University of Bath

PHD

Boundary conditions in Abelian sandpiles

Gamlin, Samuel

Award date:
2016

Awarding institution:
University of Bath

Boundary conditions in Abelian sandpiles.

submitted by

Samuel Lee Gamlin

for the degree of Doctor of Philosophy

of the

University of Bath

Department of Mathematical Sciences

November 2015

SUMMARY

The focus of this thesis is to investigate the impact of the boundary conditions on configurations in the Abelian sandpile model. We have two main results to present in this thesis.

Firstly we give a family of continuous, measure preserving, almost one-to-one mappings from the wired spanning forest to recurrent sandpiles. In the special case of Z^d, $d \geq 2$, we show how these bijections yield a power law upper bound on the rate of convergence to the sandpile measure along any exhaustion of Z^d.

Secondly we consider the Abelian sandpile on ladder graphs. For the ladder sandpile measure, ν, a recurrent configuration on the boundary, I, and a cylinder event, E, we provide an upper bound for $\Big|\nu(E|I) - \nu(E)\Big|$.

ACKNOWLEDGEMENTS

I would like thank everyone that has helped and supported through my PhD. I have no doubt that I would not of reached this stage without the people around me to help me along the way.

In particular my supervisor, Antal Járai, for all his guidance and his patience as I slowly learnt how to communicate my ideas clearly.

My family for the morale support and for letting me ramble on about my work to them whether they had any idea what I was talking about or not.

Finally to everyone who ever came to Drinks in the Parade, without who I probably would have lost my mind many times during this process.

CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION.

Let us begin by informally introducing the model that is the focus of this thesis, the formal definition is postponed until Chapter 2. The Abelian sandpile model is a stochastic particle model defined on a graph by a cellular automaton. It starts by assigning each vertex a number of particles to hold, known as the vertex's height. Each vertex has a maximum capacity of particles it can hold and if it ever has more than its capacity it sends particles to its neighbours, this is known as toppling. Particles are lost when vertices on the boundary topple meaning that there is a limited capacity in the graph and it will eventually stop toppling. If all vertices have less particles than their capacity we say the configuration is stable. Generally a vertex's capacity is taken to be the degree of the vertex and thus in this case the system will be stable when each vertex, v, has height in $\{0, \dots \deg(v) - 1\}$. When the system is stable we randomly choose a vertex and give it an extra particle.

This model, and similar variations, have arisen in several different contexts, probably most notably as the chip-firing game, see [18]. Our motivation for studying this model follows from the statistical physics background.

In [3] Bak, Tang and Wiesenfeld introduced the idea of self-organized criticality. Many dynamical systems in nature have been found to be attracted to some critical point where it demonstrates power law behaviour. The authors wanted to describe a dynamical system that would be robust, in the sense that perturbations in the original state would not be observable after a reasonable period of time had past. This would mean that fine tuning of any input parameters would not be needed in order for the power law behaviour to arise. Many of the existing models at the time studied phase transitions, these required this fine tuning which is at odds with the abundance of natural occurrences of this kind of behaviour. This robustness of self-organized criticality would suggest that this is a more plausible explanation for their existence. In their paper they gave a "toy example" that demonstrated the properties they were after, this example would later be named the Abelian sandpile model.

Dhar [11] was the first to study the model in its own right, realising that it was imbued with many advantageous properties that made explicit calculations possible. This means it has the capacity to demonstrate potentially important underlying principles of self-organized criticality and is a useful tool for further study into these kinds of systems. This has led to it being one of the primary models used to study self-organised criticality.

The model has shown connections to a variety of different areas, further suggesting that it may be able to highlight interesting properties of such systems and thus making it a model worthy of further research. Crucial to our analysis will be the close relationship between the sandpile model and the uniform spanning tree of a graph. This in turns leads to a strong connection to (loop-erased) random walks.

The objective of this thesis, as suggested by its title, was to try and answer the question "How far away does the configuration feel the impact of the boundary conditions in the sandpile model?" In particular we are interested in the stable configurations that could occur after the dynamics have been running for a long time, we refer to these as the "recurrent" states. We postpone a more formal definition to later in this chapter.

The reason that this kind of question is difficult is due to the fact that there are inherent global aspects to a recurrent stable sandpile configuration. For example, to see if a configuration is recurrent it does not suffice to consider any number of subsets of the configuration and it can only be checked when the configuration is viewed as a whole. However the fact that any one step in the dynamics of the model involve a finite number of vertices gives hope that these global factors should not be the dominant aspect in the model's behaviour. This does indeed seem to be the case. It is the influence of these global factors that most of the work in this thesis is aimed towards helping us control.

1.1 Outline of Thesis.

The results of this thesis are split into two main chapters, each takes a different approach to try and partially answer the question about boundary conditions.

Firstly we are concerned with a general graph with a particular type of boundary condition, namely taking a subset of a recurrent configuration. Secondly we consider a much more general boundary condition, where we specify the heights of the vertices in the sandpile at the boundary, but we can only show that this holds on a particular type of graph. A more detailed break down is as follows.

In Chapter 2 we formally introduce the sandpile model as well as results that are fundamental to the work in this thesis, including Wilson's algorithm and the standard burning bijection.

The main result of Chapter 3 is the construction of a family of continuous, measure

preserving, injective mappings between spanning forests and sandpiles up to a set of zero measure. Our mappings can be constructed on general infinite graphs G satisfying a common condition. The advantage of our new maps, as opposed to the known standard mapping, will be that it behaves well when we take the limit with respect to an infinite graph.

As an application of our new bijection, we show that it yields a coupling between the uniform measures on recurrent sandpiles on an infinite graph and a subgraph, that we can analyse on $\mathbb{Z}^d$, $d \geq 2$. This leads to a power law upper bound on the rate of convergence of the measure on the subgraph to the measure defined on the whole graph.

Hence if we are interested in a sandpile event it suffices to consider the sandpile configuration in a sufficiently large subgraph that includes the vertices that determine the event. This implies that the global aspect in the recurrence of a sandpile has limited influence on the configuration in this setting.

In Chapter 4 we consider the Abelian sandpile on ladder graphs, a graph of the form $[1, N] \times \mathbb{N} \subset \mathbb{Z}^2$. The purpose of this chapter is to investigate the dependence of the sandpile configuration on heights of vertices in a different part of the graph.

Suppose we are given two sandpile configuration events E and F that are determined by the height at vertices in some disjoint sets V_1 and V_2 respectively. We show that if V_1 and V_2 are sufficiently far apart, then $\mathbb{P}(E|F) \approx \mathbb{P}(E)$. Moreover we show that the error in this approximation is exponentially decreasing with respect to the distance between V_1 and V_2. We provide a quantitative estimate for the rate of decay with respect to the distance between them.

In the appendix A we present an encoding of the burning process on Ladder graphs and propose a use for it by stating a conjecture.

1.2 Details of publications

Chapter 3 is based on joint work that has been published, [15]. Some adaptations have been made to it so that the thesis reads better when viewed as a whole.
Gamlin, Samuel L. ; Járai, Antal A.
Anchored burning bijections on finite and infinite graphs.
Electronic Journal of Probability 19 (2014), no. 117, 23 pp. MR3296533

CHAPTER 2

BACKGROUND INFORMATION.

2.1 Graphs, spanning forests and groves.

The Abelian sandpile model is defined with respect to an underlying graph, therefore for clarity we will state the main notations that we use relating to graphs.

A *graph*, $G = (V, E)$, consists of a set of vertices, V, and a set of edges, E, that connect the vertices.

Definition 2.1. *i) A* planar *graph is one that can be embedded into* $\mathbb{R}^2$ *in such a way that none of the edges of the graph intersect except at vertices.*
ii) A multigraph *is a graph that allows two vertices to be connected by more than one edge.*
iii) A graph is connected *if for any two vertices in the graph there is a path of edges in* E *between them.*
iv)A graph $G_1 = (V_1, E_1)$*, such that* $V_1 \subset V$ *and* $E_1 \subset E$*, is* simply connected *if* G_1 *is connected and* $(V \setminus V_1, E \setminus E_1)$ *is also connected.*
v) A directed *graph is one where each edge* $e \in E$ *has a fixed direction. We say that* e *starts at* $\mathrm{tail}(e)$ *and ends at* $\mathrm{head}(e)$*.*
vi) Two vertices are said to be neighbours if they are connected by an edge.
vii) Given a set W *we write* ∂W *for the set of vertices in* $W^c := V \setminus W$ *that have a neighbour in* W*. Whilst* $\partial_i W$ *is a subset of vertices in* W *that have a neighbour in* W^c*.*
viii) $\deg_W(v)$ *is the degree of* v *in the subgraph induced by* $W \subset V$*. If a subscript is not specified then, unless clear from the surroundings to which graph it is referring, we will assume we mean in the whole graph so here* $\deg(v) = \deg_V(v)$*.*
ix) The graph distance between two vertices x, y *is denoted* $\mathrm{dist}_G(x, y)$*, if the graph is clear from the context we will drop the subscript* G*. For a set* A *we define* $\mathrm{dist}(x, A) := \min\{\mathrm{dist}(x, y) : y \in A\}$*.*
x) A graph is said to be transient *if it contains a vertex,* v*, such that a random walk started from* v *is transient. A graph that is not transient is said to be* recurrent*. xi) A*

graph $G_1 = (V_1, E_1)$ *is a* subgraph *of* $G = (V, E)$ *if* $V_1 \subset V$ *and* E_1 *is the set of edges between the vertices of* V_1 *induced from* E*.*

For finite graphs we will often define our graph as $G = (V \cup \{s\}, E)$, that is the graph has a distinguished vertex s, called the sink.

For an infinite graph $G = (V, E)$ we introduce the concept of an *exhaustion*. This is defined in terms of an increasing sequence of finite sets $V_1 \subset V_2 \subset V_3 \subset \dots V$ such that $\cup_{n=1}^{\infty} V_n = V$.

There are two main boundary conditions that a subgraph will have in this thesis, called free and wired. Using an exhaustion there are natural identifications to subgraphs for each of the respective boundary conditions.

When we consider the free boundary condition we define $G_n^F = (V_n, E_n)$ where $E_n \subset E$ is the set of edges whose end vertices are both contained in V_n. When we prescribe the wired boundary condition we define $G_n^W = (V_n \cup \{s\}, \tilde{E}_n)$. Similar to E_n, $\tilde{E}_n$ consists of all edges whose end vertices are both in V_n but also contains a new set of edges which attach to the sink. For each edge in E that has one end attached to a vertex, v, in V_n and the other end attached to a vertex in $V \setminus V_n$ we include an edge from v to s in $\tilde{E}_n$. Equivalently this can be seen as identifying all vertices in $V \setminus V_n$ to a single vertex s and removing any loop edges.

Another boundary condition that is worth briefly mentioning is the so called Partially wired boundary. This is prescribed as a mixture of the other two boundary conditions, where some subsets of ∂V_n have been wired together and had loop edges removed whilst other vertices act as a free boundary.

An important tool for extracting the relevant information from an underlying graph is to consider their spanning trees.

Definition 2.2. *i) A* loop *is a sequence of distinct edges that define a path with the same start and end vertex.*
ii) A spanning tree *of a connected graph* $G = (V, E)$ *is a subgraph* $T = (V, F)$ *with* $F \subset E$ *such that* T *is connected and there is no loop in* T*.*
ii) A spanning forest *of* G *is a subgraph* $T = (V, F)$*, where we drop the restriction that* T *is connected, so only require that* $F \subset E$ *does not contain any loops.*

We denote the uniform measure on spanning trees of a finite connected graph, G_n, by $\mathsf{UST}_{G_n^F}$ or $\mathsf{UST}_{G_n^W}$ respective to the boundary conditions. Let $T_n = (V_n, F_n)$ be a spanning tree on G_n. If we then take the weak limit with respect to an exhaustion we produce a measure on infinite graphs which is denoted FSF, respectively WSF, which is concentrated on spanning forests of G. This was shown on $\mathbb{Z}^d$ in [44]. Formally let $T = (V, F)$ be a spanning forest of G and for any finite $K \subset B \subset E$

$$\mathsf{WSF}(F \cap B = K) = \lim_{n \to \infty} \mathsf{UST}_{G_n^W}(F_n \cap B = K)$$
$$\mathsf{FSF}(F \cap B = K) = \lim_{n \to \infty} \mathsf{UST}_{G_n^F}(F_n \cap B = K).$$

On $\mathbb{Z}^d$ the limit is independent of the choice of exhaustion and boundary condition, free, wired or partial, this is implicitly shown by the proofs of [44] but was first explicitly shown by Häggström [17]. See [37] for details with regards to the more general case. Unless otherwise specified we will henceforth assume that we have a wired graph.

It is well-known and easy to see that in a spanning forest of an infinite graph WSF-a.s. all components are infinite trees.

A component is said to have *one end* if there is finite symmetric difference between any two infinite self-avoiding paths in the component.

We will primarily be interested in graphs that satisfy the condition

$$\text{WSF-a.s. all components have one end.} \tag{2.1}$$

While, in general, condition (2.1) is difficult to verify, it is known to hold on a large class of graphs, including $\mathbb{Z}^d$, $d \geq 2$; see [44, 7, 38, 37].

Another important property of a graph is how many trees a uniform spanning forest, T, will have. For example, Pemantle, [44], investigated this in the case of $\mathbb{Z}^d$. For $d = 1, 2, 3, 4$, WSF-a.s T is connected, that is it is a spanning tree. For $\mathbb{Z}^d$, $d \geq 5$, WSF-a.s T is a spanning forest with an infinite number of spanning trees.

Note that in the case of $d = 1$ condition (2.1) is not satisfied, namely it has two ends. This means that when we require this property, which we do for all results relating to an infinite graph in the whole of this thesis, the proofs would not hold for sandpile configurations on $\mathbb{Z}$. However as sandpiles on $\mathbb{Z}$ are trivial, there is little merit to trying to extend the results to hold in this case.

An important property of a spanning tree is where paths from different vertices first meet. Let $\mathfrak{T}$ be a spanning tree of a finite graph, $G = (V \cup \{s\}, E)$. The *earliest common ancestor* of a set A, denoted $\mathsf{eca}(A)$, is a vertex $v \in \mathfrak{T}$ such that from every vertex in A the directed path to s passes through v. Moreover any other vertex that has this property must be on the path from v to s.

Note that for a spanning tree of an infinite graph we can similarly define the earliest common ancestor by considering paths to infinity instead of paths to s. For a graph with the one-end property this will be well defined. However for a spanning forest we can only define the concept of an earliest common ancestor when we consider vertices that are restricted to one component.

2.2 Wilson's algorithm.

Wilson's algorithm is a method to generate a spanning forest for a graph. It relies on the idea of a Loop erased random walk to do this.

The *loop-erased random walk*, LERW, is produced from a random walk. Let (X_n) be a random walk of finite length. The idea is to chronologically travel along the path removing any cycles as they are formed. Suppose $j = \min\{n \in \mathbb{N} : \exists i <$

n with $X_n = X_i\}$ then this is the first loop that needs to be erased leaving the path $(X_1, X_2, \dots X_i, X_{j+1}, X_{j+2} \dots)$. This path can then be relabelled and the process repeated until no more cycles exist. Given a finite path π its loop-erasure is denoted $\mathrm{LE}(\pi)$.

It is worth noting that the distribution of a LERW is not the same as that of a self avoiding random walk, see [32] or [39] for more information about self-avoiding walks. However we do have the useful fact which is usually called the Domain Markov property, [35]. A LERW from x to y on the domain D, has the same distribution as taking the first step according to the LERW transition probabilities, say to z, and then, if $P_z(\tau_y < \tau_x) > 0$, running a LERW from z to y on the domain $D \setminus \{x\}$.

Wilson's algorithm can be split into three different cases depending on whether the graph is finite, recurrent or transient.

Wilson's algorithm on finite graphs.

Let $G = (V \cup \{s\}, E)$ be a finite connected graph. Start by defining a set $\mathfrak{T}_0 := \{s\}$

Choose a vertex $v_1 \in V$. Run a random walk from v_1 until the first time it hits s, take the loop erasure of this path, denote it by α_1. Set $\mathfrak{T}_1 := \{s\} \cup \alpha_1$, this is therefore a subgraph containing vertices and edges.

Assume that $\mathfrak{T}_{i-1}$ has been defined and let $V(\mathfrak{T}_{i-1})$ be the vertex set for this subgraph.

Choose a vertex $v_i \in V \setminus V(\mathfrak{T}_{i-1})$ in a manner that depends only on past walks.

Next run a random walk from v_i until the first time it hits a vertex in $V(\mathfrak{T}_{i-1})$. Take the loop erasure of this path, denote it by α_i. Then define a new set $\mathfrak{T}_i := \mathfrak{T}_{i-1} \cup \alpha_i$.

As the graph is finite there exists a $k \in \mathbb{N}$ such that $V(\mathfrak{T}_k) = V$. Therefore we can inductively define $\mathfrak{T}_i$ for $i \leq k$ at which point the algorithm terminates.

As shown in the next lemma the method of choosing the next vertex from which to start a new walk from is independent from the final distribution as long as it only relies on the past. One common strategy for making this choice is to fix an ordering of V before the algorithm begins. When it is time to chose v_i the first vertex, with respect to this ordering, that is not an element of $V(\mathfrak{T}_{i-1})$ is chosen.

Lemma 2.3. *The subgraph, $\mathfrak{T} := \mathfrak{T}_k$, Wilson's algorithm generates is a spanning tree. Moreover $\mathfrak{T}$ is distributed accordingly to UST_G and is independent of the choices made in the algorithm.*

Proof. This was first proven by Wilson [50], see also [37].

Firstly the output of the algorithm will be a spanning tree because every vertex is connected to s and there can not be a loop in $\mathfrak{T}$ by construction. By recurrence of random walks this occurs in finite time with probability 1. In order to consider the distribution we introduce the idea of cycle popping.

To each vertex $v \in V$ we associate an i.i.d. sequence of arrows $\{e_i^v : i = 1, 2, \dots\}$, where e_i^v is an oriented edge with $\mathrm{tail}(e_i^v) = v$ and $\mathrm{head}(e_i^v)$ uniformly random among

the neighbours of v. The stacks associated to different v are independent. We say that e_i^v has colour i, and we envision e_1^v lying directly above e_2^v in the stack, and similarly, for all k, e_k^v lying directly above e_{k+1}^v. An oriented cycle $\mathcal{C}$ is associated the *weight* $w(\mathcal{C}) = \prod_{u\in\mathcal{C}} \deg_G(u)^{-1}$. Sometimes we will need to consider *coloured cycles*, that is, a cycle consisting of some arrows $e_{i_1}^{v_1}, \ldots, e_{i_r}^{v_r}$. We will use bold characters, like $\mathbf{C}$, to denote coloured cycles. In this case, $\mathcal{C}$ will denote the cycle obtained from $\mathbf{C}$ by ignoring the colours.

Wilson's algorithm [50] is based on the idea of *cycle popping* that we now describe. We start with a configuration of stacks of arrows, as described above. We refer to the arrows *in position* 1 as lying on top of the stack and say that initially e_i^v is *in position* i. Suppose that arrows $e_1^{v_1}, \ldots, e_1^{v_r}$ on top of the stacks form a coloured cycle $\mathbf{C}$. By *popping* $\mathbf{C}$, we mean removing the arrows in $\mathbf{C}$ from the stacks, and shifting the positions of the arrows beneath them upwards. That is: after popping $\mathbf{C}$, $e_k^{v_j}$ will be in position $k-1$ for $j = 1, \ldots, r$, $k \geq 2$. Similarly, if at any later time some arrows $e_{i_1}^{v_1}, \ldots, e_{i_r}^{v_r}$ are all in position 1 and form an oriented cycle $\mathbf{C}$, we may pop them and shift the arrows beneath them upwards.

If we trace the edges given by the stacks of arrows, by following the i^{th} arrow in the stack upon the i^{th} visit to a vertex, we will create a random walk path. Thus one way of generating the stacks is by running successive random walks. Observe that popping cycles on top of the stacks in the order that a random walk following arrows reaches them is equivalent to running a LERW on the graph.

This relationship between random walks and stacks of arrows makes it clear that Wilson's algorithm corresponds to one particular method for choosing the order in which to pop cycles. Also that the method for choosing which vertex to start a walk from in Wilson's algorithm simply corresponds to selecting a different order in which to pop cycles. Hence it is necessary to show that the order that cycles are popped does not influence the spanning tree obtained; uniformity of the spanning tree will then follow from the uniform selection of the stacks of arrows as explained later.

Observe that if two disjoint cycles are popped then the order of popping is irrelevant, whilst for two intersecting coloured cycles there is only one order in which they could be popped.

Now to see that the order of cycle poppings is irrelevant we will consider the sequence, $\mathcal{C}_1, \ldots \mathcal{C}_N$, generated when Wilson's algorithm has been used to decide which cycle to pop at each step. As Wilson's algorithm terminates in finite time, with probability 1, we can assume that after a finite number of cycles have been popped there are no more coloured cycles on top of the stacks.

Suppose we have another sequence of poppings $\mathbf{C}'_1, \mathbf{C}'_2, \ldots$. If $\mathcal{C}_1 = \mathcal{C}'_1$ then we can perform the first popping in both sequences and consider $\mathbf{C}_2, \ldots \mathbf{C}_N$ and $\mathbf{C}'_2, \mathbf{C}'_3, \ldots$. Hence, without loss of generality, assume that $\mathbf{C}'_1 \neq \mathbf{C}_1$. There exists $k = \min\{j \leq N : \mathcal{C}'_1 \cap \mathcal{C}_j \neq \emptyset\}$, if no such cycle existed then $\mathcal{C}'_1$ would be disjoint from the other cycles and thus all of its arrows would remain on top of the stacks after $\mathbf{C}_1, \ldots \mathbf{C}_N$

had been popped which would mean after they had been popped $\mathbf{C}'_1$ could be popped. This contradicts the fact that Wilson's algorithm terminates when there are no more coloured cycles on top of the stacks.

Arbitrarily choose $v \in \mathcal{C}'_1 \cap \mathcal{C}_k$. The arrow on top of the stack at v will be used in both cycles, otherwise k would not be minimal or $\mathbf{C}'_1$ could not be popped. Next consider the vertex, w, that the arrow pointed towards. By the same argument both cycles would use the arrow with colour 1 at w because this is the first cycle to visit w that could of been popped. Hence by following the arrows and repeating this argument for each vertex in the cycle we see that $\mathbf{C}'_1 = \mathbf{C}_k$.

Therefore we could use the order $\mathbf{C}'_1, \mathbf{C}_1, \dots \mathbf{C}_{k-1}, \mathbf{C}_{k+1}, \dots \mathbf{C}_N$ to pop cycles instead of the sequence given by Wilson's algorithm without changing the arrows that will be on top of the stacks at the end of the process. This is because $\mathbf{C}'_1$ is disjoint from $\mathbf{C}_1, \dots \mathbf{C}_{k-1}$, so by the above observation the order they are popped can be exchanged. Now $\mathbf{C}'_1, \mathbf{C}_1, \dots \mathbf{C}_{k-1}, \mathbf{C}_{k+1}, \dots \mathbf{C}_N$ and $\mathbf{C}'_1, \mathbf{C}'_2, \dots$ have the same cycle to begin their respective sequences so we could pop this cycle and repeat the above argument with the remaining sequences of cycles. After N iterations we would have popped $\mathbf{C}'_1, \dots \mathbf{C}'_N$, moreover we would have popped $\mathbf{C}_1, \dots \mathbf{C}_N$, although possibly in a different order. By choice of the sequence this means that there are no more cycles to be popped on top of the stacks of arrows. Thus the two sequences must be permutations of the same set of coloured cycles.

If we think of the stacks of arrows as being given to us beforehand it is clear that the choices made during Wilson's algorithm can not impact on the output of the process.

Suppose we have a set of coloured cycles $\mathfrak{C}$ and a spanning tree T. Then we can deduce the colour of edges in the spanning tree from knowledge of the cycles popped, enabling us to recreate the observed parts of the stacks of arrows. Note if we had any other spanning tree T' then $(\mathfrak{C}, T')$ also defines a set of stacks of arrows. Now the probability of the stacks of arrows generating the pair $(\mathfrak{C}, T)$ is equal to the probability of having the correct coloured arrow at every height in each stack. As each arrow is chosen independently and uniformly the probability of having $(\mathfrak{C}, T)$ is proportional to the weights of the coloured cycles and the tree. To define the weight of a tree, T, firstly note we can assign a unique direction to each edge in T by having it oriented towards s. Then the weight is given by $w(T) = \frac{1}{Z} \prod_{e \in T} \deg(\text{tail}(e))^{-1}$, where Z is a normalisation constant that is fixed by the underlying graph G. Therefore we have that

$$\mathbb{P}(\text{output the spanning tree } T) = \sum_{\substack{\mathfrak{C}\text{: set of} \\ \text{coloured cycles}}} w(\mathfrak{C}) w(T).$$

Finally as the cycles that are popped and the final spanning tree are independent we can conclude that the probability that Wilson's algorithm outputs a given spanning tree is proportional to the weight of that tree and hence the outputted trees are distributed according to UST_G. □

This algorithm can be generalised for use on infinite graphs where it will produce spanning forests distributed according to WSF. There are two ways to generalise the algorithm depending on the underlying graph.

Wilson's algorithm on recurrent graphs.

The algorithm proceeds in the same way as it did for the finite case with the only difference occurring in the first step. Rather than starting with the sink, which does not exist in the infinite graph, we can choose any vertex v_0 and take $\mathfrak{T}_0 = \{v_0\}$. Then following the above algorithm we will generate a spanning tree. This follows from observing that from the point of view of an undirected spanning tree of a finite graph there is nothing special about the vertex s, therefore we could have equally chosen any other vertex to be the initial vertex to be placed in the tree. Also although the graph is infinite, due to recurrence every walk will eventually hit v_0 or a previous path and hence each step will terminate in finite time almost surely.

If we were interested in the restriction of the spanning tree onto a finite subset, D, of this infinite graph then we would chose the start vertex, $v_0 \in D$ and then by recurrence each random walk would visit a finite number of vertices and once a random walk has been run from each vertex of D they will be connected to the spanning tree. Moreover the restriction of T to D is now fixed. This is the idea behind proving the existence of WSF and that WSF = FSF in recurrent graphs.

Wilson's algorithm on transient graphs.

If the underlying graph is transient then a slightly different method is needed because a random walk may not hit the previous paths. In this case the algorithm is started by running an infinite loop erased random walk. Note that due to the transient nature of the graph that a vertex will only be visited a finite number of times almost surely and thus after the final visit to a vertex the edge connected to it in the tree will be fixed. Therefore the LERW will converge to a loop erased path. Then using this as the first path the algorithm can proceed in the same manner as the finite case except that the random walks are run until they either hit the previous paths or if a random walk does not intersect the paths we again take the loop erasure of its infinite path. Due to some walks being infinite, this method is also sometimes known as Wilson's algorithm rooted at infinity. This method was first described in [7, Theorem 5.1], where, similar to the finite case, the idea of cycle popping was used.

The key to this generalisation is that Loop-erasure also makes sense for infinite paths π, as long as π visits every vertex finitely often. To describe Wilson's method rooted at infinity, order the vertices of $\mathbb{Z}^d$ arbitrarily as $v_1, v_2, \ldots$. Starting from v_1, follow the arrows on top of the stacks, and whenever a cycle is completed, pop that cycle. The trajectory traced by this walk is a simple random walk $\{S^{(1)}(m)\}_{m\geq 0}$ under $\mathbb{P}$, the underlying probability measure for the stacks of arrows. Due to transience, every

vertex is visited only finitely many times, with probability 1. Hence, on this event, there is a well-defined configuration of stacks of un-popped arrows, after the entire trajectory of $S^{(1)}$ has been traced. On top of the stacks now lie $\mathfrak{F}_1 := \mathrm{LE}(S^{(1)}[0,\infty))$, and unexamined arrows everywhere else. Next, starting from v_2, again follow the arrows on top of the stacks, popping any cycles that are completed. The trace of the path will now be a random walk $S^{(2)}[0,\tau^{(2)}]$, where $\tau^{(2)} \in [0,\infty]$ is the first hitting time of $\mathfrak{F}_1$. Upon hitting $\mathfrak{F}_1$, a segment of $\mathfrak{F}_1$ is retraced without encountering any further cycle, and on top of the stacks will lie $\mathfrak{F}_2 := \mathfrak{F}_1 \cup \mathrm{LE}(S^{(2)}[0,\tau^{(2)}])$, with unexamined arrows everywhere else. Continue this way with $v_3, v_4, \ldots$. With probability one, from each stack only finitely many arrows get popped, hence the procedure reveals a random spanning forest T. Due to [7, Theorem 5.1], T is distributed according to WSF.

2.2.1 Generating groves.

In Chapter 4 we will be interested in a particular kind of spanning forest known as a grove. The terminology of a *Grove* was introduced in [9], [45] and [31].

Definition 2.4. *A grove on a graph G with respect to $\Lambda \subset \partial_i G$ is a spanning forest of G such that every component of the forest contains at least one vertex of Λ.*

For a grove, $\mathfrak{G}$, with respect to Λ, denote the components of $\mathfrak{G}$ by $\mathfrak{g}_1, \mathfrak{g}_2 \ldots$. Then $\mathfrak{G}$ induces a partition on Λ, where a component in the partition is given by $\Lambda_i := \mathfrak{g}_i \cap \Lambda$, we will refer to this partition as the *connection pattern* of G. When G is planar then the induced partition will be non-crossing.

Definition 2.5. *The* backbone, $\mathfrak{b}_{\mathfrak{G}}$, *of a grove $\mathfrak{G}$ with respect to Λ is a subset of the graph where only vertices and edges that are on a path between vertices of Λ are included. This includes all vertices in Λ even those which are only connected to themselves via a empty path. Let $\mathfrak{b}_i$ be the part of the backbone that is connected to Λ_i.*

We want to instil a sense of direction onto the groves, therefore in each component we will fix a vertex of Λ to be the *root*, such that all edges on the backbone are directed towards the root in their component.

Henceforth when we discuss partitions of Λ we will assume that we also know which vertex is a root in each component, if we wish to emphasise that we know the root we will call it a *directed partition*. When such a partition forms part of a grove we use the terminology *rooted grove*.

We want to use the idea of Wilson's algorithm to be able to generate a grove conditioned upon a given connection pattern on Λ, that is groves that induce the partition $(\Lambda_1, \Lambda_2 \ldots)$.

Definition 2.6. *Let $\mathfrak{G}$ be a grove with partition $(\Lambda_1, \Lambda_2 \ldots)$ which contains a component $\mathfrak{g}_1$, with $\Lambda_1 = \{v_1, \ldots v_n\}$, $n \geq 1$. By* resampling $\mathfrak{b}_1$ *we mean the process which*

takes the backbone $\mathfrak{b}_{\mathfrak{G}}$ removes $\mathfrak{b}_1$ and replaces it with the set of LERWs, $\mathfrak{b}'_1$, we now describe.

If $n = 1$ then the component $\mathfrak{b}_1$ consists solely of the vertex v_1 and so we must have that $\mathfrak{b}'_1 := \{v_1\} = \mathfrak{b}_1$. Therefore the process of resampling $\mathfrak{b}_1$ could not change the backbone in this case.

Now for the non-trivial cases when $n \geq 2$ there will be a root in Λ_1 say it is v_i. Take the loop erasure of a random walk from v_1 conditioned to hit v_i before intersecting $\mathfrak{b}_2 \cup \mathfrak{b}_3 \cup \dots$, call this path $\mathfrak{b}'_1(1)$. Note this walk is only allowed to hit Λ at a vertex in $\{v_1, \dots v_n\}$. We now proceed iteratively.

Suppose v_j is not on a previous path to v_i but v_k is already on a path for all $k < j$. Let $\mathfrak{b}'_1(j-1)$ be the set of paths connected to $v_1, \dots, v_{j-1}$ Then run a random walk from v_j conditioned to avoid vertices in $\mathfrak{b}_2 \cup \mathfrak{b}_3 \cup \dots$. Terminate the walk when it first hits a vertex that is in $\mathfrak{b}'_1(j-1)$, call this path α. Define $\mathfrak{b}'_1(j) := \mathrm{LE}(\alpha) \cup \mathfrak{b}'_1(j-1)$.

When $v_1, \dots v_n$ are all connected to v_i the process will terminate and it will have produced a new component of the backbone $\mathfrak{b}'_1 := \mathfrak{b}'_1(n)$. The set $\mathfrak{b}'_1 \cup \mathfrak{b}_2 \cup \dots$ will be a backbone of the graph.

Lemma 2.7. *Let $\mathfrak{b}_1 \cup \mathfrak{b}_2 \cup \dots$ be a backbone of a grove $\mathfrak{G}$ that is chosen uniformly from the set of all backbones that induce the partition $(\Lambda_1, \Lambda_2, \dots)$. If $\mathfrak{b}'_1$ is a resampling of $\mathfrak{b}_1$ then $\mathfrak{b}'_1 \cup \mathfrak{b}_2 \cup \mathfrak{b}_3 \cup \dots$ is also distributed uniformly amongst backbones of $\mathfrak{G}$ that induce the partition $(\Lambda_1, \Lambda_2, \dots)$.*

Before we prove this lemma we will introduce an adaptation of Wilson's algorithm that generates a grove with a given partition on Λ.

Suppose we have a graph G and a desired partition of Λ, $p = (\Lambda_1, \Lambda_2 \dots)$.

Step 1. Choose a set $\Theta \subset \Lambda$ such that $\Theta \cap \Lambda_i = \{v_i\}$ for each $i \geq 1$. Set an ordering, $\prec$, of the vertices of G that depends only on the structure of G.

Step 2. Identify the vertices $v_1, v_2, v_3, \dots$ into one vertex which we will call the sink, s, this can be done in such a way that for any edge connecting to s we know which of the vertices $v_1, v_2, v_3, \dots$ the edge was connected to in the original graph. Call this new graph G_s.

Step 3. Run Wilson's algorithm using vertices from Λ_2 as the start vertices, with the order determined by $\prec$. When all vertices in Λ_i are connected we iteratively proceed by using vertices in Λ_{i+1} as the start vertices.

Step 4. Once all vertices in $\Lambda \setminus \Lambda_1$ are connected use vertices of Λ_1 as the start vertices in Wilson's algorithm. We use $\prec$ to decide on the order within Λ_1.

Step 5. When all vertices in Λ are connected to s, continue with Wilson's algorithm using $\prec$ to chose the next start vertex at each stage.

Step 6. Separate the sink into its original vertices. If the grove does not induce p on Λ discard the grove and restart the algorithm. When we have a grove with partition p the algorithm terminates.

By Lemma 2.3 at the end of step 5 the algorithm will generate a spanning tree of G_s distributed according to UST_{G_s}.

As G_s was constructed in such a way to retain knowledge of the corresponding edges in the original graph we can separate the sink in the spanning tree of G_s into its original components. The edges would remain unchanged except for each edge connecting to s which would be replaced with an edge connecting to one of $v_1, v_2, v_3 \dots$ accordingly. If we did this the spanning tree would become a spanning forest with each element of Θ being in a different component.

When the algorithm finishes it will have produced a grove with partition p. Moreover as we are simply discarding elements from a uniform distribution that do not satisfy a given property this algorithm will generate groves uniformly amongst groves which induce a partition p on Λ as desired.

Proof of Lemma 2.7. This property is stated in [31] for $n = 2$ and follows from Wilson's algorithm.

Let us consider the above algorithm in terms of the backbone of a graph. In this context we can see that Step 3 is generating the components $\mathfrak{b}_2, \mathfrak{b}_3, \dots$ chronologically. Whilst Step 4 generates the component $\mathfrak{b}_1$.

With this observation it is clear that if we consider the construction of a grove that was not discarded and look at the state of the construction at the end of Step 3, it will have the same distribution as taking the backbone of a uniformly chosen grove with partition p and discarding $\mathfrak{b}_1$.

Now running step 4 and asking for a path that does not cause the grove to be discarded is exactly the process of resampling $\mathfrak{b}_1$. Therefore if we replaced step 4 with that of the resampling process the final distribution would be unchanged. Hence we can conclude that $\mathfrak{b}'_1 \cup \mathfrak{b}_2 \cup \mathfrak{b}_3 \cup \dots$ and $\mathfrak{b}_1 \cup \mathfrak{b}_2 \cup \dots$ have the same distribution. □

2.3 Abelian sandpile model.

We know formally define the Abelian sandpile model.

Let $G = (V \cup \{s\}, \mathsf{E})$ be a finite, connected multigraph, with the wired boundary condition. A *sandpile configuration*, that we usually denote by η, consists of assigning an integer number of particles $\eta(v) \in \{0, 1, 2, \dots\}$ to every non-sink vertex $v \in V$. The sandpile η is *stable*, if $\eta(v) \in \{0, 1, \dots, \deg_G(v) - 1\}$, for all $v \in V$, where $\deg_G(v)$ is the degree of the vertex v in the graph G.

The dynamics of the model consist of two ingredients. The first is called *toppling*. This occurs when a vertex has at least as many particles as its degree. For such a vertex v, its height is reduced by its degree and one particle is sent along every edge incident with v with the neighbouring vertex's height increasing accordingly (i.e. vertices with multiple edges connecting them to v receive more than one particle). Particles reaching the sink are lost (i.e. we do not keep track of them). The toppling of v is summarised

by the mapping $S_v : \eta(w) \to \eta(w) - \Delta_{v,w}$, $w \in V$, using the Laplacian matrix

$$\Delta_{v,w} = \begin{cases} \deg_G(w) & \text{if } w = v; \\ -a_{v,w} & \text{if } w \neq v; \end{cases}$$

where $a_{v,w}$ is the number of edges connecting vertices $v, w \in V$.

Lemma 2.8. *Toppling all unstable vertices in a sandpile configuration will yield a unique stable configuration independent of the order of topplings.*

Proof. This was first shown in [11] and is the reason for the occurrence of Abelian in the model's name. Firstly note that as particles are lost to the sink the process of toppling all unstable vertex must terminate in a finite number of steps and thus it reaches a stable configuration. Next observe that if we had two vertices, v, w, that were unstable it is clear that $S_v \circ S_w = S_w \circ S_v$ as both can be represented by the map $\eta(x) \to \eta(x) - \deg_G(v)\delta_{x,v} - \deg_G(w)\delta_{x,w} + a_{v,x} + a_{w,x}$ for $x \in V$, where $\delta_{y,z}$ is 1 if $y = z$ and 0 otherwise.

Using this observation we need to show that the set of vertices that are toppled and the number of times each vertex is toppled does not change by altering the order that we choose to topple vertices.

Suppose we have a sequence of vertices $X = x_1, \ldots, x_N$ that stabilise the sandpile η when toppled. We now proceed to induct upon N, with the basis case of $N = 2$ having been shown above. We need to show any other order of vertices that can be toppled define a map that is equivalent to $S_{x_N} \circ S_{x_{N-1}} \circ \cdots \circ S_{x_1}$.

