



HAL
open science

A Symmetry Preserving Algorithm for Matrix Scaling

Daniel Ruiz, Bora Uçar

► **To cite this version:**

Daniel Ruiz, Bora Uçar. A Symmetry Preserving Algorithm for Matrix Scaling. [Research Report] RR-7552, 2011. inria-00569250v1

HAL Id: inria-00569250

<https://inria.hal.science/inria-00569250v1>

Submitted on 24 Feb 2011 (v1), last revised 30 Jan 2015 (v4)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

*A Symmetry Preserving Algorithm for
Matrix Scaling*

Daniel Ruiz — Bora Uçar

N° 7552

February 2011

Distributed and High Performance Computing

*R*apport
de recherche

A Symmetry Preserving Algorithm for Matrix Scaling

Daniel Ruiz^{*}, Bora Uçar[†]

Theme : Distributed and High Performance Computing
Équipe-Projet GRAAL

Rapport de recherche n° 7552 — February 2011 — 22 pages

Abstract: We present an iterative algorithm which asymptotically scales the infinity-norm of each row and each column of a matrix to one. This scaling algorithm preserves symmetry of the original matrix and shows fast linear convergence with an asymptotic rate of $1/2$. We discuss extensions of the algorithm to the one-norm, and by inference to other norms. For the one-norm case, we establish convergence results that are enjoyed by some other scaling algorithms. We demonstrate experimentally that the scaling algorithm improves the conditioning of the matrix and that it helps direct solvers by reducing the need for pivoting.

Key-words: Sparse matrices, matrix scaling, equilibration

^{*} ENSEEIHT-IRIT, 2 rue Camichel, 31071, Toulouse, France

[†] CNRS and ENS Lyon, France

Un algorithme pour mettre des matrices à l'échelle tout en préservant la symétrie

Résumé : Nous décrivons un algorithme itératif qui, asymptotiquement, met une matrice à l'échelle de telle sorte que chaque ligne et chaque colonne sont de taille 1 dans la norme infini. Cet algorithme préserve la symétrie. De plus, il converge assez rapidement avec un taux asymptotique de $1/2$. Nous discutons la généralisation de l'algorithme à la norme 1 et, par inférence, à d'autres normes. Pour le cas de la norme 1, nous établissons des résultats de convergence qui sont confirmés par certains algorithmes de mise à l'échelle. Nous démontrons expérimentalement que notre algorithme améliore le conditionnement de la matrice et qu'il aide les méthodes directes de résolution en réduisant le pivotage.

Mots-clés : Matrices creuses, mise en échelle, factorisation des matrices creuses

1 Introduction

Scaling a matrix consists in pre- and post-multiplying the original matrix by two diagonal matrices. We consider the following scaling problem: given a large, sparse matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$, find two positive diagonal matrices \mathbf{D}_1 and \mathbf{D}_2 such that all rows and columns of the scaled matrix $\hat{\mathbf{A}} = \mathbf{D}_1 \mathbf{A} \mathbf{D}_2$ have the same length in the ∞ -norm. We propose an iterative algorithm for this purpose and investigate the convergence properties of the proposed algorithm. We discuss the extension of the algorithm to the 1-norm, and describe how it can be used for other norms and multidimensional matrices.

Scaling or equilibration of data in linear systems of equations of the form $\mathbf{A}\mathbf{x} = \mathbf{b}$ is a topic of great importance. In this setting, if $\hat{\mathbf{A}}$ denotes the scaled matrix $\hat{\mathbf{A}} = \mathbf{D}_1 \mathbf{A} \mathbf{D}_2$, then one solves the equation $\hat{\mathbf{A}}\hat{\mathbf{x}} = \hat{\mathbf{b}}$, where $\hat{\mathbf{x}} = \mathbf{D}_2^{-1}\mathbf{x}$ and $\hat{\mathbf{b}} = \mathbf{D}_1\mathbf{b}$, with the expectation that the scaled system has a better conditioning and hence the results are more accurate. Assuming a scaled system also becomes helpful in designing algorithms for linear systems. For example, Bunch and Parlett [12] prove the stability of their factorization method by assuming that the rows (hence the columns) of the symmetric input matrix have ∞ -norm equal to 1.

There are several well-known algorithms (see for example [15, Section 4.12], [20, Section 3.5.2] and [26]) for scaling matrices. The well-known row and column scaling methods are the most classical scaling methods. In the row scaling method, each row in the original matrix is divided by its norm. Different norms, such as the ∞ -norm or the 1-norm, may be considered, depending on the application. The column scaling is identical to the row scaling, except that it works on the columns of the original matrix. A more elaborate scaling method is proposed in [13] and is implemented as MC29 in the HSL Mathematical Software Library (<http://www.hsl.rl.ac.uk/>). This method aims to make the nonzeros of the scaled matrix close to one by minimizing the sum of the squares of the logarithms of the moduli of the nonzeros [13]. MC29 reduces that sum in a global sense and therefore should be useful on a wide range of sparse matrices. The routine MC30 in the HSL library is a variant of MC29 for symmetric matrices. Scaling can also be combined with permutations (see [17] and the HSL routine MC64). The matrix is first permuted so that the product of the absolute values of entries on the diagonal of the permuted matrix is maximized. Then the matrix is scaled so that the diagonal entries are one and the off-diagonals are less than or equal to one. This has been shown [18] to be useful for finding a good sequence of pivots for sparse direct solvers and for building good incomplete LU preconditioners for iterative methods.

In the 60s, some optimality properties in terms of condition numbers for scaled matrices with all rows or all columns of norm of 1 has been shown [6, 7, 32]. In particular, the optimal scaling of a matrix \mathbf{A} which minimizes the condition number in the ∞ -norm is characterized by both $\mathbf{D}_1 \mathbf{A} \mathbf{D}_2$ and $\mathbf{D}_2^{-1} \mathbf{A}^{-1} \mathbf{D}_1^{-1}$ having equal row sums of absolute values [6]. Other optimality results for one sided scaling, i.e., $\mathbf{D}_1 \mathbf{A}$ or $\mathbf{A} \mathbf{D}_2$ are also shown [32], again based on the equivalence of the norms of rows or columns in the ∞ - and 1-norms.

The organization of the paper is as follows. The proposed algorithm is introduced in detail in Section 2. The convergence towards the stationary state mentioned above is at least linear, with an asymptotic rate of convergence of $\frac{1}{2}$, and this is clearly demonstrated in the same section. We also indicate some

particular properties that this algorithm yields for the scaled matrices. For example, we highlight the case of symmetric matrices since the algorithm preserves symmetry. In Section 3, we extend the algorithm to the case of other norms. Following the discussion in [22], we establish under which hypothesis the algorithm is also convergent in the case of the 1-norm, and we comment on the generalization of these results with respect to what was stated in [22]. We present numerical results in Section 4 in which we experimentally show that the proposed scaling algorithm improves the conditioning of matrices in general, that the proposed algorithm is often faster than a similar, known algorithm, and that the proposed algorithm is useful in direct linear solvers. Concluding remarks are given in Section 5.

2 The algorithm

Consider a general $m \times n$ real matrix \mathbf{A} . For $i = 1, \dots, m$, let $\mathbf{r}_i = \mathbf{a}_{i*}^T \in \mathbb{R}^{n \times 1}$ denote the i th row of \mathbf{A} as a vector, and for $j = 1, \dots, n$, let $\mathbf{c}_j = \mathbf{a}_{*j} \in \mathbb{R}^{m \times 1}$ denote the j th column of \mathbf{A} . Furthermore, let \mathbf{D}_R and \mathbf{D}_C denote the $m \times m$ and $n \times n$ diagonal matrices given by:

$$\mathbf{D}_R = \text{diag} \left(\sqrt{\|\mathbf{r}_i\|_\infty} \right)_{i=1, \dots, m} \quad \text{and} \quad \mathbf{D}_C = \text{diag} \left(\sqrt{\|\mathbf{c}_j\|_\infty} \right)_{j=1, \dots, n} \quad (1)$$

where $\|\cdot\|_\infty$ stands for the ∞ -norm of a real vector (that is the maximum entry in absolute value; sometimes called the max norm). If a row (or a column) in \mathbf{A} has all entries equal to zero, we replace the diagonal entry in \mathbf{D}_R (or \mathbf{D}_C respectively) by 1. In the following, we will assume that this does not happen, considering that such cases are fictitious in the sense that zero rows or columns should be taken away to reduce the linear system.

