Analysis of a Classical Matrix Preconditioning Algorithm

LEONARD J. SCHULMAN, California Institute of Technology
ALISTAIR SINCLAIR, University of California, Berkeley

We study a classical iterative algorithm for balancing matrices in the L_∞ norm via a scaling transformation. This algorithm, which goes back to Osborne and Parlett & Reinsch in the 1960s, is implemented as a standard preconditioner in many numerical linear algebra packages. Surprisingly, despite its widespread use over several decades, no bounds were known on its rate of convergence. In this article, we prove that, for any irreducible $n \times n$ (real or complex) input matrix A, a natural variant of the algorithm converges in $O(n^3 \log(n\rho/\varepsilon))$ elementary balancing operations, where ρ measures the initial imbalance of A and ε is the target imbalance of the output matrix. (The imbalance of A is $\max_i |\log(a_i^{\text{out}}/a_i^{\text{in}})|$, where $a_i^{\text{out}}, a_i^{\text{in}}$ are the maximum entries in magnitude in the ith row and column, respectively.) This bound is tight up to the $\log n$ factor. A balancing operation scales the ith row and column so that their maximum entries are equal, and requires $O(m/n)$ arithmetic operations on average, where m is the number of nonzero elements in A. Thus, the running time of the iterative algorithm is $\widetilde{O}(n^2 m)$. This is the first time bound of any kind on any variant of the Osborne-Parlett-Reinsch algorithm. We also prove a conjecture of Chen that characterizes those matrices for which the limit of the balancing process is independent of the order in which balancing operations are performed.

CCS Concepts: • **Mathematics of computing → Mathematical analysis**; • **Theory of computation → Design and analysis of algorithms;**

Additional Key Words and Phrases: Numerical linear algebra, preconditioning matrices, matrix balancing, matrix scaling

ACM Reference Format:
Leonard J. Schulman and Alistair Sinclair. 2017. Analysis of a classical matrix preconditioning algorithm. J. ACM 64, 2, Article 9 (May 2017), 23 pages.
DOI: http://dx.doi.org/10.1145/2988227

1. INTRODUCTION

1.1. Background and Discussion of Results

In numerical linear algebra, it is standard practice to *precondition* an $n \times n$ real or complex matrix A by performing a similarity transform $D^{-1}AD$ for some diagonal scaling matrix D, prior to performing other computations on A such as computing its eigenvalues. The goal is that $D^{-1}AD$ should be *balanced*, in the sense that the norm of the ith row is equal to the norm of the ith column, for all i. The point here is that standard linear algebra algorithms tend to be numerically unstable for unbalanced matrices. Diagonal scaling achieves balance without affecting the eigenvalues of A.

L. J. S. was supported in part by NSF grants 1038578, 1319745, and 1618795 and by the Simons Institute for the Theory of Computing, where most of this work was performed. A. S. was supported in part by NSF grants 1016896 and 1420934. A preliminary version of this work, with weaker results, appeared in STOC'15 [Schulman and Sinclair 2015a].
Authors' addresses: L. J. Schulman, California Institute of Technology, 1200 E. California Blvd., MC 305-16, Pasadena, CA 91125; email: schulman@caltech.edu; A. Sinclair, Computer Science Division, University of California, Soda Hall, Berkeley CA 94720-1776; email: sinclair@cs.berkeley.edu.
Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or permissions@acm.org.
© 2017 ACM 0004-5411/2017/05-ART9 $15.00
DOI: http://dx.doi.org/10.1145/2988227

(Preconditioning typically also involves a separate process of row and column permutations, which we ignore here.)

This idea goes back to Osborne [1960], who suggested an iterative algorithm for finding a D that balances A in the L_2 norm, and proved that it converges in the limit; he also proposed using the analogous iteration in the L_∞ norm. Parlett and Reinsch [1969] generalized the algorithm to other norms. Neither of these papers gave any bound on the convergence time of the algorithm (in any norm). In the decades since then, this algorithm has been implemented as standard in almost all numerical linear algebra software, including EISPACK, LAPACK, and MATLAB. For further background, see, for example, Trefethen and Embree [2005] and Kressner [2005]. Surprisingly, despite its widespread use in practice, no bounds are known on the running time of any variant of this iterative balancing algorithm.

Our goal in this article is to initiate a quantitative study of the Osborne-Parlett-Reinsch algorithm. We emphasize that our motivation is to understand an existing method that has emerged as the leading choice of practitioners, rather than to devise a new competitor in the asymptotic regime; however, the bounds we obtain show that even asymptotically this method is not far off the theoretically best (and much more complex) algorithms (see the Related Work section for a discussion).

The iterative algorithm is very easy to describe, and involves repeated execution of a simple local balancing operation. Let $\|\cdot\|$ be a vector norm. For an index $i \in [n]$, let $\|a_{i\cdot}\|$ and $\|a_{\cdot i}\|$ denote the norms of the ith row and ith column of A, respectively. To avoid technicalities, we make the standard assumption that A is irreducible (see below), which ensures these norms are always nonzero; otherwise, earlier (and faster) stages of the preconditioning decompose A into irreducible components. A *balancing operation at i* scales each entry in column i of A by $r_i = \sqrt{\|a_{i\cdot}\|/\|a_{\cdot i}\|}$, and each entry in row i by r_i^{-1} (thus leaving a_{ii} unchanged). Note that this operation corresponds to the diagonal transformation $D^{-1}AD$, where $D = \mathrm{diag}(1, \ldots, r_i, \ldots, 1)$. The iterative algorithm simply performs the following step repeatedly:[1] *Pick an index i and perform a balancing operation at i.*

We focus in this article on the case of balancing in the L_∞ norm, where the goal is to find D so that in $D^{-1}AD$ the largest entry (in magnitude) in the ith row is equal to the largest entry in the ith column, for all i. We refer to such a matrix as *balanced*. The scaling factor in the algorithm is then just $r_i = \sqrt{a_i^{\mathrm{out}}/a_i^{\mathrm{in}}}$, where $a_i^{\mathrm{out}} = \max_j |a_{ij}|$ and $a_i^{\mathrm{in}} = \max_j |a_{ji}|$ are the maximum elements in the ith row and column, respectively. This version of the algorithm is particularly simple to implement, which makes it an attractive alternative to (say) the L_2 version. Since the algorithm depends only on the magnitudes of the a_{ij}, and multiplies them only by positive reals, we will simplify from now on by assuming that all entries a_{ij} are non-negative, keeping in mind that the balancing operations are actually performed on the original matrix A.[2]

It is instructive to view this procedure in terms of the directed weighted graph G_A, in which there is an edge (i, j) of weight a_{ij} if $a_{ij} > 0$ (and no edge if $a_{ij} = 0$). Irreducibility of A corresponds to G_A being strongly connected. The above balancing operation

[1]Note that we leave open here the rule by which the index i is chosen in each step. The original algorithm in Osborne [1960] and Parlett and Reinsch [1969] picks indices in a fixed cyclic order; in this article, we shall consider both that variant and the natural variant in which indices are picked uniformly at random with replacement (u.a.r.), and obtain essentially the same convergence rate results for both.

[2]It is sometimes also assumed that the diagonal entries of A are replaced by zeros before the balancing process, since balancing never alters them. This replacement may of course change the problem since (e.g.) a diagonally dominant matrix is already balanced in L_∞. We do not assume such a replacement has been made; our analysis of the balancing algorithm applies whether or not diagonal entries are zero.

$$A = \begin{pmatrix} 0 & 2 & 0 & 0 \\ 8 & 0 & 2 & 0 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 8 & 0 \end{pmatrix}; \qquad B_1 = \begin{pmatrix} 0 & 4 & 0 & 0 \\ 4 & 0 & 2 & 0 \\ 0 & 1 & 0 & 4 \\ 0 & 0 & 4 & 0 \end{pmatrix}; \qquad B_2 = \begin{pmatrix} 0 & 4 & 0 & 0 \\ 4 & 0 & 1 & 0 \\ 0 & 2 & 0 & 4 \\ 0 & 0 & 4 & 0 \end{pmatrix}$$

Fig. 1. Example of a non-UB matrix A. The balanced matrix B_1 results from A by balancing at indices 1,4, while B_2 results by balancing at indices 2,4.

scales all incoming (respectively, outgoing) edge weights at vertex i by $r_i = \sqrt{a_i^{\text{out}}/a_i^{\text{in}}}$ (r_i^{-1}, respectively), where a_i^{out}, a_i^{in} are the maximum outgoing and incoming weights, respectively, at i. At first sight (taking logarithms of the edge weights), this process resembles a diffusion on G_A; however, the fact that the scaling at each step depends on the *maximum* incoming/outgoing edge weight is a nonlinearity that makes the process much harder to analyze.

Clearly, any fixed point of the above iteration must be a balanced matrix A, that is, $a_i^{\text{out}} = a_i^{\text{in}}$ for all i; in other words, in the weighted graph G_A, the maximum incoming edge weight is equal to the maximum outgoing edge weight at every vertex. Note that the product of the edge weights along any cycle in G_A is an invariant of the algorithm. It is thus natural to call two matrices *equivalent* if they agree on all these invariants. The fact that the algorithm always converges asymptotically to a balanced matrix was proved, surprisingly recently, by Chen [1998][3]; in particular, this implies the existence of a balanced matrix in each equivalence class. Chen also conjectured that a worst-case input (in terms of rate of convergence) is one in which G_A is simply a directed cycle; in this case each vertex has only one incoming and one outgoing edge, so the balancing operation is linear and convergence is easily seen to occur within $\Theta(n^3)$ operations. (The Θ here hides factors that depend on the maximum imbalance in the input matrix and the desired bound on how close the output matrix is to being balanced.) In most cases the process seems to converge much faster, but the problem of analyzing the convergence rate in general graphs, or even in any strongly connected graph other than a directed cycle, has remained open until now. For example, even the case of two directed cycles that share a common vertex is quite nontrivial and has proved resistant to standard arguments.

The first difficulty one faces in analyzing convergence rates is that the balanced matrix to which the algorithm converges is in general not uniquely defined, but may depend on the sequence of indices i selected (see Figure 1 for a simple example that illustrates this phenomenon). Our first result proves a conjecture of Chen [1998], which characterizes all cases in which the balancing problem has a unique solution. For a balanced matrix $B = \{b_{ij}\}$ and any real $w > 0$, let G_B^w denote the subgraph of G_B consisting of those edges of magnitude at least w, with isolated vertices removed. (A vertex with a self-loop is not considered isolated.)

