Derandomized Squaring of Graphs*

Eyal Rozenman
Department of Computer Science & Applied Mathematics
Weizmann Institute, Rehovot, Israel

Salil Vadhan†
Division of Engineering & Applied Sciences
Harvard University, Cambridge, Massachusetts

Abstract

We introduce a "derandomized" analogue of graph squaring. This operation increases the connectivity of the graph (as measured by the second eigenvalue) almost as well as squaring the graph does, yet only increases the degree of the graph by a constant factor, instead of squaring the degree.

One application of this product is an alternative proof of Reingold's recent breakthrough result that S-T Connectivity in Undirected Graphs can be solved in deterministic logspace.

1 Introduction

"Pseudorandom" variants of graph operations have proved to be useful in a variety of settings. Alon, Feige, Wigderson, and Zuckerman [AFWZ] introduced "derandomized graph products" to give a more illuminating deterministic reduction from approximating clique to within relatively small (eg constant) factors to approximating clique to within relatively large (eg n^ϵ) factors. Reingold, Vadhan, and Wigderson [RVW] introduced the "zig-zag graph product" to give a new construction of constant-degree expander graphs. The zig-zag product and its relatives found a number of applications, the most recent and most dramatic of which is Reingold's deterministic logspace algorithm [Rei2] for connectivity in undirected graphs.

*An extended abstract of this paper has appeared in *RANDOM '05* [RV].

†Supported by NSF grant CCF-0133096, ONR grant N00014-04-1-0478, and US-Israel BSF grant 2002246.

In this paper, we present a pseudorandom analogue of graph squaring. The *square* X^2 of a graph X is the graph on the same vertex set whose edges are paths of length 2 in the original graph. This operation improves many connectivity properties of the graph, such as the diameter and mixing time of random walks of the graph (both of which roughly halve). However, the degree of the graph squares. In terms of random walks on the graph, this means that although half as many steps are needed to mix, each step costs twice as many random bits. Thus, there is no savings in the amount of randomness needed for mixing.

Our derandomized graph squaring only increases the degree by a constant factor rather than squaring it. Nevertheless, it improves the connectivity almost as much as the standard squaring operation. The measure of connectivity for which we prove this is the second eigenvalue of the graph, which is well-known to be a good measure of the mixing time of random walks, as well as of graph expansion. The standard squaring operation squares the second eigenvalue; we prove that the derandomized squaring does nearly as well.

The main application of derandomized squaring we give here is a new logspace algorithm for connectivity in undirected graphs, thereby giving a new proof of Reingold's theorem [Rei2]. Our algorithm, while closely related to Reingold's algorithm, is arguably more natural. Reingold's algorithm is based on the *zig-zag product*, and constructs a sequence of graphs with an increasing number of vertices. Our analysis, based on derandomized squaring, only works on the vertex set of the original input graph, and has a simpler analysis of the space requirements. More significantly, it can be viewed as applying a natural pseudorandom generator, namely that of Impagliazzo, Nisan, and Wigderson [INW], to random walks on the input graph. Reingold himself [Rei1] conjectured that it should be possible to use INW generator to solve undirected connectivity in logspace; we confirm his conjecture by the relating the INW generator to derandomized squaring.

Below we describe the derandomized squaring and its application to undirected s-t connectivity in more detail.

1.1 Derandomized Graph Squaring

Let X be an undirected regular graph of degree K.[1] The square X^2 of X has an edge for every path of length 2 in X. One way to visualize it is that for every vertex v in X, we place a clique on its K neighbours (this connects every pair of vertices that has a length 2 path through v). The degree thus becomes K^2. (Throughout the paper, we allow multiple edges and self-loops.)

[1] Actually, following [RTV], we actually work with regular *directed* graphs in the technical sections of the paper, but thinking of undirected graphs suffices for the informal discussion here.

In derandomized squaring, we use an auxiliary graph G on K vertices and place it instead of a clique on the K neighbours of every vertex v (thus connecting only *some* of the pairs of vertices which have a length 2 path through v). We denote the resulting graph by $X ⓢ G$.

If the degree of G is D, the derandomized square will have degree KD, which will be smaller than K^2. We will see, however, that if G is an expander, then even if D is much smaller than K, the derandomized square of X with respect to G improves connectivity similarly to standard squaring.

Our measure of connectivity is the second eigenvalue $\lambda \in [0, 1]$ of (the random walk on) the graph; small λ indicates that the random walk mixes rapidly and that the graph has good expansion (i.e. is highly connected). If the second eigenvalue of X is λ then the second eigenvalue of X^2 is λ^2. The second eigenvalue of the derandomized square is not very far. For example, we prove that it is at most $\lambda^2 + \mu$ where μ is the second eigenvalue of G. In fact, we give a tight analysis of the second eigenvalue of the derandomized square as a function of λ and μ.

1.2 A New Logspace Algorithm for Undirected Connectivity

Recall that the problem of undirected st-connectivity is: given an undirected graph G and two vertices s, t, decide whether there is a path from s to t in G. The time complexity of this problem is well understood — search algorithms like breadth-first search (BFS) and depth-first search (DFS) solve it in linear time. The space complexity is harder to tackle. A line of research starting in the $\log^2(N)$-space deterministic algorithm of Savitch [Sav] and the randomized $\log(N)$-space algorithm of Aleliunas et. al. [AKL$^+$] culminated in Reingold's optimal deterministic $\log(N)$-space algorithm [Rei2] (See Reingold's paper and the references therein for more on the history and applications of this problem). We now shortly describe Reingold's algorithm, then present our algorithm and compare the two.

Reingold's Algorithm.

Notice that undirected connectivity is solvable in log-space on bounded-degree graphs with *logarithmic diameter* (simply enumerate all paths of logarithmic length in the graph out of the origin vertex). Examples of graphs with logarithmic diameter are expander graphs, i.e. graphs whose second eigenvalue is bounded away from 1. Reingold's idea is to transform the input graph into a bounded-degree expander by gradually decreasing its second eigenvalue.

A natural attempt would be to square the graph. This indeed decreases the second eigenvalue, but increases the degree. To decrease the degree, Reingold uses the *zig-zag graph product* of [RVW], or the related *replacement product*. We describe his algorithm in terms of the latter product.

Given a K-regular graph X on N vertices, and an auxiliary D-regular graph G on K vertices, the replacement product $X \textcircled{r} G$ is a $D+1$-regular graph on NK vertices. On each edge (v, w) in X put two vertices, one called e_v "near" v and another called e_w "near" w, for a total of NK vertices. This can be thought of as splitting each vertex v into K vertices forming a "cloud" near v. Place the graph G on each cloud. Now for each edge $e = (v, w)$ of X, put an edge between e_v and e_w. The result is a $(D+1)$-regular graph. Notice that $X \textcircled{r} G$ is connected if and only if both X and G are.

The replacement product reduces the degree from K to $D+1$. It is proven in [RVW] (and also follows from [MR]) that when G is a good enough expander, replacement product roughly preserves the second eigenvalue of X. Suppose that X is $(D+1)$ regular and G has $(D+1)^2$ vertices and degree D. Then $X^2 \textcircled{r} G$ is again a $(D+1)$-regular graph, whose second eigenvalue is roughly the square of the second eigenvalue of X . Iterating this procedure $\log N$ times leads to a constant-degree expander on *polynomially many vertices*, since at each iteration the number of vertices grows by a factor of about D^2. On the resulting expander we can therefore solve connectivity in logarithmic space. (One also must confirm that the iterations can be computed in logarithmic space as well).

Our Algorithm.

Our approach also follows from this idea of increasing connectivity by squaring the graph. However, instead of squaring, and then reducing the degree by a zigzag product (and thus increasing the number of vertices) we will replace the squaring by derandomized squaring, which maintains the vertex set (but increases the degree). Iterating the derandomized squaring operation yields highly connected graphs with relatively small degree compared to doing the same iterations with standard squaring. In the next two paragraphs we compare the resulting graphs in each case.