We then scale the matrix \mathbf{A} on both sides, forming the scaled matrix $\widehat{\mathbf{A}}$ in the following way

$$\widehat{\mathbf{A}} = \mathbf{D}_R^{-1} \mathbf{A} \mathbf{D}_C^{-1}. \quad (2)$$

Algorithm 1 Simultaneous row and column scaling in the ∞ -norm

- 1: $\widehat{\mathbf{A}}^{(0)} = \mathbf{A}$
 - 2: $\mathbf{D}_R^{(0)} = \mathbf{I}_m$
 - 3: $\mathbf{D}_C^{(0)} = \mathbf{I}_n$
 - 4: **for** $k = 0, 1, 2, \dots$ **until** convergence **do**
 - 5: $\mathbf{D}_1 = \text{diag} \left(\sqrt{\|\mathbf{r}_i^{(k)}\|_\infty} \right)_{i=1, \dots, m}$ ▶ $\mathbf{r}_i^{(k)}$ is the i th row of $\widehat{\mathbf{A}}^{(k)}$
 - 6: $\mathbf{D}_2 = \text{diag} \left(\sqrt{\|\mathbf{c}_j^{(k)}\|_\infty} \right)_{j=1, \dots, n}$ ▶ $\mathbf{c}_j^{(k)}$ is the j th column of $\widehat{\mathbf{A}}^{(k)}$
 - 7: $\widehat{\mathbf{A}}^{(k+1)} = \mathbf{D}_1^{-1} \widehat{\mathbf{A}}^{(k)} \mathbf{D}_2^{-1}$
 - 8: $\mathbf{D}_R^{(k+1)} = \mathbf{D}_R^{(k)} \mathbf{D}_1^{-1}$
 - 9: $\mathbf{D}_C^{(k+1)} = \mathbf{D}_C^{(k)} \mathbf{D}_2^{-1}$
-

The idea of the algorithm we propose is to iterate on that process, resulting in Algorithm 1. The convergence is obtained when

$$\max_{1 \leq i \leq m} \left\{ |1 - \|\mathbf{r}_i^{(k)}\|_\infty| \right\} \leq \varepsilon \quad \text{and} \quad \max_{1 \leq j \leq n} \left\{ |1 - \|\mathbf{c}_j^{(k)}\|_\infty| \right\} \leq \varepsilon \quad (3)$$

for a given value of $\varepsilon > 0$.

We note that in an actual implementation one does not need to store the iterates $\widehat{\mathbf{A}}^{(k)}$, rather one can access it through left and right multiplications, respectively, with the current scaling matrices $\mathbf{D}_R^{(k)}$ and $\mathbf{D}_C^{(k)}$.

2.1 A salient property

We highlight that the proposed iterative scaling procedure preserves the symmetry in the original matrix. In fact, this was one of our main motivations. If the given matrix \mathbf{A} is symmetric, then the diagonal matrices \mathbf{D}_R and \mathbf{D}_C in (1) are equal and, consequently, matrix $\widehat{\mathbf{A}}$ in (2) is symmetric, as is the case for the matrices $\widehat{\mathbf{A}}^{(k)}$ at any iteration in Algorithm 1. This is not the case for most scaling algorithms which alternately scale rows followed by columns or vice-versa.

In the case of unsymmetric matrices, one may consider the use of the Sinkhorn-Knopp iterations given in [29] with the ∞ -norm in place of the 1-norm. This method simply normalizes all rows and then all columns of \mathbf{A} , and iterates on this process until convergence. In ∞ -norm, the convergence is achieved after one single step. Because of its simplicity, this method is very appealing. Notice, however, that the Sinkhorn-Knopp iteration may provide very different results when applied to \mathbf{A} or to \mathbf{A}^T . As opposed to that, and this is linked to the first comment above, Algorithm 1 does provide exactly the same results when applied to \mathbf{A} or \mathbf{A}^T in the sense that the scaled matrix obtained on \mathbf{A}^T is the transpose of that obtained on \mathbf{A} . We have quoted the Sinkhorn-Knopp method [29] in particular because it has been originally proposed by the authors to obtain *doubly stochastic* matrices (that is nonnegative matrices with all rows and columns of 1-norm equal to 1), and we shall come back to this issue with respect to Algorithm 1 later.

2.2 Convergence rate

The particular case when the matrix \mathbf{A} has all its rows and columns with ∞ -norm equal to one is clearly a fixed point for the iterations in Algorithm 1. Also, if \mathbf{A} is a square matrix in which the absolute value of each diagonal element (after a permutation of columns) is greater than or equal to the absolute value of any other entry in the corresponding row and column, then it can easily be seen that the algorithm converges in one iteration, with a resulting scaled matrix $\widehat{\mathbf{A}}^{(1)}$ with all ones on the diagonal (after the same column permutation).

Concerning the rate of convergence of Algorithm 1 in the more general case, we shall now verify that the algorithm converges in all cases towards the above mentioned stationary point with an asymptotic linear rate of $\frac{1}{2}$.

The first point in the demonstration is to notice that, after the first iteration of the algorithm, all the entries in $\widehat{\mathbf{A}}^{(1)}$ are less than or equal to one in absolute value. This is very easy to see, since all entries a_{ij} in \mathbf{A} are divided by the square

roots of two numbers, $\|\mathbf{r}_i^{(k)}\|_\infty$ and $\|\mathbf{c}_j^{(k)}\|_\infty$ respectively, each one of them being greater than or equal to $|a_{ij}|$ itself.

Then, for any subsequent iteration ($k \geq 1$), if we consider the ∞ -norm of any row $\mathbf{r}_i^{(k)}$ or column $\mathbf{c}_j^{(k)}$, and if we denote by $a_{ij_0}^{(k)}$ the entry in row i for which the equality $|a_{ij_0}^{(k)}| = \|\mathbf{r}_i^{(k)}\|_\infty$ holds, and by $a_{i_0j}^{(k)}$ the entry in column j such that $|a_{i_0j}^{(k)}| = \|\mathbf{c}_j^{(k)}\|_\infty$ holds, we can easily verify that both entries $a_{ij_0}^{(k+1)}$ and $a_{i_0j}^{(k+1)}$ in the scaled matrix $\widehat{\mathbf{A}}^{(k+1)}$ are greater, in absolute value, than the square root of the corresponding value at iteration k , and are still less than one. Indeed, we can write

$$1 \geq |a_{ij_0}^{(k+1)}| = \frac{|a_{ij_0}^{(k)}|}{\sqrt{\|\mathbf{r}_i^{(k)}\|_\infty \sqrt{\|\mathbf{c}_{j_0}^{(k)}\|_\infty}}} = \frac{\sqrt{|a_{ij_0}^{(k)}|}}{\sqrt{\|\mathbf{c}_{j_0}^{(k)}\|_\infty}} \geq \sqrt{|a_{ij_0}^{(k)}|}$$

since $|a_{ij_0}^{(k)}| = \|\mathbf{r}_i^{(k)}\|_\infty$ and $\|\mathbf{c}_{j_0}^{(k)}\|_\infty \leq 1$ for any $k \geq 1$. A similar short demonstration enables us to show that

$$\sqrt{|a_{i_0j}^{(k)}|} \leq |a_{i_0j}^{(k+1)}| \leq 1,$$

for any $k \geq 1$. From this, we can finally write that the iterations in Algorithm 1 provide scaled matrices $\widehat{\mathbf{A}}^{(k)}$, $k = 1, 2, \dots$ with the following properties

$$\forall k \geq 1, 1 \leq i \leq m, \sqrt{\|\mathbf{r}_i^{(k)}\|_\infty} \leq |a_{ij_0}^{(k+1)}| \leq \|\mathbf{r}_i^{(k+1)}\|_\infty \leq 1, \quad (4)$$

and

$$\forall k \geq 1, 1 \leq j \leq n, \sqrt{\|\mathbf{c}_j^{(k)}\|_\infty} \leq |a_{i_0j}^{(k+1)}| \leq \|\mathbf{c}_j^{(k+1)}\|_\infty \leq 1, \quad (5)$$

which shows that both row and column norms must converge to 1. To conclude our demonstration, we just need to see that

$$1 - \|\mathbf{r}_i^{(k+1)}\|_\infty = \frac{1 - \|\mathbf{r}_i^{(k+1)}\|_\infty^2}{1 + \|\mathbf{r}_i^{(k+1)}\|_\infty} \leq \frac{1 - \|\mathbf{r}_i^{(k)}\|_\infty}{1 + \|\mathbf{r}_i^{(k+1)}\|_\infty},$$

and that similar equations hold for the columns as well. This completes the proof of the linear convergence of Algorithm 1 with an asymptotic rate of $\frac{1}{2}$.