THEOREM 1.1. *A balanced matrix B is the unique balanced matrix in its equivalence class if and only if G_B^w is strongly connected for all w.*

Note that this result gives an implicit criterion for whether a given input matrix A can be uniquely balanced: namely, that it is equivalent to a balanced matrix B with

[3]Using the methods developed in this article, we give an alternative proof of this fact: see Theorem 4.5 at the end of Section 4. Both proofs apply to any sequence of balancing operations that is "fair" (see later for a definition). Our proof of convergence has the advantage that it also gives a (very weak, but so far the only known) bound on the rate of convergence for any fair sequence.

the above property. We say that such matrices A satisfy the *Unique Balance (UB)* condition.

In an earlier version of the present article [Schulman and Sinclair 2015a], we focused on the UB case and showed that a certain variant of the Osborne-Parlett-Reinsch algorithm converges in $O(n^3 \log n)$ balancing operations on any UB input matrix A. In this article, we extend that analysis to all input matrices, for a slightly different variant of the algorithm. (The term "variant" here refers only to the order in which balancing operations are performed.)

To describe this variant, we need the notion of "raising" and "lowering" operations, which are one-sided versions of the standard balancing operation. A *raising operation* at i performs a standard balancing operation at i if $a_i^{\text{out}} > a_i^{\text{in}}$ and does nothing otherwise; similarly, a *lowering operation* at i performs a standard balancing operation at i if $a_i^{\text{out}} < a_i^{\text{in}}$ and does nothing otherwise. Our variant of the algorithm will perform a sequence of raising operations followed by a sequence of lowering operations; equivalently, it can be viewed as the standard algorithm in which some of the balancing operations are *censored* according to a very simple condition.

To specify the algorithm precisely, we also need a measure of how far a matrix is from balanced. We thus define the *imbalance* of A as $\max_i |\log(a_i^{\text{out}}/a_i^{\text{in}})|$, and say that A is ε-*balanced* if its imbalance is at most ε. The target balance parameter $\varepsilon > 0$ is provided to the algorithm as an additional input. We also need one-sided versions of these definitions: namely, A is ε-*raising-balanced* if $\max_i \log(a_i^{\text{out}}/a_i^{\text{in}}) \le \varepsilon$ (or *raising-balanced* if $\varepsilon = 0$), and ε-*lowering-balanced* if $\max_i \log(a_i^{\text{in}}/a_i^{\text{out}}) \le \varepsilon$ (or *lowering-balanced* if $\varepsilon = 0$). Plainly, A is ε-balanced if and only if it is both ε-raising-balanced and ε-lowering-balanced.

ALGORITHM 1: 2PhaseBalance(A, ε)

Input: An $n \times n$ matrix A with (unknown) imbalance ρ; and a target $\varepsilon > 0$.
Output: A matrix equivalent to A that is ε-balanced with high probability.

```
1 Compute ρ in time O(n^2)
2 Repeat T = O(n^3 log(ρ/ε)) times                    //Raising Phase//
3     Pick an index i and apply a raising operation at i
4 Repeat T = O(n^3 log(ρ/ε)) times                    //Lowering Phase//
5     Pick an index i and apply a lowering operation at i
6 Output the resulting matrix
```

Our variant of the algorithm is specified in Algorithm 1. We note that even the asymptotic convergence of this variant of the algorithm is not immediately clear, and does not follow from Chen's proof [Chen 1998] or from our proof in Theorem 4.5. This is because, as indicated earlier, those proofs assume that the algorithm is *fair*, in the sense that balancing is performed at each index infinitely often. In the two-phase version, fairness fails since some balancing operations are censored; in particular, a "fair" sequence of *raising* operations is not a fair sequence of balancing operations, since the lowering operations are censored. Instead, convergence of the two-phase algorithm is a consequence of the following theorem.

THEOREM 1.2. *For any input matrix A, any fair sequence of raising (respectively, lowering) operations converges to a unique raising-balanced (respectively, lowering-balanced) matrix, independent of the order of the operations.*

In lines 3 and 5 of Algorithm 1, the indices i are picked either in deterministic cyclic order (as in the original algorithm in Osborne [1960] and Parlett and Reinsch [1969]), or u.a.r.; note that both these sequences are fair (with probability 1), so Theorem 1.2

implies that each phase converges asymptotically. The value T is chosen to guarantee that A is ε-raising-balanced (respectively, ε-lowering-balanced) at the end of a phase.[4] Since it is easy to see that the lowering phase cannot increase the raising imbalance of A (see Proposition 4.1), at the conclusion of both phases A will indeed be ε-balanced (w.h.p.).

Theorem 1.2 is actually a somewhat surprising result: even when A is not UB (so that the balanced matrix to which the original version of the algorithm converges depends on the sequence of operations performed), if we perform only raising (or only lowering) operations we are guaranteed a unique limit. It is this fact that ultimately enables us to obtain a tight analysis of the rate of convergence.

We are now able to state the main result of the article, which bounds the rate of convergence of Algorithm 1.

THEOREM 1.3. *Suppose Algorithm 1 is run on input* (A, ε)*, where* A *is an arbitrary irreducible nonnegative matrix with imbalance at most* ρ *and the indices in lines 3,5 are picked in deterministic cyclic order (respectively, picked u.a.r.). Then the algorithm performs* $O(n^3 \log(n\rho/\varepsilon))$ *balancing operations and outputs a matrix equivalent to* A *that is* ε*-balanced (respectively,* ε*-balanced w.h.p.).*

This is the first time bound (except on the cycle) for any variant of the Osborne-Parlett-Reinsch algorithm (under any norm). Moreover, the bound on the convergence rate is actually tight up to a factor $O(\log n)$, in view of the $\Omega(n^3)$ lower bound we mentioned earlier for the cycle. Indeed, Theorem 1.3 implies that the cycle is a worst-case input for the algorithm (up to the $\log n$ factor), as conjectured by Chen [1998].

Remark. Throughout the article, we quantify the rate of convergence of the algorithm in terms of the number of balancing operations performed. Since each balancing operation (raising or lowering) involves finding a maximum element in one row and column and then scaling the row and column, it can be performed on average in $O(m/n)$ arithmetic operations, where m is the number of edges in G_A (i.e., the number of nonzero entries in A). Thus, in terms of arithmetic operations, the (average) running time of the algorithm is $O(n^2 m \log n)$.

We close this section with a brief outline of our approach to analyzing the algorithm. As we have observed previously, it suffices to analyze only the raising phase; by symmetry, an identical analysis applies to the lowering phase.[5] It is convenient for the analysis to take logarithms of the matrix entries, so that balancing operations are additive rather than multiplicative. The first key observation is that the uniqueness of the limit in the raising phase allows us to assign a well-defined *height* to every vertex in G_A; the height of a vertex measures the "amount of raising" that needs to be performed at that vertex in order to reach the unique raising-balanced configuration. Each raising operation can then be viewed as smoothing these heights locally. In a diffusion process, this naturally leads to an associated Laplacian potential function consisting of the sum of squares of local height differences, whose convergence is captured by the eigenvalues of the heat kernel operator. Attempts to conduct a similar analysis here fail. One serious problem is that a balancing operation at vertex i depends only on the incoming and outgoing edges of *maximum* weight, but has side effects on all other edges incident at i. Another problem is that there are, essentially, "levels" within the graph: roughly speaking, vertices that lie in the "lower" levels of the graph cannot

[4]In the case where indices are picked u.a.r., these properties hold *with high probability (w.h.p.)*, that is, with probability tending to 1 as $n \to \infty$.

[5]It is an interesting open question whether the introduction of phases in the Osborne-Parlett-Reinsch algorithm leads to more rapid convergence in practice.

reliably converge toward (raising) balance until the vertices in the "higher" levels have converged. While phenomena similar to this can arise in diffusion processes, in our nonlinear setting standard tools such as eigenvalues are lacking to capture them.

The chief technical challenge in our analysis is to relate the local height changes achieved by raising operations to an improvement in a global potential function (the sum of the heights), thus ensuring significant progress over time. The details of our analysis ensure an improvement that is a factor $\Omega(n^{-3})$ of the current value of the potential function in each step (amortized or expected, depending on the version of the algorithm), thus leading to global convergence in $\tilde{O}(n^3)$ raising operations.

To establish the above local-global connection, we need a two-dimensional representation of the current state of the algorithm, which records not only the height of a vertex but also its *level*; unlike the heights, which change over time, the level of a vertex is fixed. We also introduce an intermediate quantity that we call the *momentum* of a vertex, which can be viewed as a time-dependent notion of its level. The momentum is a key ingredient in relating local improvements to the global potential function.

1.2. Related Work

As mentioned previously, the idea of iterative diagonal balancing was introduced by Osborne [1960], who was motivated by the observation that minimizing the Frobenius norm of A is equivalent to balancing A in the L_2 norm. Osborne formulated an L_2 version of the iterative algorithm and proved that it converges in the limit (but with no rate bound); he also proposed the L_∞ algorithm discussed in this paper but left the question of convergence open. Convergence was first proved by Chen [1998] almost 40 years later. Parlett and Reinsch [1969] generalized Osborne's algorithm to other norms (without proving convergence) and discussed a number of practical implementation issues for preconditioning. For the L_1 version of the Osborne-Parlett-Reinsch algorithm, convergence was proved by Grad [1971], uniqueness of the balanced matrix by Hartfiel [1971], and a characterization of it by Eaves et al. [1985], but again no bounds on the running time of L_1 balancing were given.

Despite the scarcity of theoretical support, these algorithms are implemented as standard in many numerical linear algebra packages and experience shows them to be both useful and fast.

A sequence of two papers offers an alternative, and substantially more complex, noniterative algorithm for matrix balancing in the L_∞ norm. Schneider and Schneider [1991] gave an $O(n^4)$-time algorithm based on finding maximum mean-weight cycles in a graph; the running time was improved to $O(nm + n^2 \log n)$ using Fibonnaci heaps and other techniques by Young et al. [1991]. This is asymptotically faster than the $\widetilde{O}(n^2 m)$ worst-case running time we establish for the iterative algorithm; and for some graphs (e.g., the directed cycle) it is faster than the actual running time of the iterative algorithm.

However, the iterative method has been favored in practice. This may be justified by the empirical distribution of inputs. Moreover, it is definitely driven by the fact that the iterative method offers steady partial progress, and so can deliver, without being run to completion, a matrix that is sufficiently balanced for the subsequent linear algebra computation. In practice, indeed the method is usually run for far fewer iterations than are needed in the worst case. Our purpose in this article is to provide the first theoretical understanding of the widely used iterative methods, rather than to derive new theoretical bounds for the underlying problem.

In other work on matrix balancing, Kalantari et al. [1997] considered balancing in the L_1 norm, and provided the first polynomial time algorithm by reduction to convex programming: $\widetilde{O}(n^4)$ via the ellipsoid method. Following the appearance of

earlier versions of the current article [Schulman and Sinclair 2015a, 2015b], Ostrovsky et al. [2017] gave a convex-programming-based analysis of the L_1 variant of the iterative algorithm. We note that this approach apparently fails in our case, in part because the set of L_∞-balanced matrices for a given input may fail to be convex.