Let X be a regular graph on N vertices. Squaring the graph $\log N$ times, results in the graph $X^{2^{\log N}} = X^N$ (whose edges are all paths of length N in X). This graph is extremely well connected; it contains an edge between every two vertices which are connected by a path in X. The degree however, is huge — exponential in N. We want to simulate the behavior of X^N with a graph that has much smaller degree.

Suppose that instead of standard squaring at each step we apply derandomized squaring to obtain a sequence of graphs $X_1, X_2, \ldots$. At each step the degree increases by a constant factor (instead of the degree squaring at each step).[2] For $m = O(\log N)$ the degree of X_m is only *polynomial* in N. But we will show that is as well-connected as X^N (as measured by the second eigenvalue). In particular, X_m will contain an edge between every pair of vertices s, t that are connected by a path in X. Deciding whether s, t are connected therefore reduces to enumerating all neighbors of s in X_m and looking for t. There are only polynomially many neighbors, so the search can be done in logarithmic space. We will show that computing neighbors in X_m can also be done in logarithmic space. These two facts yield a logarithmic space algorithm for undirected connectivity.

Comparing our approach to Reingold's original solution, the main way in which our algorithm differs from (and is arguably more natural than) Reingold's algorithm is that all the graphs we construct are on the *same* vertex set. Edges in the graph X_m correspond to paths of length 2^m in X. The price we pay is that the degree increases, but, thanks to the use of *derandomized* squaring, only by a constant factor (which we can afford). In contrast, each step of Reingold's algorithm creates a graph that is larger than the original graph (but maintains constant degree throughout).

1.3 Embedding Expander Graphs in Arbitrary Graphs

Another consequence of our algorithm is a logspace algorithm to find an "embedding" of an expander graph in every graph. Specifically, if X has spectral gap γ (i.e., second eigenvalue $1-\gamma$), then for $k = O(\log(1/\gamma))$, the graph X_k is an expander in the sense that it has constant spectral gap. It is embedded in X in the sense that edges in X_k correspond to paths of length $\ell = 2^k = \mathrm{poly}(1/\gamma)$ in X, and if X has degree d, then the graph X_k has degree $d \cdot t$ for $t = 2^{O(k)} = \mathrm{poly}(1/\gamma)$. In addition, it can be shown that this embedding has low congestion, in the sense that every edge of X is contained in precisely $\ell \cdot t$ of the paths. This embedding has a similar spirit to the "expander flows" of [ARV], though it does not seem to provide a better algorithm or certificate for approximating a graph's expansion.

1.4 Derandomized Squaring as a Pseudorandom Generator

Impagliazzo, Nisan, and Wigderson [INW] proposed the following pseudorandom generator. Let G be an expander graph with K vertices and degree D. Choose a random vertex $x \leftarrow [K]$, a random edge label $a \leftarrow [D]$, and output $(x, x[a]) \in [K] \times [K]$. This pseudorandom generator is at the heart of derandomized squaring.

[2]Actually, for the last $\log \log N$ steps, we use auxiliary graphs of nonconstant degree and thus the degree increases by nonconstant factors, but the degrees are chosen in such a way that the total increase is still polynomial in N.

Notice that using this pseudorandom generator to generate a pseudorandom walk of length 2 in a graph X of degree K is *equivalent* to taking a random step in the derandomized square of X using auxiliary graph G.

Impagliazzo, Nisan, and Wigderson [INW] suggested to increase the stretch of the above generator by recursion. They proved that when the graphs G used in the construction are sufficiently good expanders of relatively *large degree*, this construction fools various models of computation (including randomized logspace algorithms).[3] However, the resulting generator has seed length $O(\log^2 n)$, and hence does not prove that RL=L.

Our construction of the graph X_m in our st-connectivity algorithm is precisely the graph corresponding to using the INW generator to derandomize random walks of length 2^m in X. [4] However, we are able to use *constant-degree* expanders for G (for most levels of recursion), thereby obtaining seed length $O(\log n)$ and hence a logspace algorithm (albeit for undirected st-connectivity rather than all of RL).

Moreover, it follows from our analysis that taking the pseudorandom walk in X corresponding to a random step in X_m (equivalently, according to the INW generator with appropriate parameters) will end at an almost-uniformly distributed vertex. A pseudorandom generator with such a property was already given in [RTV] based on Reingold's algorithm and the zig-zag product, but again it is more natural in terms of derandomized squaring.

1.5 Relation to Other Graph Products

The Zig-Zag Product. The reader may have noticed a similarity between the derandomized squaring and the zig-zag product of [RVW] (which we define precisely later in the paper). Indeed, they are very closely related. When we use a square graph G^2 as auxiliary graph, the derandomized square $X \circledS G^2$ turns out to be a "projection" of the square of the zigzag product $(X \text{ ⓩ } G)^2$. In the conference version of this paper [RV], we used this observation to prove the expansion property of the derandomized squaring by reduction to that of the zig-zag product. In this version, however, we provide a direct analysis, which gives a cleaner and tight bound.

We note that the derandomized squaring has complementary properties to the zigzag product. In the zigzag product we are given a graph X and can decrease its degree while (nearly) maintaining its second eigenvalue. We must pay by slightly increasing the number of vertices. In the derandomized squaring we manage to

[3] Specifically, to fool an algorithm running in space $\log n$, they use expanders of degree $\mathrm{poly}(n)$.

[4] This holds provided that the labelling of edges in X satisfies a certain consistency condition, to be described in Sect. 3. The st-connectivity problem in general undirected graphs can easily be reduced to st-connectivity in graphs with such a consistent labelling.

decrease the second eigenvalue while maintaining the number of vertices, and we pay by slightly increasing the degree.

Derandomized Graph Products. Alon, Feige, Wigderson, and Zuckerman [AFWZ] studied a "derandomization" of a different kind of graph product, where given a graph $G = (V, E)$, we consider the graph $G^{(k)}$ whose vertex set is V^k and whose edges are $((u_1, \ldots, u_k), (v_1, \ldots, v_k))$ such that $\{u_1, \ldots, u_k, v_1, \ldots, v_k\}$ is a clique in G. A nice property of this product is that the clique number of $G^{(k)}$ is precisely the k'th power of the clique number of G, and thus this allows one to "amplify" inapproximability results for the Clique problem. A problem, however, is that the number of vertices grows exponentially with k. Thus, Alon et al. [AFWZ] showed, using random walks on expanders, how to pick a much smaller "pseudorandom" subset of vertices of $G^{(k)}$ such that the clique number behaves in much the same way. Thus, their "derandomization" is concerned with saving on the number of vertices, whereas ours is concerned with the degree, and they are interested in maintaining the clique number and similar parameters, whereas we are interested in maintaining expansion.

2 Overview of the Paper

In Section 3, we set notation and definitions, and state basic lemmas we will need later. In Section 4, we define derandomized squares and state the main lemma on them. In Section 5, we give a log-space algorithm for connectivity via iterated applications of derandomized squaring, and deduce a pseudorandom generator for walks in a graph. Section 6 extends the results to a more general notion of labelled graphs, where at each vertex, both incoming edges and outgoing edges are numbered (whereas the earlier sections only consider labellings of outgoing edges, and require the labelling to satisfy a certain consistency condition). In Section 7, we give a logspace algorithm to find an expander embedded as paths in a regular graph, with small dilation and congestion.

3 Preliminaries

Reingold, Trevisan, and Vadhan [RTV] generalized Reingold's algorithm and the zig-zag product to (regular) *directed* graphs, and working in this more general setting turns out to be useful for us, too (even if we are only interested in solving st-connectivity for undirected graphs). We present the necessary background on such graphs in this section.

Let X be a directed graph (*digraph* for short) on N vertices. We say that X is K*-outregular* if every node has outdegree K, K*-inregular* if every node has indegree K, and K*-regular* if both conditions hold. Graphs may have self-loops and multiple edges, where a self-loop is counted as both an outgoing and incoming edge. All graphs in this paper are outregular directed graphs (and most are regular).

