AN INTRODUCTION TO SANDPILES

JOSEF ELYOUSSOUFI

ABSTRACT. Sandpiles are graphs with pieces of "sand" put on the vertices that follow certain simple rules. Sandpiles were first introduced by Bak, Tang, and Wiesenfeld as an example of self-organized criticality. They help illustrate how very simple rules can lead to complexity. This paper will establish the definition of a sandpile, the properties of sandpiles, and how complexity can come from them.

INTRODUCTION

The Abelian Sandpile Model was first discussed by Bak, Tang, and Wiesenfeld in 1987 as an example of self-organized criticality [BTW87]. Self-organized criticality is a property of systems that creates complexity through very simple interactions. This is important because it is a way of creating complexity in a completely natural way, thus making it a plausible candidate to explain complexity in nature. Since being originally discovered, self-organized criticality has appeared all over science and even economics.

Despite the scientific importance of self-organized criticality, this paper will focus more on the mathematics of the Abelian Sandpile Model. Since Bak, Tan, and Wiesenfeld's original paper, the Abelian Sandpile model has been heavily studied though even now there are many open-ended questions about it. The goal of this paper is to understand the definition of a sandpile and learn many of its properties to eventually lead us to how these properties lead to self-organized criticality.

The first section will introduce the rules that define a sandpile as a connected graph with a globally accessible vertex called the sink. It will then introduce important mathematical notation for sandpiles and rules of vertex-firing. The second section will then go over many of the important sandpile properties and their proofs.

The third section will then define sandpile addition, define the rules that make a sandpile recurrent, and discuss the group that is formed by the set of recurrents. The fourth section will look at grid representations of sandpiles. This is to help visually illustrate sandpiles and show the the complex fractal patterns that emerge from the identity element of recurrents. This section will help illustrate miraculous power of self-organized criticality and the beautifully complex shapes that can come from such a simply defined system.

ACKNOWLEDGMENTS

The author would like to thank Simon Rubinstein-Salzedo and Rajiv Nelakanti.

1. SANDPILES

We begin by defining the rules that make something a *sandpile*. Let's start by taking a look at this graph:

Date: July 11, 2022.

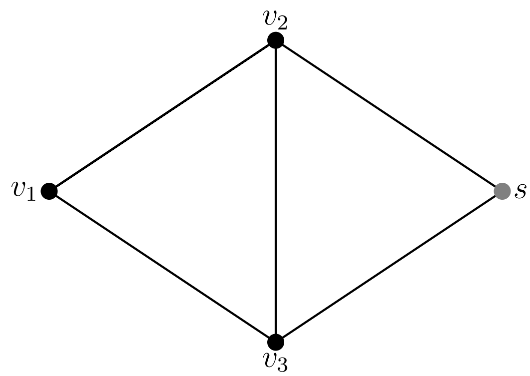


Figure 1.1

Sandpiles have *chips* or *sand* that are put on top of a vertex. Figure 1.2 shows a sandpile where vertex v_1 and v_2 have two pieces of sand on them.

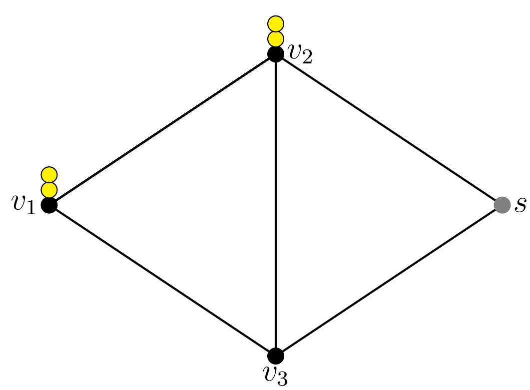


Figure 1.2

Now, we introduce the rules that define a sandpile. A vertex is considered *unstable* if the amount of sand at the vertex is greater or equal to the vertex's *degree*. The degree of a vertex is the number of edges coming out of that vertex. v_1 has degree 2 and 2 pieces of sand, thus it is unstable. v_2 has degree 3 and 2 pieces of sand, thus it is *stable*. When a vertex is unstable, we *fire* or *topple* sand to its neighbors. For example, because v_1 is unstable it will lose 2 pieces of sand: one that is given to v_2 and the other to v_3.

Another part of the sandpile is the *sink* represented by the s vertex. The sink vertex takes sand and removes it from the graph. The importance of the sink variable is to make sure we don't have configurations where sand topples infinitely.

Our goal is to make sure a sandpile is completely stable. A sandpile is unstable when there is at least one vertex that is unstable. We *stabilize* a sandpile by toppling all of the unstable vertices until the remaining vertices are stable. The resulting sandpile after this process is called the *stabilization* of the original sandpile. Let's now look at the stabilization of the sandpile in Figure 1.2:

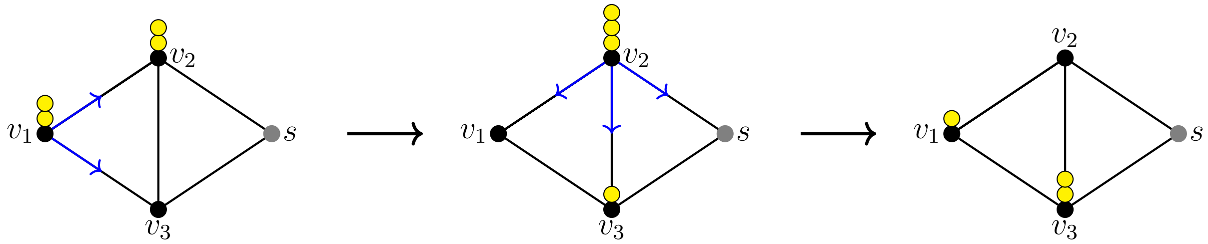


Figure 1.3

1.1. **Sandpile Graphs.** We previously described the rules of sandpiles through an example, but this section will focus more on defining the graph that describes sandpiles.

Definition 1.1. A *sandpile graph* is a triple $G = (V, E, s)$ where (V, E) is a graph where V describes the vertices and E describes the edges. s represents the sink that we defined earlier.

The reason we define the sink along with (V, E) is because we require that the sink is *globally accessible*. What this means is that for any $v \in V \setminus \{s\}$, there is a path that connects v to the sink. Let's define another term for $V \setminus \{s\}$ (vertices not including the sink) to help us.

Definition 1.2. We define $\widetilde{V}$ as the set of all non-sink vertices.

$$\widetilde{V} := V \setminus \{s\}.$$

We describe a *configuration* of sand on a graph G is an element of a group on its non-sink vertices. What this means is that a configuration of sand is a specific way to put sand on top of a graph G.