Diagonal scaling has also been used to minimize matrix norms without regard to balancing. For example, Ström [1972] considers the problem of finding diagonal scaling matrices to minimize the max and Frobenius norms of the matrix. In particular, for the max norm he proves that (when A is irreducible) the optimal diagonal scaling matrix is obtained from the principal eigenvector. Chen and Demmel [2000] show that a suitable notion of *weighted* balancing can be used to minimize the 2-norm, and discuss Krylov-based algorithms that work efficiently on sparse matrices. Boyd et al. [1994] formulate the problem of minimizing the Frobenius norm as a generalized eigenvalue problem.

A different notion of matrix balancing asks for diagonal scaling matrices D_1 and D_2 such that D_1AD_2 has prescribed row and column sums. A natural iterative process, somewhat analogous to the Osborne-Parlett-Reinsch process, appears to have been invented several times independently, and is often named after Sinkhorn, who gave the first analysis of its convergence [Sinkhorn 1964]. The process has been extensively studied (see, e.g., Sinkhorn and Knopp [1967], Franklin and Lorenz [1989], Kalantari and Khachiyan [1993], and Kalantari et al. [2008]); in particular, Linial et al. [2000] gave strongly polynomial bounds (after a preprocessing step) and derived a surprising approximation scheme for the permanent of a nonnegative matrix. For much more on this topic, see the survey by Idel [2016].

2. PRELIMINARIES

In this section, we introduce some terminology and notation that we will use throughout the article. We begin with an equivalent reformulation of the Osborne-Parlett-Reinsch balancing algorithm from the Introduction.

Let A be an irreducible $n \times n$ matrix. As described in the Introduction, the algorithm operates by iteratively picking an index i and geometrically averaging the maximum entries (in magnitude) a_i^{out} and a_i^{in}. It is convenient to switch to arithmetic averages by setting $\alpha_{ij} = \log |a_{ij}|$ (so that 0 entries of A become $-\infty$). Letting $\alpha_i^{\text{out}} = \max_j \alpha_{ij}$ and $\alpha_i^{\text{in}} = \max_j \alpha_{ji}$, a *balancing operation at* i then consists of adding $(\alpha_i^{\text{out}} - \alpha_i^{\text{in}})/2$ to the ith column and subtracting the same quantity from the ith row. (In the actual implementation of the algorithm, this corresponds to the multiplicative diagonal transformation $D = \text{diag}(1, \ldots, r_i, \ldots, 1)$ applied to the original matrix A. However, for the rest of the article, we will assume that the algorithm operates additively on the logarithms of the matrix entries and ignore the details of this trivial translation.)

We reuse the notation G_α from the Introduction to denote the digraph with edges (i, j) such that $\alpha_{ij} > -\infty$. (The passage from roman to greek letters will always resolve this abuse of notation.) Irreducibility of A again corresponds to G_α being strongly connected. We shall regard α as a real-valued function defined only on the edges of G_α, and call it a *graph function*. Note that if balancing operations change the graph function α to α', we always have $G_\alpha = G_{\alpha'}$. For the remainder of the article, we use letters u, v, etc., rather than i, j to denote vertices as our focus will be on the graph G_α rather than on the matrix A.

We say that a graph function α is *balanced* if $\alpha_v^{\text{out}} = \alpha_v^{\text{in}}$ for all v. For each v, define $\rho_v = |\alpha_v^{\text{out}} - \alpha_v^{\text{in}}|$. The *imbalance* of α is $\rho := \max_v \rho_v$, and α is ε-*balanced* if $\rho \leq \varepsilon$. (These definitions are equivalent to those in the Introduction in terms of the matrix entries a_{ij}.) A balancing operation is available at v if and only if $\rho_v > 0$.

As indicated in the Introduction, we analyze a slightly modified version of the basic Osborne-Parlett-Reinsch algorithm, which has two phases and performs only certain balancing operations (raising or lowering) during a phase. Let $\rho_v^R = \max\{0, \alpha_v^{\text{out}} - \alpha_v^{\text{in}}\}$ and $\rho_v^L = \max\{0, \alpha_v^{\text{in}} - \alpha_v^{\text{out}}\}$ be the *raising* and *lowering imbalances*, respectively, at v. Note that $\rho_v = \rho_v^R + \rho_v^L = \max\{\rho_v^R, \rho_v^L\}$, since one of ρ_v^R, ρ_v^L must be zero. A *raising operation* at v performs a standard balancing operation at v if $\rho_v^R > 0$ and does nothing otherwise; similarly, a lowering operation at v performs a standard balancing operation at v if $\rho_v^L > 0$ and does nothing otherwise. Such an operation is said to *raise* (respectively, *lower*) v by an amount $\rho_v^R/2$ (respectively, $\rho_v^L/2$); note that a raising operation at v reduces its raising imbalance to zero unless the maximum outgoing edge from v is a self-loop, in which case the raising imbalance is reduced by a factor of 2. The *raising imbalance* of α is $\rho^R := \max_v \rho_v^R$, and the *lowering imbalance* of α is $\rho^L := \max_v \rho_v^L$. We say that α is ε-*raising-balanced* if $\rho^R \le \varepsilon$, and ε-*lowering-balanced* if $\rho^L \le \varepsilon$. Clearly, α is ε-balanced if and only if it is both ε-raising-balanced and ε-lowering-balanced.

For any real w, denote by G_α^w the subgraph of G_α consisting of edges (u, v) such that $\alpha_{uv} \ge w$, with isolated vertices removed. (A vertex with a self-loop is not considered isolated.)

Let α, γ be two graph functions on the same graph G. We say that α, γ are *equivalent*, $\alpha \sim \gamma$, if their sums around all cycles of G are equal, i.e., if $\sum_{\ell=1}^{k}(\alpha_{v_\ell, v_{\ell+1}} - \gamma_{v_\ell, v_{\ell+1}}) = 0$ for all cycles $(v_1, \ldots, v_k)$, where $v_{k+1} = v_1$.

It is convenient for our analysis to introduce a generalization of the standard local balancing operation at a vertex. Let α be a graph function. For any $x \in \mathbb{R}$ and $S \subseteq [n]$, denote by $\alpha + xS$ ("α shifted by x at the set S") the graph function given by $(\alpha + xS)_{uv} = \alpha_{uv} + x \cdot (I_S(u) - I_S(v))$, where I_S is the indicator function for membership in S. Thus, the balancing operation at v is equivalent to shifting by $(\alpha_v^{\text{in}} - \alpha_v^{\text{out}})/2$ at the set $\{v\}$. It is readily seen that two graph functions are equivalent if and only if there is a sequence of shifts converting one to the other. Note that, like balancing operations, shifts of α do not change G_α.

We close this section with a simple but useful fact, which says that the maximum and minimum edge weights remain bounded under any sequence of balancing operations.

PROPOSITION 2.1. *Given any graph function α, there are finite values $b_{\max}$, $b_{\min}$ such that, under any sequence of balancing operations, all edge weights remain in the interval* $[b_{\min}, b_{\max}]$.

PROOF. Let $\alpha_{\max} = \max_{(u,v)\in G_\alpha} \alpha_{uv}$ and $\alpha_{\min} = \min_{(u,v)\in G_\alpha} \alpha_{uv}$; the definition of graph function ensures that both are finite.

A balancing operation at vertex v reduces $\max\{\alpha_v^{\text{in}}, \alpha_v^{\text{out}}\}$, so no such operation can increase the maximum edge value above its original value $\alpha_{\max}$. Thus, we may take $b_{\max} = \alpha_{\max}$. Now recall that the average weight of any cycle in G_α remains invariant under any sequence of balancing operations; thus, in particular, the minimum average weight of any cycle is always at least $\alpha_{\min}$. Consider the value of any edge (u, v) under any graph function α' that is reached by balancing operations from α. Let (u, v) belong to some (simple) cycle of average weight c and length ℓ. Then $(\alpha'_{uv} + (\ell - 1)\alpha_{\max})/\ell \ge c \ge \alpha_{\min}$, so $\alpha'_{uv} \ge \alpha_{\min} - (\ell - 1)(\alpha_{\max} - \alpha_{\min})$. Thus, we may take $b_{\min}$ to be this last value. □

3. THE UNIQUE BALANCE PROPERTY

As shown by Chen [1998] (and reproved with a different argument in the present article—see Theorem 4.5), the iterative balancing algorithm is guaranteed to converge to a balanced matrix provided the sequence of vertices at which balancing is performed is *fair*, in the sense that every vertex appears infinitely often. However, the limit matrix is in general not unique and depends on the sequence (see Figure 1 for a simple

example). This nonuniqueness is one major obstacle to obtaining good bounds on the rate of convergence.

For a broad class of inputs, however, a unique limit does exist. The characterization of this class was conjectured by Chen [1998] and is our first result. This was already stated as Theorem 1.1 in the Introduction; we restate it here in slightly different notation.

THEOREM 1.1. *A balanced graph function β is the unique balanced graph function in its equivalence class if and only if G_β^w is strongly connected for all w.*

PROOF. Let β be balanced. Suppose G_β^w is not strongly connected, and that w is the largest value for which this is the case. Note that since β is balanced, G_β^w has no sources or sinks, and hence must contain at least one source strongly connected component (scc) S and one sink scc T, both of which are nontrivial (i.e., contain at least two vertices or one self-loop). Moreover, by maximality of w, the weights of external edges (between scc's) in G_β^w are all exactly w. Choose vertices $s \in S$ and $t \in T$, and consider a directed simple path P from s to t in G_β. (Such a path exists because G_β is strongly connected.) Let $\varepsilon > 0$ be smaller than the gap between any two distinct values of β. Then in $G_{\beta+\varepsilon T}$, the length of P is ε less than it is in G_β.

Now apply balancing operations in any order to the graph function $\beta + \varepsilon T$ (which is equivalent to β), to balance it (in the limit). We claim that no balancing operation will ever be applied at any vertex inside a nontrivial scc of G_β^w, and hence at either s or t. To see this, note that such a balancing operation can only occur if the weight of some external edge incident on the scc reaches a value larger than w. But in $\beta + \varepsilon T$, no external edge has weight larger than w (this was true in β and the shift at T has increased the weight only of edges leaving T, but to a value less than w), so balancing operations at the vertices outside the scc's cannot increase the weight of edges above w. Hence, in the limiting (and hence balanced) graph function, the length of the path P from s to t will remain ε less than it is in β. (The length of a path can be altered only by a balancing operation at one of its endpoints.) Hence, this limiting graph function is different from β.

Let $\alpha \sim \beta$ be distinct balanced graph functions. We show that there is a w such that neither G_β^w nor G_α^w is strongly connected.