2.3 Comparisons with another algorithm

As we have state before, Algorithm 1 is well suited for symmetric scaling of symmetric matrices. For the ∞ -norm case, Bunch [11] also developed an efficient symmetric scaling algorithm. Bunch's algorithm processes the rows (of the lower triangular part of the given matrix) sequentially in such a way that the largest entry seen in each row is made to be ± 1 in the scaled matrix. After this processing of the rows, the scaling found for the i th row is applied to the i th column of the whole matrix. The sequential nature of the algorithm renders it sensitive to symmetric permutations applied to the original matrix. For example, any diagonal nonzero can be symmetrically permuted to the first position so that that entry is one in the scaled matrix. On the other hand, it is easy to see that the proposed algorithm (Algorithm 1) is independent of any

permutations (not even necessarily symmetric) applied to the original matrix in the sense that the scaling matrices computed for the permuted matrix would be equal to the permuted scaling matrices computed for the original matrix. Bunch's algorithm runs in $\mathcal{O}(\text{nnz})$ -time, equivalent to one iteration of the proposed algorithm. However, Algorithm 1 is amenable to parallelism (as we have shown in [2]), whereas Bunch's algorithm is intrinsically sequential.

3 Extensions to other norms

A natural idea would be to change the norm used in Algorithm 1, and to try for instance the 2-norm or the 1-norm because of the optimal properties they induce (see [22, 32]), and still expect convergence towards an equilibrated situation with all rows and columns of length 1 in the corresponding norm. We shall see, in the remainder of this section, that this will usually, but not always, work and we investigate the potential and limitations of such extensions.

3.1 The 1-norm

The idea of equilibrating a matrix such that the 1-norm of the rows and columns are all 1 is not new, and has been the subject of constant efforts since the 1960's, and even before. Here, we briefly review some of the the previous work, and give convergence proof of Algorithm 1 for the 1-norm.

3.1.1 Background

Sinkhorn and Knopp [29] studied a method for scaling square nonnegative matrices to doubly stochastic form, that is a nonnegative matrix with all rows and columns of equal 1-norm. In [27], Sinkhorn originally showed that: *Any positive square matrix of order n is diagonally equivalent to a unique doubly stochastic matrix of order n , and the diagonal matrices which take part in the equivalence are unique up to scalar factors.* Later, a different proof for the existence part of Sinkhorn's theorem with some elementary geometric interpretations was given [9].

This result was further extended to the case of nonnegative, nonzero matrices [29]. A few definitions are necessary to state the result. A square $n \times n$ nonnegative matrix $\mathbf{A} \geq 0$ is said to have *support* if there exists a permutation σ such that $a_{i,\sigma(i)} > 0$, $1 \leq i \leq n$. Note that matrices not having support are matrices for which no full transversal can be found (see [15, page 107]), i.e., there is no column permutation making the diagonal zero-free, and are thus *structurally singular*. A matrix \mathbf{A} is said to have *total support* if every positive entry in \mathbf{A} can be permuted into a positive diagonal with a column permutation. A nonnegative nonzero square matrix \mathbf{A} of size $n > 1$ is said to be *fully indecomposable* if there does not exist permutation matrices \mathbf{P} and \mathbf{Q} such that \mathbf{PAQ} is of the form

$$\begin{pmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{O} & \mathbf{A}_{22} \end{pmatrix},$$

with \mathbf{A}_{11} and \mathbf{A}_{22} being square matrices. The term bi-irreducible is also used for fully indecomposable matrices [15, Chapter 6]. Sinkhorn and Knopp [29] established that their scaling algorithm, which simply iterates on normalizing all

rows and columns in the matrix \mathbf{A} alternately, converges to a doubly stochastic limit $\widehat{\mathbf{A}}$ if and only if the matrix \mathbf{A} has support. The doubly stochastic limit $\widehat{\mathbf{A}}$ can be represented as $\mathbf{D}_R \mathbf{A} \mathbf{D}_C$ (meaning that \mathbf{A} and $\widehat{\mathbf{A}}$ are diagonally equivalent) if and only if \mathbf{A} has total support and, if the support of \mathbf{A} is not total, then there must be a positive entry in \mathbf{A} converging to 0. Additionally, it has been shown that the diagonal matrices \mathbf{D}_R and \mathbf{D}_C , when they exist, are unique up to some scalar factors if and only if the matrix \mathbf{A} is fully indecomposable [29]. Brualdi et al. [10] independently showed the same diagonal equivalence between \mathbf{A} and a doubly stochastic matrix when \mathbf{A} is a direct sum of fully indecomposable matrices. It has been shown that a matrix \mathbf{A} has total support if and only if there exist permutation matrices \mathbf{P} and \mathbf{Q} such that $\mathbf{P} \mathbf{A} \mathbf{Q}$ is a direct sum of fully indecomposable matrices [23, Theorem 1 (ii)].

Different contributions have also been made in the study of convergence of the Sinkhorn-Knopp method under various hypothesis. Geometric convergence rate for positive matrices has been shown in [28]. Geometric convergence of the method for the nonnegative matrices with total support has been shown in [31]. In [1], the converse of the above result has been established, i.e., geometric convergence of the Sinkhorn-Knopp algorithm implies total support for a nonnegative matrix. The explicit rates of convergence for fully indecomposable matrices are given in [21].

In [22], Parlett and Landis present three iterative scaling algorithms with experimental evidence of better average and worst-case convergence behavior than the Sinkhorn-Knopp method (in at least one of the three algorithms). They also give a generalized version of the convergence theorem of Sinkhorn and Knopp [29], including a characterization of scaling algorithms that will converge to a doubly stochastic matrix when the starting matrix \mathbf{A} has support. Such algorithms are called *diagonal product increasing* (DPI) algorithms, and we shall comment on this characterization in the end of this section, since our algorithm also satisfy the DPI property. We note that the discussion in the rest of this section follows very closely that of [22], with a few slight differences that we found worth mentioning in the exposure.

3.1.2 Convergence in 1-norm

We will follow the approach in [22] to show the converge of Algorithm 1. Recall that Algorithm 1 produces a sequence of matrices diagonally equivalent to the starting matrix $\mathbf{A} = \mathbf{A}^{(0)}$ with the following iterates:

$$\begin{aligned} \mathbf{A}^{(k)} &= \left(a_{ij}^{(k)} \right) = \mathbf{D}_R^{(k)} \mathbf{A} \mathbf{D}_C^{(k)}, \quad k = 1, 2, \dots, \\ \mathbf{D}_R^{(k)} &= \text{diag} \left(d_{R_1}^{(k)}, \dots, d_{R_n}^{(k)} \right), \\ \mathbf{D}_C^{(k)} &= \text{diag} \left(d_{C_1}^{(k)}, \dots, d_{C_n}^{(k)} \right), \end{aligned} \tag{6}$$

where $\mathbf{D}_R^{(0)} = \mathbf{D}_C^{(0)} = \mathbf{I}$. For notational convenience let $r_i^{(k)}$, $i = 1, \dots, n$, and $c_j^{(k)}$, $j = 1, \dots, n$, denote the 1-norm of rows and columns respectively, thus:

$$\begin{aligned} r_i^{(k)} &= \|\mathbf{r}_i^{(k)}\|_1 = \sum_{j=1}^n |a_{ij}^{(k)}|, \\ c_j^{(k)} &= \|\mathbf{c}_j^{(k)}\|_1 = \sum_{i=1}^n |a_{ij}^{(k)}|. \end{aligned} \tag{7}$$

We also assume for simplicity that $\mathbf{A} \geq 0$, since scaling \mathbf{A} or $|\mathbf{A}|$ will be the same. Under this simplification, the 1-norm of the rows and columns reduces to the row and column sums respectively, $r_i^{(k)} = \sum_{j=1}^n a_{ij}^{(k)}$ and $c_j^{(k)} = \sum_{i=1}^n a_{ij}^{(k)}$, and to generalize our results to any matrix, one just needs to extend the definition of a doubly stochastic matrix so that the absolute value of the matrix under consideration is doubly stochastic in the usual sense.

Theorem 3.1 *Given the sequence (6) of diagonal equivalents for \mathbf{A} , in which*

$$a_{ij}^{(k+1)} = \frac{a_{ij}^{(k)}}{\sqrt{r_i^{(k)}} \sqrt{c_j^{(k)}}}, \quad 1 \leq i, j \leq n,$$

with $r_i^{(k)}$ and $c_j^{(k)}$ given by (7):

1. If \mathbf{A} has support, then $\mathbf{S} = \lim_{k \rightarrow \infty} \mathbf{A}^{(k)}$ exists and is doubly stochastic.
2. If \mathbf{A} has total support, then both $\mathbf{D}_R = \lim_{k \rightarrow \infty} \mathbf{D}_R^{(k)}$ and $\mathbf{D}_C = \lim_{k \rightarrow \infty} \mathbf{D}_C^{(k)}$ exist and $\mathbf{S} = \mathbf{D}_R \mathbf{A} \mathbf{D}_C$.