Suppose $Y = y_1, \ldots y_M$ is another sequence of vertices that could be toppled. Now from X we know that x_1 is unstable in η and therefore x_1 must appear in $y_1, \ldots y_M$ at least once, because toppling other vertices can not reduce the height at x_1 which needs to happen before the sandpile is stable. Let $k := \min\{i \in \mathbb{N} : y_i = x_1\}$.

Now consider the map given by Y,

$$S_{y_m} \circ \cdots \circ S_{y_1} = S_{y_m} \circ \cdots \circ S_{y_{k+1}} \circ S_{x_1} \circ S_{y_{k-1}} \circ \cdots \circ S_{y_1}.$$

However we know that x_1 is unstable in η thus it could be toppled before $y_1, \ldots y_{k-1}$, which are all distinct from x_1 by choice of k. Therefore by repeated application of being able to change the order of two distinct vertices toppling, we can deduce that the same sandpile is achieved by the mapping $S_{y_m} \circ \cdots \circ S_{y_{k+1}} \circ S_{y_{k-1}} \circ \cdots \circ S_{y_1} \circ S_{x_1}$. Now comparing this to the map $S_{x_N} \circ S_{x_{N-1}} \cdots \circ S_{x_1}$ we see that they now have the same first step and then we have a sequence of $N-1$ and $M-1$ topplings respectively to perform.

Repeating this for each x_i in turn shows that $M \geq N$ and will eventually produce the map $S_{y'_{M-N}} \circ \cdots \circ S_{y'_1} \circ S_{x_N} \circ S_{x_{N-1}} \cdots \circ S_{x_1}$, where $y'_1, \ldots y'_{M-N}$ are the remaining cycles that have not been matched. However once we have performed the first N steps we know that the configuration is stable so no more topplings can be performed,

because the sequence $x_1, \dots x_N$ has been toppled. Thus we can conclude that $M = N$ and so any sequence of vertices that stabilise a sandpile result in the same configuration as claimed. □

We use the notation η° to denote the unique configuration that η stabilises to via toppling all unstable vertices.

The second ingredient of the model is *particle additions*. Given a stable sandpile η, we add a particle at a randomly chosen vertex $v \in V$, and then stabilize via topplings, if necessary. Successive particle additions yield a Markov chain on the set of stable sandpiles.

We can then define a set of recurrent stable configurations for this Markov chain. It is the properties of these recurrent configurations that has been an important area of research and is the subject of this thesis.

We denote the set of recurrent states of this Markov chain by $\mathcal{R}_G$, and by ν_G the unique stationary distribution, that is the uniform distribution on $\mathcal{R}_G$ [11].

The quotient $\mathbb{Z}^V/\mathbb{Z}^V\Delta_G$ defines a set of equivalence classes with respect to the graph Laplacian, where elements in the same class differ by integer linear combinations of rows of Δ_G.

Lemma 2.9. *Each equivalence class contains exactly one element of $\mathcal{R}_G$. In particular this means that $|\mathcal{R}_G| = \det(\Delta_G)$.*

Before we prove this, following the method in [24], let us draw the readers attention to the following two important sandpile configurations.

Lemma 2.10. *(i) Define the sandpile configuration $\eta^{\max}$ by $\eta^{\max}(v) = \deg(v) - 1$ $\forall v \in V$, then $\eta^{\max}$ is stable and recurrent.*
(ii) Let $\delta(v) = \deg(v)$, then the configuration $\zeta := \delta - \delta^\circ$ is everywhere positive and it is an integer sum of rows of Δ_G. Also for any stable recurrent configuration η we have that $(\eta + \zeta)^\circ = \eta$.

Proof. (i) $\eta^{\max}$ satisfies the definition of a stable sandpile. In $\eta^{\max}$ each vertex has the maximum number of particles possible in a stable configuration, thus given any other recurrent stable configuration it is clear that using particle additions $\eta^{\max}$ can be reached and is therefore itself recurrent.

(ii) As every vertex is unstable in δ each vertex must lose at least one particle to reach δ° so the difference between the configurations will be greater than 1 at each vertex. As observed above, toppling an unstable vertex corresponds to subtracting the corresponding row of Δ_G from the configuration. Thus for some constants $c_i \geq 0$ we can write $\delta^\circ = \delta - \sum_i c_i\Delta_i$ where Δ_i is the i^{th} row of Δ_G.

To see the final statement consider the configuration $\delta + \zeta + \epsilon$, where $\epsilon(v) > 0$ for every $v \in V$ and it is chosen such that $\delta + \epsilon$ stabilises to η, such a configuration exists by definition of η being recurrent.

As $\zeta(v) > 0 \ \forall v \in V$ we can begin by toppling the vertices that are needed to stabilise $\delta + \epsilon$. This would produce the configuration $\eta + \zeta$ which can then be further stabilised to yield some recurrent configuration, $(\eta + \zeta)^\circ$.

Alternatively we could begin by toppling vertices involved in stabilising δ, this is allowed as $(\zeta + \epsilon)(v) > 0 \ \forall v \in V$, this then produces the configuration $\delta^\circ + \zeta + \epsilon = \delta + \epsilon$ but this can then be further stabilised using that $(\delta + \epsilon)^\circ = \eta$.

By Lemma 2.8 the configuration $\delta + \zeta + \epsilon$ has a unique stabilisation and so $(\eta + \zeta)^\circ = \eta$. □

Proof of Lemma 2.9. Observe that for any sandpile configuration by adding $n\zeta(v) > 0$ $\forall v \in V$ to the configuration, for n sufficiently large, we can obtain an equivalent configuration whose vertices all have heights greater than their degree. This configuration would then stabilise to a recurrent configuration, because it can be reached from $\eta^{\max}$ via particle additions and the fact that every configuration can be stabilised. Hence every equivalence class has at least one stable recurrent sandpile configuration.

It remains to show that any two recurrent stable configurations that are equivalent must in fact be equal.

If we have two recurrent stable sandpile configurations, η_1, η_2 in the same equivalence class we can find constants $c_i, d_i \geq 0$ such that $\eta_3 := \eta_1 + \sum_i c_i \Delta_i = \eta_2 + \sum_i d_i \Delta_i$.

Take $M = \max\{\max_{i \in \mathbb{N}} c_i, \max_{j \in \mathbb{N}} d_j\}$ and consider the configuration $\eta_3 + M\zeta$. By choice of M this will be a positive configuration and as it can be reached from η_1, equivalently η_2, via a sequence of particle additions it must be recurrent.

By first performing the topplings corresponding to subtracting $\sum_i c_i \Delta_i$ from the configuration, this is a valid selection of topplings as M is chosen large enough such that even after performing these subtractions all vertices will have a non-negative height. This would yield the configuration $\eta_1 + M\zeta$ which can be further stabilised to η_1 by Lemma 2.10.

Alternatively from $\eta_3 + M\zeta$ we could start by performing the topplings corresponding to subtracting $\sum_i d_i \Delta_i$, which is allowed by choice of M. This would leave $\eta_2 + M\zeta$ this can then be stabilised to η_2.

Finally due to Lemma 2.8 there is a unique stabilisation of $\eta_3 + M\zeta$ and so $\eta_1 = \eta_2$. It therefore follows that each equivalence class contains exactly one stable recurrent configuration and hence $|\mathcal{R}_G| = \det(\Delta_G)$. □

By the matrix tree theorem, see [37], it is also known that the number of spanning trees on a graph G is $\det(\Delta_G)$. It is this observation which first suggested that a relationship between recurrent sandpile configurations and spanning trees existed.

2.3.1 Burning bijection.

We now introduce a fundamental tool for investigating sandpile configurations which is the burning algorithm of Dhar [11].

The following combinatorial characterization of $\mathcal{R}_G$ follows from [11, 40] (see also [18]):

$$\mathcal{R}_G = \left\{ \eta \in \prod_{x \in V} \{0, \dots, \deg_G(x) - 1\} : \eta \text{ is ample for all } \emptyset \neq F \subset V \right\}.$$

Here η is called *ample for* F, if there exists $x \in F$ such that $\eta(x) \geq \deg_F(x)$.

Given $\eta \in \mathcal{R}_G$, at time 0 we declare the sink to be "burnt". Following this, we successively "burn" vertices where $\eta(x)$ is at least as much as the number of edges leading from x to any unburnt neighbours. More precisely, we set

$$B_0 := \{s\}, \qquad U_0 := V,$$

and for $j \geq 1$ we inductively set:

$$B_j := \left\{ v \in U_{j-1} : \eta(v) \geq \deg_{U_{j-1}}(v) \right\}, \qquad U_j := U_{j-1} \setminus B_j.$$

Here B_j (resp. U_j), are the sets of vertices burnt, (resp. unburnt), at time j. Since η is ample for any non-empty U_{j-1}, we have $U_j = \emptyset$ eventually, at which time the algorithm terminates. We say $v \in B_j$ has *burning time* j.

Majumdar and Dhar [40], following the above burning algorithm, constructed a bijection $\varphi_G : \mathcal{R}_G \to \mathcal{T}_G$, where $\mathcal{T}_G$ is the set of spanning trees of G. The map $\varphi_G : \eta \mapsto t$ can be defined as follows. Fix for each $v \in V$ an ordering $\prec_v$ of the oriented edges $\{f_i : \text{tail}(f_i) = v\}$.

If $v \in B_j$, let

$$\begin{aligned} m_v &:= \left| \left\{ f : \text{tail}(f) = v, \text{head}(f) \in \bigcup_{j' < j} B_{j'} \right\} \right|, \\ F_v &:= \{ e : \text{tail}(e) = v, \text{head}(e) \in B_{j-1} \}. \end{aligned} \tag{2.2}$$

We can use the ordering to enumerate F_v, so we have $e_0^{(v)} \prec_v e_1^{(v)} \prec_v \cdots \prec_v e_{\deg(v)-1}^{(v)}$. Due to the burning rule, we have

$$\eta(v) = \deg(v) - m_v + \ell_v \quad \text{ for some } 0 \leq \ell_v < |F_v|. \tag{2.3}$$

With ℓ_v as above, we then place the directed edge $e_{\ell_v}^{(v)}$ in t, for each $v \in V$, and forget the orientation of the edges. Observe that the burning time of a vertex $v \in V$ equals $\text{dist}_t(v, s)$.

The image of ν_G under φ_G is the uniform spanning tree measure UST_G, i.e. the uniform distribution on $\mathcal{T}_G$.

This procedure can be reversed in order to find the inverse mapping, which we now briefly describe.

Given a spanning tree $\mathfrak{T}$ and a set of vertices W we will now describe how to compute the sandpile configuration on W. To find the height of a vertex v we need to know the distance from v and its neighbours to the sink. We do this by assigning every vertex a burning time. Set $B_0 := \{s\}$, and then for $i \in \mathbb{N}$ set $B_i := \{v : \mathrm{dist}_t(v, s) = i\}$. Note that these sets correspond to the same sets generated by the burning algorithm. Hence for each vertex we know m_v and F_v and so we can use the burning rule $\eta(v) = \deg(v) - m_v + \ell_v$ to compute the height at each vertex v.

If we can find the earliest common ancestor of a vertex and its neighbours then we can use the above argument but instead of asking for the distance to s we only require the distances to the e.c.a. This is a useful observation that simplifies the algorithm when we want to compute the sandpile configuration on a finite subgraph of certain infinite graphs.

The burning bijection has been very fruitful in proving things about the sandpile model; see e.g. [46, 29, 2, 26], it was also the starting point for much of the work in this thesis.

See the surveys [12, 47, 22] for further background on the sandpile model.

CHAPTER 3

ANCHORED BURNING BIJECTION.

3.1 Introduction

Having introduced the map between sandpile configurations and spanning trees on finite graphs it is natural to look for an extension of the burning bijection to infinite graphs, and this leads to some highly non-trivial questions. The main difficulty in trying to do this is that on finite graphs the burning algorithm starts from the sink, so the analogous process on infinite graphs should start from infinity. This chapter will be concerned with a particular way of overcoming this problem. However, as we outline below, some very natural questions remain open.

For the remainder of this chapter, let $G = (V, \mathsf{E})$ be a locally finite, connected, infinite graph that satisfies the one-end property. Given an exhaustion by finite subgraphs: $V_1 \subset V_2 \subset \cdots \subset V$, $\cup_{n=1}^{\infty} V_n = V$, let $G_n = (V_n \cup \{s\}, \mathsf{E}_n)$ denote the *wired graph* obtained by identifying the vertices in $V \setminus V_n$, that becomes the sink s, and removing loop-edges at s. Note that there is a natural identification between E_n and those edges in E that have an end vertex in V_n. Recall we denote by WSF the weak limit of the measures UST_{G_n} [37], called the *wired uniform spanning forest measure* on G.

We denote

$$\mathcal{T} := \left\{ \begin{matrix} \text{spanning subgraphs of } G \text{ such that all} \\ \text{components are infinite one-ended trees} \end{matrix} \right\}.$$

The counterpart of $\mathcal{T}$ for the sandpile model will be

$$\mathcal{R} := \left\{ \eta \in \prod_{x \in V} \{0, \ldots, \deg_G(x) - 1\} : \eta \text{ is ample for all finite } \emptyset \neq F \subset V \right\},$$

that we call the *recurrent configurations on* G.

Athreya and Járai [2] considered the case of $\mathbb{Z}^d$, $d \geq 2$, with $V_n = [-n, n]^d \cap \mathbb{Z}^d$,

and they showed that ν_{G_n} has a weak limit ν that concentrates on $\mathcal{R}$. When $2 \le d \le 4$ the argument is particularly transparent. It was shown by Pemantle [44] that when $2 \le d \le 4$, the measure WSF concentrates on the set

$$\mathcal{T}^{\mathrm{conn}} := \left\{\text{one-ended spanning trees of } \mathbb{Z}^d\right\} \subset \mathcal{T}.$$

In this case the limiting sandpile measure ν is exhibited as the image of WSF under a map $\psi : \mathcal{T}^{\mathrm{conn}} \to \mathcal{R}$. Here ψ is defined essentially by inverting the relationships (2.2)–(2.3), that can be made sense of in $\mathbb{Z}^d$ for $t \in \mathcal{T}^{\mathrm{conn}}$. Namely, fix $t \in \mathcal{T}^{\mathrm{conn}}$ and $v \in \mathbb{Z}^d$. Let v^* denote the unique vertex such that all infinite paths starting at a neighbour of v pass through v^*, and v^* is nearest to v with respect to dist_t (this is the earliest common ancestor of the neighbours of v, such a vertex exists because $t \in \mathcal{T}^{\mathrm{conn}}$). Orient all edges of t towards infinity (this makes sense, because t has one end). Let

$$\begin{aligned} m'_v &:= \left|\left\{f : \mathrm{tail}(f) = v,\, \mathrm{dist}_t(\mathrm{head}(f), v^*) < \mathrm{dist}_t(v, v^*)\right\}\right|, \\ F'_v &:= \left\{e : \mathrm{tail}(e) = v,\, \mathrm{dist}_t(\mathrm{head}(e), v^*) = \mathrm{dist}_t(v, v^*) - 1\right\}. \end{aligned}$$

Enumerate F'_v as $e_0 \prec_v \cdots \prec_v e_{|F'_v|-1}$, and let $0 \le \ell' < |F'_v|$ be the unique index such that $e_{\ell'} \in t$. Then we set

$$\psi(t)(v) := \eta(v) := 2d - m'_v + \ell', \quad v \in \mathbb{Z}^d.$$

It is not difficult to see that ψ is continuous on $\mathcal{T}^{\mathrm{conn}}$. Where our understanding of continuous is in the sense that if $t_1, t_2 \in \mathcal{T}^{\mathrm{conn}}$ then $\forall M \in \mathbb{N}\ \exists N \in \mathbb{N}$ such that if t_1 and t_2 agree on $[-n, n]^d$ for $n > N$ then $\psi(t_1)$ and $\psi(t_2)$ agree in $[-M, M]^d$. (In a certain sense, ψ is the limit of the inverse bijections $\psi_{G_n} := \varphi_{G_n}^{-1} : \mathcal{T}_{G_n} \to \mathcal{R}_{G_n}$.) Moreover, ψ is equivariant under translations of $\mathbb{Z}^d$, if the orderings $\{\prec_v : v \in \mathbb{Z}^d\}$ are chosen equivariant. It is tempting to conjecture that ψ is almost one-to-one, i.e. injective up to sets of measure 0. We do not have a proof of this.

Open Question 1. Is ψ almost one-to-one in the case of $\mathbb{Z}^d$, $2 \le d \le 4$?

When $d > 4$, it turned out to be necessary to add extra randomness to the WSF in order to construct ν [2], so there is no natural mapping $\mathcal{T} \to \mathcal{R}$, a priori.

The main result of this chapter is the construction of a family of measure preserving mappings between spanning forests and sandpiles that are almost one-to-one. Our mappings can be constructed on general infinite graphs G satisfying condition (2.1), in particular, also on some non-transitive graphs. In this general setting, Járai and Werning [28] showed that ν_{G_n} converges weakly to a limit ν, that is independent of the exhaustion. Our construction is a natural extension of the one in [28], that in turn was based on an observation of Majumdar and Dhar [41] and Priezzhev [46]. In general, when $G = (V, \mathsf{E})$ is transitive, our mappings will *not* be invariant under all

graph automorphism.

Definition 3.1. *An* anchor *is a sequence $\mathcal{D} = \{D_1, D_2, \ldots\}$ of finite subsets of vertices such that*

(i) $D_1 \subset D_2 \subset \ldots$ *and* $\cup_{k\geq 1} D_k = V$*;*

(ii) D_k *is simply connected for each* $k \geq 1$*, i.e. all connected components of* $V \setminus D_k$ *are infinite.*

In Section 3.2 we will associate to any anchor $\mathcal{D}$ a particular burning rule. That is, for any finite $\Lambda \subset V$ and configuration $\eta \in \mathcal{R}_\Lambda$ we define burning times $\tau^{\mathcal{D}}(x, \Lambda; \eta)$, $x \in \Lambda$ in such a way that at each time only vertices that are burnable in the sense of Dhar [11] are burnt, (but it may be that some burnable vertices are not burnt at the first opportunity). The advantage of our rule will be that it is easy to pass to the limit $\Lambda \uparrow V$, i.e. we can define a consistent set of burning times $\tau^{\mathcal{D}}(x; \eta) \in (\mathbb{Z}, \mathbb{Z})$ for ν-a.e. $\eta \in \mathcal{R}$. The reason for requiring (ii) in Definition 3.1 is that for general D_k, our burning rule will be identical if we replace D_k by the smallest simply connected set containing it.

Theorem 3.2. *Assume that the infinite graph G satisfies condition* (2.1)*. The burning rule arising from any anchor $\mathcal{D}$ defines a continuous, measure preserving, injective map $\psi_{\mathcal{D}}$ from $(\mathcal{T}, \mathsf{WSF})$ to $(\mathcal{R}, \nu)$.*

The precise meaning of "defines" will become clear in Section 3.2, where we introduce the anchored bijection and the map $\psi_{\mathcal{D}}$. Indeed, the anchor will serve to prescribe a "preferred direction" for the burning of configurations on V starting from infinity. The following question complements Open Question 1.

Open Question 2. For $\mathbb{Z}^d$, $d > 4$, is there a continuous measure preserving map from $(\mathcal{T}, \mathsf{WSF})$ to $(\mathcal{R}, \nu)$ that is equivariant with respect to translations?

Open Questions 1 and 2 are connected to a result of Schmidt and Verbitskiy [48]. They constructed, for any $d \geq 2$, a family of $\mathbb{Z}^d$-equivariant continuous surjective mappings from $\mathcal{R}$ onto the so called harmonic model, i.e. functions from $\mathbb{Z}^d$ to the unit circle that are harmonic modulo 1. The image of ν under their maps is the unique measure of maximum entropy of the harmonic model [48, Theorem 5.9].

As an application of the anchored bijection, we show that combined with Wilson's stacks of arrows construction [50] it yields a coupling between ν_{G_n} and ν that we can analyse on $\mathbb{Z}^d$, $d \geq 2$. This leads to a power law upper bound on the rate of convergence of ν_{G_n} to ν.

Theorem 3.3. *Let $d \geq 2$ and $k \in \mathbb{N}$. Take E to be any cylinder event that depends only on the heights within distance k of the origin. Then fix a sufficiently large, but finite, $\Lambda \subset \mathbb{Z}^d$. Let N be the radius of the largest ball centred at the origin that is contained in Λ. There exists $\alpha = \alpha(d) > 0$ such that we have*

$$|\nu_\Lambda(E) - \nu(E)| \leq C(k, d) N^{-\alpha}. \tag{3.1}$$

The exponent α and the dependence on k are explicit, although not optimal; see Theorem 3.12 and Theorem 3.20 for more detailed statements. Estimates analogous to (3.1), but restricted to $d = 2, 3$, have been given in the context of the zero dissipation limit in the abelian avalanche model [21, 27]. We believe that our approach will lead to a significant simplification, and an extension to all $d \geq 2$, of the arguments of [21].

As mentioned earlier, we will define burning processes on both finite and infinite configurations in such a way that these behave well with respect to taking limits. In particular, restricting an infinite recurrent configuration to distinct large finite sets Λ_1, Λ_2, the anchored burning processes on Λ_1 and Λ_2 couple with high probability, in the following sense:

$$\lim_{\Lambda \uparrow V} \left[\tau^{\mathcal{D}}(x, \Lambda; \eta) - \tau^{\mathcal{D}}(y, \Lambda; \eta)\right] = \Big(c_1(x, y; \eta), c_2(x, y; \eta)\Big). \tag{3.2}$$

This property will be proven in Lemma 3.11. We do not know whether the same statement is true for Dhar's original burning algorithm, where at each step *every* burnable vertex is burnt simultaneously.

Open Question 3. Let $\tau(x, k; \eta)$ denote the burning time of x with respect to Dhar's original burning algorithm in the ball of radius k centred at the origin in $\mathbb{Z}^d$. Does the analogue of (3.2) hold for $\mathbb{Z}^d$, $2 \leq d \leq 4$, as $k \to \infty$?

If the answer is yes, this would imply an affirmative answer to Open Question 1. This is because the coupling defines a burning time from infinity (unique up to a time shift) and this can be used to define the inverse map. Note that the arguments of [2] show that the statement of Open Question 3 fails for $\mathbb{Z}^d$, $d > 4$.

We close this introduction by remarking that a certain analogue of the statement of Open Question 3 holds on graphs of the form $G = G_0 \times \mathbb{Z}$, with G_0 a finite connected graph. Indeed, with respect to the left-burnable measure studied by Járai and Lyons [25], it is not difficult to construct a sandpile configuration on the subgraph $G_0 \times \{i, i+1, i+2\}$ for some $i \in \mathbb{Z}$, that forces the burning times of vertices in $G_0 \times [i+3, \infty]$ to be independent of the burning times of vertices in $G_0 \times [-\infty, i-1]$, and hence coupling occurs. It was in fact by studying this case that we arrived at the idea of anchored bijections.

The rest of this chapter has the following structure. In Section 3.2 we define the anchored bijection in the finite case and then show how this extends to give a bijection in the infinite case. In Section 3.3 we present the quantitative bounds on $\mathbb{Z}^d$ when $d \geq 3$. In Section 3.4 we give the bounds on $\mathbb{Z}^2$. Throughout the remainder of this chapter C_d will stand for an unspecified positive constant dependent only on d, whilst $C, C_1, C_2 > 0$ are unspecified constants that do not depend on d.

3.2 Anchored bijections

Recall that $G = (V, \mathsf{E})$ is a locally finite infinite graph satisfying (2.1); we allow parallel edges. Let $\mathcal{D} = \{D_1, D_2, \dots\}$ be an anchor, and let $D_0 := \emptyset$. Given any finite $\Lambda \subset V$, we form the wired graph $G_\Lambda = (\Lambda \cup \{s\}, \mathsf{E}_\Lambda)$, and denote

$$\begin{aligned}\mathcal{T}_\Lambda &= \text{collection of spanning trees in } G_\Lambda,\\ \mathcal{R}_\Lambda &= \text{recurrent sandpiles in } G_\Lambda.\end{aligned}$$

We first define a bijection between $\mathcal{R}_\Lambda$ and $\mathcal{T}_\Lambda$ that is an extension of the one considered in [28].

Anchored bijection in finite Λ.

Let $K = \max\{k \geq 0 : D_k \subset \Lambda\}$. Fix $\eta \in \mathcal{R}_\Lambda$. Our definitions will depend on $\mathcal{D}$, but we will not always indicate this in our notation.

Phase 1. We apply the usual burning algorithm to η with the restriction that we do not allow any vertex of D_K to burn. That is, we define

$$\begin{aligned}B_0^{(1)} &:= \{s\},\\ U_0^{(1)} &:= \Lambda,\end{aligned}$$

and for $j \geq 1$ we inductively set:

$$\begin{aligned}B_j^{(1)} &:= \left\{ v \in U_{j-1}^{(1)} \setminus D_K : \eta(v) \geq \deg_{U_{j-1}^{(1)}}(v) \right\},\\ U_j^{(1)} &:= U_{j-1}^{(1)} \setminus B_j^{(1)}.\end{aligned}$$

We have $B_j^{(1)} = \emptyset$ eventually. Note that there may be vertices in $\Lambda \setminus D_K$ that do not burn in Phase 1. These vertices, together with the vertices in D_K, will burn in later phases.

Assuming Phase $i-1$ has already been defined for some $2 \leq i \leq K+1$, we inductively define Phase i as follows.

Phase i**.** We continue the burning algorithm on η with the restriction that no vertex of D_{K-i+1} is allowed to burn. That is, we set

$$\begin{aligned}B_0^{(i)} &:= \cup_{j \geq 0} B_j^{(i-1)},\\ U_0^{(i)} &:= \Lambda \setminus B_0^{(i)},\end{aligned}$$

and for $j \geq 1$ we inductively set:

$$\begin{aligned}B_j^{(i)} &:= \left\{ v \in U_{j-1}^{(i)} \setminus D_{K-i+1} : \eta(v) \geq \deg_{U_{j-1}^{(i)}}(v) \right\},\\ U_j^{(i)} &:= U_{j-1}^{(i)} \setminus B_j^{(i)}.\end{aligned}$$

We have $B_j^{(i)} = \emptyset$ eventually. Note that if $i \leq K$, there may be vertices in $\Lambda \setminus D_{K-i+1}$ that do not burn in Phase i, only later.

Since η is recurrent, all vertices that did not burn in Phases $1, \ldots, K$, do burn in Phase $K+1$ (if this was not true, we would have found a subset that is not ample for η). Hence we have $\cup_{j \geq 0} B_j^{(K+1)} = \Lambda \cup \{s\}$

We now define a map $\varphi_{\mathcal{D},\Lambda} : \mathcal{R}_\Lambda \to \mathcal{T}_\Lambda$. Regard G_Λ as an oriented graph, with each edge being present with both possible orientations. We fix for each $v \in \Lambda$ a linear ordering $\prec_v$ of the oriented edges e such that $\mathrm{tail}(e) = v$. Given the burning of η as above, we define what oriented edges will be present in the tree $t = \varphi_{\mathcal{D},\Lambda}(\eta)$.

If $v \in B_j^{(i)}$ for some $1 \leq i \leq K+1$ and $j \geq 1$, then we place an oriented edge pointing from v to some $w \in B_{j-1}^{(i)}$. In the case $j = 1$ such an edge exists, because v must have a neighbour outside $U_0^{(i)}$, and hence in $B_0^{(i)}$. In the case of $j \geq 2$ such an edge also exists, because the requirement to burn v at step j implies that the degree of v in $U_{j-1}^{(i)}$ is strictly smaller than its degree in $U_{j-2}^{(i)}$. Hence v has a neighbour in $B_{j-1}^{(i)} = U_{j-2}^{(i)} \setminus U_{j-1}^{(i)}$. If there is more than one $w \in B_{j-1}^{(i)}$ neighbouring v, we make the choice of the edge dependent on $\eta(v)$, similarly to the usual burning bijection. Formally, we let:

$$m_v := \left| \left\{ f : \mathrm{tail}(f) = v,\, \mathrm{head}(f) \in \bigcup_{j'<j} B_{j'}^{(i)} \right\} \right|,$$
$$F_v := \left\{ e : \mathrm{tail}(e) = v,\, \mathrm{head}(e) \in B_{j-1}^{(i)} \right\}.$$

Due to the burning rule, we have

$$\eta(v) = \deg(v) - m_v + \ell \quad \text{for some } 0 \leq \ell < |F_v|.$$

With ℓ as above, let $e_v \in F_v$ be that edge e such that $|\{f \in F_v : f \prec_v e\}| = \ell$. Then we place the directed edge e_v in t.

Once an edge has been included for each vertex in Λ the collection of edges that give t is complete and so $\varphi_{\mathcal{D},\Lambda}(\eta)$ is defined.

Lemma 3.4. *For any $\eta \in \mathcal{R}_\Lambda$ the collection of edges t (disregarding their orientations) is a spanning tree of G_Λ, and the map $\varphi_{\mathcal{D},\Lambda} : \eta \mapsto t$ is injective. Consequently, $\varphi_{\mathcal{D},\Lambda}$ is a bijection between $\mathcal{R}_\Lambda$ and $\mathcal{T}_\Lambda$.*

Proof. It is clear from the definitions that there are no cycles in t, since the sets $B_j^{(i)}$, are disjoint and "lexicographically ordered" by the indices (i,j) for $1 \leq i \leq K+1$, $j \geq 1$. In order to show injectivity, suppose that $\eta_1 \neq \eta_2$. There is a first time (i,j) in the burning processes of η_1 and η_2, where the "two processes differ". That is, there exists a lexicographically smallest (i,j) such that $B_{j'}^{(i')}(\eta_1) = B_{j'}^{(i')}(\eta_2)$ for all $i' < i$,

$j' \geq 1$ and for all $i' = i$, $j' < j$, and $\eta_1(v) = \eta_2(v)$ for all elements v of these sets, but

$$\text{there exists } v \in B_j^{(i)}(\eta_1) \cup B_j^{(i)}(\eta_2) \text{ such that } \eta_1(v) \neq \eta_2(v).$$

It is easy to check that our definition of $\varphi_{\mathcal{D},\Lambda}$ assigns different oriented edges emanating from v for η_1 and η_2. Since all edges are oriented towards the sink, this implies that the two trees also differ as unoriented trees, proving injectivity. Since $\mathcal{R}_\Lambda$ and $\mathcal{T}_\Lambda$ have the same number of elements, namely $\det(\Delta)$ [11], it follows that $\varphi_{\mathcal{D},\Lambda}$ is a bijection. □

Given $\eta \in \mathcal{R}_\Lambda$, we define the *burning time* $\tau^{\mathcal{D}}(x, \Lambda; \eta)$ as the index of the pair (i, j) in the lexicographic order, where $B_j^{(i)} \ni x$, $1 \leq i \leq K+1$, $j \geq 1$ (we restrict to the non-empty $B_j^{(i)}$'s). Note that in general this differs from the graph distance of x from s in the tree $\varphi_{\mathcal{D},\Lambda}(\eta)$. This is because at Step 1 of Phase i, we may be connecting a vertex $v \in B_1^{(i)} \cap D_{K-i+2}$ to a vertex w that was not burnt in the last step of Phase $i-1$.

Given $D \subset \Lambda$ and a spanning tree t of G_Λ, we write $\mathrm{desc}_t(D)$ for the set of descendants of D in t, that is, the collection of vertices w such that the path in t from w to s has a vertex in D.

Lemma 3.5. *For any finite $\Lambda \subset V$, $1 \leq i \leq K+1$, and $\eta \in \mathcal{R}_\Lambda$, the set of vertices that did not burn by the end of Phase i are precisely the descendants of D_{K-i+1}. That is, we have $U_0^{(i+1)} = \mathrm{desc}_{\varphi_{\mathcal{D},\Lambda}(\eta)}(D_{K-i+1})$.*

Proof. Observe that all vertices in $B_1^{(i+1)}$ are in D_{K-i+1}, otherwise they could have been burnt in Phase i. Since the oriented edges assigned by the bijection respect the lexicographic order, and the orientation is towards the sink, this implies that all vertices burnt in Phases $i+1, \ldots, K+1$ are in $\mathrm{desc}_{\varphi_{\mathcal{D},\Lambda}(\eta)}(D_{K-i+1})$. On the other hand, if a vertex v was burnt in one of the Phases $1, 2, \ldots, i$, then all vertices on the oriented path from v to s were also burnt in one of these Phases, and hence $v \notin \mathrm{desc}_{\varphi_{\mathcal{D},\Lambda}(\eta)}(D_{K-i+1})$. This completes the proof. □

We next formulate a consistency property between the sandpile configurations on the sets $\mathrm{desc}_{\varphi_{\mathcal{D},\Lambda}(\eta)}(D_k)$, $k \geq 1$, that will help us to take the limit $\Lambda \uparrow V$.

Definition 3.6. *Given $k \geq 1$ and a finite simply connected set W with $D_k \subset W \subset V$ and $\partial D_k \cap V \neq \emptyset$, we define the graph $G^*_{W,k} = (W \cup \{s\}, \mathsf{E}^*_{W,k})$ as follows. It contains all the edges that W induces in the graph V, and for each edge $e \in \mathsf{E}$ that connects a vertex $u \in D_k$ with a vertex $v \in V \setminus W$, there is an edge in $\mathsf{E}^*_{W,k}$ between u and s. Note that there is a natural identification between $\mathsf{E}^*_{W,k}$ and a subset of E, and we will use this identification freely in what follows.*

Lemma 3.7. *(i) Suppose $D_k \subset W \subset V$ with W simply connected. There is a mapping $\psi_{W,k} : \mathcal{T}_{G^*_{W,k}} \to \mathcal{R}_{G^*_{W,k}}$ such that whenever $\Lambda \supset W$, $t \in \mathcal{T}_\Lambda$ and $W = \mathrm{desc}_t(D_k)$ holds,*

$$\text{the restriction of the sandpile } \varphi^{-1}_{\mathcal{D},\Lambda}(t) \text{ to } W \text{ equals } \psi_{W,k}(t_{W,k}),$$

where $t_{W,k}$ denotes the restriction of t to the edges in $\mathsf{E}^*_{W,k}$.
(ii) Suppose $D_{k'} \subset D_k \subset W$. Let $t \in \mathcal{T}_{G^*_{W,k}}$. If $W' = \mathrm{desc}_t(D_{k'})$, then

the restriction of $\psi_{W,k}(t)$ to W' is given by $\psi_{W',k'}(t_{W',k'})$.

Proof. (i) Firstly observe that for every W there exists a $t \in \mathcal{T}_\Lambda$ such that $W = \mathrm{desc}_t(D_k)$. Write $\eta = \varphi^{-1}_{\mathcal{D},\Lambda}(t)$. Due to Lemma 3.5, the statement $W = \mathrm{desc}_t(D_k)$ is equivalent to the statement that in the sandpile η, W is precisely the set of vertices that did not burn in Phase $K - k + 1$. It is easy to check using the burning rules that as η varies over all sandpiles with this property, the restriction η_W ranges over $\mathcal{R}_{G^*_{W,k}}$, and $t_{W,k}$ is a spanning tree of $G^*_{W,k}$. It follows from our definition of Phases $K-k+2, \ldots, K+1$ of the anchored bijection that $t_{W,k}$ is entirely determined by η_W, in a way independent of Λ. The map $\eta_W \mapsto t_{W,k}$ is injective, and since $|\mathcal{R}_{G^*_{W,k}}| = |\mathcal{T}_{G^*_{W,k}}|$ it is bijective. Hence $\psi_{W,k}$ can be defined as the inverse of this map.

(ii) This follows similarly to part (i), because if $\Lambda \supset W$ and η is as in part (i), then the restriction of η_W to W' is $\eta_{W'}$. □

We are now ready to extend the bijection to G.

Anchored bijection on G.

Observe that for every $t \in \mathcal{T}$ and $v \in V$ there is a unique infinite path in t starting at v. Hence for any finite $D \subset V$, we can define $\mathrm{desc}_t(D)$ as those vertices for which the infinite path starting at v has a vertex in D.

Given $t \in \mathcal{T}$, for every $k \geq 1$ let $W_k = \mathrm{desc}_t(D_k)$. Observe that due to the one-end property (2.1) of elements of $\mathcal{T}$, W_k is finite for all $k \geq 1$, $\mathsf{WSF}a.s.$ Denote by $t_{W_k,k}$ the restriction of t to the edges in $\mathsf{E}^*_{W_k,k}$. Due to Lemma 3.7(ii), the configurations $\psi_{W_k,k}(t_{W_k,k})$ consistently define a stable configuration η on V. This η will be an element of $\mathcal{R}$, because for any finite $F \subset V$ there exists $k \geq 1$ such that $D_k \supset F$, and $\psi_{W_k,k}(t_{W_k,k}) = \eta_{W_k}$ is ample for F. We denote the configuration obtained by $\psi_{\mathcal{D}}(t)$, so $\psi_{\mathcal{D}} : \mathcal{T} \to \mathcal{R}$.

Remark 3.8. *Whenever $\Lambda \supset W_k = \mathrm{desc}_t(D_k)$, we have the following property. If we start burning $\psi_{\mathcal{D}}(t)|_\Lambda$ with the restriction that no vertex of D_k is allowed to burn, then the set of vertices that cannot be burnt is exactly W_k. This follows by considering the burning process in some $W_{k'} \supset \Lambda$.*

Lemma 3.9. *The map $\psi_{\mathcal{D}}$ is injective and continuous.*

Proof. Suppose that $t_1, t_2 \in \mathcal{T}$ such that $\psi_{\mathcal{D}}(t_1) = \psi_{\mathcal{D}}(t_2)$. Let us denote $W^{(1)}_k = \mathrm{desc}_{t_1}(D_k)$ and $W^{(2)}_k = \mathrm{desc}_{t_2}(D_k)$, and let $\Lambda = W^{(1)}_k \cup W^{(2)}_k$. By Remark 3.8, if we start the burning process on $\psi_{\mathcal{D}}(t_1)|_\Lambda = \psi_{\mathcal{D}}(t_2)|_\Lambda$ in Λ (with the restriction that D_k is not allowed to burn), then the set of vertices that do not burn equals both $W^{(1)}_k$ and $W^{(2)}_k$. In particular, these sets are equal, that is, $W^{(1)}_k = W^{(2)}_k$. Denoting their common

value by W_k, we have

$$\psi_{W_k,k}(t_1|_{\mathsf{E}^*_{W_k,k}}) = \psi_{\mathcal{D}}(t_1)|_{W_k} = \psi_{\mathcal{D}}(t_2)|_{W_k} = \psi_{W_k,k}(t_2|_{\mathsf{E}^*_{W_k,k}}).$$

Hence t_1 equals t_2 on $\mathsf{E}^*_{W_k,k}$. Since k is arbitrary, it follows that $t_1 = t_2$, and therefore $\psi_{\mathcal{D}}$ is injective.