Let w be the largest value such that $G_\beta^w \neq G_\alpha^w$ (the supremum is achieved as the graph is finite), and suppose without loss of generality there is an edge (u, v) in $G_\beta^w - G_\alpha^w$. Then G_β^w cannot be strongly connected because if it were, consider any cycle in G_β^w that includes (u, v). The weight of this cycle is larger in G_β than in G_α because all weights larger than w in the cycle are identical in both functions, and in G_β all remaining weights equal w, while in G_α all are at most w and at least one is strictly less than w. Hence $\alpha \not\sim \beta$, which is a contradiction.

Likewise, if there is an edge in $G_\alpha^w - G_\beta^w$, then G_α^w is not strongly connected. This leaves only the case $G_\alpha^w \subset G_\beta^w$. Let U be the set of vertices from which u can be reached in G_β^w, and V be the set of vertices reachable from v in G_β^w. As shown in the previous paragraph, these sets are disjoint. Each set must contain a nontrivial scc (because β is balanced and therefore G_β^w contains no sources or sinks). These scc's (call them U' and V') must also be strongly connected in G_α^w; in fact, we argue that α equals β on the edges of G_β^w within each scc. For, consider any cycle using edges of G_β^w in the scc. The cycle may contain edges heavier than w (which are identical in α and β), and some edges that are equal to w in β and $\leq w$ in α. But then since $\alpha \sim \beta$, these edges must also equal w in α. So α and β are identical on the edges of G_β^w in the scc. Now, if G_α^w is

strongly connected, then it contains a path from V' to U'. Then since $G^w_\alpha \subset G^w_\beta$, this path also lies in G^w_β—contradiction. □

In light of Theorem 1.1, we say that a graph function α has the *UB* property if there is a balanced $\beta \sim \alpha$ such that G^w_β is strongly connected for all w. By the convergence theorem (Chen [1998], or Theorem 4.5 of the present article), on any input matrix with the UB property the Osborne-Parlett-Reinsch balancing algorithm will converge to a unique balanced matrix. In a preliminary version of the present article [Schulman and Sinclair 2015a], we used this fact to analyze the rate of convergence of (a suitable variant of) the algorithm in the UB case. In the remainder of this article, we turn our attention to the rate of convergence for general input matrices (which will subsume the result of Schulman and Sinclair [2015a]), and leave Theorem 1.1 as an interesting structural result.

4. CONVERGENCE OF THE TWO-PHASE ALGORITHM

The goal of this section is to prove that the two-phase variant of the iterative balancing algorithm described in the Introduction converges and outputs a ε-balanced graph function. (As previously noted, convergence of this variant does not follow from the general convergence proofs of Chen [1998] or of Theorem 4.5 of the present article, since the sequence of balancing operations in this case is not "fair" due to the censoring of operations in each phase.)

Convergence of the two-phase variant follows immediately from Theorem 1.2 stated in the Introduction, which we restate here in graph function terms.

THEOREM 1.2. *For any initial graph function α, any fair sequence of raising (respectively, lowering) operations converges to a unique raising-balanced graph function α^R (respectively, lowering-balanced graph function α^L), independent of the order of the operations.*

To deduce that the output of the two-phase algorithm is indeed ε-balanced, we need the following additional simple observation.

PROPOSITION 4.1. *A lowering operation cannot increase any raising imbalance ρ^R_v. Symmetrically, a raising operation cannot increase any lowering imbalance ρ^L_v.*

PROOF. We give the proof only for lowering operations; the proof for raising operations is symmetrical. Consider a lowering operation at vertex v, and assume that it has a nonzero effect, i.e., $\rho^L_v > 0$. Lowering at v reduces ρ^L_v (to zero unless the maximum incoming edge at v is a self-loop in which case the reduction is to $\rho^L_v/2$) and keeps ρ^R_v at zero. The only other imbalances that can be affected are at neighbors of v. Lowering at v decreases edge weights α_{uv}, which cannot increase ρ^R_u. Similarly, lowering at v increases edge weights α_{vw}, which again cannot increase ρ^R_w. □

It is now easy to see that the two-phase algorithm terminates with an ε-balanced graph function. By Theorem 1.2 each of the two phases terminates with the appropriate balance condition. At the end of the raising phase the function is ε-raising balanced, and by Proposition 4.1 it remains so throughout the lowering phase. At the end of the lowering phase the resulting graph function is both ε-raising balanced and ε-lowering balanced, and hence ε-balanced as required.

The remainder of this section is devoted to proving Theorem 1.2. We focus on raising operations; the proof for lowering operations follows by a symmetrical argument.

We introduce the following notation. Fix the initial graph function α and a particular fair infinite sequence of raising operations. Let $\alpha = \alpha(0), \alpha(1), \ldots, \alpha(t), \ldots$ be the resulting sequence of graph functions obtained. Also, for each $t \geq 0$ let the *raising vector*

$r(t) = (r_v(t))$ record the cumulative amount by which each vertex has been raised. That is, $r(0) = 0$ and, if the tth raising operation is at vertex v, then $r_v(t) = r_v(t-1) + \rho_v^R/2$ and $r_u(t) = r_u(t-1)$ for all $u \neq v$. Note that the vector $r(t)$ completely specifies[6] $\alpha(t)$ via the relation

$$\alpha_{uv}(t) = \alpha_{uv}(0) + r_v(t) - r_u(t). \tag{1}$$

LEMMA 4.2. *The sequence $r(t)$ is increasing and bounded above, and hence converges to a limit vector r^* as $t \to \infty$.*

PROOF. The fact that $r(t)$ is increasing is immediate from its definition. To see that it is bounded, first recall from Proposition 2.1 that the graph function $\alpha(t)$ is bounded above and below for all t.

Fix an arbitrary total order $<_E$ on the edges of the graph. At any time t define a total order $<_t$ on the edges of the graph as follows: $(u_1, v_1) <_t (u_2, v_2)$ if $\alpha_{u_1v_1}(t) < \alpha_{u_2v_2}(t)$ or if $\alpha_{u_1v_1}(t) = \alpha_{u_2v_2}(t)$ and $(u_1, v_1) <_E (u_2, v_2)$. Let $N_{uv}(t)$ be the number of edges (u', v') such that $(u', v') \leq_t (u, v)$.

Now define the function

$$\Phi(t) = \sum_{(u,v)\in E, u\neq v} N_{uv}(t)!\,\alpha_{uv}(t). \tag{2}$$

We claim that, if the tth raising operation is at vertex v, the change in Φ is bounded by

$$\Phi(t+1) - \Phi(t) \leq -\rho_v^R/2. \tag{3}$$

Since by our earlier observations $\Phi(t)$ is always bounded, and the increase in $r_v(t)$ is exactly $\rho_v^R/2$, we conclude from Equation (3) that $r_v(t)$ is also bounded.

To prove Equation (3), represent each edge (u, v) by a particle on the real line at position $\alpha_{uv}(t)$. We will view $[t, t+1]$ as a continuous time interval, during which each particle (u, v) moves at constant speed from $\alpha_{uv}(t)$ to $\alpha_{uv}(t+1)$. We now compute the rate of change of Φ under a raising operation at vertex v (assuming that $\rho_v^R > 0$ so that the raising operation is nontrivial; otherwise, Φ does not change). The only particles that move are (u, v) and (v, w) as u, w range over neighbors of v; each particle (u, v) moves a distance $\rho_v^R/2$ to the right, and each (v, w) moves the same distance to the left. Let $\bar{u}, \bar{w}$ be such that $(\bar{u}, v)$ is maximal in the $<_t$ order among edges (u, v), while $(v, \bar{w})$ is maximal in the $<_t$ order among edges (v, w). Thus, $\rho_v^R = \alpha_{v\bar{w}}(t) - \alpha_{\bar{u}v}(t)$. The rate of change of $\Phi(t')$ for t' in the interval $[t, t+1)$ is given by

$$\begin{aligned}
\frac{\partial \Phi(t')}{\partial t'} &= \frac{\rho_v^R}{2}\left(\sum_{(u,v)} N_{uv}(t')! - \sum_{(v,w)} N_{vw}(t')!\right) \\
&\leq \frac{\rho_v^R}{2}\big((N_{v\bar{w}}(t')-1)(N_{v\bar{w}}(t')-1)! - N_{v\bar{w}}(t')!\big) \\
&\leq -\frac{\rho_v^R}{2}.
\end{aligned}$$

The negative term in the second line arises from the contribution of the particle $(v, \bar{w})$, whose edge remains heaviest throughout. The positive term arises because there are at most $N_{v\bar{w}}(t') - 1$ particles to the left of $(v, \bar{w})$, each of which contributes at most $(N_{v\bar{w}}(t') - 1)!$.

As the particles move and pass other particles, the coefficients $N_{uv}(t')$ change. However, observe that the preceding calculation is valid at all times $t' \in [t, t+1)$, since it

[6]However, the converse is not quite true: given $\alpha(t)$ we can deduce $r(t)$ only up to an additive constant.

relies only on the fact that $(v, \bar{w})$ is the moving particle that is greatest in the $<_{t'}$ order. Hence we deduce the bound (3) claimed earlier, which completes the proof. □

It follows immediately from Lemma 4.2 and Equation (1) that $\alpha(t)$ also converges to a limit α^*, and by continuity and fairness α^* must be raising-balanced. It remains to show that α^* is independent of the sequence of raising operations. For this we require the following monotonicity property of raising operations.

LEMMA 4.3. *Let r, s be raising vectors obtained from a common initial graph function α via different sequences of raising operations. Suppose that $r \le s$, and let r', s' be the new raising vectors obtained from r, s after one additional raising operation at a common vertex v. Then $r' \le s'$.*

PROOF. Clearly, we have $r'_u = r_u \le s_u = s'_u$ for all $u \ne v$, so we need only show that $r'_v \le s'_v$. By definition of the raising operation at v, we have (using Equation (1))

$$\begin{aligned} r'_v &= r_v + \frac{\rho_v^R}{2} \\ &= r_v + \frac{1}{2}\max\Big\{0, \max_{w:v\to w}(\alpha_{vw} + r_w - r_v) - \max_{u:u\to v}(\alpha_{uv} + r_v - r_u)\Big\} \\ &= \frac{1}{2}\max\Big\{2r_v, \max_{w:v\to w}(\alpha_{vw} + r_w) + \min_{u:u\to v}(r_u - \alpha_{uv})\Big\}. \end{aligned}$$

(Here, and subsequently, we use the notation $u \to v$ to indicate that there is an edge (u, v) in G_α.) Since the right-hand side here is monotonically increasing in all entries of r, and $r \le s$, we conclude that $r'_v \le s'_v$ as required. □

LEMMA 4.4. *For any initial graph function α and sequence of raising vectors $r(1), r(2), \ldots, r(t), \ldots$ obtained from a fair sequence of raising operations, the limit $r^* = \lim_{t\to\infty} r(t)$ exists and depends only on α, not on the sequence of raising operations.*

PROOF. Consider some fair sequence of raising operations starting from α, and let $\alpha(t)$, $r(t)$ be the corresponding sequences of graph functions and raising vectors, respectively. By Lemma 4.2, the limits $\alpha^* = \lim_{t\to\infty} \alpha(t)$ and $r^* = \lim_{t\to\infty} r(t)$ exist and α^* is raising-balanced.