Definition 1.3. We define the group of all configurations as $\text{Config}(G, s)$.

$$\text{Config}(G, s) := \mathbb{Z}\widetilde{V} := \left\{ \sum_{v \in \widetilde{V}} c(v)v : c(v) \in \mathbb{Z} \text{ for all } v \right\}.$$

$c(v)$ for a configuration c is the amount of sand at vertex v. The zero of $\text{Config}(G, s)$ is the configuration 0 where $c(v) = 0$ for all $v \in \widetilde{V}$ (there are zero amounts of sand at every vertex). Notice that we defined $c(v) \in \mathbb{Z}$, thus we are allowing there to be negative amounts of sand on a vertex. This is important in defining $\text{Config}(G, s)$ as a group (because then every sandpile would have an inverse), but we will be mostly ignoring configurations that have negative numbers of sand at any of the vertices.

Definition 1.4. We define the *degree* of a configuration to be the number of total sand on the configuration:

$$\deg(c) = \sum_{v \in \widetilde{V}} c(v).$$

.

If we define that $a \leq b$ where a and b are configurations, what we mean is that for all $v \in \widetilde{V}$, we have that $a(v) \leq b(v)$.

Definition 1.5. We define a *sandpile* to be a configuration c where $c \geq 0$.

We will now define *toppling* and *firing* within our new definition of a sandpile. Recall that a toppling of vertex v happens when that vertex is *unstable*. A vertex is unstable when $c(v) \geq \deg(v)$, meaning the amount of sand on v is greater or equal to the number of edges stemming from v. When a vertex is unstable, a piece of sand is given to each of its neighbors. We can define a toppling mathematically to fit our definition of a configuration.

Definition 1.6. We let c be a configuration on a graph G and vertex v such that $v \in \widetilde{V}$. A toppling of v will produce a new configuration c' that is define as

$$c' = c - \deg(v)v + \sum_{vw \in E, w \neq s} w.$$

This can be alternatively written as

$$c \xrightarrow{v} c'$$

What this means is that the vertex v loses $\deg(v)$ pieces of sand, while all of the w non-sink vertices that share an edge with v, are given one piece of sand. For these equations, we signify adding x amount of sand to a vertex v by writing $c' = c + xv$. Removing x amount of sand from v would be described as $c' = c - xv$.

We say that a firing of vertex v is *legal* if vertex v is unstable. We define a *firing sequence* to be a list of vertices that are fired in the order of the list. For example, the firing sequence for the stabilization in Figure 1.3 would be v_1, v_2.

Definition 1.7. A legal firing sequence is a sequence of vertex firings. A firing sequence $v_1, v_2, \cdots, v_n$ for configuration c would look like

$$c \xrightarrow{v_1} c_2 \xrightarrow{v_2} c_3 \xrightarrow{v_3} \cdots \xrightarrow{v_n} c_n.$$

A legal firing sequence where the resulting configuration is stable is said to be *stabilizing* for that configuration. There is a unique stabilization for each configuration as we will show later.

1.2. **The Toppling Matrix.** While we won't use it much in this paper, it can be incredibly important to understand how sandpiles can be visualised as vectors. Our main tool to do this is the Laplacian for a graph G.

Definition 1.8. The Laplacian L of a graph $G = (V, E)$ is defined as

$$L = D - A$$

where D is the *degree matrix* and A is the *adjacency matrix*.

Definition 1.9. The *degree matrix* of a graph $G = (V, E)$ with n vertices is an $n \times n$ matrix D where

(1) for all $i \in V$, $D_{ii} = \deg(i)$
(2) for all $i, j \in V$, $D_{ij} = 0$ if $i \neq j$.

Definition 1.10. The *adjacency matrix* of a graph $G = (V, E)$ with n vertices is an $n \times n$ matrix A where for all $i, j \in V$:

(1) if $ij \in E$, $D_{ij} = 1$
(2) $D_{ij} = 0$ otherwise

Let's look at the graph in Figure 1.1 and find its Laplacian (ordering the vertices as v_1, v_2, v_3, s) given our newfound formula:

$$D = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix}$$

$$A = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

$$L = D - A = \begin{bmatrix} 2 & -1 & -1 & 0 \\ -1 & 3 & -1 & -1 \\ -1 & -1 & 3 & -1 \\ 0 & -1 & -1 & 2 \end{bmatrix}$$

The importance of the Laplacian when it comes to understanding sandpiles can only be demonstrated by defining the *reduced Laplacian.*

Definition 1.11. The reduced Laplacian $\widetilde{L}$ for a graph G with n vertices is an $n-1 \times n-1$ matrix which is defined as the Laplacian L of that graph G where the row and column representing the sink vertex is removed.

For example, we can say that the reduced Laplacian for the graph in Figure 1.1 is:

$$\widetilde{L} = \begin{bmatrix} 2 & -1 & -1 \\ -1 & 3 & -1 \\ -1 & -1 & 3 \end{bmatrix}.$$

What makes the reduced Laplacian so important is that it can be used to define any firing sequence. The way this works if that we can define a sandpile as a vector v where the nth entry corresponds to the amount of sand v_n. We can show the vector for the sandpile in Figure 1.2:

$$v = \begin{bmatrix} 2 \\ 2 \\ 0 \end{bmatrix}.$$

We can define a toppling of this sandpile using the Laplacian. It's easiest to show this through an example, so let's fire v_1 and show how we do this:

$$\begin{aligned} v' &= v - \widetilde{L} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} 2 \\ 2 \\ 0 \end{bmatrix} - \begin{bmatrix} 2 & -1 & -1 \\ -1 & 3 & -1 \\ -1 & -1 & 3 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} 2 \\ 2 \\ 0 \end{bmatrix} - \begin{bmatrix} 2 \\ -1 \\ -1 \end{bmatrix} \\ &= \begin{bmatrix} 0 \\ 3 \\ 1 \end{bmatrix} \end{aligned}$$

As we know from Figure 1.3, we can stabilize the graph by firing v_1 and the v_2. We can do this in one fell swoop:

$$\begin{aligned} v' &= v - \widetilde{L} \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} 2 \\ 2 \\ 0 \end{bmatrix} - \begin{bmatrix} 2 & -1 & -1 \\ -1 & 3 & -1 \\ -1 & -1 & 3 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} 2 \\ 2 \\ 0 \end{bmatrix} - \begin{bmatrix} 1 \\ 2 \\ -2 \end{bmatrix} \\ &= \begin{bmatrix} 1 \\ 0 \\ 2 \end{bmatrix} \end{aligned}$$

This is the exact same sandpile that we got from the firings in Figure 1.3. The reason the reduced Laplacian can do this comes from its structure. While the definition $L = D - A$ is the standard one, we can define the Laplacian like this to better understand why this works:

(1) For all $i \in V$, $L_{ii} = \deg(i)$
(2) For all $i, j \in E$, $L_{ij} = -1$

So what is happening when you fire a vertex i, we you are removing $\deg(i)$ amount of sand at the ith entry and adding 1 to all the other entries that are connected to i with an edge, which is the exact same thing as firing. Because of the distributive property of matrix multiplication, just by showing that it works for one firing shows that the reduced Laplacian will work for firing sequences (because you can do multiple single firings then distribute out the reduced Laplacian and add the vectors).