Proof of point (1) in Theorem 3.1: We recall the arithmetic-geometric mean inequality. It states that if $x_i \geq 0$ for $i = 1, \dots, n$ then

$$\prod_{i=1}^n x_i \leq \left(\sum_{i=1}^n \frac{x_i}{n} \right)^n, \quad (8)$$

with equality holding if and only if $x_1 = x_2 = \dots = x_n$.

Now, using (8), we can write the following for all k

$$\prod_{i=1}^n r_i^{(k+1)} \leq \left(\frac{1}{n} \sum_{i=1}^n r_i^{(k+1)} \right)^n = \left(\frac{1}{n} \sum_{1 \leq i, j \leq n} \frac{a_{ij}^{(k)}}{\sqrt{r_i^{(k)}} \sqrt{c_j^{(k)}}} \right)^n,$$

with the same inequality for $\prod_{j=1}^n c_j^{(k+1)}$ since

$$\sum_{i=1}^n r_i^{(k+1)} = \sum_{j=1}^n c_j^{(k+1)} = \sum_{1 \leq i, j \leq n} a_{ij}^{(k+1)}.$$

Additionally, using the Cauchy-Schwarz inequality on the dot-product of the two n^2 -vectors $\mathbf{v} = \left(\sqrt{a_{ij}^{(k)}} / \sqrt{r_i^{(k)}} \right)_{1 \leq i, j \leq n}$ and $\mathbf{w} = \left(\sqrt{a_{ij}^{(k)}} / \sqrt{c_j^{(k)}} \right)_{1 \leq i, j \leq n}$, we can write that

$$\sum_{1 \leq i, j \leq n} \frac{a_{ij}^{(k)}}{\sqrt{r_i^{(k)}} \sqrt{c_j^{(k)}}} \leq \sqrt{\sum_{1 \leq i, j \leq n} \frac{a_{ij}^{(k)}}{r_i^{(k)}}} \sqrt{\sum_{1 \leq i, j \leq n} \frac{a_{ij}^{(k)}}{c_j^{(k)}}} = \sqrt{n} \sqrt{n},$$

and thus, for all $k \geq 0$, we have

$$\prod_{i=1}^n r_i^{(k+1)} \leq \left(\frac{1}{n} \sum_{i=1}^n r_i^{(k+1)} \right)^n \leq 1 \quad \text{and} \quad \prod_{j=1}^n c_j^{(k+1)} \leq \left(\frac{1}{n} \sum_{j=1}^n c_j^{(k+1)} \right)^n \leq 1. \quad (9)$$

Finally, if we introduce

$$s_k = \prod_{i=1}^n d_{R_i}^{(k)} d_{C_i}^{(k)}, \quad k = 1, 2, \dots$$

we easily see that

$$\frac{s_k}{s_{k+1}} = \prod_{i=1}^n \sqrt{r_i^{(k)} c_i^{(k)}} \leq 1, \quad (10)$$

which shows that the sequence $(s_k)_{k=1,2,\dots}$ is monotonically increasing (unless all row and column sums are equal).

Since \mathbf{A} has support, there exists a permutation σ such that $a_{i,\sigma(i)} > 0$, $1 \leq i \leq n$. Let $a = \min_{1 \leq i \leq n} (a_{i,\sigma(i)})$. Then, for all $k \geq 1$,

$$\sum_{i=1}^n d_{R_i}^{(k)} d_{C_{\sigma(i)}}^{(k)} a \leq \sum_{i=1}^n d_{R_i}^{(k)} d_{C_{\sigma(i)}}^{(k)} a_{i,\sigma(i)} = \sum_{i=1}^n a_{i,\sigma(i)}^{(k)} \leq n,$$

the last inequality being simply due to the fact that, after the first iteration, all the entries in matrix $\mathbf{A}^{(k)}$ are less than or equal to 1. Then, by the arithmetic-geometric mean inequality (8),

$$s_k = \prod_{i=1}^n d_{R_i}^{(k)} d_{C_i}^{(k)} \leq a^{-n}, \quad k = 1, 2, \dots$$

and the monotonically increasing sequence $(s_k)_{k=1,2,\dots}$ is bounded. Therefore

$$\lim_{k \rightarrow \infty} s_k = L > 0$$

exists, and

$$\lim_{k \rightarrow \infty} \frac{s_k}{s_{k+1}} = 1. \quad (11)$$

Using (8) and (9), we can write

$$\frac{s_k}{s_{k+1}} = \prod_{i=1}^n \sqrt{r_i^{(k)} c_i^{(k)}} \leq \left\{ \sum_{i=1}^n \frac{1}{2n} (r_i^{(k)} + c_i^{(k)}) \right\}^n \leq 1,$$

and because of (11) we can conclude that

$$\lim_{k \rightarrow \infty} \prod_{i=1}^n r_i^{(k)} c_i^{(k)} = 1 \quad \text{and} \quad \lim_{k \rightarrow \infty} \sum_{i=1}^n \frac{1}{2n} (r_i^{(k)} + c_i^{(k)}) = 1. \quad (12)$$

Now, since all the elements in $\mathbf{A}^{(k)}$ are less than 1 after the first iteration, we know that each of the two sequences $(r_i^{(k)})_{k=1,2,\dots}$ and $(c_j^{(k)})_{k=1,2,\dots}$, for all $1 \leq i, j \leq n$, are bounded (note that this is also implied by (9)). Let us introduce the sequence $(\mathbf{v}^{(k)})_{k=1,2,\dots}$ of the $2n$ -vectors

$$\mathbf{v}^{(k)} = (r_1^{(k)}, \dots, r_n^{(k)}, c_1^{(k)}, \dots, c_n^{(k)}),$$

which is also bounded in \mathbb{R}^{2n} of finite dimension. Consider then any convergent subsequence $(\mathbf{v}^{(\hat{k})})_{\hat{k}}$ and let

$$x_i = \lim_{\hat{k} \rightarrow \infty} r_i^{(\hat{k})}, \quad 1 \leq i \leq n,$$

and

$$y_j = \lim_{\hat{k} \rightarrow \infty} c_j^{(\hat{k})}, \quad 1 \leq j \leq n.$$

From (12), we can write

$$\prod_{i=1}^n x_i y_i = \left\{ \sum_{i=1}^n \frac{1}{2n} (x_i + y_i) \right\}^{2n} = 1,$$

and since the arithmetic-geometric mean equality only holds if all the elements are equal, we easily see that $x_1 = \dots = x_n = 1 = y_1 = \dots = y_n$. Therefore any convergent subsequence of the bounded sequence $(\mathbf{v}^{(k)})_{k=1,2,\dots}$ in finite dimensional space must have the same limit (made with all ones), which implies that the sequence $(\mathbf{v}^{(k)})_{k=1,2,\dots}$ is necessarily convergent and that

$$\lim_{k \rightarrow \infty} r_i^{(k)} = 1 \quad \text{and} \quad \lim_{k \rightarrow \infty} c_j^{(k)} = 1, \quad 1 \leq i, j \leq n. \quad (13)$$

Again, since all $a_{ij}^{(k)}$ are less than 1 after the first iteration, the sequence of matrices $(\mathbf{A}^{(k)})_{k=1,2,\dots}$ is also bounded in the finite dimensional space of real, $n \times n$ matrices. Let us consider any convergent subsequence $(\mathbf{A}^{(\hat{k})})_{\hat{k}}$ of $(\mathbf{A}^{(k)})_{k=1,2,\dots}$, and define

$$\mathbf{S} = \lim_{\hat{k} \rightarrow \infty} \mathbf{A}^{(\hat{k})}.$$

Then, paraphrasing the results in [22] for the proof of their Corollary 1, we can state the following: because of (13), \mathbf{S} is doubly stochastic and, since the set of $n \times n$ doubly stochastic matrices is the convex hull of the set of $n \times n$ permutation matrices (see [8]), \mathbf{S} must thus have total support. Therefore, $s_{ij} = \lim_{\hat{k} \rightarrow \infty} a_{ij}^{(\hat{k})} = 0$ whenever $a_{ij} > 0$ cannot be permuted onto a positive diagonal. Now, consider the matrix $\widehat{\mathbf{A}}$ in which all entries $a_{ij} > 0$ in \mathbf{A} that cannot be permuted onto a positive diagonal have been set to zero, the others remaining the same. From the previous remark, it is clear that $\mathbf{S} = \lim_{\hat{k} \rightarrow \infty} \mathbf{A}^{(\hat{k})}$ is also equal to

$$\mathbf{S} = \lim_{\hat{k} \rightarrow \infty} \mathbf{D}_R^{(\hat{k})} \widehat{\mathbf{A}} \mathbf{D}_C^{(\hat{k})}.$$

