HAL
open science

Preconditioning techniques based on the Birkhoff–von Neumann decomposition

Michele Benzi, Alex Pothen, Bora Uçar

To cite this version:

Michele Benzi, Alex Pothen, Bora Uçar. Preconditioning techniques based on the Birkhoff–von Neumann decomposition. [Research Report] RR-8914, Inria - Research Centre Grenoble – Rhône-Alpes. 2016. hal-01318486v2

HAL Id: hal-01318486

https://inria.hal.science/hal-01318486v2

Submitted on 11 Oct 2016 (v2), last revised 5 Jan 2017 (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.

Preconditioning techniques based on the Birkhoff–von Neumann decomposition

Michele Benzi, Alex Pothen, Bora Uçar

RESEARCH REPORT
N° 8914
May 2016
Project-Team ROMA

ISSN 0249-6399 ISRN INRIA/RR--8914--FR+ENG

Preconditioning techniques based on the Birkhoff–von Neumann decomposition

Michele Benzi*, Alex Pothen†, Bora Uçar‡

Project-Team ROMA

Research Report n° 8914 — May 2016 — 19 pages

Abstract: We introduce a class of preconditioners for general sparse matrices based on the Birkhoff–von Neumann decomposition of doubly stochastic matrices. These preconditioners are aimed primarily at solving challenging linear systems with highly unstructured and indefinite coefficient matrices. We present some theoretical results and numerical experiments on linear systems from a variety of applications.

Key-words: preconditioning, doubly stochastic matrix, Birkhoff–von Neumann decomposition

* Department of Mathematics and Computer Science, Emory University, Atlanta, USA (benzi@mathcs.emory.edu).

† Department of Computer Science, Purdue University, West Lafayette, USA (apothen@purdue.edu).

‡ CNRS and LIP (UMR5668 CNRS-ENS Lyon-INRIA-UCBL), 46, allée d'Italie, ENS Lyon, Lyon F-69364, France.

RESEARCH CENTRE
GRENOBLE – RHÔNE-ALPES

Inovallée
655 avenue de l'Europe Montbonnot
38334 Saint Ismier Cedex

Une méthode de préconditionnement basée sur la décomposition de Birkhoff-von Neumann des matrices bistochastiques

Résumé : Nous introduisons une classe de préconditionnement pour matrices creuses générales fondées sur la décomposition de Birkhoff–von Neumann des matrices bistochastiques. Ces préconditionnements visent principalement à résoudre des systèmes linéaires difficiles dont les matrices sont non-structurées et non-définies. Nous présentons quelques résultats théoriques et des expériences numériques pour des systèmes linéaires issus de tout un ensemble d'applications.

Mots-clés : préconditionnement, matrices bistochastiques, décomposition de Birkhoff-von Neumann

1 Introduction

We consider the solution of linear systems $Ax = b$ where $A = [a_{ij}] \in \mathbb{R}^{n\times n}$, b is a given vector and x is the unknown vector. Our aim is to develop and investigate preconditioners for Krylov subspace methods for solving such linear systems, where A is highly unstructured and indefinite.

For a given matrix A, we first preprocess it to get a doubly stochastic matrix (whose row and column sums are one). Then using this doubly stochastic matrix, we select some fraction of some of the nonzeros of A to be included in the preconditioner. Our main tools are the well-known Birkhoff-von Neumann (BvN) decomposition (this will be discussed in Section 2 for completeness), and a splitting of the input matrix in the form $A = M - N$ based on its BvN decomposition. When such a splitting is defined, M^{-1} or solvers for $My = z$ are required. We discuss sufficient conditions when such a splitting is convergent and discuss specialized solvers for $My = z$ when these conditions are met. In case the conditions become restrictive in practice, we build preconditioners based on this splitting by using the LU decomposition of M. Our motivation is that that the preconditioner M^{-1} can be applied to vectors by a number of highly concurrent steps, where the number of steps is controlled by the user. Therefore, the preconditioners (or the splittings) can be advantageous for use in many-core computing systems. In the context of splittings, the application of N to vectors can also enjoy the same property. These motivations are shared by recent work on ILU preconditioners, where their fine-grained computation [8] and approximate application [1] are investigated for GPU-like systems.

The paper is organized as follows. We first give necessary background (Section 2) on doubly stochastic matrices and the BvN decomposition. We then develop splittings for doubly stochastic matrices (Section 3), where we analyze convergence properties and discuss algorithms to construct the preconditioners. Later in the same Section, we discuss how the preconditioners can be used for arbitrary matrices by some preprocessing. Here our approach results in a generalization of the Birkhoff-von Neumann decomposition for matrices with positive and negative entries where the sum of the absolute values of the entries in any given row or column is one. This generalization could be of interest in other areas. Then, we give experimental results (Section 4) with nonnegative and also arbitrary matrices, and then conclude the paper.

2 Background and definitions

Here we define several properties of matrices: irreducible, full indecomposable, and doubly stochastic matrices.

An $n \times n$ matrix A is reducible if there exists a permutation matrix P such that

$$PAP^T = \begin{bmatrix} A_{1,1} & A_{1,2} \\ O & A_{2,2} \end{bmatrix},$$

where $A_{1,1}$ is an $r \times r$ submatrix, $A_{2,2}$ is an $(n-r) \times (n-r)$ submatrix, and $1 \le r < n$. If such a permutation matrix does not exist, then A is irreducible [21, Ch. 1]. When A is reducible, either $A_{1,1}$ or $A_{2,2}$ can be reducible as well, and we can recursively identify their diagonal blocks, until all diagonal blocks are irreducible. That is, we can obtain

$$PAP^T = \begin{bmatrix} A_{1,1} & A_{1,2} & \cdots & A_{1,s} \\ 0 & A_{2,2} & \cdots & A_{2,s} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & A_{s,s} \end{bmatrix}, \tag{1}$$

where each $A_{i,i}$ is square and irreducible. This block upper triangular form, with square irreducible diagonal blocks is called Frobenius normal form [15, p. 532].