2. Sandpile Properties

This section will deal with proving many important properties of Sandpiles. We will first show that vertex firings commute.

Theorem 2.1. *If c' is the resulting configuration after the firing of v_1, v_2 and c'' is the resulting configuration after the firing of v_2, v_1, we know c_1 and c_2 are equivalent.*

Proof. This stems naturally from our definition of a vertex firing from earlier. Let's take the two firing scripts v_1, v_2 and v_2, v_1 and use our equation for a vertex firing for each. Here's the resulting configuration after firing v_1, v_2 :

$$c \xrightarrow{v_1} c' = c - \deg(v_1)v_1 + \sum_{v_1 w \in E, w \neq s} w$$
$$c' \xrightarrow{v_1, v_2} c''' = c - \deg(v_1)v_1 - \deg(v_2)v_2 + \sum_{v_1 w \in E, w \neq s} w + \sum_{v_2 u \in E, u \neq s} u.$$

Doing the same thing for v_2, v_1 will give the same exact result as shown:

$$c \xrightarrow{v_2} c'' = c - \deg(v_2)v_2 + \sum_{v_2 w \in E, w \neq s} w$$
$$c' \xrightarrow{v_2, v_1} c'''' = c - \deg(v_2)v_2 - \deg(v_1)v_1 + \sum_{v_2 w \in E, w \neq s} w + \sum_{v_1 u \in E, u \neq s} u.$$

While the variables are switched around, these are all the same exact sums, thus showing that both firing sequences have the exact same effects on the same variables.

■

One thing noticeable about the theorem is that it doesn't specify that the firings have to be legal. If we allow for negative amounts of sand, it doesn't really matter if a vertex firing is legal or not. This is shown in Figure 2.1

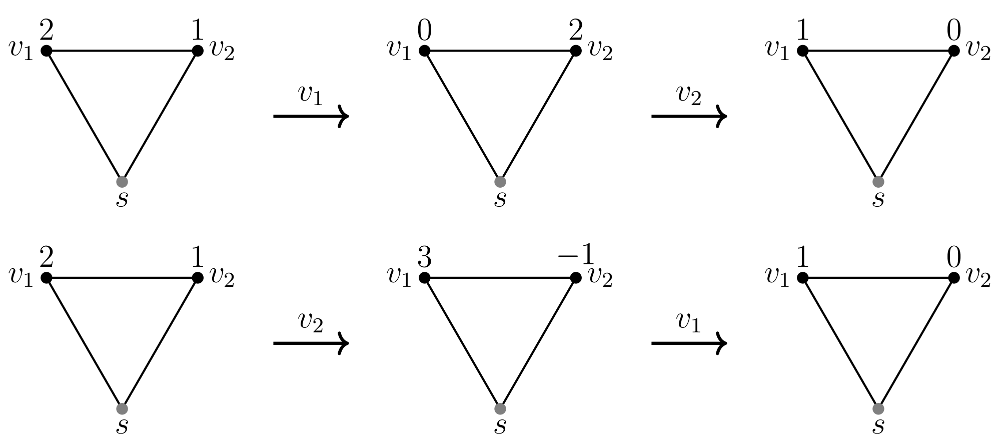


Figure 2.1

2.1. **The Least Action Principle.** We will now prove one of the most important properties of sandpiles, the *least action principle.*

Theorem 2.2 (Least Action Principle)**.** *Let $x = x_1, \cdots, x_k$ and $y = y_1, \cdots, y_n$ be firing sequences for configuration c where x is a legal firing and y is stabilizing. Then we have that $k \leq n$ and for all x_i, there exist some y_j such that $x_i = y_j$.*

Proof. We will solve this using induction. We begin with the first term x_1. Because we know that x is a legal firing sequence, we know that $c(x_1) \geq \deg(x)$. Notice that the number of sand at a vertex can only decrease if it is toppled. Thus, because y stabilizes, we know that there must exist some y_j such that $y_j = x_1$. Because multiple firings are commutative, we know that the firing sequence $y_j, y_1, \cdots, y_{j-1}, y_{j+1}, \cdots y_n$ is stabilizing. Thus, we have that the sequence $y_1, \cdots, y_{j-1}, y_{j+1}, \cdots y_n$ must be stabilizing for the configuration $c_2 = c - -\deg(x_1)x_1$. Because x is legal, that means $x_2, \cdots, x_k$ must be legal for c_2. We finish the proof by continuing our method for $c_2, \cdots, c_k$. Because there is a y_k for each x_i, we know that $n \geq k$. ■

One of the important results that comes directly from the Least Action Principle is the property that all legal stabilizing sequences are permutations of each other.

Theorem 2.3 (Uniqueness of Stabilization)**.** *Let c be a configuration where σ and σ' are firing sequences. Suppose that $c \xrightarrow{\sigma} \overline{c}$ and $c \xrightarrow{\sigma'} \overline{c}'$, where both $\overline{c}$ and $\overline{c}'$ are stable. Then we must have that $\overline{c} = \overline{c}'$ and $\sigma = \sigma'$.*

Proof. This is proven very quickly by applying the least action principle to ω and ω' twice. Once where we consider ω to be the stabilizing sequence and the other where we consider ω' the stabilizing sequence. ■

2.2. **Proof of Stabilization Existence.** Earlier, we said we defined the sink to make sure that there were no sandpiles without a stabilization such as Figure 2.2. While it makes sense that a *globally accessible* sink would mean that every sandpile has a stabilization, we have yet to prove this fact.

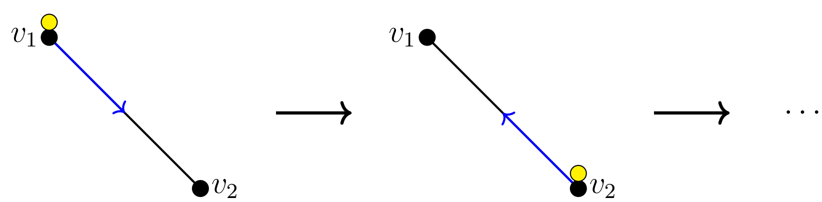


Figure 2.2

Our first step to proving that every sandpile has a stabilization is to define an *ordering* for configurations of a certain graph. This ordering will allow us to compare two configurations similar to how we can compare numbers a and b by saying $a < b$.