Additionally, from the proof in [22, Theorem 1, pages 67–68, proof of point (2)], which simply exploits properties (8), (9), (10), and the fact that any nonzero element in $\widehat{\mathbf{A}}$ can be permuted into a zero-free diagonal (as by construction $\widehat{\mathbf{A}}$ has total support), we know that there exists a strictly positive constant γ such that, for each nonzero entry a_{ij} in $\widehat{\mathbf{A}}$, we have:

$$\forall k \geq 1, \quad d_{R_i}^{(k)} d_{C_j}^{(k)} \geq \gamma > 0. \quad (14)$$

We mention, however, a very little difference in (9) and the hypothesis exploited by Parlett and Landis, which comes from the fact that they were considering “scaled algorithms” in general, thus imposing the property

$$(P3) \quad \mu_k = \frac{1}{n} \sum_{i=1}^n r_i^{(k+1)} = 1,$$

which in our case holds just as an inequality in (9). However, since they only use property (P3) as an upper bound in some parts of their demonstration, the inequality (9) is sufficient and all conclusions remain the same. Note also that the result (14) is valid for the entire sequence and not just the subsequence indexed by \hat{k} . Consequently, for each nonzero entry a_{ij} in $\widehat{\mathbf{A}}$, we can write:

$$\lim_{\hat{k} \rightarrow \infty} d_{R_i}^{(\hat{k})} d_{C_j}^{(\hat{k})} = \frac{s_{ij}}{a_{ij}} > 0.$$

Then, applying Lemma 2 of [22] (which is itself paraphrased from [29, page 345]), we know that there exist positive sequences $(x_i^{(\hat{k})})_{\hat{k}}$ and $(y_i^{(\hat{k})})_{\hat{k}}$ with positive limits such that

$$d_{R_i}^{(\hat{k})} d_{C_j}^{(\hat{k})} = x_i^{(\hat{k})} y_j^{(\hat{k})}, \quad \forall a_{ij} > 0 \text{ in } \widehat{\mathbf{A}}, \text{ and } \forall \hat{k}.$$

Then, taking

$$\begin{aligned} \mathbf{X}^{(\hat{k})} &= \text{diag} \left(x_1^{(\hat{k})}, \dots, x_n^{(\hat{k})} \right), \\ \mathbf{Y}^{(\hat{k})} &= \text{diag} \left(y_1^{(\hat{k})}, \dots, y_n^{(\hat{k})} \right), \\ \mathbf{X} &= \lim_{\hat{k} \rightarrow \infty} \mathbf{X}^{(\hat{k})} \text{ and } \mathbf{Y} = \lim_{\hat{k} \rightarrow \infty} \mathbf{Y}^{(\hat{k})}, \end{aligned}$$

we have

$$\mathbf{S} = \lim_{\hat{k} \rightarrow \infty} \mathbf{D}_R^{(\hat{k})} \widehat{\mathbf{A}} \mathbf{D}_C^{(\hat{k})} = \lim_{\hat{k} \rightarrow \infty} \mathbf{X}^{(\hat{k})} \widehat{\mathbf{A}} \mathbf{Y}^{(\hat{k})} = \mathbf{X} \widehat{\mathbf{A}} \mathbf{Y},$$

showing that $\widehat{\mathbf{A}}$ is diagonally equivalent to \mathbf{S} . Now, consider any other convergent subsequence in $(\mathbf{A}^{(k)})_{k=1,2,\dots}$. For the same reasons as above, its limit will also be doubly stochastic and diagonally equivalent to $\widehat{\mathbf{A}}$, and since doubly stochastic equivalents are unique (see [30]), the two limits must be the same. Therefore, we can conclude that $\lim_{k \rightarrow \infty} \mathbf{A}^{(k)}$ exists and is doubly stochastic, which completes the proof of point (1). ■

Proof of point (2) in Theorem 3.1: As a direct consequence of the demonstration of point (1), we can state that, if \mathbf{A} has total support, then $\mathbf{S} = \lim_{k \rightarrow \infty} \mathbf{A}^{(k)}$ is diagonally equivalent to \mathbf{A} , since in this case $\mathbf{A} = \widehat{\mathbf{A}}$ in the previous discussion.

To prove that, under the same hypothesis, both $\mathbf{D}_R = \lim_{k \rightarrow \infty} \mathbf{D}_R^{(k)}$ and $\mathbf{D}_C = \lim_{k \rightarrow \infty} \mathbf{D}_C^{(k)}$ exist and $\mathbf{S} = \mathbf{D}_R \mathbf{A} \mathbf{D}_C$, we first make the assumption that \mathbf{A} not only has total support but is also fully indecomposable. In this case, \mathbf{A} is not only diagonally equivalent to the doubly stochastic limit $\mathbf{S} = \lim_{k \rightarrow \infty} \mathbf{A}^{(k)}$, but we also know from [29] that the diagonal matrices which take place in this equivalence are unique up to a scalar factor.

Without loss of generality, we can assume that the matrix \mathbf{A} has a full diagonal, since \mathbf{A} has support and Algorithm 1 is independent of any permutations

on \mathbf{A} . Let us suppose now that one of the sequences $\left(d_{R_i}^{(k)}\right)_{k=1,2,\dots}$ is unbounded for some i —the same could be done with one of the $\left(d_{C_j}^{(k)}\right)_{k=1,2,\dots}$. In such a case, there exist a subsequence $\left(d_i^{(\hat{k})}\right)_{\hat{k}}$ such that

$$\lim_{\hat{k} \rightarrow \infty} \left(d_i^{(\hat{k})}\right)_{\hat{k}} = +\infty .$$

As the matrix is fully indecomposable, for any index j , $1 \leq j \leq n$, there exist a subset of nonzero entries in \mathbf{A}

$$a_{ij_1}, a_{j_1 j_2}, \dots, a_{j_{q-1} j_q}, a_{j_q j}$$

“connecting” index i to index j . Consequently, because of (14), we can show by using each of the nonzero intermediate entries $a_{j_p j_{p+1}}$ in the subset above as well as each of the corresponding nonzero diagonal elements, that there exist $\alpha_{ij} > 0$ and $\beta_{ij} > 0$ such that

$$\lim_{\hat{k} \rightarrow \infty} \frac{d_{R_j}^{(\hat{k})}}{d_{R_i}^{(\hat{k})}} = \alpha_{ij} \quad \text{and} \quad \lim_{\hat{k} \rightarrow \infty} d_{R_i}^{(\hat{k})} d_{C_j}^{(\hat{k})} = \beta_{ij} .$$

Since this can be done for any j , we can conclude that the subsequence $\left(\prod_{j=1}^n d_{R_j}^{(\hat{k})}\right)_{\hat{k}}$ goes to infinity as $\left(d_{R_i}^{(\hat{k})}\right)^n \prod_{j=1}^n \alpha_{ij}$ and that the subsequence $\left(\prod_{j=1}^n d_{C_j}^{(\hat{k})}\right)_{\hat{k}}$ (which is strictly positive) goes to zero as $\left(d_{R_i}^{(\hat{k})}\right)^{-n} \prod_{j=1}^n \beta_{ij}$. The last conclusion is in contradiction with (9) which shows that both sequences $\left(\prod_{j=1}^n d_{R_j}^{(k)}\right)_{k=1,2,\dots}$ and $\left(\prod_{j=1}^n d_{C_j}^{(k)}\right)_{k=1,2,\dots}$ are monotonically increasing. Consequently, each of the sequences $\left(d_{R_i}^{(k)}\right)_{k=1,2,\dots}$ and $\left(d_{C_i}^{(k)}\right)_{k=1,2,\dots}$ are bounded.