An $n \times n$ matrix A is fully indecomposable if there exists a permutation matrix Q such that AQ has a zero-free diagonal and is irreducible [6, Chs. 3 and 4]. If A is not fully indecomposable, but nonsingular, it can be permuted into the block upper triangular form

$$PAQ^T = \begin{bmatrix} A_{1,1} & A_{1,2} \\ O & A_{2,2} \end{bmatrix},$$

where each $A_{i,i}$ is fully indecomposable or can be further permuted into the block upper triangular form. If the coefficient matrix of a linear system is not fully indecomposable, the block upper triangular form should be obtained, and only the small systems with the diagonal blocks should be factored for simplicity and efficiency [11, Ch. 6]. We therefore assume without loss of generality that matrix A is fully indecomposable.

An $n \times n$ matrix A is doubly stochastic if $a_{ij} \geq 0$ for all i, j and $Ae = A^T e = e$, where e is the vector of all ones. This means that the row sums and column sums are equal to one. If these sums are less than one, then the matrix A is doubly substochastic. A doubly stochastic matrix is fully indecomposable or is block diagonal where each block is fully indecomposable. By Birkhoff's Theorem [3], there exist coefficients $\alpha_1, \alpha_2, \ldots, \alpha_k \in (0, 1)$ with $\sum_{i=1}^{k} \alpha_i = 1$, and permutation matrices $P_1, P_2, \ldots, P_k$ such that

$$A = \alpha_1 P_1 + \alpha_2 P_2 + \cdots + \alpha_k P_k\,. \tag{2}$$

Such a representation of A as a convex combination of permutation matrices is known as a *Birkhoff–von Neumann decomposition* (BvN); in general, it is not unique. The *Marcus–Ree Theorem* states that there are BvN decompositions with $k \leq n^2 - 2n + 2$ for dense matrices; Brualdi and Gibson [5] and Brualdi [4] show that for a fully indecomposable sparse matrix with τ nonzeros, we have BvN decompositions with $k \leq \tau - 2n + 2$.

An $n \times n$ nonnegative, fully indecomposable matrix A can be uniquely scaled with two positive diagonal matrices R and C such that RAC is doubly stochastic [20].

3 Splittings of doubly stochastic matrices

3.1 Definition and properties.

Let $b \in \mathbb{R}^n$ be given and consider solving the linear system $Ax = b$ where A is doubly stochastic. Hereafter we assume that A is invertible. After finding a representation of A in the form (2), pick an integer r between 1 and $k - 1$ and split A as

$$A = M - N, \tag{3}$$

where

$$M = \alpha_1 P_1 + \cdots + \alpha_r P_r, \qquad (4)$$
$$N = -\alpha_{r+1} P_{r+1} - \cdots - \alpha_k P_k.$$

Note that M and $-N$ are doubly substochastic matrices.

Definition 1. *A splitting of the form (3) with M and N given by (4) is said to be a* doubly substochastic splitting.

Definition 2. *A doubly substochastic splitting $A = M - N$ of a doubly stochastic matrix A is said to be* standard *if M is invertible. We will call such a splitting an* SDS splitting.

In general, it is not easy to guarantee that a given doubly substochastic splitting is standard, except for some trivial situation such as the case $r = 1$, in which case M is always invertible. We also have a characterization for invertible M when $r = 2$.

Theorem 1. *Let $M = \alpha_1 P_1 + \alpha_2 P_2$. Then, M is invertible if (i) $\alpha_1 \neq \alpha_2$, or (ii) $\alpha_1 = \alpha_2$ and all the fully indecomposable blocks of M have an odd number of rows (and columns). If any such block is of even order, M is singular.*

Proof. We investigate the two cases separately.

Case (i): Without loss of generality assume that $\alpha_1 > \alpha_2$. We have

$$M = \alpha_1 P_1 + \alpha_2 P_2 = P_1(\alpha_1 I + \alpha_2 P_1^T P_2).$$

The matrix $\alpha_1 I + \alpha_2 P_1^T P_2$ is nonsingular. Indeed, its eigenvalues are of the form $\alpha_1 + \alpha_2 \lambda_j$, where λ_j is the generic eigenvalue of the (orthogonal, doubly stochastic) matrix $P_1^T P_2$, and since $|\lambda_j| = 1$ for all j and $\alpha_1 > \alpha_2$, it follows that $\alpha_1 + \alpha_2 \lambda_j \neq 0$ for all j. Thus, M is invertible.

Case (ii): This is a consequence of the Perron–Frobenius Theorem. To see this, observe that we need to show that under the stated conditions the sum $P_1 + P_2 = P_1(I + P_1^T P_2)$ is invertible, i.e., $\lambda = -1$ cannot be an eigenvalue of $P_1^T P_2$. Since both P_1 and P_2 are permutation matrices, $P_1^T P_2$ is also a permutation matrix and the Frobenius normal form $T = \Pi(P_1^T P_2)\Pi^T$ of $P_1^T P_2$, i.e., the block triangular matrix

$$T = \begin{bmatrix} T_{1,1} & T_{1,2} & \cdots & T_{1,s} \\ 0 & T_{2,2} & \cdots & T_{2,s} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & T_{s,s} \end{bmatrix}, \qquad (5)$$

has $T_{i,j} = 0$ for $i \neq j$. The eigenvalues of $P_1^T P_2$ are just the eigenvalues of the diagonal blocks $T_{i,i}$ of T. Note that there may be only one such block, corresponding to the case where $P_1^T P_2$ is irreducible. Each diagonal block $T_{i,i}$ is also a permutation matrix. Thus, each $T_{i,i}$ is doubly stochastic, orthogonal, and irreducible. Any matrix of this kind corresponds to a cyclic permutation and has its eigenvalues on the unit circle. If a block has size $r > 1$, its eigenvalues are the pth roots of unity, ε^h, $h = 0, 1, \ldots, p-1$, with $\varepsilon = e^{\frac{2\pi i}{p}}$, e.g., by the Perron–Frobenius Theorem (see [14, page 53]). But $\lambda = -1$ is a pth root of unity if and only if p is even. Since M is a scalar multiple of $P_1 + P_2 = P_1(I + P_1^T P_2)$, this concludes the proof. □

Note that the fully indecomposable blocks of M mentioned in the theorem are just connected components of its bipartite graph.

It is possible to generalize condition (i) in the previous theorem as follows.