In order to see continuity, fix $t \in \mathcal{T}$, let $\eta = \psi_{\mathcal{D}}(t)$, and let $k \geq 1$ be fixed. Let $W_k = \mathrm{desc}_t(D_k)$. Suppose that $t' \in \mathcal{T}$ has the property that t' agrees with t on all edges in E that have an end vertex in W_k. Then it follows that $\mathrm{desc}_{t'}(D_k) = W_k$, and $t'_{W_k,k} = t_{W_k,k}$. Therefore

$$\psi_{\mathcal{D}}(t')|_{W_k} = \psi_{W_k,k}(t'_{W_k,k}) = \psi_{W_k,k}(t_{W_k,k}) = \psi_{\mathcal{D}}(t)|_{W_k}.$$

Since $k \geq 1$ is arbitrary, $W_k \supset D_k$ and $\cup_{k\geq 1} D_k = V$, this implies that for all $M > 0$ there exists Λ such that if t_1 equals t_2 on Λ then $\psi_c D(t_1)|_{D_M} = \psi_{\mathcal{D}}(t_2)|_{D_M}$. □

The following lemma follows directly from the proof of [28, Theorem 3]. We provide a sketch of the proof for the reader's convenience.

Lemma 3.10. *The image of* WSF *under* $\psi_{\mathcal{D}}$ *equals* $\nu = \lim_{\Lambda\uparrow V} \nu_\Lambda$.

Sketch of the proof. Let E be a cylinder event that only depends on the sandpile heights in D_k for some $k \geq 1$. For any $\Lambda \supset D_k$, let $W_{\Lambda,k}$ be the random set of vertices that are unburnt just before the phase in which we first allow vertices in D_k to burn, that is, $U_0^{(K-k+2)}$. Due to Lemma 3.5, $W_{\Lambda,k}$ also equals the set of descendants of D_k in $\psi^{-1}_{\mathcal{D},\Lambda}(\eta_\Lambda)$, where η_Λ is the sandpile configuration in Λ. Recall the auxiliary graph $G^*_{W,k}$ from Definition 3.6. Due to the proof of Lemma 3.7(i), for any fixed set $D_k \subset W \subset \Lambda$, the conditional distribution of η_W, given the event $\{W_{\Lambda,k} = W\}$ is given by $\nu_{G^*_{W,k}}$. Hence, conditioning on the value of $W_{\Lambda,k}$, we have:

$$\nu_\Lambda(E) = \sum_{D_k \subset W \subset \Lambda} \nu_\Lambda(W_{\Lambda,k} = W)\nu_{G^*_{W,k}}(\eta_W \in E). \tag{3.3}$$

Note that, in the notation of Lemma 3.7, we have

$$\begin{aligned}\nu_{G^*_{W,k}}(\eta_W \in E) &= \mathsf{UST}_{G^*_{W,k}}(t : \psi_{W,k}(t) \in E)\\ &= \mathsf{WSF}(t : \psi_{W,k}(t_{W,k}) \in E \,|\, \mathrm{desc}_t(D_k) = W)\\ &= \mathsf{WSF}(t : \psi_{\mathcal{D}}(t) \in E \,|\, \mathrm{desc}_t(D_k) = W).\end{aligned}$$

In particular, this probability does not depend on Λ. We also have

$$\lim_{\Lambda\uparrow V} \nu_\Lambda(W_{\Lambda,k} = W) = \lim_{\Lambda\uparrow V} \mathsf{UST}_\Lambda(t : \mathrm{desc}_t(D_k) = W) = \mathsf{WSF}(t : \mathrm{desc}_t(D_k) = W).$$

This is because for a fixed finite set W, the event $\mathrm{desc}_t(D_k) = W$ is spanning-tree-local: it only depends on the status of the edges in $\mathsf{E}^*_{W,k}$. Finally, note that due to the one-end

property (2.1) the family $\{W_{\Lambda,k} : \Lambda \supset D_k\}$ is tight, in the sense that

$$\lim_{M\to\infty} \sup_{\Lambda\supset D_M} \mathsf{UST}_\Lambda(t : \mathrm{desc}_t(D_k) \not\subset D_M) = 0.$$

This allows us to pass to the limit in (3.3) and obtain

$$\begin{aligned}\lim_{\Lambda\uparrow V} \nu_\Lambda(E) &= \nu(E)\\ &= \sum_{\substack{W: W \text{ is finite}\\ W\supset D_k}} \mathsf{WSF}(t : \mathrm{desc}_t(D_k) = W)\, \mathsf{WSF}(\psi_{\mathcal{D}}(t) \in E \,|\, \mathrm{desc}_t(D_k) = W)\\ &= \mathsf{WSF}(t : \psi_{\mathcal{D}}(t) \in E).\end{aligned}$$

□

Lemmas 3.9, 3.10 imply Theorem 3.2.

Our final lemma shows the coupling property (3.2).

Lemma 3.11. *Fix $o \in D_1$. For any $t \in \mathcal{T}$ and $x \in V$ the limit*

$$\lim_{\Lambda\uparrow V} \left[\tau^{\mathcal{D}}(x, \Lambda; \psi_{\mathcal{D}}(t)) - \tau^{\mathcal{D}}(o, \Lambda; \psi_{\mathcal{D}}(t))\right] =: \tau^{\mathcal{D}}(x; \eta) \in (\mathbb{Z}, \mathbb{Z})$$

exist.

Proof. Let $k \geq 1$ be the smallest index such that $x \in D_k$, let $W = \mathrm{desc}_t(D_k)$, and suppose that $\Lambda \supset W$. Due to Remark 3.8, for any such Λ the last $k+1$ phases of the burning of η_Λ have identical history. This implies the claim. □

3.3 Rate of convergence in $\mathbb{Z}^d$, $d \geq 3$.

Henceforth we consider the graphs $G = \mathbb{Z}^d$, and in this section we assume $d \geq 3$. Let D_k be the intersection of the Euclidean ball of radius k about the origin with $\mathbb{Z}^d$.

Let $\Lambda \subset \mathbb{Z}^d$ be finite. We consider the realizations of WSF and UST_{G_Λ} via stacks of arrows, as introduced in chapter 2. Recall the notation we defined there.

Define $\mathbb{P}$ as the underlying probability measure for the stacks of arrows. An oriented cycle $\mathcal{C}$ in $\mathbb{Z}^d$ is associated the *weight* $w(\mathcal{C}) = (2d)^{-|\mathcal{C}|}$, where $|\mathcal{C}|$ denotes the number of arrows in $\mathcal{C}$. Sometimes we will need to consider *coloured cycles*, that is, a cycle consisting of some arrows $e_{i_1}^{v_1}, \ldots, e_{i_r}^{v_r}$. We will again use bold characters, like $\mathbf{C}$, to denote coloured cycles. In this case, $\mathcal{C}$ will denote the cycle obtained from $\mathbf{C}$ by ignoring the colours.

Also recall that we described how Cycle popping can be made sense of in transient graphs. Here we will need the following alternative way of popping cycles in $\mathbb{Z}^d$:

first pop all cycles contained in D_1, then pop all cycles contained in D_2, etc. (3.4)

Wilson's proof for finite graphs [50], stated here as Lemma 2.3, can be adapted to show that on the probability 1 event when T is a well-defined sample from WSF, the procedure (3.4) reveals exactly the same forest T as the more standard algorithm that was presented in subsection 2.2. In particular, for any finite $\Lambda \subset \mathbb{Z}^d$, cycle popping in Λ also terminates with probability 1, resulting in a spanning tree T_Λ, distributed according to UST_{G_Λ}. Thus, using the same stacks of arrows for cycle popping in Λ and in $\mathbb{Z}^d$ provides the required coupling of WSF and UST_{G_Λ}.

Given a cylinder event $E \subset \{0, \dots, 2d-1\}^{D_k}$ only depending on sandpile heights in D_k, let us write $E_{\mathbb{Z}^d} = \{T : \psi_{\mathcal{D}}(T) \in E\}$ and $E_\Lambda = \{T_\Lambda : \psi_{\mathcal{D},\Lambda}(T_\Lambda) \in E\}$. We have $\mathbb{P}(E_\Lambda) = \nu_\Lambda(E)$, due to Lemma 3.4 and $\mathbb{P}(E_{\mathbb{Z}^d}) = \nu(E)$, due to Lemma 3.10.

Theorem 3.12. *Let E be a cylinder event depending only on the sandpile heights in D_k. Let $d \geq 3$, let $\Lambda \subset \mathbb{Z}^d$ be a finite set and let N be the radius of the largest ball centred at the origin that is contained in Λ. We have*

$$|\nu_\Lambda(E) - \nu(E)| \leq \mathbb{P}(E_\Lambda \Delta E_{\mathbb{Z}^d}) \leq \begin{cases} C_d k^{d-1} N^{\frac{2-d}{2d}} & \text{if } d \geq 5; \\ C k^{26/9} N^{-2/9} & \text{if } d = 4; \\ C k^{25/13} N^{-1/13} & \text{if } d = 3. \end{cases}$$

Here Δ denotes symmetric difference.

The proof is broken down into a number of propositions and lemmas. Let us write W_k for the random set of descendants of D_k in T,

Proposition 3.13. *Suppose $d \geq 3$, $1 \leq k < n < N$, and $\Lambda \supset D_N$. Let $\mathsf{E}^*_{W,k}$ be as defined in Definition 3.6, then there is a constant $C_d > 0$, which depends only on d, such that*

$$\mathbb{P}\left(W_k \subset D_n \text{ but } W_k \neq W_{k,\Lambda} \text{ or } T|_{\mathsf{E}^*_{W,k}} \neq T_\Lambda|_{\mathsf{E}^*_{W,k}}\right) \leq C_d \frac{k^{d-2} n^2}{(N-n)^{d-2}}. \tag{3.5}$$

Proof. If we successively pop all cycles in D_n, then in D_{n+1}, then in D_{n+2}, etc., then we see that $\mathbb{P}$-a.s. on the event $W_k \subset D_n$ we have $W_{k,\Lambda'} = W_k$ and $T|_{E^*_{W,k}} = T_\Lambda|_{E^*_{W,k}}$ for all large enough finite Λ'. Therefore, it is enough to show that for all finite $\Lambda' \supset \Lambda$ we have

$$\mathbb{P}\left(W_{k,\Lambda'} \subset D_n \text{ but } W_{k,\Lambda'} \neq W_{k,\Lambda} \text{ or } T_{\Lambda'}|_{\mathsf{E}^*_{W,k}} \neq T_\Lambda|_{\mathsf{E}^*_{W,k}}\right) \leq C_d \frac{k^{d-2} n^2}{(N-n)^{d-2}}, \tag{3.6}$$

with C_d independent of Λ, Λ'.

In order to prove (3.6), we first pop all cycles we can that are contained in Λ. This leaves on top of the stacks in Λ the wired spanning tree T_Λ of G_Λ. Let $\mathfrak{L}$ denote the collection of remaining coloured cycles contained in Λ' that need to be popped in order to obtain the wired spanning tree $T_{\Lambda'}$ in Λ'. For convenience, the cycles in $\mathfrak{L}$ are regarded as having colours according to their *current positions* in the stacks, i.e. after

all cycles contained in Λ have been popped. We claim that the probability distribution of $\mathfrak{L}$ is proportional to total weight and that $\mathfrak{L}$ is independent of the wired spanning tree $T_{\Lambda'}$ in Λ', that is:

$$\mathbb{P}(\mathfrak{L} = \{\mathbf{C}_1, \dots, \mathbf{C}_K\},\, T_{\Lambda'} = t_{\Lambda'}) = \mathsf{UST}_{G_{\Lambda'}}(t_{\Lambda'}) \frac{1}{Z} \prod_{j=1}^{K} w(\mathcal{C}_j), \tag{3.7}$$

where Z is a normalization factor. Indeed, we show that this follows from Wilson's theorem [50]. Let us write $\mathfrak{L}^0_\Lambda$, respectively $\mathfrak{L}^0_{\Lambda'}$, for the collection of coloured cycles contained in Λ, respectively Λ', that we need to pop in order to reveal T_Λ, respectively $T_{\Lambda'}$. Then $\mathfrak{L}$ is a deterministic function of $\mathfrak{L}^0_{\Lambda'}$ (recall that the colours of cycles in $\mathfrak{L}$ are according to their positions acquired after cycle popping in Λ is complete). By Wilson's theorem, $T_{\Lambda'}$ is independent of $\mathfrak{L}^0_{\Lambda'}$, and hence of $\mathfrak{L}$, and is distributed according to $\mathsf{UST}_{G_{\Lambda'}}$. Therefore, the left hand side of (3.7) equals

$$\mathsf{UST}_{G_{\Lambda'}}(t_{\Lambda'})\, \mathbb{P}(\mathfrak{L} = \{\mathbf{C}_1, \dots, \mathbf{C}_K\}).$$

In order to show that the second factor is proportional to weight, first observe that $\mathfrak{L}^0_{\Lambda'}$ and the pair $(\mathfrak{L}^0_\Lambda, \mathfrak{L})$ are deterministic functions of each other. We show that $\mathfrak{L}^0_\Lambda$ and $\mathfrak{L}$ are independent. This is because, using Wilson's Theorem again, $\mathfrak{L}^0_\Lambda$, T_Λ, the stacks of arrows beneath T_Λ, and the stacks of arrows in $\Lambda' \setminus \Lambda$ are mutually independent, and $\mathfrak{L}$ is a deterministic function of the latter three. We have

$$\mathbb{P}(\mathfrak{L}^0_\Lambda = \{\mathbf{C}^0_1, \dots, \mathbf{C}^0_{K^0}\},\, \mathfrak{L} = \{\mathbf{C}_1, \dots, \mathbf{C}_K\}) = \frac{1}{Z^0_{\Lambda'}} \times \prod_{\ell=1}^{K^0} w(\mathcal{C}^0_\ell) \times \prod_{j=1}^{K} w(\mathcal{C}_j).$$

Summing over all instances of $\mathfrak{L}^0_\Lambda$, the independence of $\mathfrak{L}^0_\Lambda$ and $\mathfrak{L}$ implies

$$\mathbb{P}(\mathfrak{L} = \{\mathbf{C}_1, \dots, \mathbf{C}_K\}) = \frac{1}{Z} \prod_{j=1}^{K} w(\mathcal{C}_j).$$

This proves the claim made in (3.7)

We introduce a partial order on elements of $\mathfrak{L}$ as follows: we say that $\mathbf{C} \prec \mathbf{C}'$, if there exist $j \geq 1$ and a sequence of coloured cycles $\mathbf{C} = \mathbf{C}_j, \mathbf{C}_{j-1}, \dots, \mathbf{C}_0 = \mathbf{C}'$ all in $\mathfrak{L}$, such that for each $1 \leq r \leq j$, the coloured cycles $\mathbf{C}_{r-1}$ and $\mathbf{C}_r$ share at least one vertex whose colour in $\mathbf{C}_r$ is one greater than its colour in $\mathbf{C}_{r-1}$. The meaning of the relation $\prec$ is the following:

$$\mathbf{C} \prec \mathbf{C}' \qquad \Longleftrightarrow \qquad \text{regardless of the order of popping, } \mathbf{C}' \text{ is popped before } \mathbf{C}. \tag{3.8}$$

(Recall that the set $\mathfrak{L}$ does not depend on the order of popping.) The direction $\Longrightarrow$ of this equivalence is immediate from the definition of $\prec$. To see the $\Longleftarrow$ direction, let us pop every cycle we can without popping $\mathbf{C}'$. This does not reveal $\mathbf{C}$. Now pop $\mathbf{C}'$, and

note that any cycle that is revealed as a result of popping $\mathbf{C}'$ necessarily shares a vertex with $\mathbf{C}'$. Popping further cycles it holds that any cycle that is revealed has a chain of cycles leading to $\mathbf{C}'$. In particular, $\mathbf{C}$ must have this property. The equivalence (3.8) makes it clear that $\prec$ is a partial order on $\mathfrak{L}$.

We apply a parallel popping procedure to reveal $\mathfrak{L}$, defined in stages. In each stage, we pop all cycles on top of the stacks, simultaneously. If the event on the left hand side of (3.6) occurs, we pop some cycle that intersects $\overline{W} := W_{k,\Lambda'} \cup \partial W_{k,\Lambda'}$. Indeed, if we never popped any such cycles, then the arrows attached to all the vertices in $\overline{W}$ would have the same direction as they had in T_Λ, which would force $W_{k,\Lambda'} = W_{k,\Lambda}$ and $T_{\Lambda'}|_{\mathsf{E}^*_{W,k}} = T_\Lambda|_{\mathsf{E}^*_{W,k}}$. Let us select, according to some fixed arbitrary rule, a cycle $\mathbf{D}_1 \in \mathfrak{L}$ such that $\mathbf{D}_1 \cap \overline{W} \neq \emptyset$ and a vertex $w \in \mathbf{D}_1 \cap \overline{W}$. Let

$$\mathfrak{M} := \{\mathbf{D} \in \mathfrak{L} : \mathbf{D} \succeq \mathbf{D}_1\}. \tag{3.9}$$

Observe that $\mathfrak{M}$ can be popped from $\mathfrak{L}$ (without popping any other cycles), since by construction, $\mathfrak{M}$ is closed under domination in the partial order $\prec$. Define $\widetilde{\mathfrak{L}}$ to be the collection of coloured cycles left after popping $\mathfrak{M}$ from $\mathfrak{L}$.

Lemma 3.14. *The map $\mathfrak{L} \mapsto (\mathfrak{M}, \widetilde{\mathfrak{L}})$ is injective.*

Proof. This immediately follows from the definition of the map. □

We are going to join the cycles in $\mathfrak{M}$ into a single loop γ in $\mathbb{Z}^d$, and then bound the probability of the possible arising loops in Lemma 3.16 below. Note that by the definition of $\mathfrak{M}$, the arrow at w on the top of its stack is included in a loop that we want to pop. There is also a unique edge in $\mathbf{D}_1$ directed towards w, label this edge as $e_{\mathbf{D}_1}$. We set $\gamma(0) = w$. We define γ by following the arrows, starting with the one on the top of the stack of w, and whenever we visit a vertex v for the i-th time, we use the i-th coloured arrow at v. The walk stops when it uses the edge $e_{\mathbf{D}_1}$. We call γ the *loop associated to* $\mathfrak{M}$. The purpose of the next lemma is to show that γ is well-defined and the map $\mathfrak{M} \to \gamma$ is injective.

Lemma 3.15. *Let $W \subset D_n$ be a fixed set and let $w \in \partial W$ be a fixed vertex. Suppose that $\mathfrak{L}$ is a collection of coloured cycles that can be popped, and $\mathbf{D}_1 \in \mathfrak{L}$ has the property that $w \in \mathcal{D}_1 \cap (W \cup \partial W)$, with $e_{\mathbf{D}_1}$ the unique edge in $\mathbf{D}_1$ directed towards w. Let $\mathfrak{M}$ be defined by formula* (3.9)*. Then we have:*
(i) The loop associated to $\mathfrak{M}$ is well-defined in that the walk does return to w.
(ii) Every coloured edge in $\mathfrak{M}$ is used exactly once by the loop.
(iii) The map $\mathfrak{M} \mapsto \gamma$ is injective.

Proof. (i), (ii) We prove the two statements together by induction on the number of cycles in $\mathfrak{M}$. If $\mathfrak{M}$ consists of the single cycle $\mathbf{D}_1$, the statements are trivial. Otherwise, consider the first time we return to a vertex v that we visited before. Then the cycle just found, $\mathbf{D}$, say, is necessarily on top of the stacks and $\mathbf{D} \neq \mathbf{D}_1$, in particular $e_{\mathbf{D}_1}$

has not been used. Also, since the walk starts with an arrow belonging to a cycle in $\mathfrak{M}$, it is easy to see that $\mathbf{D} \in \mathfrak{M}$. Now pop $\mathbf{D}$, and define $\mathfrak{L}'$, $\mathfrak{M}'$, $\mathbf{D}_1'$ by moving the arrows in the stacks of the vertices of $\mathcal{D}$ up by one (and removing the arrows in $\mathbf{D}$). Observe that $\mathfrak{L}'$, $\mathfrak{M}'$, $\mathbf{D}_1'$ also satisfy the hypotheses of the Lemma, so by the induction hypothesis, the walk γ' defined by $\mathfrak{M}'$ visits each arrow of $\mathfrak{M}'$ exactly once. Hence inserting into γ' the cycle $\mathcal{D}$ at v we get the walk γ defined by $\mathfrak{M}$. This implies the statements (i) and (ii).

(iii) This follows from the fact that by construction, following the history of the loop-erasure process on γ (started at w) the loops erased are precisely the loops in $\mathfrak{M}$. □

We continue with the proof of Proposition 3.13. We bound the left hand side of (3.6) from above as follows. Let Π denote the class of all sets of coloured loops that are possible values of $\mathfrak{L}$. Let Γ_w denote the collection of loops in $\mathbb{Z}^d$ that start and end at w and visit Λ^c. Let $\Gamma_{w,\Lambda'}$ denote those loops in Γ_w that stay inside Λ'. By the stated independence of the spanning tree in Λ' and $\mathfrak{L}$, we have

$$\begin{aligned}\mathbb{P}\left(W_{k,\Lambda'} \subset D_n \text{ and } W_{k,\Lambda'} \neq W_{k,\Lambda} \text{ or } T_{\Lambda'}|_{\mathsf{E}^*_{W,k}} \neq T_{\Lambda}|_{\mathsf{E}^*_{W,k}}\right)\\ \leq \sum_{W \subset D_n} \mu_{\Lambda'}(W_{k,\Lambda'} = W)\frac{1}{Z} \sum_{\substack{\mathfrak{L}\in\Pi:\exists \mathbf{D}_1\in\mathfrak{L},\\ \mathbf{D}_1\cap\partial W\neq\emptyset}} \prod_{\mathbf{C}\in\mathfrak{L}} w(\mathcal{C}).\end{aligned} \tag{3.10}$$

We fix W, and estimate the sum over $\mathfrak{L}$. To every $\mathfrak{L}$ occurring in the sum, we have associated (by our arbitrary rule), a choice of $w \in \partial W$ and $\mathfrak{M} \subset \mathfrak{L}$ containing w. This $\mathfrak{M}$, in turn determines a loop γ based at w. Observe that

$$\prod_{\mathbf{C}\in\mathfrak{L}} w(\mathcal{C}) = \prod_{\mathbf{D}\in\mathfrak{M}} w(\mathcal{D}) \times \prod_{\widetilde{\mathbf{C}}\in\widetilde{\mathfrak{L}}} w(\widetilde{\mathcal{C}}) = w(\gamma) \times \prod_{\widetilde{\mathbf{C}}\in\widetilde{\mathfrak{L}}} w(\widetilde{\mathcal{C}}).$$

Hence, using the injectivity statements in Lemma 3.14 and Lemma 3.15(iii), the last three terms in the right hand side of (3.10) is at most

$$\begin{aligned}\frac{1}{Z}\sum_{w\in\partial W}\sum_{\gamma\in\Gamma_{w,\Lambda'}} w(\gamma)\sum_{\widetilde{\mathfrak{L}}\in\Pi}\prod_{\widetilde{\mathbf{C}}\in\widetilde{\mathfrak{L}}} w(\widetilde{\mathcal{C}}) &\leq \sum_{w\in\partial W}\sum_{\gamma\in\Gamma_{w,\Lambda'}} w(\gamma)\\ &\leq \sum_{w\in\partial W}\sum_{\gamma\in\Gamma_w} w(\gamma).\end{aligned} \tag{3.11}$$

Lemma 3.16. *For any $w \in D_n$, we have*

$$\sum_{\gamma\in\Gamma_w} w(\gamma) \leq \frac{C_d}{(N-n)^{d-2}}. \tag{3.12}$$

Proof. The weight of a loop is equal to the probability of each step present occurring. Therefore the sum of the weights over loops Γ_w equals the sum of the probabilities of

random walk paths that start and end at w and exit Λ. Letting S_v denote a simple random walk started at time 0 at v and let ξ_N be the first exit time of D_N we get

$$\begin{aligned}
&\sum_{\gamma\in\Gamma_w} w(\gamma)\\
&= \sum_{m\geq 0}\sum_{z\in\partial D_N}\sum_{r>m} \mathbb{P}(\xi_N = m,\, S_w(m) = z)\mathbb{P}(S_w(r) = w,\, |\, \xi_N = m,\, S_w(m) = z)\\
&= \sum_{m\geq 0}\sum_{z\in\partial D_N} \mathbb{P}(\xi_N = m,\, S_w(m) = z)\sum_{r>0}\mathbb{P}(S_z(r) = w)\\
&= \sum_{m\geq 0}\sum_{z\in\partial D_N} \mathbb{P}(\xi_N = m,\, S_w(m) = z)G(z,w)\\
&\leq \frac{C_d}{(N-n)^{d-2}}\sum_{m\geq 0}\sum_{z\in\partial D_N} \mathbb{P}(\xi_N = m,\, S(m) = z)\\
&= \frac{C_d}{(N-n)^{d-2}}.
\end{aligned}$$

Here $G(z,w)$ is Green's function, see [35, Section 4.3] for a proof of the bound on $G(z,w)$. □

Inserting (3.12) and (3.11) into (3.10) we get

$$\begin{aligned}
&\mathbb{P}\left(W_{k,\Lambda'} \subset D_n \text{ and } W_{k,\Lambda'} \neq W_{k,\Lambda} \text{ or } T_{\Lambda'}|_{\mathsf{E}^*_{W,k}} \neq T_{\Lambda}|_{\mathsf{E}^*_{W,k}}\right)\\
&\quad\leq \sum_{W\subset D_n} \mu_{\Lambda'}(W_{k,\Lambda'} = W)\frac{1}{Z}\sum_{\substack{\mathfrak{L}\in\Pi:\exists \mathbf{D}_1\in\mathfrak{L},\\ \mathbf{D}_1\cap\partial W\neq\emptyset}}\prod_{\mathbf{C}\in\mathfrak{L}} w(\mathcal{C})\\
&\quad\leq \sum_{W\subset D_n} \mu_{\Lambda'}(W_{k,\Lambda'} = W)\sum_{w\in\partial W}\sum_{\gamma\in\Gamma_w} w(\gamma)\\
&\quad\leq \sum_{W\subset D_n} \mu_{\Lambda'}(W_{k,\Lambda'} = W)\sum_{w\in\partial W}\frac{C_d}{(N-n)^{d-2}} \qquad (3.13)\\
&\quad\leq \frac{C_d}{(N-n)^{d-2}}\sum_{W\subset D_n} \mu_{\Lambda'}(W_{k,\Lambda'} = W)|\partial W|\\
&\quad\leq \frac{C_d}{(N-n)^{d-2}}\mathbb{E}_{\mu_{\Lambda'}}\left[|\partial W_{k,\Lambda'}| : W_{k,\Lambda'}\subset D_n\right]\\
&\quad\leq \frac{C_d}{(N-n)^{d-2}}\mathbb{E}_{\mu_{\Lambda'}}\left[|W_{k,\Lambda'}| : W_{k,\Lambda'}\subset D_n\right].
\end{aligned}$$

We estimate the right hand side in the last equation in the following lemma.

Lemma 3.17. *We have*

$$\mathbb{E}_{\mu_{\Lambda'}}\left[|W_{k,\Lambda'}| : W_{k,\Lambda'}\subset D_n\right] \leq C_d k^{d-2}n^2.$$

Proof. By considering running Wilson's algorithm, as described in Section 2.2, with the first walk starting from x it follows that that the probability that a vertex $x \in D_n \setminus D_k$ is in $W_{k,\Lambda'}$ is at most the probability that a simple random walk started at x hits D_k.

This is bounded by $C_d k^{d-2}/|x|^{d-2}$. Summing over $x \in D_n$ gives

$$\begin{aligned}\mathbb{E}_{\mu_{\Lambda'}}\left[|W_{k,\Lambda'}| : W_{k,\Lambda'} \subset D_n\right] &\leq |D_k| + \mathbb{E}_{\mu_{\Lambda'}}\left[|W_{k,\Lambda'} \cap (D_n \setminus D_k)|\right] \\ &\leq C_d k^d + C_d n^2 k^{d-2} \\ &\leq C_d n^2 k^{d-2}.\end{aligned}$$

□

The above lemma and (3.13) completes the proof of Proposition 3.13. □

Proposition 3.18. *Suppose $d \geq 3$. Then for all $k > 0$ there exits a sufficiently large n such that*

$$\mathbb{P}(W_k \not\subset D_n) \leq C_d k^{d-1} n^{\frac{2-d}{2d}}.$$

We prove this proposition by extending the argument of [38, Theorem 4.1], that requires a couple of alterations.

Proof. Condition on the event that the restriction of the uniform spanning forest to D_k, denoted $T|_{D_k}$, is a fixed forest K. Let K_j, $j = 1, 2, \ldots$ denote the connected components of K. Then

$$\begin{aligned}\mathbb{P}(\mathrm{desc}(D_k) \not\subset D_n \,|\, T|_{D_k} = K) &= \mathbb{P}\big(\cup_j \left\{\mathrm{desc}(K_j) \not\subset D_n\right\} \,\big|\, T|_{D_k} = K\big) \\ &\leq \sum_j \mathbb{P}(\mathrm{desc}(K_j) \not\subset D_n \,|\, T|_{D_k} = K).\end{aligned}$$

In order to deal with the summand in the last expression, we need to generalize [38, Lemma 3.2]. Given a graph G, and V a subset of the vertices, we denote by G/V the graph obtained from G by identifying all the vertices in V to a single vertex and removing loop-edges.

Lemma 3.19. *Let G be a finite graph containing D_k as a subgraph and s a vertex of G with $s \notin D_k$. Let T_K denote the uniform spanning tree of G conditioned on its restriction to D_k being K. Let $L_j(T_K)$ denote the unique path from K_j to s in T_K. Then on the set of edges not belonging to K_j, the graph $T_K \setminus L_j(T_K)$ is stochastically dominated by the uniform spanning tree of $G/(K_j \cup \{s\})$, conditioned on the event that its restriction to D_k/K_j equals K/K_j.*

Proof. First we further condition on $L_j(T_K) = L$. Note that under this conditioning, $T_K \setminus L$ has the same distribution as the uniform spanning tree of $G/\mathrm{Vert}(L)$ given K, where $\mathrm{Vert}(\cdot)$ denotes the vertex set of a graph. By the negative association theorem of Feder and Mihail [14], [37, Chapter 4], conditioning on an edge being present makes the remaining set of edges stochastically smaller. As $\mathrm{Vert}(L)$ contains both K_j and s we can repeatedly apply this result to deduce that on the edges not belonging to $K_j \cup L$ the set of edges $T_K \setminus L$ is dominated by the uniform spanning tree of $G/(K_j \cup \{s\})$

given K/K_j. We can now average over all possible paths $L = L_j(T_K)$ to remove this part of the conditioning and get the stated lemma. □

We will use the following corollary of Lemma 3.19 that can be deduced by taking weak limits. Let $\mathfrak{F}_{K,j}$ denote the wired spanning forest conditioned on K with K_j wired to infinity (defined as the weak limit of uniform spanning trees conditioned on K with K_j wired to the sink).

> The set of descendants of K_j in the wired uniform spanning forest conditioned on K is stochastically dominated by the connected component of K_j in $\mathfrak{F}_{K,j}$.

The rest of the proof follows an outline similar to the proof of [38, Theorem 4.1]. We define edge sets $E_1 \subset E_2 \subset \dots$ as follows. Let $E_0 = K_j$. Assuming E_n has been defined, let S_n be the set of vertices of the connected component of $\mathfrak{F}_{K,j} \cap E_n$ containing K_j. If all edges incident with S_n are in E_n, we set $E_{n+1} = E_n$. If not, let e be an edge incident with S_n that minimizes $\min\{r : e \subset B_r\}$, where $B_r = \{x \in \mathbb{Z}^d : \|x\|_\infty \le r\}$, and set

$$E_{n+1} := \begin{cases} E_n \cup \{e\} & \text{if } e \text{ does not connect } S_n \text{ with a component } K_i,\ i \neq j; \\ E_n \cup \{e\} \cup K_i & \text{if } e \text{ connects } S_n \text{ with } K_i. \end{cases}$$

When in the above $E_n \subset B_{r-1}$, i.e. a new part of the anchor is visited by the process, we make the further requirement that e be the edge along which the unit current flow from S_n to ∞ is maximal.

Let M_n be the effective conductance from S_n to ∞ in the complement of E_n, with the end points of edges of K identified and any loops erased:

$$M_n := \mathcal{C}(S_n \leftrightarrow \infty \text{ in } (\mathbb{Z}^d/K) \setminus E_n).$$

Then by [38, Lemma 3.3], [43, Theorem 7], $(M_n)_{n\ge 0}$ is a martingale with respect to the filtration $\mathcal{F}_n$ generated by E_n, $\mathfrak{F}_{j,K} \cap E_n$.

The M_0 term is no longer constant, as in the original proof. Nevertheless, the argument of [38, Theorem 4.1] gives:

$$\mathbb{P}(\mathrm{desc}(K_j) \not\subset D_n \,|\, T|_{D_k} = K) \le C_d n^{\frac{2-d}{2d}} M_0(K_j)$$

We now bound $M_0(K_j)$ still with the conditioning that on D_k we have the forest K. Therefore we can work on the graph produced by deleting any edges from D_k that do not appear in K and contracting each component of K to a distinct vertex. By definition, the effective conductance from K_j to ∞ is the infimum of the energy of functions that are zero on K_j and one except on finitely many vertices. Therefore consider the function defined by $g(v) = 0$ if $v \in K_j$ and one otherwise. This is clearly a

valid function with regards to the infimum and will have energy equal to the number of edges connected to K_j. As all edges in D_k that are not present in K have been deleted and K_j is a connected component of K, the only edges will be those connected to K_j from the outside of D_k. An upper bound for $M_0(K_j)$ is therefore provided by the size of this set which is at most $C_d|\partial D_k \cap K_j|$.

Summing over the connected components, and using the fact that the K_j's are disjoint and cover all of D_k, we get

$$\sum_j M_0(K_j) \leq C_d|\partial D_k| \leq C_d k^{d-1}.$$

Then as this bound is independent of K we can average over all possible K to get the unconditioned result:

$$\mathbb{P}(\mathrm{desc}(D_k) \not\subset D_n) \leq C_d n^{\frac{2-d}{2d}} k^{d-1}.$$

This completes the proof of Proposition 3.18. □

Proof of Theorem 3.12. If $W_k = W_{k,\Lambda}$ and T and T_Λ agree on $E^*_{W,k}$, then $\psi_{\mathcal{D}}$ and $\psi_{\mathcal{D},\Lambda}$ will produce the same sandpile configuration on D_k. Therefore to bound the difference of the measures on any cylinder event E defined on D_k it suffices to bound the probability that the descendants in the spanning trees differ, or the trees differ on that set of descendants.

$$\begin{aligned}
|\nu(E) - \nu_\Lambda(E)| &\leq \mathbb{P}(E_{\mathbb{Z}^d} \Delta E_\Lambda) \\
&\leq \mathbb{P}\left(W_k \neq W_{k,\Lambda} \text{ or } T|_{\mathsf{E}^*_{W,k}} \neq T_\Lambda|_{\mathsf{E}^*_{W,k}}\right) \\
&\leq \mathbb{P}\left(W_k \subset D_n \text{ but } W_k \neq W_{k,\Lambda} \text{ or } T|_{\mathsf{E}^*_{W,k}} \neq T_\Lambda|_{\mathsf{E}^*_{W,k}}\right) + \mathbb{P}(W_k \not\subset D_n) \\
&\leq C_d \frac{k^{d-2} n^2}{(N-n)^{d-2}} + C_d k^{d-1} n^{\frac{2-d}{2d}}.
\end{aligned} \tag{3.14}$$

The final step follows from applications of Propositions 3.13 and 3.18.

We now optimise the choice of n. We may assume $N \geq 2n$, in which case $(N - n)^{d-2} \geq c_d N^{d-2}$.

When $d \geq 5$, we take $n = \frac{1}{2}N$, which gives the bound $C_d k^{d-1} N^{\frac{2-d}{2d}}$.

When $d = 4$, the two terms in the right hand side of (3.14) are of the same order if $n = k^{4/9} N^{8/9}$. This gives the bound $C k^{26/9} N^{-2/9}$.

When $d = 3$, we take $n = k^{6/13} N^{6/13}$. This yields the bound $C k^{25/13} N^{-1/13}$. □

3.4 Rate of convergence in $\mathbb{Z}^2$.

In this section we bound the rate of convergence on $\mathbb{Z}^2$ in Theorem 3.20 below. As was the case for $d \geq 3$, the result will follow directly from the bijections and a bound on the probability that, in a suitable coupling, the descendants of D_k in $\mathbb{Z}^2$ differ from those in Λ. This bound is given in Proposition 3.21. Due to recurrence, we cannot use Wilson's method rooted at infinity, so the construction of the coupling is more involved. Write $G = (\Lambda \cup \{s\}, \mathsf{E}_\Lambda)$ for the graph on which the sandpile is defined. Recall that given a cylinder event E determined by the sandpile heights in D_k, we write $E_{\mathbb{Z}^2} = \{T : \psi_{\mathcal{D}}(T) \in E\}$ and $E_\Lambda = \{T_\Lambda : \psi_{\mathcal{D},\Lambda}(T_\Lambda) \in E\}$, where T is a sample from WSF and T_Λ is a sample from UST_G.

Theorem 3.20. *Let E be a cylinder event determined by the sandpile heights in D_k, and let $\Lambda \subset \mathbb{Z}^2$ be a finite set. Let N be the largest integer such that $D_N \subset \Lambda$. Given $\varepsilon > 0$, there is a constant $C = C(\varepsilon) > 0$ and a coupling $\mathbb{P} = \mathbb{P}_{\Lambda,k,\varepsilon}$ of T and T_Λ, such that in this coupling we have*

$$|\nu(E) - \nu_\Lambda(E)| \leq \mathbb{P}(E_{\mathbb{Z}^2} \Delta E_\Lambda) \leq C \frac{k^{5/32}}{N^{1/16-\varepsilon}}.$$

We will write W_k, respectively $W_{k,\Lambda}$, for the set of descendants of D_k in T, respectively T_Λ. Then Theorem 3.20 follows immediately from the following proposition.

Proposition 3.21. *For any $k, \varepsilon > 0$ there exists Λ' such that for all $\Lambda \supset \Lambda'$, with N being the largest integer such that $D_N \subset \Lambda$ then there exists $C = C(\varepsilon) > 0$ and a coupling $\mathbb{P} = \mathbb{P}_{\Lambda,k,\varepsilon}$ of T and T_Λ such that in this coupling*

$$\mathbb{P}\left(W_k \neq W_{k,\Lambda} \text{ or } T \text{ and } T_\Lambda \text{ differ on some edge touching } W_k\right) \leq C \frac{k^{5/32}}{N^{1/16-\varepsilon}}.$$

The coupling will be achieved by passing to the planar dual graphs. The idea is to construct paths in the dual tree that together surround D_k in such a way that all descendants of D_k are necessarily in the interior of the region defined by the paths. Then it will be sufficient to couple the dual trees in the interior of that region.

