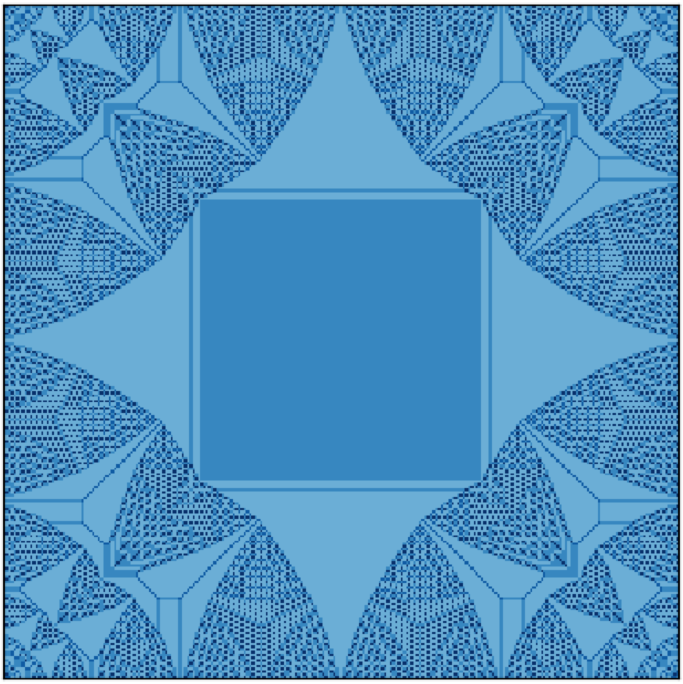
The Abelian sandpile model
Connecting probabilistic and geometric approaches

Master Thesis Mathematical Sciences

E.C.J. van der Laan



25 June 2024

Supervisor: Dr. W.M. Ruszel
Second reader: Prof. dr. G.L.M. Cornelissen

Utrecht University

Abstract

The Abelian sandpile model (ASM) is a toy model from statistical physics used to study self-organized criticality. It has been studied both in terms of a Markov chain and in terms of divisors. In this thesis we will introduce both this probabilistic and graph geometric approach to the ASM and discuss some connections between the two approaches. We will discuss a new Torelli theorem for graphs studied by Griffith in 2023 and a newfound relation between the non-special divisors from this theorem and the minimal configurations of the Markov chain. We will give a new version of this Torelli theorem, which will for certain graphs state that two graphs are isomorphic if and only if their groups of recurrent configurations admit an isomorphism that induces a bijection on minimal configurations.

Contents

Introduction

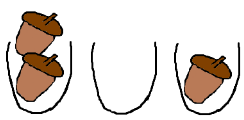
In 1987, Bak, Tang and Weisenfeld published a paper in which they introduced the first example of a dynamical system that has "self-organized criticality", a system that has a critical point as an attractor on a lattice graph [BTW87]. Models displaying self-organized criticality are characterised by power-law behaviour of e.g. the correlation function without fine-tuning any parameter such as the temperature. In nature we can observe power-law behaviour for example in earthquakes [MWZ19]. The model of Bak, Tang and Weisenfeld has been generalized to any graph in 1990 by Deepak Dhar, who discovered its Abelian properties and called it "the Abelian sandpile model" [Dha90]. We will now give an illustrative example of the model.

Imagine a squirrel that wants to stash away its found acorns. For this purpose, it has dug N pits in a row. To distribute the acorns, it chooses one of the pits uniformly at random and puts an acorn in that pit. There is one restriction however: these pits only have space for one acorn. Therefore, if the squirrel wants to put a second acorn in the pit, it doesn't fit and the squirrel has to redistribute. For this redistribution, it puts one of the two acorns in the pit left and one in the pit right of the current overflowing pit. When this happens at one of the pits at one of the ends of the row, it puts one acorn in the neighbouring pit and disposes of another acorn. This distributing behaviour can lead to an "avalanche", where the pit left and/or right also has to be redistributed after having gotten an extra acorn from its neighbouring pit. This redistributing stops when all pits have at most one acorn in it. A fixed distribution of acorns in pits is called a configuration. A configuration for which the redistribution process has ended is called a stable configuration. Let us look at an example of this redistribution process.

Example. *If we take $N = 3$, the starting configuration looks like this:*

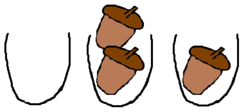
Now the squirrel chooses the left pit and after that it chooses the right pit. The configuration then looks like this:

Now if the squirrel chooses the leftmost pit again, we get the following configuration:

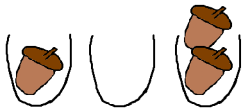


The left pit is now too full, so the squirrel redistributes by taking two of the acorns, putting one in the middle pit and disposes of one acorn. After this redistribution, we see that the stable configuration we get is the following:

With adding an acorn to the middle pit in the next step, we will experience the avalanche effect. The configuration will look like this:



The squirrel now redistributes two of the acorns from the middle pit:



Upon this redistribution, the right pit is now too full, so the squirrel has to redistribute further. It takes two acorns from the right pit, puts one of them in the middle pit and disposes of the other. The final configuration will then look like this:

$\diamondsuit$

Whilst the model has its roots in theoretical physics, connections to a bunch of different fields in mathematics have since been found. Via the theory of Markov chains, a lot of probabilistic features of the model have been determined, such as the recurrent states, the mass of states and avalanche clusters [Red05]. A relation between the Tutte polynomial and the generating function of the mass of a configuration has been found [Big99b]. In general, it is difficult to describe all recurrent configurations. There is a burning bijection mapping the recurrent configurations to spanning trees [MD92].

Minimal configurations are a subset of recurrent configurations that have attracted special attention. In Theorem 1.10 of this thesis, we will prove that the minimal configurations are in bijection with acyclic orientations with a unique source. On a Ferrers graph, the minimal configurations are in 1-1 correspondence to the set of EW-tableaux on the corresponding Ferrers diagram [SSS18]. On the hexagonal lattice, the minimal configurations and the set of maximally oriented spanning trees on the triangular sublattice are in bijection [PK97]. The concentration of minimal configurations and the probability that the height of two sites at distance r would have minimum values have been calculated in 1999 by Majumdar and Dhar [MD99]. On an infinite graph G, the limiting probability on G for generalized minimal configurations exists [JW12]. One can compute correlations and probabilities of events with stationary distributions of the Markov chain. The minimal configurations are used to prove that a weak limit of the sandpile measure exists [JW12].

In algebraic graph theory, a group called the "Jacobian group" appeared in the study of flows and cuts in graphs [BLN97], an object inspired by the Jacobian group on curves in the field of algebraic geometry. With this Jacobian, efforts have been made to translate theorems from algebraic geometry to theorems on the alike objects on graphs. The first example is an analogue of Abel-Jacobi's theorem introduced in [BLN97], which states that the Abel-Jacobi map, which is a map that relates the graph to its Jacobian, is birational and surjective. Later, in 2007, Baker and Norine proved a graph analoguous version of the Riemann-Roch theorem [BN07]. This theorem states that for any divisor D on a graph G of genus g, we have

$$rank(D) - rank(K_G) = \deg(D) - g + 1,$$

where K_G is the canonical divisor on G. Some efforts have also been made to use the Jacobian on finite graphs for cryptographic purposes [Sho10; Big07], which have until now only provided insecure cryptoschemes. Connections between divisors on curves and divisors of graphs are studied, using techniques from tropical geometry [Bak08; BJ16].

Another point of interest is translating Torelli's theorem to the graph setting. At its core, this theorem in algebraic geometry states that if the Jacobian of curves admits a certain isomorphism, the corresponding curves are isomorphic as well [Mil86]. In [BLN97, Sec. 3] the authors noted that a naive analogue would not work, i.e. isomorphic Jacobians do not imply isomorphic graphs. A Torelli theorem that holds for 3-connected graphs has been proven in 2010 by Su and Wagner [SW10] and by Caporaso and Viviani [CV10]. In 2023, Griffith proved a Torelli theorem that holds for 2-connected graphs, where the Jacobians should admit a discrete theta divisor isomorphism for graphs to be isomorphic [Gri23]. One of the contributions of this thesis is a connection between the discrete theta divisors and minimal configurations. This will allow us to extend this Torelli theorem to the probabilistic setting in Theorem 5.5.

For a specific overview of all bijections studied in this thesis, one can consult the figure on the next page. In this picture, a blue arrow denotes an original contribution of this thesis. A purple arrow implies that the bijection is not new, but we have included an original proof. A black arrow denotes that a known result and proof have been (re)formulated in this thesis.

Outline

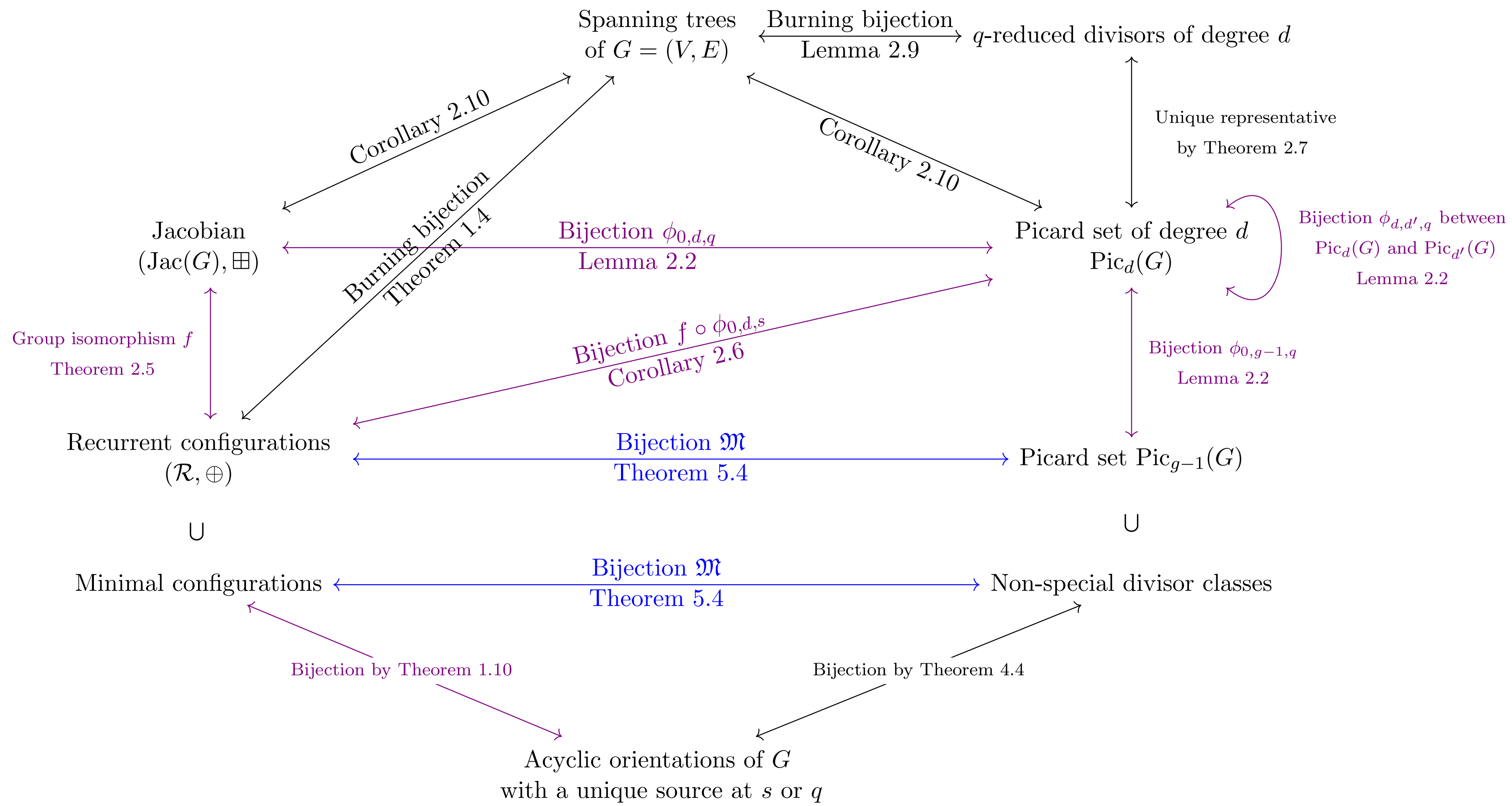
In the first chapter we will formally introduce the Abelian sandpile model, which we will express in terms of a Markov chain. We will look into Dhar's burning algorithm, which gives us a bijection between the recurrent configurations of the Markov chain and spanning trees of the graph. A group action on the recurrent states of the Markov chain is defined. The weight of recurrent states, specifically of minimal and non-minimal configurations is determined, and finally a bijection between acyclic orientations and minimal configurations is proven.

In the second chapter, divisors, Picard sets and the Jacobian on graphs are introduced, which are alike the objects known under the same name in algebraic geometry. A new isomorphism between the Jacobian and the group of recurrent states of section one is proven. At the end of chapter two we introduce q-reduced divisors, which uniquely represent equivalence classes in the Picard sets and the Jacobian and which are also in bijection with the spanning trees of the graph.

In the third chapter we will treat two different algorithmic approaches to calculate the identity element of the group of recurrent configurations.

In chapter four, we will look into a Torelli theorem on graphs by [Gri23]. We will define the discrete theta divisor and (non-)special divisors. We will highlight and rephrase parts from the paper of [Gri23].

In chapter five, we will determine some previously unknown properties of special and non-special divisors. We will find a new connection between non-special divisors and minimal configurations. This connection will allow us to obtain a new Torelli theorem for graphs using the objects from the probabilistic approach from the first section.


Spanning trees of $G = (V, E)$
Burning bijection
Lemma 2.9
q-reduced divisors of degree d
Corollary 2.10
Corollary 2.10
Unique representative
by Theorem 2.7
Burning bijection
Theorem 1.4
Jacobian
$(\mathrm{Jac}(G), \boxplus)$
Bijection $\phi_{0,d,q}$
Lemma 2.2
Picard set of degree d
$\mathrm{Pic}_d(G)$
Bijection $\phi_{d,d',q}$ between
$\mathrm{Pic}_d(G)$ and $\mathrm{Pic}_{d'}(G)$
Lemma 2.2
Group isomorphism f
Theorem 2.5
Bijection $f \circ \phi_{0,d,s}$
Corollary 2.6
Bijection $\phi_{0,g-1,q}$
Lemma 2.2
Recurrent configurations
$(\mathcal{R}, \oplus)$
Bijection $\mathfrak{M}$
Theorem 5.4
Picard set $\mathrm{Pic}_{g-1}(G)$
$\cup$
$\cup$
Minimal configurations
Bijection $\mathfrak{M}$
Theorem 5.4
Non-special divisor classes
Bijection by Theorem 1.10
Bijection by Theorem 4.4
Acyclic orientations of G
with a unique source at s or q

Chapter 1

The probabilistic approach: Markov chain and minimal configurations

In this chapter, we will introduce the Abelian sandpile model. This model can be written in terms of a Markov chain. We look into the recurrent states of the Markov chain. The set of recurrent states together with a group action $\oplus$ will form an Abelian group. A bijection between the recurrent states and the spanning trees of the graph, the so-called Dhar's burning bijection, will be discussed. Finally, we will discuss minimal configurations and their properties. This chapter is mostly based on [Red05] and [Jár18].

1.1 The Abelian sandpile model

Let $G = (V \cup \{s\}, E)$ be a finite, connected graph without loops. One of the vertices will be marked. We will call this vertex s the *sink*.

Usually we will consider regular, non-weighted, non-oriented graphs, where between any two vertices that are not the sink, there is at most one edge, but it is possible to define the model any finite, connected graph.

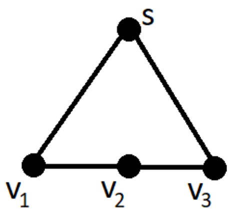
Definition. *A height configuration defined on the graph $G = (V \cup \{s\}, E)$ is given by:*

$$\eta : V \to \mathbb{Z}_{\geq 0}.$$

We call $\eta(v)$ the height of a vertex $v \in V$.

We will denote the set of all configurations by $\mathcal{H}$. In some literature, for example [Red05], only configurations with $\eta(v) \geq 1$ are considered. Although this difference might seem trivial, the two approaches sometimes have a different looking identity element, which will be further discussed in Section 3.3.1. In this chapter, we will only consider the model where all vertices start at height 0.

Example 1.1. *Let us again consider the example from the introduction. What we were previously considering as the pits, will now be the vertices V of the graph. An edge is present where the pits are neighbouring and we will connect a vertex to the sink if previously one of the acorns would be disposed by lack of a neighbour when this pit toppled. The example from the introduction would therefore translate to the following graph:*



The configuration

we now write as η *with* $\eta(v_1) = 0$ *and* $\eta(v_2) = \eta(v_3) = 1$. $\Diamond$

We say that η is *stable at vertex* v whenever $\eta(v) < \deg_G(v)$, where $\deg_G(v)$ denotes the degree of a vertex v in the graph G, i.e. the amount of edges that are incident to the vertex. We call a configuration *stable* when it is stable for all vertices except the sink. We denote the set of all stable configurations by Ω.

Example 1.2. *Consider again the graph from Example 1.1. The degree of all vertices in this graph is* 2. *The configuration*

$$\eta_1(v) = \begin{cases} 0 & \text{if } v = v_1, \\ 1 & \text{else }, \end{cases}$$

for all $v \in V$ *is therefore a stable configuration, as* $\eta(v_i) < 2$ *for all* $i \in \{1, 2, 3\}$. *An example of an unstable configuration is*

$$\eta_2(v) = \begin{cases} 2 & \text{if } v = v_1, \\ 1 & \text{else.} \end{cases}$$

$\Diamond$

To transform any configuration into a stable configuration, we want a tool to reduce the height of unstable vertices.

Definition. *Let* $G = (V \cup \{s\}, E)$ *be a finite, connected, simple graph. A toppling operator* T *at vertex* $a \in V$ *on a configuration* η *on* G *results in the height of any* $v \in V$ *is defined as:*

$$T_a(\eta)(v) = \begin{cases} \eta(a) - \deg_G(a) & \text{if } v = a, \\ \eta(v) - 1 & \text{if } v \sim a, \\ \eta(v) & \text{else,} \end{cases}$$

where $\sim$ *denotes the two vertices being neighbours in* G.

Using multiple of these toppling operations, we can go from any configuration to a stable configuration by toppling any unstable vertex and repeating that process until the height of all vertices is smaller than their degree and the configuration is stable. This procedure is called *stabilization*. We will denote the stable configuration obtained after stabilizing configuration η by $S(\eta)$. In the next lemma we will investigate whether stabilization always is a finite procedure and whether it is independent of the order of topplings. The following lemma and its proof are based on [Jár18, Thm 2.3]. We will reformulate the proof here to give a few more details.

Lemma 1.1. *[Jár18, Thm 2.3] The stabilization procedure*

$$S : \mathcal{H} \to \Omega, \eta \mapsto S(\eta),$$

is well-defined and does not depend on the order of toppling.

Proof. First, we will argue why the stabilization procedure is finite by contradiction. If the stabilization procedure would not be finite, there is a vertex $v_0 \in V$ which topples an infinite amount of times in this procedure. As the graph G is connected and finite, there is a path

$$v_0 v_1 ... v_n s,$$

where $v_i \sim v_{i+1}$ for $0 \leq i \leq n-1$ and $v_n \sim s$. As we topple v_0 an infinite amount of times, its neighbour v_1 would get an infinite amount of particles and would also have to be toppled an infinite amount of times. Following this pattern all the way through the path, we see that all of the v_i would have to be toppled an infinite amount of times. Especially, v_n has to be toppled an infinite amount of times. For every topple made at v_n, one particle will move to the sink. For these vertices to stay unstable, they must have at least a positive height at all times. But, as we are losing an infinite amount of particles by toppling v_n an infinite amount of times, this cannot stay true and we arrive at a contradiction. Therefore we conclude that after a finite amount of topplings, we will arrive at a configuration without unstable vertices.

For proving the stabilization procedure being well-defined, it is important that the operation of toppling is Abelian, i.e. first toppling vertex v and then vertex w gives the same configuration as first toppling vertex w and thereafter vertex v. Let $v, w \in V$ be two vertices at which we want to topple. If $v = w$, then clearly these topplings commute. Assume that $v \neq w$. If $v \sim w$, we get:

$$T_w(T_v(\eta))(u) = \begin{cases} \eta(u) - \deg_G(u) + 1 & \text{if } u \in \{v, w\}, \\ \eta(v) + 2 & \text{if } u \sim v \text{ and } u \sim w, \\ \eta(v) + 1 & \text{if } u \sim v \text{ or } u \sim w, \\ \eta(v) & \text{else,} \end{cases}$$

for all vertices $u \in V$. Also note that

$$T_v(T_w(\eta))(u) = \begin{cases} \eta(u) - \deg_G(u) + 1 & \text{if } u \in \{v, w\}, \\ \eta(v) + 2 & \text{if } u \sim v \text{ and } u \sim w, \\ \eta(v) + 1 & \text{if } u \sim v \text{ or } u \sim w, \\ \eta(v) & \text{else,} \end{cases}$$

for all vertices $u \in V$. We conclude that these topplings commute. This same way we can explicitly write down what happens to every vertex when toppling both v and w in the case $v \not\sim w$ to see again we get $T_v T_w = T_w T_v$.

Now, to conclude the argument of stabilization being well-defined, we have to prove that given two stabilizing sequences $v_1, ..., v_n$ and $w_1, ..., w_m$ of a configuration η for some n, m, we have that

$$T_{v_n} T_{v_{n-1}} ... T_{v_1}(\eta) = T_{w_m} T_{w_{m-1}} ... T_{w_1}(\eta).$$

First note that if η itself is stable, the two given sequences are both empty and therefore coincide. Assume that η is unstable and hence the stabilizing sequences are non-empty.
As v_1 is the first vertex at which we topple η in the first sequence, we have that $\eta(v_1) \geq \deg_G(v_1)$. This implies that in the second stabilizing sequence, this vertex also has to be toppled at some

point, so let i_1 be the smallest index such that $v_1 = w_{i_1}$. We concluded above that toppling operators are commutative, so we can move terms around to get:

$$T_{w_m}T_{w_{m-1}}...T_{w_{i_1+1}}T_{w_{i_1}}T_{w_{i_1-1}}...T_{w_1}(\eta) = T_{w_m}T_{w_{m-1}}...T_{w_{i_1+1}}T_{w_{i_1-1}}...T_{w_1}T_{w_{i_1}}(\eta).$$

Now we can repeat the same argument on the configuration $T_{v_1}(\eta) = T_{w_{i_1}}(\eta)$ to find the smallest index i_2 such that $v_2 = w_{i_2}$ and bring this to the beginning of the stabilization sequence. Repeating this argument for n times, we see that there are some pairwise distinct indices $i_3, ..., i_n$ such that $w_{i_k} = v_k$. By commutating toppling operators, we get

$$\begin{aligned} T_{w_m}T_{w_{m-1}}...T_{w_1}(\eta) &= T_{w_{j_{n-m}}}...T_{w_{j_1}}T_{w_{i_n}}...T_{w_{i_1}}(\eta) \\ &= T_{w_{j_{n-m}}}...T_{w_{j_1}}T_{v_n}...T_{v_1}(\eta) \end{aligned}$$

for some $j_1, ..., j_{n-m}$. For every v_k we have that the configuration

$$T_{v_{k-1}}...T_{v_2}T_{v_1}(\eta) = T_{w_{i_{k-1}}}...T_{w_{i_1}}(\eta)$$

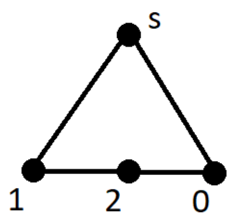
is unstable at v_k, so it should be toppled in both stabilizing sequences still, so we cannot have that $m < n$. Now, as $v_n, ..., v_1$ is a stabilizing sequence, we see that $T_{v_n}...T_{v_1}(\eta)$ is stable, we must have that $T_{w_{i_n}}...T_{w_{i_1}}$ is stable as well. We conclude that $m = n$ and have

$$T_{w_m}T_{w_{m-1}}...T_{w_1}(\eta) = T_{v_n}T_{v_{n-1}}...T_{v_1}(\eta).$$

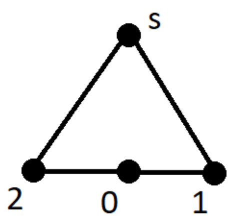
We conclude that stabilization procedure is well-defined. □

We will now look at the stabilization procedure in a concrete example.

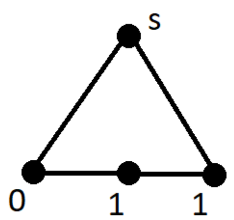
Example 1.3. *Consider the following configuration:*



The middle vertex is unstable, as it has degree 2 but also height 2. Therefore we can now topple this vertex to obtain



Now we see that the left-most vertex is unstable. Toppling this vertex results in the following configuration:



Now this configuration is stable, as the height of all vertices strictly smaller then the amount of incident edges they have. ◊

1.1.1 The Laplacian matrix

A different way to write down the stabilization is using the Laplacian matrix on $V \cup \{s\} \times V \cup \{s\}$.

Definition. *The (unnormalized) Laplacian matrix L for a simple graph $G = (V \cup \{s\}, E)$ on $V \cup \{s\} \times V\{s\}$ is*

$$L_{i,j} = \begin{cases} \deg_G(v_i) & \text{if } v_i = v_j, \\ -1 & \text{if } v_i \sim v_j, , \\ 0 & \text{else.} \end{cases}$$

for all $i, j \in \{1, ..., |V| + 1\}$ and $v_n = s$.

As we do not consider the height at the sink s and we also cannot topple at s, we will remove the row and column corresponding to s from the Laplacian matrix. We will denote this matrix by

$$L' := L_{V \times V},$$

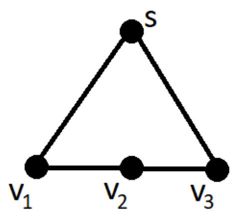
and call it the Laplacian matrix as well.

Now, if we regard a configuration η as a column vector in $\mathbb{Z}_{\geq 0}^{|V|}$, with $\eta_i = \eta(v_i)$, we can write down a stabilization procedure as an operation using the Laplacian matrix:

$$S(\eta) = \eta - L' \cdot \Delta_\eta,$$

where Δ_η is a column vector with $\Delta_\eta(v_i)$ equals the amount of times v_i has to be toppled for η to stabilize. We call Δ_η the *odometer function* of the stabilization of η. Note that by Lemma 1.1 we know that Δ_η is uniquely determined.

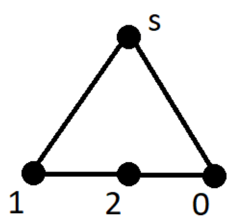
Example 1.4. *For the graph $G = (V \cup \{s\}, E)$ defined as:*



the Laplacian matrix L' is

$$L' = \begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{pmatrix}.$$

The (unstable) configuration



can be written as:

$$\eta = \begin{pmatrix} 1 \\ 2 \\ 0 \end{pmatrix}.$$

As can be seen from Example 1.3, to stabilize this graph, we topple once both v_1 and v_2 to get the stable configuration

$$S(\eta) = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}.$$

Indeed, we can write:

$$S(\eta) = \eta - L' \cdot \Delta_\eta = \begin{pmatrix} 1 \\ 2 \\ 0 \end{pmatrix} - \begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix},$$

where we see indeed that the vector Δ_η "counts" the amount of topplings of each vertex in the stabilization process. ♢

1.2 Markov chain and recurrent states

In this section, we will first introduce the general concept of Markov chain and then we will write the Abelian sandpile model defined above as a Markov chain. This basic introduction of Markov chains is based on [Nor97].

Let I be some countable set, which we will call the *state space* for the Markov chain. Let X be a random variable that takes on values in I, i.e. X is a function

$$X : \Psi \to I,$$

where we work in a probability space $(\Psi, \mathcal{F}, \mathbb{P})$. Let X be distributed via some *distribution* λ, i.e.

$$\mathbb{P}(X = i) = \lambda_i,$$

for all $i \in I$, such that $\sum_{i \in I} \lambda_i = 1$ and $\lambda_i \geq 0$ for all i.

We say that $P \in \mathbb{R}_{\geq 0}^{I \times I}$ is a *stochastic* matrix if each row $(p_{i,j} : j \in I)$ sums up to 1 and all entries are non-negative.

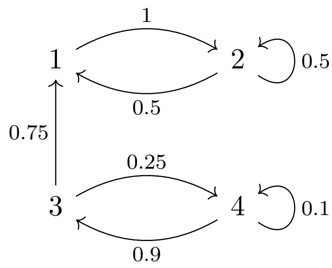
Definition. *A discrete-time Markov chain is a sequence random variables $(X_n)_{n \geq 0}$ with an initial distribution λ and transition matrix P if*

1. *X_0 has distribution λ,*
2. *for $n \geq 0$, X_{n+1} has distribution $(p_{ij} : j \in I)$, conditional on $X_n = j$ and is independent of $X_0, ..., X_{n-1}$, i.e. $p_{ij} = \mathbb{P}(X_{n+1} = i | X_n = j)$.*

We will now take a look at an example of a Markov chain.

Example 1.5. *A magic flower has been found that blooms in some colour every day. There are four colours the flower can bloom in, namely blue (state 1), red (state 2), yellow (state 3) and sparkling pink (state 4). We can calculate the probabilities the colour will bloom in tomorrow fully by knowing what colour it bloomed in yesterday. For instance, if the colour bloomed red yesterday, we know that it is going to bloom blue today by a probability $\frac{1}{2}$. We can describe this process by a Markov chain. The state space of the Markov chain are the colours, i.e. the state space is $I = \{1, 2, 3, 4\}$.*

We will draw our Markov chain as a graph with directed edge with will be the transition probability between two states:



This graph describes what the probabilities are of the colour changes from day to day. For instance, if yesterday the flower bloomed sparkly pink (state 4), we know that it will bloom again sparkly pink today by a probability of 0.1. With a probability of 0.9 it will bloom yellow (state 3) today. As there is no edge from state 4 to state 1 or state 2, we know that the probability of the flower blooming blue or red today when it bloomed sparkly pink yesterday is zero.

From this graph, we can derive the transition matrix of the Markov chain:

$$P = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0.5 & 0.5 & 0 & 0 \\ 0.75 & 0 & 0 & 0.25 \\ 0 & 0 & 0.9 & 0.1 \end{pmatrix}.$$

Note that the entries in each row sum up to 1, so these rows are indeed well-defined conditional probabilities. ♢

In a Markov chain it is often interesting to look at the stationary distribution of X_n as $n \to \infty$, because the stationary distribution models the proportion of time that the Markov chain has spend in each state. If we return to a state s an infinite number of times after having started at s, then s is called a *recurrent* state. A configuration that is not recurrent is called *transient.*

We say that from $s \in \Psi$ we can *reach* $t \in \Psi$ if, when we take s as starting configuration, we have

$$\mathbb{P}(\exists n : X_n = t, X_0 = s) > 0.$$

the probability that for some n we have $X_n = t$ is strictly positive. If we can reach s from t and t from s, we say that s and t are *communicating.* This defines an equivalence relation on the state space and therefore separates Ψ into *communicating classes.* Being recurrent or transient is a class property [Nor97].

Example 1.6. *Returning to the previous example, we see that every state can reach* 1. *For instance, starting from state 4, we see that*

$$\mathbb{P}(X_2 = 1 | X_0 = 4) = \mathbb{P}(X_2 = 1 | X_1 = 3) \cdot \mathbb{P}(X_1 = 3 | X_0 = 1) = 0.9 \cdot 0.75 > 0.$$

The other way around, we see that we can only reach state 1 and state 2 if we start at state 1. Therefore state 1 and 2 together form a communicating class. Note that state 3 and 4 also form a communicating class together.

Now note that if we start in state 3, we will move to state 1 with a probability of 0.75, after which we will never be able to return to state 3 again. Therefore state 3 is a transient state. Note that if we start at state 4, we can move with a probability 0.9 to state 3 in the next step, after which we can move to state 1 with a probability 0.75 next. Once we are in state 1, we will never return to state 4. Therefore the probability that we return to state 4 after we started there is strictly smaller than 1. We conclude that state 4 is a transient state as well. This was expected, as state 3 and 4 are in the same communicating class.

If we start in state 2, we will in the next step either be back at state 2, or we will have moved to state 1. If we moved to state 1, we move back to state 2 in the next step. We conclude that *the probability we return to state 2 after having started there, is 1. So state 2 is a recurrent state.*

As state 1 is in the same communicating class as state 2, state 1 must be recurrent as well. We can see this by looking at the probability of never returning to state 1 after having started there. We see that

$$\mathbb{P}(X_n = 2, .., X_2 = 2, X_1 = 2 | X_0 = 1) = \prod_{n \geq 1} 0.5^{n-1},$$

for all n. We see that this probability goes to 0 in the limit, which means that the probability that we return to state 1 after having started there is indeed equal to 1 and state 1 is a recurrent state. ◊

1.2.1 The Abelian sandpile model written as a Markov chain

We define a Markov chain on the state space $\Psi = \Omega$ the stable configurations by defining the configuration at time k to be

$$\eta_k = S\left(\eta_0 + \sum_{i=1}^{k} \delta_{X_i}\right), \tag{1.1}$$

where $\eta_0 \in \Omega$ is some starting configuration and δ_{X_i} denotes adding one particle to vertex X_i, where X_i are i.i.d. uniformly distributed on the V.

For this Markov chain we will denote the set of all recurrent configurations by $\mathcal{R}$.

Let η_{max} be the configuration such that $\eta_{max}(v) = \deg_G(v) - 1$. The following theorem will outline some properties specific to the recurrent states of this Markov chain.

Theorem 1.2. *[Dha90] Let $G = (V \cup \{s\}, E)$ be a finite connected graph on which a Markov chain is defined as in Equation 1.1. The following holds:*

1. *There is a unique recurrent class.*
2. *The following statements are equivalent for $\eta \in \Omega$:*
   (a) *η is a recurrent configuration.*
   (b) *There exists a configuration $\nu \geq \eta_{max}$ such that $S(\nu) = \eta$.*
   (c) *Any configuration $\nu \in \Omega$ can reach η, i.e. by adding some particles to ν and stabilizing we can obtain η.*

For a proof of this theorem, see [Dha90].