Now consider a second sequence of raising operations also starting from α, and let $s(t)$ be the corresponding sequence of raising vectors. Again, by Lemma 4.2, the limit $s^* = \lim_{t\to\infty} s(t)$ exists.

Suppose we apply this second sequence of raising operations, starting on the one hand from α and on the other hand from α^*. The resulting raising vectors (with respect to α) are s^* and r^*, respectively (the latter because α^* is already raising-balanced). Applying Lemma 4.3 to compare the raising vectors at each step of this process yields $s^* \le r^*$. But by symmetry we must also have $r^* \le s^*$, and hence $r^* = s^*$. This completes the proof. □

Since the graph function α^* is completely determined by the raising vector r^*, Lemma 4.4, and its symmetric counterpart for lowering operations, proves Theorem 1.2, which was the main goal of this section.

Figure 2 shows an example of a graph function and its associated unique corresponding raising- and lowering-balanced functions.

We end this section with a theorem that uses the preceding technology to derive a new proof of convergence of the original Osborne-Parlett-Reinsch algorithm (in which there are no phases, and balancing operations are performed regardless of whether they are raising or lowering). This proof applies for any sequence of balancing operations provided only that it is *fair*, i.e., every index i appears infinitely often; note that

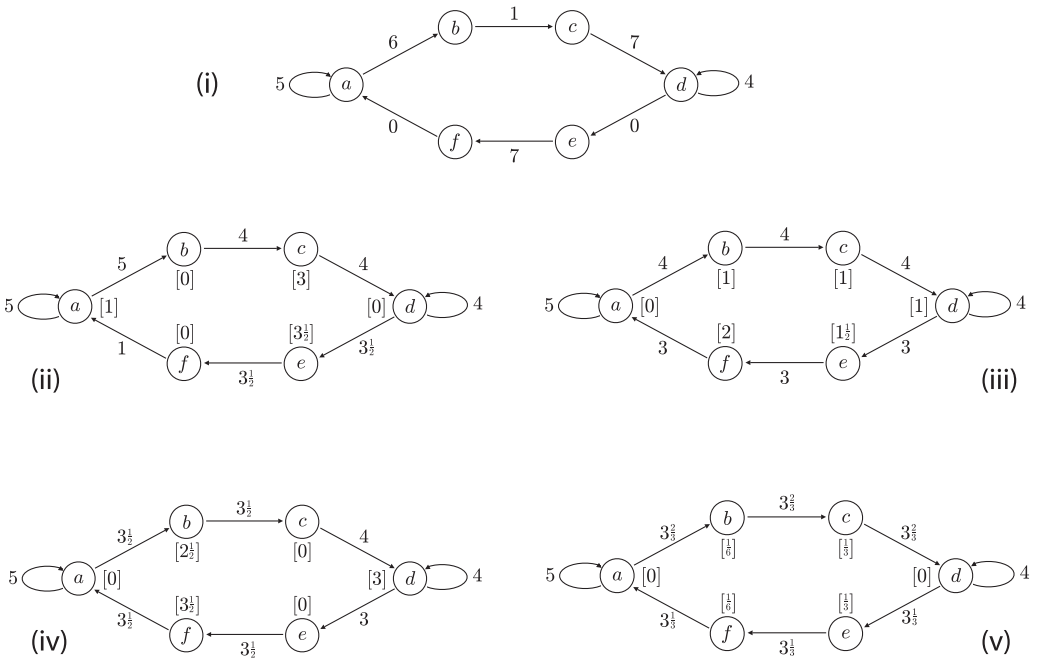

Fig. 2. Example showing convergence of raising and lowering operations. (i) A graph function α. (ii) The raising-balanced graph function α^R obtained in the limit from α by raising operations; numbers in square brackets at the vertices show the limiting raising vector r^*. (iii) The balanced graph function $(\alpha^R)^L$ obtained in the limit by applying lowering operations to α^R, together with its associated lowering vector. (This is the unique balanced graph function to which the two-phase algorithm "converges" if both phases are run to convergence.) Figures (iv) and (v) show the graph functions α^L and $(\alpha^L)^R$, respectively (and their associated lowering/raising vectors), obtained by reversing the order of the raising and lowering phases. Note that the balanced graph functions $(\alpha^R)^L$ (in (iii)) and $(\alpha^L)^R$ (in (v)) are not the same; while both versions (raising-lowering and lowering-raising) converge to a unique balanced graph function, these limits are not equal. The graph function α is not UB.

both the cyclic sequence and the u.a.r. sequence are fair (with probability 1). In contrast to the earlier proof of convergence by Chen [1998], which is not quantitative, our proof also provides an (albeit very weak) bound on the number of balancing operations required to reach an almost balanced matrix. In the next section, we will obtain a tight polynomial time bound on this number for the two-phase variant of the algorithm.

Given a fair sequence of indices, define the first *epoch* to be the minimal initial segment of the sequence that contains each index $1, 2, \ldots, n$ at least once. Define subsequent epochs in the same way. All epochs are of finite length, and there are infinitely many epochs.

THEOREM 4.5. *Consider the Osborne-Parlett-Reinsch L_∞ balancing algorithm applied to any irreducible (real or complex) input matrix A, and assume that balancing operations are performed according to any fair sequence. Let $A(t)$ denote the matrix obtained after t steps of the algorithm. Then $A(t)$ converges to a balanced matrix B (which in general depends on the sequence of balancing operations). Moreover, if the number of epochs in the first t balancing operations is at least $4nm \cdot m! \cdot (\alpha_{\max} - \alpha_{\min})/\varepsilon$, then for some $t' \leq t$, $A(t')$ is ε-balanced. (Here m is the number of nonzero entries in A and α is the graph function corresponding to A.)*

The proof is deferred to Appendix A.

5. RATE OF CONVERGENCE

In this section, we will prove the main result of the article, Theorem 1.3 from the Introduction, which gives a tight bound on the rate of convergence of the iterative balancing algorithm. We restate the result here for convenience.

THEOREM 1.3. *Suppose Algorithm 1 is run on input* (A, ε)*, where* A *is an arbitrary irreducible nonnegative matrix with imbalance at most* ρ *and the indices in lines 3,5 are picked in deterministic cyclic order (respectively, picked u.a.r.). Then the algorithm performs* $O(n^3 \log(n\rho/\varepsilon))$ *balancing operations and outputs a matrix equivalent to* A *that is* ε*-balanced (respectively,* ε*-balanced w.h.p.).*

We give here a brief, high-level outline of our analysis. First note that, by Proposition 4.1 in the previous section, it suffices to analyze each phase of the algorithm separately. Accordingly, we will analyze only the raising phase, showing that it achieves an ε-raising-balanced graph function after $O(n^3 \log(\rho/\varepsilon))$ raising operations. By symmetry, the same bound holds for the lowering phase: simply observe that lowering on the graph G_α is equivalent to raising on the transpose graph G_α^T.

In the Introduction, we already discussed some of the difficulties which prevent us from analyzing the dynamics of the algorithm by tracking a scalar quantity at each vertex, as would be the case for the analogous but much simpler heat-kernel dynamics. A key ingredient in our solution to the problem is to introduce a two-dimensional representation of the graph function α, together with an associated potential function that will enable our analysis of the raising phase. One coordinate (the vertical coordinate) in this representation, called the "height" of a vertex, is based on the unique raising vector r^* whose existence is guaranteed by Lemma 4.4 of the previous section; the second (horizontal) coordinate is the "level" of the vertex, as alluded to in the Introduction. This representation, which is specified in detail in Section 5.1, will be instrumental in obtaining upper and lower bounds relating the (global) potential function to the local imbalances (which govern the effect of a single raising operation); these bounds are derived in Sections 5.2 and 5.3, respectively. Finally, in Section 5.4 we conclude the proof of Theorem 1.3.

5.1. A Two-Dimensional Representation

Throughout this and the next two subsections, we focus exclusively on the raising phase. As we observed previously, exactly the same rate of convergence will apply to the lowering phase.

Recall from Lemma 4.4 that, for any initial graph function α, the limiting raising vector r^*, which records the asymptotic total amount of raising performed at each vertex by any infinite fair sequence of raising operations, is uniquely defined and specifies a unique raising balanced graph function α^R. We may therefore use r^* to define a *height function*

$$y_v := -r_v^*,$$

i.e., $-y_v$ is the amount of raising that remains to be performed at vertex v in order to reach the raising-balanced graph function α^R. (Note that, throughout, all heights are nonpositive and converge monotonically to zero. This convention turns out to be convenient in our analysis.) The heights are related to α and α^R by

$$\alpha_{uv}^R - \alpha_{uv} = y_u - y_v, \tag{4}$$

which is just a restatement of Equation (1).

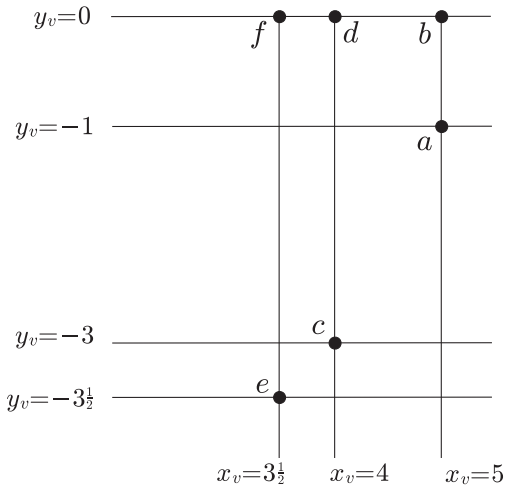

Fig. 3. Two-dimensional representation of the graph function α shown in Figure 2(i).

In addition to the heights y_v, we will need to introduce a second coordinate x_v defined by

$$x_v := \max_{u:u\to v} \alpha^R_{uv}. \tag{5}$$

We refer to x_v as the *level* of vertex v. Note that (intuitively at least), lower level vertices can only reliably be close to raising-balanced after higher level ones are close, so the levels capture one of the distinguishing features of the problem. Note also that, unlike its height, the level of a vertex does not change over time. The two-dimensional representation (x_v, y_v) of vertices v will be key to the remainder of our argument; in particular, the role of the level coordinate x_v will become apparent in Section 5.3.