Theorem 2. *A sufficient condition for $M = \sum_{i=1}^{r} \alpha_i P_i$ to be invertible is that one of the α_i with $1 \le i \le r$ be greater than the sum of the remaining ones.*

Proof. Indeed, assuming (without loss of generality) that $\alpha_1 > \alpha_2 + \cdots + \alpha_r$, we have

$$\begin{aligned} M &= \alpha_1 P_1 + \alpha_2 P_2 + \cdots + \alpha_r P_r \\ &= P_1(\alpha_1 I + \alpha_2 P_1^T P_2 + \cdots + \alpha_r P_1^T P_r). \end{aligned}$$

This matrix is invertible if and only if the matrix $\alpha_1 I + \sum_{i=2}^{r} \alpha_i P_1^T P_i$ is invertible. Observing that the eigenvalues λ_j of $\sum_{i=2}^{r} \alpha_i P_1^T P_i$ satisfy

$$|\lambda_j| \le \|\alpha_2 P_1^T P_2 + \cdots + \alpha_r P_1^T P_r\|_2 \le \sum_{i=2}^{r} \alpha_i < \alpha_1$$

(where we have used the triangle inequality and the fact that the 2-norm of an orthogonal matrix is one), it follows that M is invertible. □

Again, this condition is only a sufficient one. It is rather restrictive in practice.

3.2 Convergence conditions.

Let $A = M - N$ be an SDS splitting of A and consider the stationary iterative scheme

$$x^{k+1} = Hx^k + c, \qquad H = M^{-1}N, \qquad c = M^{-1}b \tag{6}$$

where $k = 0, 1, \ldots$ and x^0 is arbitrary. As is well known, the scheme (6) converges to the solution of $Ax = b$ for any x^0 if and only if $\rho(H) < 1$. Hence, we are interested in conditions that guarantee that the spectral radius of the iteration matrix

$$H = M^{-1}N = -(\alpha_1 P_1 + \cdots + \alpha_r P_r)^{-1}(\alpha_{r+1} P_{r+1} + \cdots + \alpha_k P_k)$$

is strictly less than one. In general, this problem appears to be difficult. We have a necessary condition (Theorem 3), and a sufficient condition (Theorem 4) which is simple but restrictive.

Theorem 3. *For the splitting $A = M - N$ with $M = \sum_{i=1}^{r} \alpha_i P_i$ and $N = -\sum_{i=r+1}^{k} \alpha_i P_i$ to be convergent, it must hold that $\sum_{i=1}^{r} \alpha_i > \sum_{i=r+1}^{k} \alpha_i$.*

Proof. First, observe that since $P_i e = e$ for all i, both M and N have constant row sums:

$$Me = \sum_{i=1}^{r} \alpha_i P_i e = \beta e, \quad Ne = -\sum_{i=1}^{r} \alpha_i P_i e = (\beta - 1)e,$$

with $\beta := \sum_{i=1}^{r} \alpha_i \in (0, 1)$. Next, observe that $M^{-1}Ne = \lambda e$ is equivalent to $Ne = \lambda Me$ or, since we can assume that $\lambda \neq 0$, to $Me = \frac{1}{\lambda} Ne$. Substituting βe for Me and $(\beta - 1)e$ for Ne we find

$$\beta e = \frac{1}{\lambda}(\beta - 1)e,$$

or

$$\lambda = \frac{\beta - 1}{\beta}.$$

Hence, $\frac{\beta-1}{\beta}$ is an eigenvalue of $H = M^{-1}N$ corresponding to the eigenvector e. Since $|\lambda| = \frac{1-\beta}{\beta}$, we conclude that $\rho(H) \geq 1$ for $\beta \in (0, \frac{1}{2}]$. This concludes the proof. □

Theorem 4. *Suppose that one of the α_i appearing in M is greater than the sum of all the other α_i. Then $\rho(M^{-1}N) < 1$ and the stationary iterative method (6) converges for all x^0 to the unique solution of $Ax = b$.*

Proof. Assuming (without loss of generality) that

$$\alpha_1 > \alpha_2 + \cdots + \alpha_k \,, \tag{7}$$

(which, incidentally, insures that the matrix M is invertible) we show below that

$$\|H\|_2 = \|(\alpha_1 P_1 + \cdots + \alpha_r P_r)^{-1}(\alpha_{r+1} P_{r+1} + \cdots + \alpha_k P_k)\|_2 < 1. \tag{8}$$

This, together with the fact that $\rho(H) \leq \|H\|_2$, ensures convergence. To prove (8) we start by observing that

$$\begin{aligned} M &= \alpha_1 P_1 + \cdots + \alpha_r P_r \\ &= \alpha_1 P_1 \left(I + \frac{\alpha_2}{\alpha_1} Q_2 + \cdots + \frac{\alpha_r}{\alpha_1} Q_r \right) \,, \end{aligned}$$

where $Q_i = P_1^T P_i$. Thus, we have

$$M^{-1} = (\alpha_1 P_1 + \cdots + \alpha_r P_r)^{-1} = \frac{1}{\alpha_1}(I - G)^{-1} P_1^T \,,$$

where we have defined

$$G = -\frac{1}{\alpha_1} \sum_{i=2}^{r} \alpha_i Q_i \,.$$

Next, we observe that $\|G\|_2 < 1$, since

$$\|G\|_2 \leq \frac{1}{\alpha_1} \sum_{i=2}^{r} \alpha_i \|Q_i\|_2 = \frac{\alpha_2 + \cdots + \alpha_r}{\alpha_1} < 1$$

as a consequence of (7). Hence, the expansion

$$(I - G)^{-1} = \sum_{\ell=0}^{\infty} G^\ell$$

is convergent, and moreover

$$\begin{aligned} \|(I - G)^{-1} P_1^T\|_2 = \|(I - G)^{-1}\|_2 &\leq \frac{1}{1 - \|G\|_2} \\ &\leq \frac{1}{1 - \left(\frac{\alpha_2}{\alpha_1} + \cdots + \frac{\alpha_r}{\alpha_1} \right)} \,. \end{aligned}$$

The last inequality follows from the fact that $\|G\|_2 \leq \sum_{i=2}^{r} \frac{\alpha_i}{\alpha_1}$, as can be seen using the triangle inequality and the fact that $\|Q_i\|_2 = 1$ for all $i = 2, \ldots, r$.