Definition 2.4. Let $g_1, \cdots, g_n$ be an ordering of the vertices for a graph $G = (V, E, s)$ where $i < j$ if $d(s, u_i) < d(s, u_j)$ (i is closer to the sink than j). We define the *sandpile ordering* $\prec$ of the configurations of G as follows. Let's take sandpile configurations a and b equal to $(a_{g_1}, \cdots, a_{g_n})$ and $(b_{g_1}, \cdots, b_{g_n})$ We have $a \prec b$ if

(1) $\deg(a) < \deg(b)$, or
(2) $\deg(a) = \deg(b)$ and $a_{g_k} - b_{g_k} > 0$ for the smallest k such that $a_{g_k} - b_{g_k} \neq 0$.

Another way to explain this is that a configuration is less than another if it has less sand or a has more sand at the vertex closest to the sink where a and b don't have the same amount of sand. This ordering is very important because it has a lot of very important properties. One of them that we will use is described below.

Property 2.5. *If c is a configuration with $0 \leq c$, there are finitely many configurations c' where $0 \leq c$ and $c' \preceq c$.*

While we won't do a rigorous proof for this property, it stems from the fact that c has a finite degree and finite number of vertices. Another property about $\prec$ is proven below.

Lemma 2.6. *Let $\prec$ be a sandpile ordering and c, c' be sandpile configurations of graph G. If $c \to c'$ through a vertex firing, then we have that $c' \prec c$.*

Proof. Let's say that the vertex we fired is vertex v. There are two options for this vertex, it is either next to a sink or not next to a sink.

(1) If it is next to a sink, then c will lose sand from the vertex firing, thus we have $c' \prec c$ because $\deg(c') < \deg(c)$.
(2) Now, let's look at the case where v is not next to a sink. Because v isn't next to a sink, we have that $\deg(c) = \deg(c')$ thus we will have to use the second condition to show $c' \prec c$. Because we know that the sink is globally accessible, there must be a path from v to s, which means that one of v's neighbors must be closer to s than v because it is part of the path from v to s. Let m be the vertex neighboring v that is closest to the sink, thus we have $c'_m > c_m$. Because toppling v only affects its neighbors, we have that $c'_{g_k} - c_{g_k} = 0$ until $g_k = m$, thus we have $c' \prec c$.

■

Theorem 2.7 (Existence of Stabilization)**.** *Every configuration has a stabilization (which is unique by Theorem 2.3).*

Proof. For sandpile configurations ($0 \leq c$), this is very natural. There are finitely many sandpiles that are $\prec c$, and every single legal firing is $\prec c$, thus there must be a finite toppling sequence that stabilizes c.

However, this doesn't completely solve our problem if we include configurations with vertices that have negative values. Accounting for these configurations though can be done by introducing a configuration c^+ that is defined as:

$$c^+(v) = \max\{0, c(v)\}.$$

This new configuration is a sandpile ($0 \leq c$) because all of the negative pieces of sand are replaced with a 0 vertex. As we've shown c^+ is stabilizable. Because c^+ is stabilizable, by the Least Action Principle (Theorem 2.2), we have that all legal firing sequences are finite. Because every legal firing sequence for c is legal for c^+, we know that every legal firing sequence for c has to be finite. Thus, it must stabilize because you cannot fire forever. ■

Our final proof is an even more powerful version of Theorem 2.3 that we can show now that we have shown the existence of stabilization for every configuration c.

Theorem 2.8. *Let c be a configuration, and σ and τ be legal firing sequences that result in the same configuration c':*

$$c \xrightarrow{\sigma} c' \quad \textit{and} \quad c \xrightarrow{\tau} c'.$$

Then, we have that σ and τ are rearrangements of each other.

Proof. By Theorem 2.7, we know that there is a firing sequence α that stabilizes c'. Thus, the firing sequences σ, α and τ, α must be stabilizing for c. Then, by Theorem 2.3 we have that σ, α and τ, α are rearrangements of each other. Because α is the same sequence in both of those firing sequences, we have then that σ and τ are rearrangements of each other. ∎

This final proof is a reason that people refer to sandpiles as *Abelian sandpiles*. The commutative property of sandpiles where the order of firings doesn't matter is where the Abelian comes from.

3. Recurrent Sandpiles

We now will look special kinds of sandpiles called *recurrent sandpiles* and the fascinating results we get by examining these special kinds of sandpiles. Before we get to defining recurrent sandpiles, we will take a look at *sandpile addition*.

3.1. **Sandpile Addition.**

Definition 3.1. The *stable addition* of two sandpiles a and b, written as $a \oplus b$ denotes the stabilization of $a + b$:

$$a \oplus b = (a + b)^\circ,$$

where $(a + b)^\circ$ represents the stabilization of $a + b$.

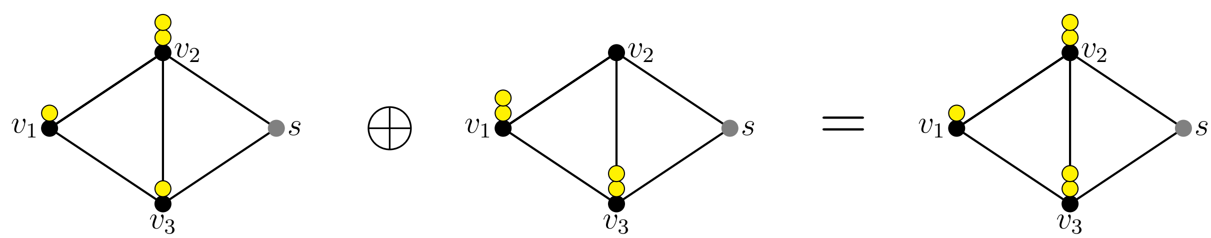


Figure 3.1

Stable addition is commutative because it's really just addition on a certain number of vertices (remember the vector representation of a sandpile). Proving it is associative is slightly more complicated and requires some of our past results.

Theorem 3.2. *Stable addition is associative.*

Proof. This theorem stems from the uniqueness of stabilization (Theorem 2.3) that we proved earlier. Let's take the three sandpiles a, b, c. Because c is a sandpile ($0 \leq c$), we know that $(a + b)^\circ$ must be legal for $a + b + c$. Thus we have:

$$a + b + c \to (a + b)^\circ + c \to ((a + b)^\circ + c)^\circ \to (a \oplus b) \oplus c.$$

We can use the exact same logic and stabilize $b + c$ first:

$$a + b + c \to a + (b + c)^\circ \to (a + (b + c)^\circ)^\circ \to a \oplus (b \oplus c).$$

By the uniqueness of stabilization, we have that:

$$(a + b + c)^\circ = (a \oplus b) \oplus c = a \oplus (b \oplus c).$$

■

Definition 3.3. The *sandpile monoid* is the set of all stable sandpile configurations for a graph G with the operation of stable addition.