For a K-regular graph X on N vertices, we denote by M_X the transition matrix of the random walk on X, i.e. the adjacency matrix divided by K. Let $u_N = (1/N, \ldots, 1/N) \in \mathbb{R}^N$ be the uniform distribution on the vertices of X. Then, by regularity, $M_X u_N = u_N$ (so u_N is an eigenvector of eigenvalue 1).

Following [Mih], we consider the following measure of the rate at which the random walk on X converges to the stationary distribution u_N:

$$\lambda(X) = \max_{v \perp u_N} \frac{\|M_X(v)\|}{\|v\|} \in [0, 1]$$

where $v \perp u_N$ refers to orthogonality with respect to the standard dot product $\langle x, y\rangle = \sum_i x_i y_i$ on $\mathbb{R}^N$ and $\|x\| = \sqrt{\langle x, x\rangle}$ is the L_2 norm. The smaller $\lambda(X)$, the faster the random walk converges to the stationary distribution and the better "expansion" properties X has. Hence, families of graphs with $\lambda(X) \leq 1 - \Omega(1)$ are referred to as *expanders*.

In case X is undirected, $\lambda(X)$ equals the second largest eigenvalue of the symmetric matrix M_X in absolute value. In directed graphs, it equals the square root of the second largest eigenvalue of $M_X^T M_X$.

A K-regular directed graph X on N vertices with $\lambda(X) \leq \lambda$ will be called an (N, K, λ)-graph. We define $\mathrm{g}(X) = 1 - \lambda(X)$ to be the *spectral gap* of X.

The "best mixing" graph on N vertices is a clique with a loop on each vertex. The transition matrix is J_N, which has all elements equal $1/N$. A random walk on this graph reaches uniform distribution after a single step, and the second eigenvalue is 0. The next proposition shows that the transition matrix of any graph can be decomposed into a convex combination of J_N and another matrix with matrix norm at most 1.

Definition 3.1. *For an $N \times N$ matrix C define the matrix norm $\|C\| = \max_{v \in \mathbb{R}^n} \|Cv\|/\|v\|$*

The matrix norm satisfies

- $\|AB\| \leq \|A\| \cdot \|B\|$ for every pair of matrices A, B.
- $\|A \otimes B\| \leq \|A\| \cdot \|B\|$.
- If A is the transition matrix of a graph then $\|A\| = 1$.

Proposition 3.2. *Let A be the transition matrix of an (N, D, λ) - graph. Let J_N be the $N \times N$ matrix with all entries equal $1/N$. Then $A = (1-\lambda)J_N + \lambda C$ where $\|C\| \leq 1$.*

Intuitively, this proposition says that a random step on the graph can be viewed as going to the uniform distribution with probability $1 - \lambda$ and "not getting any further from uniform" with probability λ. This intuition would be precise if C were a stochastic matrix, but it need not be.

Proof. Write $C = (A - (1-\lambda)J)/\lambda$. Since $Au_N = J_N u_N = u_N$ it follows that $C(u_N) = u_N$. For $v \perp u_N$ we have $Jv \perp u_N$ and $Av \perp u_N$ which implies that $Cv \perp u_N$. It therefore suffices to show that $\|Cv\| \leq \|v\|$ for every $v \perp u_N$. Since $Jv = 0$ and $\|Av\| \leq \lambda\|v\|$ then $\|Cv\| \leq \|v\|$, which proves the proposition. ∎

A *labelling* of a K-outregular graph X is an assignment of a number in $[K]$ to every edge of X, such that the edges exiting every vertex have K distinct labels. For a vertex v of X and an edge label $x \in [K]$ we denote by $v[x]$ the neighbor of v via the outgoing edge labelled x. We say that a labelling is *consistent* if for every vertex all incoming edges have distinct labels. Notice that if a graph has a consistent labelling, then it is K-inregular (and hence K-regular). Conversely, it can be shown (using matching theory) that every K-regular digraph has a consistent labelling.

The notion of consistent labelling described above is the same as in [HW] and [RTV]. We will work with consistently labelled graphs in this extended abstract for simplicity and to make the connection between derandomized squaring and the INW pseudorandom generator [INW] more apparent. But this condition can be relaxed by allowing each edge (u, v) to have two labels, one as an outgoing edge from u and one as an incoming edge to v, as formalized using the "rotation maps" of [RVW, RTV]. We present generalizations of our results to this setting in Section 6.

If G is a K-regular *undirected* graph, then we view it as a K-regular directed graph by replacing each undirected edge $\{u, v\}$ with two directed edges (u, v) and (v, u). One can then consider a stronger notion of consistent labelling whereby (u, v) is required to have the same label as (v, u). We call this an *undirected consistent labelling*. Note that such a labelling has the property that $v[i][i] = v$, and can be viewed as decomposing the set of edges into the union of K perfect matchings. However, this property has the disadvantages that (a) not all undirected graphs possess such a labelling (eg graphs with an odd number of vertices), (b) it is not preserved under the operations we perform (such as the squaring operation below). Therefore, even for undirected graphs, we will typically work with the basic notion of consistency given in the previous paragraphs.

The square X^2 of a graph X is the graph whose edges are paths of length 2 in X. The square of a K-regular graph is K^2-regular, and a consistent labelling of X induces a consistent labelling of X^2 in a natural way. Specifically, for a label $(x, y) \in [K]^2$, we define $v[x, y] = v[x][y]$. Notice that $\lambda(X^2) \leq \lambda(X)^2$. (This is always an equality for undirected graphs, but not necessarily so for directed graphs). We similarly define the n-th power X^n using paths of length n in X.

Like undirected graphs, the mixing time of *regular* connected directed graphs is bounded by a polynomial. One can give an inverse polynomial bound on the spectral gap provided the graph has a self-loop on every vertex.[5]

Lemma 3.3. *Let X be a connected D-regular graph with a loop on every vertex. Then $\lambda(X) \leq 1 - 1/(2D^2N^2)$.*

Proof. We will prove this by reduction to the bound for the undirected case, given by [AS]. As mentioned above, $\lambda(X)^2 = \lambda(M^T M)$. The matrix $M^T M$ is the adjacency matrix of a D^2-regular *undirected* graph Y on the vertex set of X, whose edges are pairs $\{v, w\}$ such that there exist edges (v, z) and (w, z) are edges of X (counted with multiplicity according to the number of such pairs). In other words, to obtain a neighbor of a vertex in Y, take a step on an edge X and followed by an inverse of an edge of X.

Since X contains a loop on every vertex, the graph Y contains an undirected edge $\{v, w\}$ for every directed edge $(v, w) \in E(X)$. In particular, in Y there is a loop on every vertex. It follows that Y is connected, non-bipartite, and D^2-regular. From [AS], every such graph satisfies $\lambda(Y) \leq 1 - 1/D^2N^2$. Therefore, $\lambda(X) \leq \sqrt{1 - 1/D^2N^2} \leq 1 - 1/2D^2N^2$. ∎

The next proposition shows that when the second eigenvalue is very small, the graph is very well connected - it contains a clique.