The stationary distribution of the Abelian sandpile model is the uniform measure on all recurrent configurations [Red05, Sec. 2.4].

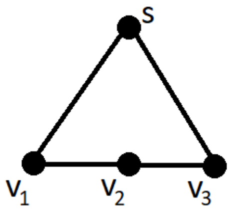
A concept closely related to recurrent configurations are forbidden subconfigurations.

Definition. *Let $\eta \in \Omega$. Let $W \subset V$ a non-empty subset. (W, η_W) is called a forbidden subconfiguration (FSC) if for all $w \in W$:*

$$\eta(w) \leq \sum_{v \in W \setminus \{w\}} (-L'_{wv}),$$

where L' is the $V \times V$ Laplacian matrix on G. If such a W exists, we say that η contains a FSC. A configuration is called allowed if it contains no forbidden subconfigurations.

Example 1.7. *For this example, consider again the graph*



Let η be the configuration such that

$$\eta(v) = \begin{cases} 1 & \text{if } v = v_2, \\ 0 & \text{else.} \end{cases}$$

Then for $W = \{v_1, v_2\}$ we see that (W, η_W) is a forbidden subconfiguration, as

$$\eta(v_1) = 0 \leq \sum_{v \in W \setminus \{v_1\}} (-L_{v_1 v}) = -L_{v_1 v_2} = 1,$$

and

$$\eta(v_2) = 1 \leq \sum_{v \in W \setminus \{v_2\}} (-L_{v_2 v}) = -L_{v_2 v_1} = 1.$$

$\Diamond$

In the following theorem we will connect the concept of FSC's to recurrent configurations.

Theorem 1.3. *[Dha90] A stable configuration $\eta \in \Omega$ is recurrent if and only if it contains no forbidden subconfigurations.*

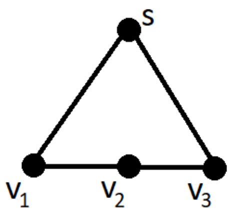
For a proof of this theorem, see [Dha90].

1.3 Dhar's burning algorithm

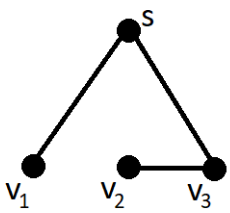
In this section the relationship between spanning trees and recurrent configurations will be investigated.

Definition. *A spanning tree of a graph $G = (V \cup \{s\}, E)$ is a graph $G' = (V \cup \{s\}, E')$ with $E' \subseteq E$ such that G' is a connected graph and $|V| = |E'| + 1$.*

Example 1.8. *Let G be*



An example of a spanning tree T of G is



$\diamondsuit$

Recurrent states of the Markov chain can be bijectively mapped to the spanning trees of G, using the so-called burning algorithm, which was first introduced in 1992 by Majumdar and Dhar [MD92]. This algorithm will also provide us with an easy way to check whether or not a stable configuration is recurrent or not.

The input of the burning algorithm is a graph $G = (V \cup \{s\}, E)$, a configuration η on V and an ordering on the edges $(e_{i_1} > e_{i_2} > ... > e_{i_n})$. The ordering of the edges is some formal ordering $e_{i_1} > e_{i_2} > ... > e_{i_n}$, where $\{i_1, ..., i_n\} = \{1, ..., n\}$ as sets. This will be important in proving the bijection between the recurrent states of the Markov chain and spanning trees that is induced by this algorithm and we will show in Theorem 1.4.

The output will be a subset $V' \subseteq V \cup \{s\}$ containing s and a spanning tree on the vertices in V' formed by the edge set $E' \subseteq E$. The algorithm runs in the following way:

- First, let $V_0 = \{s\}$ and $E_0 = \{e \in E | s = e\}$, i.e. E_0 consists of all the edges that are incident to the sink. We can see this as "burning" the sink and all edges that have s as an endpoint.

- In the i-th step, we will "burn" all vertices v in $V \setminus V_{i-1}$ and its adjacent unburned edges for which the number of neighbours in $V \setminus V_{i-1}$ is smaller than or equal to $\eta(v)$. Let W_i be the set of these v. Let $V_i = V_{i-1} \cup W_i$.

  For every vertex v in W_i, we want to add one edge to E' to form the spanning tree. Let $N_v \subset E_{i-1}$ be the set of edges in E_{i-1} which are incident to v. As v will burn in this step, some neighbour must have burned in the previous step and therefore N_v is not empty. Using the ordering on the edges, number the edges in N_v such that $e_1 > e_2 > ... > e_n$. Let bn_v be the number of neighbours of v that have burned in all the previous steps, i.e.

  $$bn_v := |\{w \in V_{i-1} | w \sim v\}.$$

  Now let $j = \eta(v) + 1 - (\deg_G(v) - bn_v)$. As $\deg_G(v) - bn_v$ equals the amount of neighbours that have not been burned in a previous step, in which the vertex was not yet able to burn, we have $j \leq n$. Note we also have that $j > 0$ as the burning condition ($\eta(v) \geq \deg_G(v) - bn_v$) holds. Therefore we have $0 < j \leq n$ and we can add e_j to E'. We will burn all the edges that have a $v \in W_i$ as an endpoint which have not been burned previously and we will collect these edges in the set E_i.

- This process will end when in a step no vertex is burned, i.e. when W_i is empty for some i. We will set $V' = V_n$. Note that we have the following chain of sets:

  $$\{s\} = V_0 \subset V_1 \subset ... \subset V_{n-1} = V_n = V'.$$

  As G is a finite graph, this algorithm will always terminate.

We call the step i in which a vertex v burns the *burning time* of v, which we will denote by b_v. Pseudocode of the burning algorithm can be found in Algorithm 1.

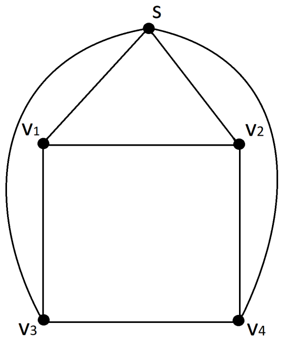
Algorithm 1 The burning algorithm

Input: A graph $G = (V \cup \{s\}, E)$, an ordering on E and a stable configuration η on V.
Output: A subgraph $G' = (V', E')$ of G, which is a spanning tree when η is a recurrent configuration.

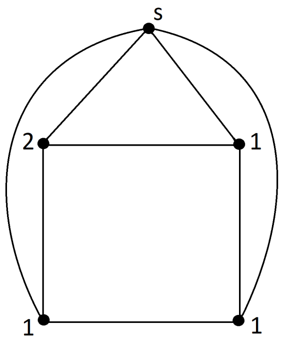
```
i := 0
V_0 := {s}
E_0 := {e ∈ E | s ∈ e}
E' := ∅
while V_i ≠ ∅ do
    i = i + 1
    W_i = ∅
    V_i := V_{i−1}
    E_i := ∅
    for v ∈ V \ V_{i−1} do
        bn_v := |{w ∈ V_{i−1} | v ∼ w}|
        if η(v) ≥ deg_G(v) − bn_v then
            V_i = V_i ∪ {v}                    ▷ Burn the vertex and adjacent unburnt edges
            W_i = W_i ∪ {v}
            E_b := {e ∈ E \ (∪_{j<i} E_j) | v ∈ e}
            E_i = E_i ∪ E_b

            N_v := {e ∈ E_i | v ∈ e}           ▷ Find the edge for the spanning tree
            Index N_v = {e_1, ..., e_n} such that
                e_1 > ... > e_n by the ordering on E.
            j := η(v) + 1 − deg_G(v) + bn_v
            E' = E' ∪ {e_j}
        end if
    end for
end while
V' = V_i
Output: G' = (V', E')
```

Let us look at the burning algorithm in practice by looking at an example.

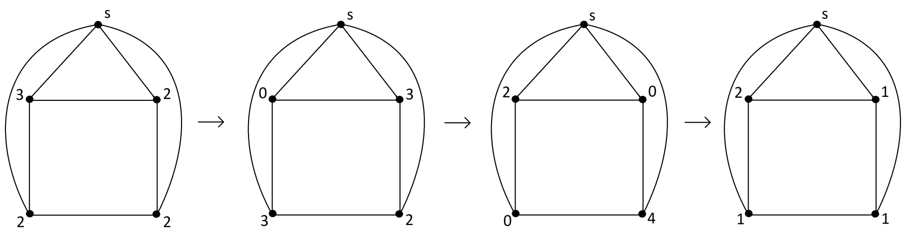
Example 1.9. *Let G be the following graph:*



Now let us consider the following recurrent configuration η on G.



But why is this configuration recurrent? First, note that η_{max}, the maximal configuration, where every vertex v has height $\deg_G(v)-1$, can be reached from any configuration in the Markov chain. Now, by Theorem 1.2 we know that there is only one class of recurrent configurations, so any configuration we can reach from η_{max} in the Markov chain is recurrent. Let us add one height to η_{max} at the vertex v_1 and look at the stabilization:



We conclude that the configuration η is indeed recurrent.

Now, before we execute the burning algorithm, we have to determine an ordering of the edges. Let the edges be ordered in the following way:

$$sv_1 > sv_2 > sv_3 > sv_4 > v_1v_2 > v_1v_3 > v_2v_4 > v_3v_4.$$

Now let us start the algorithm. First, we will burn the sink and we get $V_0 = \{s\}$, $E_0 = \{sv_1, sv_2, sv_3, sv_4\}$, $V' = \{s\}$ and $E' = \emptyset$. Now for this first step, we see that the amount of neighbours of v_1 that have not been burned is two, which equals $\eta(v_1)$. Therefore, we can now burn v_1. To do this, we get $V_1 = \{s, v_1\}$ and $E_1 = \{v_1v_2, v_1v_3\}$. We see that sv_1 gets added to E' as $N_{v_1} = \{sv_1\}$.

Note that we cannot burn the other vertices. For instance, for vertex v_3 we see that at the start of this step there were still two neighbours (v_1 and v_4) but its height is only 1. Therefore, the first step is now finished.

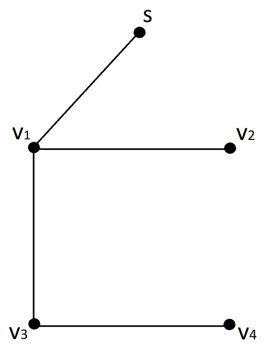
In the second step we can burn two vertices: v_2 and v_3. They both only have one neighbour, namely v_4, left that has not been burned yet and they both have height $\eta(v_2) = \eta(v_3) = 1$. First, to burn v_2, we get $V_2 = \{s, v_1, v_2\}$ and $E_2 = \{v_1v_2\}$. We add v_1v_2 to E', as $N_{v_2} = \{v_1v_2\}$. To burn v_3, we get $V_2 = \{s, v_1, v_2, v_3\}$ and $E_2 = \{v_1v_2, v_1v_3\}$. We also add the only vertex in $N_{v_3} = \{v_1v_3\}$ to E', which now equals $\{sv_1, v_1v_2, v_1v_3\}$. Note that we cannot burn vertex v_4, as its neighbours v_2 and v_3 were not burned yet at the start of this step.

To continue the algorithm, we will in the next step burn our last remaining vertex v_4. Note that as there are no unburned neighbours of v_4 remaining, we are allowed to do so. To burn the vertex, we add v_4 to V_3. As there are no unburned edges, E_3 will be the empty set. We have $N_{v_4} = \{v_2v_4, v_3v_4\}$. Now note as $|N_{v_4}| = 2$, we have to determine which of these two edges will be added to the subgraph G'. By the ordering, we have $v_2v_4 > v_3v_4$, so let $e_1 = v_2v_4$ and $e_2 = v_3v_4$. We have that

$$j = \eta(v_4) + 1 - (\deg_G(v_4) - bn_v) = 1 + 1 - (3 - 3) = 2.$$

We will add $e_2 = v_3v_4$ to the spanning tree edges E'.

Note that in the next step, there is no vertex left to burn. We set $V' = V_3 = \{s, v_1, v_2, v_3, v_4\}$. The algorithm will terminate with returning (V', E'), which is the subgraph:



◊

In the example we saw how inputting a recurrent configuration into the burning algorithm returned a spanning tree on all of the vertices $V \cup \{s\}$ of G, starting from s. In the following theorem we will prove that this algorithm gives a bijection between the recurrent configurations of G and the spanning trees of G. During the proof, we will also see why upon input of a non-recurrent stable configuration the algorithm will not return a spanning tree of G, but rather a spanning tree of a subgraph. This theorem and the burning algorithm are a well-known theorem, which were first introduced by Majumdar and Dhar in 1992. We will reformulate their proof here.

Theorem 1.4. *[MD92, Sec. 2] The burning algorithm provides bijection between the recurrent configurations and the spanning trees of G.*

Proof. First we will prove that upon input of a recurrent configuration the algorithm will always return a spanning tree of G. Let η be a recurrent configuration and $G' = (V', E')$ the output of the algorithm. If $V' = V \cup \{s\}$, then we have that G' is a spanning tree of G, as it is connected and $|E'| = |V| = |V'| - 1$. Now assume, for contradiction sake, that $V' \neq V \cup \{s\}$. Let $W = V \cup \{s\} \setminus V'$. Note that W is nonempty and $s \notin W$. As these edges were not able to burn in any step, we know that for all $w \in W$

$$\eta(w) < \deg_G(w) - \sum_{v \sim w, v \in V'} 1 = \sum_{v \sim w, v \in W} 1 = \sum_{v \in W \setminus \{w\}} -L_{vw}.$$

This means that (W, η_W) is a forbidden subconfiguration for η. Using Theorem 1.3 we now arrive at a contradiction, as a recurrent configuration cannot contain a FSC. We conclude that upon the input of a recurrent configuration the algorithm returns a spanning tree of G. Note also how Theorem 1.3 in the same way implies that if we input a transient configuration, a subgraph $G' = (V', E')$ with $V' \subsetneq V \cup \{s\}$ will be outputted.

Now we will argue why this mapping from recurrent configurations to spanning trees is injective. Let η_1 and η_2 be two different configurations. Let $v \in V$ be the first vertex that burns in the algorithm upon input of η_1 with $\eta_1(v) \neq \eta_2(v)$. Such a v exists because the heights of some vertices differ and all of the vertices burn in the algorithm, as η_1 and η_2 are recurrent configurations. Because of this height difference, either the vertex will burn at different times in the algorithm or a different edge will be added to the spanning tree with this vertex, as the ordering of the edges is fixed. Therefore we see that different recurrent configurations will result in different spanning trees.

We will now describe how to get from a spanning tree of G to a recurrent configuration. Let G and an ordering on the edges of G be given. Let $G' = (V, E')$ be a given spanning tree of G. From the spanning tree, the burning time (the step in which the vertex burned) can immediately be determined for every vertex $v \in V$ by determining the distance between the sink and the vertex. Call this burning time b_v. Let $b_s = 0$. For every $v \in V$, define

$$n_v := |\{w \in V \cup \{s\} | w \sim v, b_w + 1 < b_v\}|,$$

and

$$B_v := \{w \in V \cup \{s\} | w \sim v, b_w + 1 = b_v\},$$

where $u \sim v$ denote that u and v are neighbours in G. Now order the vertices in $B_v = \{w_1, ..., w_k\}$ such that

$$w_1 v > w_2 v > ... > w_k v$$

in the given ordering on the edges of G. Note that there is one edge in the spanning tree between v and a vertex in B_v. Let i such that $w_i v \in E'$ is this edge. Now let

$$\eta(v) = n_v + i - 1.$$

Note that $0 \leq \eta(v) < \deg_G(v)$, as $n_v \geq 0$, $i \geq 1$ and $n_v + i \leq n_v + |B_v| \leq \deg_G(v)$. If we input this configuration into the burning Algorithm 1, it will output the spanning tree we started with. Therefore we can in this way find for any spanning tree a recurrent configuration that gets mapped to it and therefore the mapping is bijective.

□

1.4 Counting recurrent configurations

Determining the amount of recurrent configurations of the Markov chain is now an equivalent problem to finding the amount of spanning trees of the graph. For this second problem, we can use Kirchhoff's matrix tree theorem [HHM08, Thm. 1.19].

Theorem 1.5 (Kirchhoff's matrix tree theorem)**.** *The amount of spanning trees of a graph G equals the determinant of the Laplacian matrix L'.*

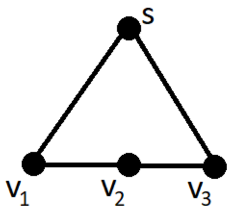
For a proof of this theorem, see [HHM08, Thm. 1.19].

We can combine this result with the bijection between spanning trees and recurrent configurations given by the burning algorithm to directly obtain the following result.

Corollary 1.6. *The amount of recurrent configurations equals the determinant of L'.*

We will now look at an example of all recurrent configurations of a graph.

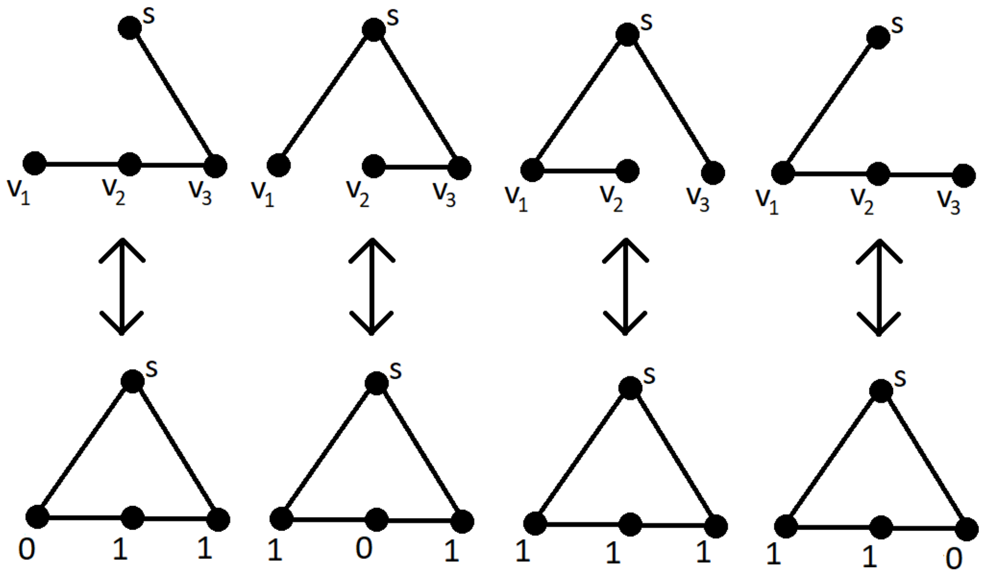
Example 1.10. *Consider the graph:*



We saw above that its Laplacian matrix L' is

$$L' = \begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{pmatrix}.$$

The determinant of this matrix equals $\det(L') = 4$. *Using the corollary above, we know that there are 4 spanning trees and therefore 4 recurrent configurations. Let the edge ordering be* $sv_1 > v_1v_2 > v_2v_3 > v_3s$. *All the spanning trees and recurrent configurations corresponding to them are:*



◊

1.5 The recurrent sandpile group

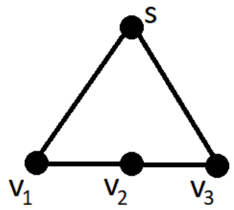
We can give more structure to the set of recurrent configurations by making it into a group. Let us define a group action on $\mathcal{R}$ by

$$\eta \oplus \zeta := S(\eta + \zeta) \text{ for } \eta, \zeta \in \mathcal{R},$$

where + denotes a pointwise addition of the two configuration. This operation makes $(\mathcal{R}, \oplus)$ into an Abelian group.

Let us look at an example of this group action.

Example 1.11. *Consider again the graph* $G = (V \cup \{s\}, E)$ *defined by:*



Using the vector notation of a configuration as introduced in Section 1.1.1, consider the two configurations

$$\eta := \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}, \nu := \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}.$$

We see that the pointwise addition of the two configuration equals

$$\eta + \nu = \begin{pmatrix} 2 \\ 2 \\ 1 \end{pmatrix}$$

The stabilization of this element is

$$S(\eta + \nu) = S\left(\begin{pmatrix} 2 \\ 2 \\ 1 \end{pmatrix}\right) = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}.$$

And therefore

$$\eta \oplus \zeta = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}.$$

We can also make an addition table of the whole group of the graph. To make it slightly more compact, we will write the configurations in a row form. For example, we will write $\eta = 110$ *and* $\nu = 111$. *With this notation, we can write down the whole addition table for this group:*

$\oplus$	011	101	110	111
011	111	011	101	110
101	011	101	110	111
110	101	110	111	011
111	110	111	011	101

Note that the configuration 101 *acts as the identity element in this group. Also note that none of the elements have order two. We conclude that in this case, the group is isomorphic to the cyclic group of order 4:*

$$(\mathcal{R}, \oplus) \cong \mathbb{Z}_4.$$

$\Diamond$

1.6 Weights

In this section we will define the weight of configurations and then prove upper and lower bounds for recurrent configurations.

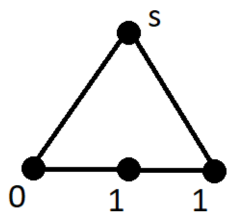
Definition. *The weight of a configuration* η *is defined as:*

$$w(\eta) = \sum_{v \in V} \eta(v),$$

which is just the sum of all heights.

Let us look at an example of weight.

Example 1.12. *Consider the configuration*



The weight of this configuration is $\sum_{v \in V} \eta(v) = 0 + 1 + 1 = 2.$ ◊

In the following lemma, we will discuss bounds on the weight of a recurrent configuration. This result is suggested but not proved in [Jár18, Ex.4.23]. We develop the proof here.

Lemma 1.7. *Let* η *be a recurrent configuration on a simple graph* $G = (V \cup \{s\}, E)$*. Then we have*

$$|E| - \deg_G(s) \leq w(\eta) \leq 2|E| - \deg_G(s) - |V|.$$

Proof. The upper bound follows quite directly from the fact that a recurrent configuration is a stable one. As it is stable, we know that for every vertex $v \in V$ we have

$$\eta(v) < \deg_G(v).$$

Summing this for all vertices, we get

$$\sum_{v \in V} \eta(v) \leq \sum_{v \in V} (\deg_G(v) - 1).$$

The right-hand side of this equation in some way counts the amount of edges that are incident to a vertex in V. Summing over all degrees in a graph would give $2|E|$, as every edge is incident to two vertices. However, some edges are incident to the sink, so we have to subtract $\deg_G(s)$ from this total. Therefore we can rewrite the equation to

$$\sum_{v \in V} (\deg_G(v) - 1) = \left(\sum_{e \in E} 2 \right) - \deg_G(s) - \sum_{v \in V} 1 = 2|E| - \deg_G(s) - |V|.$$

We conclude that the upper bound holds.

For the lower bound, we can zoom in on the burning algorithm. As η is a recurrent configuration, we know that all the vertices will "burn" during the algorithm. For vertex $v \in V$ to be able to burn, we know that the height of $\eta(v)$ should be bigger or equal to its degree minus the amount of burned neighbours. We get that

$$\sum_{v \in V} \eta(v) \geq \sum_{v \in V} \left(\deg_G(v) - \sum_{\substack{w \in V \\ w \sim v \\ b_w < b_v}} 1 \right). \tag{1.2}$$

If we now consider this sum from the perspective of edges, we see that for the first part of the sum again we have

$$\sum_{v \in V} \deg_G(v) = 2|E| - \deg_G(s).$$

Let us also bound the second part of this sum:

$$\sum_{v \in V} \sum_{\substack{w \in V \\ w \sim v \\ b_w < b_v}} 1$$

using the edges of G. For every edge in E, we have that it will be burned either when one of two vertices incident to it get burned, or when both of them will get burned at the exact same time. Only in the first case, we see that this will count towards this sum, as it counts exactly how many neighbours of v get burned before v does. Therefore, we get

$$\sum_{v \in V} \sum_{\substack{w \in V \\ w \sim v \\ b_w < b_v}} 1 \leq |E|.$$

In total, we see that

$$\sum_{v \in V} \eta(v) \geq 2|E| - \deg_G(s) - |E| = |E| - \deg_G(s).$$

This proves the lower bound and concludes the proof. □

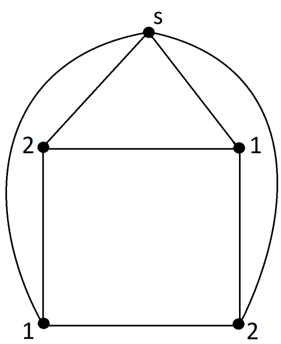
1.7 Minimal configurations

We will now define minimal configurations, which are a subset of $\mathcal{R}$, the recurrent configurations of the Markov chain. Let $\mathbb{1}_v$ be the configuration that has height one at v and height 0 on all other vertices.

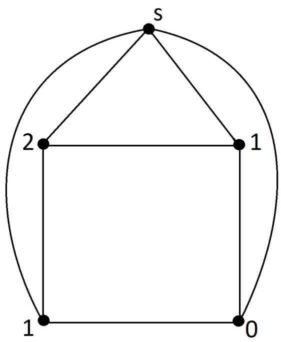
Definition. *A configuration η is called minimal if for all $v \in V$, we have that $\eta - \mathbb{1}_v \notin \mathcal{R}$, i.e. if we remove one height from any vertex, the configuration is not recurrent anymore.*

Now we will give an example of a minimal and a non-minimal configuration.

Example 1.13. *Let us take a look at the following configuration η_1:*



If we input this configuration into the burning Algorithm 1 with any ordering of the edges, we see that all vertices burn, so this is a recurrent configuration. However, if we remove two particles from the lower right vertex, we see that it will still fully burn. Therefore this configuration is non-minimal. On the other hand, let η_2 be η_1 with two particles removed from the lower right vertex:



This configuration η_2 is in fact minimal. We see that all vertices burn in the burning algorithm, so it is itself recurrent, but when we remove one particle from any of the vertices, it will no longer be recurrent. $\Diamond$

From the example we can already expect that minimal configurations have in some way a smaller weight than some other configurations. In the following lemma we will see that this is not just some local minimum of weight, but in fact all minimal configurations have the same weight which is the minimal weight possible for a recurrent configuration. This result is suggested but not proven in [Jár18, Ex. 4.23]. The proof is written out here.

Lemma 1.8. *Let $\eta \in \mathcal{R}$. Then $w(\eta) = |E| - \deg_G(s)$ if and only if η is a minimal configuration.*

Proof. "⇒" Let η be a recurrent configuration with $w(\eta) = |E| - \deg_G(s)$. Let $v \in V$ be any vertex. Then we see that

$$w(\eta - \mathbb{1}_v) = w(\eta) - 1 = |E| - \deg_G(s) - 1.$$

By Lemma 1.7 above, we conclude that $\eta - \mathbb{1}_v \notin \mathcal{R}$. As this is true for every $v \in V$, we have that η must be a minimal configuration and the implication from left to right holds.

"⇐" To prove the other implication, let η be a minimal configuration. Let us again consider the lower bound of Equation 1.2 of the previous proof:

$$\sum_{v \in V} \eta(v) \geq \sum_{v \in V} \left(\deg_G(v) - \sum_{\substack{w \in V \\ w \sim v \\ b_w < b_v}} 1 \right) = 2|E| - \deg_G(s) - \sum_{v \in V} \sum_{\substack{w \in V \\ w \sim v \\ b_w < b_v}} 1.$$

We counted the rightmost term (the double sum) in this equation by considering the edges in the burning algorithm. Using the burning times of edges, the rightmost term counts the number of edges of which one of the incident vertices burned earlier than the other. We will now argue that if η is minimal, no two neighbouring vertices burn at the same time in the algorithm.

Assume for contradiction sake that there are two neighbouring vertices $v_1 \sim v_2$ that have the same burning time in the algorithm upon input of a minimal configuration η. At the step they both burn, we know that the amount of neighbours of v_1 and v_2 that have burned in an earlier step are such that

$$\eta(v_1) \geq \deg_G(v_1) - bn_{v_1}, \quad \eta(v_2) \geq \deg_G(v_2) - bn_{v_2}.$$

Now let us consider the configuration $\eta - \mathbb{1}_{v_2}$. If v_2 has the same burning time upon input of η and upon input of $\eta - \mathbb{1}_{v_2}$, all vertices will burn in the same manner and we have that $\eta - \mathbb{1}_{v_2}$ is a recurrent configuration, which implies that η is not minimal. Now if v_2 has a bigger burning time upon input of $\eta - \mathbb{1}_{v_2}$ than upon input of η, it must burn one step later in the case of

$\eta - \mathbb{1}_{v_2}$ than in the case of η. This is because we know that all vertices that have a burning time equal or smaller than the burning time of v_2 in η will burn in the same matter in the algorithm upon input of $\eta - \mathbb{1}_{v_2}$. In the algorithm upon input of η, we had

$$\eta(v_2) \geq \deg_G(v_2) - bn_{v_2}.$$

Now, after the step in which v_1 burns, the amount of burned vertices adjacent to v_2 increases by at least one. Therefore we see that in this step we have

$$\eta(v_2) - 1 \geq \deg_G(v_2) - bn'_{v_2},$$

where bn'_{v_2} are the amount of burned neighbours in this step upon input of $\eta - \mathbb{1}_{v_2}$, from which it follows that v_2 will burn. As all the other vertices burned in the algorithm upon input of η, we get that all vertices will burn upon input of $\eta - \mathbb{1}_{v_2}$ as well, albeit some will burn at a later step in the latter. This means that $\eta - \mathbb{1}_{v_2}$ is a recurrent configuration, which implies that η is not a minimal configuration. We have reached a contradiction and conclude that no two vertices burn at the same time in the algorithm upon input of a minimal configuration.

As one of the two vertices incident to the edge burns earlier than the other adjacent vertex for all edges, we see that the double sum in the equation above is exactly equal to $|E|$. Therefore we see that if η is a minimal configuration, then

$$w(\eta) = \sum_{v \in V} \eta(v) = 2|E| - \deg_G(s) - \sum_{v \in V} \sum_{\substack{w \in V \\ w \sim v \\ b_w < b_v}} 1 = 2|E| - \deg_G(s) - |E| = |E| - \deg_G(s).$$

□

Example 1.14. *Let us return to the to configurations we saw in Example 1.13.*

The first example η_1 was a non-minimal configuration. Its weight is

$$w(\eta_1) = 2 + 1 + 1 + 2 = 6 \neq 4 = 8 - 4 = |E| - \deg_G(s).$$

The second configuration η_2 from Example 1.13 is a minimal configuration. It has weight:

$$w(\eta_2) = 2 + 1 + 1 = 4 = 8 - 4 = |E| - \deg_G(s).$$

This coincides with the statement of the theorem. ◊

1.8 Acyclic orientations

Whilst we only looked into non-oriented graphs up until now, we will in this section introduce (partial) orientations of graphs. In particular, we will look into acyclic orientations, i.e. orientations without cycles. This concept will be linked to recurrent configurations, where we will prove a bijection between acyclic orientations and minimal configurations. We will also prove that there is an injective mapping from non-minimal configurations to partial acyclic orientations.

Definition. *An orientation of a graph $G = (V \cup \{s\}, E)$ is a set of two functions (o, t) called the origin and target function $o : E \to V, t : E \to V$, such that $o(e)$ and $t(e)$ are the two vertices incident to the edge e.*

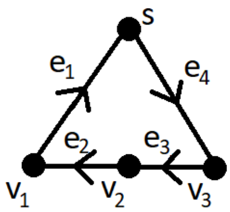
We will denote an orientation by $\Gamma = (o, t)$. An *oriented graph* (G, Γ) is a graph $G = (V \cup \{s\}, E)$ together with an orientation Γ. An edge e is called *outgoing* for vertex $o(v)$ and *incoming* for vertex $t(v)$.

An *oriented cycle* in the oriented graph (G, Γ) is a path $e_1 e_2 ... e_n$ in the graph such that

$$t(e_1) = o(e_2), t(e_2) = o(e_3), ..., t(e_{n-1}) = o(e_{n-1}), t(e_n) = o(e_1).$$

An orientation without any cycles is called an *acyclic orientation.* A *source* in a oriented graph (G, Γ) is a vertex $v \in V \cup \{s\}$ such that there exist no edges $e \in E$ for which $t(e) = v$, i.e. v has only outgoing edges. A subgraph $G_W = (W, E_W)$ of G will have *inherited orientation* from the oriented graph (G, Γ) if $o_{G_W} = o_G|_{E_W}$ and $t_{G_W} = t_G|_{E_W}$, i.e. the orientation of an edge in G_W is the same as the orientation in (G, Γ). By slight abuse of orientation, we will denote a subgraph with inherited orientation by (G_W, Γ).

Example 1.15. *An example of a graph with an orientation is:*



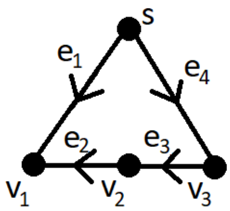
We see here that for example

$$o(e_3) = v_3, t(e_1) = s.$$

This graph contains an oriented cycle, namely $e_4e_3e_2e_1$*, as*

$$t(e_4) = o(e_3) = v_3, \quad t(e_3) = o(e_2) = v_2, \quad t(e_2) = o(e_1) = v_1, \quad t(e_1) = o(e_4) = s.$$

Moreover, there is no source, as every vertex has an incoming edge. Now let us consider another orientation of this same graph:



We see that this orientation contains no cycle, therefore this orientation is called acyclic. Note that s is a source in this orientation, as it has only outgoing edges. ◊

From this example, one might notice how there seems to be a negative association between the existence of sources and the existence of cycles. In the following lemma we will prove that an orientation without any sources must be acyclic. We will need this lemma as a tool in proving a bijection between acyclic orientations and minimal configurations. Whilst the statement of the lemma is probably not new, we will state and prove the lemma here ourselves.