This monoid is basically a group, however there are no inverses. There are no ways to go from two non-zero sandpiles to get back to the identity sandpile. We will introduce recurrent sandpiles in the next section to help fix this problem and highlight some fascinating properties of sandpiles.

3.2. **Recurrent Sandpiles.**

Definition 3.4. We define a *recurrent* configuration for a graph G to be a configuration c where:

(1) $c \geq 0$,
(2) c is stable,
(3) For any configuration a, there is another configuration $b \geq 0$ such that $c = (a+b)^\circ$.

Now that we have defined a recurrent sandpile, let's start by finding one. One sandpile that is inherently recurrent no matter what graph we have is the *maximal stable configuration* defined below.

Definition 3.5. The *maximal stable configuration* for a graph G is a sandpile defined as:

$$c_{\max} = \sum_{v \in \widetilde{V}} (\deg(v) - 1)\ v.$$

You can visualize the maximal stable configuration as the stable configuration with the most amount of sand possible on each vertex. Figure 3.1 shows the maximal stable configuration for the sandpile for the graph we defined in Figure 1.1.

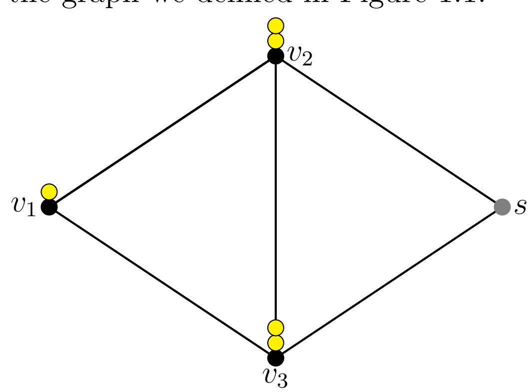


Figure 3.2

Two properties of $c_{\max}$ that are immediately recognizable is that it's stable and all other stable configurations c are smaller or equal to $c_{\max}$. These important realizations will help us show that $c_{\max}$ is a recurrent configuration.

Theorem 3.6. *The maximal stable configuration $c_{\max}$ for a graph G must be recurrent.*

Proof. The maximal stable configuration $c_{\max}$ follows the first two conditions to be a recurrent sandpiles just from its definition. The degree of a vertex v in a connected graph must be great than 0, thus $\deg(v) - 1$ will always be a non-negative number therefore $c \geq 0$. Also,

$c_{\max}$ must be stable because every vertex v has $\deg(v)-1$ amount of sand, which is less than its degree.

Now, the third condition takes slightly more work to prove. Let's first look at *stable* sandpiles a. By the definition of the maximal stable configuration, we have that $a \leq c_{\max}$, thus we can define a sandpile b, where $b(v) = c_{max}(v) - a(v)$. By this definition, we have that $a + b = c_{\max}$ and that $b \geq 0$. We can now extend this to all configurations in general by just stabilizing them before finding b using the method above. This works because $(a+b)^\circ = (a^\circ + b)^\circ$ because of the uniqueness of stabilization (Theorem 2.3). ■

Finding a configuration that is inherently recurrent is a lot more important than one would think. Due to the properties of recurrent sandpiles, we can define all of the recurrent sandpiles just by using $c_{\max}$.

Theorem 3.7. *A configuration c is recurrent if and only if there exists a configuration $b \geq 0$ such that $c = (c_{\max} + b)^\circ$*

Proof. The forward direction of the theorem is just property three of recurrent sandpiles, and we already know that $c_{\max}$ is recurrent. The reverse direction also stems from property three. We know from our initial condition that $c = (c_{\max} + b)^\circ$. Let's take an arbitrary configuration a. Because $c_{\max}$ is recurrent, we know that there exists sandpile d such that:

$$c_{\max} = (a + d)^\circ.$$

We will now stable add b to both side and use the uniqueness of a stabilization to finish the proof:

$$\begin{aligned} c_{\max} \oplus b &= (a+d)^\circ \oplus b \\ (c_{\max} + b)^\circ &= ((a+d)^\circ + b)^\circ \\ c &= ((a+d)^\circ + b)^\circ \\ &= (a + b + d)^\circ. \end{aligned}$$

Now we know that for any configuration a, we can find another positive configuration ($b + d$ in the example) such that a plus that configuration stabilizes to c. The first two conditions are automatically resolved by our definition of c as the *stable* addition of $c_{\max}$ and b, which are both greater than 0. Thus we have that c is recurrent, completing the proof. ■

We can extend this proof using the same logic to all recurrent sandpiles instead of just $c_{\max}$ but that would be unnecessary considering using that property would require us to find another recurrent sandpile that isn't $c_{\max}$, which would be a waste of time.

Theorem 3.8. *The set of all recurrents under stable addition is an Abelian group.*

Proving this is beyond the scope of this paper, but the fact that the set of recurrents is a group is going to be the basis for the rest of the paper. A group is defined as a set with an associative operation $*$ where there is an identity element e where for all elements g we have $e * g = g = g * e$. Also, every element must have an inverse g^{-1} where $g * g^{-1} = e$.

For the set of all sandpiles with stable addition as the operation, we almost satisfy the properties that make a group. We have an identity element, the zero sandpile, and an associative operation, $\oplus$, but we do not have inverses. The recurrents, however, do have an identity sandpile which satisfies all of the properties we need.

Definition 3.9. The *identity sandpile* for the set of recurrents is:

$$(2c_{\max} - (2c_{\max})^{\circ})^{\circ}.$$

Again, proving this is outside of the scope of this paper. You can find proofs for Theorem 3.8 and Definition 3.9 in [Per16]. We can use this formula to find the identity sandpile for any sandpile graph. Using it for the graph in Figure 1.1, gives us the sandpile in Figure 3.3.

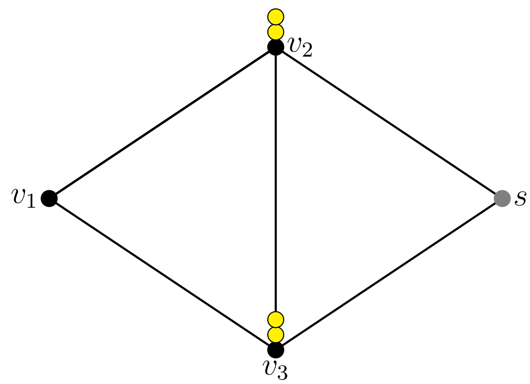


Figure 3.3

We can illustrate how the identity sandpile acts as a zero by adding it to c_{max}. This is shown in Figure 3.4 below.