Let $G^* = (\Lambda^*, \mathsf{E}^*_\Lambda)$ denote the planar dual of G. The vertex set Λ^* is naturally identified with a subset of the dual lattice $(\mathbb{Z}^2)^* = \mathbb{Z}^2 + (1/2, 1/2)$. The planar graph G^* has one unbounded face: the face corresponding to the sink s via duality. The dual spanning tree T^*_Λ is defined on G^*, by including a dual edge e^* in T^*_Λ if and only if the corresponding edge e is not in T_Λ. Then T^*_Λ is a sample from UST_{G^*} (i.e. with free boundary conditions). It is well known that as $\Lambda \uparrow \mathbb{Z}^2$, the measure UST_{G^*} converges weakly to the free spanning forest measure FSF, which for $\mathbb{Z}^2$ coincides with WSF [44, 37]. Let T^* denote a sample from this measure on the graph $(\mathbb{Z}^2)^*$. We refer to paths in $\mathbb{Z}^2$ as *primal* paths, and paths in $(\mathbb{Z}^2)^*$ as *dual* paths. Let o^* be the dual vertex $o + (1/2, 1/2) \in (\mathbb{Z}^2)^*$, where o is the origin in $\mathbb{Z}^2$. For any $m \geq 0$ we define the

balls in the dual graph:

$$D_m^* := \{w \in (\mathbb{Z}^2)^* : |w - o^*| \leq m\}.$$

The construction of the coupling is broken down into a sequence of steps, and the required estimates stated as lemmas. We collect the estimates at the end and prove Proposition 3.21. The integers $\ell \geq 1$ and $k < n < r < R < N$ will be parameters that we choose at the end to optimize the bound.

Step 1. Coupling the backbones inside D_r^***.** We will need to work with fixed "backbones" in our trees. Since T^* has one end WSF-a.s., there is a unique infinite path γ^* in T^* that starts at o^*. We call γ^* the *backbone* of T^*. The free spanning tree on Λ^* does not have a unique backbone (there are typically several paths from o^* to the boundary of Λ^*). Therefore, we will first work with the *wired* boundary condition in the *dual graph*, i.e. we consider the graph $\widetilde{G}^* = (\Lambda^* \cup \{s^*\}, \widetilde{\mathsf{E}}_\Lambda^*)$ obtained by connecting each vertex in Λ^* to s^* by as many edges as it needs, for its degree to be 4. Then we will compare $\mathsf{UST}_{\widetilde{G}^*}$ to UST_{G^*} using the well known monotone coupling between them [44, 37]. Let $\widetilde{T}_\Lambda^*$ denote a sample from $\mathsf{UST}_{\widetilde{G}^*}$. Let γ_Λ^* denote the unique path between o^* and s^* in $\widetilde{T}_\Lambda^*$. We call γ_Λ^* the *backbone* of $\widetilde{T}_\Lambda^*$.

We fix a coupling between γ^* and γ_Λ^* that maximizes the probability that their first ℓ steps are identical. The next lemma collects some LERW estimates from the literature that we use to estimate the probability that the restrictions of γ_Λ^* and γ^* to the ball D_r^* differ from each other.

Lemma 3.22. *(i) There exists a constant $C > 0$ such that for all $\ell < \sqrt{N}$, we have*

$$\mathbb{P}(\textit{first } \ell \textit{ steps of } \gamma^* \textit{ and } \gamma_\Lambda^* \textit{ are not identical}) \leq C\frac{\ell^2}{N}\ln\left(\frac{N}{\ell}\right).$$

(ii) There exists a constnat $C > 0$ such that if $R > 4r$, we have

$$\mathbb{P}(\gamma_\Lambda^* \textit{ returns to } D_r^* \textit{ after its first exit from } D_R^*) \leq C\frac{r}{R}$$

and

$$\mathbb{P}(\gamma^* \textit{ returns to } D_r^* \textit{ after its first exit from } D_R^*) \leq C\frac{r}{R}.$$

(iii) For all $\varepsilon > 0$ there exists $M > 0$ such that for all $R > M$ we have that

$$\mathbb{E}[\textit{number of steps of } \gamma^* \textit{ until first exit from } D_R^*] = R^{\frac{5}{4}+\varepsilon}$$

(iv) For all $\lambda, \varepsilon > 0$, $N > 4R$ we have that there exist $C(\varepsilon), C_1, C_2 > 0$ such that

$$\mathbb{P}(\textit{number of steps of } \gamma_\Lambda^* \textit{ until first exit from } D_R^* > \lambda C(\varepsilon) R^{\frac{5}{4}+\varepsilon}) \leq C_1 e^{-C_2\lambda}.$$

Remark 3.23. *Note that in contrast with [35, Proposition 11.3.1], the above bounds*

give us power law (rather than logarithmic) control on the errors, since we are free to discard a collection of "bad paths" in D_r^* of small probability on which convergence to the infinite LERW would be much slower.

Proof of Lemma 3.22. (i) The statement follows from [33, Proposition 7.4.2]. Note that although the exact statement is not present in the reference, it immediately follows from the proof presented there.

(ii) This is [5, Lemma 2.4].

(iii) This result was first shown by Kenyon [30] (stated there in the upper half plane). It also follows by combining [4, Proposition 6.2(2)] and [42, Theorem 5.7].

(iv) This follows from [4, Corollary 3.4], [4, Theorem 5.8(4)] and part (iii). □

The next lemma puts the above estimates together and bounds the probability that the restrictions of γ_Λ^* and γ^* to the ball D_r^* are not identical.

Lemma 3.24. *There exist constants $C, C(\varepsilon), C_1, C_2 > 0$ such that for all $\lambda, \varepsilon > 0$, and sufficiently large N, with $N > 4R > 16r$, we have that*

$$\mathbb{P}(\gamma_\Lambda^* \cap D_r^* \neq \gamma^* \cap D_r^*) \leq C\frac{\lambda^2 C(\varepsilon)^2 R^{\frac{5}{2}+2\varepsilon}}{N} \ln\left(\frac{N}{\lambda C(\varepsilon) R^{5/4+\varepsilon}}\right) + C_1 \exp(-C_2\lambda) + 2C\frac{r}{R}$$

Proof. Let F_1 be the event that the first ℓ steps of γ^* and γ_Λ^* coincide, the event maximized by our choice of coupling. We therefore need to choose ℓ appropriately to get the desired result.

Let F_2 be the event that the length of γ_Λ^* until first exit of D_R^* is less than ℓ.

Let F_3 be the event that neither γ_Λ^* nor γ^* return to D_r^* after their first exits from D_R^*.

On the event $F_2 \cap F_3$, we have that the first ℓ steps of γ_Λ^* includes $\gamma_\Lambda^* \cap D_r^*$. If F_1 also occurs, then we have $\gamma_\Lambda^* \cap D_r^* = \gamma^* \cap D_r^*$. We choose $\ell = \lambda C(\varepsilon) R^{5/4+\varepsilon}$. By Lemma 3.22(i),(iv),(ii) we have

$$\begin{aligned}
&\mathbb{P}(\gamma_\Lambda^* \cap D_r^* \neq \gamma^* \cap D_r^*) \\
\leq &\mathbb{P}(F_1^c) + \mathbb{P}(F_2^c) + \mathbb{P}(F_3^c) \\
\leq &C\frac{\lambda^2 C(\varepsilon)^2 R^{\frac{5}{2}+2\varepsilon}}{N} \ln\left(\frac{N}{\lambda C(\varepsilon) R^{5/4+\varepsilon}}\right) + C_1 \exp(-C_2\lambda) + 2C\frac{r}{R}.
\end{aligned}$$

□

Step 2. Constructing the dual paths that surround D_k^*. On the event $\gamma_\Lambda^* \cap D_r^* \neq \gamma^* \cap D_r^*$, we extend the coupling of γ_Λ^* and γ^* to a coupling of WSF and $\mathsf{UST}_{\widetilde{G}_\Lambda^*}$ in an arbitrary way. (For example: make them conditionally independent given the backbones.) On the event $\gamma_\Lambda^* \cap D_r^* = \gamma^* \cap D_r^*$, we extend the coupling via Wilson's stacks of arrows construction. For each $x \in D_r^* \setminus \gamma^*$, we assign identical stacks for the constructions in Λ^* and $(\mathbb{Z}^2)^*$, respectively. For all other vertices, the stacks in Λ^* are

assigned independently from those in $(\mathbb{Z}^2)^*$. This defines a coupling of WSF and $\mathsf{UST}_{\widetilde{G}^*}$ on $(\mathbb{Z}^2)^*$.

We now construct the required set of dual paths. Write γ_r^* for the portion of γ^* up to its first exit from D_r^*.

Definition 3.25. *By a* block*, we mean a set U of dual edges with the properties:*
(i) $U \subset D_n^* \setminus D_k^*$;
(ii) $U \cup \gamma_r^*$ is a connected set of edges;
(iii) the set of vertices of $U \cup \gamma_r^*$ disconnects D_k^* from $(D_n^*)^c$.

Lemma 3.26. *There exists $C > 0$ such that for $r > 4n > 16k$ and $\gamma_\Lambda^* \cap D_r^* = \gamma^* \cap D_r^*$ we have that*

$$\mathbb{P}\begin{pmatrix}\text{we can pop a set of coloured cycles contained in}\\ D_n^* \setminus D_k^* \text{ so that the arrows revealed contain a block}\end{pmatrix} \geq 1 - C\left(\frac{k}{n}\right)^{1/4} - C\frac{n}{r}.$$

Proof. Due to Lemma 3.22(ii), we have $\mathbb{P}(\gamma^* \cap D_n^* \neq \gamma_r^* \cap D_n^*) \leq C(n/r)$. Henceforth assume that we are on the event when $\gamma^* \cap D_n^* = \gamma_r^* \cap D_n^*$.

We start with a minor adaptation of the argument of [1, Lemma 6.1]. Let $v \in (\mathbb{Z}^2)^*$ be a vertex at distance $\sqrt{kn}$ from o^*, and let $\{S(n)\}_{n\geq 0}$ be simple random walk starting at v. Let τ be the first time when either S exits $D_n^* \setminus D_k^*$, or when the loop-erasure of S has made a non-contractible loop around D_k^*. Let us use the sequence $S(1), S(2), \ldots, S(\tau)$ as our successive choices in Wilson's algorithm, where γ_r^* is already part of the tree to be constructed. That is, whenever a random walk step is to be made, we use the next step of S for the random walk step, and whenever a new vertex is to be chosen in the algorithm, we use the next vertex not in the tree that is visited by S as the new vertex.

We claim that on the event $S[0,\tau] \subset D_n^* \setminus D_k^*$ the set of edges, U, that this algorithm has included in the tree by time τ is a block. Indeed, condition (i) holds because the walk never left $D_n^* \setminus D_k^*$. Also, observe that the set of vertices of $\mathrm{LE}(S[0,\tau))$ do not get erased, and hence condition (iii) holds. Finally, condition (ii) holds, because each piece of the tree we create gets joined to γ_r^* (here is where we use that $\gamma^* \cap D_n^* = \gamma_r^* \cap D_n^*$). Note that since $S(\tau - 1)$ does not get erased, the last piece is also joined. This proves the claim. Interpreting the construction in terms of stacks of arrows, we see that the probability of the event in part (i) is at least the probability that $S[0,\tau] \subset D_n^* \setminus D_k^*$.

The probability that a non-contractible loop is created could be bounded by $\geq 1 - C(k/n)^\zeta$ with some $\zeta, C > 0$, by ideas similar to [35, Exercise 3.3], showing the statement (i) with ζ in place of $1/4$. In order to get the explicit exponent $1/4$, we combine the argument with an idea that was inspired by [6].

Again we are going to start with γ_r^* as our initial tree. Choose a subpath $\gamma_{k,n}^*$ of γ_r^* that forms a crossing from D_k^* to $(D_n^*)^c$. Write H_ρ for the circle of radius ρ centred at

o^*. Define the annulus:

$$A_{k,n} = \{z \in \mathbb{R}^2 : k+1 < |z - o^*| < n-1\}.$$

Choose a point, not necessarily a vertex, $Q \in \gamma^*_{k,n} \cap H_{\sqrt{kn}}$, and let $\alpha_0 = H_{\sqrt{kn}} \setminus \{Q\}$. Let P_1 be the mid-point of α_0, and let v_1 be a vertex of $(\mathbb{Z}^2)^*$ closest to P_1. Run a random walk $S^{(1)}$ from v_1 to $\gamma^*_{k,n}$, and add edges to the tree in the same way as we did with S. Let π_1 be the set of edges added. Note that π_1 is not necessarily a connected set of edges, however, $\gamma^*_r \cup \pi_1$ is. From the two subarcs of α_0 defined by P_1, throw away the one that is on the same side of $\gamma^*_{k,n}$ as where π_1 hit, and let us call the other arc α_1. On the event when $\{S^{(1)}\} \subset A_{k,n}$, the arc α_1 has the property that any dual lattice path from H_k to H_n that is vertex-disjoint from $\gamma^*_r \cup \pi_1$ has to intersect α_1.

Continue inductively in the following way. Suppose that for some $i \geq 1$ the arc α_i and the sets of edges $\pi_1, \ldots, \pi_i$ have been defined. Let P_{i+1} be the mid-point of α_i and let v_{i+1} be the vertex of $(\mathbb{Z}^2)^*$ closest to P_{i+1}. Run a random walk $S^{(i+1)}$ from v_{i+1} to $\gamma^*_{k,n} \cup \pi_1 \cup \cdots \cup \pi_i$, and let π_{i+1} be the set of edges that get added to the tree. From the two subarcs of α_i, throw away the one that is on the same side of γ^*_r as where π_{i+1} hit, and call the other one α_{i+1}. On the event when $\{S^{(i+1)}\} \subset A_{k,n}$, the arc α_{i+1} has the property that any dual lattice path from H_k to H_n that is vertex-disjoint from $\gamma^*_r \cup \pi_1 \cup \cdots \cup \pi_{i+1}$ has to intersect α_{i+1}.

The construction is well defined until a time when the length of the arc α_i becomes of order 1. Stop the construction the first time when the arc length of α_i is less than 10, say. We can select further vertices $v_{i+1}, \ldots, v_{i+K}$ (with K a fixed constant, say, $K = \lceil 10\sqrt{2} + 4 \rceil$) such that if we start further random walks at these vertices, then $\gamma^* \cup \pi_1 \cup \cdots \cup \pi_{i+K}$ contains a block. An example of the start of this construction is shown in Figure 3.1.

It remains to bound the probability that the walks $S^{(1)}, S^{(2)}, \ldots$ all remain inside $D^*_n \setminus D^*_k$. The i-th walk $S^{(i)}$ starts at distance $O(2^{-i}\sqrt{kn})$ from the current tree $\mathfrak{T}_{i-1} := \gamma^*_{k,n} \cup \pi_1 \cup \cdots \cup \pi_{i-1}$. If it were to leave $D^*_n \setminus D^*_k$ without hitting $\mathfrak{T}_{i-1}$, it would first have to leave the ball

$$B^*(v_i; (1/4)\sqrt{kn}) := \{w \in (\mathbb{Z}^2)^* : |w - v_i| \leq (1/4)\sqrt{kn}\}.$$

without hitting $\mathfrak{T}_{i-1}$. Using Beurling's estimate [35, Section 6.8], the probability of this can be controlled

$$\mathbb{P}(\ S^{(i)} \text{ hits } \mathfrak{T}_{i-1} \text{ before exitting } B^*(v_i; (1/4)\sqrt{kn}) \leq C(2^{-i}\sqrt{kn}/\sqrt{kn})^{1/2}.$$

Regardless of where the walk exits $B^*(v_i; (1/4)\sqrt{kn})$, the exit point z^*_i is still at distance $\asymp \sqrt{kn}$ from o^*. It follows, again using Beurling's estimate, that the probability that the continuation of the walk from z^*_i exits D^*_n without hitting $\mathfrak{T}_{i-1}$ is at most $C(\sqrt{kn}/n)^{1/2}$. Similarly, together with a time-reversal argument, the probability that the walk started

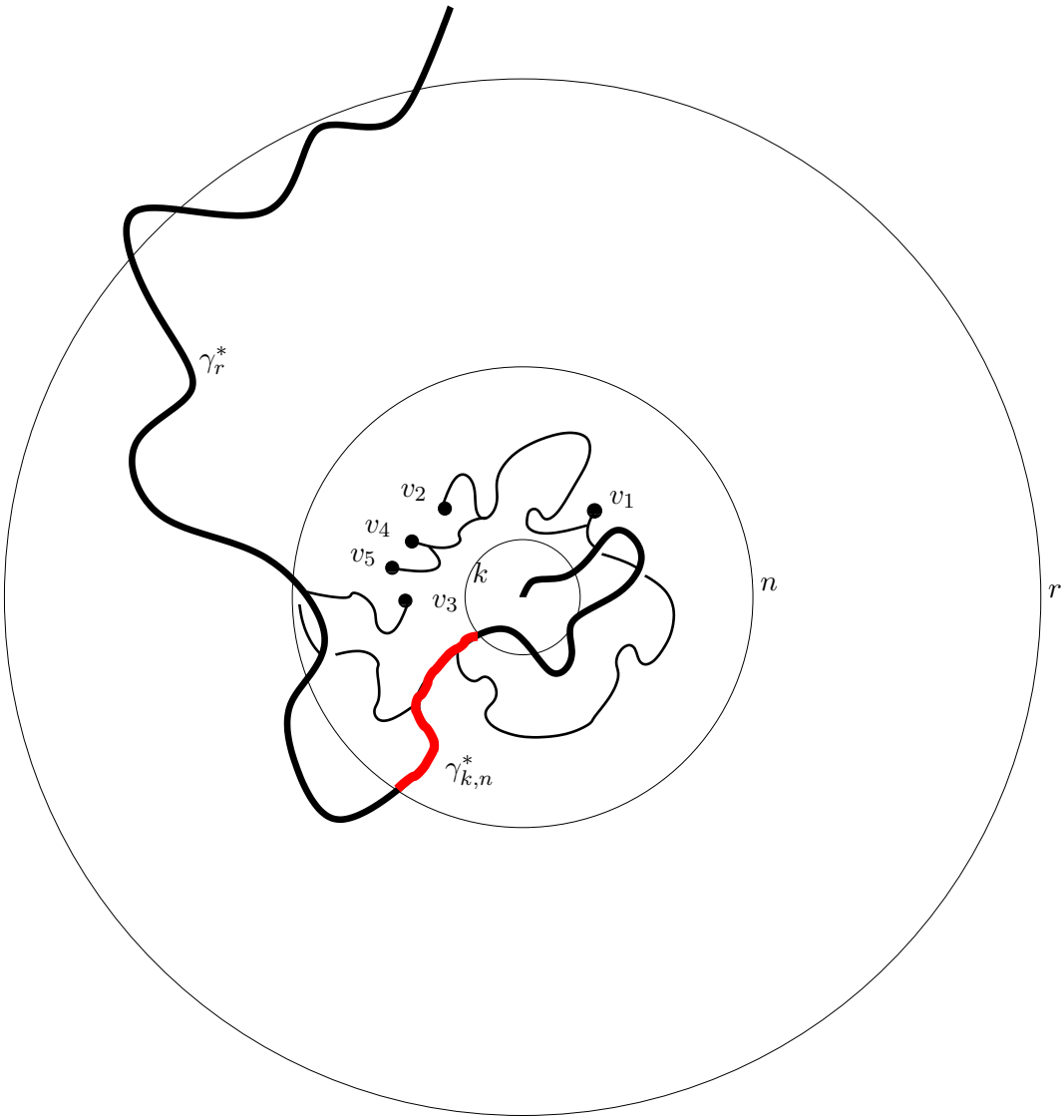

Figure 3.1: *An example of the construction of a block. The thick line is γ_r^*, and the red piece is $\gamma_{k,n}^*$. LERWs were started successively at v_1, v_2, etc. Note the gaps between pieces in some of the LERWs, where an intersection with $\gamma_r^* \setminus \gamma_{k,n}^*$ has occurred.*

at z_i^* hits D_k^* before hitting $\mathfrak{T}_{i-1}$ is at most $C(k/\sqrt{kn})^{1/2}$. Combining these three estimates we get the bound

$$\mathbb{P}(\{S^{(i)}\} \not\subset D_n^* \setminus D_k^* \,|\, \mathfrak{T}_{i-1}) \le C \left(\frac{2^{-i}\sqrt{kn}}{\sqrt{kn}}\right)^{\frac{1}{2}} \times \left[\left(\frac{\sqrt{kn}}{n}\right)^{\frac{1}{2}} + \left(\frac{k}{\sqrt{kn}}\right)^{\frac{1}{2}}\right]$$
$$= C 2^{1-i/2} \left(\frac{k}{n}\right)^{\frac{1}{4}}.$$

Summing over i we get the claimed bound $C_1(k/n)^{1/4}$. □

Step 3. Coupling the set of descendants. We now complete the definition of the coupling of T_Λ and T. Fix a monotone coupling between $\widetilde{T}_\Lambda^*$ and T_Λ^*, such that $e^* \in \widetilde{T}_\Lambda^* \cap \mathsf{E}_\Lambda^*$ implies $e^* \in T_\Lambda^*$; see [37, Chapter 10]. Define T_Λ and T as the dual trees of T_Λ^* and T^*. This completes the definition of required coupling $\mathbb{P}_{\Lambda,k,\varepsilon}$.

Lemma 3.27.
(i) When $\widetilde{T}_\Lambda^*$ contains a block, we have $W_{k,\Lambda} \subset D_r$.
(ii) When T^* contains a block, we have $W_k \subset D_r$.
(iii) When the event in Lemma 3.26 occurs, we have $W_{k,\Lambda} = W_k \subset D_r$ and T and T_Λ agree on the set of edges with at least one end vertex in W_k.

Proof. (i) Since T_Λ^* is stochastically larger than $\widetilde{T}_\Lambda^*$, the edges in the block are also present in T_Λ^*. Since the the union of the block with γ_r^* is connected, any two dual vertices in the interior of the block are connected by a path in $\widetilde{T}_\Lambda^*$. Hence no new edges are added in the interior of the block when passing from $\widetilde{T}_\Lambda^*$ to T_Λ^*.

Suppose that D_k had a descendant $v \in D_r^c$ in T_Λ. Then there would be a primal path β starting at v that visits D_k and ends outside D_r. Since the block surrounds D_k, this would contradict the connectivity of the block (as a set of edges).

(ii) The same argument as in the previous paragraph applies here.

(iii) Since we are using the same stacks of arrows in $D_r^* \setminus \gamma_r^*$, the same block exists in Λ^* and in $(\mathbb{Z}^2)^*$, and the trees coincide in the interior of the region defined by the block. Therefore, the trees T_Λ and T also coincide in this region. By parts (i) and (ii), the set of descendants are contained in this region and are equal in T_Λ and T. □

Proof of Proposition 3.21. By Lemma 3.27 we have $W_{k,\Lambda} = W_k$ if the event in Lemma 3.26 occurred which in turn assumed that the event in Lemma 3.24 did not occur. Therefore we have

$$\mathbb{P}\left(W_k \neq W_{k,\Lambda} \text{ or } T \text{ and } T_\Lambda \text{ differ on some edge with an end vertex in } W_k\right)$$
$$\le C\frac{\lambda^2 C(\varepsilon)^2 R^{\frac{5}{2}+2\varepsilon}}{N} \ln\left(\frac{N}{\lambda C(\varepsilon) R^{5/4+\varepsilon}}\right) + C_1 \exp(-C_2\lambda) + C\frac{r}{R} + C\left(\frac{k}{n}\right)^{1/4} + C\frac{n}{r}.$$

We can now optimise our choice of parameters by taking $n = (kr^4)^{1/5}$, $r = (R^5k)^{1/6}$, $R = (kN^6)^{1/16}$ and choose λ such that $\lambda^2 R^{2\varepsilon} = N^{\epsilon}$, note these choices mean that the necessary criteria we required for intermediary results are satisfied. □

CHAPTER 4

BOUNDARY CONDITIONS ON THE STRIP.

4.1 Introduction.

In this section we are interested in the dependencies between the heights in a recurrent sandpile configuration.

The definition of a sandpile configuration being recurrent shows that there is a global influence effecting the sandpile configuration. However the existence of the thermodynamic limit for the sandpile measure, for example, suggests that the model is dominated by local factors. We want to try and quantify which of these two aspects is the dominant factor in its behaviour.

For the case of sandpiles on the graph $\mathbb{Z}^d$, $d \geq 2$, Járai and Redig, [26], showed that the sandpile measure, ν, was tail trivial, equivalently that for any cylinder event E and for all $\varepsilon > 0$ there exists an $n \in \mathbb{N}$ such that for any event F, depending only on vertices that are not in $[-n, n]^d$,

$$|\nu(E \cap F) - \nu(E)\nu(F)| \leq \varepsilon.$$

The aim of this chapter was to strengthen this result to give control of $\nu(E|F)$. We believe a statement of the following form should hold true but we are currently unable to prove it.

Open Question 4. Let $G = (V, E)$ be a graph with $\Lambda \subset V$ and ν the associated sandpile measure. Under what constraints on G does the following hold? Let E be any cylinder event determined by vertices in Λ. For all $\varepsilon > 0$ there exists an $n \in \mathbb{N}$ such that for any cylinder event, F, that is determined by vertices which are at least a graph distance of n from vertices in Λ, we have that

$$\big|\nu(E|F) - \nu(E)\big| \leq \varepsilon.$$

However as an intermediary step we have been able to prove this result in the special

case of a ladder graph, which is a subgraph of $\mathbb{Z}^2$.

Definition 4.1. *A ladder graph is of the form $G \times \mathbb{N} \cup \{s\}$ where the vertices in the left most copy of G, $G \times \{1\}$, are connected to s. A copy of the graph G is called a* rung*, with the i^{th} rung from the left being denoted by R_i.*

In this chapter we will take the rungs to be a finite subset of $\mathbb{Z}$.

Járai and Lyons showed in [25] that there exists a sandpile measure on the full ladder graph, which is where the graph is instead given by $G \times \mathbb{Z}$. As this graph has two-ends it does not have a unique natural measure and depending on the interpretation of the intuition for burning an infinite graph of "starting the burning process at infinity" the resulting measure can change. We will work with what they called in their paper the left-burnable measure.

Given a finite set of vertices V, a vertex $v \in V$ is allowed to burn if it satisfies two properties. Firstly its height is greater than the number of its neighbours that are currently unburnt, this is the understanding of burnable from Majumdar and Dhar's burning bijection [40]. Secondly we require that there is a nearest neighbour path from v to $R_{-\infty}$ that only uses vertices that have already been burnt or are in V^c. We will henceforth refer to this as the standard burning rule and the bijection it induces as the standard bijection.

The existence of the measure on infinite graphs was shown by considering rungs whose vertices all have maximal heights which are "renewals", in the sense that the sandpile configurations on either side of a maximal rung would be conditionally independent.

It therefore immediately follows that for the one-sided ladder graphs, $G \times \mathbb{N}$, that we consider in this thesis, there exists a measure for the sandpiles on the infinite graph. This is true because the measure of a sandpile, η, on our graph will have the same distribution as asking for the configuration on the positive rungs to correspond to η in the full ladder graph under the left-burnable measure conditioned on a maximal rung occurring at rung 0.

The existence of these "renewals" would immediately lead to an upper bound on the influence of the boundary conditions but it would be a very weak bound especially for large N.

The aim of this chapter is to prove the following related theorems which will provide better control over the influence of the boundary.

Theorem 4.2. *Let η be a recurrent sandpile configuration on the graph $[0, N] \times [-q_1, \infty) \subset \mathbb{Z}^2$, for $q_1 > 0$. Let η_i denote the configuration on the vertices in R_i. Let ν be the measure on recurrent sandpile configurations. We define $I := \{\eta_j = E_j \text{ for } j \in [-q, 0]\}$, the event that the sandpile between rung $-q$ and rung 0 is some known configuration, E. Further define $\eta(i, j)$ to be any event that is determined by the configuration of the sandpile between rungs i and j. Then with $k \geq N^2 + \gamma N$ for sufficiently large $\gamma > 0$ and some constants $c_1, c_2, c_3 > 0$, we have that*

$$\Big|\nu\big(\eta(k,\infty)|I\big) - \nu\big(\eta(k,\infty)\big)\Big| \leq c_1 \exp(-c_2\gamma + c_3 N).$$

To prove this we will first prove the result on the finite graph.

Theorem 4.3. *Let η be a recurrent sandpile configuration on the graph $[0,N] \times [-q_1,q_2] \subset \mathbb{Z}^2$, for some $q_1, q_2 > 0$. Let η_i denote the configuration on the vertices in R_i. Let $\nu_{[-q_1,q_2]}$ be the measure on recurrent sandpile configurations on the graph $[0,N] \times [-q_1,q_2] \subset \mathbb{Z}^2$. We define $I := \{\eta_j = E_j \text{ for } j \in [-q_1,0]\}$, the event that the sandpile between rung $-q_1$ and rung 0 is some known configuration, E. Further define $\eta(i,j)$ to be any event that is determined by the configuration of the sandpile between rungs i and j. Then with $q_2 > k \geq N^2 + \gamma N$ for sufficiently large $\gamma > 0$ and some constants $c_1, c_2, c_3 > 0$, we have that*

$$\Big|\nu_{[-q_1,q_2]}\big(\eta(k,q_2)|I\big) - \nu_{[-q_1,q_2]}\big(\eta(k,q_2)\big)\Big| \leq c_1 \exp(-c_2\gamma + c_3 N). \tag{4.1}$$

The key to the proof of our theorems is to use a variation of the standard burning bijection. This allows us to convert the events in (4.1) into statements about spanning trees.

However the sandpile event we wish to condition upon does not convert into an event that is easily dealt with, therefore it will be beneficial to further condition upon a stronger spanning tree event which is a subset of I under this modified bijection.

It will turn out that the stronger conditioning we want to use is to specify the connection pattern of a grove at R_1. A grove and spanning forest can both be constructed using LERWs hence we can use our knowledge of random walks, which are much more malleable objects than sandpile configurations, to study the probability of the relevant events.

Using these observations the proofs of the theorems effectively come down to showing that we can bound the impact of the dependence of the spanning tree at R_k on the spanning tree at R_0 when conditioned on the partition induced by the grove at R_1. This is done in two parts.

Firstly we will give an upper bound for the probability that a path in a grove on R_1 will reach R_{KN}.

Having shown this it suffices to bound the probability that the spanning tree at R_k will depend on the spanning tree up to R_{KN}. This is done using the idea of a block, similar to the construction in the previous chapter.

The remainder of the chapter has the following structure.
Section 4.2 contains the statement of our notation and some preliminary results.
In Section 4.3 we introduce and justify the adaptation of the burning bijection.
In Section 4.4 we provide the bounds for the distance that paths in a grove will reach.
Section 4.5 contains a bound for the probability of having a block in a spanning tree.
Section 4.6 brings these results together to conclude the proofs of theorems 4.2 and

4.3.
In Section 4.7 we briefly consider an equivalent statement on graphs with different boundary conditions.

4.2 Preliminary results.

Throughout this chapter we will use $c, c_1, c_2 \dots$ to represent unspecified positive constants which have unless otherwise stated will have no dependence on any other variables. We may use the same notation to represent different constants in different results throughout the chapter, however we will maintain consistency of the constants within a calculation.

We will begin by defining the random walk terminology we will require.

Definition 4.4. *$\big(S_v(j)\big)_{j\in\mathbb{N}}$ denotes a simple random walk started at v, if the start vertex is obvious from the setting or not important we will often drop the v from the notation. We denote by τ the first hitting time with respect to the random walk, and by ξ the first exit time. More precisely we define*

$$\begin{aligned}\tau_x &:= \min\{j \geq 0 : S(j) = x\}\\ \tau_x^+ &:= \min\{j > 0 : S(j) = x\}\\ \tau_A &:= \min\{j \geq 0 : S(j) \in A\}\\ \xi_A &:= \min\{j \geq 0 : S(j) \notin A\}\end{aligned}$$

Let $G_D(x, y)$ be the Green's function of a simple random walk that is stopped upon first exiting a domain D.

In order to state some preliminary result we introduce the following notation with respect to two distinct vertices $a, w \in R_1$.
Take π_1 to be an arbitrary loop-erased random walk in $\cup_{i\geq 1} R_i$ which was started at v_1 and terminated upon hitting v_2, with $v_1, v_2 \in R_1$ such that they are in the interval between a and w.
Set π_3 to be an arbitrary loop-erased random walk in $\cup_{i\geq 1} R_i$ which starts at v_3 and ends at v_4 with $v_3, v_4 \in R_1$ such that a and w are in the interval between v_3 and v_4.

Define the sets

$$\begin{aligned}Z &:= \pi_1 \cup \pi_3 \cup (R_1 \setminus \{a\})\\ Z' &:= \pi_1 \cup \pi_3 \cup R_1.\end{aligned}$$

Let the domain D_1 be the connected component of $\{[0, N] \times [1, \infty)\} \setminus \{\pi_1 \cup \pi_3\}$ that contains the vertices a and w.

The exact construction of the instigating paths, π_1, π_3 and the reason we need to consider such sets, will become apparent later in this chapter. However as it will turn

out that the results we will get will be uniform amongst the possible paths, it is not necessary to formally define how they arise yet and it suffices to state the results in terms of arbitrary paths that satisfy the above restrictions.

4.2.1 Decomposition of the domain.

In Section 4.4 we will decompose the domain D_1 into different sections based upon a sequence of balls that a random walk would have to travel through. We now describe the criteria for the decomposition.

Definition 4.5. *Let B_k be a ball with centre o_k and radius $0 < r_k \leq N/2$ (we do not insist that o_k is a lattice point).*
Define B'_k to be the ball of radius $\left(\sqrt{2-\sqrt{2}}\right) r_k$ centred at o_k.
Let α be a path from a vertex in R_1 that hits B_k, denote the first intersection point as $v_k \in \partial_i B_k$. Then define $B''_{k,\alpha}$ to be the ball of radius r_k centred at v_k.

We wish to choose the balls B_k in such a way that we maximise the total number of balls whilst still having each ball satisfy the following criteria.
For all k:
(i) $B_k \cap Z' = \emptyset$
(ii) Every path from a vertex in R_1 to $R_{|o_k|+r_k}$ that does not intersect Z' must intersect B_k.
(iii) Given B_{k-1} and for any path α, from R_1 that hits B_{k-1} before Z', we require that $B_k \cap B''_{k-1,\alpha} = \emptyset$.

See Figure 4.1 for an example decomposition.

Remark 4.6. *The radii for B'_k and $B''_{k,\alpha}$ have been chosen such that the part of $\partial_i B''_{k,\alpha}$ that intersects B'_k is given by an arc with angle $\pi/2$. This will be required later in the chapter so that we will have the correct set up in order to be able to apply Lemma 4.7 which is stated next.*

We now state some preliminary results that will be needed for proofs later in the chapter.

Lemma 4.7. *There exists a constant $c > 0$ such that for all $\theta \in [-\pi, \pi]$ there exists $R \in \mathbb{N}$ such that for all $r > R$ the following holds. Let B be a ball with centre $(0,0)$ and radius r. Suppose there exists a set $\mathfrak{A}$ such that $\mathfrak{A} \subset B$ but does not intersect the ball centred at $(r,0)$ of radius r. Then*

$$\mathbb{P}\Big(\arg(S_{(0,0)}(\xi_B)) \in [\theta - \pi/4, \theta + \pi/4] \Big| \xi_B < \tau_{\mathfrak{A}} \Big) \geq c.$$

Proof. The case for $\theta = 0$ is Proposition 3.5 in [42]. Although originally stated for a ball with an integer radius the extension to a real-valued radius follows immediately from the proof provided there. Consequently the same proof will hold for all θ by symmetry of the random walk and domain. □

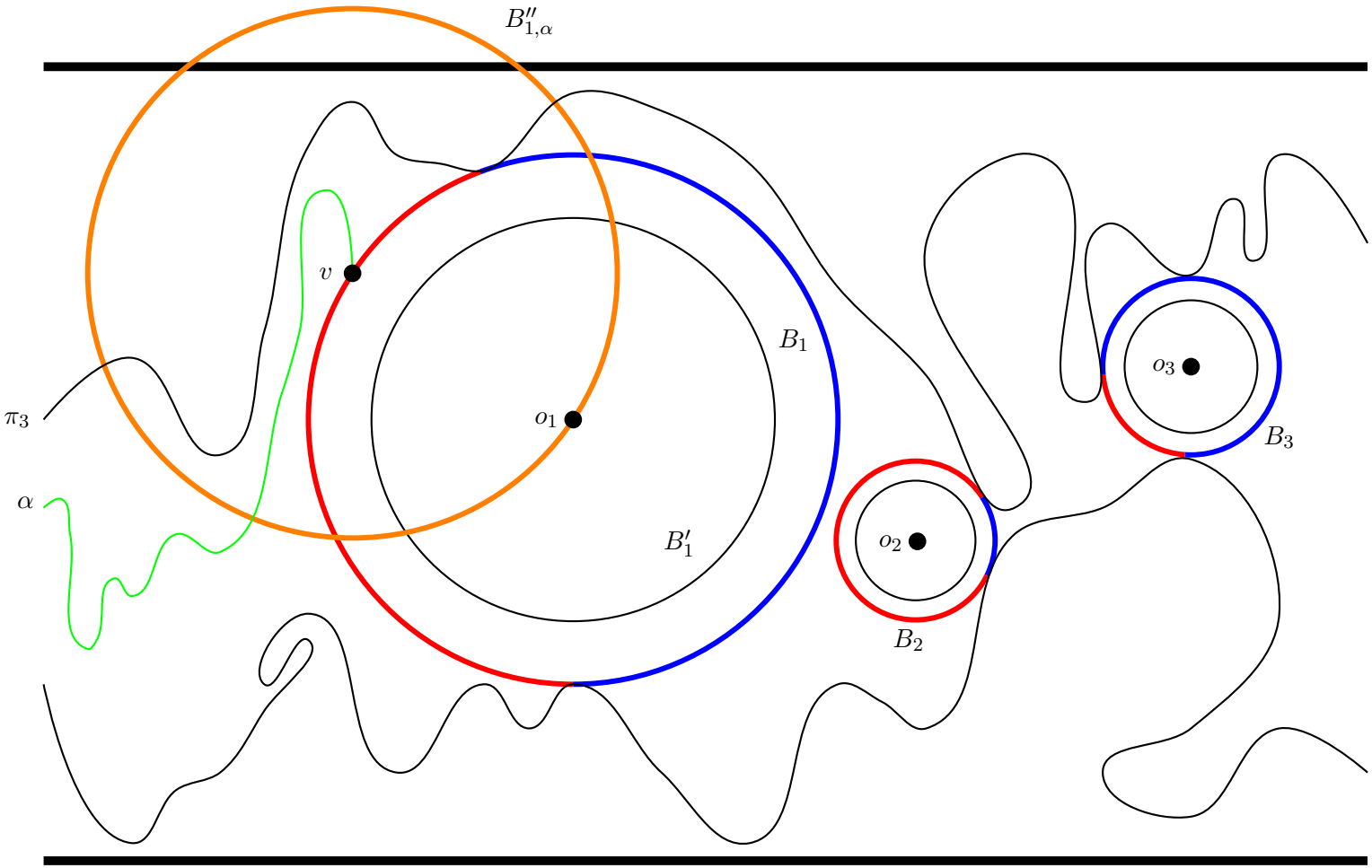

Figure 4.1: *An illustrative example of a potential decomposition of the domain, although it may be a non-optimal choice of balls. The red part of the ball's boundary are where a ball B'' may be centred.*

Lemma 4.8. *Let $c > 0$ be a positive constant, $u_1, w_1 \in \partial_i B'_k$ and let $\mathfrak{B}$ be a ball centred at u_1 which is contained in B_k and has a radius $r_k/8$, or a constant radius if $r_k/8 < 1$. If $x \in \partial_i \mathfrak{B}$ then*

$$\mathbb{P}_{w_1}(\tau_{u_1} < \tau_Z) \geq c\mathbb{P}_x(\tau_{u_1} < \tau_Z)$$

Proof. Firstly note that if $r_k < 8$ that the result is trivially true as the probability a random walk goes from w_1 to x without hitting Z or u_1 is at least a constant as we can find a viable path of constant length between them, so

$$\mathbb{P}_{w_1}(\tau_{u_1} < \tau_Z) \geq \mathbb{P}_{w_1}(\tau_x < \tau_{Z\cup\{u_1\}})\mathbb{P}_x(\tau_{u_1} < \tau_Z) \geq c\mathbb{P}_x(\tau_{u_1} < \tau_Z).$$

Henceforth we can assume $\mathfrak{B}$ has a radius of $r_k/8$, which is a sufficient condition to ensure that $\text{dist}(y, Z) \geq c_1 r_k \ \forall y \in \mathfrak{B}$.