Now, since the two sequences $(\mathbf{D}_R^{(k)})_{k=1,2,\dots}$ and $(\mathbf{D}_C^{(k)})_{k=1,2,\dots}$ are bounded in the finite dimensional space of real, $n \times n$, diagonal matrices, they have convergent subsequences. Let us consider two convergent subsequences:

$$\left(\mathbf{D}_R^{(\hat{k})}, \mathbf{D}_C^{(\hat{k})}\right) \xrightarrow{\hat{k} \rightarrow +\infty} (\mathbf{D}_1, \mathbf{E}_1) ,$$

and

$$\left(\mathbf{D}_R^{(\tilde{k})}, \mathbf{D}_C^{(\tilde{k})}\right) \xrightarrow{\tilde{k} \rightarrow +\infty} (\mathbf{D}_2, \mathbf{E}_2) .$$

Obviously, $\mathbf{D}_1 \mathbf{A} \mathbf{E}_1 = \mathbf{S} = \mathbf{D}_2 \mathbf{A} \mathbf{E}_2$, and thus, because of the uniqueness (shown in [29]), there exists $\alpha > 0$ such that $\mathbf{D}_1 = \alpha \mathbf{D}_2$ and $\mathbf{E}_1 = (1/\alpha) \mathbf{E}_2$. Then, as mentioned above, since both sequences $\left(\prod_{p=1}^n d_{R_p}^{(k)}\right)_{k=1,2,\dots}$ and $\left(\prod_{p=1}^n d_{C_p}^{(k)}\right)_{k=1,2,\dots}$ are monotonically increasing, it is clear that α must be equal to 1 and the two limits must be equal. Therefore, we can conclude that $\mathbf{D}_R = \lim_{k \rightarrow \infty} \mathbf{D}_R^{(k)}$ and $\mathbf{D}_C = \lim_{k \rightarrow \infty} \mathbf{D}_C^{(k)}$ exist and point (2) in Theorem 3.1 holds for fully indecomposable matrices.

Finally, we know from [23] that matrices with a total support can be permuted into a direct sum of fully indecomposable matrices. Now, since Algorithm 1 is independent of any permutation, we get the result from what precedes by applying it independently to each fully indecomposable matrix in such a direct sum. This completes the proof of point (2). ■

We would like to recall some of the results contained in [22], which can also be collected from the proof of Theorem 3.1 above.

Corollary 3.2

1. If \mathbf{A} is diagonally equivalent to a doubly stochastic matrix, \mathbf{S} , then

$$\mathbf{S} = \lim_{k \rightarrow \infty} \mathbf{A}^{(k)}.$$

2. If \mathbf{A} has support and is not diagonally equivalent to a doubly stochastic matrix, then for each pair of indices (i, j) such that a_{ij} cannot be permuted onto a positive diagonal by a column permutation,

$$\lim_{k \rightarrow \infty} a_{ij}^{(k)} = 0.$$

Parlett and Landis [22], in their Theorem 1, additionally gave some characterizations for iterative scaling algorithms that would converge to a doubly stochastic matrix when the matrix \mathbf{A} has support. They introduced three properties (P1), (P2), and (P3) which, when they hold for the iterates (6), enable one to prove convergence of the sequence $(\mathbf{A}^{(k)})_{k=1,2,\dots}$ to a doubly stochastic limit. Property (P1) is simply equation (10), or equivalently that the sequence $(s_k)_{k=1,2,\dots}$ is monotonically increasing. Property (P2) implies that, under the hypothesis (11), both (13) and

$$\lim_{k \rightarrow \infty} d_{R_i}^{(k+1)} / d_{R_i}^{(k)} = \lim_{k \rightarrow \infty} d_{C_j}^{(k+1)} / d_{C_j}^{(k)} = 1, \quad 1 \leq i, j \leq n, \quad (15)$$

hold. Property (P3) is simply to consider that the algorithm applies a normalization with

$$(P3) \quad \mu_k = \frac{1}{n} \sum_{i=1}^n r_i^{(k+1)} = 1$$

at each iteration. In our case, this only holds as an inequality and, as we already mentioned, it is sufficient to complete the proof. It is important to note also that we did not need to establish the second set of conclusions (15) contained in (P2) to complete the proof of Theorem 3.1. This was the main reason why we wanted to re-derive the proof of the convergence which, as one may easily see, can be extended to any algorithm that satisfies (P1), the first part of (P2), and just an inequality in (P3). Strictly speaking, (P3) is not even necessary, and it is only sufficient to prove that all the elements $a_{ij}^{(k)}$ are bounded. However, the inequality in (P3) was necessary in our case to prove (13), and even if (P2) is a consequence of (P3), it has to be shown by some means.

Finally, the proof that each of the sequences $(d_{R_i}^{(k)})_{k=1,2,\dots}$ and $(d_{C_j}^{(k)})_{k=1,2,\dots}$ converges under the hypothesis that \mathbf{A} has a total support is not contained in [22, Theorem 1]. This convergence result relies on the added property induced by (9), which we call (P4). The property (P4) says that both sequences

$\left(\prod_{p=1}^n d_{R_p}^{(k)}\right)_{k=1,2,\dots}$ and $\left(\prod_{p=1}^n d_{C_p}^{(k)}\right)_{k=1,2,\dots}$ are monotonically increasing. Notice that this property also implies (P1). The convergence result also relies on the property, which we call (P5), that the iterates (6) are permutation independent—otherwise these results only hold for fully indecomposable matrices. Now, looking carefully at the convergence proof above, it is clear that we can extend the results in Theorem 3.1 to all sequences (6) of diagonal equivalents for \mathbf{A} satisfying the first part of (P2), the inequality in (P3), and (P4)+(P5) instead of (P1).

We have one last comment for the symmetric case, considering the fact that Algorithm 1 preserves symmetry.

Corollary 3.3 1. *If \mathbf{A} is symmetric and has support, then Algorithm 1 in the 1-norm builds a sequence of symmetric scalings of \mathbf{A} converging to a symmetric doubly stochastic limit.*

2. *If \mathbf{A} is symmetric and has total support, then \mathbf{A} is symmetrically equivalent to a symmetric doubly stochastic matrix \mathbf{S} , and Algorithm 1 in the 1-norm builds a convergent sequence of diagonal matrices $\mathbf{D}^{(k)}$ such that $\mathbf{S} = \lim_{k \rightarrow \infty} \mathbf{D}^{(k)} \mathbf{A} \mathbf{D}^{(k)}$.*

3.2 Other norms and multidimensional matrices

Rothblum et al. have shown [25, page 13] that the problem of scaling a matrix \mathbf{A} in the ℓ_p -norm, for $1 < p < \infty$ can be reduced to the problem of scaling the p th Hadamard power of \mathbf{A} , i.e., the matrix $\mathbf{A}^{[p]} = [a_{ij}^p]$, in the 1-norm. We apply that discussion to Algorithm 1 by replacing the matrix \mathbf{A} with $\mathbf{A}^{[p]}$ and then by taking the Hadamard p th root, e.g., $\mathbf{D}_1^{[1/p]} = [d_{ii}^{1/p}]$, of the resulting iterates. Hence, we conclude that all of the convergence results shown in the previous section for the 1-norm hold for any of the ℓ_p norms for $1 < p < \infty$.

Extensions of Algorithm 1 to the scaling of multidimensional matrices (see for example [5, 19, 24]) can also be done in a very simple manner. Consider for example a three dimensional nonnegative matrix $\mathbf{A} = (a_{ijk})$, and suppose that its one-dimensional marginals $x_i = \sum_j \sum_k a_{ijk}$, $y_j = \sum_i \sum_k a_{ijk}$, and $z_k = \sum_i \sum_j a_{ijk}$ are all positive, then the idea is to scale \mathbf{A} to $\hat{\mathbf{A}}$ by setting

$$\hat{a}_{ijk} = \frac{a_{ijk}}{\sqrt[3]{x_i} \sqrt[3]{y_j} \sqrt[3]{z_k}}$$

for all i, j, k , and to iterate on that. Generalization to p -dimensional matrices can be done with the p th root of each of the p one-dimensional marginals. We must mention, however, that we did not investigate the extension to multidimensional matrices, and we cannot state whether they are convergent or not.

4 Numerical experiments

We have implemented the proposed algorithm in MATLAB, and compared it against our implementation of Bunch's and Sinkhorn-Knopp algorithms. We have experimented with a set of matrices obtained from the University of Florida Sparse Matrix Collection (UFL) available at <http://www.cise.ufl.edu/research/>

Table 1: The numbers of instances in which the 2-norm condition number (estimated) of the scaled matrix is larger than (the first half of the table) and smaller than (the second half) the condition number (estimated) of the original matrix by differing amounts, for the symmetric matrices in the data set.

ratio	B	A-inf	A-1	A-2
$\geq 1.0e+3$	1	0	0	0
$\geq 1.0e+2$	2	0	1	0
$\geq 1.0e+1$	10	2	1	3
$> 1.0e+0$	41	36	23	29
$< 1.0e-3$	34	37	42	44
$\leq 1.0e-4$	25	27	33	36
$\leq 1.0e-5$	15	15	21	21
$\leq 1.0e-6$	14	14	16	17

[sparse/matrices/](#). We present two different sets of experiments. In the first one, we compare different scaling methods. In the second one, we investigate the merits of using the proposed scaling method in solving linear systems with direct methods.