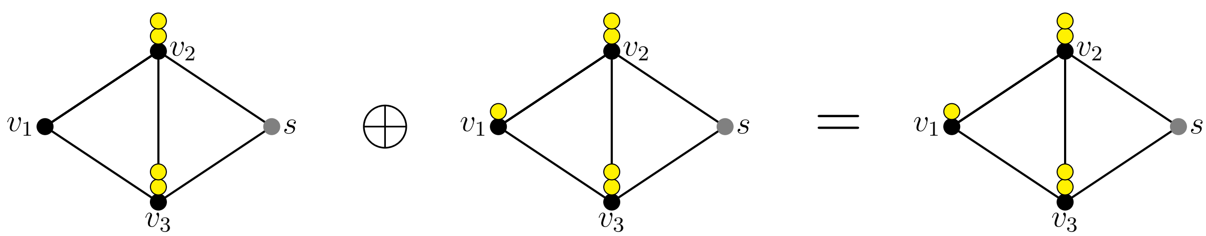


Figure 3.4

The identity sandpile for a graph has a lot of fascinating properties that can fully appreciated if viewed as an image. We will show this in the following section.

4. Grid Sandpile Graphs

This section will use a specific kind of sandpile graph that is slightly different from the ones we've been using so far. What we will be using are *sandpile grid graphs*, which are important for visualizing sandpiles. Using grids as graphs allows us to view each grid as pixels, where the amount of sand on each vertex represents a color for example.

Definition 4.1. The $m \times n$ *sandpile grid graph* has nonsink vertexes defined as:

$$\widetilde{V} = \{(i, j) \in \mathbb{Z}^2 : 1 \leq i \leq m, 1 \leq j \leq n\}$$

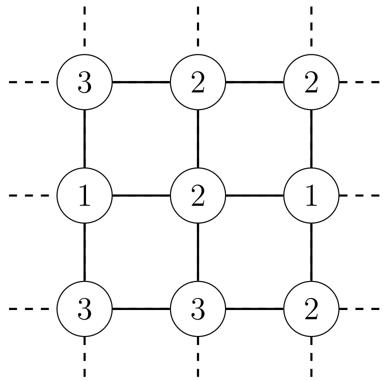


Figure 4.1

Figure 4.1 illustrates a graph a 3×3 sandpile grid graph. We have yet to define what the sink is for grid graphs. The sink is connected once to all of the vertices on the edge on the boundary other than the corners and twice to the corners. Edges connecting to the sink are represented as dashed lines in Figure 4.1.

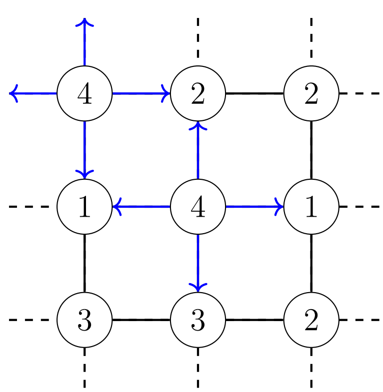


Figure 4.2

Figure 4.2 helps illustrate what a vertex firing looks like using grid graph. What we are interested in however, are the identity elements for these sandpiles. The identity element for the 3×3 grid is shown below in Figure 4.3.

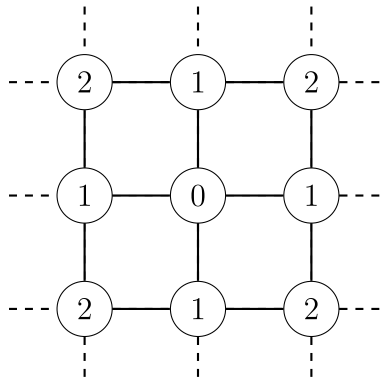


Figure 4.3

There are some interesting symmetries in the identity matrix, but as I said earlier, we can treat each vertex as a ”pixel” in grid graphs. If we assign colors to each amount of sand, for example yellow for one piece of sand, we can create colorful representations of sandpiles. Figure 4.4 shows this for the identity of the 3×3 identity sandpile.