Hence, we have

$$\|M^{-1}N\|_2 \leq \frac{1}{\alpha_1} \frac{1}{1 - \left(\frac{\alpha_2}{\alpha_1} + \cdots + \frac{\alpha_r}{\alpha_1}\right)} \|\alpha_{r+1}P_{r+1} + \cdots + \alpha_k P_k\|_2 \ .$$

Using once again the triangle inequality (applied to the last term on the right of the foregoing expression) we obtain

$$\|M^{-1}N\|_2 \leq \frac{\alpha_{r+1} + \cdots + \alpha_k}{\alpha_1 - (\alpha_2 + \cdots + \alpha_r)} \ .$$

Using the condition (7) we immediately obtain that

$$\frac{\alpha_{r+1} + \cdots + \alpha_k}{\alpha_1 - (\alpha_2 + \cdots + \alpha_r)} < 1 \ ,$$

therefore $\|M^{-1}N\|_2 < 1$ and the iterative scheme (6) is convergent. □

Note that if condition (7) is satisfied, then the value of r in (4) can be chosen arbitrarily; that is, the splitting will converge for any choice of r between 1 and k. In particular, the splitting $A = M - N$ with $M = \alpha_1 P_1$ and $N = -\sum_{i=2}^{k} \alpha_i P_i$ is convergent. It is an open question whether adding more terms to M (that is, using $M = \sum_{i=1}^{r} \alpha_i P_i$ with $r > 1$) will result in a smaller spectral radius of H (and thus in faster asymptotic convergence); notice that adding terms to M will make application of the preconditioner more expensive.

Unfortunately, condition (7) is very restrictive. It implies that $\alpha_1 > 1/2$, a very strong restriction. It is clear that given a doubly substochastic matrix, in general it will have no representation (2) with $\alpha_1 > 1/2$. On the other hand, it is easy to find examples of splittings of doubly substochastic matrices for which $\alpha_1 = 1/2$ and the splitting (4) with $r = 1$ is not convergent. Also, we have found examples with $k = 3$, $r = 2$ and $\alpha_1 + \alpha_2 > \alpha_3$ where the splitting did not converge. It is an open problem to identify other, less restrictive conditions on the α_i (with $1 \leq i \leq r$) that will ensure convergence, where the pattern of the permutations could also be used. Another option is to renounce convergence and use M as a preconditioner for a Krylov subspace method like GMRES.

3.3 Solving linear systems with M.

The stationary iterations (6) for solving $Ax = b$ or Krylov subspace methods using M as a preconditioner require solving linear systems of the form $Mz = y$. Assume that $M = \alpha_1 P_1 + \alpha_2 P_2$ and $\alpha_1 > \alpha_2$. The stationary iterations $z^{k+1} = \frac{1}{\alpha_1} P_1^T (y - \alpha_2 P_2 z^k)$ are convergent for any starting point, with the rate of $\frac{\alpha_2}{\alpha_1}$. Therefore, if $M = \alpha_1 P_1 + \alpha_2 P_2$ and M is as described in Theorem 1(i), then we use the above iterations to apply M^{-1} (that is, solve linear systems with M). If M is as described in Theorem 4, then we can still solve $Mz = y$ by stationary iterations, where we use the splitting $M = \alpha_1 P_1 - \sum_{r=2}^{k} \alpha_r P_r$. We note that application of $\sum_{r=2}^{k} \alpha_r P_r$ to a vector y, that is $z = (\sum_{r=2}^{k} \alpha_r P_r) y$ can be effectively computed in $k - 1$ steps, where at each step, we perform $z \leftarrow z + (\alpha_r P_r) y$. This last operation takes n input entries, scale them and add to n different positions in z. As there is no read or write dependencies between these n operations, each step is trivially parallel; especially in shared memory environments, the only parallelization overhead is

the creation of threads. Either input can be read in order, or the output can be written in order (by using the inverse of the permutation P_r).

As said before, splitting is guaranteed to work when $\alpha_1 > 1/2$. Our experience showed that it does not work in general when $\alpha_1 < 1/2$. Therefore, we suggest using M as a preconditioner in Krylov subspace methods, when $\alpha_1 < 1/2$. In this case, the application of M requires solving linear systems with M. Considering that the number of nonzeros in M would be much less than that of A, we can use a direct method for M. In other words, we use a complete factorization of an incomplete version of A as a preconditioner. Another alternative is to build M in such a way that one of the coefficients is bigger than the sum of the rest. This can be doable by adding one permutation matrix at a time until the condition is met. This way, we have M with $\frac{\alpha_1}{\sum_{i=1}^{k} \alpha_i} > 1/2$ and M^{-1} can be applied with splitting iterations.

3.4 Algorithms for constructing the preconditioners.

It is desirable to have a small number k in the Birkhoff–von Neumann decomposition while designing the preconditioner. This is because of the fact that if we use splitting, then k determines the number of steps in which we compute the matrix vector products. If we do not use all k permutation matrices, having a few with large coefficients should help to design the preconditioner. The problem of obtaining a Birkhoff–von Neumann decomposition with the minimum number k is a difficult one, as pointed out by Brualdi [4] and has been shown to be NP-complete [12]. This last reference also discusses a heuristic which delivers a small number of permutation matrices. We use this heuristic to obtain a Birkhoff–von Neumann decomposition, or the first r permutation matrices corresponding to the largest r coefficients in the complete Birkhoff-von Neumann decomposition resulting from this heuristic. The heuristic is shown in Algorithm 1. As seen in Algorithm 1, the heuristic proceeds step by step, where at each step j, a bottleneck matching is found to determine α_j and P_j. A bottleneck matching can be found using MC64 [9, 10].

Algorithm 1: A greedy algorithm for BvN decomposition.

Input : A, a doubly stochastic matrix
Output: A BvN decomposition (2)

$k \leftarrow 0$
while $\mathrm{nnz}(A) > 0$ **do**
  $k \leftarrow k + 1$
  $P_k \leftarrow$ the pattern of a bottleneck perfect matching M in A
  $\alpha_k \leftarrow \min M_{i,P_k(i)}$
  $A \leftarrow A - \alpha_k P_k$

In this heuristic, at step (4) M is a bottleneck perfect matching; that is, the minimum weight of an edge in M is the maximum among all minimum elements of perfect matchings. Also, at step (6), α_k is equal to the bottleneck value of the perfect matching M. The worst case running time of a step of this heuristic can be bounded by the worst case running time of a bottleneck matching algorithm. For matrices where $\mathrm{nnz}(A) = O(n)$, the best known algorithm is of time complexity $O(n\sqrt{n \log n})$ [13]. We direct the reader to [7, p. 185] for other cases.