Figure 3 shows the two-dimensional representation of the graph function in Figure 2(i). The horizontal coordinates can be read off from the maximum incoming edges at each vertex in Figure 2(ii), and the vertical coordinates from the square brackets in the same figure.

We will measure progress of the raising phase by means of the global potential function $\Psi = -\sum y_v$, which is the total amount of raising that remains to be done to reach α^R. In order to relate Ψ to the local imbalances ρ^R_v, it will be convenient to introduce a closely related global quantity h, which is just the difference between the maximum and minimum heights. That is, setting $y_{\min} = \min_u y_u$ and $y_{\max} = \max_u y_u$, we define

$$h := y_{\max} - y_{\min}.$$

As we shall see shortly (Corollary 5.4), it is always the case that $y_{\max} = 0$, so in fact $h = -y_{\min}$. Note that Ψ and h differ by at most a factor of n, namely:

$$\Psi \geq h \geq \frac{1}{n}\Psi. \tag{6}$$

The key ingredients in our analysis are the following two bounds relating h to the local imbalances ρ^R_v. Recall from Section 2 the notation $\rho^R = \max_v \rho^R_v$.

Proposition 5.1. *For any graph function α, we have $\rho^R/2 \leq h$.*

Proposition 5.2. *For any graph function α, we have $h \leq (n-1)\sum_v \rho^R_v$.*

Proposition 5.1 confirms that, to ensure ε-raising-balance, it suffices to achieve $\Psi \leq \varepsilon/2$. The much more subtle Proposition 5.2 implies that we make steady progress reducing Ψ. This implication depends on the update sequence and will be given in detail in Section 5.4, but a brief outline is as follows. First, when the vertex v at which the raising operation is performed is chosen u.a.r., Ψ decreases at each step by an expected $(1 - \Omega(\frac{1}{n^3}))$ factor; to see this, note that the expected decrease in Ψ in one raising operation is $\frac{1}{2n}\sum_v \rho_v^R$, and apply Proposition 5.2 and Equation (6). Second, when v is chosen according to the cyclic order, the monotonicity from Proposition 4.1 ensures that, on average, progress is at least as fast as in the u.a.r. case.

In the next two subsections, we prove Propositions 5.1 and 5.2 respectively.

5.2. Proof of Proposition 5.1

The proposition is an almost immediate consequence of the following straightforward lemma relating heights to local imbalances. Since we are dealing exclusively with the raising phase, in this and the next two subsections we will write ρ_v, ρ in place of ρ_v^R, ρ^R to simplify notation.

LEMMA 5.3. *Let α be a graph function and v a vertex with $\rho_v > 0$. Then there exist edges (u, v) and (v, w) in G_α such that*

$$y_v + \frac{\rho_v}{2} \leq \frac{y_u + y_w}{2}. \tag{7}$$

PROOF. Let u, w be vertices such that $\alpha_{uv} = \alpha_v^{\text{in}}$ and $\alpha_{vw} = \alpha_v^{\text{out}}$ (i.e., (u, v) and (v, w) are incoming and outgoing edges of maximum weight at v). Note that $\rho_v = \alpha_{vw} - \alpha_{uv}$.

To prove Equation (7), we let u' be a vertex such that the edge (u', v) achieves the maximum in Equation (5), i.e., $\alpha_{u'v}^R = x_v$, and write

$$\begin{aligned} y_v + \frac{\rho_v}{2} &= y_{u'} + \alpha_{u'v} - \alpha_{u'v}^R + \frac{\rho_v}{2} \\ &= y_{u'} + \alpha_{u'v} - \alpha_{u'v}^R + \frac{\alpha_{vw} - \alpha_{uv}}{2} \\ &\leq y_{u'} - \alpha_{u'v}^R + \frac{\alpha_{vw} + \alpha_{u'v}}{2}, \end{aligned} \tag{8}$$

where in the first line we used Equation (4) and in the last line the fact that $\alpha_{u'v} \leq \alpha_{uv}$. Now two further applications of Equation (4) give

$$\begin{aligned} \alpha_{vw} &= y_w - y_v + \alpha_{vw}^R; \\ \alpha_{u'v} &= y_v - y_{u'} + \alpha_{u'v}^R. \end{aligned}$$

Plugging these into Equation (8) yields

$$y_v + \frac{\rho_v}{2} \leq y_{u'} - \alpha_{u'v}^R + \frac{y_w - y_{u'} + \alpha_{vw}^R + \alpha_{u'v}^R}{2} = \frac{y_{u'} + y_w}{2} + \frac{\alpha_{vw}^R - \alpha_{u'v}^R}{2} \leq \frac{y_{u'} + y_w}{2},$$

where the last inequality follows from the facts that α^R is raising-balanced and $\alpha_{u'v}^R$ is maximal among incoming edges at v. This completes the proof. □

Before proving Proposition 5.1, we pause to observe a consequence of Lemma 5.3:

COROLLARY 5.4. $y_{\max} = 0$.

PROOF. A raising operation applied at some vertex v increases y_v to $y_v + \frac{\rho_v}{2}$. By Lemma 5.3 this is $\leq \frac{y_u + y_w}{2} \leq y_{\max}$. Therefore, $y_{\max}$ can never increase during the raising

phase. Choose a vertex u so that $y_u = y_{\max}$ initially. Then u can never rise, and so $r^*_u = 0$. Thus, $y_{\max} = y_u = -r^*_u = 0$. □

We conclude this subsection by proving Proposition 5.1.

Proof of Proposition 5.1. Let v be any vertex with raising imbalance $\rho_v > 0$. By Lemma 5.3 we have

$$\frac{\rho_v}{2} \leq \frac{y_u + y_w}{2} - y_v \leq y_{\max} - y_{\min} = h.$$

Together with the fact that, by definition, $h \geq 0$, this completes the proof. □

5.3. Proof of Proposition 5.2

Now we turn to the reverse inequality, Proposition 5.2, which is considerably more delicate. It makes crucial use of the two-dimensional representation introduced earlier, as well as the concept of "momentum," whose role will emerge shortly. The *momentum* of a vertex v is defined by

$$m_v := \alpha^{\text{in}}_v = \max_{u:u\to v} \alpha_{uv}. \tag{9}$$

Note that in the limit of the raising-balanced graph function α^R, the momentum of a vertex v is equal to its level x_v.

One key property of momentum is that it can be upper bounded at a vertex in terms of the sum of the momentum and the local imbalance at the predecessors of the vertex in the graph. (The form of this bound is complicated by a dependence on the levels of the vertices; see part (i) of Lemma 5.5.) The second key property of momentum is that, along any edge, height can increase only at a rate bounded by momentum and local imbalance. (Again, there is an additional term in this bound; see part (ii) of Lemma 5.5.) Putting these two properties together will allow us to show that, if the height difference between some two vertices in the graph is substantial, then the aggregate local imbalance $\sum_v \rho_v$ must be large, which is exactly the content of Proposition 5.2.

Lemma 5.5. *The momentum satisfies the following inequalities:*

(*i*) *For any vertex* v, $m_v \leq \max\{x_v, \max_{u:u\to v, y_u < y_v}\{m_u + \rho_u\}\}$.
(*ii*) *For any edge* (u, v) *in* G_α, $y_v \leq y_u + m_u + \rho_u - \alpha^R_{uv}$.

Proof. Both parts rely on the following elementary observation. For any edge (u, v), by the definition of ρ_u we have

$$\alpha_{uv} \leq \max_{u':u'\to u} \alpha_{u'u} + \rho_u = m_u + \rho_u, \tag{10}$$

where we have used the definition of momentum (9).

For part (i), we thus have

$$m_v = \max_{u:u\to v} \alpha_{uv} \leq \max_{u:u\to v}\{m_u + \rho_u\}.$$

Now note that if $y_u \geq y_v$ we have, recalling Equations (4) and (5),

$$\alpha_{uv} = y_v - y_u + \alpha^R_{uv} \leq x_v. \tag{11}$$

This justifies the restriction on the right-hand side of (i) to maximization over u such that $y_u < y_v$. This completes the proof of part (i).

For part (ii), apply Equations (4) and (10) to deduce that

$$y_v = y_u + \alpha_{uv} - \alpha^R_{uv} \leq y_u + m_u + \rho_u - \alpha^R_{uv},$$

as required. □

Remark. We pause to provide motivation for our choice of the term "momentum." Consider the special case in which all vertices have the same level, i.e., $x_v = \max_{u:u\to v} \alpha^R_{uv} = c$ for all v. By rescaling, we may take $c = 0$. Considering $y_v - y_u$ as the height gain along edge (u, v), part (ii) of Lemma 5.5 says that the height gain is bounded by the momentum m_u (plus ρ_u, which generally is a lower order term). So m_u can naturally be viewed as the momentum of a particle as it rises along edges. Pursuing this analogy further, part (i) of Lemma 5.5 indicates that ρ_v plays roughly the role of acceleration.

To complete the proof of Proposition 5.2, we need to translate Lemma 5.5, which gives local bounds on the increase of momentum and height, into global bounds in G_α, as specified in Lemma 5.6 below. Note that part (i) of Lemma 5.5 immediately implies that m_v is bounded above by the sum of local imbalances at vertices below v, except for a penalty that depends on the levels x_u of those vertices. The key to the translation is then to apply part (ii) of Lemma 5.5 to a carefully chosen sequence of edges which span the heights in the graph, and for each of which the term $-\alpha^R_{uv}$ in the lemma can be used to cancel the penalty; these edges will be identified in Lemma 5.7 below.

For a vertex v, let S_v denote the set of vertices whose height is strictly smaller than v, i.e., $S_v = \{u : y_u < y_v\}$. Also, let $\rho(S_v) = \sum_{u\in S_v} \rho_u$.

We are now ready to state our global bounds.

LEMMA 5.6. *For any vertex v, the following two bounds hold:*

(i) $m_v \le \rho(S_v) + \max_{u:y_u\le y_v} x_u$;
(ii) $y_v - y_{\min} \le |S_v| \cdot \rho(S_v)$.

PROOF. Both parts (i) and (ii) are proved by induction on vertices v in order of height. We begin with part (i), which will be used in the proof of part (ii).

Part (i). We use induction on the height y_v of v. If $y_v = y_{\min}$, the statement is true since $m_v \le x_v$ (this follows from part (i) of Lemma 5.5). Assuming $y_v > y_{\min}$, and beginning with part (i) of Lemma 5.5, we may write

$$m_v \le \max\Big\{x_v, \max_{u:u\to v, y_u<y_v} \{m_u + \rho_u\}\Big\}.$$

We can use induction to bound each term in the inner maximization (noting that there $y_u < y_v$) as follows:

$$m_u + \rho_u \;\le\; \rho_u + \rho(S_u) + \max_{u':y_{u'}\le y_u} x_{u'} \;\le\; \rho(S_v) + \max_{u':y_{u'}\le y_u} x_{u'} \;\le\; \rho(S_v) + \max_{u':y_{u'}\le y_v} x_{u'},$$

where in the second inequality we used the fact that $u \in S_v \setminus S_u$ to deduce that $\rho(S_u) + \rho_u \le \rho(S_v)$. Noting that $\rho(S_v) \ge 0$, this completes the inductive proof of part (i).