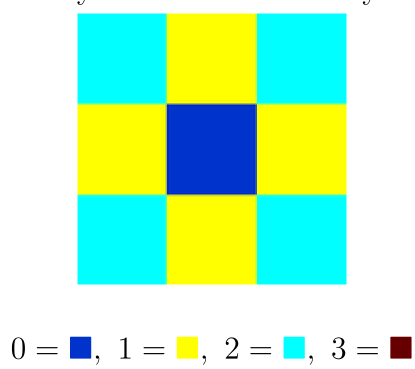
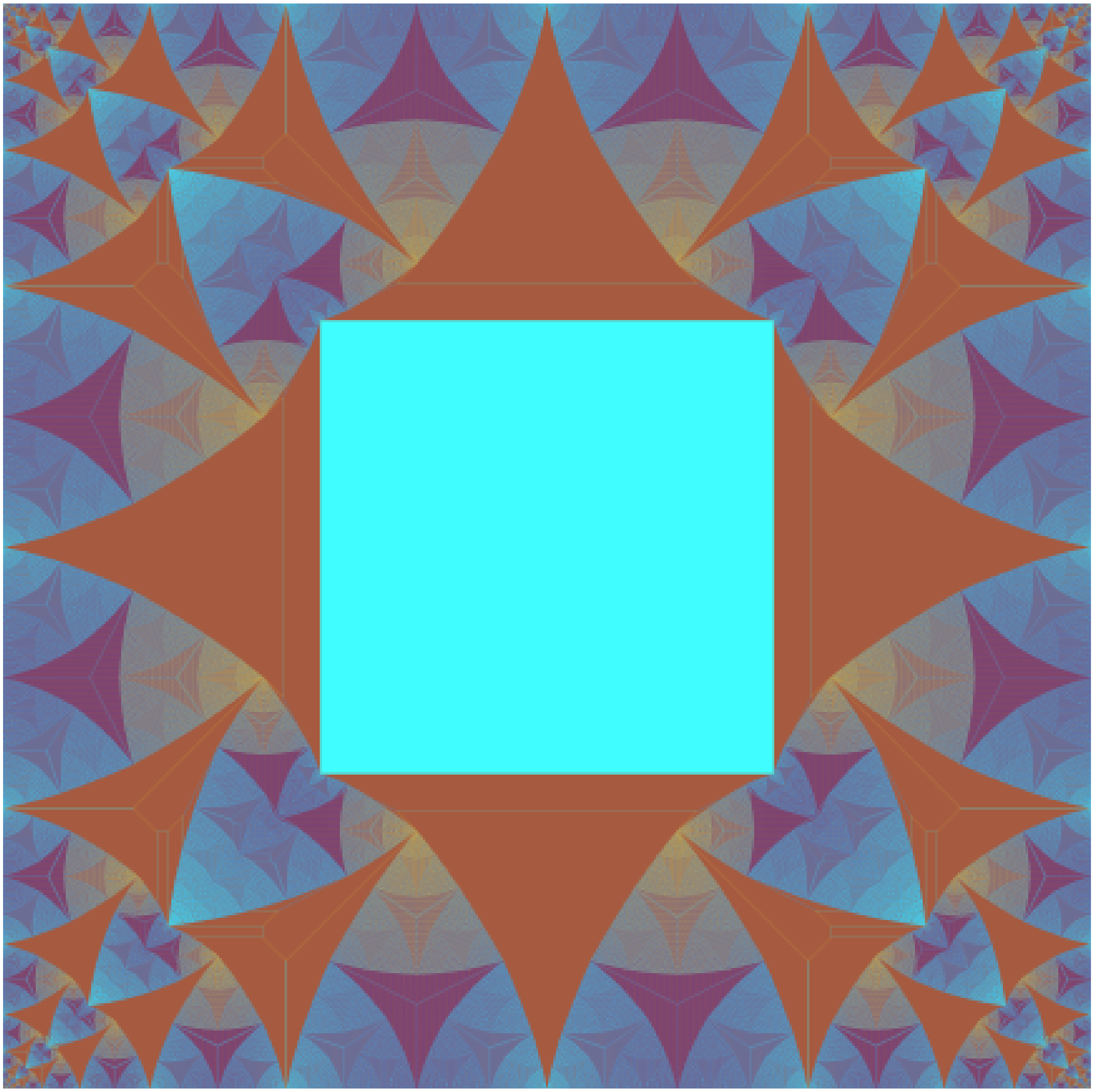
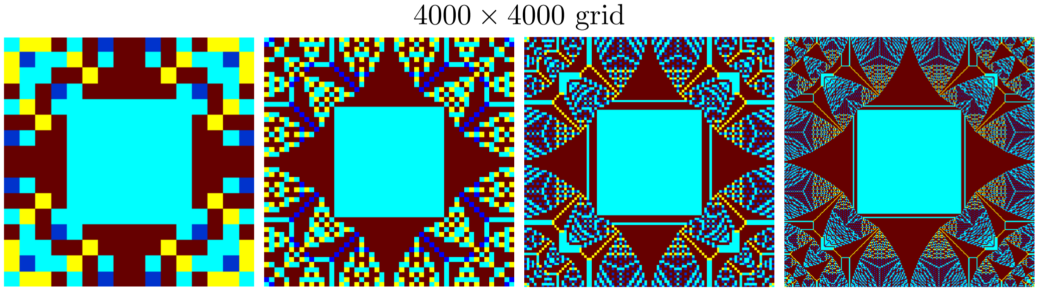


Figure 4.4

This doesn't look very impressive, but still there are some patterns in it. Everything is symmetric, which might not seem very interesting, but it still makes one wonder if that is true for all sandpiles in general.