3.5 Arbitrary coefficient matrices.

Here we discuss how to apply the proposed preconditioning technique for all fully indecomposable sparse matrices.

Let $A \geq 0$ be an $n \times n$ fully indecomposable matrix. The first step is to scale the input matrix with two positive diagonal matrices R and C so that RAC is doubly stochastic, or nearly so. For this step, there are different approaches [16, 17, 20]. Next the linear system $Ax = b$ can be solved by solving $RACx' = Rb$ and recovering $x = Cx'$, where we have a doubly stochastic coefficient matrix.

Suppose A is an $n \times n$ fully indecomposable matrix with positive and negative entries. Then, let $B = \text{abs}(A)$ and consider R and C making RBC doubly stochastic, which has a Birkhoff–von Neumann decomposition $RBC = \sum_i^k \alpha_i P_i$. Then, RAC can be expressed as

$$RAC = \sum_{i=1}^{k} \alpha_i Q_i \,.$$

where $Q_i = [q_{jk}^{(i)}]_{n\times n}$ is obtained from $P_i = [p_{jk}^{(i)}]_{n\times n}$ as follows:

$$q_{jk}^{(i)} = \text{sign}(a_{jk}) p_{jk}^{(i)} \,.$$

That is, we can use a convex combination of a set of signed permutation matrices to express a matrix A where abs(A) is doubly stochastic. We can then use the same construct to use r matrices to define M (for splitting or for defining the preconditioner).

We note that the Theorems 2 and 4 remain valid without changes, since the only property we use is the orthogonality of the P_i (not the nonnegativity). Theorem 1, on the other hand, needs some changes. All we can prove in this more general setting is that if $\alpha_1 \neq \pm\alpha_2$, then $M = \alpha_1 P_1 + \alpha_2 P_2$ is nonsingular. We could not find a simple condition for the case $\alpha_1 = \pm\alpha_2$ since we cannot use the Perron-Frobenius Theorem to conclude anything about 1 (or -1) being an eigenvalue of $P_1^T P_2$.

4 Experiments

We are interested in testing our preconditioner on challenging linear systems that pose difficulties for "standard" preconditioned Krylov subspace methods. We conducted experiments with the preconditioned GMRES of Matlab, without restart. We used the matrices given in Table 1. The first 22 matrices in the table were used by Benzi et al. [2], and the last six matrices were used by Manguoglu et al. [18]. All matrices, except the last four in the first part, are available from the UFL Sparse Matrix Collection. Manguoglu et al. present experiments with larger matrices but we chose only those with $n \leq 20000$. The table also shows the size and the number of nonzeros of the largest fully indecomposable block (these are the largest square blocks in the Dulmage-Mendelsohn decomposition of the original matrices). The experiments are conducted with those largest blocks; hence from now on a matrix refers to its largest block.

We present three sets of experiments: with nonnegative matrices, with general matrices, and with nonnegative matrices using a specialized solver. In the first two, we obtain a (generalized) BvN decomposition and try a few values of "r" in defining the preconditioner M. In this case, we cannot say anything specific about M and hence we apply M^{-1} with the standard solve methods.

matrix	n_A	nnz_A	n	nnz
west0655	655	2808	452	1966
west0989	989	3518	720	2604
west1505	1505	5414	1099	3988
west2021	2021	7310	1500	5453
lhr01	1477	18427	1171	15914
lhr02	2954	36875	1171	15914
bayer09	3083	11767	1210	6001
bayer10	13436	71594	10803	62238
mahindas	1258	7682	589	4744
orani678	2529	90158	1830	47823
bp_200	822	3802	40	125
gemat11	4929	33108	4578	31425
gemat12	4929	33044	4552	31184
circuit_3	12127	48137	7607	34024
sherman2	1080	23094	870	19256
lns_3937	3937	25407	3558	24002
utm5940	5940	83842	5794	83148
venkat25	62424	1717763	62424	1717763
slide	20191	1192535	19140	1191421
two-dom	22200	1188152	20880	1186500
watson4a	468	2459	364	1480
watson5a	1854	8626	1765	6387
appu	14000	1853104	14000	1853104
bundle1	10581	770811	10581	770811
dw8192	8192	41746	8192	41746
fp	7548	834222	7548	834222
Simon/raefsky4	19779	1316789	19779	1316789
FEM_3D_th1	17880	430740	17880	430740

Table 1: Test matrices, their original size n_A and the size of the largest fully indecomposable block n. The experiments were done using the largest fully indecomposable block (n is the effective size in the following experiments). The last matrix's full name is FEM_3D_thermal1.

In the first two cases, we checked the flag returned by GMRES of Matlab and considered those with `flag=0` to be successful—we did not check if there were other warnings. In the third set of experiments, we put $\alpha_1 P_1$ to M, and then we keep adding subsequent terms as long as α_1 is larger than the sum of the rest. The M created this way satisfies the condition in Theorem 4, and M^{-1} can be applied with splitting. For this third set of experiments, we used `fgmres` [19].

4.1 Nonnegative matrices.

The first set of experiments are conducted on nonnegative matrices. Let A be a matrix from Table 1 (recall that this is the largest fully indecomposable block), and $B = \mathrm{abs}(A)$, that is $b_{ij} = |a_{ij}|$. We scaled B to a doubly stochastic form with the method of Knight et al. [17]; we specified the maximum number of scaling iterations as n and asked a tolerance of 10^{-3} (so that row and column sums can deviate from 1 by 10^{-3}). We then obtained the Birkhoff–von Neumann decomposition by using Algorithm 1. When the bottleneck value found at a step was smaller than 10^{-6}, we stopped the process—hence we obtain an "approximate" Birkhoff–von Neumann decomposition of an "approximately" doubly stochastic matrix. This way we obtained $B \approx \alpha_1 P_1 + \alpha_2 P_2 + \cdots + \alpha_k P_k$. Then, let $x^\star$ be a random vector whose entries are from the uniform distribution on the open interval $(0, 1)$, generated using `rand` of Matlab. We then defined $b = Bx^\star$ to be the right hand side of $Bx = b$. Preconditioned GMRES was used to solve the linear systems $Bx = b$ using M as a preconditioner with an error tolerance of 10^{-6}. Here M is constructed by taking the r largest α_i along with the corresponding permutations for $r \leq k$, that is $M = \sum_{i=1}^{r} \alpha_i P_i$.