Part (ii). Again we use induction on the height of v. The base case is $y_v = y_{\min}$, in which case the statement holds trivially.

Now let v be any vertex with $y_{\min} < y_v$, and suppose part (ii) is established for all u with $y_u < y_v$. Let U be the subset of vertices $u \in S_v$ for which $x_u = x_U$, where $x_U = \max_{u\in S_v} x_u$. Of course, $U \ne \emptyset$. We require the following additional fact:

LEMMA 5.7. *There is a vertex $u' \in U$ having an edge (u', v') where $\alpha^R_{u'v'} = x_U$ and $y_{v'} \ge y_v$.*

We defer the proof of Lemma 5.7 to the end of the subsection and continue with the proof of Lemma 5.6(ii). Applying part (ii) of Lemma 5.5 to the edge (u', v') provided by

Lemma 5.7, we have

$$\begin{aligned} y_v \;\le\; y_{v'} \;&\le\; y_{u'} + m_{u'} + \rho_{u'} - \alpha^R_{u'v'} \\ &\le\; y_{u'} + \rho(S_{u'}) + \left(\max_{u:y_u \le y_{u'}} x_u\right) + \rho_{u'} - x_U \\ &\le\; y_{u'} + \rho(S_{u'}) + \rho_{u'} \\ &\le\; y_{\min} + |S_{u'}|\rho(S_{u'}) + \rho(S_{u'}) + \rho_{u'} \\ &\le\; y_{\min} + |S_v|\rho(S_v). \end{aligned}$$

In the second line we have used part (i) of the current lemma as well as the guarantee of Lemma 5.7 that $\alpha^R_{u'v'} = x_U$; in the third line we have used the definition of x_U; in the fourth line we have used the inductive hypothesis applied to $y_{u'}$ (which is strictly less than y_v); and in the last line we have used the fact that $u' \in S_v \setminus S_{u'}$. This completes the inductive proof of part (ii) of the lemma. □

Our main goal in this subsection, Proposition 5.2, now follows trivially from Lemma 5.6.

Proof of Proposition 5.2. By part (ii) of Lemma 5.6 we have, for any vertex v, $y_v - y_{\min} \le (n-1)\sum_u \rho_u$. □

It remains only for us to supply the missing proof of Lemma 5.7.

Proof of Lemma 5.7. The proof rests upon defining the graph G' formed by the set of edges $\{(u,w) : u \in U, \alpha^R_{uw} \ge x_U\}$, and showing:

Claim: There is an edge $(u,w) \in G'$ such that $w \notin U$.

The first thing to note is that for all edges $(u,w) \in G'$, in fact $\alpha^R_{uw} = x_U$; this is because α^R is raising-balanced.

To see that the Lemma follows from the Claim, with (u',v') instantiated by (u,w): first, since $(u,w) \in G'$, $\alpha^R_{uw} = x_U$. Then $x_w \ge x_U$, which taken together with $w \notin U$ implies that $y_w \ge y_v$.

To show the Claim we start by arguing that every vertex $u \in U$ has an outgoing edge $(u,w) \in G'$, and therefore such that $\alpha^R_{uw} = x_U$. Since $u \in U$, we know that $y_u < 0$ and hence some raising must eventually be performed at u, i.e., eventually $\rho^R_u > 0$ and then $\rho^L_u = 0$. But once this happens, Proposition 4.1 shows that $\rho^L_u = 0$ at all future times, and hence in α^R we must have $\rho^R_u = \rho^L_u = 0$. Thus, if (u,w) is a maximum outgoing edge at u in α^R, we have $\alpha^R_{uw} = \max_v \alpha^R_{vu} = x_U$.

Now suppose the Claim to be false, so $w \in U$ for all edges of G'. Then since every $u \in U$ has an outgoing edge in G', there is a sink scc on vertices $U' \subseteq U$. (U' may be as small as a single vertex with a self-loop.)

Let $\delta_0 = -\max_{u' \in U'} y_{u'}$. Note that $\delta_0 > 0$. Also, let $\delta_1 = \min\{\alpha^R_{rr'} - \alpha^R_{ss'} : \alpha^R_{rr'} - \alpha^R_{ss'} > 0\}$; if this set is empty let $\delta_1 = \infty$. Finally, select $0 < \delta < \min\{\delta_0, \delta_1\}$.

Now fix any infinite fair sequence of raising operations, and consider the first operation in the sequence that increases any y_u, $u \in U'$, to a value larger than $-\delta$. (This occasion is well defined as all $y_u, u \in U'$, increase monotonically to zero.) Let u' be the vertex raised by this operation, and let $\bar{\alpha}$ be the graph function immediately after the operation and $\bar{y}$ the height function corresponding to $\bar{\alpha}$. By Theorem 1.2, $\bar{\alpha}^R = \alpha^R$.

By construction there is an edge (t,u') in G', with $t \in U'$. Applying Equation (4) to the edge (t,u'), we see that

$$\bar{\alpha}_{tu'} = \alpha^R_{tu'} + \bar{y}_{u'} - \bar{y}_t \ge x_U + \bar{y}_{u'} - \bar{y}_t \ge x_U, \tag{12}$$

where the first inequality is because the edge is in G', and the second inequality is by choice of u'. (Note that equality can hold here only if $t = u'$.)

Now let (u', w) be a maximum (in $\bar{\alpha}$) outgoing edge at u'. Since a nontrivial raising operation has just been performed at u', it must be the case that $w \neq u'$. (To see this, note that raising preserves the order of weights among outgoing edges. So, if a self-loop is a maximal outgoing edge from u' in $\bar{\alpha}$, then it must also have been maximal before raising. But in that case there could have been no nontrivial raising at u'.)

Since we have just raised at u', $\rho^L_{u'} = 0$, so $\bar{\alpha}_{u'w} \geq \bar{\alpha}_{tu'}$. Combining with Equation (12) we have

$$\bar{\alpha}_{u'w} \geq x_U. \tag{13}$$

Applying Equation (4) to the edge (u', w) we get

$$\alpha^R_{u'w} = \bar{\alpha}_{u'w} + \bar{y}_{u'} - \bar{y}_w \geq \bar{\alpha}_{u'w} + \bar{y}_{u'} \geq \bar{\alpha}_{u'w} - \delta \geq x_U - \delta,$$

where the first inequality is because heights are nonpositive, the second is by the choice of u', and the third is by Equation (13). But since $\delta < \delta_1$, this in fact implies that $\alpha^R_{u'w} \geq x_U$ and therefore that (u', w) belongs to G' and $\alpha^R_{u'w} = x_U$. Since U' is a sink scc it must be the case that $w \in U'$. By the choice of u', then, and the fact that $w \neq u'$, $\bar{y}_w < \bar{y}_{u'}$.

Hence,

$$\bar{\alpha}_{u'w} = \alpha^R_{u'w} - \bar{y}_{u'} + \bar{y}_w = x_U - \bar{y}_{u'} + \bar{y}_w < x_U,$$

which in light of Equation (13) gives us the desired contradiction. □

5.4. Proof of Theorem 1.3

We are now finally in a position to prove our main result, Theorem 1.3.

Proof of Theorem 1.3. We begin with the variant of Algorithm 1 in which vertices are chosen u.a.r.; the analysis of the cyclic variant is similar and follows. It suffices to show that, with high probability, $O(n^3 \log(\rho n/\varepsilon))$ raising operations suffice to achieve an ε-raising-balanced graph function. As we observed at the beginning of the section, by symmetry the same holds for the lowering phase, and hence the output of the algorithm will be ε-balanced w.h.p..

To analyze the raising phase, we index all quantities by t, the number of raising operations that have so far been performed. Thus, in particular, $y_v(t)$ is the height of vertex v after t raising operations. We also introduce the potential function $\Psi(t) = -\sum_v y_v(t)$. Note that $\Psi(t) \geq 0$ and that $\Psi(t)$ decreases monotonically to 0 as $t \to \infty$. Moreover, by Proposition 5.1 we have

$$\Psi(t) = -\sum_v y_v(t) \geq h(t) \geq \rho^R(t)/2.$$

Hence in order to achieve raising imbalance at most ε after t raising steps it suffices to ensure that $\Psi(t) \leq \varepsilon/2$.

Since a raising operation at vertex v increases y_v by $\rho^R_v/2$, the expected decrease in $\Psi(t)$ in one raising operation is

$$\mathrm{E}[\Psi(t) - \Psi(t+1)] = \frac{1}{2n}\sum_v \rho^R_v(t) \geq \frac{h(t)}{2n(n-1)} = \frac{-y_{\min}(t)}{2n(n-1)} \geq \frac{1}{2n^3}\Psi(t), \tag{14}$$

where in the first inequality we have used Proposition 5.2, in the next step Corollary 5.4, and in the last inequality the fact that $y_{\min}$ is less than the average of the y_v. Iterating for t steps yields

$$\mathrm{E}[\Psi(t)] \leq \left(1 - \frac{1}{2n^3}\right)^t \Psi(0).$$

To get an upper bound on $\Psi(0)$, we observe that

$$\Psi(0) = -\sum_v y_v(0) \le -ny_{\min}(0) = nh(0) \le n(n-1)\sum_v \rho_v^R(0) \le n^3\rho^R(0) \le n^3\rho,$$

where we have again used Corollary 5.4 and Proposition 5.2. (Recall that $\rho = \max\{\rho^R(0), \rho^L(0)\}$ is the imbalance of the original matrix.) Hence, for any $\delta \in (0,1)$, after $t = 6n^3 \ln(2\rho n/\varepsilon\delta)$ raising operations we have

$$\mathrm{E}[\Psi(t)] \le \varepsilon\delta/2.$$

By Markov's inequality this implies that $\Psi(t) \le \varepsilon/2$ with probability at least $1-\delta$. To obtain the result w.h.p., it suffices to take $\delta = o(1)$.