We need to consider two cases depending on the distance between u_1 and w_1.

If $w_1 \notin \mathfrak{B}$ then $|u_1 - w_1| \geq c_2 r_k$ and so we can immediately apply the Harnack inequality. This is because the hitting probability is a harmonic function and there is a compact connected set containing w_1 and x whose vertices are at least $c_3 r_k$ from $\{u_1\} \cup Z$. Hence $\mathbb{P}_{w_1}(\tau_{u_1} < \tau_Z) \geq c\mathbb{P}_x(\tau_{u_1} < \tau_Z)$ holds.

If on the other hand u_1 and w_1 are both contained in $\mathfrak{B}$, then we need to take an extra step. This is to ensure that the distance to u_1 is at least $c_3 r_k$, where upon we are able to apply Harnack's inequality and compare to the walk from x.

$$\begin{aligned}
&\mathbb{P}_{w_1}(\tau_{u_1} < \tau_Z) \\
&= \mathbb{P}_{w_1}(\tau_{u_1} < \xi_{\mathfrak{B}}) + \mathbb{P}_{w_1}(\xi_{\mathfrak{B}} < \tau_{u_1}) \sum_{y\in\partial\mathfrak{B}} \mathbb{P}_{w_1}(S(\xi_{\mathfrak{B}}) = y | \xi_{\mathfrak{B}} < \tau_{u_1})\mathbb{P}_y(\tau_{u_1} < \tau_Z) \\
&\geq \mathbb{P}_{w_1}(\tau_{u_1} < \xi_{\mathfrak{B}})\mathbb{P}_x(\tau_{u_1} < \tau_Z) \\
&\qquad + \mathbb{P}_{w_1}(\xi_{\mathfrak{B}} < \tau_{u_1}) \sum_{y\in\partial\mathfrak{B}} \mathbb{P}_{w_1}(S(\xi_{\mathfrak{B}}) = y | \xi_{\mathfrak{B}} < \tau_{u_1})c\mathbb{P}_x(\tau_{u_1} < \tau_Z) \\
&\geq c\mathbb{P}_x(\tau_{u_1} < \tau_Z)(\mathbb{P}_{w_1}(\tau_{u_1} < \xi_{\mathfrak{B}}) + \mathbb{P}_{w_1}(\xi_{\mathfrak{B}} < \tau_{u_1})).
\end{aligned}$$

Thus the result holds in both cases. □

Let $\mathfrak{D} = [0, 1] \times \mathbb{R}^+ \subset \mathbb{R}^2$. Let $\tilde{\tau}_A$ and $\tilde{\xi}_A$ be the hitting and exit times of a set by A by a standard two dimensional Brownian motion respectively. By $v \in \mathfrak{R}_j$ we mean that $v \in \mathbb{R}^2$ has x co-ordinate j.

Lemma 4.9. *There exists a constant $C_1 < 1$ such that for any $v \in \mathfrak{R}_j$*

$$\mathbb{P}_v(\tilde{\tau}_{\mathfrak{R}_{(j+1)}} < \tilde{\xi}_{\mathfrak{D}}) \leq C_1.$$

Note that the constant bound does not need to depend on the x co-ordinate.

Lemma 4.10. *Let v be a vertex in R_{MN} then, with $K \geq M+1$ and for sufficiently large N, there exists a constant $d > 0$ such that*

$$\mathbb{P}_v(\tau_{R_{KN}} < \tau_{Z'}) \leq \exp(-d(K-M)). \tag{4.2}$$

Proof. Firstly we will rescale the domain so the random walk is on the lattice $\mathfrak{D} \cap \frac{1}{N}\mathbb{Z}^2$, under this rescaling $v \in R_{KN}$ becomes $\mathfrak{v} \in \mathfrak{R}_K$. Next note that increasing the usable domain of the random walk will yield an upper bound for the event in the left hand side of (4.2), so in particular if Z' was replaced by $\mathfrak{D}^c$, we would obtain the relationship

$$\mathbb{P}_v(\tau_{R_{KN}} < \tau_{Z'}) \leq \mathbb{P}_{\mathfrak{v}}(\tau_{\mathfrak{R}_K} < \xi_{\mathfrak{D}}).$$

Now we can decompose the domain into disjoint boxes of size 1×1. Therefore we can see in order for the event to be successful that the random walk will have to cross at least $K - M$ disjoint boxes without hitting the boundary. Note that the probability of successfully crossing a box is dependent on the path through the previous boxes only through the starting vertex. Therefore we have

$$\mathbb{P}_{\mathfrak{v}}(\tau_{\mathfrak{R}_K} < \xi_{\mathfrak{D}}) \leq \prod_{i=M}^{K-1} \max_{w_i \in \mathfrak{R}_i} \mathbb{P}_{w_i}(\tau_{\mathfrak{R}_{(i+1)}} < \xi_{\mathfrak{D}}).$$

To bound this we consider the walk in each box separately. As N tends to infinity a random walk on $\mathfrak{D} \cap \frac{1}{N}\mathbb{Z}^2$ will converge to a standard two dimensional Brownian motion on $\mathfrak{D}$. Therefore we can approximate the probability of a random walk to cross a box with that given for a Brownian motion and then apply Lemma 4.9 to get

$$\mathbb{P}_{w_i}(\tau_{\mathfrak{R}_{(i+1)}} < \xi_{\mathfrak{D}}) \leq c_2 \mathbb{P}_{w_i}(\tilde{\tau}_{\mathfrak{R}_{(i+1)}} < \tilde{\xi}_{\mathfrak{D}}) \leq c_2 C_1.$$

By taking N sufficiently large we can make c_2 as near to 1 as required to make $c_2 C_1 < 1$.

Hence we can conclude that for sufficiently large N, $\exists c < 1$ such that

$$\mathbb{P}_v(\tau_{R_{KN}} < \tau_{Z'}) \leq c^{K-M}.$$

□

Remark 4.11. *The restriction of $K \geq M+1$ could be weakened however as this would complicate the proof and is an unnecessary case for what we require we have chosen not to prove the more general result.*

4.3 Burning based on rungs.

We now introduce a modification to the standard burning bijection. The motivation for the alteration is to more easily enable the separate treatment of the parts of the

graph where we have information, i.e the sandpile configuration up to and including R_0, and the part we have no more information than that which is derived through the influence of the first part. The advantage of this new bijection over the standard burning bijection is explained in Lemma 4.13 and Remark 4.14 after the construction.

Definition 4.12. *Let the* burning bijection centred at R_0 *be the following adaptation of the standard bijection, it will be denoted by* ϕ_{R_0} *a map from recurrent sandpile configurations to spanning trees.*

Suppose the underlying graph $G = [0, N] \times [-q_1, q_2] \cup \{s\}$ is a finite but arbitrarily large ladder graph and η is a sandpile configuration on G. The idea is to use the standard burning algorithm but only on the negative or positive rungs at each stage. Define two subgraphs of G,

$$\Lambda_1 := \cup_{i=0}^{q_1} R_{-i},$$
$$\Lambda_2 := \cup_{i=1}^{q_2} R_i.$$

Phase 1. We apply the usual burning algorithm to η with the restriction that we do not allow any vertex of Λ_2 to burn. That is, we define

$$B_0^{(1)} := \{s\},$$
$$U_0^{(1)} := \Lambda_1 \cup \Lambda_2,$$

and for $j \geq 1$ we inductively set:

$$B_j^{(1)} := \left\{ v \in U_{j-1}^{(1)} \cap \Lambda_1 : \eta(v) \geq \deg_{U_{j-1}^{(1)}}(v) \right\},$$
$$U_j^{(1)} := U_{j-1}^{(1)} \setminus B_j^{(1)}.$$

Note that there may be vertices in Λ_1 that do not burn in Phase 1. These vertices, together with the vertices in Λ_2, will burn in later phases.

There will exist a $k > 0$ such that $B_k^{(1)} = \emptyset$ eventually. The set of vertices that remain unburnt in Λ_1 at the end of phase 1 is defined as

$$U^{(1)} := U_k^{(1)}.$$

Assuming Phase $i-1$ has already been defined, then for even i we inductively define Phase i and $i+1$ as follows.

Phase i**.** We continue the burning algorithm on η with the restriction that no vertex of Λ_1 is allowed to burn. That is, we set

$$B_0^{(i)} := \cup_{j \geq 0} B_j^{(i-1)},$$
$$U_0^{(i)} := U^{(i-1)},$$

and for $j \geq 1$ we inductively set:

$$B_j^{(i)} := \left\{ v \in U_{j-1}^{(i)} \cap \Lambda_2 : \eta(v) \geq \deg_{U_{j-1}^{(i)}}(v) \right\},$$
$$U_j^{(i)} := U_{j-1}^{(i)} \setminus B_j^{(i)}.$$

There exists a smallest $k > 0$ such that $B_k^{(i)} = \emptyset$ at which point any vertices that can be burnt under the restrictions have been, so define the set of unburnt vertices in Λ_2 at the end of phase i as

$$U^{(i)} := U_k^{(i)}.$$

Phase i + 1**.** We continue the burning algorithm on η in much the same manner, except now with the restriction that no vertex of Λ_2 is allowed to burn. That is, we set

$$B_0^{(i+1)} := \cup_{j \geq 0} B_j^{(i)},$$
$$U_0^{(i+1)} := U^{(i)},$$

and for $j \geq 1$ we inductively set:

$$B_j^{(i+1)} := \left\{ v \in U_{j-1}^{(i+1)} \cap \Lambda_1 : \eta(v) \geq \deg_{U_{j-1}^{(i+1)}}(v) \right\},$$
$$U_j^{(i+1)} := U_{j-1}^{(i+1)} \setminus B_j^{(i+1)}.$$

There exists a smallest $k' > 0$ such that $B_{k'}^{(i+1)} = \emptyset$, this means no more vertices can currently be burned in Λ_1. For this value of k' we define

$$U^{(i+1)} := U_{k'}^{(i+1)}.$$

After a finite number of phases all vertices in Λ_1 and Λ_2 will be burnt. We now use these sets to construct the spanning tree.

If $v \in B_j^{(i)}$ for some $i \geq 1$ and $j \geq 1$, then we place an oriented edge pointing from v to some $w \in B_{j-1}^{(i)}$. In the case $j = 1$ such an edge exists, because v must have a neighbour outside $U_0^{(i)}$, and hence in $B_0^{(i)}$. In the case $j \geq 2$ such an edge also exists, because the requirement to burn v at step j implies that the degree of v in $U_{j-1}^{(i)}$ is strictly smaller than its degree in $U_{j-2}^{(i)}$. Hence v has a neighbour in $B_{j-1}^{(i)} = U_{j-2}^{(i)} \setminus U_{j-1}^{(i)}$. If there is more than one $w \in B_{j-1}^{(i)}$ neighbouring v, we make the choice of the edge dependent on $\eta(v)$, via the burning rule as we did in the usual burning bijection.

If $v \in B_j^{(i)}$, let

$$m_v := \left| \left\{ f : \text{tail}(f) = v, \text{head}(f) \in \bigcup_{j' < j} B_{j'}^{(i)} \right\} \right|,$$
$$F_v := \left\{ e : \text{tail}(e) = v, \text{head}(e) \in B_{j-1}^{(i)} \right\}.$$

Then there exists some $0 \leq \ell_v < |F_v|$ such that

$$\eta(v) = \deg(v) - m_v + \ell_v.$$

With respect to a well defined ordering of the vertices of F_v, we then place the ℓ_v^{th} edge of F_v in the tree.

Forgetting the orientation of all the edges will then yield a spanning tree of G, denoted $\phi_{R_0}(\eta)$, this can therefore be used to define the map ϕ_{R_0}. The proof that ϕ_{R_0} is indeed a bijection immediately follows from the one provided for Lemma 3.4.

Having described this modified burning bijection we now explain why it is necessary.

Recall the definition of a rooted grove and that a grove with respect to R_1 determines a partition of R_1, with a direction from each vertex towards the corresponding root vertex in their component. It will be these *rooted partitions* and set of all possible groves that could be associated to it that will be key to our argument.

Let $G = (V, E)$, define the graph Λ_1^+ by the vertex set of $V \cap \bigcup_{i \leq 1} R_i$ and the induced edges from E between two vertices in Λ_1 and those between a vertex of Λ_1 and a vertex of R_1.

Lemma 4.13. *Given a restriction of a recurrent configuration to Λ_1, η_{Λ_1}, and a rooted partition of R_1, p. Define a set of sandpile configurations on G by*

$$\Omega := \Omega(\eta_{\Lambda_1}, p) = \{\zeta : \zeta|_{\Lambda_1} = \eta_{\Lambda_1} \text{ and } \phi_{R_0}(\zeta) \text{ induces the partition } p \text{ on } R_1\}.$$

There is a unique spanning forest, F on Λ_1^+, such that $\phi_{R_0}(\zeta)|_{\Lambda_1^+} = F \quad \forall \zeta \in \Omega$.

Proof. Let $\zeta_1, \zeta_2 \in \Omega$, we will show that the spanning forest they produce under ϕ_{R_0} will be the same when restricted to Λ_1^+ .

During Phase 1 of the burning procedure we only burn vertices in Λ_1 and so as $\zeta_1|_{\Lambda_1} = \zeta_2|_{\Lambda_1}$ the process will be identical and thus so will the sets $B_j^{(1)}(\zeta_1) = B_j^{(1)}(\zeta_2)$, for $j > 0$.

Now when we reach Phase 2 the procedures may deviate because the configurations differ on Λ_2. However we can still deduce which vertices in R_1 will burn in this phase, because $\phi_{R_0}(\zeta_1)$ and $\phi_{R_0}(\zeta_2)$ induce the same partition p on R_1.

By definition of a rooted partition we know that the root is the first vertex that burns in a component and also that every other vertex in that component is connected via a path through Λ_2 to the root. This means that the root must burn in the first step of a phase and the other vertices in the component must burn in later steps in the same phase. Let p_1 be a component of p and v be the root in p_1. Then if v has a neighbour in R_0 that burnt in Phase 1 then $v \in B_1^{(2)}(\zeta_i)$ and $p_1 \subset \cup_{j \geq 1} B_j^{(2)}(\zeta_i)$ for $i = 1, 2$. Therefore $\cup_{j \geq 1} B_j^{(2)}(\zeta_1) \cap R_1 = \cup_{j \geq 1} B_j^{(2)}(\zeta_2) \cap R_1$.

Now as we know the subset of R_1 that burnt in Phase 2 we can perform Phase 3. The burning of ζ_1 will again match the burning of ζ_2 in this phase because for each vertex in Λ_1 the same set of neighbours have burnt in ζ_1 and ζ_2.

Proceeding inductively with respect to the number of phases we can conclude that the burning times of vertices in Λ_1 will agree for all sandpiles in Ω as will the phase in which vertices in R_1 burnt.

Next we can follow the burning procedure to start producing an oriented spanning tree on G, specifically the forest restricted to Λ_1^+. If i is odd then for $v \in B_j^{(i)} \cap \Lambda_1$ we know which neighbours burnt in $B_{j-1}^{(i)}$ and so we know which oriented edge to include in the spanning forest. This covers all edges oriented away from vertices in Λ_1.

For $w \in R_1$ we can also specify some of the oriented edges that will be included. If w is a root then an edge is placed directed from w to its neighbour in R_0, as this is the only neighbouring vertex that burnt in the previous phase.

If w is not a root then no oriented edges are directed out of it in F. This is because edges from w would be directed towards the root through vertices in Λ_2 and so no edge would be directed from w to vertices in Λ_1.

Therefore there is a unique forest F such that $F = \phi_{R_0}(\zeta)|_{\Lambda_1^+}$ for all $\zeta \in \Omega$. □

Remark 4.14. *Recall we are interested in conditioning on a sandpile configuration on Λ_1. By Lemma 4.13 we have seen that further conditioning on a rooted partition fixes the spanning forest on Λ_1^+ under ϕ_{R_0}. This allows us to interpret conditioning on a sandpile configuration in terms of a spanning tree event, namely having a rooted partition as a boundary condition.*

In contrast the standard bijection would not fix the forest on Λ_1 and so we could not reduce the problem to a question about spanning trees. This is because the spanning tree on Λ_1 under the standard bijection depends on the length of paths through Λ_2 not just which vertices are connected. Trying to condition on the length of paths between vertices in R_1 would make the following approach too unwieldy.

With this observation we now turn our attention to controlling aspects of the grove's distribution when conditioned upon having a given rooted partition.

4.4 Bounds for paths in a grove.

The next step towards being able to prove Theorem 4.3 is to control how far the paths in a spanning forest will reach when conditioned upon having a given rooted partition on R_1. This is done in the following Proposition and the remainder of this section is devoted to proving it.

Proposition 4.15. *Let $c_1, c_2, c_3 > 0$ be positive constants. Let $\mathfrak{g}$ be a grove with respect to R_1, with $\mathfrak{b}_\mathfrak{g}$ being the backbone of the grove. Let $\mathfrak{p}$ denote a rooted partition on R_1.*

Then for $K, N > 0$ sufficiently large

$$\mathbb{P}(\mathfrak{b}_\mathfrak{g} \cap R_{KN+N^2} \neq \emptyset |\ \mathfrak{b}_\mathfrak{g} \text{ induces } \mathfrak{p}) \leq c_1 \exp(-c_2 K + c_3 N).$$

We first wish to consider what type of paths connected to R_1 we will find in a grove.

Definition 4.16. *Given a set of $2m$ vertices, labelled $a_1, \dots a_m, b_1, \dots b_m$ with a path connecting a_i to b_i for each $i \in \{1, \dots m\}$, then there are two main ways that the paths can occur in relation to each other:*
*i)*Nested paths *are such that the vertices are ordered as $a_1, a_2, \dots a_m, b_m, \dots b_2, b_1$.*
*ii)*Adjacent paths *are such that the vertices are ordered as $a_1, b_1, a_2, b_2 \dots a_m b_m$.*

We say a component, $\mathfrak{g}_1$ is nested inside of another, $\mathfrak{g}_2$, if there exists a path in $\mathfrak{g}_1$ that is nested inside of a path in $\mathfrak{g}_2$. We say two components are adjacent if they are not nested.

Clearly these are not all of the possible realisations of how components are connected to a rung, however they are the main ones we need to consider. This is because the only other kinds of components will either be trivial or, as we will see later, can be bounded using results on nested and adjacent paths.

Next recall the definition of a grove and that a grove determines a partition of R_1, with a special vertex selected in each component. It will be these rooted partitions and the set of all possible groves that could be associated to a partition that we want to work with. Also recall that we have the resampling process to construct a grove uniformly using random walks.

Therefore if we are interested in paths in the grove we need to have control of LERWs with fixed start and end vertices conditioned to avoid other paths.

In the next two subsections we will provide bounds for the two main types of path before combining the results to say something about the grove as a whole.

4.4.1 Bound for a nested path.

Proposition 4.17. *Take $c_1, c_2 > 0$ as positive constants. Let w and a be distinct vertices in R_1 and π_1, π_3, Z satisfying Definition 4.4. Let $K, L \in \mathbb{N}$ be such that $K > L + 1$ and $\pi_1 \cap R_{LN} = \emptyset$. Then for sufficiently large N*

$$\mathbb{P}_w(\textit{LERW to } a \textit{ intersects } R_{KN} | \tau_a < \tau_Z^+) \leq c_1 \exp(-c_2(K - L))$$

Proof. Firstly note that if π_3 disconnects w from R_{KN} then the statement is trivially true, hence assume that that w is not disconnected from R_{KN}.

Next we observe that if a loop-erased random walk is to hit R_{KN} before hitting a then the random walk that generates it must also hit R_{KN} before a. Therefore we can say that

$$\mathbb{P}_w(\text{LERW to } a \text{ intersects } R_{KN} | \tau_a < \tau_Z^+) \leq \mathbb{P}_w(\tau_{R_{KN}} < \tau_a | \tau_a < \tau_Z^+). \tag{4.3}$$

The key to understand whether this event is successful is to split it into three parts, the start, middle and end. There will be a path from w, denoted S_w, and a path from a, denoted S_a, that both reach R_{KN} and then there will be a path that joins them together. All of the paths must not hit Z. The idea of the proof is to generate the two

paths to R_{KN} simultaneously and utilise the fact that with high probability at some point the paths get close to each other whilst being away from the boundary. When the paths are near each other we will be able to simplify the expression by cancelling similar terms.

We first consider the following decomposition of a successful path.

$$\begin{aligned}&\mathbb{P}_w(\tau_{R_{KN}} < \tau_a < \tau_Z^+)\\ &= \mathbb{P}_w(\tau_{R_{KN}} < \tau_{Z'}^+) \sum_{\mathfrak{v}\in R_{KN}} \mathbb{P}_w(S_w(\tau_{R_{KN}}) = \mathfrak{v} | \tau_{R_{KN}} < \tau_{Z'}^+)\mathbb{P}_\mathfrak{v}(\tau_a < \tau_Z)\end{aligned} \tag{4.4}$$

Then we can manipulate the final term using the reversibility of a random walk.

$$\begin{aligned}\mathbb{P}_\mathfrak{v}(\tau_a < \tau_Z) &= \mathbb{P}_\mathfrak{v}(\tau_a < \tau_Z | \tau_\mathfrak{v}^+ > \tau_a \wedge \tau_Z) G_{D_1}(\mathfrak{v},\mathfrak{v})\\ &= \mathbb{P}_a(\tau_\mathfrak{v} < \tau_Z | \tau_a^+ > \tau_\mathfrak{v} \wedge \tau_Z) G_{D_1}(\mathfrak{v},\mathfrak{v}) \frac{G_{D_1}(a,a)}{G_{D_1}(a,a)}\\ &= \mathbb{P}_a(\tau_\mathfrak{v} < \tau_Z) \frac{G_{D_1}(\mathfrak{v},\mathfrak{v})}{G_{D_1}(a,a)}.\end{aligned} \tag{4.5}$$

Now substitute (4.5) into (4.4) to get

$$\begin{aligned}&\mathbb{P}_w(\tau_{R_{KN}} < \tau_a < \tau_Z^+)\\ &= \mathbb{P}_w(\tau_{R_{KN}} < \tau_{Z'}^+) \sum_{\mathfrak{v}\in R_{KN}} \mathbb{P}_w(S_w(\tau_{R_{KN}}) = \mathfrak{v} | \tau_{R_{KN}} < \tau_{Z'}^+)\mathbb{P}_a(\tau_\mathfrak{v} < \tau_Z) \frac{G_{D_1}(\mathfrak{v},\mathfrak{v})}{G_{D_1}(a,a)}.\end{aligned} \tag{4.6}$$

Now we split the domain from R_{LN} to $R_{(K+L)N/2}$ into a sequence of balls and a set of the remaining space. The balls $B_1, B_2, \ldots B_n$, have radii $r_k = r(k, \pi_3)$ and are as described at the start of subsection 4.2.1. The maximum diameter of a ball is N so there is a decomposition with $n \geq c(K-L)$ balls for any possible domain D_1.

Decomposition of paths from R_1 to R_{KN} that do not intersect Z.
We want to use these balls to decompose the paths from a and w to R_{KN}. As any walk that hit Z can not satisfy the event it suffices to decompose the pairs of paths such that the path starting from a does not intersect Z before hitting R_{KN} and that, after time 1, the path from w does not hit Z' before hitting R_{KN}. We wish to decompose the paths in a way that will highlight when the two paths are both near to each other but more importantly are also away from the boundary. See Figure 4.2 for an example decomposition.

First consider the path from w until it first hits R_{KN} without hitting Z' after it takes its first step away from w. We will decompose this into a sequence of paths, $(\gamma_i)_{i\in\mathbb{N}}$, which in turn consists of three paths, $\gamma_i := \gamma_i^0 \cup \gamma_i^1 \cup \gamma_i^2$.

Define γ_1^0 to be the path from w until it first hits B_1, say at v_1. This means that B''_{1,γ_1^0} is defined. Define γ_1^1 to be the path from v_1 until the first time it hits B_1' or exits

B''_{1,γ_1^0}. Denote the vertex where γ_1^1 ends by w_1, then define γ_1^2 to be the path from w_1 until it first exits B''_{1,γ_1^0}. Set x_1 to be the end vertex of this path. Note this means that if γ_1^1 ended upon exiting B''_{1,γ_1^0} then $\gamma_1^2 = \{w_1\}$, i.e the walk takes no steps, and $x_1 = w_1$.

Now suppose we have constructed γ_{i-1} and it ended at x_{i-1}, then we inductively define γ_i for $i \leq n$.

Define γ_i^0 to be the path from x_{i-1} until the path first hits B_i, say at v_i. This means B''_{i,γ_i^0} is defined. Define γ_i^1 to be the path from v_i until it first hits B'_i or exits B''_{i,γ_i^0}. If γ_i^1 ends at w_i, then set γ_i^2 to be the path from w_i until it first exits B''_{i,γ_i^0}. Note this means that if γ_i^1 ended upon exiting B''_{i,γ_i^0} then $\gamma_i^2 = \{w_i\}$.

Having defined $\gamma_1, \dots \gamma_n$ we define a further couple of related paths.

Suppose γ_n ended at x_n then we define γ_{n+1} to be the path from x_n until it first reaches $R_{\frac{K+L}{2}N}$, say at x_{n+1}. Finally define γ_{n+2} to be the path from x_{n+1} until it first intersects R_{KN}.

Let the last vertex in γ_{n+2} be denoted by v. Now we similarly decompose the walk from a to v that does not intersect Z.

Define β_1^0 to be the path from a until it first hits B_1, say at y_1. This means that B''_{1,β_1^0} is defined. Set β_1^1 to be the path from y_1 until it first hits B'_1 or exits B''_{1,β_1^0}. Let the end vertex of β_1^1 be denoted a_1, then set β_1^2 to be the path from a_1 until it first exits B''_{1,β_1^0}, call the vertex it ends at u_1. Note this means that if β_1^1 ended upon exiting B''_{1,β_1^0} then $\beta_1^2 = \{a_1\}$, i.e it takes no steps, and $u_1 = a_1$.

Now suppose we have constructed β_{i-1}^2 and it ended at u_{i-1}, then we can inductively define β_i^j for $j = 0, 1, 2$ and $i \leq n$ as follows.

Define β_i^0 to be the path from u_{i-1} until the path first hits B_i, say at y_i. This means B''_{i,β_i^0} is defined. Define β_i^1 to be the path from y_i until it first hits B'_i or exits B''_{i,β_i^0}. Let the end vertex of β_i^1 be denoted a_i, then set β_i^2 to be the path from a_i until it first exits B''_{i,β_i^0}. Label the last vertex visited by this path as u_i. Note this means that if β_i^1 ended upon exiting B''_{i,β_i^0} then $\beta_i^2 = \{u_i\}$. Define $\beta_i := \beta_i^0 \cup \beta_i^1 \cup \beta_i^2$ for $1 \leq i \leq n$.

Suppose β_n ended at u_n then we define β_{n+1} to be the path from u_n until it first reaches $R_{\frac{K+L}{2}N}$, say at u_{n+1}. Finally define β_{n+2} to be the path from u_{n+1} until it first hits v.

Given a path α, with the number of steps used in α denoted $|\alpha|$, we define the associated weight of the path by

$$\omega(\alpha) := \left(\frac{1}{4}\right)^{|\alpha|}.$$

Now we use this decomposition of the paths and the same notation for end vertices to decompose the probabilities in (4.6). Firstly we can separate out the first part of

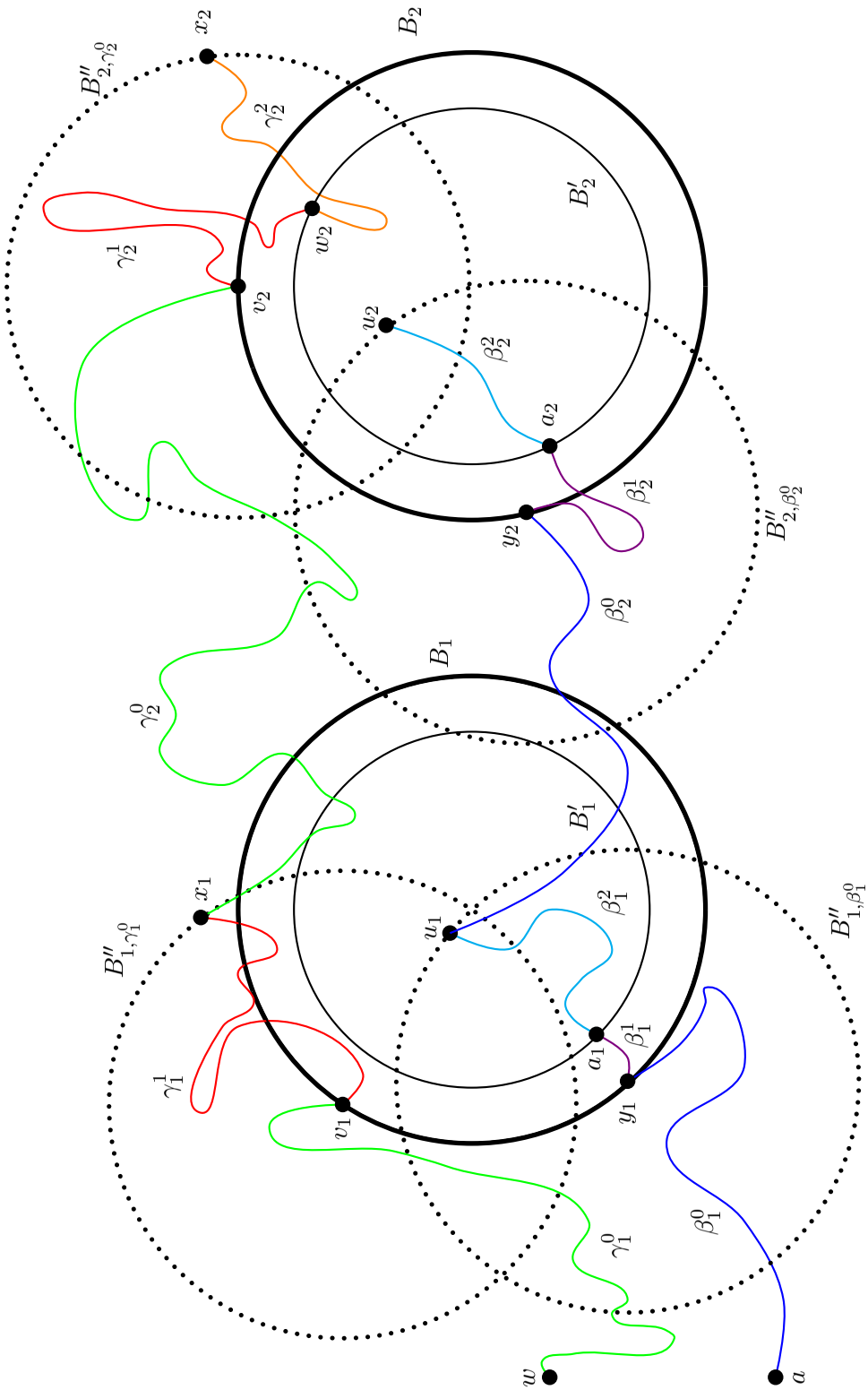

Figure 4.2: *An example of a potential decomposition of the paths over the first two balls. We have removed the boundary and previous paths to make the diagram clearer. Later in this subsection we define some sets E_j,F_j, in this example the pair of paths $(\gamma_1, \beta_1) \in F_1$ whilst $(\gamma_1 \cup \gamma_2, \beta_1 \cup \beta_2) \in E_2$.*

the walk.

$$\mathbb{P}_w(\tau_{R_{KN}} < \tau^+_{Z'}) = \sum_{\gamma_1} \omega(\gamma_1)\mathbb{P}_{x_1}(\tau_{R_{KN}} < \tau_{Z'}).$$

For clarity we are summing over the paths that start at w and exit B''_{1,γ_1^0} without returning to w or hitting any other vertex in Z'.

Observe we could repeat this process for each of the remaining paths in the decomposition to get

$$\mathbb{P}_w(\tau_{R_{KN}} < \tau^+_{Z'}) = \sum_{\gamma_1,\gamma_2\dots\gamma_{n+2}} \prod_{i=1}^{n+2} \omega(\gamma_i). \tag{4.7}$$

Then similarly for the walk from a we get

$$\mathbb{P}_a(\tau_v < \tau_Z) = \sum_{\beta_1,\dots\beta_{n+2}} \prod_{i=1}^{n+2} \omega(\beta_i).$$

Substituting these into (4.6), with v being the last vertex in γ_{n+2}, and noting that $\mathbb{P}_w(S_w(\tau_{R_{KN}}) = \mathfrak{v} | \tau_{R_{KN}} < \tau^+_{Z'})$ becomes the indicator function on the event $\mathfrak{v} = v$, we get

$$\mathbb{P}_w(\tau_{R_{KN}} < \tau_a < \tau^+_Z) = \sum_{\substack{\gamma_1,\gamma_2\dots\gamma_{n+2} \\ \beta_1,\dots\beta_{n+2}}} \prod_{i=1}^{n+2} \omega(\gamma_i)\omega(\beta_i)\frac{G_{D_1}(v,v)}{G_{D_1}(a,a)}. \tag{4.8}$$

We now want to use the decomposition to choose useful times to evaluate the probabilities, namely these times will be based on the collection of paths E_j and F, that we now define.

Define E_i to be the collection of pairs of paths that have the property that the two walks both end a section of their decomposition in B'_i but do not for any previous balls, and such that they do not intersect Z, Z' respectively. Formally, with w_k being the end point of γ^1_k and a_k the end point of β^1_k, set

$$E_i := \{(\cup_{j=1}^i \gamma_j, \cup_{j=1}^i \beta_j) : w_i, a_i \in B'_i, w_k \notin B'_k \text{ or } a_k \notin B'_k \ \forall k < i\}.$$

Alternatively given a random walk S_w that ends upon hitting a and satisfies $\tau_{B'_i} < \tau_a < \tau_Z$, we can ask if $S_w \in E_i$. This makes sense as (4.6) and the following decomposition give a method for producing the paths γ_j, β_j, from such a walk, and we can then ask if these uniquely determined paths satisfy E_i.

Note that E_i is therefore determined by the paths $\gamma_1, \dots \gamma_i$ and $\beta_1 \dots \beta_i$.

Define F to be the set of pairs of walks that reach $R_{(K+L)N/2}$ without having a section of their decomposition where they both end in a ball B'_j, i.e the paths that do

not satisfy the criteria for any E_j. Formally

$$F := \{(\cup_{k=1}^{n}\gamma_j, \cup_{k=1}^{n}\beta_n) : w_k \notin B'_k \text{ or } a_k \notin B'_k \; \forall k \leq n\}. \tag{4.9}$$

Thus in order for F to be satisfied the pair of paths (γ_k, β_k) must satisfy the corresponding event $F_k := \{\gamma_k \cap B'_k = \emptyset \text{ or } \beta_k \cap B'_k = \emptyset\}$, for all $k \leq n$. Hence $F = \bigcap_{k\leq n} F_k$. In the same way as we did for the set E_j we can extend this definition to the case of a single path between w and a if it reached $R_{(K+L)N/2}$.

It is worth noting that we could define these sets of paths in terms of a stopping time of the Markov chain consisting of the two random walks and an index variable to keep track of which part of the walk you are considering. We will not state this formally as we do not explicitly require this property.

If we wanted the event E_j or F to also occur we can rewrite (4.8) in the following ways

$$\mathbb{P}_w(\tau_{R_{KN}} < \tau_a < \tau_Z^+, E_j) = \sum_{\substack{\gamma_1,\gamma_2\ldots\gamma_j \\ \beta_1,\ldots\beta_j \\ E_j}} \sum_{\substack{\gamma_{j+1},\ldots\gamma_{n+2} \\ \beta_{j+1},\ldots\beta_{n+2}}} \prod_{i=1}^{n+2} \omega(\gamma_i)\omega(\beta_i)\frac{G_{D_1}(v,v)}{G_{D_1}(a,a)}, \tag{4.10}$$

$$\mathbb{P}_w(\tau_{R_{KN}} < \tau_a < \tau_Z^+, F) = \sum_{\substack{\gamma_1,\gamma_2\ldots\gamma_n \\ \beta_1,\ldots\beta_n \\ F}} \sum_{\substack{\gamma_{n+1},\gamma_{n+2} \\ \beta_{n+1},\beta_{n+2}}} \prod_{i=1}^{n+2} \omega(\gamma_i)\omega(\beta_i)\frac{G_{D_1}(v,v)}{G_{D_1}(a,a)}. \tag{4.11}$$

We now define some related terminology. Suppose instead of having a walk between w and a that visited R_{KN} we instead had a walk from w to a that hit B'_j. Then note we could use the same decomposition as above to split this walk into a set of pairs of paths $(\hat{\gamma}_i, \hat{\beta}_i)$ for $i \leq j$ and a walk between $\hat{w}_j$ and $\hat{a}_j$, where $\hat{w}_j$ is the end point of $\hat{\gamma}_j^1$ and $\hat{a}_j$ is the end point of $\hat{\beta}_j^1$.