Proposition 3.4. *Let X be an $(N, D, 1/2N^{1.5})$-graph. Then X contains an edge between any pair of vertices. Indeed, for a pair of vertices v, w the probability that a random neighbor of v is equal w is at least $1/N - 1/N^2$.*

Proof. The probability distribution of a random neighbor of vertex v is the vector $M_X e_v$ where e_v is the vector which has 1 in coordinate v and 0 in the other coordinates. We need to show that every coordinate of $M_X e_v$ has value at least $1/N - 1/N^2$. Let u be the vector with value $1/n$ in all coordinates. Since $M_X u = u$, it suffices to show that $M_X(e_v - u)$ has absolute value at most $1/N^2$

[5] In the conference version of our paper [RV], instead of assuming that every vertex has a self-loop, we erroneously used the standard notion of aperiodicity (the gcd of all cycle lengths is 1). In that case, the the spectral gap can actually be zero, as shown by the following example: $G = (V, E)$ where $V = \{a, b, c, d\}$ and $E = \{(a, b), (a, c), (b, b), (b, d), (c, c), (c, d), (d, a), (d, a)\}$.

in each coordinate. As $e_v - u$ has coordinate sum zero and $\|e_v - u\| \le 2$ we know that $\|M_X(e_v - u)\|^2 \le 1/N^3$. Let m be the minimal absolute value of a coordinate of $M_X(e_v - u)$. Then $Nm^2 \le \|M_X(e_v - u)\|^2 \le 1/N^3$ which proves the result. ∎

4 Derandomized Squaring

After giving a formal definition of derandomized squaring, we will show in Theorem 4.4 that it decreases the second eigenvalue of a graph in a way comparable to squaring it.

Definition 4.1. *Let X be a labelled K-regular graph on vertex set $[N]$, let G be a labelled D-regular graph on vertex set $[K]$. The* derandomized square graph $X Ⓢ G$ *has vertex set $[N]$ and is KD-outregular. The edges exiting a vertex v are paths $v[x][y]$ of length two in X such that y is a neighbor of x in G. Equivalently, when $x \in [K]$ is an edge label in X and $a \in [D]$ is an edge label in G, the neighbor of $v \in [N]$ via the edge labelled (x, a) is $v[x][x[a]]$.*

The derandomized square may, in general, not produce an in-regular graph. However, it will do so provided that X is consistently labelled.

Proposition 4.2. *If X is consistently labelled, then $X Ⓢ G$ is KD-regular. If, in addition, G is consistently labelled, then $X Ⓢ G$ is consistently labelled.* ∎

Notice that even if X and G are consistently labelled and undirected, i.e. for every edge (u, v) there is a corresponding reverse edge (v, u), then the derandomized square $X Ⓢ G$ need not be undirected.[6] In Section 6, we present a more general formulation that is more amenable to maintaining undirectedness.

A *Cayley graph* is a graph whose vertices are elements of a group $\mathcal{G}$ and whose edges are all pairs (g, gu) for all $g \in \mathcal{G}$ and all $u \in U$ where U is a subset of $\mathcal{G}$. A Cayley graph is consistently labelled, by labelling the edge (g, gu) by u. The next observation states that if X is a Cayley graph then so is $X Ⓢ G$. We will not use Cayley graphs in other parts of the paper.

Observation 4.3. *Let X be a Cayley graph given by a group $\mathcal{G}$ and subset $U \subset \mathcal{G}$, and let G be a consistently labelled $|U|$-regular graph. Then $X Ⓢ G$ is a Cayley graph given by the same group $\mathcal{G}$ and subset $\{u_i u_j | (i, j) \in E(G)\}$.*

[6] If X satisfied the stronger notion of consistent labelling where (u, v) and (v, u) are required to have the same label, then $X Ⓢ G$ would be undirected. Alas, this stronger notion is not preserved under the derandomized square (or even the standard squaring).

Our main result on derandomized squares is that when G is a good expander, then the expansion of $X \circledS G$ is close to that of X^2.

Theorem 4.4. *If X is a consistently labelled (N, K, λ)-graph and G is a (K, D, μ)-graph, then $X \circledS G$ is an $(N, KD^2, f(\lambda, \mu))$-graph, where*

$$f(\lambda, \mu) = 1 - (1 - \lambda^2) \cdot (1 - \mu)$$

The function f is monotone increasing in λ and μ, and satisfies

- $f(\lambda, \mu) \leq \lambda^2 + \mu$,
- $1 - f(1 - \gamma, 1/100) \geq (3/2) \cdot \gamma$ *, when* $\gamma < 1/4$.

Notice that when $\mu \to 0$ (i.e. G is a good expander), then $f(\lambda, \mu) \to \lambda^2$ (i.e. $X \circledS G$ is nearly good an expander as we expect X^2 to be.). After proving the theorem, we show (Proposition 4.5) that the upper bound $f(\lambda, \mu)$ is tight in a very strong sense. No such tightness result is known for the bounds on the second eigenvalue of the zig-zag product.

In the conference version of this paper [RV], we analyze the derandomized square by reduction to the zig-zag product, obtaining a weaker bound than above. Below, we present a direct proof, which uses some of the ideas from the analysis of the zig-zag product in [RVW, RTV], but is significantly simpler. Specifically, it applies Proposition 3.2 to the expander G. Intuitively, this says that we can view the random step on G in the derandomized square as going to the uniform distribution on $[K]$ with probability $1 - \mu$, and otherwise doing no harm. In case the step on G goes to the uniform distribution, the derandomized square is identical to two independent, random steps on X. This suggests a bound of $(1 - \mu) \cdot \lambda^2 + \mu \cdot 1$, which equals $f(\lambda, \mu)$. The proof below makes this intuition formal.

Proof. Let M be the transition matrix of the random walk on $X \circledS G$. We must show that, for every vector $v \in \mathbb{R}^N$ orthogonal to the uniform distribution u_N, Mv is shorter than v by a factor of $f(\lambda, \mu)$.

In order to relate M to the transition matrices of X and G, we think of a random step on $X \circledS G$ started at a vertex u as consisting of the following steps:

1. Choose a uniformly at random in $[K]$, to go to "state" $(u, a) \in [N] \times [K]$.
2. Go to state $(u[a], a)$.
3. Go to state $(u[a], b)$, where b is a random neighbor of a in G.
4. Go to state $(u[a][b], b)$.

5. Output $u[a][b]$.

Step 1 corresponds to mapping L that "lifts" probability distributions on $[N]$ to probability distributions on $[N] \times [K]$ given by $L(v) = v \otimes u_K$, where $v \in \mathbb{R}^N$ is a probability distribution on $[N]$, $\otimes$ is tensor product and u_K is the uniform distribution on $[K]$. Step 2 corresponds to the $NK \times NK$ matrix $\tilde{A}$, where $\tilde{A}_{(u,a),(u',a')}$ is 1 iff $a' = a$ and $u' = u[a]$. Since X is consistently labelled, $\tilde{A}$ is a permutation matrix. Step 3 corresponds to the matrix $\tilde{B} = I_N \otimes B$, where I_N is the $N \times N$ identity matrix and B is the transition matrix for G. Step 4 is again given by $\tilde{A}$. Step 5 is given by the linear map P that "projects" probability distributions on $[N] \times [K]$ to probability distributions on $[N]$ given by $(Pz)_u = \sum_a z_{u,a}$. (This is inverse to Step 1 in the sense that $PL(v) = v$ for any $v \in \mathbb{R}^N$). Thus,

$$M = P\tilde{A}\tilde{B}\tilde{A}L.$$

By Proposition 3.2 we can decompose $B = (1 - \mu)J + \mu C$ where $\|C\| \leq 1$, which induces the decomposition $\tilde{B} = I_N \otimes B = (1-\mu)(I_N \otimes J) + \mu(I_N \otimes C) = (1 - \mu)\tilde{J} + \mu\tilde{C}$. Therefore

$$M = (1 - \mu)P\tilde{A}\tilde{J}\tilde{A}L + \mu P\tilde{A}\tilde{C}\tilde{A}L.$$

Now, the key observation is that

$$P\tilde{A}\tilde{J}\tilde{A}L = P\tilde{A}LP\tilde{A}L = A^2,$$

because $\tilde{J} = LP$ and $P\tilde{A}L = A$. Since $\|\tilde{A}\|, \|\tilde{C}\| \leq 1$, $\|L\| = 1/\sqrt{K}$, and $\|P\| = \sqrt{K}$, we conclude that $\|P\tilde{A}\tilde{C}\tilde{A}L\| \leq 1$. Therefore, for some matrix D with $\|D\| \leq 1$

$$M = (1 - \mu)A^2 + \mu D.$$

The last equation implies that $\lambda(M) \leq (1 - \mu)\lambda^2 + \mu$, which is equal $f(\lambda, \mu)$. ∎

The next proposition shows that the bound of Theorem 4.4 is tight in a strong sense. The proof of the proposition also clarifies the intuition of the proof of Theorem 4.4.