We have compared the proposed scaling method with a few others on a set of matrices from UFL collection. The matrices in our data set satisfy the following properties: real, symmetric, $1000 \leq n < \text{nnz} \leq 1500000$, without explicit zeros, with total support. There were a total of 256 such matrices at the time of experimentation. In 10 out of those 256 matrices, MATLAB's `condest` function failed to compute an estimated condition number (due to insufficient memory). Therefore, we give results on 246 matrices; among which 172 are symmetric and 74 are unsymmetric.

First, we comment on the condition number estimates of the scaled matrices. For a given scaling algorithm, we compute the scaled matrix and run MATLAB's `condest` function on it. We used the performance profiles discussed in [14] to generate the plot in Fig. 1. The plot contains condition estimates after the application of four different scaling algorithms and those of the original matrices. For a given τ , the plot shows the probability for a scaling algorithm that the condition estimate due to this algorithm is within τ times the best (among all 5 condition estimates). Therefore, the higher the probability the more preferable the scaling method is. As seen in the curves, the estimated condition number is almost always improved after the scaling algorithms. We provide Table 1 for details. As seen from the table, all of the scaling algorithms improve the condition number in most of the instances; in the remaining ones, for the Bunch's algorithm the highest increase is $1.12e+4$ times the original condition estimate, for the other algorithms the highest increases are less than 450. As seen in the plots, 1-norm and 2-norm scalings are more likely to result in the better condition number (estimated). Therefore, we suggest the use of 1- or 2-norm scalings, for obtaining better (estimated) condition numbers.

We present the number of iterations for convergence with a tolerance of $1.0e-6$, that is $\varepsilon = 1.0e-6$ in (3), for 1-, 2-, and ∞ -norms. The algorithm converges very fast for the ∞ -norm. The maximum number of iterations with $\varepsilon = 1.0e-6$ is observed to be 26 in our data set. Figure 2 shows the histogram of the number

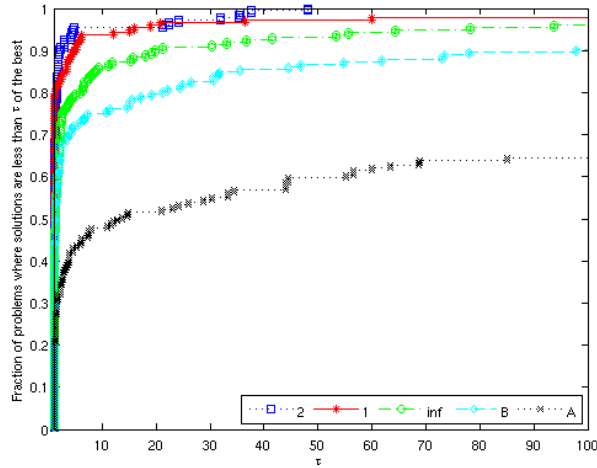


Figure 1: Performance profiles for the condition number estimates with limit 100, for the symmetric data set. A marks the condition number estimate of the original matrix; B marks that of Bunch’s algorithm; *inf*, 1, and 2 mark that of the proposed algorithm with ∞ -, 1-, and 2-norms. To follow the curves, we note that at $\tau = 40$, the order of curves correspond to, from top to bottom, the order in the legend.

of iterations for the ∞ -norm. The number of iterations with the 1- and 2-norms are close to each other. The algorithm did not converge for 10 matrices in 5000 iterations (maximum error was $2.07e-4$) with the 1-norm. With the 2-norm, the algorithm did not converge for 17 matrices, 10 of them being common with the 1-norm, in 5000 iterations (maximum error was $3.08e-4$). The average numbers of iterations for the converged matrices are 206 and 257 for the 1- and 2-norms, respectively. It is worth noting that those matrices that cause trouble come from two matrix sets *GHS_indef* (represented with 33 matrices in our data set) and *Schenk_IBMNA* (represented with 27 matrices in our data set). Excluding these two sets, the algorithm converges in 26 and 29 iterations with the 1- and 2-norms, respectively. We have also performed experiments with Sinkhorn-Knopp algorithm both in 1- and 2-norms. We have seen that for the same set of matrices (the symmetric matrices in our data set), the algorithm did not converge in 5000 iterations for 75 matrices in the 1-norm and for 101 matrices in the 2-norm. This evidently shows faster convergence with the proposed algorithm than the Sinkhorn-Knopp algorithm for doubly stochastic scaling; we, however, did not investigate this outcome theoretically. We note however that for 74 unsymmetric matrices in our data set, the proposed algorithm converged to $1.0e-6$ in less than 5000 iterations for 28 and 24 matrices in, respectively, 1- and 2-norms, whereas the Sinkhorn-Knopp algorithm converged for 31 and 20 matrices in, respectively, 1- and 2-norms.

We now comment on the use of the proposed scaling algorithms in the solution of linear systems with direct methods. We used a large set of square matrices from UFL each satisfying the following assertions: $1000 \leq n \leq 121000$,

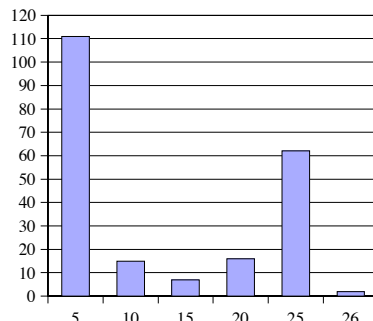


Figure 2: Histogram of the number of iterations with $\varepsilon = 1.0e-6$ for the ∞ -norm

also $n \leq \text{nnz} \leq 1790000$, real, with full structural rank, not binary nor a matrix with nonzeros only in $\{-1, 1\}$. There were a total of 260 symmetric matrices and 455 unsymmetric matrices with these properties at the time of experimentation. We use the solver MUMPS [3, 4] version 4.8.4 to test the effects of the proposed scaling method. We use MUMPS with default parameter settings except for the following. For the unsymmetric matrices if there are zeros in the diagonal, we find a maximum traversal using `dmpperm` of MATLAB and permute the matrix to put that transversal into the diagonal, before calling MUMPS and set the related parameter `ICNTL(6)` to 0 in MUMPS. For the symmetric matrices, we set the parameter `ICNTL(12)` to 2 to use the compressed ordering strategies discussed in [16]. We set `ICNTL(14)` to 80 (a memory relaxation parameter that permits the use of 80% more memory than estimated) and deemed the factorizations that pass this limit unsuccessful. The necessity of more memory than the estimated amount arises when pivoting is performed for the numerical stability. A scaling strategy deemed to be better, if it reduces pivoting.

We have investigated different strategies that use the proposed scaling algorithm in order to be able to suggest a solid scaling strategy. These scaling strategies we have investigated can be specified with three integers $[i_1, i_2, i_3]$. The scaling strategy will proceed in three phases: in phase 1, it will apply i_1 many steps of ∞ -norm scaling; then in phase 2, it will apply i_2 many steps of 1-norm scaling; and finally in phase 3, it will apply i_3 many steps of ∞ -norm scaling. It is understood that each phase continues using the scaling factors found so far, and if convergence occurs within a phase then the next phase is started. The results are shown in Table 2. As seen from these results, the proposed scaling method helps MUMPS perform less pivoting and thus the memory estimates are closer to the actual ones, when the scaling is used. In general, the higher the number of 1-norm iterations, the more robust the estimates are. In other words, by increasing the number of 1-norm iterations, one can expect to have more robust solver performance. We note that the current version of MUMPS uses $[1, 3, 0]$ as the default scaling strategy.