Lemma 1.9. *Let (G, Γ) be a finite, oriented graph in which every vertex v has an incoming edge. Then (G, Γ) contains an oriented cycle.*

Proof. Assume for contradiction sake that (G, Γ) contains no oriented cycle. Let v_1 be any vertex. We know that v has at least one incoming edge e_1. Let v_2 be the origin of e_1. We know that v_2 also has an incoming edge e_2. As we assumed that the graph contains no oriented cycle, we know that $o(e_2)$ is not v_1, but some other vertex v_3. This vertex again has an incoming edge, which origin is not any of the already labeled vertices, as we would get a cycle in that case. Using an argument of induction, we can repeat this procedure to label all vertices $V \cup \{s\} = \{v_1, ..., v_n\}$ such that

$$v_n v_{n-1} ... v_3 v_2 v_1$$

is an oriented path in G. But now we arrive at a contradiction, as v_n has an incoming edge e_n. The origin $o(e_n) \in V \cup \{s\}$ must be a vertex of G, which is already part of the path, so this forms an oriented cycle. □

1.8.1 Acyclic orientations and minimal configurations

Acyclic orientations of a graph are closely connected to configurations in the Abelian sandpile model. In the next theorem we will prove a connection between acyclic orientations and minimal configurations. This result is suggested, but not proved, in [Jár18, Ex. 4.26]. We have provided a proof here.

Theorem 1.10. *There is a bijection between minimal configurations and acyclic orientations of G with a unique source at s.*

Proof. Let $\Gamma = (o, t)$ be an acyclic orientation of G with a unique source at s. Then define the configuration η to be:

$$\eta(v) := |\{e \in E | o(e) = v\}|, \tag{1.3}$$

for all $v \in V$. Note that $\eta(v) \in \mathbb{Z}_{\geq 0}$, so it is indeed a configuration. As the configuration has a unique source at s, we see that

$$\eta(v) < |\{e \in E | o(e) = v \text{ or } t(e) = v\}| = \deg_G(v),$$

so η is a stable configuration.

We will now use a variant of the burning algorithm on acyclic orientations to show that the configuration η is recurrent. In this burning algorithm, we will burn a vertex (and all its adjacent edges) when it has only outgoing edges, i.e. the vertices that are a source in the remaining unburnt subgraph. Note that we start with burning the sink, just as in the burning Algorithm 1 as introduced in Section 1.3. For this proof, we ignore the choosing of a specific edge for the spanning tree and will focus on determining why all vertices will burn in the algorithm and why they burn in the same order as the just given configuration η would in the burning Algorithm 1 for stable configurations. Note that in this new version of the burning algorithm, an oriented edge e will only burn at the time that vertex $o(e)$ burns, as vertex $t(e)$ is only able to burn after this edge has been burned. Therefore we see that for every vertex v the quantity

$$\deg_G(v) - bn_v,$$

where bn_v denotes the amount of burned neighbours in a previous step, coincides with the amount of incoming edges it has. In the burning Algorithm 1 for stable configurations, the burning condition for η is given by

$$\eta(v) \geq \deg_G(v) - bn_v.$$

As we choose $\eta(v)$ equal to the amount of outgoing edges to v, we see that the burning conditions coincide and the vertices will burn in the exact same order. Therefore if burning the vertex when it has only outgoing edges remaining will burn all edges, we get that η is a recurrent configuration, as this implies that all vertices will burn when using Algorithm 1 from Section 1.3 with input η.

Now we will prove by contradiction that using this new burning algorithm, an acyclic orientation will burn all the vertices. Assume for contradiction sake that upon input of an acyclic orientation the algorithm would not burn all vertices of the graph. This implies there is a subset $W \subseteq V$ for which all vertices have an incoming edge remaining in the subgraph G_W with inherited orientation from Γ. Using Lemma 1.9 we have that the subgraph G_W must contain a cycle, which implies that the graph (G, Γ) contains a cycle. This is a contradiction as Γ is an acyclic orientation. We conclude that this burning algorithm burns all vertices of an acyclic oriented graph. We conclude that η defined by Equation 1.3 is a recurrent configuration.

Now let us look at the weight of the configuration. By definition we have that

$$w(\eta) = \sum_{v \in V} \eta(v) = \sum_{v \in V} |\{e \in E | o(e) = v\}|.$$

Now note that for every edge e not connected to the sink, we have $o(e) \in V$, so there is a vertex that gets an added particle in η from this. Therefore we see that

$$w(\eta) = \sum_{v \in V} |\{e \in E | o(e) = v\}| = |E| - \deg_G(s).$$

Using Lemma 1.8 we conclude that η is a minimal configuration.

We now have a mapping from acyclic orientations of G with a unique source at s to minimal configurations of G. We will denote this mapping by ϕ_{ac}. We will now prove that the map ϕ_{ac} is bijective.

First, let us prove that this mapping is injective. Let Γ_1 and Γ_2 be two acyclic orientations with a unique source at s such that $\phi_{ac}(\Gamma_1) = \phi_{ac}(\Gamma_2)$. Assume, for contradiction sake, there is an edge e_1 that is differently oriented in Γ_1 and Γ_2, so $o_{\Gamma_1}(e_1) = t_{\Gamma_2}(e_1)$ and $t_{\Gamma_1}(e_1) = o_{\Gamma_2}(e_1)$ for this edge. As $\phi_{ac}(\Gamma_1) = \phi_{ac}(\Gamma_2)$, we must have that every vertex has the same amount of incoming and outgoing vertices in the two orientations. This means that the vertex $o_{\Gamma_1}(e)$ must have some edge that is incoming in Γ_1 and outgoing in Γ_2. In fact, there is a set of such vertices $W \subseteq V \cup \{s\}$ for which there are some edges that have a different orientation in Γ_1 and Γ_2. Note that all these vertices have both incoming and outgoing edges in Γ_1 that have flipped direction in Γ_2, as the total amount of incoming edges of a vertex is the same in Γ_1 and Γ_2. Now if we look at the oriented subgraph $(G_W, \Gamma_{1,W})$, we have that it is a finite graph in which every vertex has an incoming edge. Using Lemma 1.9 we can now conclude that Γ_1 contains a cycle, which is a contradiction as Γ_1 is an acyclic orientation. Therefore our assumption must be wrong and we conclude that every edge is oriented in the same way in both Γ_1 and Γ_2, i.e. $\Gamma_1 = \Gamma_2$. We conclude that the map ϕ_{ac} is injective.

Now we will prove this mapping is surjective. Let η be a minimal configuration. In the burning Algorithm 1 for configurations, we have that two neighbouring vertices in G will not burn at the same time upon input of a minimal configuration. This holds because if two neighbouring vertices would burn at the same time, the height of one of the vertices could be reduced and all vertices would still burn, which would imply that the configuration is not minimal. We proved this behaviour in the proof of Lemma 1.8. We can use the fact that no two neighbouring vertices burn at the same time to our advantage by defining the orientation of an edge e by letting $o(e)$ be the incident vertex that burns first and $t(e)$ be the other vertex. Note that in this case we have that

$$\eta(v) = |\{e \in E | o(e) = v\}| \tag{1.4}$$

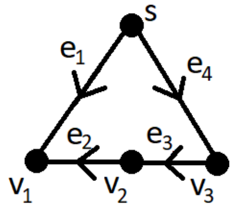
for all $v \in V$. Note that determining the orientations in this way implies there is no oriented cycle in this orientation, because for an oriented cycle $v_1 v_2 ... v_n$ in G, we would have that

$$b_{v_1} < b_{v_2} < b_{v_3} < ... < b_{v_n} < b_{v_1},$$

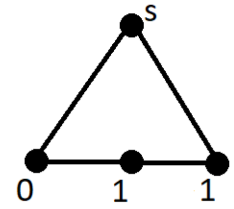
where b_v denotes the burning time of vertex v. Also note that s is a source in this orientation, as it is the first vertex that burned, so all edges will be outgoing for s. Also note that no other vertices will be sources, as all other vertices only burn upon the burning of some of its neighbours, as $\eta(v) < \deg(v)$ for all $v \in V$. We conclude that the orientation $\Gamma = (o, t)$ we obtain this way is an acyclic configuration with a unique source at s. Note that $\phi_{ac}(\Gamma) = \eta$ by the observation made in Equation 1.4. We conclude that ϕ_{ac} is surjective. □

Now we will show in an example how to go from an acyclic orientation to a minimal configuration.

Example 1.16. *Consider the following acyclic orientation with a unique source at s:*



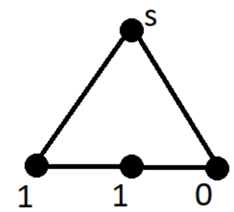
This orientation gets mapped to a configuration η in which the height at every vertex equals the amount of outgoing vertices. We get the following configuration:



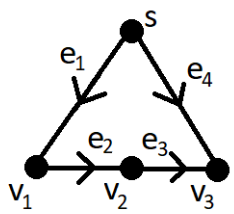
Note that this configuration is recurrent, as it will fully burn if we input it to the burning Algorithm 1. It is also minimal, as

$$w(\eta) = \sum_{v \in V} \eta(v) = 0 + 1 + 1 = 2 = |E| - \deg_G(s).$$

Now let us take a look at a different minimal configuration:



If we would input this configuration in the algorithm, we would see that in the first step s would burn, then in the second step v_1. Then in the next step v_2 would burn and in the last step v_3 would burn. We orient the edges such that the origin is the incident vertex that burns first and the target the other vertex. We get the following orientation of G:



Note that it is indeed an acyclic orientation with a unique source at s. ◊

Now that we have found a connection between acyclic orientations and minimal configurations, one might wonder if there is a way to connect orientations to recurrent configurations that are not minimal. In this thesis, a connection between partial acyclic orientations and non-minimal configurations has been found. We will now introduce the concept of partial orientations.

Definition. *A partial orientation of a graph* $G = (V \cup \{s\}, E)$ *is a subset of edges* $E' \subsetneq E$ *together with an orientation* $\Gamma = (o, t)$ *of these edges, consisting of a origin function*

$$o : E' \to V \cup \{s\},$$

and a target function

$$t : E' \to V \cup \{s\}.$$

We will denote a graph with a partial notation by (G, E', Γ).

A vertex v of a partial orientation (G, E', Γ) is called a *source* if there is no edge $e \in E'$ such that $t(e) = v$. An *oriented cycle* in a partial orientation (G, E', Γ) is a path $e_1 e_2 ... e_n$ in the graph such that all edges are oriented, i.e. $e_i \in E'$ for all i and

$$t(e_1) = o(e_2), t(e_2) = o(e_3), ..., t(e_{n-1}) = o(e_{n-1}), t(e_n) = o(e_1).$$

A partial orientation without any oriented cycles is called an *partial acyclic orientation.*

Theorem 1.11. *There is a surjective map* ϕ_{pac} *from partial acyclic orientations with a unique source at* s *to non-minimal recurrent configurations.*

Proof. Let (G, E', Γ) be a partial acyclic orientation with a unique source at s. Then let η be the configuration defined by

$$\eta(v) := |\{e \in E' | o(e) = v)\}| + |\{e \in E \setminus E' | v \in e\}|$$

for all $v \in V$. Comparing this with the mapping of acyclic orientations to minimal configurations, note that this is like considering a non-oriented edge as outgoing to both vertices it is incident to.

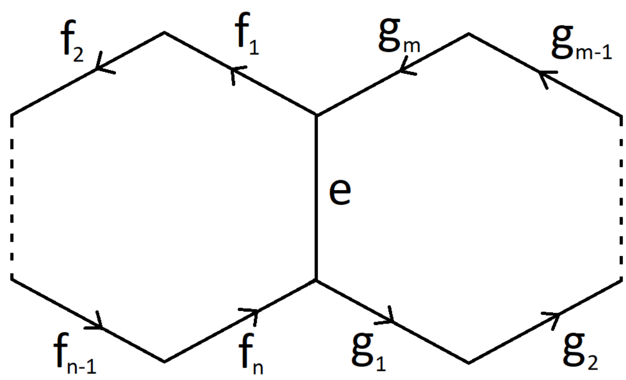
To be able to show this η is a non-minimal configuration, we will first show that we are able to add orientations to edges in $E \setminus E'$ such that we get an acyclic configuration Γ^* with a unique source at s, which inherits the orientation on E' from the partial orientation. Let $e \in E \setminus E'$ be any edge that is not oriented in Γ. We will prove that we can give some orientation to e such that $(E' \cup \{e\}, \Gamma)$ is a (partial) acyclic orientation with a unique source at s. If e is incident to s, we will orient it outward from s. This will not create an oriented cycle, as s has no incoming edges. If e is not incident to s it is less clear how to orient e, but we will prove by contradiction that there is at least one orientation of e that does not create a cycle. For contradiction sake, assume there is no orientation we can give to the edge such that the orientation stays acyclic. This means that there is an oriented cycle

$$e f_1 f_2 ... f_n$$

for some $f_1, ..., f_n \in E'$ if we orient e in one way and another oriented cycle

$$e g_1 g_2 ... g_m$$

for some $g_1, ..., g_m \in E'$ if we orient e in the other way. The subgraph containing these two paths looks like this:



Now note that

$$f_1 f_2 ... f_n g_1 g_2 ... g_m$$

is a oriented cycle. This is a contradiction with the partial orientation (E', Γ) being acyclic. We conclude that there must be some way to orient e such that $(E' \cup \{e\}, \Gamma^*)$ is a (partial) acyclic orientation with a unique source at s. Repeating this for every edge in $E \setminus E'$ will result in an acyclic orientation (G, Γ^*) with a unique source at s.

Now from this acyclic orientation with a unique source at s, we obtain a minimal configuration η' using the construction from the construction of Theorem 1.10. Note that

$$\eta'(v) = |\{e \in E | o_{\Gamma^*}(e) = v)\}| \leq |\{e \in E' | o_{\Gamma}(e) = v)\}| + |\{e \in E \setminus E' | v \in e\}| = \eta(v).$$

With every added orientation of an edge to the partial orientation Γ, we "lost" one particle at the vertex that became the target of the newly oriented edge. We have that

$$\eta = \eta' + \sum_{e \in E \setminus E'} \mathbb{1}_{t_{\Gamma^*}(e)}.$$

By definition of the Markov chain, this means that η is a recurrent configuration. As $E \setminus E'$ is not empty, we know it is not minimal. Therefore

$$\phi_{pac} : (G, E', \Gamma) \mapsto \eta$$

is a well-defined map from a partial orientations with a unique source at s to non-minimal configurations.

Now we will show why this mapping is surjective. Let η be a non-minimal recurrent configuration. Then we can subtract some particles from some vertices $V' = \{v_{i_1}, ..., v_{i_n}\}$ with $v_{i_j} \in V$ for all $j \in \{1, ..., n\}$ to obtain a minimal configuration η' with

$$\eta' = \eta - \sum_{j=1}^{n} \mathbb{1}_{v_{i_j}}.$$

Using the bijection of Theorem 1.10 we can obtain an acyclic orientation Γ with a unique source at s. Now we want to replace an incoming edge of each v_{i_j} with a non-oriented edge to "add" more particles to the configuration that is the image $\phi_{pac}((G, E', \Gamma))$. If we would replace all incoming edges of a vertex v with non-oriented edges, we would get that

$$\phi_{pac}((G, E', \Gamma))(v) = |\{e \in E' | o(e) = v)\}| + |\{e \in E \setminus E' | v \in e\}| = \deg_G(v),$$

as it would then only have neutral and outgoing edges. This would make the configuration unstable at v. As we want to have $\phi_{pac}((G, E', \Gamma)) = \eta(v)$, all sites have to be stable. Therefore, there are enough incoming edges incident to all of the v_i to replace with non-oriented edges such that every vertex $v \in V$ will still have at least one incoming edge remaining, as

$$\eta(v) = |\{e \in E'|o(e) = v\}| + |\{e \in E \setminus E'|v \in e\}| = \deg_G(v) - |\{e \in E'|t(e) = v\}| < \deg_G(v).$$

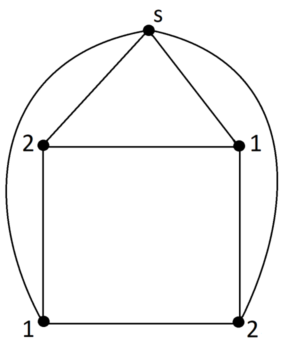
Therefore we see that the partial orientation (G, E', Γ) we obtain after removing the orientation of one edge for every $v_i \in V'$ still has a unique source at s. Also, with the mapping we see that

$$\phi_{pac}((G, E', \Gamma)) = \eta.$$

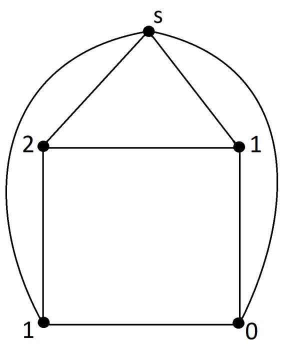
We conclude that ϕ_{pac} is surjective. □

In the next example we will show how to find a partial orientation that corresponds to a given non-minimal recurrent configuration using the proof of surjectivity above. In the example we will also see why the mapping of acyclic orientations to non-minimal recurrent configurations is not injective.

Example 1.17. *Take the following recurrent configuration η:*



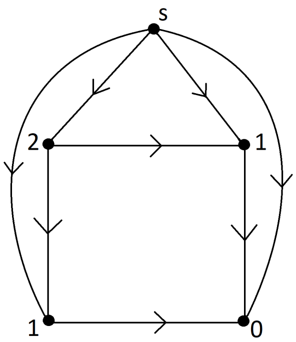
Note that if we would apply the burning algorithm to this configuration, all vertices would burn, so this is indeed a recurrent configuration. To find the partial acyclic orientation that maps to this configuration, we have to reduce some heights of the configuration to get a minimal configuration. If we reduce the height of the lower right vertex by 2, we get configuration η':



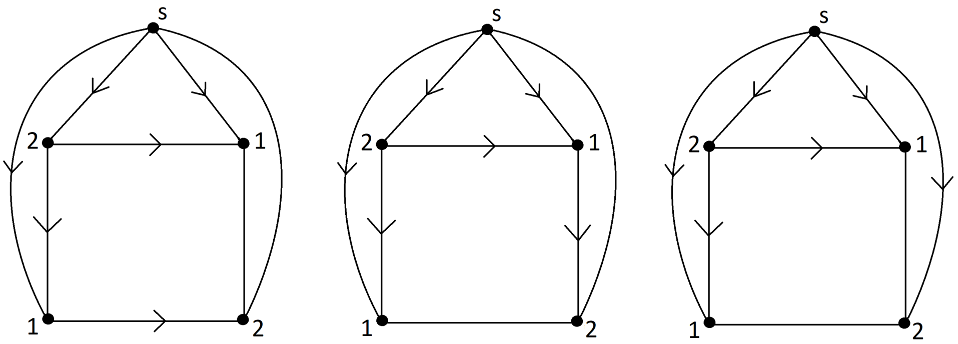
This configuration is still recurrent, as all vertices burn if we apply the burning algorithm. Note that

$$w(\eta') = \sum_{v \in V} \eta'(v) = 4 = |E| - \deg_G(s).$$

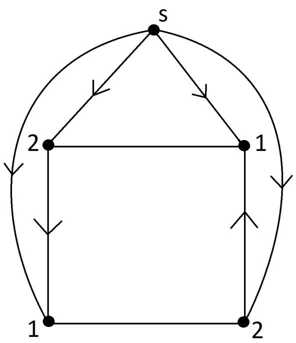
Using Lemma 1.8 we know this implies that η' is a minimal configuration. Using the bijection from Theorem 1.10 we get the following acyclic orientation with a unique source at s corresponding to η':



Now to get to the partial orientation that corresponds to η, we have to remove orientation of some edges that are incident to the vertices of which we removed a particle to get to η'. As we removed two particles from the lower right vertex to get to η', we will now remove the orientation of two incoming edges to this vertex. There are three ways to remove these orientations:



Note that all these are partial orientations with a unique source at s that all get mapped to the configuration η by ϕ_{pac}. These are not even all partial orientations that map to η by ϕ_{pac}, as we could have also chosen different vertices to reduce height from to get a minimal configuration. For instance, the following partial orientation also maps to η under ϕ_{pac}:



We conclude that ϕ_{pac} is not an injective map. $\Diamond$

Chapter 2

Geometric approach: the Jacobian and q-reduced divisors

In this section we will define objects which are inspired by objects from algebraic geometry. We will start by defining divisors on a graph and an equivalence relation on these divisors. With this equivalence relation, we will define the Picard sets and the Jacobian group, which are sets of equivalence classes of divisors. We will show that there is an isomorphism between the Jacobian group and the recurrent sandpile group from the previous chapter. After this, we will take a look into q-reduced divisors, which uniquely represent all the equivalence classes in the Picard sets. We will discuss a burning algorithm for q-reduced divisors, which gives a bijection to spanning trees.

Almost all of the objects we will define in this chapter have a counterpart in algebraic geometry with the same name. Some connections between the objects in algebraic geometry and the graph theoretical versions we will describe in this section are known [Bak08; BJ16; Bak08; Jen21], but are out of scope for this thesis.

An active field of research the last two decades is to reformulate theorems from algebraic geometry to theorems on graphs using the same-called objects. After having introduced the concepts of divisors, Picard sets and the Jacobian on graphs in this section, we will do exactly that by diving into the Torelli theorem in Chapter 4. This theorem finds its origin in algebraic geometry and has been translated to the graph geometric setting in multiple ways [CV10; Gri23].

2.1 Divisors

In this section, we will introduce divisors and some of their properties. Let us start with defining divisors, which are formal sums of the vertices of the graph.

Definition. *Given a finite, connected graph $G = (V, E)$, we define a divisor D to be*

$$D = \sum_{v \in V} a_v \cdot v,$$

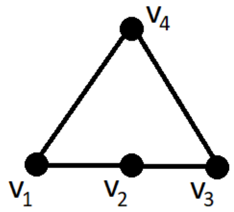
where $a_v \in \mathbb{Z}$ are some constants. The set of all divisors on G is called $Div(G)$. The degree of the divisor is defined by

$$\deg : Div(G) \to \mathbb{Z}, \quad \deg(D) = \sum_{v \in V} a_v$$

for $D \in Div(G)$. The set of divisors of a certain degree $d \in \mathbb{Z}$ is denoted by $Div_d(G)$.

A divisor D is called *effective* if $a_v \geq 0$ for all $v \in V$. The set of effective divisors of degree $d \in \mathbb{Z}$ is denoted by $\mathrm{Div}_d^+(G)$. Given a vertex $q \in V$, the *q-weight* of a divisor D is $w_q(D) := \sum_{v \in V \setminus \{q\}} a_v$. Let us look at an example of divisors.

Example 2.1. *Consider the following graph:*



An example of a divisor on G is

$$D = 3v_1 + v_2 - 5v_3 + 20v_4.$$

The degree of the divisor is $\deg(D) = 3 + 1 - 5 + 20 = 19$. *The v_1-weight of the divisor is* $w_{v_1} = 1 - 5 + 20 = 16$. ♢

A subset of the divisors are the *principal divisors.*

Definition. *A divisor $D_f = \sum_{v \in V} a_v \cdot v$ is called principal if there exists a function $f : V \to \mathbb{Z}$, such that for all $v \in V$*

$$a_v = \sum_{(v,w) \in E} (f(v) - f(w)).$$

Note that the degree of a principal divisors is zero. The set of principal divisors is denoted by $\mathrm{Prin}(G)$. Now let us look at an example of a principal divisor.

Example 2.2. *Consider again the graph of the previous example. Let $f : V \to \mathbb{Z}$ be defined as*

$$f(v) = \begin{cases} 2 & \text{if } v = v_1 \text{ or } v = v_2, \\ 1 & \text{if } v = v_3, \\ -3 & \text{if } v = v_4. \end{cases}$$

Then we have that

$$\begin{aligned} a_{v_1} &= f(v_1) - f(v_4) + f(v_1) - f(v_2) = 2 + 3 + 2 - 2 = 5, \\ a_{v_2} &= f(v_2) - f(v_1) + f(v_2) - f(v_3) = 2 - 2 + 2 - 1 = 1, \\ a_{v_3} &= f(v_3) - f(v_2) + f(v_3) - f(v_4) = 1 - 2 + 1 + 3 = 3, \\ a_{v_4} &= f(v_4) - f(v_3) + f(v_4) - f(v_1) = -3 - 1 - 3 - 2 = -9. \end{aligned}$$

The principal divisor that corresponds with this f is therefore

$$D_f = 5v_1 + v_2 + 3v_3 - 9v_4.$$

Note that the degree of this divisor is indeed zero:

$$\deg(D_f) = 5 + 1 + 3 - 9 = 0.$$

♢

Using these principal divisors, we can define an equivalence relation on divisors. In the following lemma we will define this relation and prove that it is indeed an equivalence relation. This relation can be found in a lot of literature, for instance in [BS13] and [Sho10].

Definition. *Let $D_1, D_2 \in Div(G)$. We say $D_1 \sim D_2$ if and only if $D_1 - D_2$ is a principal divisor, i.e. there exists an $f : V \to \mathbb{Z}$ such that*

$$D_1 - D_2 = D_f.$$

We want to prove this is in fact an equivalence relation. This detailed proof is a quick verification of what is assumed as a fact in literature.

Lemma 2.1. *The relation given by $D_1 \sim D_2$ if and only if $D_1 - D_2$ is a principal divisor, i.e. there exists an $f : V \to \mathbb{Z}$ such that*

$$D_1 - D_2 = D_f,$$

is an equivalence relation.

Proof. Clearly $\sim$ is reflexive, i.e. $D \sim D$ for all $D \in \mathrm{Div}(G)$, as

$$D - D = 0 = D_{f_0},$$

where $f_0(v) := 0$ for all $v \in V$.

Let $D_1, D_2 \in \mathrm{Div}(G)$. Note that if

$$D_1 - D_2 = D_f,$$

for some $f : V \to \mathbb{Z}$, then we have that

$$D_2 - D_1 = -D_f = D_{-f},$$

where $(-f)(v) := -f(v)$ is a function $-f : V \to \mathbb{Z}$. Therefore we conclude that $\sim$ is symmetric.

What is left to prove is transitivity. Let $D_1, D_2, D_3 \in \mathrm{Div}(G)$ such that $D_1 \sim D_2$ and $D_2 \sim D_3$. This means there exist $f, g : V \to \mathbb{Z}$ such that

$$D_1 - D_2 = D_f, \quad D_2 - D_3 = D_g.$$

Then we see that

$$D_1 - D_3 = D_f + D_g.$$

Let $v \in V$ some vertex. Then we see that the constant a_v in $D_f + D_g$ equals:

$$a_v = \sum_{(v,w)\in E} (f(v) - f(w) + g(v) - g(w)) = \sum_{(v,w)\in E} ((f+g)(v) - (f+g)(w)),$$

where $f + g : V \to \mathbb{Z}$, $(f + g)(v) := f(v) + g(v)$. It yields that $D_1 \sim D_3$, which concludes this proof. □

With this equivalence relation, we can define the rank of a divisor.

Definition. *A divisor D has at least rank $d \in \mathbb{Z}$ if for every effective divisor E of degree d, $D - E$ is equivalent to an effective divisor E'. D has rank d if it has at least rank d but not $d + 1$, i.e. there exists an effective divisor E of degree $d + 1$ such that $D - E$ is not equivalent to any effective divisor E'. We denote the rank of a divisor D by rank(D).*

Let us look at an example of the rank of a divisor.

Example 2.3. *Let us again consider the graph from Example 2.1. Consider the divisor*

$$D = v_1 + v_4 \in Div_2(G).$$

As D is itself an effective divisor (all its constants are non-negative), it has at least rank 0. The next step is to determine whether D has at least rank 1. All effective divisors of degree 1 on G are:

$$E_1 := v_1, \quad E_2 := v_2, \quad E_3 := v_3, \quad E_4 = v_4.$$

To prove that D has at least rank 1, we have to prove that $D - E_i$ is equivalent to an effective divisor for every $i \in \{1, 2, 3, 4\}$. We see that

$$D - E_1 = v_4, \quad D - E_4 = v_1,$$

are effective divisors. For $D - E_2$ we get

$$D - E_2 = v_1 - v_2 + v_4 \sim v_3 = E',$$

as

$$D - E_2 - E' = v_1 - v_2 - v_3 + v_4 = D_f,$$

for

$$f(v) = \begin{cases} 1 & \text{if } v \in \{v_1, v_4\}, \\ 0 & \text{else.} \end{cases}$$

Now for $D - E_3$ we get that

$$D - E_3 = v_1 - v_3 + v_4 \sim v_2 = E^*,$$

as

$$D - E_3 - E^* = v_1 - v_2 - v_3 + v_4 = D_f,$$

where f is the same as above. We see that for every divisor E of degree 1, $D - E$ is equivalent to an effective divisor. Therefore we conclude that D is of rank at least 1.

To prove that D is of rank of at least 2, we want to prove that $D - E$ is equivalent to an effective divisor for every effective divisor E of degree 2. First, let us consider the degree two divisor $E_5 = v_1 + v_4$. Note that

$$D - E_5 = 0,$$

is an effective divisor. On the other hand, consider for the divisor $E_6 = v_1 + v_2$ that

$$D - E_6 = v_4 - v_2 \in Div_0(G).$$

If $D - E_6$ is equivalent to an effective divisor, we must have that it is equivalent to the divisor 0, as that is the only effective divisor of degree 0 and the divisor cannot be equivalent to a divisor of a different degree. Note how $v_4 - v_2$ is not a principal divisor, therefore it is not equivalent to the divisor 0. We conclude that $D - E_6$ is not equivalent to an effective divisor. As E_6 is a divisor of degree 2, we get that D is not of at least rank 2. We conclude that D has rank 1.

In the next part of this example, we will see why it is possible for a divisor to have a negative rank. Consider the divisor

$$D' := -v_1 - v_2 \in Div_{-2}(G).$$

As D' is of degree -2, it is not equivalent to an effective divisor, as all divisors which are equivalent to D' also have degree -2. We conclude that D' does not have at least rank 0 and therefore has a negative rank. $\diamondsuit$

2.2 The Jacobian

Using the equivalence relation from the previous section, we can define a set of equivalence classes of divisors. In this section we will introduce Picard sets and a special Picard set called the Jacobian, which turns out to be a group. The Picard set of degree d is the set of divisors of degree d up to linear equivalence.

Definition. *The Picard set of degree d for any $d \in \mathbb{Z}$ is defined by*

$$Pic_d(G) := Div_d(G)/\sim .$$

For a divisor class $[D]$, all $D' \sim D$ are called *representatives* of the class $[D]$. Note that the degree of a divisor class is fixed, i.e. every representative in a class has the same degree, as the difference between two representatives is a principal divisor, which has degree zero. Also note how rank is a class property, as an equivalence relation is transitive. Let us now look at an example of divisors in the same equivalence class.

Example 2.4. *Let us again consider the graph $G = (V, E)$ as defined in Example 2.1. Two divisors on this graph are:*

$$D_1 = 3v_1 + 2v_2 - 5v_3 + v_4, \quad D_2 = -2v_1 + v_2 - 8v_3 + 10v_4.$$

Note that the degree of both these divisors is $3 + 2 - 5 + 1 = -2 + 1 - 8 + 10 = 1$. Therefore we know that

$$[D_1], [D_2] \in Pic_1(G).$$

Let us now check whether or not these divisors are equivalent. Since

$$D_1 - D_2 = 5v_1 + v_2 + 3v_3 - 9v_4,$$

we see that $D_1 - D_2 = D_f$ with

$$f(v) = \begin{cases} 2 & \text{if } v = v_1 \text{ or } v = v_2, \\ 1 & \text{if } v = v_3, \\ -3 & \text{if } v = v_4, \end{cases}$$

as can be seen in Example 2.2. We conclude that $D_1 \sim D_2$ and therefore

$$[D_1] = [D_2].$$

$\Diamond$

Though it might seem evident, we will see in the next proof that the different Picard groups can be bijectively mapped to one another. This proof might seem redundant, as some might say it is quite clear that for all d, we have that

$$\text{Pic}_d(G) \cong \mathbb{Z}^{|V|}/\sim,$$

but for completeness sake we added the following explicit theorem here.

Lemma 2.2. *Let $d, d' \in \mathbb{Z}$ and $q \in V$ be given. The map*

$$\phi_{d,d',q} : Pic_d(G) \to Pic_{d'}(G),$$

$$[D] \mapsto [D + (d' - d) \cdot q],$$

is a bijection.

Proof. First of all, note that if $D \in \text{Pic}_d(G)$ is a divisor of degree d, then we have that $D + (d' - d) \cdot q$ is a divisor of degree $d + (d' - d) = d'$, so indeed $D + (d' - d) \cdot q \in \text{Pic}_{d'}(G)$.

Now let $D_1 \sim D_2 \in \text{Pic}_d(G)$. Then we have that there exists a function $f : V \to \mathbb{Z}$, such that

$$D_1 - D_2 = D_f.$$

We see that:
$$D_1 + (d' - d) \cdot q - (D_2 + (d' - d) \cdot q) = D_1 - D_2 = D_f.$$
We get that $D_1 + (d' - d) \cdot q \sim D_2 + (d' - d) \cdot q$. We conclude that $\phi_{d,d',q}$ is a well-defined function.

Now we will prove that $\phi_{d,d',q}$ is an injective function. Let $D_1, D_2 \in \text{Pic}_d(G)$ such that $\phi_{d,d',q}(D_1) = \phi_{d,d',q}(D_2)$. This means that

$$D_1 + (d' - d) \cdot q \sim D_2 + (d' - d) \cdot q,$$

from which by definition follows that

$$D_1 + (d' - d) \cdot q - (D_2 + (d' - d) \cdot q) = D_f,$$

for some $f : V \to \mathbb{Z}$. From this it immediately follows that

$$D_1 - D_2 = D_f,$$

i.e. $D_1 \sim D_2$. We conclude that $\phi_{d,d',q}$ is injective.

We will now show that the inverse of $\phi_{d,d',q}$ is

$$\phi^{-1}_{d,d',q} := \phi_{d',d,q}.$$

This is indeed the inverse of $\phi_{d,d',q}$, as

$$\phi^{-1}_{d,d',q}(\phi_{d,d',q}([D])) = \phi^{-1}_{d,d',q}([D+(d'-d)\cdot q]) = \phi_{d',d,q}([D+(d'-d)\cdot q]) = [D+(d'-d+d-d')]\cdot q = [D],$$

and

$$\phi_{d,d',q}(\phi^{-1}_{d,d',q}([D])) = \phi_{d,d',q}([D + (d - d') \cdot q]) = [D + (d - d' + d' - d) \cdot q] = [D].$$

Using the same argument as above, we know $\phi^{-1}_{d,d',q}$ is also injective. We conclude that $\phi_{d,d',q}$ is a bijective function. □

When $d = 0$, the Picard group has a special name: the Jacobian. That is, the Jacobian is defined as
$$\text{Jac}(G) := \text{Div}_0(G)/\sim.$$
As $\text{Prin}(G) \subseteq \text{Div}_0(G)$, we can write this definition as a quotient group.