We used GMRES without restart and with the maximum number of iterations set to be the minimum of 3000 and $n-1$. The number of GMRES iterations with M constructed using different numbers m of permutation matrices are seen in Table 2. In this table we also give the number of GMRES iterations with ILU(0) preconditioners. We created ILU(0) preconditioners by first permuting the coefficient matrices to have a maximum product matching in the diagonal (using MC64), and then calling `ilu` of Matlab with the option `nofill`. In the table, we highlighted the cases where the number of iterations with M were fewer than the number of iterations with ILU(0). As seen in this table, there are four matrices where the proposed preconditioners results in fewer iterations, consistently with at least 4–8 permutation matrices in M; in `bayer10`, having four permutation matrices were more effective than having eight of them. In 18 of the matrices, adding more permutation matrices to M results in improved number of iterations; by this we mean that for all $r_1 < r_2$, having r_2 permutation matrices is more helpful than having r_1 of them.

Although our major concern is a proof-of-concept but not the running time (recall that we aim for preconditioners that can be applied in efficiently in parallel), we investigate the running time of the solver and also the time spent in setting up the preconditioner in a sequential Matlab environment (on a core of an Intel Xeon E5-2695 with 2.30 GHz clock speed). Table 3 presents the complexity of the preconditioners and the running time of the solver (total time spent in the iterations). The complexity of the preconditioner is given as the ratio $(\mathrm{nnz}(L+U)) - n/\mathrm{nnz}$, where L and U are the factors of the preconditioner. As seen in this table, in a few matrices the LU-factors of M with $r > 8$ have many more nonzeros than A. This is especially worrying for `appu` where, even with 8 permutation matrices, the LU factors of M have more than $20 \cdot \mathrm{nnz}(A)$ nonzeros, while $\mathrm{nnz}(M)/\mathrm{nnz}(A) = 0.04$ and the full LU factorization of A contains $85.67 \cdot \mathrm{nnz}(A)$ nonzeros. The running time trend is closely related to what we see in the iteration table: in cases where M results in smaller number of iterations, the running time is also improved (except the unique cases

		Number r of permutation matrices in M						
matrix	ILU(0)	1	2	4	8	16	32	64
west0655	71	306	232	158	132	84	80	78
west0989	2	200	157	93	58	32	16	11
west1505	1	285	238	151	87	49	23	15
west2021	1	326	266	174	114	59	32	19
lhr01	6	566	296	184	139	81	39	24
lhr02	7	568	296	176	140	75	41	21
bayer09	123	484	303	214	**101**	**45**	**39**	**37**
bayer10	230	–	–	**14**	343	**16**	**44**	**18**
mahindas	37	225	158	87	66	43	**32**	**18**
orani678	23	172	140	99	93	71	58	49
bp_200	2	38	31	25	24	22	23	23
gemat11	176	1916	1606	1204	858	620	297	281
gemat12	254	2574	1275	1121	688	570	386	357
circuit_3	341	–	–	–	–	2614	1792	854
sherman2	19	374	260	173	85	43	21	**15**
lns_3937	348	1702	801	**314**	**167**	**48**	**36**	**37**
utm5940	133	2684	1960	1045	634	347	144	**82**
venkat25	1	–	–	–	–	–	155	201
slide	124	1040	1119	1055	879	699	523	132
two-dom	1567	–	–	–	–	2492	1783	**1359**
watson4a	46	171	170	134	167	142	118	144
watson5a	7	598	732	841	954	909	403	282
appu	31	49	53	56	70	103	173	219
bundle1	25	129	139	135	115	100	79	59
dw8192	79	–	–	1698	1058	495	287	290
fp	245	–	–	–	**22**	**6**	**4**	**2**
raefsky4	8	–	–	–	2948	2553	1716	1101
FEM_3D_th1	7	38	38	54	204	124	11	**5**

Table 2: The number of GMRES iterations with different number of permutation matrices in M with tolerance 10^{-6}. We ran GMRES (without restart) for at most 3000 iterations; the symbol "–" in the Table flags cases where GMRES did not converge to the required tolerance. **All matrices are nonnegative.**

	Complexity of M						Running time				
	M with r permutation matrices						M with r permutation matrices				
matrix	4	8	16	32	64	ILU(0)	4	8	16	32	64
west0655	0.75	1.15	1.51	1.72	1.72	0.05	0.17	0.13	0.06	0.05	0.05
west0989	0.52	0.77	0.99	1.18	1.27	0.00	0.08	0.04	0.02	0.01	0.01
west1505	0.56	0.82	1.02	1.24	1.36	0.01	0.23	0.08	0.03	0.02	0.01
west2021	0.55	0.82	1.05	1.24	1.31	0.01	0.38	0.16	0.05	0.02	0.01
lhr01	0.17	0.28	0.52	1.28	1.91	0.01	0.29	0.19	0.08	0.02	0.02
lhr02	0.17	0.28	0.52	1.44	2.10	0.01	0.34	0.20	0.08	0.05	0.02
bayer09	0.54	0.91	1.32	1.66	1.66	0.17	0.43	0.12	0.03	0.03	0.03
bayer10	0.48	0.72	1.15	1.61	2.18	6.79	0.23	13.92	0.24	0.50	0.27
mahindas	0.25	0.34	0.44	0.61	0.96	0.03	0.06	0.04	0.02	0.01	0.01
orani678	0.05	0.05	0.07	0.09	0.16	0.04	0.13	0.12	0.07	0.07	0.08
bp_200	0.93	1.01	1.01	1.01	1.01	0.00	0.01	0.01	0.01	0.01	0.01
gemat11	0.34	0.58	1.28	1.70	1.78	2.22	135.96	59.22	34.48	8.24	6.84
gemat12	0.31	0.54	1.16	1.61	1.64	5.12	96.30	24.01	21.24	13.29	10.32
circuit_3	–	–	0.47	0.58	0.96	9.09	–	–	752.28	362.16	84.86
sherman2	0.10	0.14	0.23	0.45	0.73	0.01	0.25	0.08	0.03	0.01	0.02
lns_3937	0.56	1.78	3.03	4.39	6.92	7.15	5.06	2.36	0.28	0.23	0.24
utm5940	0.24	0.86	2.01	6.00	7.83	1.55	127.01	28.81	9.56	1.81	1.21
venkat25	–	–	–	7.20	6.92	0.76	–	–	–	18.05	26.45
slide	0.05	3.89	18.78	30.28	30.29	2.59	229.14	168.96	138.11	104.92	14.25
two-dom	–	–	10.35	21.94	27.90	398.12	–	–	1160.53	876.66	401.56
watson4a	0.26	0.28	0.31	0.37	0.37	0.03	0.20	0.16	0.15	0.14	0.11
watson5a	0.35	0.41	0.60	0.71	0.75	0.02	5.78	9.08	8.78	1.81	0.94
appu	0.48	20.16	34.74	54.53	68.70	0.84	1.21	7.21	16.71	39.45	68.46
bundle1	0.01	0.02	0.03	0.07	0.14	0.38	1.97	1.61	1.38	0.87	0.64
dw8192	0.58	0.64	1.58	6.18	7.85	0.85	399.33	96.71	22.77	7.57	7.54
fp	–	0.19	0.33	0.39	0.82	4.93	–	0.23	0.15	0.15	0.14
raefsky4	–	1.46	7.75	12.58	11.85	0.33	–	1908.78	1490.82	970.97	292.71
FEM_3D_th1	0.12	1.15	5.56	12.81	21.49	0.26	0.80	7.64	3.67	0.44	0.38

