Pattern Formation in Growing Sandpiles

Aimee Ross

University of Massachusetts Dartmouth

Abstract

The topic of Pattern Formation in Growing Sandpiles was first looked at in the 1980's through a model of sand being added to a two dimensional system at a constant rate.[1] How this sand creates patterns of inactivity and unpredictability is an interesting natural phenomena. The Abelian Sandpile model is one model created to model these patterns, showing that the ending stable state of the model can create extensive and beautiful images.

1 Introduction

What is referred to as granular physics is the study of small but visible particles. One thing often studied is grains of rice, or grains of sand. This type of physics is interesting because the particles in part behave like a solid, but also in part behave like a liquid. Cellular automata can be used to explain how simple local interactions in a system can produce complex behavior. In this paper, we will examine the topic of two dimensional cellular automata, specifically the chip-firing game. After a brief introduction to self-organized criticality, we will look at the Bak-Tang-Wiesenfeld sandpile model and the Abelian Sandpile Model. Finally we will look at the MATLAB code that can be implemented and some images that are produced of the Abelian Sandpile Model.

2 Cellular Automata

Cellular automata are mathematical models for systems in which simple components act together to form complicated and unpredictable patterns using a set of predefined rules. Two dimensional cellular automata are important to explain pattern formation in many physical systems, including crystal growth, turbulent flow patterns, and is greatly related to the topic of the sandpile. There are two neighborhoods that are often used in 2 dimensional CA models.

The Abelian Sandpile Model resembles the CA known as the "Chip-firing game." This model is defined by two condiions:

Figure 1: The first neighboorhood approach is referred to as the "Five-neighbor square", or the von Neumann neighborhood. The second neighborhood is the "Nine-neighbor square." [4] The sandpile model uses the von Neumann approach.

1. The model exists on a directed graph
2. An initial configuration (the number of chips) is defined on each vertex

Like every CA model, the chip-firing game must depend on a predefined set of rules. The rules for the chip-firing game are as follows: If a vertex contains at least as many chips as its outgoing degree, this vertex can be fired. When fired, one chip is moved along each outgoing edge and placed on the corresponding vertex. The game will continue as long as there is at least one vertex with as many chips as the number of its outgoing edges. To model this CA model as a loop, using the rule for the chip-firing game and the five-neighbor square, the following code demonstrates how the CA model will fire chips:

```
if
A(xi,yi) ≥ 4;
A(xi,yi)= A(xi,yi) - 4;
A(xi+1,yi)=A(xi+1,yi) + 1;
A(xi,yi+1)=A(xi,yi+1) + 1;
A(xi-1,yi)=A(xi-1,yi) + 1;
A(xi,yi-1)=A(xi,yi-1) + 1;
end
```

3 Self Organized Criticality

The idea of Self Organized Criticality can be used to describe many natural systems including forest fires, earthquakes, and stock-market fluctuations. The idea of 'criticality' occurs through a critical point, where the system can experience radical behavioral change. To explain that the system is self-organized, changing the parameters of the model does not affect the criticality of the model. The final stable state is independent of changes in the time and length scales. The Oslo ricepile experiment, which is similar to the sandpile model but done with rice, was the first examination and proof that self organized criticality occurs in systems of granular matter, such as rice or sand. The frequency distribution of the size of avalanches in the sandpile model seems to follow a power law quite similar to the power law describing the magnitudes of earthquakes. [2]

4 Bak Tang Wiesenfeld sandpile model

4.1 Introduction to the model

The Bak-Tang-Wiesenfeld sandpile model was the first model of a dynamical system done with sand that exhibits self-organized criticality. It is named after Per Bak (a Danish theoretical physicist), Chao Tang (a Chinese physicist), and Kurt Wiesenfeld (an American physicist). [3] The model is defined on an LxL lattice. The model assigns a height value to each site. The model is stable if the height value at every site is less than four. When the site reaches the critical value of four, a toppling occurs. This one toppling may lead to a series of topplings. The addition of a single grain of sand may do nothing to the pile, or it may cause a massive avalanche. In a 2 dimenstional model, the avalanches can be characterized by the equations:

$$D(s) = s^{-\alpha_s} \tag{1}$$

$$D(t) = t^{-\alpha_t} \tag{2}$$

where $\alpha_s = \frac{5}{4}$ and $\alpha_t = \frac{3}{4}$. [2] $D(s)$ is the distribution of the size of avalanches and $D(t)$ is the distribution of the duration of avalanches.

4.2 Differences from a Real World Sandpile

There are two main differences between the Bak-Tang-Wiesenfeld sandpile model and a real world sandpile model:

1. In the Bak-Tang-Wiesenfeld sandpile model, the height value at each site may be thought of as the slope of a real world sandpile. However, a real world sandpile slope is continuous, where this model is discrete.

2. When a site near the boundary of the BTW sandpile model topples, sand can be lost. However, in a real world sandpile model, it does not take place with a square boundary and sand cannot be lost.

4.3 Power Laws

The BTW sandpile model displays power law behavior. This can simply be explained because there are many small avalanches, but few large ones. The law explains that the chance for an event is inversely proportional to its size. The flow of sand falling off the edges in this model also exhibits 1/f noise. If time allowed, I would have like to better examine the power laws and 1/f noise of the model.