Definition. *The Jacobian is defined by*

$$Jac(G) = Div_0(G)/Prin(G).$$

Because the divisors in this set have degree zero, we can add two divisors pointwise and the result would again be a divisor of degree zero. This operation makes the Jacobian into a group.

Example 2.5. *Let us again consider the graph $G = (V, E)$ as in Example 2.1. Two divisor classes in the Jacobian are*

$$[v_1 + 3v_3 - 4v_4], [-3v_1 + 2v_2 + 2v_3 - v_4].$$

The addition of these two classes is:

$$[v_1 + 3v_3 - 4v_4 + -3v_1 + 2v_2 + 2v_3 - v_4] = [-2v_1 + 2v_2 + 5v_3 - 5v_4].$$

Note that indeed this class is again in the Jacobian, as its degree is equal to zero. ♢

This addition of divisors within the Jacobian is not just some operation on the Jacobian. In fact, this addition operator makes the Jacobian into a group, which we will discuss in the following lemma. This is a property noticed in a lot of literature, for instance in [BS13] it is noted that $\mathrm{Prin}(G)$ and $\mathrm{Div}_0(G)$ are free Abelian groups of rank $n-1$ and therefore its quotient group, the Jacobian, must be a finite Abelian group. In addition to this statement, we have added this lemma, which is a short direct proof of the Jacobian being an Abelian group using the definition of a group.

Lemma 2.3. *The Jacobian forms an Abelian group under addition $\boxplus$ defined by*

$$[D_1] \boxplus [D_2] := [D_1 + D_2],$$

where + denotes point-wise addition.

Proof. As we are doing point-wise addition, the action is commutative and associative. Note that the class [0], the class with the divisor that has all constants equal to zero, is the neutral element of this group operation.

Now we will show that inverses exist. Let $[D_1] \in \mathrm{Jac}(G)$ be some equivalence class of divisors. Let $a_v \in \mathbb{Z}$ for all $v \in V$ such that $D_1 = \sum_{v\in V} a_v \cdot v$. Then we see that $D_2 := \sum_{v\in V} -a_v \cdot v \in \mathrm{Jac}(G)$, as $-a_v \in \mathbb{Z}$ for all $v \in V$ and $\sum_{v\in V} -a_v = -\sum_{v\in V} a_v = -0 = 0$. We see that

$$[D_1] \boxplus [D_2] = [D_1 + D_2] = \left[\sum_{v\in V}(a_v - a_v)\cdot v\right] = [0].$$

We see that D_2 is the inverse of D_1. We conclude the operation $\boxplus$ makes $\mathrm{Jac}(G)$ into an Abelian group. □

2.2.1 Isomorphism with sandpile group

In this subsection we want to prove an isomorphism between the recurrent sandpile group $(\mathcal{R}, \oplus)$ from Section 1.5 and the Jacobian. In order to do this, we want to rewrite the equivalence relation in terms of the Laplacian matrix introduced in Section 1.1.1. Using this rewriting, we obtain the following result, which we will need as a tool in the next theorem. This lemma is an original contribution.

Lemma 2.4. *Let $G = (V, E)$ and $d \in \mathbb{Z}$ be given. Let $D_1, D_2 \in Pic_d(G)$. Then $D_1 \sim D_2$ if and only if we have*

$$D_1 = D_2 + Lf,$$

for some $f \in \mathbb{Z}^{|V|}$, and where L is the $V \times V$ Laplacian matrix and D_1 and D_2 are written in vector form.

Proof. We want to express a principal divisors $D_f = \sum_{v\in V} a_v \cdot v$ in terms of the Laplacian matrix. Remember that the constants a_v for all $v \in V$ were defined as

$$a_v = \sum_{(v,w)\in E} (f(v) - f(w)).$$

We can rewrite this expression to

$$a_v = \deg_G(v) \cdot f(v) - \sum_{(v,w)\in E} f(w).$$

If we write D as a vector, we can see that

$$D = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ v_{|V|} \end{pmatrix} = L \cdot \begin{pmatrix} f(v_1) \\ f(v_2) \\ \vdots \\ f(v_{|V|}) \end{pmatrix},$$

where L is the $V \times V$ Laplacian matrix as introduced in Section 1.1.1. By definition, we know that $D_1 \sim D_2$ if and only if there is a principal divisor D_f such that:

$$D_1 - D_2 = D_f.$$

We can express the divisors as vectors to obtain that $D_1 \sim D_2$ if and only if there exists a $f \in \mathbb{Z}^{|V|}$ such that

$$D_1 - D_2 = Lf,$$

which is equivalent to

$$D_1 = D_2 + Lf.$$

□

Let us look at an example of two equivalent divisors.

Example 2.6. *In Example 2.4 we saw that the divisors*

$$D_1 = 3v_1 + 2v_2 - 5v_3 + v_4, \quad D_2 = -2v_1 + v_2 - 8v_3 + 10v_4,$$

are equivalent divisors on the graph of Example 2.1. If we write the divisors in vector form, we see that:

$$D_1 = \begin{pmatrix} 3 \\ 2 \\ -5 \\ 1 \end{pmatrix} = \begin{pmatrix} -2 \\ 1 \\ -8 \\ 10 \end{pmatrix} - \begin{pmatrix} 2 & -1 & 0 & -1 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ -1 & 0 & -1 & 2 \end{pmatrix} \begin{pmatrix} 2 \\ 2 \\ 1 \\ -3 \end{pmatrix} = D_2 + Lf.$$

◊

In Chapter 1 we saw how we could use the Laplacian matrix to relate unstable configurations to stable configurations. Now that we know the equivalence relation on divisors is also related to the Laplacian, one might wonder if we could relate divisor classes in $(\text{Jac}(G), \boxplus)$ to the group of recurrent configurations $(\mathcal{R}, \oplus)$ from the previous chapter in a way that preserves the group structure. In the following theorem, we will show that the recurrent sandpile group $(\mathcal{R}, \oplus)$ introduced in Section 1.5 is isomorphic to the Jacobian group $(\text{Jac}(G), \boxplus)$.

One big difference between (recurrent) configurations and divisors is that in a configuration we do not give a height to the sink s, while in a divisor we do have a constant in front of s. However, if we only look at divisors of a fixed the degree d and the constants a_v for all $v \in V \setminus \{s\}$ are known, then a_s must be equal to d minus the sum of all a_v for the degree of the divisor to be d. This is the principle we use to naturally map a configuration η to a divisor $D = \sum_{v \in V} a_v \cdot v$ of degree d, i.e. we define the constants a_v of D for $v \in V$ to be equal to

$$a_v := \begin{cases} \eta(v) & \text{if } v \neq s, \\ d - \sum_{v \in V \setminus \{s\}} \eta(v) & \text{if } v = s. \end{cases}$$

In the following theorem, we specifically want to look at the Jacobian group in this proof as this is the only Picard set on which we have defined a group structure. This result is suggested in the literature, for instance [Big99a, Sec. 7]. We have included a proof here.

Theorem 2.5. *Let $G = (V, E)$ be a simple graph with $s = v_n \in V$ a marked vertex and $n = |V|$. Then*

$$f : (\mathcal{R}, \oplus) \to (Jac(G), \boxplus), \eta \mapsto [\eta'],$$

where

$$\eta' := \sum_{v \in V \setminus \{s\}} \eta(v) \cdot v - \left(\sum_{v \in V \setminus \{s\}} \eta(v) \right) \cdot s \in Div_0(G),$$

is a group isomorphism between the group of recurrent configurations and the Jacobian.

Proof. Let $G = (V, E)$ a graph with $n = |V|$ and $s = v_n \in V$ a marked vertex. Let $\eta \in \mathcal{R}$. Note that the degree of η' equals

$$\deg(D) = \sum_{v \in V \setminus \{s\}} \eta(v) - \left(\sum_{v \in V \setminus \{s\}} \eta(v) \right) = 0,$$

so $[\eta']$ is indeed a divisor class in $\mathrm{Pic}_0(G)$ and therefore in the Jacobian.

Now we will argue why this function preserves the group action. Let $\eta_1, \eta_2 \in \mathcal{R}$. We want to prove that $f(\eta_1 \oplus \eta_2) = f(\eta_1) \boxplus f(\eta_2)$. We have that

$$\eta_1 \oplus \eta_2 = S(\eta_1 + \eta_2) = \eta_1 + \eta_2 - L'x',$$

for some vector $x' \in \mathbb{Z}^{n-1}$ that is the odometer function of the stabilization of $\eta_1 + \eta_2$. Then we see that:

$$f(\eta_1 \oplus \eta_2) = f(\eta_1 + \eta_2 - L'x') = [(\eta_1 + \eta_2 - L'x')'].$$

We get that:

$$\begin{aligned}(\eta_1 + \eta_2 - L'x')' = &\left(\sum_{v_i \in V \setminus \{s\}} (\eta_1(v_i) + \eta_2(v_i) + x_i \cdot \deg_G(v_i) - \sum_{v_j \in V \setminus \{s\}, v_j \sim v} x_j) \cdot v_i \right) \\ &- \left(\sum_{v_i \in V \setminus \{s\}} (\eta_1(v_i) + \eta_2(v_i) + x_i \cdot \deg_G(v_i) - \sum_{v_j \in V \setminus \{s\}, v_j \sim v_i} x_j) \right) \cdot s\end{aligned}$$

where $x_i = x_i'$ for $i \in \{1, 2, ..., n-1\}$ and $x_n = 0$. Note that

$$\begin{aligned}\sum_{v_i \in V \setminus \{s\}} \left(x_i \cdot \deg_G(v_i) - \sum_{v_j \in V \setminus \{s\}, v_j \sim v_i} x_j \right) &= \sum_{v_i \in V \setminus \{s\}} \left(\sum_{v_j \in V, v_j \sim v_i} x_i - x_j \right), \\ &= \sum_{v_i \in V \setminus \{s\}, v_i \sim s} x_i - x_n, \\ &= \sum_{v_i \in V \setminus \{s\}, v \sim s} x_i,\end{aligned}$$

as $x_n = 0$. Therefore we get that

$$\begin{aligned}(\eta_1 + \eta_2 - L'x')' = &\left(\sum_{v_i \in V \setminus \{s\}} (\eta_1(v_i) + \eta_2(v_i) + x_i \cdot \deg_G(v_i) - \sum_{v_j \in V \setminus \{s\}, v_j \sim v} x_j) \cdot v_i \right) \\ &- \left(\sum_{v_i \in V \setminus \{s\}} (\eta_1(v_i) + \eta_2(v_i) + x_i \cdot \deg_G(v_i) - \sum_{v_j \in V \setminus \{s\}, v_j \sim v_i} x_j) \right) \cdot s \\ = &\left(\sum_{v_i \in V \setminus \{s\}} (\eta_1(v_i) + \eta_2(v_i) + x_i \cdot \deg_G(v_i) - \sum_{v_j \in V \setminus \{s\}, v_j \sim v} x_j) \cdot v_i \right) \\ &- \left(\left(\sum_{v_i \in V \setminus \{s\}} (\eta_1(v_i) + \eta_2(v_i)) \right) + \sum_{v_i \in V \setminus \{s\}, v \sim s} x_i \right) \cdot s \\ = &\left(\sum_{v \in V \setminus \{s\}} (\eta_1(v) + \eta_2(v)) \cdot v \right) - \left(\sum_{v \in V \setminus \{s\}} (\eta_1(v) + \eta_2(v)) \right) \cdot s - Lx,\end{aligned}$$

as we defined x to be $x_i = x'_i$ for $i \in \{1, 2, ..., n-1\}$ and $x_n = 0$. In Lemma 2.4 we know that divisors are equivalent when their difference is the $V \times V$ Laplacian matrix multiplied with an integer-valued vector. Therefore we see that:

$$\begin{aligned}(\eta_1 + \eta_2 - L'x')' &= \left(\sum_{v \in V \setminus \{s\}} (\eta_1(v) + \eta_2(v)) \cdot v\right) - \left(\sum_{v \in V \setminus \{s\}} (\eta_1(v) + \eta_2(v))\right) \cdot s - Lx \\ &\sim \left(\sum_{v \in V \setminus \{s\}} (\eta_1(v) + \eta_2(v)) \cdot v\right) - \left(\sum_{v \in V \setminus \{s\}} (\eta_1(v) + \eta_2(v))\right) \cdot s = (\eta_1 + \eta_2)'.\end{aligned}$$

In conclusion, we have

$$\begin{aligned}f(\eta_1 \oplus \eta_2) &= [((\eta_1 + \eta_2) - L'x')'], \\ &= [(\eta_1 + \eta_2)'], \\ &= [\eta'_1 + \eta'_2], \\ &= f(\eta_1) \boxplus f(\eta_2),\end{aligned}$$

and we conclude that f preserves the group action.

What is left to prove is that f is a bijective function. First, let us prove that f is injective. Let $\eta_1, \eta_2 \in \mathcal{R}$ such that $f(\eta_1) = f(\eta_2)$. Then we have by definition that

$$[\eta'_1] = [\eta'_2].$$

Remember that $n = |V|$. By Lemma 2.4 there exists a vector $x \in \mathbb{Z}^n$ such that

$$\eta'_1 = \eta'_2 - Lx.$$

We now want to argue why there exists an $y \in \mathbb{Z}^n$ with $y_n = 0$ such

$$\eta'_1 = \eta'_2 - Ly.$$

First note that by definition of the $V \times V$ Laplacian matrix we have

$$L \cdot \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} = \begin{pmatrix} \deg_G(v_1) - \sum_{v \in V} \mathbb{1}_{v_1 \sim v} \\ \deg_G(v_2) - \sum_{v \in V} \mathbb{1}_{v_2 \sim v} \\ \vdots \\ \deg_G(v_n) - \sum_{v \in V} \mathbb{1}_{v_n \sim v} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix},$$

where $\mathbb{1}_{v,w}$ equals 1 if v and w are neighbours and 0 if they are not. Let

$$y := x - x_n \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \\ 1 \end{pmatrix}.$$

Note that $y_n = 0$. Combining the above, we get that

$$\eta'_1 = \eta'_2 - Lx = \eta'_2 - Ly.$$

It follows that

$$\eta_1 = \eta_2 - L'y',$$

where $y' \in \mathbb{Z}^{n-1}$ with $y'_i = y_i$ for $i \in \{1, 2, ..., n-1\}$. Now let us split the indices of y' in by their value being positive or negative:

$$I := \{i \in \{1, ..., n-1\} : y'_i \geq 0\},$$
$$J := \{i \in \{1, ..., n-1\} : y'_i < 0\} = \{1, ..., n-1\} \setminus I.$$

Then let $y_I \in \mathbb{Z}^{n-1}_{\geq 0}$ be the vector such that $y_{I,i} = y_i$ if $i \in I$ and $y_{I,i} = 0$ else. Likewise, let $y_J \in \mathbb{Z}^{n-1}_{<0}$ be the vector such that $y_{J,j} = y_j$ if $j \in J$ and $y_{J,j} = 0$ else. Let

$$\eta^* := \eta_2 - L'y_I.$$

By definition of stabilization, we know that

$$S(\eta^*) = \eta_2.$$

We also can rewrite $\eta_2 - L'y_I$ to get:

$$\eta_2 - L'y_I = \eta_1 + L'y' - L'y_I = \eta_1 + L'y_J = \eta_1 - L'(-y_J).$$

Therefore by definition we also see that:

$$S(\eta^*) = S(\eta_1 - L'(-y_J)) = \eta_1.$$

As both η_1 and η_2 are recurrent (stable) configurations and both y_I and $-y_J$ are vectors in $\mathbb{Z}^{n-1}_{\geq 0}$, we see that indeed the stabilization of η^* equals both η_1 and η_2. By Lemma 1.1, we know that stabilization is well-defined and has a unique result. Therefore, we must have that $\eta_1 = \eta_2$ and we conclude that f is injective.

Now let us prove that f is surjective. Let $[D] \in \mathrm{Jac}(G)$ be any divisor class. We want to find a representative of the class of the form

$$D' = \left(\sum_{v \in V \setminus \{s\}} a_v \cdot v \right) - \left(\sum_{v \in V \setminus \{s\}} a_v \right) \cdot s,$$

such that $\eta(v) := a_v$ for all $v \in V \setminus \{s\}$ is a recurrent configuration. First, we want to find a representative $D_{big} \sim D$ such that for all $v \in V \setminus \{s\}$ we have

$$D_{big}(v) \geq \deg_G(v).$$

This is possible by creating a sequence of equivalent divisors

$$D = D_0 \sim D_1 \sim D_2 \sim ... \sim D_{k+1} = D_k = D_{big},$$

where

$$D_i := D_{i-1} + Lx_i,$$

with $x_i \in \mathbb{Z}^{|V|-1}$ defined as

$$x_i(j) = \begin{cases} 1 & \text{if } v_j \neq s \text{ and } D_{i-1}(v_j) < \deg_G(v_j), \\ 0 & \text{else,} \end{cases}$$

for all $i \in \{1, ..., k\}$ and $v_j \in V \setminus \{s\}$. By Lemma 2.4 we know that these divisors are equivalent. We continue until x_k equals the zero vector, i.e. $D_k(v) \geq \deg_G(v)$ for all $v \in V \setminus \{s\}$. To prove this sequence is always finite, we have to take a look at the s-weight of the divisors. As we will only add (at most $\deg_G(v)$) height to $D_i(v)$ if $D_{i-1}(v) < \deg_G(v)$, we have that

$$D_i(v) < \max\{2 \cdot \deg_G(v), D(v)\},$$

for all $i \in \{0, ..., k\}$ and $v \in V \setminus \{s\}$. Therefore the s-weight of D_i is

$$w_s(D_i) \leq \sum_{v \in V \setminus \{s\}} \max\{2 \cdot \deg_G(v), D(v)\}.$$

Also note that the s-weight increases in the sequence, i.e.

$$w_s(D_{i-1}) \leq w_s(D_i),$$

for all $i \in \{1, .., k\}$, as $(Lx_i)_s \leq 0$. If we have $x_{i,v} = 1$ for some $v \sim s$ and some $i \in \{1, .., k\}$, we get that

$$w_s(D_{i-1}) < w_s(D_i).$$

As the graph G is connected, we have that every amount finite amount of steps in the sequence, a neighbour v of s will have $D(v) < \deg_G(v)$. Therefore, every finite amount of steps, the s-weight in the sequence strictly increases. As the s-weight is bounded, we must have that this sequence is finite. We conclude we can indeed find such a $D_{big} \sim D$.

Now let $\eta_{big}(v) := D_{big}(v)$ for all $v \in V \setminus \{s\}$. By construction, as $\eta_{big}(v) \geq \deg(v)$ for all $v \in V \setminus \{s\}$, we know that η is an unstable configuration. Let $\eta := S(\eta_{big})$ be the stabilization of the configuration. We know that $\eta_{big}(v) > \eta_{max}(v)$ for all $v \in V \setminus \{s\}$, where $\eta_{max}(v) = \deg_G(v) - 1$ for all $v \in V \setminus \{s\}$. By Theorem 1.2 (2(b)) we know that $\eta = S(\eta_{big})$ is a recurrent configuration, i.e. $\eta \in \mathcal{R}$. What is left to prove is that $f(\eta) = [D]$. As η is the stabilization of η_{big}, we can use Lemma 1.1 to find a unique vector $x' \in \mathbb{Z}^{n-1}_{\geq 0}$ for which we can write the configurations in vector form in the following way:

$$\eta = \eta_{big} - L'x',$$

where L' is the $V \setminus \{s\} \times V \setminus \{s\}$ Laplacian. As done above, it follows that

$$f(\eta) = [\eta'] = [\eta'_{big} - Lx] = [D_{big} - Lx],$$

where $x \in \mathbb{Z}^n_{\geq 0}$ with $x_i = x'_i$ for $i \in \{1, 2, ..., n-1\}$ and $x_n = 0$. By Lemma 2.4 we conclude that

$$f(\eta) = [D_{big} - Lx] = [D_{big}] = [D],$$

i.e. f is surjective. □

In Lemma 2.2 we saw that there is a bijection between Picard sets of different degrees. Combining the mapping $\phi_{o,d}$ from the Jacobian to the Picard set of degree d with the mapping f from the recurrent group to the Jacobian, we get a bijection between the recurrent group and any Picard set.

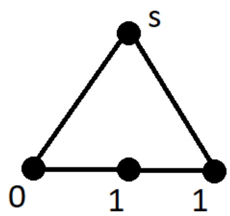
Corollary 2.6. *Let $d \in \mathbb{Z}$ and $q \in V$ be given. The map*

$$\phi_{0,d,q} \circ f : (\mathcal{R}, \oplus) \to Pic_d(G),$$

where $\phi_{0,d,q}$ is the mapping from Lemma 2.2 and f the mapping from Theorem 2.5 gives a bijection between the recurrent configurations in $\mathcal{R}$ and the equivalence classes of divisors in $Pic_d(G)$.

Let us now look at a concrete example of how to obtain the divisor in $\mathrm{Pic}_d(G)$ that a recurrent configuration gets mapped to.

Example 2.7. *Let us consider the recurrent configuration η:*



on the graph from Example 2.1. Note that for the recurrent configuration, we marked v_4 to be the sink.

This element gets mapped to

$$f(\eta) = [v_2 + v_3 - 2v_4] \in Jac(G)$$

by f. Now if we want to map this with $\phi_{0,4,q}$ to $Pic_4(G)$, we have to chose a vertex q. Natural is to choose the same vertex as the sink, so $q = s = v_4$. Then we get that

$$\phi_{0,4,v_4} \circ f : (\mathcal{R}, \oplus) \to Pic_d(G),$$
$$\eta \mapsto \left[\left(\sum_{v \in V \setminus \{v_4\}} \eta(v) \cdot v \right) + \left(4 - \sum_{v \in V \setminus \{v_4\}} \eta(v) \right) \cdot v_4 \right].$$

We see that η gets mapped to

$$\phi_{0,4,v_4}(f(\eta)) = [v_2 + v_3 + 2v_4] \in Pic_4(G).$$

It is also possible to chose q not equal to the sink. For instance if we choose $q = v_1$, we would get that

$$\phi_{0,4,v_1} \circ f : (\mathcal{R}, \oplus) \to Pic_d(G),$$
$$\eta \mapsto \left[\left(\sum_{v \in V \setminus \{v_4, v_1\}} \eta(v) \cdot v \right) + (4 + \eta(v_1)) \cdot v_1 - \sum_{v \in V \setminus \{v_4\}} \eta(v) \cdot v_4 \right].$$

In this case, we see that

$$\phi_{0,4,v_1}(f(\eta)) = [4v_1 + v_2 + v_3 - 2v_4] \in Pic_4(G).$$

Still, by the previous corollary we know that $\phi_{0,4,v} \circ f$ provides a bijection between $(\mathcal{R}, \oplus)$ and $Pic_4(G)$ for any given $v \in V$. ◊

2.3 Q-reduced divisors

In this section, we will look into a subset of divisors called q-reduced divisors. We will show that every equivalence class of divisors contains a unique q-reduced divisor. Also, we will look into a burning algorithm for q-reduced divisors, which can also be used to determine whether or not a divisor is q-reduced.

Definition. *Let $G = (V, E)$ a graph and $q \in V$ some vertex. A divisor $D \in Div(G)$ is q-reduced if it satisfies the following two conditions:*

- *$D(v) \geq 0$ for all $v \in V \setminus \{q\}$*
- *For every non-empty $A \subseteq V \setminus \{q\}$, there exists a vertex $v \in A$ such that $D(v) < outdeg_A(v)$.*

There are similarities between q-reduced divisors and stable configurations. If for the second condition we would look at subsets A of size one, i.e. $A = \{v\}$ for some $v \in V \setminus \{q\}$, we must have that for every such A it holds that

$$D(v) < \deg_G(v),$$

which is the condition for a configuration to be stable. The second condition for being q-reduced can therefore be interpreted as a version of stable configurations where not only vertices are stable, but also where subsets of all vertices except q should be stable. Let us look at this in an example.

Example 2.8. *Let us again consider the graph from Example 2.1. Let $q = v_4$. Consider the divisor:*

$$D = v_1 + v_2 \in Pic_2(G).$$

This is the image under f of the recurrent configuration η with height 1 at v_1 and v_2, height 0 at v_3 and considering v_4 as the sink. Note that as this is a recurrent configuration. As a configuration with v_4 as the sink, this is stable, as

$$\eta(v_1) = \eta(v_2) = 1 < \deg_G(v_1) = \deg_G(v_2) = 2,$$

and

$$\eta(v_3) = 0 < \deg_G(v_3) = 2.$$

However, the divisor is not a v_4-reduced divisor. Let $A = \{v_1, v_2\}$. Then we have that

$$D(v_1) = 1 \geq outdeg_A(v_1) = 1, \quad D(v_2) = 1 \geq outdeg_A(v_2) = 1.$$

We see that in vector form we get

$$D - L\begin{pmatrix}1\\1\\0\\0\end{pmatrix} = \begin{pmatrix}1\\1\\0\\0\end{pmatrix} - \begin{pmatrix}2 & -1 & 0 & -1\\-1 & 2 & -1 & 0\\0 & -1 & 2 & -1\\-1 & 0 & -1 & 2\end{pmatrix}\begin{pmatrix}1\\1\\0\\0\end{pmatrix} = \begin{pmatrix}0\\0\\1\\1\end{pmatrix}.$$

If we were considering a configuration with height one at v_1 and v_2, height zero at v_3 and the sink at v_4, we would not be allowed to topple the configuration at any vertex. However, we can now "topple" the divisor at full subsets A if this results in the constants in the divisor staying non-negative for all vertices in $V \setminus \{q\}$. We get the divisor:

$$D' = v_3 + v_4.$$

By Lemma 2.4 we know that $D' \in [D]$. Now note that D' is a v_4-reduced divisor. We have $D(v) \geq 0$ for $v \in \{v_1, v_2, v_3\}$. Also, if for instance we take $A = \{v_2, v_3\}$, we see that

$$D(v_2) = 0 < outdeg_A(v_2) = 1.$$

$\Diamond$

In the example we found a q-reduced divisor in the equivalence class of a certain divisor. The fact that we could find a q-reduced divisor is no accident. In the next theorem we will state that for every vertex q and every divisor D, there is a q-reduced divisor in the equivalence class of D.

Theorem 2.7. *[BS13, Cor. 4.13] Let $q \in V$ and $D \in Div(G)$. Then there is a unique q-reduced divisor $D' \in Div(G)$ that is linearly equivalent to D.*

For a proof of this theorem, see [BS13, Cor. 4.13].

We call the unique q-reduced divisor of a class $[D] \in \text{Pic}_d(G)$ the *q-reduced representative of $[D]$*.

In Corollary 2.6 we saw how there is a bijection between the recurrent configurations of a graph and the classes of divisors in $\text{Pic}_d(G)$ for any given d. Now that we have seen above that there is a unique q-reduced divisor in every class, we combine Corollary 2.6 and Theorem 2.7 to immediately obtain the following result.

Corollary 2.8. *Let $d \in \mathbb{Z}$ be given. There is a bijection between the recurrent configurations of a graph and the q-reduced divisors in $Pic_d(G)$.*

2.3.1 Burning algorithm for q-reduced divisors

In this section we will further explore the bijection between the recurrent configurations of G and the q-reduced divisors in $\text{Pic}_d(G)$ for any given $d \in \mathbb{Z}$. We will first look into a modified version of the burning Algorithm 1, which will take a q-reduced divisor as input. Then we will look into an explicit bijection between recurrent configurations and q-reduced divisors.

The burning algorithm has been modified for q-reduced divisors by [BS13, Alg. 2], which we will now discuss.

The input of this algorithm is a graph $G = (V, E)$ with a fixed ordering on E, a marked vertex $q \in V$ and a q-reduced divisor $D = \sum_{v \in V} a_v \cdot v$. Instead of burning all adjacent edges to a vertex when this vertex is burned as in Algorithm 1, we will burn the edges one by one and burn a vertex if the constant a_v of a vertex is equal to the amount of previously burned vertices. This translates to the pseudocode that can be found in Algorithm 2.

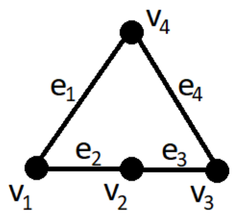
Algorithm 2 The burning algorithm for q-reduced divisors [BS13, Alg. 2]

Input: A graph $G = (V, E)$, an ordering on E, a vertex $q \in V$ and a q-reduced divisor $D = \sum_{v \in V} a_v \cdot v$.
Output: A spanning tree $G' = (V', E')$ of G.

$V' := \{q\}$
$E' := \emptyset$
$E_b := \emptyset$ ▷ The set of burned edges
while $V' \neq V$ **do**
  $e' := \min\{e = (s, t) \in E | e \notin E_b, s \in V', t \notin V'\}$
  $v := t$
  if $a_v = |\{e \in E_b | v \in e\}|$ **then**
    $V' = V' \cup \{v\}$ ▷ Burn the vertex
    $E' = E' \cup \{e'\}$
  end if
  $E_b = E_b \cup \{e'\}$ ▷ Burn the edge
end while
Output: $G' = (V', E')$

Let us look at an example of the execution of the algorithm.

Example 2.9. *Let us again consider the graph:*



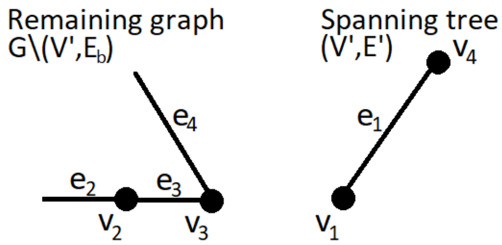
Let $q = v_4$ and the edges be ordered as $e_1 < e_2 < e_3 < e_4$. In the previous example we saw that a q-reduced divisor is

$$D' = v_4 + v_3.$$

When we start the algorithm, we have that $E_b = E' = \emptyset$ and $V' = \{q\} = \{v_4\}$. In the first step, we will take the smallest edge that is connected to the burned vertex v_4, so $e' := e_1$. We see that v_1 is the other vertex connected to this edge and

$$a_{v_1} = 0 = |\{e \in E_b | v_1 \in e\}|,$$

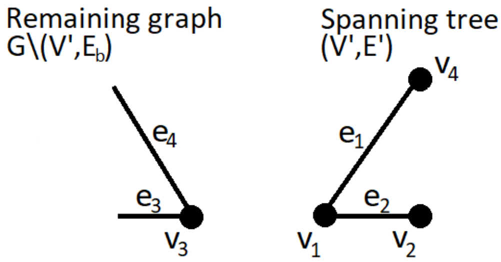
so we will burn both v_1 and e_1 in this step, which we will both add to the spanning tree. We get $E_b = E' = \{e_1\}$ and $V' = \{v_1, v_4\}$. The unburned part of the graph and the spanning tree up until now look like this:



In the next step, the smallest edge that has not been burned and adjacent to a burned vertex is $e' := e_2$. We see that v_2 is the other vertex connected to this edge and

$$a_{v_2} = 0 = |\{e \in E_b | v_2 \in e\}|,$$

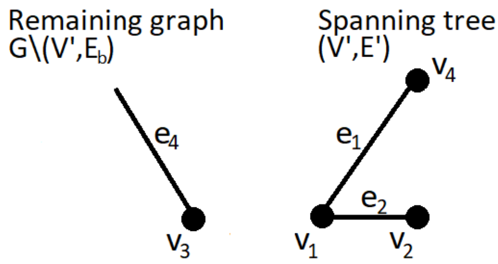
so we will burn both v_2 and e_2 in this step, which we will both add to the spanning tree. We get $E_b = E' = \{e_1, e_2\}$ and $V' = \{v_1, v_2, v_4\}$:



In the third step, the smallest edge that has not been burned and adjacent to a burned vertex is $e' := e_3$. We see that v_3 is the other vertex connected to this edge and

$$a_{v_3} = 1 \neq 0 = |\{e \in E_b | v_3 \in e\}|,$$

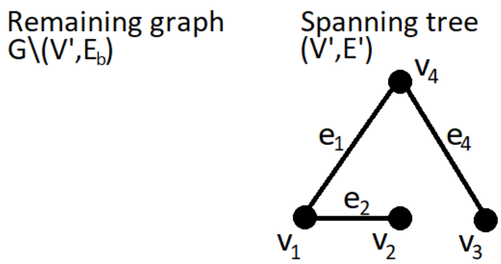
so we will not burn v_3. We get $E_b = \{e_1, e_2, e_3\}$, $E' = \{e_1, e_2\}$ and $V' = \{v_1, v_2, v_4\}$:



Now in the following step, the smallest and only unburned edge that is left is $e' := e_4$. We see that

$$a_{v_3} = 1 = |\{e \in E_b | v_3 \in e\}|,$$

so we will burn both v_3 and e_4 in this step, which we will both add to the spanning tree. We get $E_b = E$, $E' = \{e_1, e_2, e_4\}$ and $V' = V$:



As we now have $V' = V$, the algorithm terminates and outputs spanning tree $G' = (V', E')$. $\diamondsuit$

We also want to have a way to get from a spanning tree to a q-reduced divisor. The pseudocode of this reversed algorithm can be found in Algorithm 3 and is taken from [BS13, Alg. 3].

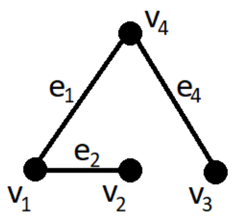
Algorithm 3 The burning algorithm: from spanning tree to q-reduced divisors [BS13, Alg. 3]

Input: A graph $G = (V, E)$, an ordering on E, a vertex $q \in V$, a degree $d \in \mathbb{Z}$ and a spanning tree (V', E') of G.
Output: A q-reduced divisor $D = \sum_{v \in V} a_v \cdot v$ of degree d.

$E_b = \emptyset$
$V_b = \{q\}$
while $V_b \neq V$ **do**
  $e' = \min\{e = (s,t) \in E | e \notin E_b, s \in V_b, t \notin V_b\}$
  if $e' \in E$ **then**
    $v = t$
    $a_v := |\{e \in E_b | v \in e\}|$ ▷ Burn the vertex
    $V_b = V_b \cup \{v\}$
  end if
  $E_b = E_b \cup \{e'\}$ ▷ Burn the edge
end while
$a_q = d - \sum_{v \in V \setminus \{q\}} a_v$
Output: $D = \sum_{v \in V} a_v \cdot v$

Let us look at an example of this algorithm in practice.