Define the related collection of pairs of paths $\hat{E}_i$,

$$\hat{E}_i := \{(\cup_{j=1}^{i}\hat{\gamma}_j, \cup_{j=1}^{i}\hat{\beta}_j) : \hat{w}_i, \hat{a}_i \in B'_i,\ \hat{w}_k \notin B'_k \text{ or } \hat{a}_k \notin B'_k \; \forall k < i\}.$$

It is therefore not difficult to see that following the same argument, as for the case with a walk via R_{KN}, would yield a similar statement when we consider walks that satisfy $\hat{E}_j$.

$$P_w(\tau_{B'_j} < \tau_a < \tau_Z, \hat{E}_j) = \sum_{\substack{\hat{\gamma}_1\ldots\hat{\gamma}_j \\ \hat{\beta}_1\ldots\hat{\beta}_j \\ \hat{E}_j}} \prod_{i=1}^{j} \omega(\hat{\gamma}_i)\omega(\hat{\beta}_i)\mathbb{P}_{\hat{w}_j}(\tau_{\hat{a}_j} < \tau_{Z'})\frac{G_{D_1}(\hat{a}_j,\hat{a}_j)}{G_{D_1}(a,a)}. \tag{4.12}$$

Next we show that we can extract an exponential term by relaxing some of the walk's criteria.

Lemma 4.18. *For all $K, N > 0$ sufficiently large there exists constants $c_1, c_2 > 0$ such that*

$$\mathbb{P}_w(\tau_{R_{KN}} < \tau_a < \tau_Z^+, E_j) \leq c_1 \exp(-c_2(K-L))\mathbb{P}_w(\tau_a < \tau_Z^+, \hat{E}_j).$$

Lemma 4.19. *For all $K, N > 0$ sufficiently large there exists a constant $c > 0$ such that*

$$\mathbb{P}_w(\tau_{R_{KN}} < \tau_a < \tau_Z^+, F) \leq \exp(-c(K-L))\mathbb{P}_w(\tau_{R_{KN}} < \tau_a < \tau_Z^+).$$

Proof of Lemma 4.18. Starting from (4.10) we want to bound the paths that occur after the walk hits B'_j by an exponential term.

Notice that by combining the equivalent statements of (4.6) and (4.7) when we also require the paths to be in E_j, we can regroup some of the paths in the following manner,

$$\begin{aligned}
&\sum_{\substack{\gamma_1,\gamma_2\ldots\gamma_j \\ \beta_1,\ldots\beta_j \\ E_j}} \sum_{\substack{\gamma_{j+1},\ldots\gamma_{n+2} \\ \beta_{j+1},\ldots\beta_{n+2}}} \prod_{i=1}^{n+2} \omega(\gamma_i)\omega(\beta_i)\frac{G_{D_1}(v,v)}{G_{D_1}(a,a)} \\
&= \sum_{\substack{\gamma_1\ldots\gamma_j \\ \beta_1\ldots\beta_j \\ E_j}} \prod_{i=1}^{j} \omega(\gamma_i)\omega(\beta_i)\mathbb{P}_{w_j}(\tau_{R_{KN}} < \tau_{Z'}) \\
&\quad \times \sum_{\mathfrak{v}\in R_{KN}} \mathbb{P}_{w_j}(S_{w_j}(\tau_{R_{KN}}) = \mathfrak{v} | \tau_{R_{KN}} < \tau_{Z'})\mathbb{P}_{a_j}(\tau_{\mathfrak{v}} < \tau_Z)\frac{G_{D_1}(\mathfrak{v},\mathfrak{v})}{G_{D_1}(a,a)} \qquad (4.13)
\end{aligned}$$

Now we will do some manipulation with two of the terms in this expression using the time reversal of a random walk.

$$\begin{aligned}
&\mathbb{P}_{a_j}(\tau_{\mathfrak{v}} < \tau_Z)G_{D_1}(\mathfrak{v},\mathfrak{v}) \\
&= G_{D_1}(a_j,a_j)\mathbb{P}_{a_j}(\tau_{\mathfrak{v}} < \tau_Z | \tau_{a_j}^+ > \tau_{\mathfrak{v}} \wedge \tau_Z)G_{D_1}(\mathfrak{v},\mathfrak{v}) \\
&= G_{D_1}(a_j,a_j)\mathbb{P}_{\mathfrak{v}}(\tau_{a_j} < \tau_Z | \tau_{\mathfrak{v}}^+ > \tau_{a_j} \wedge \tau_Z)G_{D_1}(\mathfrak{v},\mathfrak{v}) \\
&= G_{D_1}(a_j,a_j)\mathbb{P}_{\mathfrak{v}}(\tau_{a_j} < \tau_Z).
\end{aligned}$$

Let $\mathfrak{B}$ be a ball of radius $r_k/8$ centred at a_j, or a constant radius if $r_k < 8$. Now

using Lemma 4.8 we can deduce that

$$\begin{aligned}
&G_{D_1}(a_j,a_j)\mathbb{P}_{\mathfrak{v}}(\tau_{a_j}<\tau_Z)\\
&=G_{D_1}(a_j,a_j)\mathbb{P}_{\mathfrak{v}}(\tau_{\mathfrak{B}}<\tau_Z)\sum_{x\in\partial_i\mathfrak{B}}\mathbb{P}_{\mathfrak{v}}(S_{\mathfrak{v}}(\tau_{\mathfrak{B}})=x|\tau_{\mathfrak{B}}<\tau_Z)\mathbb{P}_x(\tau_{a_j}<\tau_Z)\\
&\leq G_{D_1}(a_j,a_j)\mathbb{P}_{\mathfrak{v}}(\tau_{\mathfrak{B}}<\tau_Z)\sum_{x\in\partial_i\mathfrak{B}}\mathbb{P}_{\mathfrak{v}}(S(\tau_{\mathfrak{B}})=x|\tau_{\mathfrak{B}}<\tau_Z)c\mathbb{P}_{w_j}(\tau_{a_j}<\tau_Z)\\
&\leq c_1G_{D_1}(a_j,a_j)\mathbb{P}_{w_j}(\tau_{a_j}<\tau_{Z'})
\end{aligned}$$

Note in the final step we have used that adding a to the domain increases the random walks chance of hitting a_j before terminating thus switching Z with Z' respects the direction of the inequality.

Due to the properties we imposed on the choice of balls, $B_1,\dots B_n$, we know that each ball is between R_{LN} and $R_{(K+L)N/2}$ therefore the distance between w_j and R_{KN} is at least $c_3(K-L)N$, hence we can use Lemma 4.10 to deduce

$$\mathbb{P}_{w_j}(\tau_{R_{KN}}<\tau_{Z'})\leq\exp(-c_2(K-L)).$$

Now substitute these last two upper bounds into (4.13), to deduce

$$\begin{aligned}
&\mathbb{P}_w(\tau_{R_{KN}}<\tau_a<\tau_Z^+,E_j)\\
\leq&\sum_{\substack{\gamma_1\dots\gamma_j\\ \beta_1\dots\beta_j\\ E_j}}\prod_{i=1}^{j}\omega(\gamma_i)\omega(\beta_i)\exp(-c_2(K-L))\\
&\quad\times\sum_{\mathfrak{v}\in R_{KN}}\mathbb{P}_{w_j}(S_{w_j}(\tau_{R_{KN}})=\mathfrak{v}|\tau_{R_{KN}}<\tau_{Z'})c_1\mathbb{P}_{w_j}(\tau_{a_j}<\tau_{Z'})\frac{G_{D_1}(a_j,a_j)}{G_{D_1}(a,a)}\\
=&c_1\exp(-c_2(K-L))\sum_{\substack{\gamma_1\dots\gamma_j\\ \beta_1\dots\beta_j\\ E_j}}\prod_{i=1}^{j}\omega(\gamma_i)\omega(\beta_i)\mathbb{P}_{w_j}(\tau_{a_j}<\tau_{Z'})\frac{G_{D_1}(a_j,a_j)}{G_{D_1}(a,a)}.
\end{aligned}\tag{4.14}$$

Now we compare the terms in (4.14) to those in (4.12). Note that a set of j pairs of paths satisfies E_j if and only if it satisfies $\hat{E}_j$ and the paths would have the same weight in both settings, this allows us to see that

$$\sum_{\substack{\hat{\gamma}_1 \dots \hat{\gamma}_j \\ \hat{\beta}_1 \dots \hat{\beta}_j \\ \hat{E}_j}} \prod_{i=1}^{j} \omega(\hat{\gamma}_i)\omega(\hat{\beta}_i)\mathbb{P}_{\hat{w}_j}(\tau_{\hat{a}_j} < \tau_{Z'}) \frac{G_{D_1}(\hat{a}_j, \hat{a}_j)}{G_{D_1}(a,a)}$$

$$= \sum_{\substack{\gamma_1 \dots \gamma_j \\ \beta_1 \dots \beta_j \\ E_j}} \prod_{i=1}^{j} \omega(\gamma_i)\omega(\beta_i)\mathbb{P}_{w_j}(\tau_{a_j} < \tau_{Z'}) \frac{G_{D_1}(a_j, a_j)}{G_{D_1}(a,a)}.$$

Thus by using this equivalence we can substitute (4.12) into (4.14) to conclude that

$$\begin{aligned}
&\mathbb{P}_w(\tau_{R_{KN}} < \tau_a < \tau_Z^+, E_j) \\
\leq & c_1 \exp(-c_2(K-L)) \sum_{\substack{\hat{\gamma}_1 \dots \hat{\gamma}_j \\ \hat{\beta}_1 \dots \hat{\beta}_j \\ \hat{E}_j}} \prod_{i=1}^{j} \omega(\hat{\gamma}_i)\omega(\hat{\beta}_i)\mathbb{P}_{\hat{w}_j}(\tau_{\hat{a}_j} < \tau_{Z'}) \frac{G_{D_1}(\hat{a}_j, \hat{a}_j)}{G_{D_1}(a,a)} \\
\leq & c_1 \exp(-c_2(K-L))\mathbb{P}_w(\tau_{B'_j} < \tau_a < \tau_Z, \hat{E}_j) \\
\leq & c_4 \exp(-c_2(K-L))\mathbb{P}_w(\tau_a < \tau_Z, \hat{E}_j).
\end{aligned}$$

□

Now we need to consider the control we have over the event when F occurs instead of an E_j.

Proof of Lemma 4.19. By Remark 4.6 we are able to apply Lemma 4.7 to deduce that

$$\mathbb{P}_{v_i}(S(\xi_{B''_{k,\gamma_i^0}}) \in B'_k | \xi_{B''_{k,\gamma_i^0}} < \tau_{Z'}) \geq d$$
$$\mathbb{P}_{y_i}(S(\xi_{B''_{k,\beta_i^0}}) \in B'_k | \xi_{B''_{k,\beta_i^0}} < \tau_{Z}) \geq d.$$

Therefore, as the paths are independent, we have that for any $k \leq n$

$$\mathbb{P}(\gamma_k \cap B'_k = \emptyset \text{ or } \beta_k \cap B'_k = \emptyset | \gamma_k \cap Z' = \emptyset, \beta_k \cap Z = \emptyset) \leq 1 - d^2. \tag{4.15}$$

Next look at one pair of paths and show that we can separate out the event F_k occurring.

Hence, with v_k, respectively y_k, being the end vertex of γ_k^0, respectively β_k^0, we can

use (4.15), to get

$$\begin{aligned}
&\sum_{\substack{\gamma_k,\beta_k \\ F_k}} \omega(\gamma_k)\omega(\beta_k) \\
=&\sum_{\gamma_k^0,\beta_k^0} \omega(\gamma_k^0)\omega(\beta_k^0)\mathbb{P}_{v_k}(\xi_{B''_{k,\gamma_k^0}} < \tau_{Z'})\mathbb{P}_{y_k}(\xi_{B''_{k,\beta_k^0}} < \tau_Z) \\
&\quad \times \mathbb{P}(\gamma_k \cap B'_k = \emptyset \text{ or } \beta_k \cap B'_k = \emptyset | \gamma_k^0, \beta_k^0,\ \gamma_k \cap Z' = \emptyset, \beta_k \cap Z = \emptyset) \\
\leq&(1-d^2)\sum_{\gamma_k,\beta_k} \omega(\gamma_k)\omega(\beta_k). \qquad (4.16)
\end{aligned}$$

This is true for all pairs of paths so doing this simplification for each of the first n pairs of paths, (γ_k, β_k) yields

$$\begin{aligned}
\sum_{\substack{\gamma_1,\gamma_2\dots\gamma_n \\ \beta_1,\dots\beta_n \\ F}} \prod_{i=1}^{n} \omega(\gamma_i)\omega(\beta_i) \leq&(1-d^2)^n \sum_{\substack{\gamma_1,\gamma_2\dots\gamma_n \\ \beta_1,\dots\beta_n}} \prod_{i=1}^{n} \omega(\gamma_i)\omega(\beta_i) \\
&\leq \exp(-c(K-L)) \sum_{\substack{\gamma_1,\gamma_2\dots\gamma_n \\ \beta_1,\dots\beta_n}} \prod_{i=1}^{n} \omega(\gamma_i)\omega(\beta_i)
\end{aligned}$$

as $n > c_1(K-L)$ by the choice of balls we used to decompose the domain.

Now recall (4.11), where v was the last vertex in γ_{n+2}, and substitute the previous expression into it.

$$\begin{aligned}
&\mathbb{P}_w(\tau_{R_{KN}} < \tau_a < \tau_Z^+, F) \\
=&\sum_{\substack{\gamma_1,\gamma_2\dots\gamma_n \\ \beta_1,\dots\beta_n \\ F}} \sum_{\substack{\gamma_{n+1},\gamma_{n+2} \\ \beta_{n+1},\beta_{n+2}}} \prod_{i=1}^{n+2} \omega(\gamma_i)\omega(\beta_i)\frac{G_{D_1}(v,v)}{G_{D_1}(a,a)} \\
\leq&\exp(-c(K-L)) \sum_{\substack{\gamma_1,\gamma_2\dots\gamma_n \\ \beta_1,\dots\beta_n}} \prod_{i=1}^{n} \omega(\gamma_i)\omega(\beta_i)) \sum_{\substack{\gamma_{n+1},\gamma_{n+2} \\ \beta_{n+1},\beta_{n+2}}} \prod_{i=n+1}^{n+2} \omega(\gamma_i)\omega(\beta_i)\frac{G_{D_1}(v,v)}{G_{D_1}(a,a)}
\end{aligned}$$

Comparing this to (4.8) we can conclude that

$$\mathbb{P}_w(\tau_{R_{KN}} < \tau_a < \tau_Z^+, F) \leq \exp(-c(K-L))\mathbb{P}_w(\tau_{R_{KN}} < \tau_a < \tau_Z^+) \qquad (4.17)$$

as required. □

We can now recall (4.3) and conclude the proof of Proposition 4.17 by utilising the Lemmas 4.18 and 4.19, to derive the desired result.

$$
\begin{aligned}
&\mathbb{P}_w(\tau_{R_{KN}} < \tau_a | \tau_a < \tau_Z^+) \\
&= \frac{\mathbb{P}_w(\tau_{R_{KN}} < \tau_a < \tau_Z^+)}{\mathbb{P}_w(\tau_a < \tau_Z^+)} \\
&= \frac{\sum_{i=1}^n \mathbb{P}_w(\tau_{R_{KN}} < \tau_a < \tau_Z^+, E_i) + \mathbb{P}_w(\tau_{R_{KN}} < \tau_a < \tau_Z^+, F)}{\mathbb{P}_w(\tau_a < \tau_Z^+)} \\
&\leq \frac{\sum_{i=1}^n c_1 \exp(-c_2(K-L)) \mathbb{P}_w(\tau_a < \tau_Z^+, \hat{E}_i)}{\mathbb{P}_w(\tau_a < \tau_Z^+)} \\
&\quad + \frac{\exp(-c_3(K-L)) \mathbb{P}_w(\tau_{R_{KN}} < \tau_a < \tau_Z^+)}{\mathbb{P}_w(\tau_a < \tau_Z^+)} \\
&\leq c_1 \exp(-c_2(K-L)) \frac{\mathbb{P}_w(\tau_a < \tau_Z^+, \bigcup_{i=1}^n \hat{E}_i)}{\mathbb{P}_w(\tau_a < \tau_Z^+)} + \exp(-c_3(K-L)) \frac{\mathbb{P}_w(\tau_a < \tau_Z^+)}{\mathbb{P}_w(\tau_a < \tau_Z^+)} \\
&\leq c_1 \exp(-c_2(K-L)) \frac{\mathbb{P}_w(\tau_a < \tau_Z^+)}{\mathbb{P}_w(\tau_a < \tau_Z^+)} + \exp(-c_3(K-L)) \\
&\leq c_4 \exp(-c_5(K-L)).
\end{aligned}
$$

□

4.4.2 Bound for adjacent paths.

Now we have control over nested paths in this subsection we turn our attention to adjacent paths.

Proposition 4.20. *Let the union of the set of some paths between vertices in R_1, denoted π, be such that they do not disconnect a and w from each other or the boundary of the graph. Moreover let rung LN be the smallest such that $R_{LN} \cap \pi = \emptyset$. Take $Z := \pi \cup R_1 \setminus \{a\}$. Then for sufficiently large N,*

$$\mathbb{P}_w(\textit{LERW to } a \textit{ intersects } R_{KN} | \tau_a < \tau_Z^+) \leq c_1 \exp(-c_2(K-L)).$$

Proof. The strategy is to show that it is unlikely for the parts of the walk that intersect R_{KN} to not be contained in loops.

Let $\big(S(n)\big)_{n\in\mathbb{N}}$ be a random walk started at w that terminates upon first hitting a or returning to Z. Set $M_1 := (L+2K)/3$ and $M_2 := (2L+K)/3$ and define three probability measures for $y \in R_{KN}$, $x \in R_{M_1 N}$ and a path α.

$$
\begin{aligned}
\sigma_1(y) &:= \mathbb{P}_w(\text{the last vertex in } R_{KN} \text{ visited by } \big(S(n)\big)_{n\in\mathbb{N}} \text{ is } y | \tau_{R_{KN}} < \tau_a < \tau_Z^+) \\
\sigma_2(x,y,\alpha) &:= \mathbb{P}_y(S_y(\tau_{R_{M_1 N}}) = x | \tau_{R_{M_1 N}} < \tau_\alpha^+, \tau_a < \tau_Z \wedge \tau_{R_{KN}}^+) \\
\sigma_3(u,x,\alpha) &:= \mathbb{P}_x(S_x(\tau_{R_{M_2 N}}) = u | \tau_{R_{M_2 N}} < \tau_\alpha).
\end{aligned}
$$

Firstly note that a and Z can not be in a loop as the walk would stop upon hitting them so their LERW hitting time is the same as the RW hitting time.

Next we will assume that the random walk reaches R_{KN} and will condition upon y being the last vertex in R_{KN} that the random walk visited.

Define Υ to be the loop erasure of the random walk up until the final time the random walk was at y, given that the walk hits R_{KN} before $Z \cup \{a\}$ and will then reach a before hitting Z.

As y was conditioned to be the last vertex visited by the random walk in R_{KN}, not returning to y is equivalent to not returning to R_{KN}. Introduce the notation $\hat{\tau}$ for the hitting time of a loop-erased path starting from w and terminating at a, so the event $\hat{\tau}_{R_{KN}} < \hat{\tau}_a$ is that the loop erased path from w to a hits R_{KN}. Then we have the expression

$$\begin{aligned}&\mathbb{P}_w(\hat{\tau}_{R_{KN}} < \hat{\tau}_a | \tau_{R_{KN}} < \tau_a < \tau_Z^+, \text{ last vertex visited in } R_{KN} \text{ is } y)\\ =&\sum_{\alpha\text{: path from } w \text{ to } y} \mathbb{P}(\Upsilon = \alpha)\mathbb{P}_y(\tau_a < \tau_\alpha^+ | \tau_a < \tau_Z \wedge \tau_{R_{KN}}^+).\end{aligned}$$

Therefore

$$\begin{aligned}&\mathbb{P}_w(\hat{\tau}_{R_{KN}} < \hat{\tau}_a | \tau_a < \tau_Z^+)\\ =&\mathbb{P}_w(\tau_{R_{KN}} < \tau_a | \tau_a < \tau_Z^+) \sum_{y \in R_{KN}} \sigma_1(y)\mathbb{P}_y(\hat{\tau}_{R_{KN}} < \hat{\tau}_a | \tau_a < \tau_Z, \text{last visit to } R_{KN} \text{ at } y)\\ \leq&\sum_{y \in R_{KN}} \sigma_1(y) \sum_\alpha \mathbb{P}(\Upsilon = \alpha)\mathbb{P}_y(\tau_a < \tau_\alpha^+ | \tau_a < \tau_Z \wedge \tau_{R_{KN}}^+)\\ =&\sum_{y \in R_{KN}} \sigma_1(y) \sum_\alpha \mathbb{P}(\Upsilon = \alpha)\mathbb{P}_y(\tau_{R_{M_1N}} < \tau_\alpha^+ | \tau_a < \tau_Z \wedge \tau_{R_{KN}}^+)\\ &\times \sum_{x \in R_{M_1N}} \sigma_2(x, y, \alpha)\mathbb{P}_x(\tau_a < \tau_\alpha | \tau_a < \tau_Z \wedge \tau_{R_{KN}})\\ \leq&\sum_{y \in R_{KN}} \sigma_1(y) \sum_\alpha \mathbb{P}(\Upsilon = \alpha) \sum_{x \in R_{M_1N}} \sigma_2(x, y, \alpha) \sum_{u \in R_{M_2N}} \Big(\\ &\mathbb{P}_x(\tau_{R_{M_2N}} < \tau_\alpha, S_x(\tau_{R_{M_2N}}) = u | \tau_a < \tau_Z \wedge \tau_{R_{KN}})\mathbb{P}_u(\tau_a < \tau_\alpha | \tau_a < \tau_Z \wedge \tau_{R_{KN}})\Big)\\ \leq&\sum_{y \in R_{KN}} \sum_\alpha \sum_{x \in R_{M_1N}} \sum_{u \in R_{M_2N}} \sigma_1(y)\mathbb{P}(\Upsilon = \alpha)\sigma_2(x, y, \alpha)\\ &\times \mathbb{P}_x(\tau_{R_{M_2N}} < \tau_\alpha, S_x(\tau_{R_{M_2N}}) = u | \tau_a < \tau_Z \wedge \tau_{R_{KN}}). \qquad (4.18)\end{aligned}$$

Define the harmonic function $h(v) := \mathbb{P}_v(\tau_a < \tau_Z \wedge \tau_{R_{KN}})$ which vanishes outside of the domain. Note that any vertices $u \in R_{M_2N}$ and $x \in R_{M_1N}$ are at least c_1N from a, Z and R_{KN}. Therefore the Harnack inequality holds and we can deduce that $h(u) \leq c_2h(x)$. Due to the choice of the function h we also have that the conditions of a Doob h-transform are satisfied so we can deduce that

$$\begin{aligned}
&\mathbb{P}_x(\tau_{R_{M_2N}} < \tau_\alpha, S(\tau_{R_{M_2N}}) = u | \tau_a < \tau_Z \wedge \tau_{R_{KN}}) \\
&= \mathbb{P}_x\Big(\tau_{R_{M_2N}} < \tau_\alpha, S(\tau_{R_{M_2N}}) = u\Big)\frac{h(u)}{h(x)} \\
&\leq c_2 \mathbb{P}_x\Big(\tau_{R_{M_2N}} < \tau_\alpha, S(\tau_{R_{M_2N}}) = u\Big) \\
&= c_2\sigma_3(u,x,\alpha)\mathbb{P}_x(\tau_{R_{M_2N}} < \tau_\alpha) \qquad (4.19)
\end{aligned}$$

Substituting (4.19) into (4.18) gives

$$\begin{aligned}
&\mathbb{P}_w(\hat{\tau}_{R_{KN}} < \hat{\tau}_a | \tau_a < \tau_Z^+) \\
&\leq \sum_{y\in R_{KN}} \sum_{\alpha} \sum_{x\in R_{M_1N}} \sum_{u\in R_{M_2N}} c_2\sigma_1(y)\mathbb{P}(\Upsilon = \alpha)\sigma_2(x,y,\alpha)\sigma_3(u,x,\alpha)\mathbb{P}_x(\tau_{R_{M_2N}} < \tau_\alpha)
\end{aligned}$$

Finally by Lemma 4.10,

$$\mathbb{P}_x(\tau_{R_{M_2N}} < \tau_\alpha) \leq c_3 \exp(-c_4(M_2 - M_1)) = c_3 \exp(-c_5(K-L)),$$

although the domain on which this random walk is terminated is different from the one stated in Lemma 4.10 it is clear that the proof for the result in this setting is equivalent and so the same bound also holds in this case. This allows us to conclude that

$$\begin{aligned}
&\mathbb{P}_w(\hat{\tau}_{R_{KN}} < \hat{\tau}_a | \tau_a < \tau_Z^+) \\
&\leq \sum_{y\in R_{KN}} \sum_{\alpha} \sum_{x\in R_{M_1N}} \sum_{u\in R_{M_1N}} c_2\sigma_1(y)\mathbb{P}(\Upsilon = \alpha)\sigma_2(x,y,\alpha)\sigma_3(u,x,\alpha)c_3 \exp(-c_5(K-L)) \\
&\leq c_6 \exp(-c_5(K-L)).
\end{aligned}$$

□

4.4.3 Bound for a grove.

Now we have the individual bounds we will show that we can take any combination of these paths and associated bounds to produce a uniform bound for the whole grove.

Lemma 4.21. *Suppose we have a partition of a grove consisting of m components with each component being a path nested inside of the previous one. If we uniformly choose a grove that satisfies this connection pattern then look at the furthest rung that any of the nested paths in the grove reached then we can say that*

$$\mathbb{P}(\textit{a nested paths crosses } R_{(K+m)N}) \leq \exp\left(-c_1K + c_2m\right).$$

Proof. Suppose we have a grove selected uniformly from groves that induce the desired partition on R_1 . Recall if we resample a component then the resulting grove is also

uniform. Therefore we can resample each path in turn, working from the inside out, extracting the relevant information at each step allowing us to check if the final path reaches $R_{(K+m)N}$.

Let $\mathfrak{F}_i$ contain the information of how the first i paths have been resampled. Let X_i be the distance reached by the i^{th} path beyond the furthest rung that the $i-1^{th}$ path hit upon resampling conditioned on $\mathfrak{F}_i$. Hence the furthest rung reached by the sequence of nested paths after resampling will be $\sum_{i=1}^{m} X_i$ since after resampling the grove has the same distribution as the original grove it suffices to provide a bound for this term.

The behaviour of X_i conditioned on $\mathfrak{F}_{i-1}$, for $i < m$ is bounded by Proposition 4.17 where π_1 is taken to be the $i-1^{th}$ resampled path and π_3 is some path that has not been resampled yet. For $x \in \mathbb{R}^+$ this Proposition will give us a bound for the probability that $X_i - N \geq x$. Note that the criteria for the Proposition to be applied require us to be asking about distances of at least N and hence we will work with $X_i - N$.

For the m^{th} path we need to use Propostion 4.20 instead as there will not be a path that disconnects it from the boundary to play the role of π_3.

Note we can find two constants $c_1, c_2 > 0$ such that the upper bound in Proposition 4.17 and Propostion 4.20 are the same. Hence for all i

$$\mathbb{P}(X_i > x + N | \mathfrak{F}_{i-1}) \leq c_1 \exp\left(\frac{-c_2(x+N)}{N}\right).$$

Introduce the new sigma algebra, $\mathfrak{G}_i$, which contains the information about the furthest rung reached by each of the previous i paths. In the proofs of Proposition 4.17 and Propostion 4.20 it is only $\mathfrak{G}_i$ rather than $\mathfrak{F}_i$ that we need to know. Hence the same bound holds when we condition on $\mathfrak{G}_i$ instead of $\mathfrak{F}_i$.

Introduce the i.i.d geometric random variables Z_i with parameter $1 - \exp\left(\frac{-c_2}{N}\right)$, which are independent of $\mathfrak{G}_m$. This means that

$$\mathbb{P}(X_i > x + N | \mathfrak{G}_{i-1}) \leq c_1 \exp\left(\frac{-c_2(x+N)}{N}\right) \leq c_1 \exp\left(\frac{-c_2 x}{N}\right) = c_1 \mathbb{P}(Z_i > x).$$

Hence

$$\begin{aligned}
&\mathbb{P}\Big(\sum_{i=1}^{m}(X_i - N) > KN\Big) \\
&= \sum_{\substack{x_1,\dots x_m\in\mathbb{Z}^+ \\ \sum_{i=1}^m x_i \geq KN}} \mathbb{P}(X_1 - N = x_1|\mathfrak{G}_0)\mathbb{P}(X_2 - N = x_2|\mathfrak{G}_1)\dots\mathbb{P}(X_m - N = x_m|\mathfrak{G}_{m-1}) \\
&= \sum_{\varsigma=0}^{\infty} \sum_{\substack{x_1,\dots x_{m-1}\in\mathbb{Z}^+ \\ \sum_{i=1}^{m-1} x_i = \varsigma}} \mathbb{P}(X_1 - N = x_1|\mathfrak{G}_0)\dots\mathbb{P}(X_{m-1} - N = x_{m-1}|\mathfrak{G}_{m-2}) \\
&\quad \times \sum_{\substack{x_m\in\mathbb{Z}^+ \\ x_m \geq KN-\varsigma}} \mathbb{P}(X_m - N = x_m|\mathfrak{G}_{m-1}) \\
&= \sum_{\varsigma=0}^{\infty} \sum_{\substack{x_1,\dots x_{m-1}\in\mathbb{Z}^+ \\ \sum_{i=1}^{m-1} x_i = \varsigma}} \mathbb{P}(X_1 - N = x_1|\mathfrak{G}_0)\dots\mathbb{P}(X_{m-1} - N = x_{m-1}|\mathfrak{G}_{m-2}) \\
&\quad \times \mathbb{P}(X_m - N \geq KN - \varsigma|\mathfrak{G}_{m-1}) \\
&\leq \sum_{\varsigma=0}^{\infty} \sum_{\substack{x_1,\dots x_{m-1}\in\mathbb{Z}^+ \\ \sum_{i=1}^{m-1} x_i = \varsigma}} \mathbb{P}(X_1 - N = x_1|\mathfrak{G}_0)\dots\mathbb{P}(X_{m-1} - N = x_{m-1}|\mathfrak{G}_{m-2}) \\
&\quad \times c_1\mathbb{P}(Z_m \geq KN - \varsigma) \\
&= \sum_{\varsigma=0}^{\infty} \sum_{\substack{x_1,\dots x_{m-1}\in\mathbb{Z}^+ \\ \sum_{i=1}^{m-1} x_i = \varsigma}} \mathbb{P}(X_1 - N = x_1|\mathfrak{G}_0)\dots\mathbb{P}(X_{m-1} - N = x_{m-1}|\mathfrak{G}_{m-2}) \\
&\quad \times \sum_{\substack{x_m\in\mathbb{Z}^+ \\ x_m \geq KN-\varsigma}} c_1\mathbb{P}(Z_m = x_m) \\
&= c_1 \sum_{\substack{x_1,\dots x_m\in\mathbb{Z}^+ \\ \sum_{i=1}^m x_i \geq KN}} \mathbb{P}(X_1 - N = x_1|\mathfrak{G}_0)\dots\mathbb{P}(X_{m-1} - N = x_{m-1}|\mathfrak{G}_{m-2})\mathbb{P}(Z_m = x_m).
\end{aligned}$$

As Z_m is independent of $X_1, \dots X_m$ there is no reason we now can not repeat this argument for X_{m-1}. Repeating for each X_i in turn allows us to deduce that

$$\begin{aligned}
\mathbb{P}\Big(\sum_{i=1}^{m}(X_i - N) > KN\Big) &\leq \sum_{\substack{x_1,\dots x_m\in\mathbb{Z}^+ \\ \sum_{i=1}^m x_i \geq KN}} c_1\mathbb{P}(Z_1 = x_1)c_1\mathbb{P}(Z_2 = x_2)\dots c_1\mathbb{P}(Z_m = x_m) \\
&= c_1^m\mathbb{P}(\sum_{i=1}^{m} Z_m \geq KN).
\end{aligned}$$

Next we use Markov's inequality, then insisting that $t \leq \frac{c_2}{N}$ we can use the moment

generating function of a geometric random variable and then finally take $t = \frac{c_2}{2N}$ to get

$$\begin{aligned}
&\mathbb{P}\Big(\sum_{i=1}^{m}(X_i - N) > KN\Big) \\
&\leq c_1^m \mathbb{P}\Big(\sum_{i=1}^{m} Z_i > KN\Big) \\
&\leq \min_{0<t<\frac{c_2}{N}} c_1^m \mathbb{P}\Big(\exp\Big(\sum_{i=1}^{m} Z_i t\Big) \geq \exp(KNt)\Big) \\
&\leq \min_{0<t<\frac{c_2}{N}} c_1^m \frac{\mathbb{E}[\exp(t\sum_{i=1}^{m} Z_i)]}{\exp(KNt)} \\
&= \min_{0<t<\frac{c_2}{N}} c_1^m \frac{\prod_{i=1}^{m} \mathbb{E}[\exp(Z_i t)]}{\exp(KNt)} \\
&\leq \min_{0<t<\frac{c_2}{N}} c_1^m \exp(-KNt) \left(\frac{(1-\exp(\frac{-c_2}{N}))\exp(t)}{1-(1-(1-\exp(\frac{-c_2}{N}))\exp(t)}\right)^m \\
&\leq \min_{0<t<\frac{c_2}{N}} c_1^m \exp(-KN\frac{c_2}{2N}) \left(\frac{(1-\exp(\frac{-c_2}{N}))\exp(\frac{c_2}{2N})}{1-(1-(1-\exp(\frac{-c_2}{N}))\exp(\frac{c_2}{2N})}\right)^m \\
&\leq c_1^m \exp(-c_3 K) \left(\frac{\exp(\frac{c_2}{2N}) - \exp(\frac{-c_2}{2N})}{1-\exp(\frac{-c_2}{2N})}\right)^m \\
&= c_1^m \exp(-c_3 K) \left(1 + \exp\left(\frac{c_2}{2N}\right)\right)^m \\
&\leq c_1^m \exp(-c_3 K) c_4^m \\
&\leq \exp(-c_3 K + c_5 m).
\end{aligned}$$

□

Lemma 4.22. *Suppose we have a grove with respect to R_1 that is uniformly chosen from the set of all groves that consist of m adjacent paths. For $K > 1$,*

$$\mathbb{P}(\textit{One of the } m \textit{ adjacent paths reaches } R_{KN}) \leq c_1 m \exp(-c_2 K).$$

Proof. This will be shown by resampling each path in turn and checking if any of the paths reached R_{KN}, as noted early the resulting grove will have the same distribution as the original grove chosen. Suppose that we are resampling the path between w and a and that the union of all the other adjacent paths connected to R_1, whether resampled or original, is denoted by π. We assume that the furthest rung that π reaches is R_{MN}. Then depending on the size of M we can use our previous results to handle this case.

If $M > 1 \vee K/2$, then observe that until the walk reaches R_{MN} we are in the same situation as Proposition 4.17. Whilst in the set up of that Proposition we have a path between R_1 and R_{MN} on either side in this setting we can use the paths in π to play the same role.

This is allowed because the only relevant information we need to extract from the

enclosing path in the proof of the proposition was that we could decompose the domain using balls which have a maximum allowed radius and that the probability of a walk between two rungs to avoid the paths was exponentially bounded. Therefore if we have a path either side we have exactly the same situation. Moreover even if we had a path on only one side the same proof works, because the decomposition would be the same but using the boundary to control the size of the ball and an equivalent statement to Lemma 4.9 exists when we only have one path.

More formally let $Z := \pi \cup R_1$ and recall the notation that $\hat{\tau}_{R_{KN}} < \hat{\tau}_a$ denotes the event of a LERW started from w that is stopped at a intersecting R_{KN}. As there is no path nested inside we can take $L = 0$ and conclude that there exists constants $d_1, d_2 > 0$ such that

$$\mathbb{P}_w(\hat{\tau}_{R_{MN}} < \hat{\tau}_a | \tau_a < \tau_Z^+) \leq d_1 \exp(-cM) \leq d_1 \exp(-d_2 K).$$

Once the walk passes R_{MN} then there would be no other paths outside it as all previous paths would be behind R_{MN} and we our interested if the path manages to cross R_{KN}. This is therefore the same set up as we have in Proposition 4.20 with $M = L$. Observe that for the cases $M < 1$ or $M < K/2$ we have $K - M \geq cK$. Therefore we can conclude that there exists constants $d_3, d_4, d_5 > 0$ such that

$$\mathbb{P}_w(\hat{\tau}_{R_{KN}} < \hat{\tau}_a | \tau_a < \tau_Z^+) \leq d_3 \exp(-d_4(K - M)) \leq d_3 \exp(-d_5 K).$$

Finally we note that by taking the constants $c_1 = \max\{d_1, d_3\}$ and $c_2 = \min\{d_2, d_5\}$ we can get an upper bound that holds in both settings. Therefore each path has the same bound for reaching R_{KN} and it is independent of the other paths in the backbone, thus

$$\begin{aligned}\mathbb{P}(\text{One of the } m \text{ adjacent paths reaches } R_{KN}) &\leq \sum_{i=1}^{m} \mathbb{P}_w(\hat{\tau}_{R_{KN}} < \hat{\tau}_a | \tau_a < \tau_Z^+ \wedge \tau_\alpha) \\ &\leq c_1 m \exp(-c_2 K).\end{aligned}$$

□

Remark 4.23. *The bound that this yields is significantly stronger than the final bound that we are able to prove. This offers hope that if in future work we can provide control of how many nested paths are likely to exist in a grove we will be able to make a beneficial improvement to the overall bound.*

Definition 4.24. *For a partition $\mathfrak{p}$ of R_1, define $\mathbb{P}_\mathfrak{p}$ to be the uniform probability measure on groves with respect to R_1 which have induced the partition $\mathfrak{p}$ on R_1.*

Lemma 4.25. *Let $\mathfrak{b}_\mathfrak{g}$ be the backbone of a uniformly chosen grove with respect to R_1 and let $\mathfrak{g}_\mathfrak{c}$ be one of its components. Suppose $\mathfrak{g}_\mathfrak{c} \cap R_1 = \{v_1, v_2, \dots v_n\}$, where the vertices are ordered according to their vertical co-ordinate. The path in $\mathfrak{b}_\mathfrak{g}$ that connects v_1 to*

v_n has the distribution of a loop-erased random walk started at v_1 conditioned to hit v_n before hitting hitting a vertex in $\mathfrak{b}_g \setminus \mathfrak{g}_\mathfrak{c}$.