Proposition 4.5. *For every $K \in \mathbb{N}$ and rational $\mu \in [0, 1]$, there is a $D \in \mathbb{N}$, and an undirected (K, D, μ)-graph G such that for every (N, K, λ)-graph X with an undirected, consistent labelling, we have $\lambda(X \circledS G) \geq f(\lambda, \mu)$, for $f(\lambda, \mu) = 1 - (1 - \mu)(1 - \lambda^2)$. (Recall that in an undirected labelling we have $v[i][i] = v$ for all v, i).*

Proof. We choose G to be the undirected graph whose transition matrix is $B = \mu I_K + (1-\mu)J_K$, where I_K is the $K \times K$ identity matrix and J_K is the $K \times K$ matrix all of whose entries $1/K$. That is, a random step on G stays in place with probability μ and goes to a uniformly random vertex with probability $1-\mu$. Thus a random step on the derandomized square $X \circledS G$ amounts to taking two steps on X, using the same random edge label for both steps with probability μ and using two independent edge labels with probability $1-\mu$. The fact that X has an undirected consistent labelling implies that using the same edge label twice brings you back to the same vertex. Thus the transition matrix for $X \circledS G$ is $\mu I_N + (1-\mu)A^2$, where A is the transition matrix for X, and thus has second eigenvalue $\mu + (1-\mu)\cdot\lambda^2 = f(\lambda, \mu)$. ∎

5 A Log-Space Algorithm for Undirected Connectivity

We describe how to solve undirected st-connectivity on an undirected graph X with N vertices in logarithmic space.

Overview.

We will assume that the input graph X is 4-regular, consistently labelled and contains a loop on each vertex. Prop. 5.3 shows that this assumption does not lose generality. By Lemma 3.3, every 4-regular connected graph with a loop on each vertex has second eigenvalue $1 - \Omega(1/N)$. Our goal is to use derandomized squaring to decrease the second eigenvalue (of each connected component) to less than $1/N^3$ (we will need to square $O(\log N)$ times). By Prop. 3.4, the resulting graph must contain a clique on every connected component of X. We can therefore go over all the neighbors of s in the resulting graph and search for vertex t.

Starting with (some power of) X, we define a sequence of graphs X_m, each of which is a derandomized square of its predecessor using a suitable auxiliary graph. The algorithm works in two phases. Phase one works for $m \leq 100 \log N$, and reduces the second eigenvalue to a constant $(3/4)$, by using as auxiliary graphs a sequence G_m of *fixed-degree* expanders. We will see that the spectral gap $\mathrm{g}(X_m)$ grows by at least a factor of $3/2$ at each step. Therefore, after $m_0 = O(\log N)$ steps, we obtain an expander X_{m_0} with second eigenvalue at most $3/4$ and degree polynomial in N.

At this point we cannot use fixed-degree expanders as auxiliary graphs any more. If we did, the second eigenvalue of the derandomized square would be dominated by the second eigenvalue of the auxiliary graph, which is constant. Thus we would not be able to decrease the eigenvalue to $1/N^3$. In phase two, we therefore use auxiliary graphs G_m with non-constant degrees. Specifically, for $m > m_0$, the auxiliary graph G_m will have degree doubly-exponential in $m - m_0$. The fast growth of the degree allows the eigenvalue of the auxiliary graph to remain small enough to imply that $\lambda(X_{m+1}) \leq c \cdot \lambda(X_m)^2$ for some $c > 1$ quite close to 1. Therefore, after an additional $\log \log N + O(1)$ steps we obtain a graph X_{m_1} with second eigenvalue at most $1/N^3$.

Since the graph X_{m_1} has degree polynomial in N, we can enumerate all the neighbors of s in logarithmic space. We will show (in Prop. 5.7) that neighbors in X_{m_1} are log-space computable, making the whole algorithm work in logarithmic space.

The Auxiliary Expanders.

We will need a family of logspace-constructible constant-degree expanders with the following parameters, (which can be obtained from e.g. [GG] or [RVW]).

Lemma 5.1. *For some constant $Q = 4^q$, there exists a sequence H_m of consistently labelled $(Q^m, Q, 1/100)$-graphs. Neighbors in H_m are computable in space $O(m)$ (i.e. given a vertex name $v \in [Q^m]$ and an edge label $x \in [Q]$, we can compute $v[x]$ in space $O(m)$ and time $\mathrm{poly}(m)$).*

Definition 5.2. *Let H_m be the graph sequence of Lemma 5.1. For a positive integer N, we set $m_0 = \lceil 100 \log N \rceil$, we define a graph sequence G_m by*

$$\begin{array}{ll} \textit{When } m \leq m_0\textit{:} & G_m = (H_m) \\ \textit{When } m > m_0\textit{:} & G_m = (H_{m_0 - 1 + 2^{m-m_0}})^{2^{m-m_0}}. \end{array}$$

Neighbors in G_m are computable in space $O(m + 2^{m-m_0})$.

The Algorithm.

Let (X, s, t) be an instance of undirected st-connectivity; we want to decide whether there is a path from vertex s to t in X.

Proposition 5.3. *We may assume without loss of generality that the input graph is 4-regular, contains a loop on every vertex, and is consistently labelled.*

Proof. The easy proof appears in [Rei2]. We repeat it for completeness. We are given a (not necessarily regular) *undirected* graph X. Suppose X is described by a function that, given a vertex v of X, returns the degree $\deg(v)$ of v and an array of neighbors $v[1], \ldots, v[\deg(v)]$. Define a 4-regular *directed* graph X_{reg} whose vertices are pairs (v, i) for every vertex v of X and $0 \leq i \leq \deg(v)$. The neighbors of (v, i) are

$$\begin{aligned}
(v,i)[1] &= (v, && i+1 \bmod \deg(v)) \\
(v,i)[2] &= (v, && i-1 \bmod \deg(v)) \\
(v,i)[3] &= (v[i], && \text{location of } v \text{ in the array of neighbors of } v[i]). \\
(v,i)[4] &= (v,i). &&
\end{aligned}$$

This is equivalent to replacing each vertex v by a cycle of length $\deg[v]$, and connecting each vertex on the cycle of v to exactly one of the neighbors of v, and adding a loop on each vertex. This operation can be done in logarithmic space. The result is a 4-regular directed graph X_{reg}, and the labelling used to define the graph is consistent. ∎

Let X be a 4-regular graph with a loop on each vertex, given by a consistent labelling. Given two vertices s, t connected in X, we describe a log-space algorithm that outputs a path between s and t. For simplicity, assume that X is connected (else carry out the analysis below on each connected component of X).

Define $X_1 = X^q$, where $Q = 4^q$ is from Lemma 5.1. Define inductively $X_{m+1} = X_m \circledS G_m$. It can be verified by induction that the degree D_m of X_m is equal to the number of vertices of G_m, so the operation $X_m \circledS G_m$ is indeed well-defined. Specifically, we have $D_m = Q^m$ for $m \leq m_0$, and $D_m = Q^{m_0+2^{m-m_0}-1}$ for $m > m_0$.

Phase One.

By Lemma 3.3 we have $g(X_1) \geq 1/32N^2$. We will reduce the second eigenvalue to $3/4$. From Theorem 4.4 it follows that

$$\mathrm{g}(X_{m+1}) \geq \mathrm{g}(X_m) \cdot (3/2) \geq \mathrm{g}(X_1) \cdot (3/2)^m$$

as long as $\mathrm{g}(X_{m-1}) \leq 1/4$. Therefore for some $m < 100 \log N$ we will get $\lambda(X_m) \leq 3/4$. The inequality $\lambda(X_m) \leq 1/4$ holds for all larger m due to the monotonicity mentioned in Theorem 4.4. We deduce the following corollary.