Example 2.10. *Let us consider the graph from Example 2.9 and the spanning tree:*



Let $q = v_4$, $d = 2$ and the edges be ordered as $e_1 < e_2 < e_3 < e_4$. We start the algorithm with $E_b = \emptyset$ and $V_b = v_4$. The first edge we will burn is the smallest that has one adjacent vertex in V_b and one adjacent vertex not in V_b. In this first step, this is e_1. We see that e_1 is in the spanning tree, so we will burn a vertex as well, namely vertex v_1. To burn this vertex, we get

$$a_{v_1} = |\{e \in E_b | v_1 \in e\}| = 0.$$

We add v_1 to the set of burned vertices V_b and e_1 to the set of burned edges E_b.

In the next step, we will burn edge e_2, which is now the smallest unburned edge with one burned incident vertex. We see that e_2 is in the spanning tree, so we will also burn the unburned incident vertex, which is v_2. We get

$$a_{v_2} = |\{e \in E_b | v_1 \in e\}| = 0.$$

We add v_2 to the set of burned vertices V_b and e_2 to the set of burned edges E_b.

In the next step, we will burn edge e_3. As e_3 is not in the spanning tree, we will not burn any vertex. To conclude this step, we add e_3 to the set of burned edges E_b.

In the next and final step, we will burn the only remaining edge e_4. As e_4 is in the spanning tree, we burn vertex e_3 as well. We get that

$$a_{v_3} = |\{e \in E_b | v_1 \in e\}| = |\{e_3\}| = 1.$$

To burn the edge and the vertex, we add v_3 to the set of burned vertices V_b and e_4 to the set of burned edges E_b.

As we now have that $V_b = V$, we get out of the while loop of the algorithm. We get

$$a_{v_4} = d - \sum_{v \in V \setminus \{v_4\}} a_v = 2 - (0 + 0 + 1) = 1.$$

The output of the algorithm is $D = \sum_{v \in V} a_v \cdot v = v_3 + v_4$. Note that in Example 2.9 we saw that upon input $D = v_3 + v_4$ and the same ordering on the edges, we got as an output the spanning tree we started with in this example. $\diamondsuit$

The two Algorithms 2 and 3 form a bijection between spanning trees and the q-reduced divisors of degree d. The following lemma is taken from [BS13, Thm. 5.1]

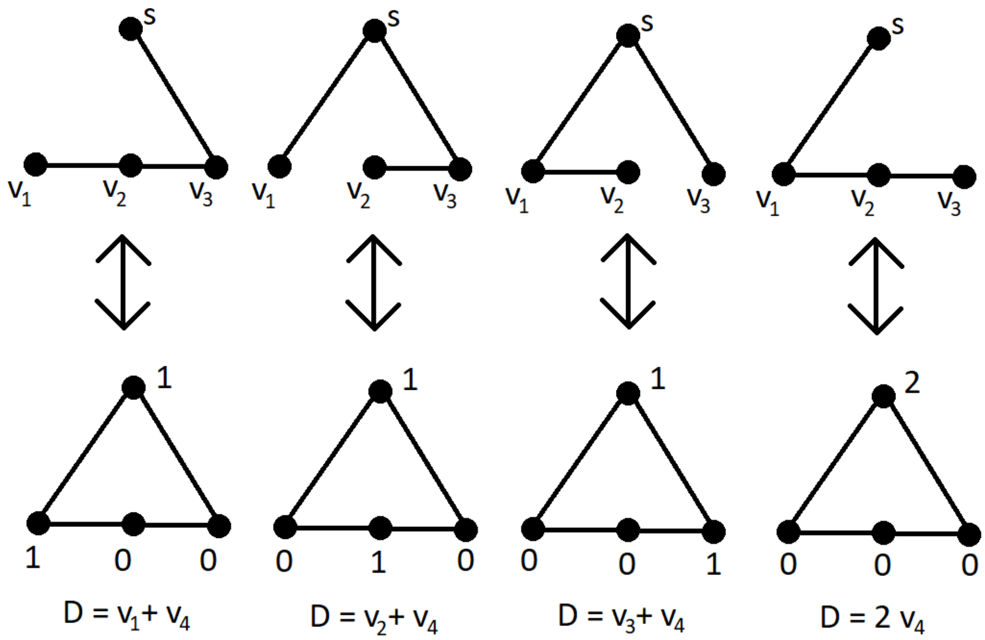
Lemma 2.9. *Let $d \in \mathbb{Z}$ and $q \in V$. Algorithm 2 gives a bijection between the q-reduced divisors of degree d and spanning trees of G, where Algorithm 3 is the inverse.*

In Theorem 2.7 we saw that every divisor class in $\mathrm{Pic}_d(G)$ have a unique q-reduced representative. We can combine this with the above result to find a bijection between $\mathrm{Pic}_d(G)$ and the spanning trees of G.

Corollary 2.10. *Let $d \in \mathbb{Z}$ and $q \in V$. Algorithm 2 and Algorithm 3 give rise to a bijection between the classes in $Pic_d(G)$ and spanning trees of G.*

In the following example, we will give a specific example of this bijection.

Example 2.11. *For this example, we will again be considering the graph from Example 2.9. Let $q = v_4$ and the edges be ordered as $e_1 < e_2 < e_3 < e_4$. All the spanning trees of the graph and the corresponding v_4-reduced divisors in $Pic_2(G)$ are:*



◊

As seen earlier in Corollary 2.8, there is a bijection between the recurrent configurations $\mathcal{R}$ and the q-reduced divisors of degree d. In Corollary 2.8 we found it via bijections with $\text{Pic}_d(G)$, but with these burning algorithms we can now also confirm this bijection by going via spanning trees, as we saw in 1.4 that the recurrent configurations are also in bijection with the spanning trees of G.

2.3.2 Weights of q-reduced divisors

In this section we will look into the q-weight of q-reduced divisors. Let us start with a small example to remind ourselves of the definition of q-weight.

Example 2.12. *In this example we will again consider the graph from Example 2.1. Let $q = v_4$. In Example 2.8 we saw that $D = v_1 + v_2$ is not a v_4-reduced divisor and that $D' = v_3 + v_4$ is the v_4-reduced representative of $[D]$ on this graph. We see that the v_4-weight of D is*

$$w_{v_4}(D) = \sum_{v \in V \setminus \{v_4\}} D(v) = 1 + 1 + 0 = 2.$$

and the v_4-weight of D' is

$$w_{v_4}(D') = \sum_{v \in V \setminus \{v_4\}} D(v) = 0 + 0 + 1 = 1.$$

◊

We see that a q-reduced divisor sometimes has a smaller q-weight than some of the other representatives in its class, whilst it will always have positive q-weight. We will see this in the following lemma, in which we will prove bounds on the q-weight of a q-reduced divisor. We have included a proof here. The proof of the upper bound is inspired by the proof of Lemma 1.7.

Lemma 2.11. *Let $G = (V \cup \{s\}, E)$ and $q \in V \cup \{s\}$. The q-weight of a q-reduced divisor D is $0 \leq w_q(D) \leq |E| - |V|$.*

Proof. Let D be any q-reduced divisor. By definition, we know that for every $v \in V \cup \{s\} \setminus \{q\}$ we have

$$D(v) \geq 0.$$

The lower bound directly follows:

$$w_q(D) = \sum_{v \in V \cup \{s\} \setminus \{q\}} D(v) \geq 0.$$

For proving the upper bound, we take a closer look at what happens in the burning algorithm for q-reduced divisors in Algorithm 2. In this algorithm, a vertex $v \neq q$ burns when the edge e' between v and an already burned neighbour burns and $D(v)$ equals the amount of adjecent edges to v that have burned before e' did. As we want all vertices to burn, we see that

$$\sum_{v \in V \cup \{s\} \setminus \{q\}} D(v) \leq |E| - |V|,$$

as otherwise not all vertices will be able to burn in the algorithm. This proves the upper bound. $\square$

Chapter 3

The identity element

For a set to be a group, there needs to be one element in the set that acts as the identity with the defined group action. In this section we will look at two algorithms to determine the identity element of the recurrent sandpile group $(\mathcal{R}, \oplus)$. The first algorithm is based on the isomorphism between the recurrent sandpile group $(\mathcal{R}, \oplus)$ and the Jacobian $(\text{Jac}(G), \boxplus)$ we saw in Theorem 2.5. The second algorithm is based on the method by Creutz from 1991 [Cre91]. After introducing these algorithms, we will analyze the results. The implementation of these algorithms in Python can be found in Appendix B. The implementation of the algorithms and analysis of the results are an original contribution of this thesis.

3.1 The Jacobian algorithm

Using the proof of Theorem 2.5, which states that the Jacobian $(\text{Jac}(G), \boxplus)$ is isomorphic to the recurrent sandpile group $(\mathcal{R}, \oplus)$, we can use the equivalence class of neutral element in the Jacobian group to find the corresponding element in the recurrent sandpile group. We want to use this equivalence as the neutral element in the Jacobian group is quite easy to find. It is the equivalence class with the empty sum, i.e. the equivalence class containing the divisor $D_0 = \sum_{v \in V} a_v \cdot v$ with $a_v = 0$ for all $v \in V$. In the proof of Theorem 2.5, we find the preimage $f^{-1}([D_0])$, by creating a sequence of equivalent divisors

$$D_0 \sim D_1 \sim ... \sim D_n,$$

where

$$D_i := D_{i-1} + Lx_i,$$

with $x_i \in \mathbb{Z}^{|V|-1}$ defined as

$$x_i(j) = \begin{cases} 1 & \text{if } v_j \neq s \text{ and } D_{i-1}(v_j) < \deg_G(v_j), \\ 0 & \text{else,} \end{cases}$$

for all $i \in \{1, ..., k\}$ and $v_j \in V \setminus \{s\}$. This sequence continues until x_k equals the zero vector, i.e. $D_k(v) \geq \deg_G(v)$ for all $v \in V \setminus \{s\}$. In the proof of Theorem 2.5 we proved this is happens after a finite number of steps.

From this sequence, we obtained $\eta(v) := D_n(v)$ for all $v \in V \setminus \{s\}$, which is an unstable configuration. The stabilization of this configuration gives us the recurrent configuration in $(\mathcal{R}, \oplus)$ that is the neutral element in this group. Example code of this algorithm to find the identity element of the sandpile group $(\mathcal{R}, \oplus)$ on the lattice graph $G = (V \cup \{s\}, E)$ with $V = [-n, n]^2 \cap \mathbb{Z}^2$ for a given n implemented in Python can be found in B.2.

3.2 Creutz's algorithm

In 1991 Creutz gave an algorithm for finding the neutral element of the Abelian sandpile group on a lattice graph $G = (V \cup \{s\}, E)$ with $V = [-n, n]^2 \cap \mathbb{Z}^2$ for a given n [Cre91]. In order to find this identity element, Creutz considers the configuration I_0, which has height two at the corners of the lattice, height one at edges and height zero everywhere else. This element is not a recurrent configuration, but does have the property that when added to a recursive element, it acts neutral. Note that when all sites of I_0 get toppled all at once, we get the configuration where the height equals zero everywhere.

Now define $I_n = I_{n-1} \oplus I_{n-1}$, where we recall that $\oplus$ is the operation of adding two configurations and then stabilizing the result. After a finite number of these recursive calculations, we will reach a point where $I_n = I_{n+1}$ [Cre91]. Note that this implies that for all $t > n$, $I_n = I_t$ as well. This means we have found the recursive neutral element. Example code of this algorithm to find the identity element of the sandpile group $(\mathcal{R}, \oplus)$ on the lattice graph $G = (V \cup \{s\}, E)$ with $V = [-n, n]^2 \cap \mathbb{Z}^2$ for a given n implemented in Python can be found in B.1.

In the paper of Creutz, only finite lattice graphs as discussed. In this thesis, we observed we can generalize this method to any finite graph $G = (V \cup \{s\}, E)$ by taking

$$I_0(v) = \mathfrak{m}(v, s),$$

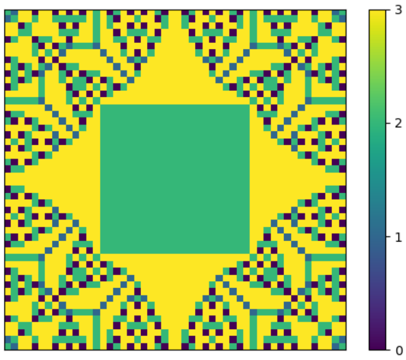
for all $v \in V$ where $\mathfrak{m}(i, j)$ denotes the edge multiplicity between i and j, which is the amount of edges between i and j. This is not a recurrent configuration, but adding I_0 to a recurrent configuration and stabilizing acts neutral, i.e. for every recurrent configuration $\eta \in \mathcal{R}$, we have

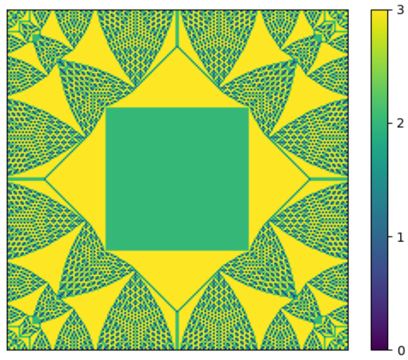
$$S(\eta + I_0) = \eta.$$

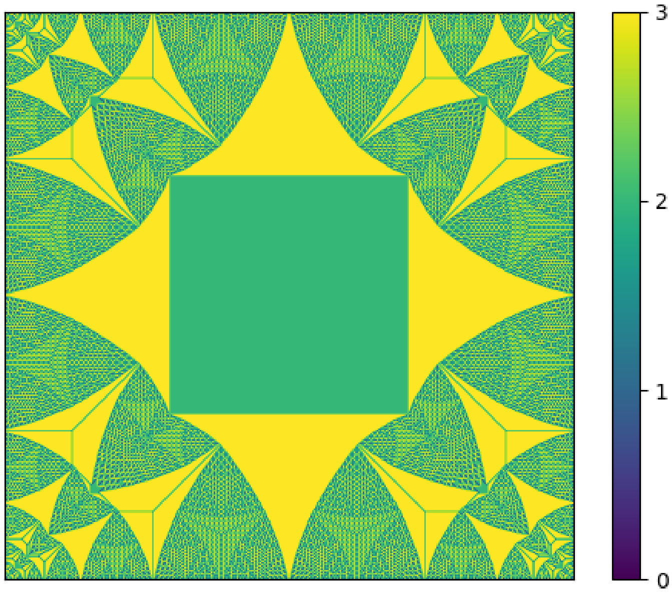
Using the iteration step $I_n = I_{n-1} \oplus I_{n-1}$, we will again after a finite amount of steps end up with the recurrent element in $(\mathcal{R}, \oplus)$ that acts as the identity.

3.3 Results

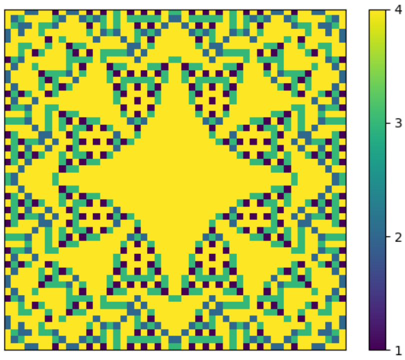
The neutral elements we found with both algorithms are identical, as we would expect. We get the following visualisations of the identity elements on the lattice graph $G = (V \cup \{s\}, E)$ with $V = [-n, n]^2 \cap \mathbb{Z}^2$ for $n = 25$, $n = 100$ and $n = 250$ respectively.

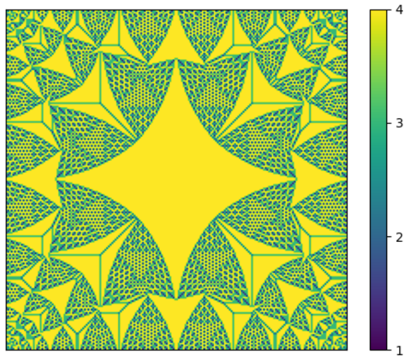


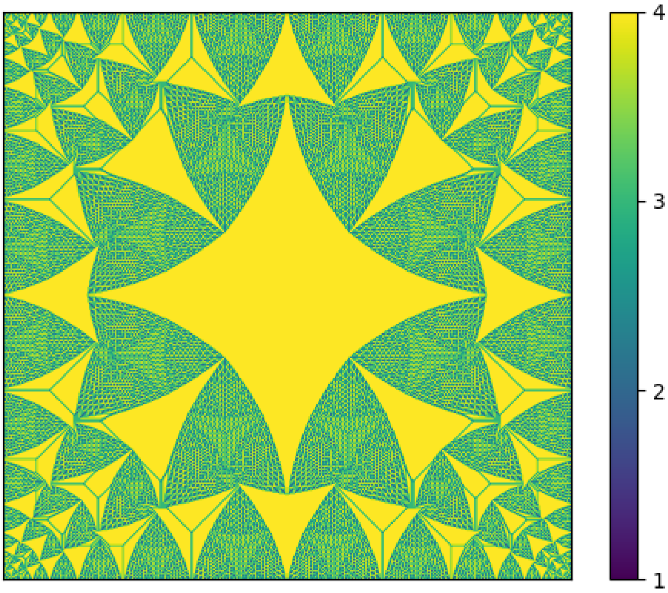




As has been mentioned previously, some literature considers every vertex of the graph in the sandpile model to start at height one and to topple when its height is greater than the degree of the vertex. The effect is that the recurrent elements of the Markov chain are different. We can make a slight adjustment to the code to calculate the neutral element of $(\mathcal{R}, \oplus)$ in this case. We get the following visualisations of the identity elements on the lattice graph $G = (V \cup \{s\}, E)$ with $V = [-n, n]^2 \cap \mathbb{Z}^2$ for $n = 25$, $n = 100$ and $n = 250$ respectively.







3.3.1 Starting at 0 or 1?

At first, one might expect that if we take the identity element of the model where all the heights are in between zero and the degree of a vertex minus one and we heighten every site with one, we get the identity element of the model where all the heights are in between one and the degree of a vertex. This would result in their visualisations looking the same, only shifted up by one, but this clearly is not the case if we look at the visualisations above. Why does this behaviour show?

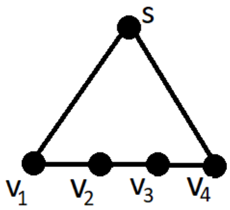
A key observation is that when we add one particle to every site of the recurrent configuration η_1 with its heights between zero and the degree minus one that acts neutral and we map this newly obtained configuration to divisor in the Jacobian using f from Theorem 2.5, we get that this divisor is in the equivalence class of $D_{\mathbb{1}}$, where

$$D_{\mathbb{1}} = \sum_{v \in V} 1 \cdot v - \left(\sum_{v \in V} 1 \right) \cdot s.$$

When $D_{\mathbb{1}}$ is not equivalent to the identity element in the Jacobian, we know that if we add one particle to every site of η_1, we get a configuration that does not act as the identity in the set of recurrent configurations of which the heights are in between 1 and the degree of a vertex. In that case we get that the identity elements of the two versions of the model look different. There are graphs for which $D_{\mathbb{1}}$ is equivalent to the identity and thus the two representatives of the identity in recurrent elements looks the same. To the author, no precise classification of graphs for which the identity elements do or do not look the same is known. For some very alike graphs, this property can differ. Take for instance a circle graph $G = (V \cup \{s\}, E)$. If $|V|$ is even, the identity elements look the same, whilst if $|V|$ odd, they do not.

Let us look at an example of a graph for which the identity elements look the same.

Example 3.1. *Consider the graph $G = (V \cup \{s\}, E)$:*



The recurrent configuration η_1 *that acts as the identity when the height of every vertex starts at 0 and we topple a vertex* v *when* $\eta(v) \geq \deg_G(v)$ *is*

$$\eta_1(v) := 1,$$

for all $v \in V$*. The recurrent configuration* η_2 *that acts as the identity when every vertex starts at 1 and we topple a vertex* v *when* $\eta(v) > \deg_G(v)$ *is*

$$\eta_2(v) := 2,$$

for all $v \in V$*. They are both the identity, so we see that*

$$f(\eta_1) = [v_1 + v_2 + v_3 + v_4 - 4s] = [0] = [2v_1 + 2v_2 + 2v_3 + 2v_4 - 8s] = f(\eta_2).$$

We see that if we add 1 particle to every site of η_1*, we get* η_2*. The fact that adding 1 particle to every site of* η_1 *results in the identity element of the model where the heights are between 1 and the degree of a vertex is because:*

$$f(\eta_1 \oplus \eta_{\mathbb{1}}) = f(\eta_1) \boxplus f(\eta_{\mathbb{1}}) = [v_1 + v_2 + v_3 + v_4 - 4s] \boxplus [D_{\mathbb{1}}] = [0] \boxplus [0] = [0]s,$$

where $\eta_{\mathbb{1}}$ *is the height 1 configuration*

$$\eta_{\mathbb{1}}(v) := 1,$$

for all $v \in V$*. As for this graph* G *we have that* $D_{\mathbb{1}}$ *is equivalent to* $D_0 = 0$*, we have that* $\eta_1 \oplus \eta_{\mathbb{1}}$ *is the recurrent element that acts as the identity when considering the recurrent configurations that start at 1 and we topple a vertex* v *when* $\eta(v) > \deg_G(v)$*, i.e.* $\eta_1 \oplus \eta_{\mathbb{1}} = \eta_2$*.* $\Diamond$

Chapter 4

Torelli theorems on graphs

As the definition of the Jacobian on a graph is inspired by algebraic geometry, a part of research on the Abelian sandpile model has been dedicated to translating theorems from algebraic geometry to graphs. In this chapter, we will look into the Torelli theorem.

In algebraic geometry, the well-known Torelli theorem states that

Theorem 4.1. *[Mil86, Cor. 12.2] Let C and C' be two curves over a perfect field k of genus at least two. If the canonically polarized Jacobian varieties of C and C' are k-isomorphic, then C and C' are isomorphic over k as well.*

In 2009 Caporaso and Viviani [CV10] and Su and Wagner [SW10] made the first statements that resemble this Torelli theorem. In 2023, Griffith published a new Torelli theorem for graphs [Gri23]. In this chapter we will focus on the result of Griffith, which requires the Jacobians to admit a discrete theta divisor isomorphism for the graphs to be isomorphic. We will go through some of the main concepts discussed in the paper. In the next chapter, we will find previously unknown properties of special and non-special divisors (which can be found using the discrete theta divisor), namely its rank and bounds on its q-weight. In the next chapter we will also find an isomorphism between non-special divisors and minimal configurations, which will enable us to translate this theorem to the probabilistic setting.

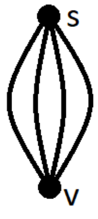
4.1 First attempt at a Torelli theorem for graphs

Inspired on the Torelli theorem on curves from Theorem 4.1, we would like to prove a similar theorem on graphs. A first attempt could be the following statement.

Statement 4.1. *Let G and H be two graphs. Then G is isomorphic to H if and only if their Jacobians are isomorphic.*

In the following example we will see that this statement does not hold.

Example 4.1. *Consider the following graph G:*



All divisors of degree 0 of this graph are of the form

$$D = d \cdot v - d \cdot s,$$

for some $d \in \mathbb{Z}$. As there are 4 edges between s and v, we see that the principal divisors are of the form

$$D_f = 4f(v) \cdot v - 4f(v) \cdot s,$$

for some $f(v) \in \mathbb{Z}$. Remember that the Jacobian $Jac(G)$ are classes of divisors of degree 0 up to equivalence, where two divisors are equivalent if their difference is a principal divisor. Therefore we see that

$$Jac(G) = \{[0], [v - s], [2v - 2s], [3v - 3s]\}.$$

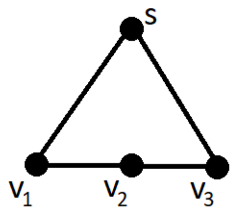
Clearly, $[0]$ is the identity element of $Jac(G)$. We see that $[v - s]$ is an element of order 4, as

$$\begin{aligned}
&[v - s] \boxplus [v - s] = [2v - 2s], \\
&[v - s] \boxplus [v - s] \boxplus [v - s] = [3v - 3s], \\
&[v - s] \boxplus [v - s] \boxplus [v - s] \boxplus [v - s] = [4v - 4s] = [0],
\end{aligned}$$

where $\boxplus$ is the group action on $Jac(G)$ as defined in Lemma 2.3. We conclude that

$$(Jac(G), \boxplus) \cong \mathbb{Z}_4.$$

Now consider the graph G':



In Example 1.11 we saw that

$$(\mathcal{R}, \oplus) \cong \mathbb{Z}_4.$$

In Theorem 2.5 we saw that the recurrent sandpile group $(\mathcal{R}, \oplus)$ is isomorphic with $(Jac(G'), \boxplus)$, so we have that:

$$(Jac(G'), \boxplus) \cong \mathbb{Z}_4.$$

The Jacobians of the graphs G and G' are isomorphic, whilst the graphs are not isomorphic, as they don't even have the same number of vertices. ♢

Later in this chapter, we will see that if the Jacobians of two graphs admit a special kind of isomorphism, namely a discrete theta divisor preserving isomorphism, we get that the graphs are isomorphic. When we define the discrete theta divisor, we will show that for these two example graphs above, the discrete theta divisor is different. The example shows that we really need this discrete theta divisor, as a naive version of the Torelli theorem as stated in Statement 4.1 does not hold.

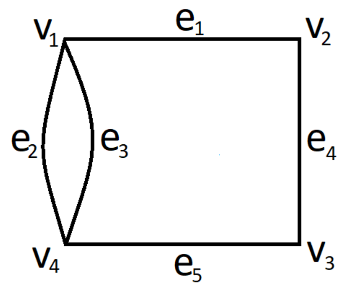
4.2 Graphic matroid

As a naive version of a Torelli theorem for graphs does not hold, we have to put some restrictions on either the graphs we look at or the kind of isomorphism the Jacobians have to admit. One of the first Torelli theorems on graphs uses *graphic matroids.*

Definition. *Let $G = (V, E)$ be a finite, undirected graph. The associated graphic matroid $M(G)$ is the edge set E of G together with a set of simple unordered cycles.*

An *isomorphism between graphic matroids* is a bijection between the edge sets that induces a bijection between the sets of simple unordered cycles. Note that an isomorphism between graphic matroids is only possible if G and H have the same amount of edges. Let us look at an example of a graphic matroid.

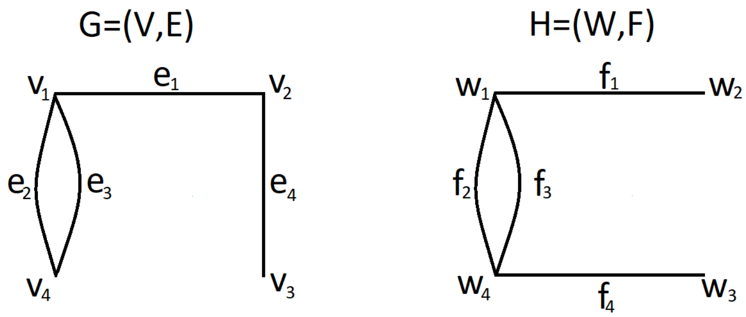
Example 4.2. *Consider the graph $G = (V, E)$:*



The graphic matroid associated with this graph is the edge set $E = \{e_1, e_2, e_3, e_4, e_5\}$ and the set of simple unordered cycles, which is

$$\{\{e_1, e_2, e_4, e_5\}, \{e_1, e_3, e_4, e_5\}, \{e_2, e_3\}\}.$$

Now let us look at a morphism between graphic matroids. Consider the two graphs $G = (V, E)$ and $H = (W, F)$



The graphic matroid of G is E together with the set $\{\{e_2, e_3\}\}$ and the graphic matroid of H is F together with the set $\{f_2, f_3\}$. We see that the graphic matroids are isomorphic, as

$$\phi : M(G) \to M(H), \quad \phi(e_i) = f_i \text{ for all } i \in \{1, 2, 3, 4, 5\},$$

is an isomorphism between the graphic matroids, as it preserves the simple cycles. However, we can see that G and H are not isomorphic graphs, as for instance vertex v_2 in G is of degree 2 and H does not have a vertex of degree 2.

We see that two graphs do not have to be isomorphic to have isomorphic graphic matroids. ◊

The first successful attempts at a Torelli theorem on graphs are by Caporaso and Viviani [CV10] and by Su and Wagner [SW10]. They proved a Torelli theorem for 3-edge connected graphs. Also, they found a weaker result that states that for 2-edge connected graphs, the graphic matroids are isomorphic if and only if the Jacobians of two graphs are isomorphic .

This second theorem was originally proved for the Albaneses of the graphs being isomorphic if and only if the matroids are isomorphic. The Albanese is the dual of the Jacobian, when we define the Jacobian in a more geometric setting. The Albeneses of two graphs are isomorphic if any only if the Jacobians of two graphs are isomorphic. Therefore, we will state the theorem here in terms of the Jacobian.

Theorem 4.2. *[CV10, Thm. 3.1.1] [Gri23, Thm 3.9] Let G and H be 2-edge connected graphs. Then $M(G) \cong M(H)$ if and only if $Jac(G) \cong Jac(H)$.*

For a proof, we refer the reader to Section 3 of Caporaso and Viviani [CV10].

We will look into a Torelli theorem on graphs by Griffith [Gri23], which roughly states:

Theorem 4.3. *Let G and H be 2-edge connected graphs of genus at least two. Then their Jacobians admit a discrete theta divisor preserving isomorphism if and only if G and H are isomorphic graphs.*

A more detailed version of this theorem can be found in Theorem 4.9, which is a reformulation of Theorem 6.6 in Griffith [Gri23]. In the rest of this chapter, we will go through some of the theory in the paper give an idea of the ingredients needed to prove the detailed version of this theorem.

4.3 Discrete theta divisor

In this section we will introduce the discrete theta divisor, which is a crucial object in the Torelli theorem by Griffith. Although its name suggests this is a divisor, it is actually a set of divisors in the Jacobian. To define the set, we need a set of maps called the *Abel-Jacobi* maps.

Definition. *Let $G = (V, E)$ be a connected graph. Let $e \in E$ be some base edge that has an orientation and let $t(e) = v_0 \in V$. Let $n \in \mathbb{Z}_{\geq 0}$. The Abel-Jacobi maps $S^n_{v_0}$ are given by:*

$$S^n_{v_0} : Div^+_n(G) \to Jac(G), \quad v_{i_1} + ... + v_{i_n} \mapsto [v_{i_1} + ... + v_{i_n} - n \cdot v_0],$$

where $Div^+_n(G)$ is the set of effective divisors of degree n.

These Abel-Jacobi maps are mapping all effective divisors of degree n to the equivalence class in the Jacobian. To make it degree 0, we subtract the target of the base edge. This is very similar to the natural mapping $\phi_{0,n,s} \circ f$ from $\text{Pic}_n(G)$ to $\mathcal{R}$ we saw in Corollary 2.6, where we would subtract n times the sink s. The base edge can be any edge, it is just an extra marking to some edge, alike the marked vertex s in the probabilistic approach from Chapter 1.

Using these Abel-Jacobi maps, we can define the discrete theta divisor. Remember that the genus of a graph G is defined as $g(G) = |E| - |V| + 1$.

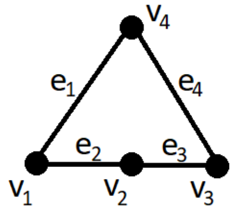
Definition. *The discrete theta divisor θ_e of a graph G for some oriented base edge e is the image*

$$\theta_e := Im(S^{g(G)-1}_{v_0}) \subseteq Jac(G),$$

where $v_0 = t(e)$.

Let us look at an example of the Abel-Jacobi maps and the discrete theta divisor.

Example 4.3. *Let us consider the graph $G = (V, E)$:*



Let e_1 be the base edge, and define $t(e_1) = v_4$.

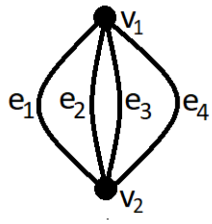
The genus of this graph is $g(G) = 4 - 4 + 1 = 1$. To find the discrete theta divisor we therefore want to look at the Abel-Jacobi map $S^0_{v_4}$.

We see that that this Abel-Jacobi map sends the only element of $Div^+_0(G) = \{0\}$ to the zero class in the Jacobian:

$$S^0_{v_4} : Div^+_0(G) \to Jac(G), \quad 0 \mapsto [0].$$

Therefore we see that in this case, the discrete theta divisor equals $\{[0]\} \subset Jac(G)$.

Let us now consider the graph $G' = (V', E')$ given by:



Let e_1 be the base edge, with $t(e_1) = v_1$.

The genus of this graph is $g(G) = 4 - 2 + 1 = 3$. To find the discrete theta divisor we therefore want to look at the Abel-Jacobi map $S^2_{v_1}$.

We see that that this Abel-Jacobi map is:

$$S^2_{v_1} : Div^+_2(G) \to Jac(G), \quad v_i + v_j \mapsto [v_i + v_j - 2v_1].$$

We have that $Div^+_2(G) = \{2v_1, v_1 + v_2, 2v_2\}$. We see that

$$S^2_{v_1}(2v_1) = [0], \quad S^2_{v_1}(v_1 + v_2) = [v_2 - v_1], \quad S^2_{v_1}(2v_2) = [2v_2 - 2v_1].$$

The discrete theta divisor equals $\{[0], [v_2 - v_1], [2v_2 - 2v_1]\} \subset Jac(G)$. We see that these two non-isomorphic graphs have a different discrete theta divisor whilst their Jacobians are isomorphic, which we saw in Example 4.1. ◊

If we want to get rid of the base edge, we can look at the set of *special divisors.*

Definition. *We define the set of special divisor classes to be*

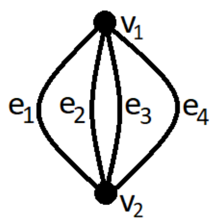
$$\mathcal{S}(G) := \{D + (g(G) - 1)v_0 | [D] \in \theta_e\} \subseteq Pic_{g-1}(G).$$

The set of non-special divisor classes $\mathcal{N}(G)$ are those that are not special, i.e.

$$\mathcal{N}(G) := Pic_{g-1}(G) \setminus \mathcal{S}(G).$$

A divisor class $[D]$ is called *special* if $[D] \in \mathcal{S}(G)$. It is called *non-special* if it is not special, i.e. when $[D] \in \mathcal{N}(G)$. Let us look at an example of (non-)special divisor classes.