We turn now to the variant in which vertices are chosen in deterministic cyclic order. In this case, we consider each cycle of n raising operations, numbered $tn+1, \ldots, (t+1)n$, one at each vertex; let the vertices be indexed $v = 1, \ldots, n$ according to the order they are visited in this cycle. We show that $O(n^2 \ln(\rho n/\varepsilon))$ such cycles suffice to reduce Ψ to $\varepsilon/2$, for the same potential function Ψ as previously. Let $\rho_v^R(s)$ be the raising imbalance at v at time s, so in particular, $\rho_v^R(tn)$ is the raising imbalance at v at the beginning of this cycle, and $\rho_v^R(tn+v-1)$ is the raising imbalance at v just before the raising step at v. By Proposition 4.1, $\rho_v^R(tn) \le \rho_v^R(tn+v-1)$. The raising operation at v decreases Ψ by $\rho_v^R(tn+v-1)/2$. Hence, the aggregated effect of the sequence of n raising operations decreases Ψ by at least $\frac{1}{2}\sum_v \rho_v^R(tn)$. As in Equation (14), this yields

$$\Psi(tn) - \Psi((t+1)n) \ge \frac{1}{2}\sum_v \rho_v^R(tn) \ge \frac{1}{2n^2}\Psi(tn)$$

for any integer $t \ge 0$, and essentially the same calculations as previously imply that $\Psi(tn) \le \varepsilon/2$ for $t = 6n^2 \ln(2\rho n/\varepsilon)$, as desired. □

6. OPEN PROBLEMS

We close the article with some remarks and open problems.

(1) The introduction of raising and lowering phases into the Osborne-Parlett-Reinsch algorithm is trivial from an implementation perspective (the effect being simply to censor some of the balancing operations), but seemingly crucial to our analysis. If balancing operations of both types are interleaved, we lose our tight control on the rate of progress and are able to prove only the much weaker exponential bound of Theorem 4.5 on the convergence time. It would be interesting to modify the tight analysis so as to handle an arbitrary sequence of balancing operations, as well as to investigate whether the introduction of phases actually improves the rate of convergence (both in theory and in practice).

(2) Recall that, without raising and lowering phases, the balanced matrix to which the algorithm converges is not unique (except, by definition, in the UB case). It would be interesting to investigate the geometric structure of the set of fixed points, which apparently can be rather complicated.

(3) As we have seen, our $\widetilde{O}(n^3)$ bound on the worst-case convergence time is essentially optimal. While this goes some way toward explaining the excellent performance of the algorithm in practice, more work needs to be done to explain its empirical dominance over algorithms with faster worst-case asymptotic running times. In particular, can one identify features of "typical" matrices that lead to much faster convergence?

(4) We have analyzed only the L_∞ variant of the Osborne-Parlett-Reinsch algorithm. We note that there is active discussion in the numerical analysis literature about which norm to use for balancing; for example, Watkins [2006] and James et al. [2014] show

that in some cases L_1 balancing leads to undesirable results. Regarding the algorithmic aspect, recent work by Ostrovsky et al. [2017], following the appearance of earlier versions of this article [Schulman and Sinclair 2015a, 2015b], bounds the convergence time of the L_1 variant (albeit with respect to a weaker definition of ε-balanced). All L_p formulations of the problem for finite p (but not for $p = \infty$) are interreducible, simply by raising all entries of the matrix to an appropriate power, so Ostrovsky et al. [2017] provides bounds on the convergence time of iterative algorithms for matrix balancing in L_p for all finite p.

APPENDIX

A. PROOF OF THEOREM 4.5

PROOF. The proof is similar to that for raising operations in Lemma 4.2. We again use the potential function $\Phi(t)$ defined in Equation (2). Recall from the proof of Lemma 4.2 that a raising operation at v decreases $\Phi(t)$ by at least $\rho_v^R/2$ (see Equation (3)). A symmetrical argument shows that a lowering operation at v also decreases $\Phi(t)$, by at least $\rho_v^L/2$. Hence, we may bound the decrease in Φ over t steps in terms of the cumulative raising and lowering vectors $r(t)$, $\ell(t)$ as follows:

$$\Phi(0) - \Phi(t) \geq \sum_v r_v(t) + \sum_v \ell_v(t).$$

Since $\Phi(t)$ is bounded (as in the proof of Lemma 4.2), we see that the vectors $r(t)$, $\ell(t)$ are monotonically nondecreasing and bounded, and hence converge to limits r^*, ℓ^* respectively.[7] This in turn implies that $\alpha(t)$ converges to a balanced graph function α^* given by

$$\alpha^*_{uv} = \alpha_{uv} + r^*_v - r^*_u - \ell^*_v + \ell^*_u.$$

To bound the rate of convergence, we obtain explicit upper and lower bounds on Φ. Recall from the proof of Proposition 2.1 that all $\alpha_{uv}(t)$ satisfy $b_{\min} \leq \alpha_{uv}(t) \leq b_{\max}$, where $b_{\max} = \alpha_{\max}$ and $b_{\min} = \alpha_{\min} - (n-1)(\alpha_{\max} - \alpha_{\min})$; therefore, $|\alpha_{uv}(0) - \alpha_{uv}(t)| \leq n \cdot (\alpha_{\max} - \alpha_{\min})$. From the definition (2) of Φ, we may therefore conclude that

$$\Phi(0) - \Phi(t) \leq nm \cdot m! \cdot (\alpha_{\max} - \alpha_{\min}). \tag{15}$$

Now suppose that, at all times within the first T epochs, the imbalance of $A(t)$ is at least ε. We claim that, in each such epoch, Φ must decrease by at least $\varepsilon/4$. Combining this claim with Equation (15) will complete the proof as it implies that $T \leq 4nm \cdot m! \cdot (\alpha_{\max} - \alpha_{\min})/\varepsilon$.

To see the preceding claim, consider one such epoch that starts at time t; let v be a vertex with imbalance $\rho_v(t) \geq \varepsilon$, and let t' be the first time in the epoch at which balancing is performed at v. If the cumulative raising and lowering during the epoch prior to time t' is at least $\varepsilon/4$, then Φ has decreased by at least this amount and we are done. Otherwise, the weight of any edge incident on v has changed by no more than $\varepsilon/4$, so the imbalance of v at time t' is still at least $\varepsilon/2$, leading to a decrease in Φ of at least $\varepsilon/4$.

ACKNOWLEDGMENTS

We thank Tzu-Yi Chen and Jim Demmel for introducing us to this problem and for several helpful pointers, Martin Dyer for useful discussions, and Yuval Rabani for help in the early stages of this work which led to

[7]Note that when raising and lowering operations are interspersed, as considered in this theorem, r^* and ℓ^* may depend on the sequence of balancing operations.

the proof of Theorem 1.1. We also thank anonymous referees for suggestions that improved the presentation of the article.

REFERENCES

S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. 1994. *Linear Matrix Inequalities in System and Control Theory*, SIAM Studies in Applied Mathematics, Vol. 15.

T.-Y. Chen. 1998. *Balancing Sparse Matrices for Computing Eigenvalues*. Master's thesis. UC Berkeley.

T.-Y. Chen and J. Demmel. 2000. Balancing sparse matrices for computing eigenvalues. *Linear Algebra Appl.* 309 (2000), 261–287.

B. C. Eaves, A. J. Hoffman, U. G. Rothblum, and H. Schneider. 1985. Line-sum-symmetric scalings of square non-negative matrices. *Math. Programm. Study* 25 (1985), 124–141.

J. Franklin and J. Lorenz. 1989. On the scaling of multidimensional matrices. *Linear Multilinear Algebra* 23 (1989), 717–735.

J. Grad. 1971. Matrix balancing. *Comput. J.* 14 (1971), 280–284.

D. J. Hartfiel. 1971. Concerning diagonal similarity of irreducible matrices. *Proc. Amer. Math. Soc.* 30 (1971), 419–425.

M. Idel. 2016. A review of matrix scaling and Sinkhorn's normal form for matrices and positive maps. (2016). http://arxiv.org/abs/1609.06349 ArXiv: 1609.06349.

R. James, J. Langou, and B. R. Lowery. 2014. On matrix balancing and eigenvector computation. (2014). ArXiv: 1401.5766v1.

B. Kalantari and L. Khachiyan. 1993. On the rate of convergence of deterministic and randomized RAS matrix scaling algorithms. *Op. Res. Lett.* 14 (1993), 237–244.

B. Kalantari, L. Khachiyan, and A. Shokoufandeh. 1997. On the complexity of matrix balancing. *SIAM J. Matrix Anal. Appl.* 18 (1997), 450–463.

B. Kalantari, I. Lari, F. Ricca, and B. Simeone. 2008. On the complexity of general matrix scaling and entropy minimization via the RAS algorithm. *Math. Program.* 112, 2 (2008), 371–401.

D. Kressner. 2005. *Numerical Methods for General and Structured Eigenvalue Problems*. Springer.

N. Linial, A. Samorodnitsky, and A. Wigderson. 2000. A deterministic strongly polynomial algorithm for matrix scaling and approximate permanents. *Combinatorica* 20 (2000), 531–544.

E. E. Osborne. 1960. On pre-conditioning of matrices. *J. ACM* 7, 4 (1960), 338–345. DOI:http://dx.doi.org/10.1145/321043.321048

R. Ostrovsky, Y. Rabani, and A. Yousefi. 2017. Matrix balancing in L_p norms: Bounding the convergence rate of Osborne's iteration. In *Proceedings of the 28th ACM-SIAM Symposium on Discrete Algorithms (SODA)*.

B. N. Parlett and C. Reinsch. 1969. Balancing a matrix for calculation of eigenvalues and eigenvectors. *Numer. Math.* 13 (1969), 293–304.

H. Schneider and M. H. Schneider. 1991. Max-balancing weighted directed graphs and matrix scaling. *Math. Oper. Res.* 16, 1 (1991), 208–222.

L. J. Schulman and A. Sinclair. 2015a. Analysis of a classical matrix preconditioning algorithm. In *Proceedings of the 47th Annual ACM Symposium on Theory of Computing (STOC'15)*.

L. J. Schulman and A. Sinclair. 2015b. Analysis of a classical matrix preconditioning algorithm. (2015). http://arxiv.org/abs/1504.03026v2, ArXiv: 1504.03026.

R. Sinkhorn. 1964. A relationship between arbitrary positive matrices and doubly stochastic matrices. *Ann. Math. Stat.* 35 (1964), 876–879.

R. Sinkhorn and P. Knopp. 1967. Concerning nonnegative matrices and doubly stochastic matrices. *Pacific J. Math.* 21 (1967), 343–348.

T. Ström. 1972. Minimization of norms and logarithmic norms by diagonal similarities. *Computing* 10 (1972), 1–7.

L. N. Trefethen and M. Embree. 2005. *Spectra and Pseudospectra: The Behavior of Nonnormal Matrices and Operators*. Princeton University Press.

D. S. Watkins. 2006. A case where balancing is harmful. *Electron. Trans. Num. Anal.* 23 (2006), 1–4.

N. Young, R. E. Tarjan, and J. B. Orlin. 1991. Faster parametric shortest path and minimum balance algorithms. *Networks* 21 (1991), 205–221.

Received December 2015; revised November 2016; accepted January 2017