Corollary 5.4. *Let m_0 be the smallest integer such that $m_0 \geq 100 \log N$. Then $\lambda(X_{m_0}) < 3/4$.*

Phase Two.

We now decrease the second eigenvalue from $3/4$ to $1/2N^3$.

Proposition 5.5. *For $m \geq m_0$ we have $\lambda(X_m) \leq (7/8)^{2^{(m-m_0)}}$.*

Proof. Define $\lambda_m = (64/65) \cdot (7/8)^{2^{(m-m_0)}}$, $\mu_m = (1/100)^{2^{m-m_0}}$. We will show that $\lambda(X_m) \le \lambda_m$ for $m \ge m_0$. This is true for $m = m_0$, and suppose by induction that it holds for some m. Since $\lambda(G_m) \le \mu_m < \lambda_m^2/64$ we can use Theorem 4.4 to deduce that

$$\lambda(X_{m+1}) \le \lambda_m^2 + \mu_m \le \lambda_m^2(1 + \frac{1}{64}) \le \left(\frac{64}{65}\right)^2 \cdot \left(\frac{7}{8}\right)^{2^{(m+1-m_0)}} \cdot \frac{65}{64} \le \lambda_{m+1}$$

which proves the proposition. ∎

Corollary 5.6. *Let $m_1 = m_0 + \log\log N + 10$. Then $\lambda(X_{m_1}) \le 1/2N^3$.*

By Proposition 3.4 the graph X_{m_1} contains a clique on the N vertices. Moreover, it has degree $D_{m_1} = Q^{100\log N + 2^{\log\log N+10}-1} = \mathrm{poly}(N)$. If we could compute neighbors in X_{m_1} in space $O(\log N)$ we could find a path from s to t in logarithmic space.

Proposition 5.7. *Neighborhoods in X_{m_1} are computable in space $O(\log N)$.*

Proof. Edge labels in X_m are vectors $y_m = (y_1, a_1, \ldots, a_{m-1})$ where y_1 is an edge label in X_1 and a_i is an edge label on G_i. Given a vertex v and an edge label y_m in X_m we wish to compute the neighbor $v[y_m]$ in X_m.

Every edge in X_m corresponds to a path of length 2^m in X. It suffices to give a (log-space) algorithm that, given v,y and an integer b in the range $[1, 2^m]$, returns the edge label in X of the b-th edge in this path of length 2^m. As we will see below, this edge label is actually independent of the vertex v (and thus can be computed given only y and b).

The path of length 2^m originating from v corresponding to the edge label y_m consists of two paths of length 2^{m-1} corresponding to two edges in X_{m-1}. These two edges in X_{m-1} have labels $y_{m-1} = (y_1, a_1, \ldots, a_{m-2})$ and $y_{m-1}[a_{m-1}]$, where the latter is a neighbor computation in G_{m-1}.

From these observations the algorithm is simple. If $b \le 2^{m-1}$ then solve the problem encoded by y_{m-1}, b in X_{m-1}. If $b > 2^{m-1}$ then instead set $y_{m-1} \leftarrow y_{m-1}[a_{m-1}]$, $b \leftarrow b - 2^{m-1}$, and now solve the problem encoded by y_{m-1}, b on X_{m-1}.

Here is a pseudo code for the algorithm. Write $b-1$ as a binary string $(b_{m-1}, \ldots, b_0)$, and let y_i be the string $y_1, a_1, \ldots, a_{i-1}$.

for $i = m-1$ to 0 **do**
  if $b_i = 1$ **then**
    set $y_i = y_i[a_i]$ (this is a computation in G_i).
  end if
end for

output y_0

Now we argue that this can be computed in space $O(\log N)$ when $m = m_1$. Notice that the input length to the algorithm is $m + \log D_{m_1} = O(\log N)$. By Lemma 5.1, the computation in the G_i-computation steps in the loop described in the code can be performed in space $O(m+2^{m-m_0}) = O(\log N)$, and we are done. ∎

This ends the log-space algorithm for undirected connectivity. We now use the same construction idea to generate a (log-space computable) pseudorandom generator for random walks on consistently labelled graphs.

A Pseudorandom Generator for Walks on Consistently Labelled Graphs.

We solved the undirected connectivity problem by using the fact that the graph X_{m_1} contains a clique on all the vertices (assuming X was connected). Actually, by Proposition 3.4, a random neighbor of a vertex v in X_{m_1} has distribution which is $1/N^2$-close to uniform. Every edge exiting v in X_{m_1} corresponds to a path with length 2^{m_1} (polynomial in N) in X. As the degree of X_{m_1} is only polynomial in N, we deduce that $O(\log N)$ uniformly random input bits (encoding an edge of X_{m_1}) suffice to generate a "pseudorandom" walk in X of poynomial length, such that the endpoint is almost uniformly distributed, as it would be for a truly random walk of polynomial length (which needs $\mathrm{poly}(N)$ random bits). Moreover, the edge labels in the walk do not depend on graph X, but only on the edge label chosen in X_{m_1} and the number of vertices N. Indeed, the algorithm given in Proposition 5.7 describes how to compute the labels in the output walk given the input edge label y_{m_1} in X_{m_1}. In fact, the map from y_{m_1} to the sequence of edge labels in the walk is precisely the Impagliazzo–Nisan–Wigderson pseudorandom generator [INW] constructed using the expanders $G_1, \ldots, G_{m-1}$.

We state the properties of this generator precisely and in a more general form in the following theorem.

Theorem 5.8. *For given parameters (N, D, λ) there is a pseudorandom generator PRG : $\{0,1\}^r \to [D]^\ell$ with seed length $O(\log(DN))$ and walk length*

$$\ell = O\left(\frac{\log N}{1 + \log(1/\lambda)}\right) \cdot \mathrm{poly}\left(\frac{1}{1-\lambda}\right),$$

such that for every consistently labelled *(N, D, λ)-graph X and every vertex v in X, if we choose a random seed $s \leftarrow \{0,1\}^r$ then following the walk PRG(s) from v ends at a distribution that is $(1/N^2)$-close to uniform. Given N, D, λ, and $1 \le i \le \ell$, the i'th step of PRG(s) is computable in space $O(\log(DN))$ and time* $\mathrm{poly}(\log N \log D)$.

The above theorem is more general than the one implicit in our undirected connectivity algorithm in that it produces shorter walks when the graphs are known to have better expansion than the bound of $\lambda = 1 - 1/(2D^2N^2)$ from Lemma 3.3. A pseudorandom walk generator with similar properties was given by Reingold, Trevisan, and Vadhan [RTV] based on Reingold's algorithm (which uses the zig-zag product). However, the generator does not have as simple a description as above. In particular, computing the i'th step in the walk seems to require computing all the previous $i-1$ labels of the walk (which may take time $\mathrm{poly}(N)$), rather than being computable directly as above (in time $\mathrm{poly}(\log(ND))$). Reingold, Trevisan, and Vadhan [RTV] also proved that if a similar pseudorandom generator could be given for walks on regular digraphs with arbitrary labellings (as opposed to consistent labellings), then every problem in solvable in randomized logspace is also solvable in deterministic logspace (i.e., $\mathbf{RL} = \mathbf{L}$).