Example 4.4. *Consider the graph $G = (V, E)$:*



Let e_1 be the base edge, with $t(e_1) = v_1$.

In Example 4.3 we saw that the discrete theta divisor for this graph is $\{[0], [v_2 - v_1], [2v_2 - 2v_1\} \subset Jac(G)$. Therefore we see that the special divisor classes are

$$\mathcal{S}(G) = \{[2v_1], [v_1 + v_2], [2v_2]\} \subset Pic_2(G).$$

There is one divisor class in $Pic_2(G)$ that is not a special divisor. We see that

$$\mathcal{N}(G) = \{[3v_1 - v_2]\} \subset Pic_2(G).$$

Note that as the genus of this graph is 3, the special and non-special divisor classes live in $Pic_2(G)$. ♢

4.4 Chern class map

In this section we will characterize the special and non-special divisors by mapping a (partial) orientation on a graph G to a divisor on G. We will make use of the Chern class map, which we will now define.

Definition. *[Gri23, Def. 3.10] Let $G = (V, E)$ be a graph and $k \in \mathbb{Z}$ such that $-|V| < k \leq g-1$. Let $\mathbb{O}_k(G)$ be the set of partial orientations on $k + |V|$ edges. If $k = g - 1$, this is the set of full orientations (as $k + |V| = g - 1 + |V| = |E|$) and we will denote this set by*

$$\mathbb{O}(G) := \mathbb{O}_{g-1}(G).$$

The Chern class map c is defined as the following:

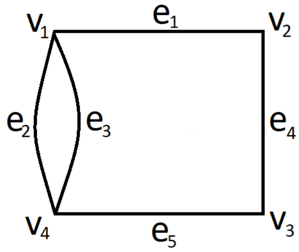
$$c : \mathbb{O}_k(G) \to Pic_k(G), \quad U \mapsto \left[\sum_{e \in E} t_U(e) - \sum_{v \in V} v\right],$$

where $t_U(e)$ is the target function in the orientation $U = (o_U, t_U)$. We call $c(U)$ the Chern class of U.

Note that k can be thought of as some tuning parameter with which we can define the Chern class map to a specific Picard set, namely to $\text{Pic}_k(G)$.

Let us look at an example of the Chern class map.

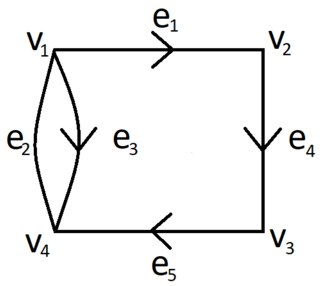
Example 4.5. *Let us consider the following graph $G = (V, E)$:*



If we take $k = -4 = -|V|$, we have that we do not orient any edges and we get the partial orientation U_1 as above. One can argue whether this should be called an orientation as there are no oriented edges, but in this thesis we call this a partial orientation. We see that this partial orientation U_1 without any oriented edges gives us the Chern map:

$$c : \mathbb{O}_{-4}(G) \to Pic_{-4}(G), \quad U_1 \mapsto [-v_1 - v_2 - v_3 - v_4].$$

Now let $k = 0$. This means that we have to orient 4 edges of G to obtain a partial orientation in $\mathbb{O}_0(G)$. Consider the partial orientation U_2:



This orientation gets mapped by the Chern map to

$$c(U_2) = \left[\sum_{e \in E} t(e) - \sum_{v \in V} v\right] = [v_2 + v_4 + v_3 + v_4 - v_1 - v_2 - v_3 - v_4] = [v_4 - v_1] \in Pic_0(G).$$

$\Diamond$

Using the Chern class map, we call two orientations in $\mathbb{O}_k$ equivalent if they map to the same divisor class. With this equivalence, we create equivalence classes of (partial) orientations.

In [Gri23, Rmk. after Def. 3.16] it is mentioned that that an orientation class is in the preimage of $\mathcal{S}(G)$ if it contains a sourceless orientation, while an orientation class is in the preimage of $\mathcal{N}(G)$ when it is an acyclic orientation.

Theorem 4.4. *The Chern class map c induces a bijection between the non-special divisor classes and acyclic orientations with a unique source of G.*

For a proof, we refer to [Bac17, Thm. 4.10].

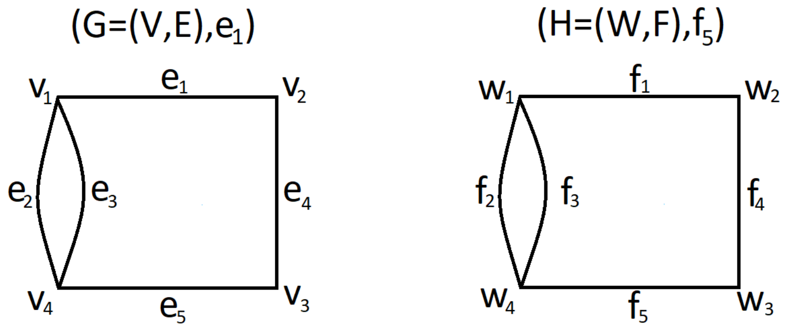
4.5 Maps on orientations

As we have proven that we can characterize the special and non-special divisors with orientations of a graph being acyclic or not, we start to focus on maps on oriented graphs that preserve acyclic orientations in some way, which we will explore in this section. Later on we will use the maps we define in this section to differentiate between theta divisor preserving and non theta divisor preserving isomorphisms of the Jacobian. We will start with defining a base preserving cyclic bijection.

Definition. *[Gri23, Def. 3.19] Let (G, e_G) and (H, e_H) be two graphs with a base edge. A base preserving cyclic bijection between (G, Γ, e_G) and (H, Γ', e_H) is a bijection from the edges of G to the edges of H that preserves simple unordered cycles and maps e_G to e_H.*

A base preserving cyclic bijection is a bijection on the edges of the graphs that induces an isomorphism between the graphic matroids of the graph. Note that a cyclic bijection is only possible if G and H have the same amount of edges. Let us look at an example of a base preserving cyclic bijection.

Example 4.6. *Let us consider the following two based graphs $(G = (V, E), e_1)$ and $(H = (W, F), f_5)$:*



An example of a base preserving cyclic bijection ϕ *is*

$$\phi(e_1) = f_5, \quad \phi(e_2) = f_3, \quad \phi(e_3) = f_2, \quad \phi(e_4) = f_4, \quad \phi(e_5) = f_1.$$

Note that ϕ *maps the base edge* e_1 *of* G *to the base edge* f_5 *of* H*. All simple unordered cycles of* G *are:*

$$C_1 = \{e_1, e_2, e_4, e_5\}, \quad C_2 = \{e_1, e_3, e_4, e_5\}, \quad C_3 = \{e_2, e_3\}.$$

We see that

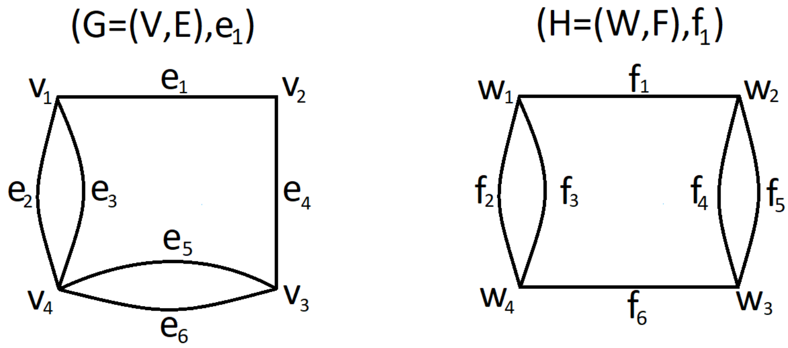
$$\phi(C_1) = \{f_5, f_3, f_4, f_1\}, \quad \phi(C_2) = \{f_5, f_2, f_4, f_1\}, \quad C_3 = \{f_3, f_2\},$$

are also all simple unordered cycles. We conclude that ϕ *is a base preserving cyclic bijection.*

Note that ϕ *induces a graph isomorphism from* G *to* H*, i.e. there exists a mapping* $\phi_V : V \to W$ *such that for all* $v \in V$

$$v \in e \in E \Rightarrow \phi_V(v) \in \phi(e) \in F.$$

It is also possible to have a base preserving cyclic bijection that does not induce such a graph isomorphism. An example, that can also be found in [Gri23, Ex.3.20], is on the graphs $(G = (V, E), e_1)$ *and* $(H = (W, F), f_1)$*:*



The mapping ϕ*:*

$$\phi(e_1) = f_1, \quad \phi(e_2) = f_2, \quad \phi(e_3) = f_3, \quad \phi(e_4) = f_6, \quad \phi(e_5) = f_4, \quad \phi(e_6) = f_5,$$

maps the base edge e_1 *of* G *to the base edge* f_1 *of* H *and preserves simple unordered edges. Therefore* ϕ *is a base preserving cyclic bijection. However,* ϕ *could never induce a graph isomorphism between* G *and* H*, as* v_4 *has degree* 4 *in* G *but there is no vertex of degree* 4 *in* H*.* $\Diamond$

If we talk about a base preserving cyclic bijection ϕ on two oriented graphs (G, Γ, e_G) and (H, Γ', e_H), ϕ is still a bijection from G to H that preserves simple unordered cycles and maps e_G to e_H, so ϕ sees nothing of the orientation on the graphs. However, we can find a unique sign function $\text{sgn}_\phi : E \to \{-1, 1\}$ that preserves the orientation on all cycles. The following theorem is a reformulation of Theorem 4.3 of Griffith, which states it in broader category theory terms [Gri23, Thm. 4.3].

Theorem 4.5. *Let $\phi : (G, \Gamma, e_G) \to (H, \Gamma', e_H)$ be a cyclic bijection on two 2-edge connected based oriented graphs. For an unoriented cycle $c = \{e_1, ..., e_{c_n}\}$, let or_{c,e_i} such that*

$$or_{c,e_1} e_1, or_{c,e_2} e_2, ..., or_{c,e_{c_n}} e_{c_n},$$

is an oriented cycle where $or_{e_i} \in \{-1, 1\}$, where or_{e_i} denotes reversing the orientation of edge e_i such that this path indeed is an oriented cycle. Note that there exist two such or. It does not matter which one of the two is chosen.

Then there exists a unique $sgn_\phi : E_G \to \{-1, 1\}$ such that $sgn_\phi(e_G) = 1$ and for every unoriented cycle $\{e_1, ..., e_c\}$ we have that

$$sgn_\phi(e_1) or_{c,e_1} \phi(e_1), sgn_\phi(e_2) or_{c,e_2} \phi(e_2), ..., sgn_\phi(e_{c_n}) or_{c,e_{c_n}} \phi(e_{c_n}),$$

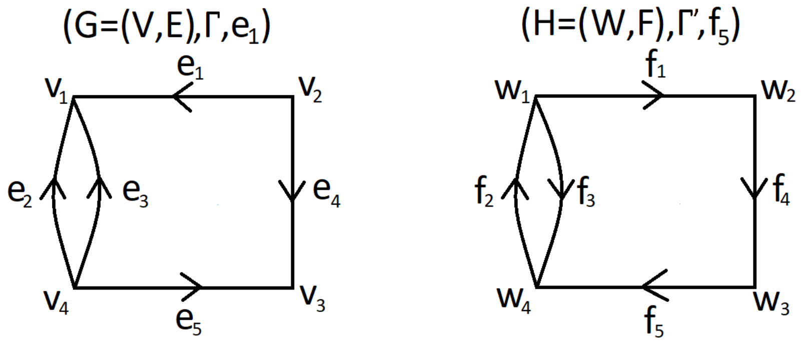
is an oriented cycle in H. This extends to a isomorphism $\phi_* : Jac(G) \to Jac(H)$.

For a proof of the theorem, see [Gri23, Thm. 4.3]. The intuition to this extending to an isomorphism, is as it maps unoriented cycles to unoriented cycles, we have that the graphic matroids are isomorphic. In Theorem 4.2 we saw that isomorphic graphic matroids imply isomorphic Jacobians. In Example 4.9 we will explore how to calculate this mapping.

If $\text{sgn}_\phi(e) = 1$ we say that ϕ is *orientation preserving* for e. If $\text{sgn}_\phi(e) = -1$ we say that ϕ is *orientation reversing* for e.

We will now give an example of finding this unique sgn_ϕ.

Example 4.7. *Let us again consider the based oriented 2-connected graphs $(G = (V, E), \Gamma, e_1)$ and $(H = (W, F), \Gamma', f_5)$:*



In Example 4.6 we saw that ϕ defined by

$$\phi(e_1) = f_5, \quad \phi(e_2) = f_3, \quad \phi(e_3) = f_2, \quad \phi(e_4) = f_4, \quad \phi(e_5) = f_1,$$

is a base preserving cyclic bijection. We saw that the unordered cycles in G are:

$$C_1 = \{e_1, e_2, e_4, e_5\}, \quad C_2 = \{e_1, e_3, e_4, e_5\}, \quad C_3 = \{e_2, e_3\}.$$

Let us first consider the unordered cycle C_1. We see that

$$e_1, -e_2, e_5, -e_4$$

is the oriented cycle in G. Therefore we have for this cycle that

$$or_{C_1,e_1} = 1, \quad or_{C_1,e_2} = -1, \quad or_{C_1,e_5} = 1, \quad or_{C_1,e_4} = -1.$$

Note that we also have chosen to have the reverse orientation of the cycle, which would lead to the values of the or function to be the opposite of the one we chose now.

In Example 4.6 we saw that the unordered cycle C_1 is mapped by ϕ to $\{f_5, f_3, f_4, f_1\}$. By definition of the sgn_ϕ function, we want for every cycle C and every $e \in C$ that

$$sgn_\phi(e) \cdot or_{C,e} = or_{\phi(C),\phi(e)},$$

where by definition, we have sgn_ϕ evaluates to 1 on the base edge. Therefore we have $sgn_\phi(e_1) = 1$. We get that

$$or_{\phi(C_1),\phi(e_1)} = or_{\phi(C_1),f_5} = sgn_\phi(e_1) or_{C_1,e_1} = 1.$$

We orient $\phi(C_1)$ such that $or_{\phi(C_1),f_5} = 1$ and obtain

$$f_5, -f_3, f_1, f_4,$$

as an oriented cycle in H. For this cycle, we get

$$or_{\phi(C_1),f_1} = 1, \quad or_{\phi(C_1),f_4} = 1, \quad or_{\phi(C_1),f_5} = 1, \quad or_{\phi(C_1),f_3} = -1.$$

To determine $sgn_\phi(e_4)$, we want that to have

$$sgn_\phi(e_4) or_{C_1,e_4} = or_{\phi(C_1),\phi(e_4)},$$

i.e.

$$sgn_\phi(e_4) \cdot -1 = or_{\phi(C_1),f_4} = 1.$$

We obtain that $sgn_\phi(e_4) = -1$. The same way, we can obtain that $sgn_\phi(e_5) = 1$ and $sgn_\phi(e_2) = 1$.

The only value left to determine is $sgn_\phi(e_3)$. For this purpose, let us take a look at unordered cycle $C_2 = \{e_1, e_3, e_4, e_5\}$. Under ϕ this is mapped to $\phi(C_2) = \{f_5, f_2, f_4, f_1\}$ as we saw in Example 4.6. Let us choose an orientation $or_{C_2,e_1} = 1$. We see that

$$e_1, -e_3, e_5, e_4,$$

is an oriented path in G, i.e. we get

$$or_{C_2,e_1} = 1, \quad or_{C_2,e_3} = -1, \quad or_{C_2,e_5} = 1, \quad or_{C_2,e_4} = -1.$$

We already know that $sgn_\phi(e_1) = 1$. Therefore we can determine the way orientation we want to walk the path $\phi(C_2)$, as

$$sgn_\phi(e_1) or_{C_2,e_1} = or_{\phi(C_2),\phi(e_1)},$$

i.e.

$$or_{\phi(C_2),\phi(e_1)} = or_{\phi(C_2),f_5} = 1 \cdot 1 = 1,$$

so we get $or_{\phi(C_2),f_5} = 1$. We get

$$f_5, f_2, f_1, -f_4,$$

is an oriented cycle in H. Therefore we have for this cycle that

$$or_{\phi(C_2),f_1} = 1, \quad or_{\phi(C_2),f_4} = -1, \quad or_{\phi(C_2),f_5} = 1, \quad or_{\phi(C_2),f_2} = 1.$$

We can now calculate $sgn_\phi(e_3)$, by considering the equation

$$sgn_\phi(e_3) or_{C_2,e_3} = or_{\phi(C_2),\phi(e_3)}.$$

We get that

$$sgn_\phi(e_3)\cdot -1 = or_{\phi(C_2),f_2} = 1.$$

It follows that $sgn_\phi(e_3) = -1$. If we fill in these equations for e_4 and e_5 with the already found values for $sgn_\phi(e_4)$ and $sgn_\phi(e_5)$, we see that they hold.

Finally, let us check that with C_3 and $\phi(C_3)$, the equations hold as well (which they should by the theorem). We see that

$$e_2 - e_3,$$

is an oriented cycle in G. We have that $or_{C_3,e_2} = 1$ and $or_{C_2,e_3} = -1$.

We can determine the orientation of the cycle $\phi(C_3)$ by calculating

$$or_{\phi(C_3),f_3} = or_{\phi(C_3),\phi(e_2)} = sgn_\phi(e_2) or_{C_3,e_2} = 1\cdot 1 = 1.$$

We get the oriented cycle $\phi(C_3)$ is

$$f_3 + f_2.$$

To check, we want the equation

$$or_{\phi(C_3),\phi(e_3)} = sgn_\phi(e_3) or_{C_3,e_3},$$

to hold. We see that

$$1 = or_{\phi(C_3),\phi(e_3)} = sgn_\phi(e_3) or_{C_3,e_3} = -1\cdot -1.$$

We conclude that there indeed is a sgn_ϕ function that fulfills the conditions of Theorem 4.5. We have that ϕ is orientation preserving for edges e_1, e_2 and e_5 and orientation reversing for edges e_3 and e_4. ♢

With this sign function, we can define a function that maps orientations of G to orientations on H using a specific manner based on a base preserving cyclic bijection.

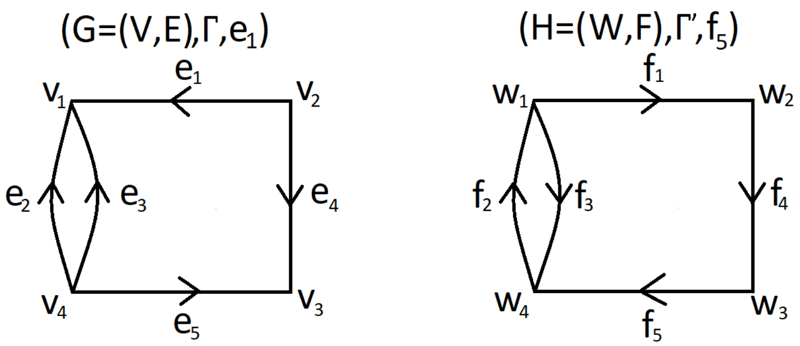
Definition. *Let $(G = (V_G, E_G), \Gamma, e)$ and $(H = (V_H, E_H), \Gamma', w)$ be two 2-edge connected oriented graphs with a base edge. Let $\phi : (G, \Gamma, e) \to (H, \Gamma', w)$ be a base preserving cyclic bijection. Given an orientation $U \in \mathbb{O}(G)$ and an edge $e \in E_G$, let $sgn_U(e)$ equal 1 if the orientation of e in U is the same as in Γ and 0 if it is not. We define a morphism $\phi_{\mathbb{O}} : \mathbb{O}(G) \to \mathbb{O}(H)$ be such that it sends the orientation $U \in \mathbb{O}(G)$ such that*

$$sgn_{\phi_{\mathbb{O}}(U)}(\phi(e)) = sgn_\phi(e) sgn_U(e),$$

for all $e \in E_G$. We extend $\phi_{\mathbb{O}}$ to partial orientations U by setting $sgn_U(e) = 0$ when e is unoriented.

Let us continue the previous example and determine $\phi_{\mathbb{O}}(U)$ for some orientation U.

Example 4.8. *Let us again consider the based oriented 2-connected graphs $(G = (V, E), \Gamma, e_1)$ and $(H = (W, F), \Gamma', f_5)$:*



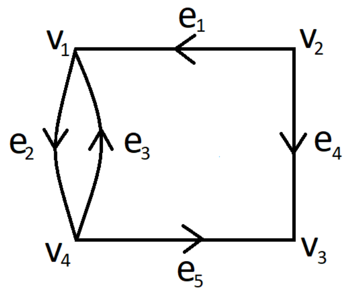
In Example 4.6 we saw that ϕ defined by

$$\phi(e_1) = f_5, \quad \phi(e_2) = f_3, \quad \phi(e_3) = f_2, \quad \phi(e_4) = f_4, \quad \phi(e_5) = f_1,$$

is a cyclic bijection. In Example 4.7 we saw that ϕ induced the sign function sgn_ϕ with

$$sgn_\phi(e_1) = 1, \quad sgn_\phi(e_2) = -1, \quad sgn_\phi(e_3) = 1, \quad sgn_\phi(e_4) = 1, \quad sgn_\phi(e_5) = -1.$$

Now let us consider the orientation U_1 on G:



We see that the orientation of the edges e_1, e_3, e_4 and e_5 in U_1 is the same as in (G, Γ, e_1) and the orientation of the edge e_2 in U_1 differs from the orientation in (G, Γ, e_1). Therefore we have

$$sgn_{U_1}(e_1) = 1, \quad sgn_{U_1}(e_2) = -1, \quad sgn_{U_1}(e_3) = 1, \quad sgn_{U_1}(e_4) = 1, \quad sgn_{U_1}(e_5) = 1.$$

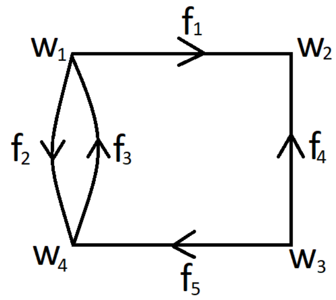
Combining these values with sgn_ϕ, we obtain

$$sgn_{\phi_{\mathbb{O}}(U_1)}(\phi(e_1)) = sgn_\phi(e_1) sgn_{U_1}(e_1) = 1 \cdot 1 = 1, \quad sgn_{\phi_{\mathbb{O}}(U_1)}(\phi(e_2)) = 1 \cdot -1 = -1,$$
$$sgn_{\phi_{\mathbb{O}}(U_1)}(\phi(e_3)) = -1 \cdot 1 = -1, \quad sgn_{\phi_{\mathbb{O}}(U_1)}(\phi(e_4)) = -1 \cdot 1 = -1, \quad sgn_{\phi_{\mathbb{O}}(U_1)}(\phi(e_5)) = 1 \cdot 1 = 1.$$

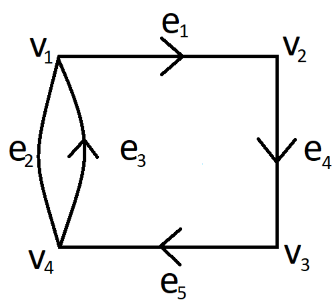
Therefore we see that

$$sgn_{\phi_{\mathbb{O}}(U_1)}(f_5) = 1, \quad sgn_{\phi_{\mathbb{O}}(U_1)}(f_3) = -1, \quad sgn_{\phi_{\mathbb{O}}(U_1)}(f_2) = -1,$$
$$sgn_{\phi_{\mathbb{O}}(U_1)}(f_4) = -1, \quad sgn_{\phi_{\mathbb{O}}(U_1)}(f_1) = 1.$$

This means that in the image $\phi_{\mathbb{O}}(U_1)$, the orientation of the edges f_2, f_3 and f_4 have to be reversed in comparison to orientation (H, Γ'). We see that $\phi_{\mathbb{O}}(U_1)$ is:



Now let us also look at how $\phi_{\mathbb{O}}$ maps partial orientations. Consider the following partial orientation U_2 on G:



As edge e_2 is not oriented, we have $sgn_{U_2}(e_2) = 0$. We see that the orientation of the edges e_3 and e_4 in U_2 is the same as in (G, Γ, e_1) and the orientation of the edges e_1 and e_5 in U_2 differs from the orientation in (G, Γ, e_1). In total we have

$$sgn_{U_2}(e_1) = -1, \quad sgn_{U_2}(e_2) = 0, \quad sgn_{U_2}(e_3) = 1, \quad sgn_{U_2}(e_4) = 1, \quad sgn_{U_2}(e_5) = -1.$$

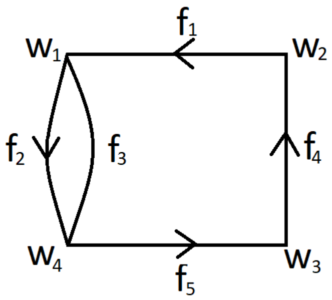
Combining these values with sgn_ϕ, we obtain

$$sgn_{\phi_{\mathbb{O}}(U_2)}(\phi(e_1)) = sgn_\phi(e_1) sgn_{U_2}(e_1) = 1 \cdot -1 = -1, \quad sgn_{\phi_{\mathbb{O}}(U_2)}(\phi(e_2)) = 1 \cdot 0 = 0,$$
$$sgn_{\phi_{\mathbb{O}}(U_2)}(\phi(e_3)) = -1 \cdot 1 = -1, \quad sgn_{\phi_{\mathbb{O}}(U_2)}(\phi(e_4)) = -1 \cdot 1 = -1, \quad sgn_{\phi_{\mathbb{O}}(U_2)}(\phi(e_5)) = 1 \cdot -1 = -1.$$

Therefore we see that

$$sgn_{\phi_{\mathbb{O}}(U_2)}(f_5) = -1, \quad sgn_{\phi_{\mathbb{O}}(U_2)}(f_3) = 0, \quad sgn_{\phi_{\mathbb{O}}(U_2)}(f_2) = -1,$$
$$sgn_{\phi_{\mathbb{O}}(U_2)}(f_4) = -1, \quad sgn_{\phi_{\mathbb{O}}(U_2)}(f_1) = -1.$$

This means that in the image $\phi_{\mathbb{O}}(U_2)$, the orientation of the edges f_1, f_2, f_4 and f_5 have to be reversed in comparison to orientation (H, Γ'). We see that $\phi_{\mathbb{O}}(U_2)$ is:



Note that in the orientation U_2 the vertices e_1, e_3, e_4 and e_5 form an oriented cycle. In $\phi_{\mathbb{O}}(U_2)$, we have that the image of these vertices is f_1, f_2, f_4 and f_5, which form an oriented cycle in $\phi_{\mathbb{O}}(U_2)$. ♢

In the example we saw how $\phi_{\mathbb{O}}$ mapped an oriented cycle to an oriented cycle. In the next theorem, we will see that this behaviour is no coincidence.

Lemma 4.6. *[Gri23, Lem. 5.3] Image of oriented cycle in U under ϕ is an oriented cycle in $\phi_{\mathbb{O}}(U)$.*

In Theorem 4.4 we saw that the non-special divisors are in bijection with the acyclic orientations without a unique source. To obtain $\phi_{\mathbb{O}}$, we start of with ϕ a cyclic bijection, we know that the amount of unoriented cycles in $U \in \mathbb{O}(G)$ is the same as in $\phi_{\mathbb{O}}(U)$. Therefore $\phi_{\mathbb{O}}$ induces a bijection between the non-special divisors of G and H and the special divisors of G and H.

Corollary 4.7. *[Gri23, Cor. 5.4] A map $\phi_{\mathbb{O}}$ induces a bijection between the non-special and special orientation classes.*

4.6 Rigidity

Using the maps defined in the previous section, we want to design a diagram that commutes exactly when the mapping $\phi_* : \mathrm{Jac}(G) \to \mathrm{Jac}(H)$ preserves the theta divisor. For this diagram, we define i to be the mapping

$$i_{e_G} : \mathrm{Pic}_{g-1}(G) \to \mathrm{Jac}(G), \quad [D] \mapsto [D - (g-1) \cdot t(e_G)]$$

where (G, e_G) is a based graph. This is alike the mapping $\phi_{g-1,0,t(e_G)}$ from Lemma 2.2. Now let us define the diagram. When this diagram commutes, we call it rigid.

Definition. *[Gri23, Def 5.7] Let $\phi : (G, \Gamma, e_G) \to (H, \Gamma', e_H)$ be a base preserving cyclic bijection between two 2-edge connected oriented graphs. Consider the following diagram:*

$$\begin{array}{ccc}
\mathbb{O}(G) & \xrightarrow{\quad \phi_{\mathbb{O}} \quad} & \mathbb{O}(H) \\
\downarrow{\scriptstyle c} & & \downarrow{\scriptstyle c} \\
Pic_{g-1}(G) & & Pic_{g-1}(H) \\
\downarrow{\scriptstyle i_{e_G}} & & \downarrow{\scriptstyle i_{e_H}} \\
Jac(G) & \xrightarrow{\quad \phi_* \quad} & Jac(H)
\end{array}$$

If this diagram commutes, we call ϕ rigid.

If we take a look at the set of sourceless orientations in $\mathbb{O}(G)$, we see that with the Chern class map this maps down to the discrete theta divisor $\theta_{e_G} \subset \text{Jac}(G)$. Therefore if ϕ_* is a theta divisor preserving isomorphism between $\text{Jac}(G)$ and $\text{Jac}(H)$, this scheme can commute.

To measure whether or not this scheme commute, a so-called *rigidity divisor* is defined next.

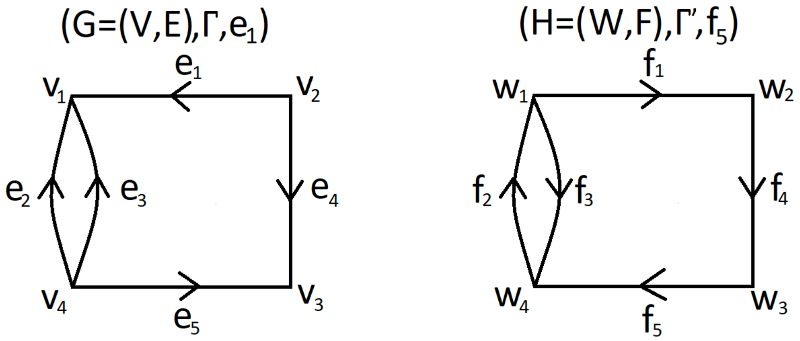
Definition. *[Gri23, Def. 5.8] Let $\phi : (G, \Gamma, e_G) \to (H, \Gamma', e_H)$ be a base preserving cyclic bijection between two 2-edge connected oriented graphs. The rigidity divisor $\mathcal{E}_\phi \in (Jac(H), \boxplus)$ given by*

$$\mathcal{E}_\phi = \phi_*(i_{e_G}(c(\Gamma))) \boxplus (-i_{e_H}(c(\Gamma'))) \boxplus \left[\sum_{e \in E_G, sgn_\phi(e) = -1} t_{\Gamma'}(\phi(e)) - o_{\Gamma'}(\phi(e)) \right],$$

where the orientation $\Gamma' = (t_{\Gamma'}, o_{\Gamma'})$.

Let us calculate the rigidity divisor of the mapping ϕ we saw previously in this chapter.

Example 4.9. *Let us again consider the based oriented 2-connected graphs $(G = (V, E), \Gamma, e_1)$ and $(H = (W, F), \Gamma', f_5)$:*



In Example 4.6 we saw that ϕ defined by

$$\phi(e_1) = f_5, \quad \phi(e_2) = f_3, \quad \phi(e_3) = f_2, \quad \phi(e_4) = f_4, \quad \phi(e_5) = f_1,$$

is a cyclic bijection. In Example 4.7 we saw that ϕ induced the sign function sgn_ϕ with

$$sgn_\phi(e_1) = 1, \quad sgn_\phi(e_2) = 1, \quad sgn_\phi(e_3) = -1, \quad sgn_\phi(e_4) = -1, \quad sgn_\phi(e_5) = 1.$$

Let us now calculate $c(\Gamma)$ and $c(\Gamma')$. We get that:

$$c(\Gamma) = \left[\sum_{e \in E} t_\Gamma(e) - \sum_{v \in V} v \right] = [3v_1 + 2v_3 - v_1 - v_2 - v_3 - v_4] = [2v_1 + v_3 - v_2 - v_4] \in Pic_1(G).$$

Also, we have that

$$c(\Gamma') = \left[\sum_{f\in F} t_{\Gamma'}(f) - \sum_{w\in W} w\right] = [w_4] \in Pic_1(H).$$

We have to map these to the Jacobians using the map i that use the base edge of the graph. We see:

$$i_{e_1}(c(\Gamma)) = [2v_1+v_3-v_2-v_4-(g(G)-1)\cdot t_\Gamma(e_1)] = [2v_1+v_3-v_2-v_4-v_1] = [v_1+v_3-v_2-v_4] \in Jac(G),$$

and

$$i_{f_5}(c(\Gamma')) = [w_4 - (g(H)-1)\cdot t_{\Gamma'}(f_5)] = [w_4 - w_4] = [0] \in Jac(H).$$

With ϕ, we only know how to map edges to H. Therefore we have to translate $i(c(\Gamma))$ to a sum of edges for the mapping with $\phi*$. This process is described Remark 5.10 in Griffith [Gri23], and we will reformulate it here as well. We use an object h_e, where e is an edge. Bluntly said, this object lives in $H^1(G,\mathbb{R})/H^1(G,\mathbb{Z})$, where $H^1(G,\mathbb{R})$ is the span of the algebraic cycles of the oriented graph G. We map these objects to divisors by

$$h_e \mapsto [t(e) - o(e)].$$

for any divisor in Jac(G), we can find a formal sum of h_e's that map to this divisor. In this case we see that

$$h_{e_2} + h_{e_4} \mapsto [v_1 + v_3 - v_2 - v_4].$$

We can translate the h_e to h_f by using the mapping

$$h_e \stackrel{\phi_*}{\longmapsto} sgn_\phi(e) h_{\phi(e)}.$$

We get that

$$h_{e_2} + h_{e_4} \mapsto h_{f_3} - h_{f_4}.$$

We can map this to a divisor in Jac(G):

$$h_{f_3} - h_{f_4} \mapsto [w_4 - w_1 - (w_3 - w_2)] = [w_4 - w_1 + w_2 - w_3].$$

In Griffith, the way of obtaining $\phi_*(i_{e_1}(c(\Gamma)))$ is defined more rigorously. We get that $\phi_*(i_{e_1}(c(\Gamma))) = [w_4 - w_1 + w_2 - w_3]$.