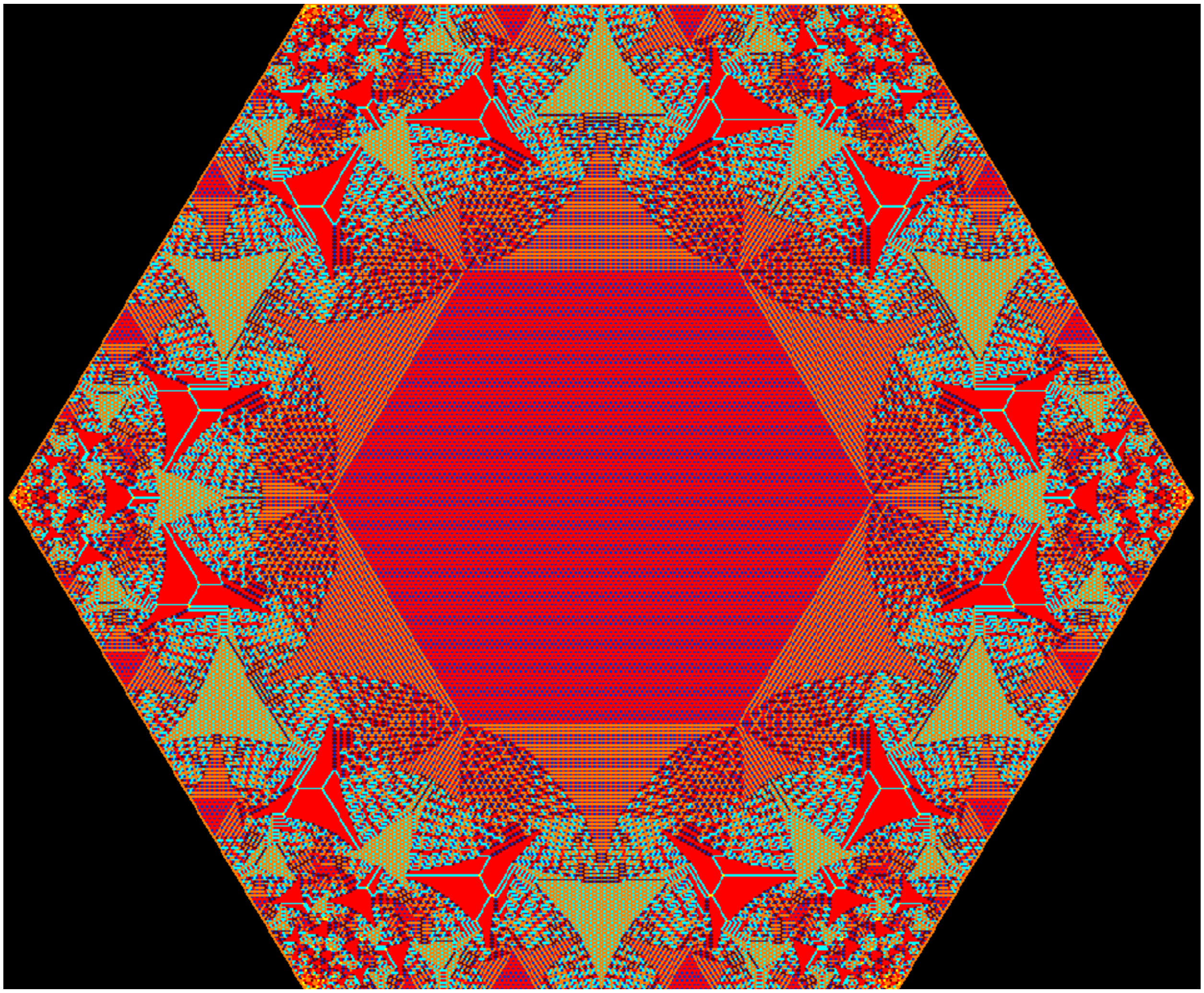


4000×4000 grid

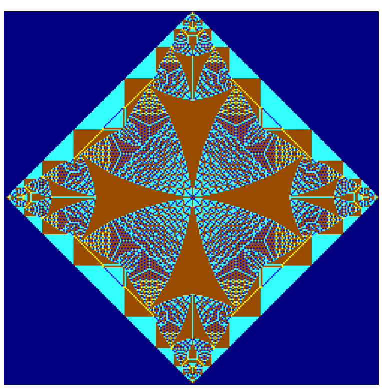


16×16 grid 50×50 grid. 100×100 grid 250×250 grid.

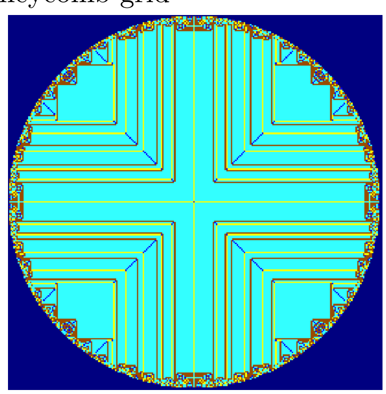
Figure 4.5. Identity elements for grids of various sizes.



250×250 honeycomb grid



250×250 diamond grid



250×250 circle grid.

Figure 4.6 Identity elements of non-rectangular grids of size 250×250

4.1. **Patterns.** As seen from Figure 4.6, the identity elements of sandpile grids can create incredible fractal patterns for very large grid sizes. What's so fascinating about these patterns are the simple mechanics of the sandpiles that produced them.

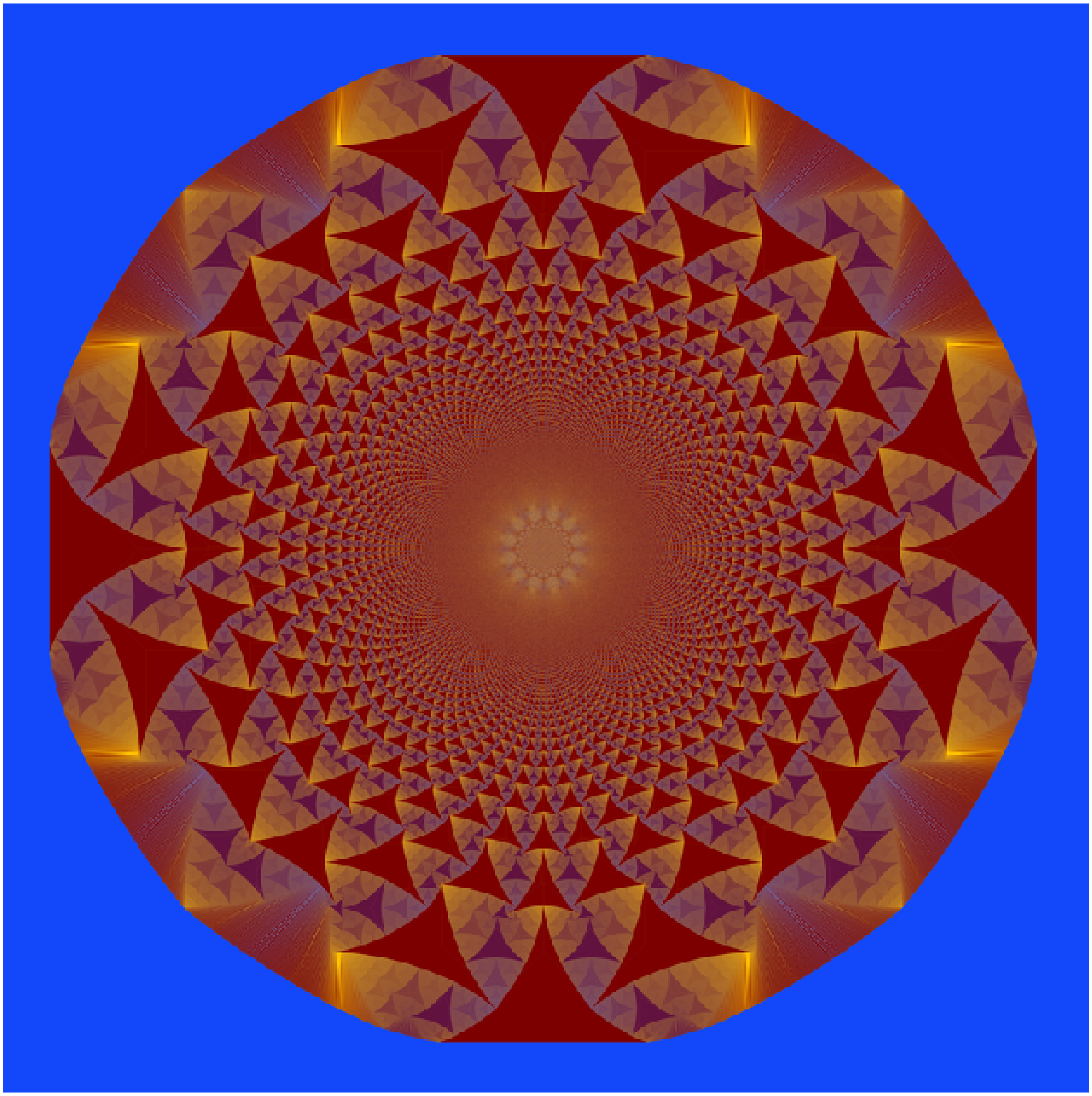


Figure 4.7 2^{30} pieces of sand dropped at the center of an infinite grid (Wesley Pegden)

Despite the many amazing discoveries that are made all of the time, there are many properties of these complex patterns that are still being explored to this day. For example, mathematicians have yet to characterize the large block of 2-vertices at the center of almost all of the identity elements. Despite all of the work that has been done, the incredible complexity of sandpiles leaves so many possibilities and avenues for exploration.

References

[BTW87] Per Bak, Chao Tang, and Kurt Wiesenfeld. Self-organized criticality: An explanation of the 1/f noise. *Phys. Rev. Lett.*, 59:381–384, Jul 1987.

[Per16] Scott Cory & David Perkinson. *Divisors and Sandpiles: Divisors and Sandpiles.* American Mathematical Society, 2016.