Proof. Firstly note to generate a uniformly distributed rooted groves we could first generate a uniformly distributed grove then uniformly choose the roots in each component.

Now recall the algorithm for generating groves conditioned on the partition induced on R_1 that we introduced in Chapter 2. The first step in the algorithm was to choose the roots in each component. We know that in Wilson's algorithm the choices we make do not effect the distribution of the spanning tree that is outputted and hence we are free to chose different roots in the components without changing the distribution of the undirected grove created.

Therefore given the partition $\{v_1, \dots v_n\}$ we are free to generate the attached backbone component with v_1 being the root. Moreover we can let v_n be the first vertex we start a LERW from. Thus the path from v_n to v_1 will have the stated distribution. □

Remark 4.26. *The path from v_1 to v_n will encompass the other paths in the backbone of $\mathfrak{g}_\mathfrak{c}$, hence if any path in $\mathfrak{g}_\mathfrak{c}$ crosses R_{KN} so must this extremal path.*

Lemma 4.27. *Suppose we have a partition $\mathfrak{p}$ of R_1 and let $\mathfrak{b}_\mathfrak{g}$ be a backbone of a uniformly selected grove with respect to R_1. Suppose that $\mathfrak{b}_\mathfrak{g}$ consists of m components with m' of them containing at least two vertices.*

Then

$$\mathbb{P}_\mathfrak{p}(\mathfrak{b}_\mathfrak{g} \cap R_{(K+m')N} \neq \emptyset) \leq c_1 \exp(-c_2 K + c_3 m'). \tag{4.20}$$

Proof. This will be proven by showing that the upper bound for the partition with all nested paths will hold for any other configuration of components.

There is one type of component that can occur in a partition which we have not yet considered and that is one which contains a solitary vertex.

The solitary vertices, by definition, can not have a path that returns to R_1 and so they can not make the event in the left-hand side of (4.20) occur. Moreover by Lemma 2.7 we know that in the resampling process we only condition on the other paths in the backbone and thus not the paths connected to solitary vertices. Therefore we would only generate the paths that attached to the solitary vertices after the other paths have been generated at which point the occurrence of the event will have already been determined. Therefore solitary vertices only influence is through the requirement that the paths in $\mathfrak{b}_\mathfrak{g}$ must avoid them.

Therefore there are m' components that we need to consider further.

Observe that in the proofs of Proposition 4.17, Proposition 4.20, Lemma 4.21 and Lemma 4.22, the dependence on the other components in the backbone comes only through the furthest rung that they reached. Thus the only information we need from each backbone component is the furthest rung that it intersects. Also note that, in

both of these Lemmas, increasing the number of components monotonically increases the upper bound produced.

If a component of $\mathfrak{p}$, $\mathfrak{p}_i$ had more than two vertices then we can apply Lemma 4.25. Suppose $\mathfrak{p}_i = \{v_1, v_2, \dots v_n\}$, with the vertices ordered according to their vertical co-ordinate. Then the furthest rung reached by the backbone attached to this component is the same as the furthest rung that the path from v_1 to v_n intersects. This has the distribution of a LERW path conditioned to avoid the other backbone components.

Recall with π_1, π_3 being the relevant nesting paths, if they exist, from the other backbone components, the proofs of Propositions 4.17 and 4.20 used $Z = \pi_1 \cup \pi_3 \cup (R_1 \setminus \{v_n\})$. Notice that the same proofs would hold if instead we had taken $Z = \pi_1 \cup \pi_3 \cup (R_1 \setminus \{v_2, \dots v_n\})$ and $Z' = \pi_1 \cup \pi_3 \cup (R_1 \setminus \{v_2, \dots v_{n-1}\})$. This would allow the LERW from v_1 to v_n to return to vertices in $\mathfrak{p}_i$, whilst still yielding the same upper bound.

These walks would therefore have the same distribution as the LERW we want to run from v_1 to v_n conditioned to avoid other backbone components. Therefore the upper bounds in Propositions 4.17 and 4.20 also hold for the distance reached by the backbone connected to $\mathfrak{p}_i$.

This means that if $\mathfrak{p}_i$ has more than two vertices in it then the exact number of vertices contained in $\mathfrak{p}_i$ does not effect our upper bound for the furthest rung reached by the associated backbone component.

Hence once we know there are m' components of the partition with at least two vertices the only further information required to utilise the results of the upper bounds we have thus far produced, is to know how the end vertices of each component are arranged in the partition with respect to each other.

There are two distinct ways that two components of a grove can be in relation to each other, either they are adjacent or they are nested. We can use Lemma 4.22 to bound the probability of any of m' successive adjacent components reaching $R_{(K+m')N}$ by $c_1 m' \exp(-c_2(K + m'))$.

Next for the case of a grove having m' nested components, we have the bound from Lemma 4.21 for a path in the backbone reaching $R_{(K+m')N}$ of $\exp(-c_3 K + c_4 m')$. By comparing these bounds, and taking $c_3 > 0$ sufficiently small and $c_4 > 0$ sufficiently large, we see that for all valid K, N, m' our weaker bound is achieved by taking all components nested over all adjacent components.

The proof is concluded by showing that taking any other possible combination of nested and adjacent components does not increase the bound given by only having nested components.

Firstly consider having two adjacent components which are nested inside the same outer component. Observe that the bound for the outer nested component depends only on the adjacent components through the furthest rung that was reached by either of the inner components. Therefore if we could replace the two adjacent components with components that had a greater probability of travelling further then the backbone as a

whole would be more likely to reach a higher rung. As we previous observed, replacing two adjacent components with two nested components will increase the bound for the distance reached. Hence the case of having adjacent components in one step of a sequence of nested components is bounded above by the case where all components are in one nested sequence.

The other example to consider is that of having two nested components being adjacent to each other. Suppose the adjacent components had m_1 and m_2 components in respectively. Then by Lemmas 4.21 and 4.22 the probability of either component reaching $R_{(K+m_1+m_2)N}$ is

$$c_5 \exp(-c_6(K+m_2)+c_7 m_1) + c_5 \exp(-c_6(K+m_1)+c_7 m_2).$$

By insisting that the constants are sufficiently large, or small respectively, this is less than $c_5 \exp(-c_6 K + c_7(m_1+m_2))$ for all K, N, m_1, m_2, which we would obtain from having all paths in one nested sequence.

From these cases any other combination of adjacent and nested components will clearly also be bounded by the case of all components being nested. □

Remark 4.28. *Using a similar argument we could have shown that if a partition $\mathfrak{p}$ had m' components with at least two vertices and the longest sequence of nested components was m'' then*

$$\mathbb{P}_{\mathfrak{p}}(\mathfrak{b}_{\mathfrak{g}} \cap R_{(K+m'')N} \neq \emptyset) \leq c_1 m' \exp(-c_2 K + c_3 m'').$$

This would not strengthen the final bound that we are able to prove in this thesis but if in future work we could control the number of nested components this bound would lead to a stronger result.

We now posses all the results we require to complete the objective of this section and prove a bound for the probability that a grove contains a path that travels too far.

Proof of Proposition 4.15. Suppose the partition $\mathfrak{p}$ of R_1 consist of the components $(\mathfrak{p}_1, \mathfrak{p}_2, \dots \mathfrak{p}_m)$, with m' components having at least two vertices.

By Lemma 4.27 an upper bound can be provided based on the size of m'. It is clear that increasing m', the number of non-solitary vertex partitions, increases the upper bound. As increasing the number of vertices in a partition beyond two does not increase the bound, taking each component $\mathfrak{p}_i$ to contain exactly two vertices, except if N is odd when we take $\mathfrak{p}_1$ to have three vertices, will provide an upper bound that holds uniformly for all possible partitions.

Hence we can take $m = m' \leq N/2$ and conclude that

$$\mathbb{P}_{\mathfrak{p}}(\mathfrak{b}_{\mathfrak{g}} \cap R_{KN+N^2} \neq \emptyset) \leq c_1 \exp(-c_2 K + c_3 N).$$

□

4.5 Bound for existence of a block.

Recall that any spanning forest is uniquely determined by the dual spanning forest it defines. The original graph we are working on, $(G \times [-q_1, q_2]) \cup \{s\}$, has free boundary conditions on the top,bottom and right-hand side and has the wired boundary on the left-hand side of the graph, this means the dual graph has a free boundary on the left-hand side and a wired boundary on the other three sides of the graph. For convenience we will use a modification of the planar dual and split the sink into multiple vertices. Denote by s_1 the vertex that the top row of vertices in the dual graph are connected to, and s_2 for the sink connected to the bottom row.

In the previous chapter we also had the concept of a block, although it is a slight variation of this we now want.

Definition 4.29. *A* block *between R_i and R_j, is a set of edges, U, in the dual tree such that it has the following properties:*
(i) U is contained between R_i and R_j.
(ii) U consists of two sets of connected edges, one component containing s_1 and the other s_2.
(iii) There exists a unique edge in the original graph that it is included in every path from R_i to R_j that does not intersect U.

Lemma 4.30. *There exists constants $c_1, c_2 > 0$ such that for all $K^* \geq 1$, $m > 0$ and for sufficiently large N*

$$\mathbb{P}(\textit{no block between } R_m \textit{ and } R_{m+K^*N}) \leq c_1 \exp(-c_2 K^*).$$

Proof. The approach used here is the same idea as we used to construct a block in the previous chapter, by bounding the probability of a given construction occurring.

Construction of a block.
We firstly describe a particular method of constructing a block between the desired rungs, then we will bound the probability of such a construction successfully occurring. See Figure 4.3 for an example of the first three steps of a typical construction.

Let v_i denote the vertex that the i^{th} random walk will start from in the dual graph.

From each start vertex, v_i, we will run a loop erased random walk until the first time it hits $s_1 \cup s_2$ or any other vertex in the previous paths. Denote the i^{th} path by α_i.

We now inductively define the start vertices, all of which will be on rung $m+K^*N/2$. Set $v_1 = (m + K^*N/2, N/2)$, the mid-point of the domain that can contain U.

Suppose the LERW from v_1 that terminates upon hitting $s_1 \cup s_2$ generated the path α_1. If α_1 contains the vertex s_1 then take v_2 to be $(m + K^*N/2, N/4)$. Alternatively if α_1 contains s_2 we would take $v_2 := (m + K^*N/2, 3N/4)$.

More generally suppose we have created the first $i-1$ start vertices and paths. The vertex v_{i-1} will be situated in an interval between two of the vertices in the set

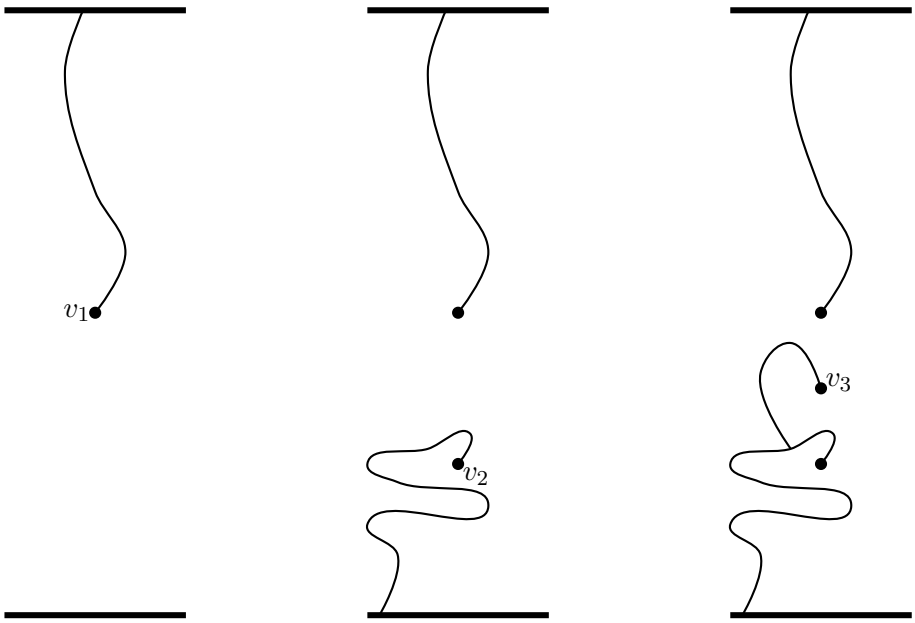

Figure 4.3: *An example of the first three steps in a construction of a block.*

$\{s_1, s_2, v_1, v_2 \dots v_{i-2}\}$. We denote the vertices it is between by u and w, note that one will be in the connected component containing s_1 and the other will be in the component connected to s_2. For a vertex v let v^h denote its vertical co-ordinate.

If α_{i-1} ends at a vertex that is connect to u we take the midpoint between v_{i-1} and w, namely $v_i = \big(m + K^*N/2, (v_{i-1}^h + w^h)/2\big)$. If instead α_{i-1} ends at a vertex that is connect to w we take the midpoint between v_{i-1} and u to be the next start vertex, so we would have $v_i = \big(m + K^*N/2, (v_{i-1}^h + u^h)/2\big)$. The path α_i is then defined to be the LERW started at v_i that terminates upon first hitting a vertex in $\bigcup_{j=1}^{i-1} \alpha_j \cup \{s_1, s_2\}$.

If the above construction of v_i does not give a vertex then take the nearest vertex in the dual graph and split any draws by taking the higher vertex, or by any other arbitrary rule.

This procedure continues until the first time we have a start vertex whose neighbouring vertices in rung $m + K^*N/2$ are both previous start vertices. The process will then terminate after generating the walk from this vertex.

If all of the paths in this construction end before they pass R_m or R_{m+K^*N}, then it will have created a block in the dual forest between these rungs.

Probabilistic bound.

We can therefore provide an upper bound for the probability of there not existing a block by the probability that this construction does not succeed in producing a block.

In order for this construction to fail it suffices to check if any of the walks exit the domain before hitting the previous paths.

By an application of Lemma 4.10 we can deduce that the probability of α_1 exiting the box before hitting s_1 or s_2 is at most $\exp(-cK^*)$.

Now consider the i^{th} walk. In order to pass R_m or R_{m+K^*N} the walk must first

reach a distance N from v_i. Observe that v_i will be at most $2^{-i}N$ from a vertex in a previous paths, in particular from v_{i-1}.

Let B_i be a ball of radius N around v_i. It is possible that this will not be completely contained within the domain, however the walk can not reach the part of the ball outside the domain without hitting s_1 or s_2 hence the walk will always terminate before leaving the domain. Therefore considering the ball and the ball restricted to the domain amount to the same thing.

By Beurling's estimate, [35, Section 6.8], we have that

$$\mathbb{P}_{v_i}(\xi_{B_i} < \tau_{\cup_{j<i}\alpha_j}) \leq c_1 \left(\frac{2^{-i}N}{N}\right)^{\frac{1}{2}}.$$

Next suppose the walk managed to reach a distance N from the start vertex without being terminated, this means we can no longer assume that it will be near any vertex in the previous path. We now no longer concern ourselves with whether the walk hits the previous paths and just ask for the walk to avoid s_1 and s_2, this is allowed as we only need to deduce an upper bound.

Consider the first vertex, x, that the random walk hits that was distance N from v_i. From x if the walk is to reach R_m without hitting s_1 or s_2 then it must first reach rung $m + (\frac{K^*}{2} - 1)N$. By Lemma 4.10, the probability that a walk from $R_{m+(\frac{K^*}{2}-1)N}$ could reach R_m without hitting $s_1 \cup s_2$ is bounded above by $\exp(-c_2(K^* - 1))$, note that this bound holds uniformly for all vertices in $R_{m+(\frac{K^*}{2}-1)N}$. Hence

$$\begin{aligned}
&\mathbb{P}_x(\tau_{R_m} \leq \tau_{s_1} \wedge \tau_{s_2}) \\
&= \sum_{y \in R_{m+(\frac{K^*}{2}-1)N}} \mathbb{P}_x(\tau_{R_{m+(\frac{K^*}{2}-1)N}} \leq \tau_{s_1} \wedge \tau_{s_2}, S_x(\tau_{R_{m+(\frac{K^*}{2}-1)N}}) = y)\mathbb{P}_y(\tau_{R_m} \leq \tau_{s_1} \wedge \tau_{s_2}) \\
&\leq \mathbb{P}_x(\tau_{R_{m+(\frac{K^*}{2}-1)N}} \leq \tau_{s_1} \wedge \tau_{s_2}) \exp(-c_2(K^* - 1)) \\
&\leq \exp(-c_2(K^* - 1)).
\end{aligned}$$

Similarly using the symmetry of the walk and the domain, for the walk to reach R_{m+K^*N} it must first hit rung $m + (\frac{K^*}{2} + 1)N$, and then we could apply Lemma 4.10 to the remaining part of the walk to deduce

$$\mathbb{P}_x(\tau_{R_{m+K^*N}} < \tau_{s_1} \wedge \tau_{s_2}) \leq \exp(-c_2(K^* - 1)).$$

Therefore

$$\begin{aligned}
\mathbb{P}_x(\tau_{R_m} \wedge \tau_{R_{m+K^*N}} \leq \tau_{s_1} \wedge \tau_{s_2}) &\leq \mathbb{P}_x(\tau_{R_m} \leq \tau_{s_1} \wedge \tau_{s_2}) + \mathbb{P}_x(\tau_{R_{m+K^*N}} < \tau_{s_1} \wedge \tau_{s_2}) \\
&\leq 2\exp(-c_2(K^* - 1)),
\end{aligned}$$

and so for $i \geq 2$

$$\begin{aligned}
&\mathbb{P}(i^{th} \text{ path fails}) \\
&\leq \mathbb{P}_{v_i}(\xi_{B_i} < \tau_{\cup_{j<i}\alpha_j}) \sum_{x \in \partial B_i} \mathbb{P}(S(\xi_{B_i}) = x | \xi_{B_i} < \tau_{\cup_{j<i}\alpha_j}) \mathbb{P}_x(\tau_{R_m} \wedge \tau_{R_m + K^* N} \leq \tau_{s_1} \wedge \tau_{s_2}) \\
&\leq c_1 2^{1-i/2} \exp(-c_2(K^* - 1)).
\end{aligned}$$

Now bounding the sum over the finite number of paths involved in the construction by an infinite sum gives

$$\begin{aligned}
\mathbb{P}(\text{no block}) \leq \mathbb{P}(\text{construction fails}) &\leq \exp(-cK^*) + c_1 \sum_{i=2}^{\infty} 2^{1-i/2} \exp(-c_2(K^* - 1)) \\
&\leq c_3 \exp(-c_2 K^*).
\end{aligned}$$

□

We now have all the concepts and results we require to prove the theorem.

4.6 Proof of Theorems.

Let $\mathfrak{b}_{\mathfrak{g}}$ be the backbone component of the grove $\mathfrak{g}$ on Λ_2. Let $\mathfrak{p}(\mathfrak{b}_{\mathfrak{g}})$ be the partition the backbone induces on R_1.

Proof of Theorem 4.3. Recall from Chapter 2 that once the spanning tree up to the earliest common ancestor of a vertex and its neighbours is known then the sandpile configuration at that vertex can be computed. Therefore to determine whether the event $\eta(k, q_2)$ occurs it suffices to know the descendants of the earliest common ancestor of vertices in R_{k-1} in the spanning tree associated via ϕ_{R_0}.

Therefore if we can find a coupling between spanning trees when conditioned on I or with no conditioning, we would have a bound for the difference between the conditioned and unconditioned measures.

By Lemma 4.13 we see that conditioning on a suitable rooted partition for the grove with respect to R_1 as well as the sandpile configuration on Λ_1 is enough to determine the spanning forest, F, on Λ_1^+. Let the set of rooted partitions on R_1 that can be involved in giving rise to I on Λ_1 be denoted by $\mathfrak{P}$.

Therefore the spanning forest on Λ_2 with respect to the burning bijection centred at R_0 conditioning on I can be studied by conditioning on a partition from $\mathfrak{P}$ and the spanning forest on Λ_1^+. We now further condition on which of the possible partitions, $\mathfrak{p} \in \mathfrak{P}$, occurs.

Now observe that the spanning forest F is only a dependent of the distribution of the groves on Λ_2 through the partition $\mathfrak{p}$. Consequently when investigating groves on

Λ_2 conditioned on I we will wish to consider groves on Λ_2 conditioned on $\mathfrak{p}$ which is determined by the backbone $\mathfrak{b}_\mathfrak{g}$.

Next consider the method of constructing a grove with a prescribed connection pattern presented in Chapter 2. If $\mathfrak{b}_\mathfrak{g}$ is constructed and it does not intersect $R_{\theta N+N^2}$ then we could continue the algorithm by running Wilson's algorithm on vertices between $R_{\theta N+N^2}$ and $R_{\theta N+N^2+K^*N}$. Note that as a spanning forest is uniquely determined by its dual we could equivalently continue the algorithm by running Wilson's algorithm on the dual graph, which will be denoted G^*.

Let us consider how knowledge of $\mathfrak{b}_\mathfrak{g}$ in the primal graph converts into a dual tree event. Observe that in the dual tree knowing that the tree on the primal graph must contain the edges in $\mathfrak{b}_\mathfrak{g}$ is equivalent to saying there is a set of edges $\mathfrak{b}^*_\mathfrak{g}$ that can not be included in the dual tree.

Therefore constructing the dual tree can be done utilising Wilson's algorithm on each connected component of the graph $G^* \setminus \mathfrak{b}^*_\mathfrak{g}$. Moreover as we are free to choose the start vertices used in Wilson's algorithm we can use the same method as for the construction described in the proof of Lemma 4.30 to construct the part of the dual tree that occurs between $R_{\theta N+N^2}$ and $R_{\theta N+N^2+K^*N}$.

Assuming that $\mathfrak{b}_\mathfrak{g}$ does not cross $R_{\theta N+N^2}$ the transition probabilities for the random walks involved in the construction of a block between $R_{\theta N+N^2}$ and $R_{\theta N+N^2+K^*N}$ in G^* are the same as those used when running Wilson's algorithm on the graph $G^* \setminus \mathfrak{b}^*_\mathfrak{g}$. This observation allows us to couple the construction of a block under conditioning and the unconditioned case, and thus we will be able to utilise Lemma 4.30 in the desired setting.

If a block existed in the dual tree, it would mean that the earliest common ancestor of $R_{N^2+\theta N+K^*N}$ would be forced to occur after $R_{N^2+\theta N}$. Thus the sandpile configuration on $R_{N^2+\theta N+K^*N}$, and therefore all later rungs, are independent of the spanning tree up to $R_{N^2+\theta N}$ and thus would be independent of I.

An upper bound could therefore be provided if we can control the events that $\mathfrak{b}_\mathfrak{g}$ does not intersect $R_{\theta N+N^2}$ and that a block exists between $R_{\theta N+N^2}$ and $R_{\theta N+N^2+K^*N}$.

Observe that the bound in Proposition 4.15 is uniform with respect to the rooted partition $\mathfrak{p}$ we have conditioned upon. Hence the same bound will also hold when we have conditioned upon the event $\mathfrak{p}(\mathfrak{b}_\mathfrak{g}) \in \mathfrak{P}$. Therefore we have control over whether $\mathfrak{b}_\mathfrak{g}$ intersects $R_{\theta N+N^2}$.

Assuming that none of the paths in $\mathfrak{b}_\mathfrak{g}$ intersect $R_{N^2+\theta N}$, we can use Wilson's algorithm to generate the uniform spanning forest on the remaining graph, conditioned on the backbone connected to R_1 being $\mathfrak{b}_\mathfrak{g}$. Due to our earlier observations in this proof about coupling the block construction in the conditioned and unconditioned case we can deduce that the probability of a block occurring between $R_{\theta N+N^2}$ and $R_{\theta N+N^2+K^*N}$ is bounded by Lemma 4.30.

Therefore for any rung k such that $k > K^*N + \theta N + N^2$ the spanning tree at R_k is independent of I if either the backbone attached to R_1 reaches $R_{N^2+\theta N}$ and conditioned

on this backbone we then have independence, or no path in the backbone attached to the vertices in R_1 reaches $R_{N^2+\theta N}$ and conditioned on this type of backbone we have a block between rungs $N^2+\theta N$ and $N^2+\theta N+K^*N$.

Therefore combining the bounds found in Proposition 4.15 and Lemma 4.30, we have some constants $c_1, c_2, c_3, c_4, c_5, c_6 > 0$ which have no dependence on θ or N such that

$$\begin{aligned}
&\left|\nu_{[-q_1,q_2]}(\eta(k,q_2)|I) - \nu_{[-q_1,q_2]}(\eta(k,q_2)\right| \\
\leq &\mathbb{P}(\mathfrak{b}_{\mathfrak{g}} \cap R_{\theta N+N^2} \neq \emptyset | \mathfrak{p}(\mathfrak{b}_{\mathfrak{g}}) \in \mathfrak{P}) \\
&+ \mathbb{P}(\text{there is not a block between } R_{\theta N+N^2} \text{ and } R_{\theta N+N^2+K^*N}) \\
\leq &c_1 \exp(-c_2\theta + c_3 N) + c_4 \exp(-c_5 K^*) \\
\leq &c_6 \exp(-c_2\theta + c_3 N).
\end{aligned}$$

By taking $K^* = \frac{c_2\theta + c_3 N}{c_5}$ and θ sufficiently large. □

Proof of Theorem 4.2. To extend the result to hold on a graph with an infinite number of rungs we need some approximations.

From convergence of the sandpile measures on Ladder graphs proven in [26], it follows that for all $\varepsilon_1, \varepsilon_2 > 0$ there exists a $Q_1 > 0$ such that when $q_2 > Q_1$ the following hold

$$\begin{aligned}
\left|\nu_{[-q_1,q_2]}(\eta(k,q_2)|I) - \nu(\eta(k,q_2)|I)\right| &\leq \varepsilon_1 \\
\left|\nu_{[-q_1,q_2]}(\eta(k,q_2)) - \nu(\eta(k,q_2))\right| &\leq \varepsilon_2.
\end{aligned}$$

Also we have that we can approximate an event depending on an infinite number of rungs by cyclinder events , so for all $\varepsilon_3, \varepsilon_4 > 0$ there exists $Q_2 > 0$ such that when $q_2 > Q_2$ the following statements are true.

$$\begin{aligned}
\left|\nu(\eta(k,q_2)|I) - \nu(\eta(k,\infty)|I)\right| &\leq \varepsilon_3 \\
\left|\nu(\eta(k,q_2)) - \nu(\eta(k,\infty))\right| &\leq \varepsilon_4.
\end{aligned}$$

Hence by taking $q_2 > max\{Q_1, Q_2\}$

$$\left|\nu(\eta(k,\infty)|I) - \nu(\eta(k,\infty))\right| \leq \varepsilon_1 + \varepsilon_2 + \varepsilon_3 + \varepsilon_4 + \left|\nu(\eta(k,q_2)|I) - \nu(\eta(k,q_2)\right|.$$

Moreover we can chose $\varepsilon_1, \varepsilon_2, \varepsilon_3, \varepsilon_4$ such that $\sum_i \varepsilon_i < \exp(-c_2\gamma + c_3 N)$ and then the result follows from Theorem 4.3. □

4.7 Periodic Boundary on the strip.

In the final section of this chapter we briefly consider what would happen if we had defined the lattice graph on the surface of a cylinder, i.e it was periodic in the vertical direction.

Theorem 4.31. *Given the graph* $[0, N+1] \times [-q, \infty) \subset \mathbb{Z}^2$ *for any* $q > 0$*, where row* 0 *and row* $N+1$ *have been identified, so we have a lattice grid on the surface of a cylinder. Let* η *be a recurrent sandpile configuration on this graph and let* η_i *denote the configuration on the vertices in the* i^{th} *rung,* R_i*. Let* ν *be the measure on recurrent sandpile configurations on this graph. We define* $I := \{\eta_j = E_j \text{ for } j \in [-q, 0]\}$*, the event that the sandpile between rung* $-q$ *and rung* 0 *is some known configuration,* E*. Further define* $\eta(i, j)$ *to be any event that is determined by the configuration of the sandpile between rungs* i *and* j*. Then with* $k \geq N^2 + \gamma N$ *for sufficiently large* $\gamma > 0$ *and some constant* $c_1, c_2, c_3 > 0$*, we have that*

$$\Big|\nu\big(\eta(k,\infty)|I\big) - \nu\big(\eta(k,\infty)\big)\Big| \leq c_1 \exp(-c_2\gamma + c_3 N). \tag{4.21}$$

We now briefly give a rough outline to how the proof would proceed, by highlighting the areas where it differs from the proof of Theorem 4.2.

Sketch of proof. Most of this proof will follow immediately from the above results for the non-periodic boundary case. The proof of Theorem 4.2 follows from Theorem 4.3, it will also be true that in the periodic boundary case that the result on an infinite ladder graph follows from the result on a finite graph. Therefore we need to see how Theorem 4.3 would need to be altered to hold in this new setting.

There are two parts to the proof that we need to check. Firstly we can again implement the same decomposition of the domain as in section 4.2.1. Note the maximum diameter of a ball would still be N. Hence we can still use the same argument to decompose the random walk into paths that may or may not hit the centre of the balls. Now observe that an equivalent bound to lemma 4.9 exists in the periodic case. Thus the same proofs as in Section 4.4 will also work in the periodic case.

The second part in Section 4.5 where we build a block would need more care and there is a distinction between the proofs. In the non-periodic case the random walk on the dual graph was always within $N/2$ steps of the sink and so we could utilise this to bound the probability of the walk reaching a distant rung without hitting the sink. Unfortunately this is no longer true in the periodic case, so to prove a result of the form of Lemma 4.30 we will require a different approach. However we can use a similar idea to one we had whilst creating a block in Chapter 3. This involved creating a "backbone" that was sufficiently "well behaved" (note this is not the same as the backbone of the grove we have discussed earlier in this chapter). By a backbone we now mean a path in the dual tree from a vertex in R_{m+K^*N} to the sink. For a random

walk starting from a vertex in R_i for some $i > m$ we say the walk is *well behaved* if once the random walk has hit R_m it does not return to R_{m+K^*N} .

Once the backbone has been created the remainder of the construction is the same as the non-periodic case except that the backbone now plays the role of $\{s_1\} \cup \{s_2\}$.

The proof found in Chapter 3 required three properties. Firstly that the infinite LERW and the LERW stopped upon exiting a subgraph Λ agreed in some domain. Secondly that once the walk was sufficiently far away it would not return to a ball about the origin. Finally that the rest of the construction was satisfied. These requirements are now translated into the current setting.

The first and second requirement can be dealt with together, by showing that the random walk that generates the backbone, from a vertex in R_{m+K^*N} that terminates upon hitting the sink in the dual tree, is well behaved. An upper bound will then follow from existing results. By conditioning on the LERW path until the walk first intersects R_m and using the domain Markov property we can see that the required behaviour of the remainder of the walk can be controlled by an application of Proposition 4.17. This is therefore exponentially bounded by $c_1 \exp(-c_2 K^*)$. Note that whilst the path until the first intersection with R_m will depend on $\mathfrak{b}_{\mathfrak{g}}$ the bound for the remainder of the walk holds uniformly for all possible paths and thus this bound also holds when conditioning upon knowledge of the backbone, $\mathfrak{b}_{\mathfrak{g}}$, in the primal tree.

The third requirement is controlled by the same method as in the non-periodic case by using Beurling's estimate and an equivalent statement to Lemma 4.10. This would yield an upper bound of $\exp(-c_3 K^*)$, which again would hold uniformly amongst all possible well behaved backbone paths.

Hence, up to some changes in constants, the same bound holds for the periodic case as for the non-periodic case. □

APPENDIX A

AN ENCODING OF THE BURNING PROCESS.

In this appendix we will give a construction on recurrent Ladder sandpiles that encode the burning process with respect to rungs.

Recall a ladder sandpile is of the form $G \times \mathbb{N}$ where a copy of the graph G is called a rung. A special vertex, s, is defined to be the sink and each vertex in rung 1 will be connected to the sink by one edge.

Also recall that in [25], Járai and Lyons showed that there existed a measure on "left-burnable" sandpiles on $[1, N] \times \mathbb{Z}$. The left-hand boundary of a finite subgraph, H, is given by those $v \in H$ such that there exists an (infinite) path from v to $R_{-\infty}$ that only uses vertices in H that have been declared burnt and vertices in $H^c \subset [1, N] \times \mathbb{Z}$. By "left-burnable" it is meant that the configuration on H can be burnt under the usual burning rule when restricted to only allowing a vertex to burn when it is on the left-hand boundary of H.

This allows us to conclude the existence of a measure on our related graph. The distribution of recurrent sandpile configurations that can occur after a maximum rung as given by the left-burnable measure is exactly the distribution on the graphs we consider.

In [16] it was shown that there is a Markov chain that encodes the possible configurations of the spanning tree for the ladder graph with rungs given by $G = \{0, 1\}$. By considering non-crossing partitions of the vertices in a rung the statespace for the spanning tree of the graph $\{1, \dots N\} \times \mathbb{N} \subset \mathbb{Z}^2$ can be chosen such that its size is $\exp(cN)$.

Ideally we would like to have an encoding of the sandpile configuration whose statespace contained configurations on a rung but this is not sufficient information to ensure the sandpile is recurrent. This is because the recurrence of a configuration has a non-local aspect which causes non-trivial constraints to be placed on which sequences of rung configuration are allowed to occur. Therefore we need to include more information; the approach we take here is to know how previous rungs can possibly burn as

well as the configuration on a rung.

In [25], the authors also showed that there existed such a Markov chain on ladder sandpiles of the form $G \times \mathbb{N} \subset \mathbb{Z}^2$. However the size of their statespace grows at a non-optimal rate. We can give an encoding for sandpiles on these graphs that we believe has an exponential number of elements in its statespace.

For a subgraph, H, of a ladder graph, we use the notation $\mathcal{R}_H$ to denote the set of sandpile configurations that can occur on the graph H as part of a recurrent configuration on the whole ladder graph.

Definition A.1. *Given two sandpile configurations $\eta, \zeta \in \mathcal{R}_{G\times[1,q]}$ we define the* extended sandpile*, $\theta := \eta \cup \zeta$ on the graph $G \times [1, 2q]$ by taking $\theta|_{\mathcal{R}_{G\times[1,q]}} = \eta$ and $\theta|_{\mathcal{R}_{G\times[q+1,2q]}} = \zeta$.*

Definition A.2. *For a set $\mathcal{S}$ and maps $\varphi_q : \mathcal{R}_{G\times[1,q]} \to \mathcal{R}_G \times \mathcal{S}$ define the set $\Phi = \{\varphi_q : q \geq 1\}$. Suppose we are also given a map $\psi : \mathcal{R}_G \times \mathcal{S} \to \mathcal{S}$. We say the triple $(\mathcal{S}, \Phi, \psi)$ is a* coding *if the following properties hold.*

(i) For $q_1, q_2 \geq 1$ let η_1, respectively η_2, be recurrent configurations on $G \times [1, q_1]$, respectively $G\times[1, q_2]$, and ζ a sandpile configuration on $G\times[1, \infty)$. If $\varphi_{q_1}(\eta_1) = \varphi_{q_2}(\eta_2)$ then the extended sandpile $\eta_1 \cup \zeta$ is a recurrent configuration on $G \times \mathbb{N}$ if and only if $\eta_2 \cup \zeta$ is a recurrent configuration.

(ii) With the projection mapping, $\pi : (\mathcal{R}_G\times\mathcal{S}) \to \mathcal{R}_G$, we have that $\pi(\varphi_k(\eta(1, k))) = \eta(k)$.

(iii) For $k \geq 2$, if $\varphi_{k-1}(\eta(0, k-1))) = (\eta(k-1), \sigma_{k-1})$ and $\varphi_k(\eta(0, k)) = (\eta(k), \sigma_k)$ then $\psi(\eta(k), \sigma_{k-1}) = \sigma_k$.

The construction in [25] would satisfy this definition for $G = [1, N] \subset \mathbb{Z}$, however it would not satisfy the following conjecture.

Conjecture A.3. *For $G = [1, N]$ there exists a triple $(\mathcal{S}, \Phi, \psi)$ that satisfies Definition A.2 such that there exists a constant $c > 0$ such that $|\mathcal{R}_{[1,N]} \times \mathcal{S}| \leq \exp(cN)$.*

If we could prove such a result then we could define an encoding which had a statespace with an exponential number of elements, which we suspect to be true due to the connection with spanning trees.

The encoding we present in this appendix satisfies Definition $A.2$ and we believe that it may also simplify the amount of information required to keep track of the burning at each stage suggesting a proof of the conjecture is possible. In our encoding the set $\mathcal{S}$ will be the set of possible multigraphs S_k which we will define shortly.

It is also hoped that once the Ladder graph case has been proven that this encoding could be further generalised to one that works on $\mathbb{Z}^d$ and provide another way of thinking about recurrent sandpiles in this important setting.

Trying to use the standard burning bijection to find such an a coding would not be useful. The reason for this, similar to the one faced in Chapter 4, is that we would

not only need to know what vertices in a rung can burn but also to have a concept of respective burning times. Therefore we use a different burning rule, which is referred to as burning with respect to rungs.

In Chapter 4 we had a modified burning rule where burning could only occur in one of the two sections at a time. Here we use a similar idea where burning can occur in only one rung at a time. This burning procedure was introduced in [25].

Burning based on rungs. Let η be a sandpile on the ladder graph $H = \{s\} \cup (G \times [0, q])$ with vertex set $V \cup \{s\}$.

Phase 1: Declare the sink to be burnt, so we can define two sets, $B(1) = \{s\}$ and $U(1) = V$.

Phase i: Using the standard burning rule we find the set of unburnt vertices, W, that are currently in a position to be burnt,

$$W := \{v \in U(i-1) : \eta(v) \geq deg_{U(i-1)}(v)\}.$$

Let $f := f(i, \eta) = \min\{n \in \mathbb{N} : W \cap R_n \neq \emptyset\}$. Burn any vertices in R_f that we can without allowing any vertex in another rung to burn. To find this we define the following sets,

$$\begin{aligned} B(i,1) &= \{v \in R_f \cap U(i-1) : \eta(v) \geq deg_{U(i-1)}(v)\} \\ U(i,1) &= U(i-1) \setminus B(i,1). \end{aligned}$$

Then inductively for $j \geq 2$ define

$$\begin{aligned} B(i,j) &= \{v \in R_f \cap U(i,j-1) : \eta(v) \geq deg_{U(i,j-1)}(v)\} \\ U(i,j) &= U(i,j-1) \setminus B(i,j) \end{aligned}$$

There exists a j^* such that $B(i, j^*) = \emptyset$ and at this step the phase terminates and we set $B(i) = \cup_{j=1}^{j^*} B(i,j)$ and $U(i) = U(i-1) \setminus B(i)$.

Let ϕ_R be the mapping that given a sandpile configuration, η will assign labels to the unburnt vertices. Namely $\phi_R(\eta, w)$ assigns w the label corresponding to the index of the step and phase that w can burn in η according to the burning based on rungs procedure.