We saw in Example 4.7 that ϕ is edge reversing on e_3 and e_4. Therefore the rigidity divisor is:

$$\begin{aligned}
\mathcal{E}_\phi &= \phi_*(i_{e_G}(c(\Gamma))) \boxplus (-i_{e_H}(c(\Gamma'))) \boxplus \left[\sum_{e\in E_g, sgn(e)=-1} t(\phi(e)) - o(\phi(e))\right] \\
&= [w_4 - w_1 + w_2 - w_3] \boxplus [0] \boxplus [t(\phi(e_3)) - o(\phi(e_3)) + t(\phi(e_4)) - o(\phi(e_4))] \\
&= [w_4 - w_1 + w_2 - w_3] \boxplus [0] \boxplus [t(f_2) - o(f_2) + t(f_4)) - o(f_4)] \\
&= [w_4 - w_1 + w_2 - w_3] \boxplus [0] \boxplus [w_1 - w_4 + w_3 - w_2] \\
&= [0]
\end{aligned}$$

In the next theorem, we will see that ϕ_* is a discrete theta divisor preserving isomorphism because the rigidity divisor $\mathcal{E}_\phi$ is the neutral element in $Jac(H)$. ◊

In the next theorem, we will see that we can actually use this rigidity divisor to determine whether ϕ is rigid. The rigidity divisor is obtained by chasing any configuration U through the diagram [Gri23, Thm. 5.10].

Theorem 4.8. *[Gri23, Thm. 5.9] Let $\phi : (G, \Gamma, e_G) \to (H, \Gamma', e_H)$ be a base preserving cyclic bijection between two 2-edge connected oriented graphs of genus at least two. The following statements are equivalent:*

1. *ϕ is rigid.*
2. *$[\mathcal{E}_\phi] = [0]$.*
3. *$\phi_*(\theta_{e_G}) = \theta_{e_H}$.*

The proof of this theorem can be found in [Gri23, Thm. 5.9].

4.7 Torelli Theorem

In the following theorem, which is the main result of the paper of Griffith [Gri23], we conclude that if ϕ is a rigid cyclic bijection, G and H must be isomorphic graphs.

Theorem 4.9. *[Gri23, Thm. 6.6] Let $\phi : (G, \Gamma, e) \to (H, \Gamma', w)$ be a base preserving cyclic bijection between 2-edge connected graphs of genus at least two. If ϕ is rigid, then G and H are isomorphic graphs.*

Whilst throughout we worked with G and H being oriented graphs to define the different maps with orientation from Section 4.5, the proof of this theorem actually does not depend on a specific orientation of the graphs. This leads us to the following corollary from Griffith.

Corollary 4.10. *[Gri23, Cor. 6.8] Rigidity is independent of base orientations Γ and Γ'.*

Now, it seems as though this theorem proves only the implication from a discrete theta divisor preserving isomorphism to isomorphic graphs, but if we ponder for a little we can quite quickly convince ourselves that the other way around is also true. If two graphs $G = (V_G, E_G)$ and $H = (V_H, E_H)$ are isomorphic, we have that the graph isomorphism

$$f : (V_G, E_G) \to (V_H, E_H),$$

maps a simple cycle in G to a simple cycle in H. Therefore we have that the graphic matroids of G and H are isomorphic as well. Using Theorem 4.2 we immediately obtain that the Jacobians are isomorphic. The isomorphism between the two graphs induces a base preserving cyclic bijection ϕ if we chose any orientation of the graphs and two base edges that get mapped to each other by the isomorphism. We get that this ϕ is rigid.

Chapter 5

Non-special divisors and minimal configurations

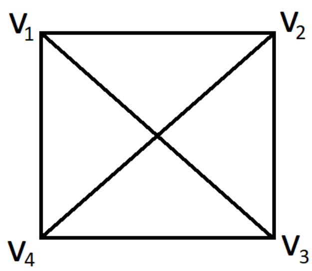
In this chapter, we will take a look at properties of special and non-special divisor classes as defined in the previous chapter. First, we will determine that the q-weight of non-special q-reduced divisors and rank of non-special divisor classes are different from their special counterpart. After this, we will prove that there is a bijection between the set of recurrent configurations $\mathcal{R}$ and the divisor classes in $\text{Pic}_{g-1}(G)$, which maps minimal configurations to non-special divisors and non-minimal configurations to special divisors. Everything in this chapter is an original contribution.

5.1 Weight and rank of non-special divisors

As defined in Chapter 4, the divisor classes can be divided into two sets: the special and the non-special divisors. In this section we will look at the weight and rank of special and non-special divisors.

In Lemma 2.11 we saw an upper and lower bound for the q-weight of q-reduced divisors. In the next example, we will investigate the q-weight of special and non-special q-reduced divisors.

Example 5.1. *Let us consider the following graph $G = (V, E)$:*



Note that the genus of the graph is

$$g(G) = |E| - |V| + 1 = 6 - 4 + 1 = 3.$$

The special and non-special divisors live in $Pic_{g-1}(G)$, therefore we will take a look at all q-reduced divisors in $Pic_2(G)$. Let $q = v_4$. In the following table we have listed all v_4-reduced divisors together with a classification of them being special or non-special and their v_4-weight.

v_4-reduced divisor	(non-)special	v_4-weight
$2v_4$	special	0
$v_1 + v_4$	special	1
$v_2 + v_4$	special	1
$v_3 + v_4$	special	1
$2v_1$	special	2
$2v_2$	special	2
$2v_3$	special	2
$v_1 + v_2$	special	2
$v_1 + v_3$	special	2
$v_2 + v_3$	special	2
$v_1 + 2v_2 - v_4$	non-special	3
$2v_1 + v_2 - v_4$	non-special	3
$v_1 + 2v_3 - v_4$	non-special	3
$2v_1 + v_3 - v_4$	non-special	3
$v_2 + 2v_3 - v_4$	non-special	3
$2v_2 + v_3 - v_4$	non-special	3

Note that there is no special v_4-reduced divisor with the same v_4-weight as a non-special v_4-reduced divisor. Also note how the weight of non-special divisors equals the upper bound of q-weight for q-reduced divisors we found in Lemma 2.11. ♢

In the example we saw that the q-weight of the non-special divisors is bigger than the q-weight of special divisors. In the next lemma, we will prove this is always the case. In fact, we will prove here the q-weight of the q-reduced representative of the non-special divisor classes is equal to the upper bound of the q-weight of q-reduced divisors found in Lemma 2.11.

Lemma 5.1. *Let $G = (V \cup \{s\}, E)$. Let $q \in V \cup \{s\}$ and let D be the q-reduced representative of some $[D] \in Pic_{g-1}(G)$. Then D is a non-special divisor if and only if the q-weight of D equals $w_q(D) = |E| - |V|$.*

Proof. Let $[D] \in \mathrm{Pic}_{g-1}(G)$ be some equivalence class and D its q-reduced representative.

"⇒" Assume that D is a non-special divisor. As $D = \sum_{v\in V\cup\{s\}} a_v \cdot v$ is a non-special q-reduced divisor, we have by definition that $\sum_{v\in V\cup\{s\}\setminus\{q\}} a_v > g - 1$. Using Lemma 2.11 we see that

$$g - 1 = |E| - |V \cup \{s\}| < \sum_{v\in V\cup\{s\}\setminus\{q\}} a_v = w_q(D) \leq |E| - |V|.$$

We conclude that the q-weight of D equals $w_q(D) = |E| - |V|$ if D is a non-special divisor.

"⇐" We will now prove the implication from right to left using contraposition. Assume that D is a special divisor. Then we know that

$$D = \sum_{v\in V\cup\{s\}} a_v \cdot v,$$

with $a_v \geq 0$ for all $v \in V \cup \{s\}$ and $\sum_{v\in V\cup\{s\}} a_v = g - 1$. Then we see that

$$w_q(D) = \sum_{v\in V\cup\{s\}\setminus\{q\}} a_v \leq g - 1 = |E| - |V \cup \{s\}| < |E| - |V|.$$

We conclude that if D is a special divisor, then $w_q(D) < |E| - |V|$. □

If we take another look at the table in Example 5.1, we can see another difference between the special and non-special q-reduced divisors: the special divisors are effective and the non-special divisors are not. Remember that a divisor $D = \sum_{v\in V} a_v \cdot v$ is effective if $a_v \geq 0$ for all $v \in V$. This observation leads us to the following lemma. This statement is also made in [Gri23, Thm 3.18] but not proven. We will provide a proof here. Interesting to note here is how in some literature, the special and non-special sets are defined based on their degree and rank [Bac17, Sec. 2], instead of via the Abel-Jacobi maps as we did in Section 4.3.

Lemma 5.2. *Let $G = (V \cup \{s\}, E)$. Let $[D] \in Pic_{g-1}(G)$. Then the rank of D is bigger or equal to 0 if and only if $[D]$ is a special divisor class.*

Proof. "$\Leftarrow$" Let $q \in V$ be given. As the rank of a divisor is a class property and every class has a q-reduced divisor by Theorem 2.7, we only have to prove the statement for D a q-reduced divisor.

By Lemma 2.11 and Lemma 5.1, we know that the weight of a q-reduced divisor D equals $|E| - |V| + 1$ if D is non-special and $0 \leq w_q(D) < |E| - |V| + 1$ if D is special. As $D \in Div_{g-1}$, we see that $a_q = -1$ when D is non-special and $a_q \geq 0$ when D is special. It follows that the rank of D is bigger or equal to 0 if D is a special q-reduced divisor, as D is an effective divisor in this case.

"$\Rightarrow$" Now assume that D is non-special q-reduced divisor. We will prove by contradiction that D has a negative rank. Assume it has a non-negative rank. Then D must be equivalent to some effective divisor D', i.e.

$$D' \in [D],$$

with $D'(v) \geq 0$ for all $v \in V \cup \{s\}$. As $D' \in \mathrm{Pic}_{g-1}(G)$, we know by definition that

$$\sum_{v \in V \cup \{s\}} D'(v) = g - 1.$$

Then by definition we know that D' is a special representative of $[D]$. This is a contradiction, as D is a non-special divisor, so its class cannot contain a special representative. We conclude that D has a negative rank. □

5.2 Relating non-special divisors and minimal configurations

In Theorem 4.4 we saw how the non-special divisors are in bijection with the acyclic orientations with a unique source. This already suggests a connection to minimal configurations, as we saw in Theorem 1.10 that there is a bijection between acyclic orientations and minimal configurations as well. In this section, we will make this connection more concise by giving a direct bijection between the minimal configurations and the non-special divisors.

First, let us look at an example that shows this connection between minimal configurations and non-special divisors.

Example 5.2. *Let us revisit the graph from Example 5.1 and let $s = q = v_4$. The natural way to map a recurrent configuration $\eta \in \mathcal{R}$ to the Picard set $Pic_{g-1}(G)$, just as defined in Corollary 2.6, is*

$$\phi_{0,g-1,v_4} \circ f : \mathcal{R} \to Pic_{g-1}(G), \quad \eta \mapsto [\eta'],$$

where η' is defined as

$$\eta' = \sum_{v \in V \setminus \{v_4\}} \eta(v) \cdot v + \left(g - 1 - \sum_{v \in V \setminus \{v_4\}} \eta(v) \right) \cdot v_4 \in Div_{g-1}(G).$$

By Corollary 2.6, we know this function gives a bijection. We will now expand the table given in Example 5.1 by adding the divisor η' that is in the class of the q-reduced divisor and whether or not the pre-image $\eta \in \mathcal{R}$ is a minimal or non-minimal configuration.

v_4-reduced divisor	(non-)special	η' in the class	η is (non-)minimal
$2v_4$	special	$2v_1 + 2v_2 + 2v_3 - 4v_4$	non-minimal
$v_1 + v_4$	special	$2v_1 + v_2 + v_3 - 2v_4$	non-minimal
$v_2 + v_4$	special	$v_1 + 2v_2 + v_3 - 2v_4$	non-minimal
$v_3 + v_4$	special	$v_1 + v_2 + 2v_3 - 2v_4$	non-minimal
$2v_1$	special	$2v_2 + 2v_3 - 2v_4$	non-minimal
$2v_2$	special	$2v_1 + 2v_3 - 2v_4$	non-minimal
$2v_3$	special	$2v_1 + 2v_2 - 2v_4$	non-minimal
$v_1 + v_2$	special	$2v_1 + 2v_2 + v_3 - 3v_4$	non-minimal
$v_1 + v_3$	special	$2v_1 + v_2 + 2v_3 - 3v_4$	non-minimal
$v_2 + v_3$	special	$v_1 + 2v_2 + 2v_3 - 3v_4$	non-minimal
$v_1 + 2v_2 - v_4$	non-special	$v_1 + 2v_2 - v_4$	minimal
$2v_1 + v_2 - v_4$	non-special	$2v_1 + v_2 - v_4$	minimal
$v_1 + 2v_3 - v_4$	non-special	$v_1 + 2v_3 - v_4$	minimal
$2v_1 + v_3 - v_4$	non-special	$2v_1 + v_3 - v_4$	minimal
$v_2 + 2v_3 - v_4$	non-special	$v_2 + 2v_3 - v_4$	minimal
$2v_2 + v_3 - v_4$	non-special	$2v_2 + v_3 - v_4$	minimal

We see how $\phi_{0,g-1,v_4} \circ f$ in this case gives a bijection between the recurrent configurations $\mathcal{R}$ and the Picard set $Pic_{g-1}(G)$ that maps minimal configurations to non-special divisors and non-minimal configurations to special divisors. ♢

In the example, we saw how there are indeed the same amount of non-special divisors and minimal configurations. It is an open question whether the mapping $\phi_{0,g-1,s} \circ f$ always maps minimal configurations to non-special divisors.

Conjecture 5.3. *Let $G = (V \cup \{s\}, E)$ be given. The mapping*

$$\phi_{0,g-1,s} \circ f : \mathcal{R} \to Pic_{g-1}(G), \quad \eta \mapsto [\eta'],$$

where

$$\eta' = \sum_{v \in V \setminus \{v_4\}} \eta(v) \cdot v + \left(g - 1 - \sum_{v \in V \setminus \{v_4\}} \eta(v) \right) \cdot v_4 \in Div_{g-1}(G),$$

is a bijection that maps minimal configurations to non-special divisors.

Whilst for it is not known if this natural mapping $\phi_{0,g-1,s} \circ f$ always maps minimal configurations to non-special divisors, we will in the next theorem prove a bijection between the recurrent configurations $\mathcal{R}$ and the Picard set $\mathrm{Pic}_{g-1}(G)$ that maps minimal configurations to non-special divisors and non-minimal configurations to special divisors. For this proof, we will use the divisor $\mathcal{M}$ defined as:

$$\mathcal{M}(v) = \begin{cases} \deg_G(v) - 1 & \text{if } v \in V, \\ \deg_G(s) - |V| - 1 & \text{if } v = s. \end{cases}$$

The degree of $\mathcal{M}$ equals

$$\begin{aligned} \sum_{v \in V} \mathcal{M}(v) &= \deg_G(s) - |V| - 1 + \sum_{v \in V \setminus \{s\}} \deg(v) - 1 \\ &= -|V| + \sum_{v \in V} \deg(v) - \sum_{v \in V} 1 \\ &= -|V| + \sum_{e \in E} 2 - \sum_{v \in V} 1 \\ &= 2|E| - 2|V| = 2(g-1). \end{aligned}$$

Hence, $\mathcal{M} \in \mathrm{Pic}_{2(g-1)}(G)$. Now we will show the bijection between the recurrent configurations $\mathcal{R}$ and the Picard set $\mathrm{Pic}_{g-1}(G)$ that maps minimal configurations to non-special divisors and non-minimal configurations to special divisors.

Theorem 5.4. *Let $G = (V \cup \{s\}, E)$ be a finite, connected graph. The function $\mathfrak{M} : \mathcal{R} \to Pic_{g-1}(G)$, defined by*

$$\mathfrak{M} : \eta \mapsto [\mathcal{M} - \eta'],$$

where

$$\eta' := \sum_{v \in V \setminus \{s\}} \eta(v) \cdot v + \left(g - 1 - \sum_{v \in V \setminus \{s\}} \eta(v) \right) \cdot s,$$

is a bijection. In particular, $\mathfrak{M}$ induces a bijection between minimal configurations in $\mathcal{R}$ and non-special divisor classes in $Pic_{g-1}(G)$ and a bijection between non-minimal configurations in $\mathcal{R}$ and special divisor classes in $Pic_{g-1}(G)$.

Proof. Let us start with proving this function is well-defined. Let η be a recurrent configuration in $\mathcal{R}$. Note that $[\eta'] \in \mathrm{Pic}_{g-1}(G)$, as its degree is

$$\deg(\eta') = \sum_{v \in V \setminus \{s\}} \eta(v) - \left(g - 1 - \sum_{v \in V \setminus \{s\}} \eta(v) \right) = g - 1.$$

As $\mathcal{M} \in \mathrm{Pic}_{2(g-1)}(G)$ we have that the degree of $\mathfrak{M}(\eta) = [\mathcal{M} - \eta']$ is $2(g-1) - (g-1) = g - 1$, so $\mathfrak{M}(\eta) \in \mathrm{Pic}_{g-1}(G)$.

Now let us prove injectivity. Let $\eta, \nu \in \mathcal{R}$ be two recurrent configurations, such that $\mathfrak{M}(\eta) = \mathfrak{M}(\nu)$. Then we have that

$$[\mathcal{M} - \eta'] = [\mathcal{M} - \nu'].$$

By Lemma 2.4 we know there exists a vector $x \in \mathbb{Z}^{|V \cup \{s\}|}$ such that we can write the divisors in vector form to get

$$\mathcal{M} - \eta' - (\mathcal{M} - \nu') = Lx,$$

where L is the $V \cup \{s\} \times V \cup \{s\}$ Laplacian matrix, as defined in Section 1.1.1. This equation reduces to

$$\nu' - \eta' = Lx.$$

We now want to find a vector $y \in \mathbb{Z}^{|V \cup \{s\}|}$ with $y_{n+1} = 0$ such that

$$\nu' - \eta' = Ly,$$

where $|V| = n$. We can repeat the argument from the proof of Theorem 2.5, where we saw that by definition of the Laplacian matrix, we have that

$$L \cdot \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} \deg_G(v_1) - \sum_{v \in V \cup \{s\}} \mathbb{1}_{v_1 \sim v} \\ \deg_G(v_2) - \sum_{v \in V \cup \{s\}} \mathbb{1}_{v_2 \sim v} \\ \vdots \\ \deg_G(v_n) - \sum_{v \in V \cup \{s\}} \mathbb{1}_{v_n \sim v} \\ \deg_G(s) - \sum_{v \in V \cup \{s\}} \mathbb{1}_{s \sim v} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix},$$

where $\mathbb{1}_{v \sim w}$ equals 1 if v and w are neighbours and 0 if they are not. Define y to be

$$y := x - x_{n+1} \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \\ 1 \end{pmatrix},$$

where $x_{n+1} \in \mathbb{Z}$ is the $n+1$-th entry in the vector x. We now have that $y_{n+1} = 0$ and

$$\nu' - \eta' = Ly.$$

It follows that

$$\nu - \eta = L'y',$$

where L' is the $V \times V$ Laplacian and $y' \in \mathbb{Z}^n$, such that $y'_i = y_i$ for all $i \in \{1, 2, ..., n\}$. Now we will, just as in the proof of Theorem 2.5, split the indices of y' into two sets dependent on their value being positive or negative:

$$I := \{i \in \{1, ..., n\} : y'_i \geq 0\},$$
$$J := \{i \in \{1, ..., n\} : y'_i < 0\} = \{1, ..., n\} \setminus I.$$

Then let $y'_I \in \mathbb{Z}^n_{\geq 0}$ be the vector such that $y'_{I,i} = y'_i$ if $i \in I$ and $y'_{I,i} = 0$ else. Likewise, let $y'_J \in \mathbb{Z}^n_{<0}$ be the vector such that $y'_{J,j} = y'_j$ if $j \in J$ and $y'_{J,j} = 0$ else. Let

$$\nu^* := \nu - L'y'_I.$$

We have that

$$\nu - L'y'_I = \eta + L'y' - L'y'_I = \eta + L'y'_J = \eta - L'(-y_J)'.$$

As η and ν are both recurrent (stable) configurations, we see that

$$S(\nu^*) = \nu, \quad S(\nu^*) = \eta,$$

by the definition of stabilization in Lemma 1.1, as y'_I is the odometer function for ν and y'_J is the odometer function for η. By Lemma 1.1, we know that stabilization is well-defined and has a unique resulting configuration. Therefore we must have that $\nu = \eta$ and we conclude that $\mathfrak{M}$ is injective.

Now we will prove that $\mathfrak{M}$ is surjective. In Corollary 2.6 we already saw that there is a bijection between the recurrent elements in $\mathcal{R}$ and the divisor classes of any Picard group, in particular the divisor classes of $\mathrm{Pic}_{g-1}(G)$. We also know that both sets are finite, as $|\mathcal{R}| = \det(L')$ by Corollary 1.6. As we have already proven that $\mathfrak{M}$ is injective and we now have that $|\mathrm{Pic}_{g-1}(G)| = |\mathcal{R}|$, we can conclude that $\mathfrak{M}$ is surjective as well.

Before we can prove that $\mathfrak{M}$ maps minimal configurations to non-special divisors and non-minimal configurations to special divisors, we will first prove by contradiction that given an $\eta \in \mathcal{R}$, we have that the s-reduced representative of class $[\mathcal{M} - \eta'] \in \mathrm{Pic}_{g-1}(G)$ is $\mathcal{M} - \eta'$. For contradiction sake, assume that $\mathcal{M} - \eta'$ is not an s-reduced divisor. As $\eta(v) \leq \deg_G(v) - 1$ for all $v \in V$, we have that $(\mathcal{M} - \eta')(v) \geq 0$ for all $v \in V$. Therefore by definition of s-reduced divisors, there must exist a non-empty set $A \subset V$ such that for all $v \in A$ we have $(\mathcal{M} - \eta')(v) \geq outdeg_A(v)$. Let $v \in A$. Then we see that

$$(\mathcal{M} - \eta')(v) = \deg_G(v) - 1 - \eta(v) \geq outdeg_A(v).$$

We rewrite this to be a restriction on the height of $\eta(v)$ and see that

$$\eta(v) \leq \deg_G(v) - 1 - outdeg_A(v) = indeg_A(v) - 1 < \sum_{y \in A \setminus \{v_i\}} (-L'_{vy}),$$

holds for every $v \in A$. As $A \neq \emptyset$, we have that η contains a forbidden subconfiguration (A, η_A). This is a contradiction, as η is a recurrent configuration, which by Theorem 1.3 we know contains no forbidden subconfigurations. Therefore our assumption must be wrong and we conclude that $\mathcal{M} - \eta'$ is a s-reduced divisor.

We will now prove that $\mathfrak{M}$ maps minimal configurations to non-special divisors. Let $\eta \in \mathcal{R}$ be a minimal configuration. By Lemma 1.8 we know that its weight equals

$$w(\eta) = \sum_{v \in V} \eta(v) = |E| - \deg_G(s).$$

Therefore the s-weight of $\mathcal{M} - \eta'$ equals

$$\begin{aligned} w_s(\mathcal{M} - \eta') &= \sum_{v \in V} \left(\mathcal{M}(v) - \eta(v)\right), \\ &= \sum_{v \in V} \left(\deg_G(v) - 1 - \eta(v)\right), \\ &= \sum_{v \in V} (\deg_G(v) - 1) - \sum_{v \in V} \eta(v), \\ &= 2|E| - \deg_G(s) - |V| - (|E| - \deg_G(s)) \\ &= |E| - |V|. \end{aligned}$$

As the s-reduced representative of class $[\mathcal{M} - \eta'] \in \mathrm{Pic}_{g-1}(G)$ is $\mathcal{M} - \eta'$, then by Lemma 5.1, we get that $[\mathcal{M} - \eta']$ is indeed a non-special divisor class.

Now let η be a non-minimal configuration. By Lemma 1.7 we know that its weight has the following bounds:

$$|E| - \deg_G(s) \le w(\eta) = \sum_{v \in V} \eta(v) \le 2|E| - \deg_G(s) - |V|.$$

As η is not minimal, we know by Lemma 1.8 that the first inequality is strict:

$$|E| - \deg_G(s) < w(\eta) = \sum_{v \in V} \eta(v) \le 2|E| - \deg_G(s) - |V|.$$

Therefore the s-weight of $\mathcal{M} - \eta'$ is:

$$\begin{aligned} w_s(\mathcal{M} - \eta') &= \sum_{v \in V} \left(\mathcal{M}(v) - \eta(v)\right), \\ &= \sum_{v \in V} \left(\deg_G(v) - 1 - \eta(v)\right), \\ &= \sum_{v \in V} (\deg_G(v) - 1) - \sum_{v \in V} \eta(v). \end{aligned}$$

We see that

$$2|E| - \deg_G(s) - |V| - (2|E| - \deg_G(s) - |V|) \le w_s(\mathcal{M} - \eta') < 2|E| - \deg_G(s) - |V| - (|E| - \deg_G(s)),$$

which reduces to

$$0 \le w_s(\mathcal{M} - \eta') < |E| - |V|.$$

We know that $\mathcal{M} - \eta'$ is the s-reduced representative of class $\mathfrak{M}(\eta) = [\mathcal{M} - \eta'] \in \mathrm{Pic}_{g-1}(G)$, we can now conclude by using Lemma 5.1 that $\mathfrak{M}(\eta)$ is a special divisor.

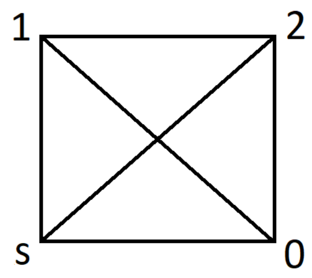
As we have already proven that $\mathfrak{M}$ is a bijection, we conclude that it induces a bijection between minimal configurations and non-special divisor classes and a bijection between non-minimal configurations and special divisor classes. □

Let us look at an example of this mapping.

Example 5.3. *Let us again consider the graph G from Example 5.1 with $v_4 = s = q$. We have that*

$$\mathcal{M} = 2v_1 + 2v_2 + 2v_3 - 2v_4.$$

Consider the configuration η_1



Note that this is a recurrent configuration, as it will output a spanning tree of G when inputted in Algorithm 1. As its weight equals 3, which is $|E| - \deg_G(v_4)$, we have that it is a minimal configuration by Lemma 1.8. Now let us calculate η_1':

$$\eta_1' = v_1 + 2v_2 - v_4 \in Div_2(G).$$

We get that

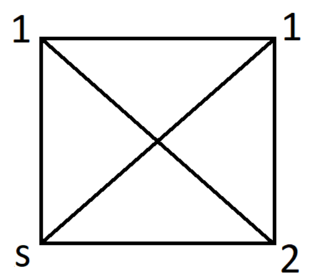
$$\mathfrak{M}(\eta_1) = [\mathcal{M} - \eta_1'] = [v_1 + 2v_3 - v_4].$$

Note that $\mathcal{M} - \eta_1'$ is indeed a v_4-reduced divisor and its v_4-weight is

$$w_{v_4}(\mathcal{M} - \eta_1') = w_{v_4}(v_1 + 2v_3 - v_4) = 3 = |E| - |V \setminus \{v_4\}|.$$

By Lemma 5.1 we conclude that $[\mathcal{M} - \eta_1']$ is a non-special divisor class.

Now let us consider a non-minimal configuration η_2



Again, note that this is a recurrent configuration, as it will output a spanning tree of G when inputted in Algorithm 1 and it is a non-minimal configuration by Lemma 1.8. We get that

$$\eta_2' = v_1 + v_2 + 2v_3 - 2v_4 \in Div_2(G).$$

We get that

$$\mathfrak{M}(\eta_2) = [\mathcal{M} - \eta_2'] = [v_1 + v_2].$$

We again see that $\mathcal{M} - \eta_2'$ is a v_4-reduced divisor and its v_4-weight is

$$w_{v_4}(v_1 + v_2) = 2 < |E| - |V \setminus \{v_4\}|.$$

By Lemma 5.1 we see that η' special divisor. $\Diamond$

5.3 Extending Torelli theorem for graphs

Now that we have found a bijection between the minimal configurations and the non-special divisors, we can state that a minimal configurations preserving isomorphism for recurrent groups $\mathcal{R}$ is as the discrete theta divisor preserving isomorphism for Jacobians. In the next theorem we will formulate a Torelli theorem on graphs using the objects from the probabilistic approach. We will state that if two graphs have a minimal configuration preserving isomorphism for the recurrent groups $(\mathcal{R}_G, \oplus)$ and $(\mathcal{R}_H, \oplus)$, then the graphs are isomorphic.

Theorem 5.5. *Let $G = (V, E)$ and $H = (W, F)$ be 2-edge connected graphs of genus at least two. Then there is a graph isomorphism between G and H that sends the sink of G to the sink of H if and only if there exists a group isomorphism*

$$\mathfrak{R} : (\mathcal{R}_G, \oplus) \to (\mathcal{R}_H, \oplus),$$

that induces a bijection on the minimal configurations of G and H.

Proof. "⇒" As with Torelli theorem, the easier implication is proving such an isomorphism exists if G and H are isomorphic. Let

$$\xi_V : V \to W, \quad \xi_E : E \to F,$$

be a graph isomorphism from G to H such that $\xi_V(s_G) = s_H$.

Define $\mathfrak{R}$ to be this mapping:

$$\mathfrak{R} : (\mathcal{R}_G, \oplus) \to (\mathcal{R}_H, \oplus), \quad \eta \mapsto \nu,$$

with $\nu(\xi_V(v)) = \eta(v)$ for all $v \in V \setminus \{s_G\}$. As ξ_V defines a bijection between V and W, this defines ν on all of $W \setminus \{s_H\}$.

Note that $\xi_V(v) \in W \setminus \{s_H\}$ has the same degree as v. Also, if $v_i \sim v_j$ are neighbours in G for some $v_i, v_j \in V$, we have that $\xi_V(v_i) \sim \xi_V(v_j)$ in H. Therefore we get that η and ν burn in the exact same way when inputted to the burning Algorithm 1. We conclude that ν is a recurrent configuration. This implies that $\mathfrak{R}$ is well-defined. By definition, this mapping preserves the group structure. Specifically, the identity element of $(\mathcal{R}_G, \oplus)$ gets mapped to $(\mathcal{R}_H, \oplus)$. Therefore, $\mathfrak{R}$ is in fact a group isomorphism.

Now we will argue that $\mathfrak{M}$ induces a bijection on minimal configurations. Note that the amount of minimal configurations on G must be equal to the amount of minimal configurations on H, as we can map acyclic orientations of G with a unique source at s_G to acyclic orientations of H with a unique source at s_H bijectively using the graph isomorphism. As the minimal configurations are in bijection with the acyclic orientation with a unique source by Theorem 1.10, we must have that G and H must have the same number of minimal configurations.

Now let us look at the weight of a minimal configuration $\eta \in \mathcal{R}_G$. By Lemma 1.8, we know that

$$w(\eta) = \sum_{v \in V \setminus \{s_G\}} \eta(v) = |E| - \deg_G(s_G).$$

Then we see that the weight of image $\mathfrak{R}(\eta)$ is:

$$w(\mathfrak{R}(\eta)) = \sum_{w' \in W \setminus \{s_H\}} \mathfrak{R}(\eta)(w') = \sum_{v \in V \setminus \{s_G\}} \eta(v) = |E| - \deg_G(s_G).$$

As our graph isomorphism ξ_V maps s_G to s_H, we know that $\deg_G(s_G) = \deg_H(s_H)$. Using again Lemma 1.8, we see that $w(\mathfrak{R}(\eta))$ is a minimal configuration of H.

We conclude that $\mathfrak{R}$ is group isomorphism from $(\mathcal{R}_G, \oplus)$ to $(\mathcal{R}_H, \oplus)$ that induces a bijection between the minimal configurations of G and H.

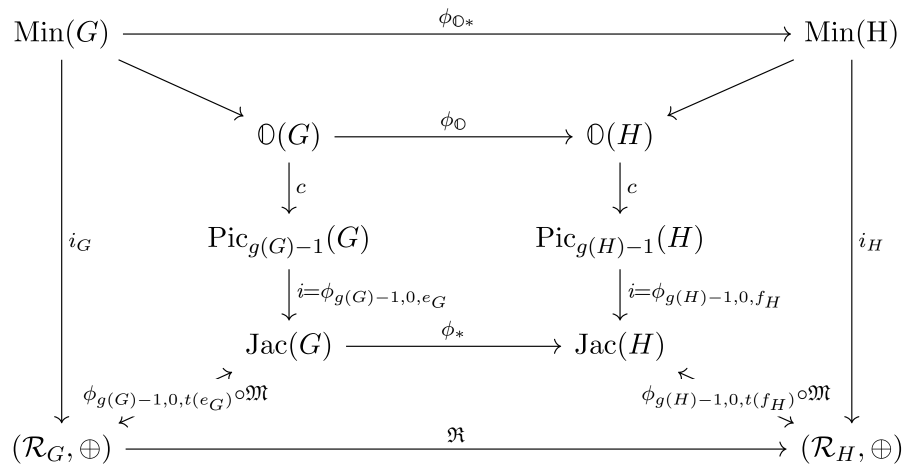


Figure 5.1: The inner diagram is the rigidity diagram from Section 4.6. We extend this with the outer layer using the bijection between the sandpile groups $(\mathcal{R}_G, \oplus)$ and $(\mathcal{R}_H, \oplus)$, which induces a bijection between the minimal configurations.