Table 3: The complexity of the preconditioners and the running time of the solver. The complexity of the preconditioner is given as the ratio $(\mathrm{nnz}(L+U)-n)/\,\mathrm{nnz}$, where L and U are the factors of the preconditioner (ILU(0) has a complexity of 1.0). **All matrices are nonnegative.**

with `two-dom` and `FEM_3D_th1`). Overall, M as a preconditioner seems to be helpful for this set of experiments, especially when more than 4–8 permutation matrices are used in M. This come however with memory overhead which should be taken into account along with the preconditioner set up and the number of iterations.

We now comment on the running time of the preconditioner set up phase. For the largest of these matrices (`venkat25`), the scaling algorithm took 619.90 seconds (with 5287 iterations), the approximate BvN decomposition heuristic with 64 permutation matrices took 332.09 seconds, and the ILU factorization of M took 1.75 seconds. We note that there are more efficient scaling algorithms [16] for the most expensive of these two steps.

4.2 General matrices.

In this set of experiments, we used the matrices listed in Table 1 while retaining the sign of the nonzero entries. We used the same set of scaling matrices as in the previous section and used the proposed generalized BvN decomposition (Section 3.5) to construct the preconditioners.

Table 4 gives the number of GMRES iterations with M constructed using different numbers r of permutation matrices. In this table we also give the number of GMRES iterations with ILU(0) preconditioners, constructed as before (in particular MC64 is used to improve ILU(0)'s effectiveness). The cases where the proposed preconditioner led to fewer iterations than ILU(0) are again shown in bold face. Most of the time, these cases are the same as the ones in Table 2. In this experiments too, there were 18 matrices where adding more permutation matrices to M results in improved number of iterations. All linear systems were solved successfully with at least 8 permutation matrices in M. We do not list the complexity and the running time results for this set of experiments. The complexity results are almost the same as in the nonnegative case. The running time results have the same trend as in the nonnegative case: where there are improvements in the iteration numbers, there are improvements in the running time as well. Again, M as a preconditioner seems to be helpful for this set of experiments, but the memory overhead, the preconditioner set up time, and the total number of iterations should be taken into account.

4.3 Special M.

We used `fgmres` without restart and with the maximum number of iterations set to be the minimum of 3000 and $n-1$, where the preconditioner M is applied using the proposed solution method (Section 3.3). Here, we performed experiments with a relaxed error tolerance of 10^{-3} for `fgmres` for shorter experimentation time. The inner solver is run with a tolerance of 10^{-1} and with at most $\min\{1000, n-1\}$ iterations. We investigated six small matrices in which the proposed preconditioner was found more effective than ILU(0) in the previous subsections. These matrices are `west0655`, `bayer09`, `mahindas`, `sherman2`, `lns_3937`, `utm5940`— we did not investigate `venkat`, `fp` and `FEM_3D_th1` as these were large for our tests. The number of iterations is shown in Table 5 for nonnegative and general (negative an positive entires) matrices; the cases where the solver did not converge to the required accuracy within the allowed iterations are shown with "–". As seen in this table, the proposed solver did not lead to convergence for `lns_3937` (both cases); this was also the case for `utm5940`. We investigated the cases and saw that α_1 was close to the sum of the rest of the coefficients and that the inner iterations were not enough; allowing n inner iterations lead to convergence in the original `utm5940` matrix in 2353 iterations. Here the number of iterations with M as a preconditioner is high with respect to ILU(0); however the application of M has much higher concurrent than that of M.

5 Conclusion

We introduced a class of preconditioners for general sparse matrices based on the Birkhoff–von Neumann decomposition of doubly stochastic matrices. These preconditioners are aimed primarily at solving challenging linear systems with highly unstructured and indefinite coefficient matrices. We presented some theoretical results and numerical experiments on linear systems from a variety of applications. We presented a proof of concept realization; there are many challenging questions remain to be investigated to render the proposed preconditioners competitive with the standard approaches.

Based on our current theoretical findings, we suggest the use of proposed preconditioners within a Krylov subspace method, where the application of the preconditioner requires an LU decomposition. The proposed preconditioner is therefore constructed as a complete factorization of an

		Number r of permutation matrices in M						
matrix	ILU(0)	1	2	4	8	16	32	64
west0655	52	263	187	118	79	60	**46**	**46**
west0989	2	202	158	100	57	33	18	9
west1505	1	290	243	135	90	46	25	14
west2021	1	328	280	188	111	51	26	18
lhr01	6	473	217	138	113	65	29	15
lhr02	7	477	216	136	111	65	30	17
bayer09	39	304	180	87	**37**	**19**	**18**	**18**
bayer10	6	–	2028	13	96	134	38	13
mahindas	26	183	124	74	50	35	**23**	**14**
orani678	18	161	134	89	86	66	51	40
bp_200	2	36	30	22	22	22	22	22
gemat11	223	–	2385	1420	891	612	352	288
gemat12	264	–	1596	1259	767	573	467	460
circuit_3	58	–	2127	–	2813	1970	1286	475
sherman2	18	294	176	106	60	28	**17**	**14**
lns_3937	140	1142	428	197	**74**	**20**	**16**	**15**
utm5940	74	2387	1500	931	571	341	123	81
venkat25	151	–	2392	1521	962	554	214	**79**
slide	208	1014	964	–	682	559	830	647
two-dom	147	585	547	538	604	610	401	316
watson4a	34	171	149	164	139	158	116	120
watson5a	7	925	919	906	1079	1084	449	298
appu	31	50	53	56	70	116	177	217
bundle1	18	121	117	80	107	90	73	58
dw8192	63	–	–	2043	983	488	297	306
fp	159	–	–	1337	480	**62**	**32**	**31**
raefsky4	513	2306	2477	2088	1863	1508	1138	814
FEM_3D_th1	7	42	45	60	216	121	12	**5**