5 Conclusion

We presented an iterative algorithm which scales the ∞ -norm of the rows and columns of a matrix to 1. The important features of the proposed algorithm are

Table 2: The effects of different scaling strategies on the direct solver MUMPS. The results are the averages for 260 matrices in the symmetric case and 455 matrices for the unsymmetric case. A scaling strategy is said to be unsuccessful, if MUMPS cannot run to completion with the settings described in the text. The minimum and average ratios of estimated and actual memory requirements are given; the closer this ratio to 1.0, the better the estimates.

scaling strategy	symmetric matrices				unsymmetric matrices			
	number of unsuccessful runs	memory reqs est/actual		number of unsuccessful runs	memory reqs est/actual			
		min	avg		min	avg		
[0, 0, 0]	24	0.510	0.852	22	0.552	0.858		
[0, 3, 0]	10	0.554	0.924	10	0.579	0.904		
[1, 3, 0]	7	0.555	0.927	8	0.592	0.907		
[1, 3, 1]	8	0.557	0.928	9	0.604	0.905		
[1, 3, 3]	7	0.559	0.926	8	0.573	0.906		
[0, 10, 0]	7	0.564	0.937	6	0.567	0.919		
[1, 10, 0]	8	0.569	0.935	2	0.572	0.922		
[1, 10, 1]	7	0.580	0.937	3	0.585	0.920		
[0, 30, 0]	7	0.598	0.944	1	0.668	0.948		
[1, 30, 0]	6	0.604	0.944	1	0.634	0.947		
[1, 30, 1]	6	0.620	0.946	1	0.639	0.948		
[1, 100, 0]	6	0.754	0.949	2	0.744	0.951		

the following: it preserves symmetry; it is permutation independent; it has fast linear convergence with an asymptotic rate of $1/2$. We discussed the extension to the 1-norm in detail and established convergence results that are enjoyed by some well-known algorithms. By following existing arguments, we addressed scaling in ℓ_p -norms, for $1 < p < \infty$ and scaling of multidimensional matrices. We have demonstrated numerical experiments in which the proposed algorithm was shown to be helpful for a direct solver in the sense that it reduces the need for numerical pivoting. In particular we have shown that one step of ∞ -norm scaling followed by a few steps of 1-norm scaling is usually good enough, where for harder problems increasing the number of 1-norm iterations seems to be a viable way. We have also experimentally demonstrated that the proposed algorithm usually converges faster than a known alternative.

Acknowledgements and availability

We thank Patrick R. Amestoy, Iain S. Duff, Nick Gould, and Mario Arioli for commenting on an earlier draft of the article. The MATLAB implementations of all scaling algorithms discussed in this paper and an MPI-based parallel implementation of the proposed scaling algorithm can be obtained from the web pages of authors.

References

- [1] Eva Achilles. Implications of convergence rates in Sinkhorn balancing. *Linear Algebra and its Applications*, 187:109–112, July 1993.
- [2] P. R. Amestoy, I. S. Duff, D. Ruiz, and B. Uçar. A parallel matrix scaling algorithm. In J. M. Palma, P. R. Amestoy, M. Daydé, M. Mattoso, and J. C. Lopes, editors, *High Performance Computing for Computational Science - VECPAR 2008: 8th International Conference*, volume 5336 of *Lecture Notes in Computer Science*, pages 301–313. Springer Berlin / Heidelberg, 2008.
- [3] Patrick R. Amestoy, Iain S. Duff, Jean-Yves L'Excellent, and Jacko Koster. A fully asynchronous multifrontal solver using distributed dynamic scheduling. *SIAM Journal on Matrix Analysis and Applications*, 23(1):15–41, 2001.
- [4] Patrick R. Amestoy, Abdou Guermouche, Jean-Yves L'Excellent, and Stéphane Pralet. Hybrid scheduling for the parallel solution of linear systems. *Parallel Computing*, 32(2):136–156, 2006.
- [5] R. Bapat. D_1AD_2 theorems for multidimensional matrices. *Linear Algebra and its Applications*, 48:437–442, 1982.
- [6] F. L. Bauer. Optimally scaled matrices. *Numerische Mathematik*, 5:73–87, 1963.
- [7] F. L. Bauer. Remarks on optimally scaled matrices. *Numerische Mathematik*, 13:1–3, 1969.

-
- [8] G. D. Birkhoff. Tres observaciones sobre el algebra lineal. *Universidad Nacional de Tucuman Revista, Serie A*, 5:147–151, 1946.
- [9] Alberto Borobia and Rafael Cantó. Matrix scaling: A geometric proof of Sinkhorn’s theorem. *Linear Algebra and its Applications*, 268(1–3):1–8, January 1998.
- [10] R. A. Brualdi, S. V. Parter, and H. Schneider. The diagonal equivalence of a nonnegative matrix to a stochastic matrix. *Journal of Mathematical Analysis and Applications*, 16:31–50, 1966.
- [11] J. R. Bunch. Equilibration of symmetric matrices in the max-norm. *Journal of the ACM*, 18(4):566–572, 1971.
- [12] J. R. Bunch and B. N. Parlett. Direct methods for solving symmetric indefinite systems of linear equations. *SIAM Journal on Numerical Analysis*, 8(4):639–655, 1971.
- [13] A. R. Curtis and J. K. Reid. On the automatic scaling of matrices for Gaussian elimination. *IMA Journal of Applied Mathematics*, 10(1):118–124, 1972.
- [14] E. D. Dolan and J. J. More. Benchmarking optimization software with performance profiles. *Mathematical Programming*, 91(2):201–213, 2002.
- [15] I. S. Duff, A. M. Erisman, and J. K. Reid. *Direct Methods for Sparse Matrices*. Oxford University Press, London, 1986.
- [16] I. S. Duff and S. Pralet. Strategies for scaling and pivoting for sparse symmetric indefinite problems. *SIAM Journal on Matrix Analysis and Applications*, 27(2):313–340, 2005.
- [17] Iain S. Duff and Jacko Koster. The design and use of algorithms for permuting large entries to the diagonal of sparse matrices. *SIAM Journal on Matrix Analysis and Applications*, 20(4):889–901, 1999.
- [18] Iain S. Duff and Jacko Koster. On algorithms for permuting large entries to the diagonal of a sparse matrix. *SIAM Journal on Matrix Analysis and Applications*, 22:973–996, 2001.
- [19] J. Franklin and J. Lorenz. On the scaling of multidimensional matrices. *Linear Algebra and its Applications*, 114/115:717–735, 1989.
- [20] Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, 3rd edition, 1996.
- [21] Philip A. Knight. The sinkhorn–knopp algorithm: Convergence and applications. *SIAM Journal on Matrix Analysis and Applications*, 30(1):261–275, 2008.
- [22] B. N. Parlett and T. L. Landis. Methods for scaling to double stochastic form. *Linear Algebra and its Applications*, 48:53–79, 1982.
- [23] H. Perfect and L. Mirsky. The distribution of positive elements in doubly stochastic matrices. *The Journal of the London Mathematical Society*, 40:689–698, 1965.

-
- [24] T. E. S. Raghavan. On pairs of multidimensional matrices. *Linear Algebra and its Applications*, 62:263–268, 1984.
 - [25] U. G. Rothblum, H. Schneider, and M. H. Schneider. Scaling matrices to prescribed row and column maxima. *SIAM Journal on Matrix Analysis and Applications*, 15(1):1–14, 1994.
 - [26] M. H. Schneider and S. Zenios. A comparative study of algorithms for matrix balancing. *Operations Research*, 38(3):439–455, 1990.
 - [27] R. Sinkhorn. A relationship between arbitrary positive matrices and doubly stochastic matrices. *The Annals of Mathematical Statistics*, 35:876–879, 1964.
 - [28] R. Sinkhorn. Diagonal equivalence to matrices with prescribed row and column sums. *American Mathematical Monthly*, 74:402–405, 1967.
 - [29] R. Sinkhorn and P. Knopp. Concerning nonnegative matrices and doubly stochastic matrices. *Pacific Journal of Mathematics*, 21(2):343–348, 1967.
 - [30] R. Sinkhorn and P. Knopp. Problems concerning diagonal products in nonnegative matrices. *Transactions of the American Mathematical Society*, 136:67–75, 1969.
 - [31] George W. Soules. The rate of convergence of Sinkhorn balancing. *Linear Algebra and its Applications*, 150:3–40, 1991. Proceedings of the First Conference of the International Linear Algebra Society (Provo, UT, 1989).
 - [32] A. van der Sluis. Condition numbers and equilibration of matrices. *Numerische Mathematik*, 14:14–23, 1969.



Centre de recherche INRIA Grenoble – Rhône-Alpes
655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)

Centre de recherche INRIA Bordeaux – Sud Ouest : Domaine Universitaire - 351, cours de la Libération - 33405 Talence Cedex
Centre de recherche INRIA Lille – Nord Europe : Parc Scientifique de la Haute Borne - 40, avenue Halley - 59650 Villeneuve d'Ascq
Centre de recherche INRIA Nancy – Grand Est : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex
Centre de recherche INRIA Paris – Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex
Centre de recherche INRIA Rennes – Bretagne Atlantique : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex
Centre de recherche INRIA Saclay – Île-de-France : Parc Orsay Université - ZAC des Vignes : 4, rue Jacques Monod - 91893 Orsay Cedex
Centre de recherche INRIA Sophia Antipolis – Méditerranée : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
<http://www.inria.fr>
ISSN 0249-6399