"$\Leftarrow$" We will expand the scheme for rigidity from Section 4.6 with mappings on the recurrent sandpile group and minimal configurations. Let $\mathrm{Min}(G)$ and $\mathrm{Min}(H)$ denote the set of minimal configurations in $\mathcal{R}_G$ and $\mathcal{R}_H$ respectively. As the minimal configurations are a subset of the recurrent configurations, we can map them to the corresponding recurrent sandpile group with an inclusion which we call $i_G : \mathrm{Min}(G) \to (\mathcal{R}_G, \oplus)$ and $i_H : \mathrm{Min}(H) \to (\mathcal{R}_H, \oplus)$. Let $\phi_{\mathbb{O}*}$ be that bijection between the minimal configurations $\mathrm{Min}(G)$ and $\mathrm{Min}(H)$ induced by $\mathfrak{R}$. In the scheme, we will also use the bijection between minimal configurations and acyclic orientations we proved in Theorem 1.10.

To map the recurrent sandpile group $(\mathcal{R}, \oplus)$ to the Jacobian, we will need a base edge e_G of G and a base edge f_H of H which we orient. A natural choice are two edges such that $t(e_G) = s$ and $t(f_H) = s$, but any edge and orientation of the edge can be chosen. We will use the mapping $\mathfrak{M}$ from $\mathcal{R}_G$ to $\mathrm{Pic}_{g(G)-1}(G)$ from Theorem 5.4, which we combine with the mapping $\phi_{g(G)-1,0,t(e_G)}$ from Lemma 2.2 to map from $\mathrm{Pic}_{g(G)-1}(G)$ to $\mathrm{Jac}(G)$.

The implied mapping from $\mathrm{Jac}(G)$ to $\mathrm{Jac}(H)$ is defined by:

$$\phi_* : \mathrm{Jac}(G) \to \mathrm{Jac}(H), \quad \phi_*([D]) = \phi_{g(H)-1,0,t(f_H)} \circ \mathfrak{M} \circ \mathfrak{R} \circ \mathfrak{M}^{-1} \circ \phi^{-1}_{g(G)-1,0,t(e_G)}([D])$$

Note that we saw in the proof of Lemma 2.2 that $\phi^{-1}_{g(G)-1,0,t(e_G)} = \phi_{0,g-1,t(e_G)}$.

To prove this scheme commutes, i.e. the implied mapping ϕ_* is rigid, we will use that $\phi_*(\theta_{e_G}) = \theta_{e_H}$ implies that ϕ_* is rigid by Theorem 4.8 (implication third to first equivalent statement).

Let $[D] \in \mathrm{Jac}(G)$ be in the discrete theta divisor θ_{e_G}. Then we know that there is a representative D' of $[D]$ that can be written as

$$D' = v_{i_1} + ... + v_{i_{g(G)-1}} - (g(G) - 1)t(e_G),$$

for some $i_1, ..., i_{g(G)-1} \in \{1, 2, ..., |V|\}$. We get that $\phi^{-1}_{g(G)-1,0,t(e_G)}([D]) = \phi_{0,g(G)-1,t(e_G)}[D] = [D + (g(G) - 1)t(e_G)]$. Then we see that

$$D' + (g(G) - 1)t(e_G) = v_{i_1} + ... + v_{i_{g(G)-1}} \in [D + (g(G) - 1)t(e_G)] \in \mathcal{S}(G).$$

We get that $\phi^{-1}_{g(G)-1,0,t(e_G)}([D]) = [D + (g(G) - 1)t(e_G)]$ is a special divisor class.

Therefore $\mathfrak{M}^{-1}(\phi^{-1}_{g(G)-1,0,t(e_G)}([D]))$ is a non-minimal configuration. As $\mathfrak{R}$ induces a bijection on minimal configurations and is a bijection between $(\mathcal{R}_G, \oplus)$ and $(\mathcal{R}_H, \oplus)$, it also induces a bijection on non-minimal configurations. Therefore we see that $\mathfrak{R}(\mathfrak{M}^{-1}(\phi^{-1}_{g(G)-1,0,t(e_G)}([D])))$ is also a non-minimal configuration. Then

$$\mathfrak{M}(\mathfrak{R}(\mathfrak{M}^{-1}(\phi^{-1}_{g(G)-1,0,t(e_G)}([D])))) \in \mathcal{S}(H)$$

is a special divisor class. We use $\phi_{g(H)-1,0,t(f_H)}$ to map this divisor class to a divisor class in $\mathrm{Jac}(H)$. As $\mathfrak{M}(\mathfrak{R}(\mathfrak{M}^{-1}(\phi^{-1}_{g(G)-1,0,t(e_G)}([D]))))$ is a special divisor class, we know there is a representative D^* of the divisor class that is of the form

$$D^* = T + (g(H) - 1)t(e_H),$$

where $[T] \in \theta_{e_H}$ is in the discrete theta divisor of H. Then we see that

$$\begin{aligned}
\phi_*([D]) &= \phi_{g(H)-1,0,t(f_H)}(\mathfrak{M}(\mathfrak{R}(\mathfrak{M}^{-1}(\phi^{-1}_{g(G)-1,0,t(e_G)}([D]))))), \\
&= \phi_{g(H)-1,0,t(f_H)}([D^*]), \\
&= [D^* - (g(G) - 1)t(e_H)] = [T] \in \theta_{e_H}.
\end{aligned}$$

As $\mathfrak{R}$, $\mathfrak{M}$, $\phi_{g(G)-1,0,t(e_G)}$ and $\phi_{g(H)-1,0,t(f_H)}$ are all bijections, we have that ϕ_* is a bijection as well.

As $\mathfrak{R}$ induces bijections between the minimal configurations of G and H and between the non-minimal configuration of G and H, we have that they have the same size. Combining this with $\mathfrak{M}$ we get that the special set of G has the same size as the special set of H. This implies that the discrete theta divisor of G is of the same size as the discrete theta divisor of H. We conclude that

$$\phi_*(\theta_{e_G}) = \theta_{f_H},$$

is a bijective mapping from the discrete theta divisor θ_{e_G} of G and the discrete theta divisor θ_{f_H} of H.

We use the implication $3 \to 1$ from Theorem 4.8, we get that ϕ_* is rigid. We conclude that G and H are isomorphic graphs using Theorem 4.9. If we choose e_G and f_H such that $t(e_G)$ is the sink in G and $t(f_H)$ is the sink in H, the isomorphism we get must map the sink of G to the sink of H.

□

Discussion

In this thesis, we introduced the Abelian sandpile model both in terms of a Markov chain as in terms of divisors. We proved a bijection between the recurrent configurations and spanning trees on G using Dhar's burning bijection in Theorem 1.4. Also, we defined a group operation on the recurrent configurations.

We looked at the special subset of the recurrent configurations called the minimal divisors, for which we provided an original proof that they are in bijection with acyclic orientations with a unique sink in Theorem 1.10. We also found a new surjective map between non-minimal configurations and acyclic partial orientations in Theorem 1.11.

After introducing divisors, the Picard sets and the Jacobian, we provided an original proof of the existence of a bijection between two Picard sets of a different degree in Lemma 2.2. In the literature, both the Jacobian and the group of recurrent sandpile group are referred to as "the sandpile group". We proved an original group isomorphism between the Jacobian and the group of recurrent configurations in Theorem 2.5.

The isomorphism between the Jacobian and the group of recurrent configurations allowed us to formulate a new algorithmic approach to calculating the identity element of the group of recurrent configurations in Chapter 3. In this chapter, we also gave an novel reflection on the difference of starting the heights of vertices at zero or one in the Markov chain.

In Chapter 4 we looked at a new Torelli theorem of graphs by Griffith [Gri23]. We defined special and non-special divisors. In Theorem 5.4 we proved a new bijection from the group of recurrent configurations $(\mathcal{R}, \oplus)$ to the Picard set of degree $g(G) - 1$. This bijection induces a bijection between non-special divisors and minimal configurations, which is a connection that has to the authors knowledge not been known previously. In Theorem 5.5 we could with the help of this bijection state an original Torelli theorem on the groups of recurrent configurations.

Outlook

One problem that remained open in this thesis is whether or not the natural embedding from the recurrent configurations to a Picard set of degree $g - 1$:

$$\phi_{0,g-1,s} \circ f : \mathcal{R} \to \mathrm{Pic}_{g-1}(G), \quad \eta \mapsto [\eta'],$$

where

$$\eta' = \sum_{v \in V \setminus \{s\}} \eta(v) \cdot v + \left(g - 1 - \sum_{v \in V \setminus \{s\}} \eta(v) \right) \cdot s \in \mathrm{Div}_{g-1}(G),$$

as discussed in Conjecture 5.3 is a mapping that sends minimal configurations to non-special divisor classes for all types of graphs.

In Theorem 1.11 we found a surjective mapping ϕ_{pac} from partial orientations with a unique source to non-minimal configurations. In Example 2.7 we saw this mapping is not injective. Perhaps there could be some equivalence defined on the partial orientations to make this mapping injective as well, but this was outside the scope of this thesis.

The mapping $\mathfrak{M}$ from Theorem 5.4 now sends recurrent configurations in $\mathcal{R}$ to the class of the s-reduced divisor in Div_{g-1} and maps minimal configurations to non-special divisors. It might also be interesting to look at bijections that send the recurrent configurations $\mathcal{R}$ to q-reduced representatives for $q \neq s$. If we make small alterations to the burning algorithms for recurrent configurations and q-reduced divisors (for instance using a specific ordering of the edges), we could cause the spanning trees corresponding to minimal configurations and non-special divisors to be of a certain form. This might be done by using the concept of internally/externally active edges from [CL03].

The Tutte polynomial counts the amount of configurations of a certain weight [Big99b; CL03]. This may give an easier tool to determine whether or not the sets of minimal configurations on two graphs are the same. If we could translate the problem of determining whether the minimal configurations of two graphs are bijective to determining some evaluation of the Tutte polynomial of the graphs, this could extent the result of Torelli theorem even further.

One of the first found counterexamples of a naive Torelli theorem are dual graphs [BLN97, Sec. 3]. On the hexagonal lattice, the minimal configurations are in one-to-one correspondence to the set of maximally oriented spanning trees of the triangular sublattice [PK97], which is the dual graph of the hexagonal lattice. As we have now seen that the Torelli theorem from Griffith is tightly connected to minimal configurations, one possible point of research is whether there are connections between minimal configurations and some property of the dual graph for finite graphs in general.

Further research could also be done into extending the results in this thesis to infinite graphs. Minimal configurations are used to prove that a weak limit of the sandpile measure exists on infinite graphs [JW12]. Not a lot of results on the Abelian sandpile model for infinite graphs are known, but the known theory of minimal configurations on infinite graphs might be extended to say something about graph isomorphisms of infinite graphs.

Connections between divisors on curves and divisors of graphs are studied, using among others things techniques from tropical geometry [Bak08; BJ16]. When given a curve C, we can get to the dual graph G using the special fiber C_0. The Baker's specialization lemma states that for a curve C and a K-divisor D on C, we have that

$$rank(D) \leq rank(Trop(D)),$$

where $Trop$ is a mapping from the Picard set of the curve to the Picard set of the dual graph G [Bak08]. The other way around we can use the constructions of metric graphs, which are metric spaces obtained from a discrete graph. Associating divisors on curves and divisors on graphs could allow us to obtain new results by transferring known properties of curves to graphs or the other way around [Jen21].

Appendix A

Prerequisites graph theory

In this chapter, we will go through some graph theoretical definitions that are prerequisites for this thesis. Let us start by defining a graph.

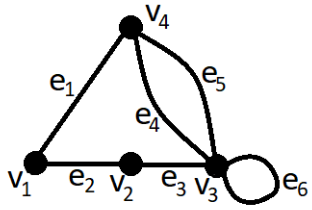
Definition. *A graph G is a finite set of points V called vertices together with a finite set of edges E that each connect to two vertices.*

An edge $e \in E$ is called *incident* to a vertex v if v is one of the two vertices that is connected by the edge. We denote this by $v \in e$. If $v, w \in V$ are two vertices that are connected by an edge, these two vertices are called *neighbours*, which we denote by $v \sim w$. The degree $\deg_G(v)$ of a vertex $v \in V$ is equal to the amount of edges v is incident to.

A graph is called *regular* if the degree of all vertices in the graph is equal. In the Abelian sandpile model, we don't allow *loops*, which are edges that connect a vertex with itself. Usually in this thesis, we will consider only *simple graphs*, which are graphs that have at most one edge in between every two vertices and that have no loops.

The *genus* of a graph is the constant $g(G) = |E| - |V| + 1$.

Example A.1. *An example of a graph $G = (V, E)$ is*



where $V = \{v_1, v_2, v_3, v_4\}$ and $E = \{e_1, e_2, e_3, e_4, e_5, e_6\}$. We see that v_1 is incident to the edges e_1 and e_2. Therefore we have that

$$\deg_G(v_1) = 2.$$

In this graph there is one loop, namely e_6, to which only v_3 is incident. This graph is not simple, as it contains a loop and there are two edges between e_4 and e_5.

An example of two neighbouring vertices are $v_1 \sim v_2$. We see that for instance v_2 and v_4 are not neighbours.

The genus of this graph is $g(G) = 6 - 4 + 1 = 3$. ◊

For a subset $A \subseteq V$, the *outer A-degree* $\text{outdeg}_A(v)$ of $v \in A$ is the amount of edges that v is incident, of which the other vertex incident to this edge is not in A. The *inner A-degree* $\text{indeg}_A(v)$ of $v \in A$ is the amount of edges that v is incident, of which the other vertex incident to this edge is also A. Note that by definition we have

$$\deg_G(v) = \text{indeg}_A(v) + \text{outdeg}_A(v),$$

for all $A \subseteq V$ and $v \in A$.

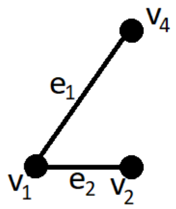
Definition. *A subgraph G_A of G induced by A is a subset $A \subseteq V$ together with all edges $e \in E$ such that both vertices that e is incident to are in A.*

Note that a subgraph is itself a graph. A subgraph of a connected graph is not necessarily connected itself.

Example A.2. *Let us again take a look at the graph of Example A.1. Consider the subset $A = \{v_1, v_2, v_4\} \subset V$. We see that*

$$outdeg_A(v_4) = 2, \quad indeg_A(v_4) = 1.$$

The subgraph $G_{\{v_1,v_2,v_4\}}$ is:



◊

Definition. *A path in $G = (V, E)$ is a sequence vertices*

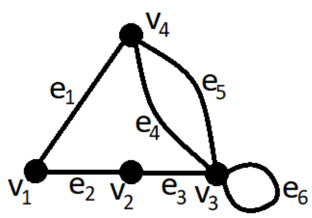
$$v_1 v_2 ... v_n$$

for some $n \in \mathbb{N}$, such that $v_i \in V$ for all $i \in \{1, .., n\}$ and $v_j \sim v_{j+1}$ for all $j \in \{1, ..., n-1\}$. Let $e_1, ..., e_{n-1} \in E$ be edges such that $v_i, v_{i+1} \in e_i$ for all $i \in \{1, ..., n-1\}$. Then $e_1, ..., e_{n-1}$ is called a path as well.

A path is called a cycle if $v_1 = v_n$. A path or cycle is called simple if no vertex is occurring in the path more than once (except for $v_1 = v_n$ in a cycle).

The *length* of a path $v_1 v_2 ... v_n$ is $n - 1$, which can be interpreted as the amount of edges "traveled" in the path. The *distance* between two vertices is the length of the smallest possible path between the two vertices.

Example A.3. *Let us again consider the graph $G = (V, E)$ from the previous example:*



A simple path in this graph is

$$v_4 v_1 v_2.$$

Note that this is also (one of) the shortest path from v_4 to v_2. Therefore, the distance between v_2 and v_4 is 2.

An example of a cycle in this graph is

$$v_4 v_1 v_2 v_3 v_4.$$

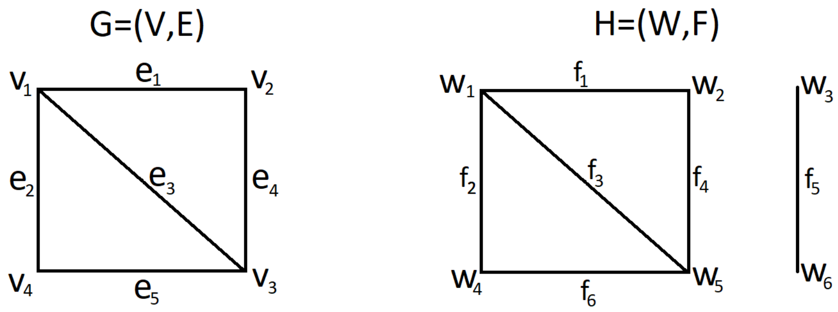
◊

If there is a path between any two vertices in your graph, we call the graph connected.

Definition. *A graph is called connected if for any two vertices $v, w \in V$, there is a path in G that starts at v and ends at w. A graph that is not connected is called disconnected.*

A graph is called *n-connected* if we can remove any $n-1$ vertices from the graph and it will stay connected. A graph is called *n-edge connected* if it is possible to remove any $n-1$ edges from the graph and the graph will stay connected.

Example A.4. *Let us consider the two graphs $G = (V, E)$ and $H = (W, F)$:*



In H there is no path from w_2 to w_3. Therefore H is a disconnected graph.

We see that there is a path between any two vertices in G. For instance, from v_4 we can go to v_2 by the path

$$v_4 v_1 v_2.$$

Therefore we conclude that G is a connected graph.

If we remove any vertex, there still is a path between all of the remaining three vertices. Therefore G is 1-connected. However, if we remove both v_1 and v_3, there is no path anymore from v_2 to v_4. Therefore G is not 2-connected.

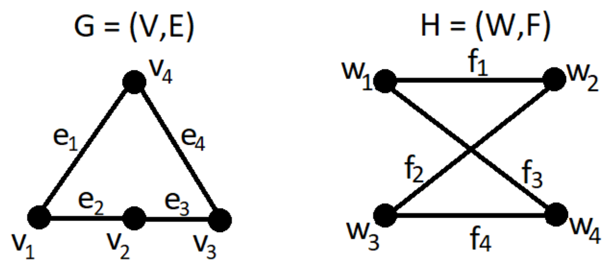
The same way, we can remove any vertex from G and still have a path between any two vertices. However, if we remove both e_1 and e_4, there is no path anymore between v_2 and any other vertex in G. We conclude that G is 1-edge connected but not 2-edge connected. ♢

We also want a way to define two graphs as being the "same". This concept is called a graph isomorphism.

Definition. *Let $G = (V_G, E_G)$ and $H = (V_H, E_H)$ be two graphs. An isomorphism from G to H is a bijection $\phi_V : V_G \to V_H$ together with a bijection $\phi_E : E_G \to E_H$ such that if $v \in e$ for any $v \in V_G$ and $e \in E_G$, then $\phi_V(v) \in \phi_E(e)$.*

Let us look at an example of a graph isomorphism.

Example A.5. *Let us consider the graph $G = (V, E)$ and $H = (W, F)$:*



Define the bijection ϕ to be:

$$\phi_V(v_1) = w_1, \quad \phi_V(v_2) = w_2, \quad \phi_V(v_3) = w_3, \quad \phi_V(v_4) = w_4,$$

and

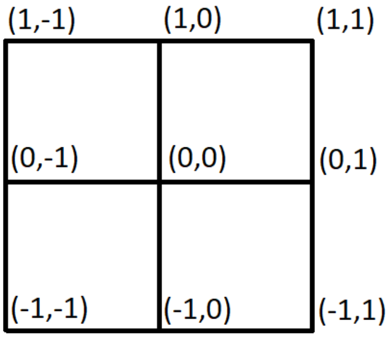
$$\phi_E(e_1) = f_3, \quad \phi_E(e_2) = f_1, \quad \phi_E(e_3) = f_2, \quad \phi_E(e_4) = f_4.$$

We have that ϕ is a graph isomorphism between G and H. Note for instance that $v_1 \in e_2$ and $\phi_V(v_1) = w_1 \in \phi_E(e_2) = f_1$. Note that both graphs are regular, as every vertex in the graphs has degree 2. ◊

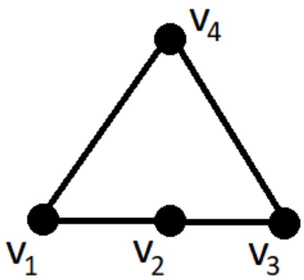
A *lattice graph* $G = (V, E)$ is a graph with $V = \mathbb{Z}^d \cap [-n, n]^d$ for some $n, d \in \mathbb{N}_{>0}$. Two vertices $v_1 = (x_1, y_1), v_2 = (x_2, y_2)$ are adjacent if and only if their Manhattan distance is 1, i.e. when $|x_1 - x_2| + |y_1 - y_2| = 1$. The circle graph $G = (V, E)$ is the graph for which

$$v_1 \sim v_2, \quad v_2 \sim v_3, \quad \dots \quad v_{n-1} \sim v_n, \quad v_n \sim v_1.$$

Example A.6. *The lattice graph with $V = \mathbb{Z}^2 \cap [-2, 2]^2$ is:*



The circle graph with $V = \{v_1, v_2, v_3, v_4\}$ is:



◊

Appendix B

Python code for determining the identity element of the sandpile group

B.1 Cruetz algorithm

This Python code is based on Cruetz[Cre91]. If one wants to look at the model where every site starts at height zero en topples when its height is greater or equal than its degree, one should change the 5 at line 51 to a 4.

```
1 from matplotlib import pyplot as plt
2 
3 #This function will topple the configuration once at site [i,j], it will negatively topple when sign is set to -1
4 def distribute_to_neighbours (configuration, i, j, sign):
5     configuration[i][j] = configuration[i][j] + sign * 4
6     if i != 0:
7         configuration[i-1][j] += sign * -1
8     if i != len(configuration)-1:
9         configuration[i+1][j] += sign * -1
10     if j != 0:
11         configuration[i][j-1] += sign * -1
12     if j != len(configuration[0])-1:
13         configuration[i][j+1] += sign * -1
14     return configuration
15 
16 #This function will find the stabilized version of a configuration
17 def find_stable (configuration, deg):
18     done = False
19     while not done:
20         done = True
21         for i in range(len(configuration)):
22             for j in range(len(configuration[0])):
23                 if configuration[i][j] >= deg:
24                     done = False
25                     configuration = distribute_to_neighbours(configuration, i, j, -1)
26     return configuration
27 
28 #Read input
29 print("For what N to you want to find the NxN neutral, recurrent element?")
30 N = int(input())
31 
32 #Initialize I to be I_0
33 I = [[0 for i in range(N)] for j in range(N)]
```

```
for i in range(N):
    I[0][i] = 1
    I[i][0] = 1
    I[i][N-1] = 1
    I[N-1][i] = 1
I[0][0] = 2
I[0][N-1] = 2
I[N-1][0] = 2
I[N-1][N-1] = 2

previousConfiguration = [[0 for i in range(N)] for j in range(N)]
configuration = I.copy()
while previousConfiguration != configuration:
    for i in range(N):
        for j in range(N):
            previousConfiguration[i][j] = configuration[i][j]
            configuration[i][j] += configuration[i][j]
    configuration = find_stable(configuration, 5)


#Print the result
for i in range(len(configuration)):
    for j in range(len(configuration[0])):
        print(configuration[i][j], end=" ")
    print()
```

B.2 Jacobian algorithm

This Python code is based on the algorithm for finding the recurrent element corresponding to a given equivalence class of the Jacobian in Theorem 2.5. If one wants to look at the model where every site starts at height zero en topples when its height is greater or equal than its degree, one should change the 5 at line 62, 66 and 70 to a 4.

```
1 from matplotlib import pyplot as plt
2
3 #This function will topple the configuration once at [i,j], it will negatively
    topple when sign is set to -1
4 def distribute_to_neighbours (configuration, i, j, sign):
5     configuration[i][j] = configuration[i][j] + sign * 4
6
7     if i != 0:
8         configuration[i-1][j] += sign * -1
9     if i != len(configuration)-1:
10        configuration[i+1][j] += sign * -1
11    if j != 0:
12        configuration[i][j-1] += sign * -1
13    if j != len(configuration[0])-1:
14        configuration[i][j+1] += sign * -1
15
16    return configuration
17
18 #This function will reduce our so that at every point in the sandpile, it will
    have less than 2 * deg
19 def max_degree_double (configuration, deg):
20    done = False
21    while not done:
22        done = True
23        for i in range(len(configuration)):
24            for j in range(len(configuration[0])):
25                if configuration[i][j] > 2 * deg:
26                    done = False
```

```
                    configuration = distribute_to_neighbours(configuration, i, j, -1)
    return configuration

#This function will find the stabilized version of a configuration
def find_stable (configuration, deg):
    done = False
    while not done:
        done = True
        for i in range(len(configuration)):
            for j in range(len(configuration[0])):
                if configuration[i][j] >= deg:
                    done = False
                    configuration = distribute_to_neighbours(configuration, i, j, -1)
    return configuration

#This function will negatively topply any site of our sandpile configuration until for every sight height > deg
def find_maximized_config(configuration, deg):
    done = False
    while not done:
        done = True
        for i in range(len(configuration)):
            for j in range(len(configuration[0])):
                if configuration[i][j] <= deg:
                    done = False
                    configuration = distribute_to_neighbours(configuration, i, j, 1)
    return configuration


#Read input
print("For what N to you want to find the NxN neutral, recurrent element?")
N = int(input())

configuration = [[0 for i in range(N)] for j in range(N)]

#Step 1 of algorithm (reduce)
configuration = max_degree_double(configuration, 5)
print("done step 1")

#Step 2 of algorithm (maximize)
configuration = find_maximized_config(configuration, 5)
print("done step 2")

#Step 3 of algorithm (stabilize)
configuration = find_stable(configuration, 5)
print("done step 3")

#Print the result
for i in range(len(configuration)):
    for j in range(len(configuration[0])):
        print(configuration[i][j], end=" ")
    print()
```

Bibliography

[Bac17] Spencer Backman. “Riemann–Roch theory for graph orientations”. In: *Advances in Mathematics* 309 (2017), pp. 655–691. ISSN: 0001-8708. DOI: https://doi.org/10.1016/j.aim.2017.01.005. URL: https://www.sciencedirect.com/science/article/pii/S0001870817300208.

[Bak08] Matthew Baker. “Specialization of linear systems from curves to graphs”. In: *Algebra & Number Theory* 2.6 (2008), pp. 613–653. DOI: 10.2140/ant.2008.2.613. URL: https://doi.org/10.2140/ant.2008.2.613.

[Big07] Norman Biggs. “The critical group from a cryptographic perspective”. In: *Bulletin of the London Mathematical Society* 39.5 (Aug. 2007), pp. 829–836. ISSN: 0024-6093. DOI: 10.1112/blms/bdm070. URL: https://doi.org/10.1112/blms/bdm070.

[Big99a] Norman Biggs. “Chip-Firing and the Critical Group of a Graph.” In: *Journal of Algebraic Combinatorics* 9 (1999), pp. 25–45. URL: https://doi.org/10.1023/A:1018611014097.

[Big99b] Norman Biggs. “The Tutte polynomial as a growth function”. In: *Journal of Algebraic Combinatorics* 10(2) (1999), pp. 115–133. URL: https://doi.org/10.1023/A:1018748527916.

[BJ16] Matthew Baker and David Jensen. “Degeneration of Linear Series from the Tropical Point of View and Applications”. In: *Nonarchimedean and Tropical Geometry*. Ed. by Matthew Baker and Sam Payne. Cham: Springer International Publishing, 2016, pp. 365–433. ISBN: 978-3-319-30945-3.

[BLN97] Roland Bacher, Pierre de La Harpe, and Tatiana Nagnibeda. “The lattice of integral flows and the lattice of integral cuts on a finite graph.” In: *Bulletin de la Société Mathématique de France* 125.2 (1997), pp. 167–198. DOI: 10.24033/bsmf.2303. URL: http://www.numdam.org/articles/10.24033/bsmf.2303/.

[BN07] Matthew Baker and Serguei Norine. “Riemann–Roch and Abel–Jacobi theory on a finite graph”. In: *Advances in Mathematics* 215.2 (2007), pp. 766–788. ISSN: 0001-8708. DOI: https://doi.org/10.1016/j.aim.2007.04.012. URL: https://www.sciencedirect.com/science/article/pii/S0001870807001454.

[BS13] Matthew Baker and Farbod Shokrieh. “Chip-firing games, potential theory on graphs, and spanning trees”. In: *Journal of Combinatorial Theory, Series A* 120.1 (Jan. 2013), pp. 164–182. ISSN: 0097-3165. DOI: 10.1016/j.jcta.2012.07.011. URL: http://dx.doi.org/10.1016/j.jcta.2012.07.011.

[BTW87] Per Bak, Chao Tang, and Kurt Wiesenfeld. “Self-organized criticality: An explanation of the 1/f noise”. In: *Phys. Rev. Lett.* 59 (4 July 1987), pp. 381–384. DOI: 10.1103/PhysRevLett.59.381. URL: https://link.aps.org/doi/10.1103/PhysRevLett.59.381.

[CL03] Robert Cori and Yvan Le Borgne. “The sand-pile model and Tutte polynomials”. In: *Advances in Applied Mathematics* 30.1 (2003), pp. 44–52. ISSN: 0196-8858. DOI: https://doi.org/10.1016/S0196-8858(02)00524-9. URL: https://www.sciencedirect.com/science/article/pii/S0196885802005249.

[Cre91] Michael Creutz. “Abelian sandpiles”. In: *Nuclear Physics B - Proceedings Supplements* 20 (1991), pp. 758–761. ISSN: 0920-5632. DOI: https://doi.org/10.1016/0920-5632(91)91016-D. URL: https://www.sciencedirect.com/science/article/pii/092056329191016D.

[CV10] Lucia Caporaso and Filippo Viviani. “Torelli theorem for graphs and tropical curves”. In: *Duke Mathematical Journal* 153.1 (May 2010). ISSN: 0012-7094. DOI: 10.1215/00127094-2010-022. URL: http://dx.doi.org/10.1215/00127094-2010-022.

[Dha90] Deepak Dhar. “Self-organized critical state of sandpile automaton models”. In: *Phys. Rev. Lett.* 64 (14 Apr. 1990), pp. 1613–1616. DOI: 10.1103/PhysRevLett.64.1613. URL: https://link.aps.org/doi/10.1103/PhysRevLett.64.1613.

[Gri23] Sarah Griffith. “A Torelli theorem for graph isomorphisms”. In: *Journal of Combinatorial Theory, Series A* 198 (Aug. 2023), p. 105764. ISSN: 0097-3165. DOI: 10.1016/j.jcta.2023.105764. URL: http://dx.doi.org/10.1016/j.jcta.2023.105764.

[HHM08] John M. Harris, Jeffry L. Hirst, and Michael J. Mossinghoff. *Combinatorics and graph theory*. Undergraduate Texts in Mathematics. Springer New York, 2008. DOI: 10.1007/978-0-387-79711-3.

[Jár18] Antal A. Járai. *Sandpile models*. 2018. arXiv: 1401.0354 [math.PR].

[Jen21] David Jensen. “Chip Firing and Algebraic Curves”. In: *Notices of the American Mathematical Society* 68 (Dec. 2021), pp. 1875–1881. DOI: 10.1090/noti2378.

[JW12] Antal A. Járai and Nicolás Werning. “Minimal Configurations and Sandpile Measures”. In: *Journal of Theoretical Probability* 27.1 (Aug. 2012), pp. 153–167. ISSN: 1572-9230. DOI: 10.1007/s10959-012-0446-z. URL: http://dx.doi.org/10.1007/s10959-012-0446-z.

[MD92] S.N. Majumdar and Deepak Dhar. “Equivalence between the Abelian sandpile model and the q→0 limit of the Potts model”. In: *Physica A: Statistical Mechanics and its Applications* 185.1 (1992), pp. 129–145. ISSN: 0378-4371. DOI: https://doi.org/10.1016/0378-4371(92)90447-X. URL: https://www.sciencedirect.com/science/article/pii/037843719290447X.

[MD99] S.N. Majumdar and Deepak Dhar. “Height correlations in the Abelian sandpile model”. In: *Journal of Physics A: Mathematical and General* 24 (Jan. 1999), p. L357. DOI: 10.1088/0305-4470/24/7/008.

[Mil86] J.S. Milne. “Jacobian Varieties”. In: *Cornell, G., Silverman, J.H. (eds) Arithmetic Geometry*. Springer, New York, NY, 1986. URL: https://doi.org/10.1007/978-1-4613-8655-1_7.

[MWZ19] F. Meng, L.N.Y. Wong, and H. Zhou. “Power law relations in earthquakes from microscopic to macroscopic scales.” In: *Scientific Reports* 9.10705 (2019). URL: https://doi.org/10.1038/s41598-019-46864-8.

[Nor97] J. R. Norris. *Markov Chains*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 1997. DOI: 10.1017/CBO9780511810633.

[PK97] V. B. Priezzhev and Dmitri V. Ktitarev. “Minimal Sandpiles on Hexagonal Lattice”. In: *Journal of Statistical Physics* 88 (1997), pp. 781–793. URL: https://doi.org/10.1023/B:JOSS.0000015171.16291.ae.

[Red05] F. Redig. *Mathematical aspects of the abelian sandpile model.* June 2005. URL: https://interacting.math.cnrs.fr/FR_sandpilelectures.pdf.

[Sho10] Farbod Shokrieh. "The monodromy pairing and discrete logarithm on the Jacobian of finite graphs". In: *Journal of Mathematical Cryptology* 4.1 (Jan. 2010). ISSN: 1862-2984. DOI: 10.1515/jmc.2010.002. URL: http://dx.doi.org/10.1515/JMC.2010.002.

[SSS18] Thomas Selig, Jason P. Smith, and Einar Steingrímsson. "EW-tableaux, Le-tableaux, tree-like tableaux and the Abelian sandpile model". English. In: *The Electronic Journal of Combinatorics* 25.3 (July 2018). ISSN: 1077-8926. DOI: 10.37236/7480.

[SW10] Yi Su and David G. Wagner. "The lattice of integer flows of a regular matroid". In: *Journal of Combinatorial Theory, Series B* 100.6 (2010), pp. 691–703. ISSN: 0095-8956. DOI: https://doi.org/10.1016/j.jctb.2010.07.003. URL: https://www.sciencedirect.com/science/article/pii/S0095895610000742.