This algorithm will terminate when no more vertices remain unburnt in a finite graph. If we applied this algorithm to an infinite graph, then the burning procedure in any finite subset of the graph will terminate in finite time almost surely, with respect to the left-burning measure. This follows from the fact that the graph has the one-end property.

In the construction that follows we can have any finite graph as the rung, including d-dimensional and non-planar graphs.

It is clear that retaining knowledge of all of the previous rungs, $\eta(1,k)$, would determine which configurations are allowed to occur on rung $k+1$. The idea behind our encoding of the sandpiles is for each rung, R_k, to be associated to a directed multigraph, S_k, whose edges have been given an integer label that encodes all the information about how the burning can continue through the previous rungs. Another interpretation, using the ideas of the previous chapter, is that the graph S_k will encode the set of possible partitions that could be induced on the rung R_k that would allow burning of the remainder of the graph.

Burning a multigraph. We now introduce the procedure of *burning* a directed multigraph whose edges are labelled with a non-negative integer number. The outcome of this procedure is to declare some vertices and edges to be burnt and to reduce the label of some of the unburnt edges. To perform this operation we would initially be given a vertex (or vertices) that have been declared burnt and a set of edges, E, that are exempt, in the sense that they will not be declared burnt.

Suppose we have a vertex, v, that is burnt then every edge not contained in E that is directed out of v is declared to be burnt.

Now consider an unburnt vertex, w. When an edge directed into w is declared burnt subtract one from the label of each edge that is directed out of w that had a non-zero label. If a vertex has an edge directed out of it that has a label of zero attached to it then the vertex is declared burnt. A vertex is also declared burnt if it has no out going edges (this situation can not arise in the following construction but can occur when we start to allow the graph to be simplified).

When no more edges or vertices can be burnt in the graph, except for those in E, it is said to have been burnt.

There is another operation on the multigraph which we will call *removal*. This works on a labelled graph that is burnt and a given subset of vertices, W, by removing any edge or vertex that are not in W but have been labelled as burnt. If an edge has its start or end vertex removed then it too is removed.

We would like to highlight a couple of points for the reader. Firstly that the edges in E can have their labels reduced, possibly even to zero, but can not be declared burnt. Secondly that there is a distinction between burnt vertices and removed vertices, in that a burnt vertex is not always removed from the graph. Also note that no removal operations can be performed until all burning has occurred in a graph.

Remark A.4. *Given the set of vertices that are initially burnt every possible order that we may choose to continue declaring vertices and edges as burnt will yield the same labelled multigraph when no further edge or vertex can be declared burnt. This follows from the fact that any vertex or edge that is burnable will still be burnable after something else has burnt.*

The multigraphs, S_k, that we will define are a reformulation of the set of heights in the sandpile configuration of unburnt vertices with respect to the burning based on

rungs procedure at the end of a certain phase. The advantage of our encoding is that a lot of the information in the multigraph can be seen to be redundant. We will now describe how to create these multigraphs before showing that they can be used to find a triple that satisfies Definition A.2.

A.1 Construction of S_k.

Given a sandpile configuration, η, defined on the graph $H = \{s\} \cup G \times \mathbb{N}$ we describe how to construct the directed multigraphs inductively.

The graph S_k consists of four parts; a set of vertices representing R_k, a set of "temporary" vertices representing R_{k+1}, a set of vertices representing all vertices in previous rungs and a set of directed edges with labels. The construction of the directed multigraph S_k is inductive. Given S_{k-1} we first take copies of the rungs R_{k-1}, R_k and R_{k+1} and determine the labelled edge set attached to them. Then these vertices and edges are combined with S_{k-1} to form a new graph, denoted Q_k. The final step is to apply the burning and removal operations to Q_k.

We have provided an example for reference at the end of the appendix for the reader to follow along with, which will hopefully make the process clearer. In the figures the label of an edge is depicted by placing the appropriate number of arrows on the edge.

We begin with the construction of S_1. We will start with the knowledge of $\eta(1)$ and that the sink, s, has been declared burnt.

Construction of S_1.

Step 1: Encode $\eta(1)$.

Start with copies of R_1 and R_2, label them R_1^1 and R_2^1 respectively. Introduce a map $I_1 : (R_1^1 \cup R_2^1) \to (R_1 \cup R_2)$ which represents the natural identification of a vertex in a copy of a rung to a vertex in H.

For each $v \in R_1^1$ place an edge from v directed towards w if $I_1(v)$ was a neighbour of $I_1(w)$ in H. The label of an edge that is directed out of v is given by $\deg_H(I_1(v)) - \eta(I_1(v))$. Also include a set of edges, $\mathfrak{E}_1$, with contains an edge directed from each $v \in R_2^1$ to its neighbour in R_1^1, we will arbitrarily give these edges a label of one. The labelled, directed multigraph this produces is denoted T_1, see figure A.2 for an example construction.

Step 2: Connecting the sink.

As each vertex in R_1 is attached to the sink we extend the graph T_1 by adding a vertex s and for each $v \in R_1^1$ place an edge directed from the sink to v. Let these edges have label 0. This graph is denoted Q_1.

Step 3: Burning Q_1.

Next we allow the graph Q_1 to burn with the exemption that no edge from a vertex in R_1^1 directed towards a vertex in R_2^1 is allowed to be declared burnt. When Q_1 is burnt we perform the removal operation on it with the restriction that no vertex in R_1^1 can

be removed. This leaves the graph which we will call S_1.

Note that the graph S_1 has the property that at least one vertex in R_1 will have been declared burnt, this is because $\eta(1)$ forms part of a recurrrent configuration and R_1 contains all neighbours of s. Also note that the vertex set of S_1, $V(S_1)$, contains R_1^1 and R_2^1.

We now proceed inductively to define the multigraphs S_k, there will be simillarities to the construction of S_1.

Construction of S_k. Suppose that we know the configuration $\eta(k)$ and the multigraph S_{k-1}. We will assume for an inductive argument that $R_{k-1}^{k-1}, R_k^{k-1} \subset V(S_{k-1})$ and that $\mathfrak{E}_{k-1} \subset E(S_{k-1})$, this will be proven for S_k during the construction. The basis case for $k = 1$ is true from the above construction. We will also assume that at least one vertex of R_{k-1}^{k-1} has been declared burnt in S_{k-1}, the inductive proof of this for S_k will be shown later in Lemma A.7.

Step 1: Encoding $\eta(k)$.

Start with a copy of the vertices in R_{k-1}, R_k and R_{k+1} and label them R_{k-1}^k, R_k^k and R_{k+1}^k respectively. Each vertex in these copies has a natural identification to a vertex in H, let this be represented by the map $I_k : (R_{k-1}^k \cup R_k^k \cup R_{k+1}^k) \to (R_{k-1} \cup R_k \cup R_{k+1})$.

For each vertex $v \in R_k^k$ place an edge directed from v towards w if $I_k(v)$ and $I_k(w)$ were neighbours in H. The label of an edge directed out of v is given by $\deg_H(I_k(v)) - \eta(I_k(v))$.

Also include an edge direct from each vertex in R_{k+1}^k to its neighbour in R_k^k, arbitrarily give these a label of one, call this set of edges $\mathfrak{E}_k$

The multigraph this yields is denoted by T_k. See figure A.4 for an example of this step.

Step 2: Connecting the previous graph.

We now wish to combine the graphs T_k and S_{k-1}. Consider the vertex and edge sets of the graphs. We have that $V(S_{k-1}) = R_{k-1}^{k-1} \cup R_k^{k-1} \cup V'$, where V' is some set of vertices. Let $E(S_{k-1})$ be the edge set of S_{k-1}, observe that this will contain $\mathfrak{E}_{k-1}$. For T_k, we have a vertex set $V(T_k) = R_{k-1}^k \cup R_k^k \cup R_{k+1}^k$ and some edge set $E(T_k)$. There is a natural identification between vertices in R_j^{k-1} and R_j^k, for $j = k-1, k$, given by by the map $I_{k+1}^{-1} \circ I_k|_{R_j^k}$, i.e two vertices can be identified if they both are identified to the same vertex in H. Using this identification we will stop referring to $R_k^{k-1} \cup R_{k-1}^{k-1}$.

We will define a new graph Q_k by specifying its vertex and edge set, namely $V(Q_k) = V(T_k) \cup V(S_{k-1}) = R_{k-1}^k \cup R_k^k \cup R_{k+1}^k \cup V'$, and $E(Q_k) = E(T_k) \cup (E(S_{k-1}) \setminus \mathfrak{E}_{k-1})$.

See figure A.5 for the example of the output of this step.

Step 3: Burning Q_k.

We now burn the graph Q_k to produce the graph S_k. However we do not want to allow the whole graph to burn, the exemption from the burning on the multigraph is the edges from vertices in R_k^k to R_{k+1}^k.

When the multigraph Q_k has been burnt we can begin to perform the removal

operation. We remove all edges that have been declared burnt and remove vertices in $Q_k \setminus R_k^k$ that have been declared burnt.

The subgraph of Q_k that this leaves is defined to be S_k. See figures A.3, A.6 and A.7 for examples of these graphs.

Observe that as $R_k^k \cup R_{k+1}^k \in V(Q_k)$ and no vertices in R_k^k are allowed to be removed, which prevents any vertex in R_{k+1}^k from being burnt, we have proved by induction that $R_k^k \cup R_{k+1}^k \in S_k$ and $\mathfrak{E}_k \in S_k$ for all $k \geq 1$ as we claimed earlier.

The reason that we need these restrictions on the burning and removal is so that when we construct the multigraph S_{k+1} we have a standardised construction and have a set of vertices that have been declared burnt so that the burning of a multigraph process can be started.

The purpose of the edges $\mathfrak{E}_k$, as we will shortly see, is so that given S_k we can compute how the burning can occur in R_k when we are given a set of R_{k+1} that is burnt without explicitly knowing the configuration on R_{k+1}.

Remark A.5. *Note that in this construction at each step of the inductive process we do not need to know where S_k came from only that it contains $R_k^k \cup R_{k+1}^k$, and certain edges between these vertices.*

A.2 Properties of S_k.

Remark A.6. *This construction respects planarity, in the sense that if G is a planar graph than S_k is also a planar graph for all $k \geq 1$.*

Let $t_k + 1$ be the index of the phase when a vertex in R_k is burnt for the first time. Let π be the projection map that gives the phase that a vertex burnt in.

Lemma A.7. *For an underlying recurrent sandpile η and for all $k \geq 1$,*
(i) The vertices of S_k are given by the union of vertices, v, which are unburnt at the end of phase t_{k+1}, i.e $v \in \bigcup_{i=1}^k R_i$ such that $\pi(\phi_R(\eta, v)) > t_{k+1}$ and the vertices, w, in R_k that have been burnt by ϕ_R, i.e $\pi(\phi_R(\eta, w)) \leq t_{k+1}$.
(ii) Suppose we are given a set of vertices that are burnt in R_k and R_{k+1} and we followed the burning based on rungs procedure until the next time a vertex of R_{k+1} needs to be burnt. The pair $(\eta(k), S_k)$ determines the set of vertices in R_k that are burnt at this time.

Proof. Firstly observe that (i) is just a special case of (ii) where no additional vertices are declared burnt. However we believe the argument is more apparent by doing this case separately.

(i) This will follow by induction on k and from the observation that S_k is just a restatement of the sandpile configuration at the end of phase t_{k+1}. To see this we will follow the process of burning Q_k inductively.

For Q_1 we start with the sink being burnt. To see why the set of vertices that are burnt when Q_1 has been burnt is equal to the set of vertices that are assigned a label with a phase $\leq t_2 = 2$ by ϕ_R we need to look at how we constructed the value of the labels on edges.

Look at the label on an edge directed out of v and we see it is equal to the number of neighbouring vertices that need to be burnt in order for v to burn in the burning procedure. Now each time a neighbour of v burns in Q_1 an edge connecting to v will burn as a result, this means that the value of the label on an edge leaving v is reduced by one. Hence at any stage of the burning procedure on Q_1, the value of the label on an edge leaving v in Q_1 is equal to the number of unburnt neighbours of v that need to be burnt before v can burn under the burning based on rungs procedure. Therefore an edge leaving v will have a label of zero if and only if the number of unburnt neighbours of v is less than or equal to the height of the sandpile at v and thus a vertex burns in Q_1 if and only if $\pi(\phi_R(\eta, v)) \leq t_2$. Therefore (i) holds for S_1.

Now suppose S_{k-1} satisfies (i). Then using the construction of Q_k we can identify each vertex to one that is unburnt at the end of phase t_k or to one that has burnt in R_k. By the previous observation on the number of arrows on an edge it again follows that the process of burning Q_k to produce S_k will force it to satisfy property (i). Hence by induction (i) does hold for all $k \geq 1$.

(ii) We now consider what we get if given an extra set in R_k and R_{k+1} that are burnt. As these vertices will have a corresponding vertex in S_k we can directly set these vertices as burnt. Now by following the above procedure we can again create a set of vertices in R_k that will burn as a result and by the same argument as when no vertices in R_{k+1} were burnt we see this corresponds to the same set of vertices that will be declared as burnt by running the burning based on rungs procedure until a new vertex in R_{k+1} needs to burn. Hence S_k encodes the burning as claimed. □

Definition A.8. *Let $\mathcal{S}$ be the set of all possible labelled, directed multigraphs that can arise in our construction. For $k \geq 2$ define $\psi : \mathcal{R}_G \times \mathcal{S} \to \mathcal{S}$ by $\psi(\eta(k), S_{k-1}) := S_k$. This is possible as the only information we used in the construction of S_k was $\eta(k)$ and S_{k-1}.*

Next we define $\Phi := \{\varphi_q : q \geq 1\}$ by defining each $\varphi_q : \mathcal{R}_{G\times[1,q]} \to \mathcal{R}_G \times \mathcal{S}$. We can define the map φ_1 by the construction of S_1, namely $\varphi_1(\eta(1)) := (\eta(1), S_1)$. Then φ_k, for $k \geq 2$, can be defined inductively by

$$\varphi_k\Big(\eta(1,k)\Big) := \Big(\eta(k), \psi\Big(\eta(k), \varphi_{k-1}\big(\eta(1,k-1)\big)\Big)\Big) = (\eta(k), S_k).$$

Lemma A.9. *The triple $(\mathcal{S}, \Phi, \psi)$ satisfies Definition A.2.*

Proof. The statements of (ii) and (iii) follow immediately from the definitions of the maps.

(i) Suppose $\varphi_{q_1}(\eta_1[1,q_1]) = \varphi_{q_2}(\eta_2[1,q_2]) = (\eta_1(q_1),\sigma_1)$, then this implies that $\eta_1(q_1) = \eta_2(q_2)$ by property (ii). First let us check whether a recurrent configuration that began with η_1 could be continued by the configuration $\zeta(1)$ on R_{q_1+1}. By applying Lemma A.7(i) to $(\eta_1(q_1),\sigma_1)$ we can find the set of vertices that are burnt in R_{q_1} when R_{q_1+1} is about to burn for the first time. We can then apply the burning rule of the burning based on rungs procedure to deduce which vertices in $\zeta(1)$ can burn. Now by repeated application of Lemma A.7(ii) we can deduce the subset of $\zeta(1)$ that is burnt when a vertex of R_{q_1+2} needs to burn. By taking R_{q_1+2} to have a maximal configuration the whole of R_{q_1+2} would burn in the next step. Then again by iteratively using the burning rule on $\zeta(1)$ and Lemma A.7 on σ_1 we can continue burning $\zeta(1)$. We can then conclude that $\zeta(1)$ is an allowed extension of η_1 if and only if by the end of this process the whole of R_{q_1+1} has been declared burnt.

Then for each $k \geq q_1$ we can inductively define $\sigma_{k+1} := \psi(\zeta(k+1),\sigma_k)$ and this process can be repeated inductively to determine if a configuration $\zeta(k+2)$ is allowed.

Moreover as $\eta_1(q_1) = \eta_2(q_2)$ and both η_1 and η_2 create the multigraph σ_1 if we had run this argument with η_1 replaced by η_2 we would have equivalently found that any ζ that was an allowed extension of η_1 was also an allowed extensions of η_2. Thus proving property (i). □

If the graphs we have created were left like this we would be adding up to $|V(G)|-1$ vertices at each step and this would not reduce the information that we must retain so would not help us investigate the sandpile. However the advantage of encoding by multigraphs is that we can find that much of the information in the graphs is redundant when we are only interested in the state of burning in R_k. Thus we can simplify the graph to reflect this fact. See figures A.7 and A.8 for an example of how a multigraph may be simplify.

Remark A.10. *By Lemma A.7 any sandpile that can give rise to the pair (η,σ) has the same set of recurrent sandpile extensions. Therefore it also makes sense to talk of extension of the pair (η,σ), which we will now formally define.*

Definition A.11. *Given a coding triple $(\mathcal{T},\Upsilon,\rho)$ and a pair (η,σ) for $\eta \in \mathcal{R}_G$, $\sigma \in \mathcal{T}$ we define the* recurrent sandpile extensions of (η,σ). *By property (i) of a coding triple the pair (η,σ) specify a unique set of sandpiles that can extend any configuration that yielded this pair. Moreover by Lemma A.7 given (η,σ) and the maps Υ and ρ we can determine if a sandpile ζ can extend a configuration that gave rise to the pair without needing to know any more information about the sandpile configuration that gave rise to it. The set of all valid ζ that this produces is defined to be the extensions of (η,σ).*

Definition A.12. *We say two labelled, directed multigraphs, σ,σ' are* equivalent, *denoted $\sigma \sim \sigma'$, if for all $\eta \in \mathcal{R}_G$ we have that (η,σ) and (η,σ') allow the same recurrent sandpile extensions.*

Note that $\sim$ defines an equivalence relation.

Definition A.13. *Given a coding triple $(\mathcal{T}, \Upsilon, \rho)$ A map $\chi : \mathcal{T} \to \mathcal{T}$ is called a* simplification *map if for all $\sigma \in \mathcal{T}$ we have that $\sigma \sim \chi(\sigma)$ and there is a natural extension of ρ such that $\rho : \mathcal{R}_G \times (\mathcal{T} \cup \chi(\mathcal{T})) \to \mathcal{T} \cup \chi(\mathcal{T})$ is well defined.*

Also for a simplification map χ define the set $\mathcal{S}' = \mathcal{S}'(\chi) := \{\chi(S_k) : S_k \in \mathcal{S}\}$.

Let χ be a simplification map with respect to the triple $(\mathcal{S}, \Phi, \psi)$. In this setting the requirement for the existence of an extension of ψ to be well defined would be equivalent to the requirement that $S_k|_{R_k^k \cup R_{k+1}^k} = \chi(S_k)|_{R_k^k \cup R_{k+1}^k}$ for all $S_k \in \mathcal{S}$. As by Remark A.5 this would allow the inductive construction of S_k from $(\eta(k), S_{k-1})$ to be applied to $(\eta(k), \chi(S_{k-1}))$. Hence the previous definition of the map ψ can be extended to be well defined on the larger domain of $\mathcal{R}_G \times (\mathcal{S} \cup \mathcal{S}')$.

Using this extended definition we claim the following property holds,

$$\chi(\psi(\eta(k), S_{k-1})) \sim \chi\Big(\psi(\eta(k), \chi(S_{k-1}))\Big)$$

Proof of claim. Firstly observe that as $(\eta(k), S_{k-1})$ and $(\eta(k), \chi(S_{k-1}))$ have the same set of extensions they must have the same subset of extensions that have the next step as $\eta(k+1)$, therefore it follows that $\psi(\eta(k), S_{k-1}) \sim \psi(\eta(k), \chi(S_{k-1}))$.

Thus

$$\chi(\psi(\eta_k, S_{k-1})) = \chi(S_k) \sim S_k = \psi(\eta_k, S_{k-1}) \sim \psi(\eta(k), \chi(S_{k-1})).$$

□

Definition A.14. *Define $\psi' := \chi \circ \psi : \mathcal{R}_G \times (\mathcal{S} \cup \mathcal{S}') \to \mathcal{S}'$ and then setting $\varphi_1' := \varphi_1$, we can also define inductively*

$$\varphi_k'\Big(\eta(1,k)\Big) := \Big(\eta(k), \psi'\Big(\eta(k), \varphi_{k-1}'\big(\eta(1,k-1)\big)\Big)\Big).$$

Let $\Phi' = \{\varphi_k' : k \geq 1\}$.

Lemma A.15. *The triple $(\mathcal{S}', \Phi', \psi')$ satisfies Definition A.2.*

Proof. (i) If $\psi'_{q_1}(\eta_1[1, q_1]) = \psi'_{q_2}(\eta_2[1, q_2]) = (\eta_1(q_1), S'_{q_1})$ then by equivalence the possible recurrent extensions are the same as though allowed by $(\eta_1(q_1), S_{q_1})$ which uniquely determines the allowed extensions of η_1 and η_2.

The validity of (ii) and (iii) again immediately follow from the definition of the maps. □

There are many ways in which these multigraphs can simplify and we give a selection of possible rules. This is not a comprehensive list but it should hopefully give the reader an idea of what is possible.

Note some simplifications may introduce into the graph configurations that are not desirable, for instance having multiple edges directed from v to w which may alter the

maximum degree of a vertex or connecting vertices in a way that makes a planar graph into one that is non-planar. Such alterations may not be advisable for future proofs so the exact list of simplifications that can be used will depend on how the proofs involving the encoding will work.

Simplifications. These rules are applied to a graph S'_k except for vertices in, and edges between, $R^k_k \cup R^k_{k+1}$.

1) A vertex without a edge directed out of it can be removed.

2) Remove any edge whose start or end vertex has been removed.

3) Remove the edge from v to w if the vertex v can not be declared burnt whilst w is still declared unburnt.

4) If the only edge leaving v is to w then we can remove the edge from w to v, if it exists.

5) If the edge from w to v has label one and it is the only edge directed into v and it is also the only edge directed out of w, then w can be deleted and any edges that were directed into w are now connected to v, with their labels unchanged.

6) Delete any vertex and edges that never need to be used in any burning of S'_k, because the vertices in R_k they help to burn can always be reached via an alternative path that requires the same set of initial vertices to have burnt.

Each of these rules will give rise to a simplification map and, by the fact that $\sim$ is an equivalence relation, any combination of the rules will also be a simplification map. To demonstrate the type of argument required to prove this we consider rule (5).

Proof that applications of rule (5) *define a simplification map.* Firstly the vertex and edges set of $R^k_k \cup R^k_{k+1}$ are not changed by this rule so this property is satisfied. It suffices to show that given equivalent inputs the continuation of the burning in the graphs will be identical.

Suppose $v, w \in S_k$ satisfied the conditions of (5) and let $\chi(S_k)$ be the multigraph found via an application of rule (5). Now consider how the burning may occur in S_k. If an edge $e_1 \in E_1$, the set of edges directed towards w, is declared burnt then the edge leaving w will now have label 0 and so w will be declared burnt. The only impact this has is that the edge directed towards v will now be declared burnt. This means the set of edges leaving v, E_2, will have their labels reduced by one.

Now let us consider what occurs in $\chi(S_k)$. The edge e_1 is now connected to v so when e_1 is burnt the label of edges in E_2 will be decreased by one.

Therefore as e_1 is the only edge leaving w and is the only edge entering v the set of edges $E_1 \cup E_2$ completely determine how w and v interact with all other vertices. Given the same initial input both multigraphs will output the same labels on $E_1 \cup E_2$ and thus $S_k \sim \chi(S_k)$.

From this argument it also follows that if we could apply rule (5) multiple times then the output would still be equivalent to the original graph. □

Conjecture A.16. *There exists a simplification map, χ, and a constant, $c > 0$, such that for any sandpile on $[1, N] \times \mathbb{N} \subset \mathbb{Z}^2$ the induced multigraphs, $\chi(S_k)$, each contains at most cN vertices.*

If this conjecture could be proven then we believe that it would be a big step towards proving Conjecture A.3. We also believe that similar bounds would exist in higher dimensions and whilst we are unable to provide an estimate for these cases, we hope understanding this encoding would be a step in the right direction.

We conclude this appendix by giving an example of the construction and a possible simplification of a sandpile configuration. Suppose we have a sandpile on the graph $[1, 6] \times \mathbb{N} \subset \mathbb{Z}^2$ with rung 1 connected to the sink. Then we will construct the multigraphs for the first three rungs. In the following figures the label of an edge is given by the number of arrows on it. To be able to determine the direction of an edge with label zero we will use the convention that edges that are curved above (resp. below) their end points are oriented from left to right (resp. right to left) and edges that are curved to the left (resp. right) of their end points are directed upwards (resp. downwards).

$\eta(1,3) =$

0	2	1	
3	3	3	
1	2	3	
2	3	2	
1	0	2	
1	2	2	

Figure A.1: *An example of a sandpile configuration on the first three rungs of the graph* $\{1, 2, 3, 4, 5, 6\} \times \mathbb{N}$

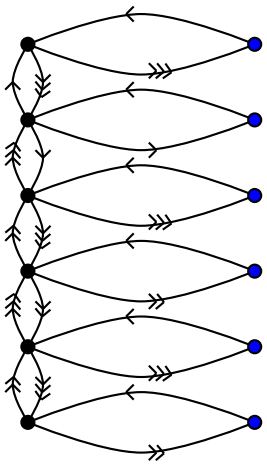

Figure A.2: *Encoding of $\eta(1)$ as T_1. Vertices are coloured according to which set they are in; Black is for R_1^1 and blue is for R_2^1*

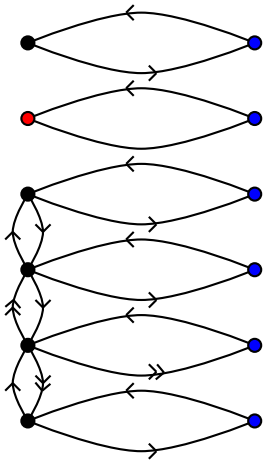

Figure A.3: *S_1, where the red vertices are ones that have been declared burnt and the blue vertices are those in the set R_2^1.*

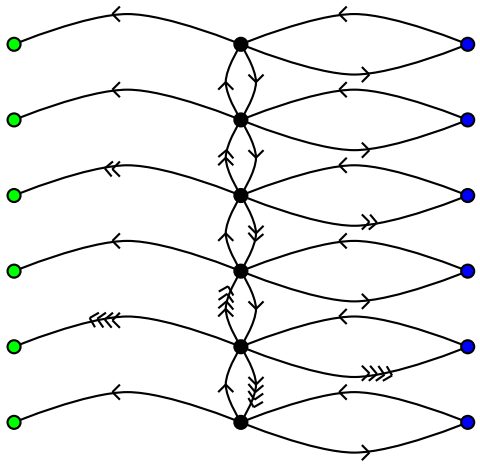

Figure A.4: *Encoding of $\eta(2)$ as T_2. Vertices are coloured according to which set they are in; Green is for R_1^2, black is for R_2^2 and blue is for R_3^2*

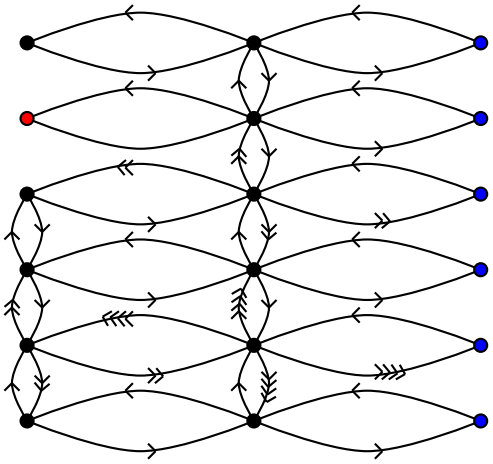

Figure A.5: *Q_2, with red vertices representing vertices that have burnt and blue vertices are those in the set R_3^2.*

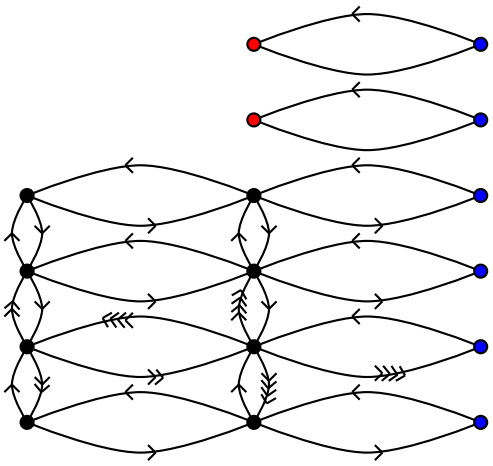

Figure A.6: *S_2, with red vertices representing vertices that have burnt and the blue vertices are those in the set R_3^2.*

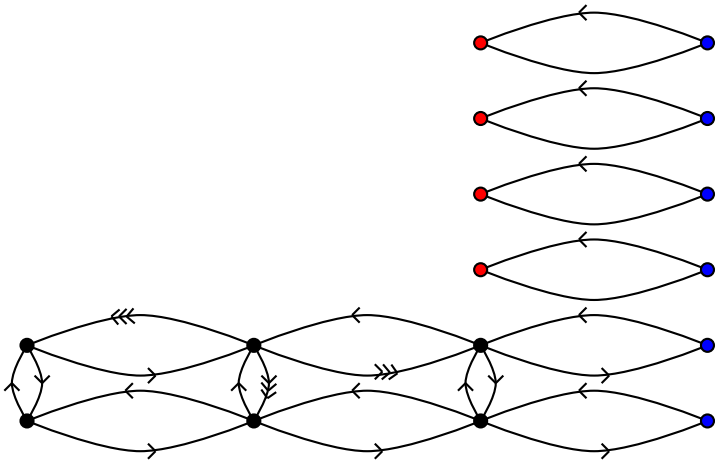

Figure A.7: *S_3, with red vertices representing vertices that have burnt and the blue vertices are those in the set R_4^3.*

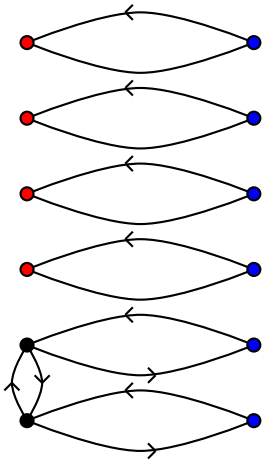

Figure A.8: *A possible simplification of S_3, with red vertices representing vertices that have burnt and the blue vertices are those in the set R_4^3.*

BIBLIOGRAPHY

[1] Michael Aizenman, Almut Burchard, Charles M. Newman, and David B. Wilson. Scaling limits for minimal and random spanning trees in two dimensions. *Random Structures Algorithms*, 15(3-4):319–367, 1999. Statistical physics methods in discrete probability, combinatorics, and theoretical computer science (Princeton, NJ, 1997).

[2] Siva R. Athreya and Antal A. Járai. Infinite volume limit for the stationary distribution of abelian sandpile models. *Comm. Math. Phys.*, 249(1):197–213, 2004.

[3] P. Bak, C. Tang, and K. Wiesenfeld. Self-organized criticality: An explanation of the 1/f noise. *Physics Review Letters*, 59:381–384, 1987.

[4] Martin T. Barlow and Robert Masson. Exponential tail bounds for loop-erased random walk in two dimensions. *Ann. Probab.*, 38(6):2379–2417, 2010.

[5] Martin T. Barlow and Robert Masson. Spectral dimension and random walks on the two dimensional uniform spanning tree. *Comm. Math. Phys.*, 305(1):23–57, 2011.

[6] Itai Benjamini. Large scale degrees and the number of spanning clusters for the uniform spanning tree. In *Perplexing problems in probability*, volume 44 of *Progr. Probab.*, pages 175–183. Birkhäuser Boston, Boston, MA, 1999.

[7] Itai Benjamini, Russell Lyons, Yuval Peres, and Oded Schramm. Uniform spanning forests. *Ann. Probab.*, 29(1):1–65, 2001.

[8] David Bleecker and George Csordas. *Basic partial differential equations*. International Press, Cambridge, MA, 1996.

[9] Gabriel D. Carroll and David Speyer. The cube recurrence. *Electron. J. Combin.*, 11(1):Research Paper 73, 31 pp. (electronic), 2004.

[10] R. Courant, K. Friedrichs, and H. Lewy. Über die partiellen Differenzengleichungen der mathematischen Physik. *Math. Ann.*, 100(1):32–74, 1928.

[11] Deepak Dhar. Self-organized critical state of sandpile automaton models. *Phys. Rev. Lett.*, 64(14):1613–1616, 1990.

[12] Deepak Dhar. Theoretical studies of self-organized criticality. *Phys. A*, 369(1):29–70, 2006.

[13] Peter G. Doyle and J. Laurie Snell. *Random walks and electric networks*, volume 22 of *Carus Mathematical Monographs*. Mathematical Association of America, Washington, DC, 1984.

[14] T. Feder and M. Mihail. Balanced matroids. In *In Proceedings of the Twenty-Fourth Annual ACM Symposium on Theory of Computing*, pages 23–38. ACM, New York, 1992.

[15] Samuel L. Gamlin and Antal A. Járai. Anchored burning bijections on finite and infinite graphs. *Electron. J. Probab.*, 19:no. 117, 23, 2014.

[16] Olle Häggström. *Aspects of Spatial Random Processes.* PhD thesis, Department of Mathematics Göteborg, 1994.

[17] Olle Häggström. Random-cluster measures and uniform spanning trees. *Stochastic Process. Appl.*, 59(2):267–275, 1995.

[18] Alexander E. Holroyd, Lionel Levine, Karola Mészáros, Yuval Peres, James Propp, and David B. Wilson. Chip-firing and rotor-routing on directed graphs. In *In and out of equilibrium. 2*, volume 60 of *Progr. Probab.*, pages 331–364. Birkhäuser, Basel, 2008.

[19] Eberhard Hopf. An inequality for positive linear integral operators. *J. Math. Mech.*, 12:683–692, 1963.

[20] J. Ben Hough, Manjunath Krishnapur, Yuval Peres, and Bálint Virág. Determinantal processes and independence. *Probab. Surv.*, 3:206–229, 2006.

[21] A. Járai. Rate of convergence estimates for the zero dissipation limit in abelian sandpiles. *Preprint*, 2011. arXiv:110.1437.

[22] A. Járai. Sandpile models. *Preprint*, 2014. arXiv:1401.0354.

[23] A. Járai and N. Werning. Minimal configurations and sandpile measures. *Journal of Theoretical Probability*, 2013.

[24] A. A. Járai. Abelian sandpiles: an overview and results on certain transitive graphs. *Markov Process. Related Fields*, 18(1):111–156, 2012.

[25] A. A. Járai and R. Lyons. Ladder sandpiles. *Markov Process. Related Fields*, 13(3):493–518, 2007.

[26] Antal A. Járai and Frank Redig. Infinite volume limit of the abelian sandpile model in dimensions $d \geq 3$. *Probab. Theory Related Fields*, 141(1-2):181–212, 2008.

[27] Antal A. Járai, Frank Redig, and Ellen Saada. Approaching criticality via the zero dissipation limit in the abelian avalanche model. *J. Stat. Phys.*, 159(6):1369–1407, 2015.

[28] Antal A. Járai and Nicolás Werning. Minimal configurations and sandpile measures. *J. Theoret. Probab.*, 27(1):153–167, 2014.

[29] M. Jeng, G. Piroux, and P. Ruelle. Height variables in the abelian sandpile model: scaling fields and correlations.. *Journal Stat. Mech. Theory Exp.*, 10, 2006. P10015.

[30] Richard Kenyon. The asymptotic determinant of the discrete Laplacian. *Acta Math.*, 185(2):239–286, 2000.

[31] Richard W. Kenyon and David B. Wilson. Boundary partitions in trees and dimers. *Trans. Amer. Math. Soc.*, 363(3):1325–1364, 2011.

[32] Gregory F. Lawler. Loop-erased self-avoiding random walk in two and three dimensions. *J. Statist. Phys.*, 50(1-2):91–108, 1988.

[33] Gregory F. Lawler. *Intersections of random walks. 2013 reprint.* Probability and its Applications. Birkhäuser Boston Inc., Boston, MA, 1991.

[34] Gregory F. Lawler and Vlada Limic. The Beurling estimate for a class of random walks. *Electron. J. Probab.*, 9:no. 27, 846–861 (electronic), 2004.

[35] Gregory F. Lawler and Vlada Limic. *Random walk: a modern introduction*, volume 123 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, Cambridge, 2010.

[36] Lionel Levine and Yuval Peres. Strong spherical asymptotics for rotor-router aggregation and the divisible sandpile. *Potential Anal.*, 30(1):1–27, 2009.

[37] R. Lyons and Y. Peres. *Probability on Trees and Networks.* Cambridge University Press, In preparation. Current version available at http://mypage.iu.edu/ rdlyons/.

[38] Russell Lyons, Benjamin J. Morris, and Oded Schramm. Ends in uniform spanning forests. *Electron. J. Probab.*, 13:no. 58, 1702–1725, 2008.

[39] Neal Madras and Gordon Slade. *The self-avoiding walk.* Modern Birkhäuser Classics. Birkhäuser/Springer, New York, 2013. Reprint of the 1993 original.

[40] S.N Majumdar and D. Dhar. Equivalence between the Abelian sandpile model and the q→0 limit of the Potts model. *Journal of Physics A*, 185:129–145, 1991.

[41] SN Majumdar and Deepak Dhar. Height correlations in the abelian sandpile model. *Journal of Physics A: Mathematical and General*, 24(7):L357, 1992.

[42] Robert Masson. The growth exponent for planar loop-erased random walk. *Electron. J. Probab.*, 14:no. 36, 1012–1073, 2009.

[43] Ben Morris. The components of the wired spanning forest are recurrent. *Probab. Theory Related Fields*, 125(2):259–265, 2003.

[44] Robin Pemantle. Choosing a spanning tree for the integer lattice uniformly. *Ann. Probab.*, 19(4):1559–1574, 1991.

[45] T. Kyle Petersen and David Speyer. An arctic circle theorem for Groves. *J. Combin. Theory Ser. A*, 111(1):137–164, 2005.

[46] V.B. Priezzhev. Structure of two-dimensional sandpile. i. height probabilities. *Journal of Statistical Physics*, 74(5-6):955–979, 1994.

[47] Frank Redig. Mathematical aspects of the abelian sandpile model. In *Mathematical statistical physics*, pages 657–729. Elsevier B. V., Amsterdam, 2006.

[48] Klaus Schmidt and Evgeny Verbitskiy. Abelian sandpiles and the harmonic model. *Comm. Math. Phys.*, 292(3):721–759, 2009.

[49] Javier Segura. Monotonicity properties and bounds for the chi-square and gamma distributions. *Appl. Math. Comput.*, 246:399–415, 2014.

[50] David Bruce Wilson. Generating random spanning trees more quickly than the cover time. In *Proceedings of the Twenty-eighth Annual ACM Symposium on the Theory of Computing (Philadelphia, PA, 1996)*, pages 296–303. ACM, New York, 1996.