Proof. To simplify the proof we will show the proof when $\lambda \leq 3/4$, and afterwards mention the approach for larger λ. Define $X_1 = X^2$, which is a consistently labelled (N, D^2, λ^2)-graph. Define X_m inductively by $X_{m+1} = X_m \,\circledS\, G_m$ as in Section 5. However, we use slightly different auxiliary graphs G_m. Similar to Lemma 5.1 one can show that for some constant Q and every D there exists a consistently labelled $(D^2Q^{m-1}, Q, 1/100)$-graph H_m such that neighbors in H_m are computable in space $O(\log D + m)$ and time $\mathrm{poly}(\log D, m)$. The auxiliary graph sequence is defined by $G_1 = H_1^k$ where $k = O(\log(1/\lambda))$ is the minimal integer such that $\lambda(H_k) \leq \lambda^4/64$ and $G_m = (H_{1+k\cdot(2^{m-1}-1)})^{k\cdot 2^{m-1}}$. Similar to the analysis of Phase two in Section 5, we obtain graphs X_m with degree $D^2Q^{k\cdot(2^{m-1}-1)}$ and second eigenvalue $\lambda(X_m) \leq (1.1\lambda)^{2^{m-1}}$. Let m_1 be the minimal integer satisfying $\lambda(X_m) \leq 1/N^3$. This holds for some m_1 satisfying $2^{m_1} = O(\log N/\log(1/\lambda))$.

We can now define the generator. The seed is an edge label in X_{m_1}, encoded by $O(\log D + k \cdot 2^{m_1})$ bits. Every edge (v, w) exiting a vertex v of X_{m_1} corresponds to a walk from v to w of length 2^{m_1-1} in X_1. This walk corresponds to a walk of length 2^{m_1} in X. This walk the is the output of the generator. As in the proof of Proposition 5.7, the edge labels in the walk do not depend on the graph X (but only on the auxiliary expanders $G_1, \ldots, G_m$).

By Proposition 3.4, walking on a random edge in X_{m_1} results in a distribution on the vertices that is $1/N^2$-close to uniform. This proves the pseudorandomness property of our generator. The seed length is $O(\log D + k \cdot 2^{m_1}) = O(\log(DN))$. The walk length is $2^{m_1-1} = O(\log N/\log(1/\lambda))$.

To compute the i-th step in the walk we use the same algorithm used in Proposition 5.7. The algorithm runs in m_1 steps, each requiring a computation in some graph G_m for some $m \leq k \cdot 2^{m_1}$, and manipulation of strings of length $O(k2^{m_1})$.

Each step requires space $O(\log(DN))$ and time $\text{poly}(\log(ND))$, and there are $m_1 = O(\log \log N)$ steps, so the total required time is $\text{poly}(\log(ND))$.

For $\lambda > 3/4$ we first take m_0 derandomized square steps with auxiliary graphs of constant degree Q. Each step increases the spectral gap by a factor of $3/2$, so when $(1-\lambda)\cdot(3/2)^{m_0} > 1/4$ we obtain a graph with spectral gap at least $1/4$. This holds for some $m_0 = O(\log(1/1-\lambda))$. We can now proceed as in the proof above. The walk length increases by a multiplicative factor of $2^{m_0} = \text{poly}(1/(1-\lambda))$, but the seed length increases only by an additive factor of $O(m_0) = O(\log N)$, since the degree of the final graph increases by a multipicative factor of Q^{m_0}. ∎

6 Extension to Two-Way Labellings

Until now, we have focused on applying the derandomized square to graphs X that are consistently labelled. Indeed, if if X is not consistently labelled, then $X \text{ⓢ} G$ may not even be inregular (in which case its stationary distribution will not be uniform). Nevertheless, working with consistently labelled graphs sufficed for our Undirected s-t Connectivity algorithm (via Proposition 5.3).

In this section, we consider a more general notion of labelling (previously used for the zig-zag product in [RVW, RTV]), and show how both the derandomized square and Theorem 4.4 can be extended to this more general notion. This extension has several benefits, and in particular addresses two deficiencies of the basic notion of consistent labelling considered in previous sections:

- Even though every K-regular digraph has a consistent labelling, it may not be possible to find such a labelling in logspace. Indeed, this problem is equivalent to decomposing a regular bipartite graph into the union of perfect matchings, and matching is not known to be in logspace. (Nevertheless, $s - t$ connectivity on regular digraphs can be reduced to $s - t$ connectivity on consistently labelled graphs, as in Proposition 5.3.) The more general labelling notion presented below is easy to achieve in logspace.

- The derandomized square of a consistently labelled *undirected* graph need not be undirected. One can impose a stronger condition on consistent labelling for undirected graphs that does ensure that the derandomized square is undirected, but alas this condition itself is not preserved under the derandomized square. (See Footnote 4.) The labelling notion presented below has an undirected analogue for which the derandomized square preserves both undirectedness as well as the labelling notion itself.

If X is a K-regular digraph, a *two-way labelling* of X provides, for each vertex v, a numbering from $1, \ldots, K$ of the K edges leaving v as well as a numbering from $1, \ldots, K$ of the K edges entering v. So each edge (u, v) has two numbers, one as an outgoing edge from u and one as an incoming edge to v. Clearly, given a K-regular digraph, a two-way labelling for it can be found in logarithmic space. A graph together with a two-way labelling can be specified by the following notion of a "rotation map," taken from [RVW, RTV].

Definition 6.1. *For a K-regular graph G on N vertices with a two-way labelling, the* **rotation map** *$Rot_G : [N]\times[K] \to [N]\times[K]$ is defined as follows: $Rot_G(v, i) = (w, j)$ if the i-th outgoing edge from vertex v leads to w, and this edge is the j-th incoming edge of w.*

Notice that the rotation function of a K-regular directed graph is a permutation on $[N] \times [K]$, and conversely, every permutation on $[N] \times [K]$ specifies a K-regular digraph on N vertices together with a two-way labelling. Observe that if a K-regular graph G has a consistent labelling, then the function $\mathrm{Rot}(v, i) = (v[i], i)$ is a permutation, corresponding to the two-way labelling that takes the incoming label for each edge to be the same as its outgoing label.

Recall that if G is a K-regular *undirected* graph, then we view it as a K-regular directed graph by replacing undirected edge $\{u, v\}$ with two directed edges (u, v) and (v, u). Then it is natural to insist that the label of (u, v) as an edge leaving u is the same as the label of (v, u) as an edge entering u. Indeed, such a two-way labelling corresponds to simply numbering the K undirected edges incident to each vertex; thus we refer to it as a *two-way labelling*. Notice that the resulting rotation map Rot is an involution, i.e. Rot^2 is the identity map. Conversely, every involution on $[N] \times [K]$ corresponds to a regular undirected graph together with an undirected labelling.

Now, we generalize the definition of the derandomized square to support two-way labellings, specified by rotation maps.

Definition 6.2. *Let X be a K-regular graph on vertex set $[N]$ with a two-way labelling, let G be a D-regular graph on vertex set $[K]$ with a two-way labelling. The* derandomized square graph *$X\circledS G$ has vertex set $[N]$ and rotation map $Rot_{X\circledS G}$ defined as follows: ($v_0 \in [N]$, $i_0 \in [K]$,$j_0 \in [D]$):*
$Rot_{X\circledS G}(v_0, (i_0, j_0))$:

1. *Let $(v_1, i_1) = Rot_X(v_0, i_0)$.*
2. *Let $(i_2, j_1) = Rot_G(i_1, j_0)$.*
3. *Let $(v_2, i_3) = Rot_X(v_1, i_2)$.*
4. *Output $(v_2, (i_3, j_1))$.*

Since the three operations above are permutations on $[N] \times [K] \times [D]$, we have indeed defined a regular directed graph, with a two-way labelling. Moreover, if X and G are undirected graphs with undirected labellings (i.e. their rotation maps are involutions), then the rotation map of $X \circledS G$ is an involution and in particular, $X \circledS G$ is undirected. Finally, we note that when the rotation maps of X and G are obtained from a consistent labelling (i.e. $\mathrm{Rot}(v, i) = (v[i], i)$), then Definition 6.2 coincides with Definition 4.1.