Table 4: The number of GMRES iterations with different number of permutation matrices in M with tolerance 10^{-6}. We ran GMRES (without restart) for at most 3000 iterations; the symbol "–" in the Table flags cases where GMRES did not converge to the required tolerance. **Matrices have negative and positive entries.**

	nonnegative			neg. and pos.	
		M			
matrix	ILU(0)	r	iter	ILU(0)	M
west0655	43	7	383	43	419
bayer09	24	12	1000	24	880
mahindas	11	6	587	11	444
sherman2	11	7	851	11	856
lns_3937	98	7	–	98	–
utm5940	6	6	2916	6	–

Table 5: The number of `fgmres` iterations with ILU(0), with M, and the number of permutation matrices in M. The proposed solver is used for applying the preconditioner M which by construction has large α_1 and contain r permutation matrices. The same number r of permutation matrices are used in both nonnegative and general cases.

incomplete matrix. What would be a good way to order these matrices (we have results for the simple case of $M = \alpha_1 P_1 + \alpha_2 P_2$)?

In our current set up, we construct the preconditioners by using MC64 repeatedly to find an approximate BvN decomposition. There is an obvious dependency from one run of MC64 to the next one, but we did not take advantage of it. What is a good algorithm to obtain (an approximate) BvN for the sake of preconditioning?

Acknowledgements

The work of Michele Benzi was supported in part by NSF grant DMS-1418889, and that of Alex Pothen by NSF grant CCF 1552323 and DOE grant DE-SC00010205. Bora Uçar was supported in part by French National Research Agency (ANR) project SOLHAR (ANR-13-MONU-0007). This work resulted from the collaborative environment offered by the Dagstuhl Seminar 14461 on High-performance Graph Algorithms and Applications in Computational Science (November 9–14, 2014).

References

[1] H. Anzt, E. Chow, and J. Dongarra. Iterative sparse triangular solves for preconditioning. In J. L. Träff, S. Hunold, and F. Versaci, editors, *Euro-Par 2015*, pages 650–661. Springer Berlin Heidelberg, 2015.

[2] M. Benzi, J. C. Haws, and M. Tuma. Preconditioning highly indefinite and nonsymmetric matrices. *SIAM Journal on Scientific Computing*, 22(4):1333–1353, 2000.

[3] G. Birkhoff. Tres observaciones sobre el algebra lineal. *Univ. Nac. Tucumán Rev. Ser. A*, (5): 147–150, 1946.

[4] R. A. Brualdi. Notes on the Birkhoff algorithm for doubly stochastic matrices. *Canadian Mathematical Bulletin*, 25(2):191–199, 1982.

[5] R. A. Brualdi and P. M. Gibson. Convex polyhedra of doubly stochastic matrices: I. Applications of the permanent function. *Journal of Combinatorial Theory, Series A*, 22(2):194–230, 1977.

[6] R. A. Brualdi and H. J. Ryser. *Combinatorial Matrix Theory*, volume 39 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, Cambridge, UK; New York, USA; Melbourne, Australia, 1991.

[7] R. Burkard, M. Dell'Amico, and S. Martello. *Assignment Problems*. SIAM, Philadelphia, PA, USA, 2009.

[8] E. Chow and A. Patel. Fine-grained parallel incomplete LU factorization. *SIAM Journal on Scientific Computing*, 37(2):C169–C193, 2015.

[9] I. S. Duff and J. 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.

[10] I. S. Duff and J. 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.

[11] I. S. Duff, A. M. Erisman, and J. K. Reid. *Direct Methods for Sparse Matrices*. Oxford University Press, London, 1986. In preprint of second edition. To appear.

[12] F. Dufossé and B. Uçar. Notes on Birkhoff–von Neumann decomposition of doubly stochastic matrices. *Linear Algebra and its Applications*, 497:108–115, 2016.

[13] H. N. Gabow and R. E. Tarjan. Algorithms for two bottleneck optimization problems. *J. Algorithms*, 9(3):411–417, 1988.

[14] F. R. Gantmacher. *The Theory of Matrices*, volume 2. Chelsea Publishing Co.,, New York, NY, 1959.

[15] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, second edition, 2013.

[16] P. A. Knight and D. Ruiz. A fast algorithm for matrix balancing. *IMA Journal of Numerical Analysis*, 33(3):1029–1047, 2013.

[17] P. A. Knight, D. Ruiz, and B. Uçar. A symmetry preserving algorithm for matrix scaling. *SIAM Journal on Matrix Analysis and Applications*, 35(3):931–955, 2014.

[18] M. Manguoglu, M. Koyutürk, A. H. Sameh, and A. Grama. Weighted matrix ordering and parallel banded preconditioners for iterative linear system solvers. *SIAM Journal on Scientific Computing*, 32(3):1201–1216, 2010.

[19] Y. Saad. A flexible inner-outer preconditioned GMRES algorithm. *SIAM Journal on Scientific Computing*, 14(2):461–469, 1993.

[20] R. Sinkhorn and P. Knopp. Concerning nonnegative matrices and doubly stochastic matrices. *Pacific J. Math.*, 21:343–348, 1967. ISSN 0030-8730.

[21] R. S. Varga. *Matrix Iterative Analysis.* Springer, Berlin, Heidelberg, New York, second (revised and expanded edition of prentice-hall, englewood cliffs, new jersey, 1962) edition, 2000.

RESEARCH CENTRE
GRENOBLE – RHÔNE-ALPES

Inovallée

655 avenue de l'Europe Montbonnot

38334 Saint Ismier Cedex

Publisher
Inria
Domaine de Voluceau - Rocquencourt
BP 105 - 78153 Le Chesnay Cedex
inria.fr

ISSN 0249-6399