Just like the analysis of the zig-zag product [RVW, RTV], the eigenvalue bound on the derandomized square given by Theorem 4.4 also holds for graphs given by rotation maps:

Theorem 6.3. *If X is an (N, K, λ)-graph with a two-way labelling and G is a (K, D, μ)-graph with a two-way labelling, then $X \circledS G$ is an $(N, KD^2, f(\lambda, \mu))$-graph, where*

$$f(\lambda, \mu) = 1 - (1 - \lambda^2) \cdot (1 - \mu) \leq \lambda^2 + \mu.$$

Proof. The only change in the proof of Theorem 4.4 is that the matrix $\tilde{A}$ should now be taken to be the permutation matrix corresponding to the permutation Rot_X. The only facts used about $\tilde{A}$ in the proof were that $\tilde{A}$ is of norm at most 1, and that $P\tilde{A}L = A$. Both of these still hold. ∎

7 Embedding expanders in general graphs

Another consequence of our algorithm for undirected connectivity is a logspace algorithm to find an "embedding" of an expander graph in every regular graph with congestion and dilation that is polynomially related to the spectral gap.

Theorem 7.1. *Let X be an $(N, D, 1-\gamma)$-graph. Then there exists an $(N, \hat{D}, 1/2)$-graph $\hat{X}$ on the same vertex set with the following properties:*

- $\hat{D}/D = \mathrm{poly}(1/\gamma)$.
- *There is an embedding function f mapping edges of $\hat{X}$ to paths of length at most $l = \mathrm{poly}(1/\gamma)$ in X.*
- *Each edge of X is contained in exactly $l \cdot \hat{D}/D = \mathrm{poly}(1/\gamma)$ paths corresponding to edges in $\hat{X}$ under f.*

Furthermore, given X and the value γ, the graph $\hat{X}$ and embedding f can be computed in space $O(\log N)$.

Proof. Add self-loops to each vertex of X until it has degree that is a power Q^b of Q (where Q is from Lemma 5.1) to obtain a graph X_1, and construct any two-way labelling of X_1. (We do not use a consistent one-way labelling, because it may not be feasible to find in logspace.) It is easy to check that $\mathrm{g}(X_1) \geq \gamma/Q$. (Recall that $\mathrm{g}(()G) = 1 - \lambda(G)$ is the spectral gap of graph G.) Similar to the construction of the sequence X_m given in Section 5, define inductively $X_{m+1} = X_m \circledS H_{m+b-1}$ where H_m are defined in Lemma 5.1 and we use the generalization of the derandomized square to two-way labellings from Section 6.

One can check that the degree of X_m and the size of H_m are both $Q^b \cdot Q^{m-1}$. Observe that $\mathrm{g}(X_{m+1}) \geq (3/2)\mathrm{g}(X_m)$ as long as $\mathrm{g}(X_m) \leq 1/4$. Take m_0 to be the smallest integer larger than $10\log(Q/\gamma)$. The graph X_{m_0} has second eigenvalue at most $3/4$, and degree $\hat{D} = Q^b \cdot Q^{m_0-1} = D \cdot \mathrm{poly}(1/\gamma)$. We will embed X_{m_0} in X with the properties claimed in the theorem. Each edge of X_{m_0} corresponds to a path of length exactly $l = 2^{m_0-1} = \mathrm{poly}(1/\gamma)$ in X_1. Each such path corresponds to a path in X by ignoring the steps on the added self-loops of X_1. The path in X therefore has length at most l.

We now prove the congestion claim in the theorem. This follows by induction from the following fact: Let X be an (N, D, λ)-graph and let G be a (D, K, μ)-graph. The edges of $X \circledS G$ correspond to paths of length 2 in X, and each edge of X is covered by exactly $2K$ of these paths of length 2. It follows by induction that if one draws all the paths in X corresponding to edges of X_{m_0}, every edge of X is covered exactly $(2Q)^{m_0-1}$ paths.

Finally, we note that, even though we have used two-way labellings, the construction of the graph X_{m_0} and the embedding f can be computed in logspace. This is not as simple to see as for the case of consistent one-way labellings, but can be shown using a similar recursive algorithm to the one presented in [Rei2]. ∎

The embedding above resembles the "expander flow" embedding of [ARV], where an $(N, D, 1/2)$-graph is embedded as paths in an input graph X on N vertices. The maximal number of times an edge of X is covered by these paths depends linearly on the edge expansion of X up to a multiplicative factor of $\sqrt{\log N}$, providing a certificate for the edge expansion of X. In our embedding the number of times each edge of X is covered by paths depends polynomially on the spectral gap of X, but does not depend on the graph size N. Furthermore, we find our embedding in X in logarithmic space (rather than polynomial time).

Acknowledgments

This work emerged from of our collaborations with Omer Reingold, Luca Trevisan, and Avi Wigderson. We are deeply grateful to them for their insights on this topic and their encouragement in writing this paper. We also thank Avi, Omer, and Nandakumar Raghunathan for helpful comments on the write-up.

References

[AKL$^+$] R. Aleliunas, R. M. Karp, R. J. Lipton, L. Lovász, and C. Rackoff. Random walks, universal traversal sequences, and the complexity of maze problems. In *20th Annual Symposium on Foundations of Computer Science (San Juan, Puerto Rico, 1979)*, pages 218–223. IEEE, New York, 1979.

[AFWZ] N. Alon, U. Feige, A. Wigderson, and D. Zuckerman. Derandomized graph products. *Comput. Complexity*, 5(1):60–75, 1995.

[AS] N. Alon and B. Sudakov. Bipartite subgraphs and the smallest eigenvalue. *Combin. Probab. Comput.*, 9(1):1–12, 2000.

[ARV] S. Arora, S. Rao, and U. Vazirani. Expander flows, geometric embeddings and graph partitioning. In *STOC '04: Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, pages 222–231, New York, NY, USA, 2004. ACM Press.

[GG] O. Gabber and Z. Galil. Explicit Constructions of Linear-Sized Superconcentrators. *J. Comput. Syst. Sci.*, 22(3):407–420, June 1981.

[HW] S. Hoory and A. Wigderson. Universal Traversal Sequences for Expander Graphs. *Inf. Process. Lett.*, 46(2):67–69, 1993.

[INW] R. Impagliazzo, N. Nisan, and A. Wigderson. Pseudorandomness for Network Algorithms. In *Proceedings of the Twenty-Sixth Annual ACM Symposium on the Theory of Computing*, pages 356–364, Montréal, Québec, Canada, 23–25 May 1994.

[MR] R. A. Martin and D. Randall. Sampling Adsorbing Staircase Walks Using a New Markov Chain Decomposition Method. In *Proceedings of the 41st Annual Symposium on Foundations of Computer Science*, pages 492–502, Redondo Beach, CA, 17–19 Oct. 2000. IEEE.

[Mih] M. Mihail. Conductance and convergence of markov chains: a combinatorial treatment of expanders. In *In Proc. of the 37th Conf. on Foundations of Computer Science*, pages 526–531, 1989.

[Rei1] O. Reingold. Personal communication. December 2004.

[Rei2] O. Reingold. Undirected ST-Connectivity in Log-Space. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pages 376–385, New York, NY, USA, 2005. ACM Press.

[RTV] O. Reingold, L. Trevisan, and S. Vadhan. Pseudorandom Walks in Biregular Graphs and the RL vs. L Problem. *Electronic Colloquium on Computational Complexity* Technical Report TR05-022, February 2005. `http://www.eccc.uni-trier.de/eccc`.

[RVW] O. Reingold, S. Vadhan, and A. Wigderson. Entropy waves, the zig-zag graph product, and new constant-degree expanders. *Ann. of Math. (2)*, 155(1):157–187, 2002.

[RV] E. Rozenman and S. Vadhan. Derandomized Squaring of Graphs. In *Proceedings of the 8th International Workshop on Randomization and Computation (RANDOM '05)*, number 3624 in Lecture Notes in Computer Science, pages 436–447, Berkeley, CA, August 2005. Springer.

[Sav] W. J. Savitch. Relationships between nondeterministic and deterministic tape complexities. *J. Comput. System. Sci.*, 4:177–192, 1970.