5 Abelian Sandpile Model

The Abelian Sandpile Model is a two-dimensional cellular automata model. It takes place on an LxL square lattice. Each site i has an initial height value, an integer between 1 and 4, assigned to it. One site is chosen and sand is added to the system at this site. If the height value of the site is greater than 4, the site will topple. The height of this site will decrease by 4 and the height of its nearest neighbors, using the Von Neumann approach, will increase by one. This process will cause chain reactions of avalanches until the system is stable.

The Abelian Sandpile Model takes into account the importance of the Abelian property. Recurrent configurations of the Abelian Sandpile Model have the structure of a finite abelian group. The Abelian group property, stated

$$a * b = b * a \tag{3}$$

meaning the operation is commutative, is related to this model in that the order of site topplings does not matter. This can be explained in that no matter which order the topplings of a system occur in, if the system has the same initial conditions, it will evolve to the same stable configuration.[5]

The Abelian Sandpile Model takes into account three important properties:

1. Toppling must terminate in a finite time.
2. The Abelian property holds true for the system.
3. New sand is not generated in the toppling process.

To code the Abelian Sandpile model, the CA rule explained above for the chip-firing game must be utilized. The ASM takes into account that sand is added to a point that is chosen by the code and is not random. To set up the code, first a matrix of dimensions mxn is set up. The values of each site in this matrix represent the initial height values of each grain of sand in the model. Next, a for loop is used to add sand particles to the chosen site of the model. Within this loop, there are two loops used to reflect the CA rule of the model, showing that at each site of the model, if the height value is greater than or equal to four, that site will topple. The imagesc command is used to plot the matrix as an image and a colormap is chosen.

6 MATLAB code

6.1 Original Code

The following code is the original code that I wrote for the Abelian sandpile model. It sets up a matrix that has dimensions 100x100 with all initial heights set to two. It has 2500 time steps and adds sand to the single point in the center where x=50 and y=50. The next section of the code is the rule for the cellular automata that resembles the chip firing game explained above.

```
function originalabeliansandpile
x=100;
y=100;
A=2*ones(x,y);
for j=1:4060
xj=50;
yj=50;
A(xj,yj)= A(xj,yj)+1;
for xi=1:100
for yi=1:100
if A(xi,yi) >= 4;
A(xi,yi)= A(xi,yi) - 4;
A(xi+1,yi)=A(xi+1,yi) + 1;
A(xi,yi+1)=A(xi,yi+1) + 1;
A(xi-1,yi)=A(xi-1,yi) + 1;
A(xi,yi-1)=A(xi,yi-1) + 1;
end
end
end
imagesc(A)
colormap(spring);
pause(0.01)
end
end
```

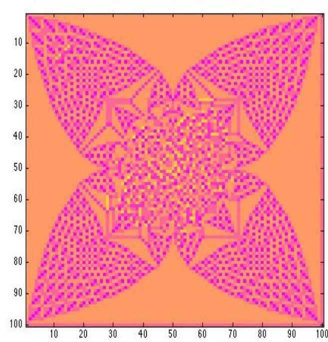

Figure 2: Abelian sandpile produced from the original code. The different colors of the colormap spring represent the different height values of the model.

6.2 Second Code

The second code I wrote changes the initial height value of the matrix to one. It keeps everything else the same except adding additional time steps. This code needed more time steps because the change in the initial height value made the model expand at a slower pace.

```
function abeliansandpile
x=100;
y=100;
A=ones(x,y);
for j=1:10000;
xj=50;
yj=50;
A(xj,yj)= A(xj,yj)+1;
for xi=1:100
for yi=1:100
if A(xi,yi) >=4;
    A(xi,yi)= A(xi,yi)  4;
    A(xi+1,yi)=A(xi+1,yi) + 1;
    A(xi,yi+1)=A(xi,yi+1) + 1;
    A(xi-1,yi)=A(xi-1,yi) + 1;
    A(xi,yi-1)=A(xi,yi-1) + 1;
end
end
end
imagesc(A)
colormap(spring);
pause(0.0001)
end
end
```

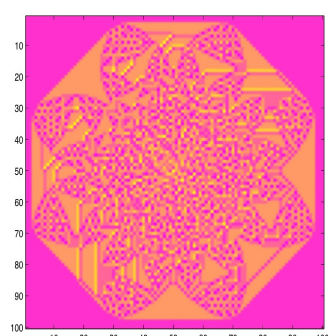

Figure 3: Abelian sandpile produced from the second code.

6.3 ASM with multiple points

When working with the Abelian Sandpile model, adding sand to multiple different points, rather than one single point, produced images that were different, but the same. The following code is a sample code adding sand to three different points.

```
function abeliansandpile
x=200;
y=200;
A=2*ones(x,y);
for j=1:12;
for k=1:12;
for l=1:12;
     xl=150;
     yl=150;
     xk=50;
     yk=50;
     xj=100;
     yj=100;
     A(xj,yj)= A(xj,yj)+1;
     A(xk,yk)= A(xk,yk)+1;
     A(xl,yl)= A(xl,yl)+1;
        for xi=1:200
        for yi=1:200
            if A(xi,yi) >=4;
            A(xi,yi)= A(xi,yi)  4;
            A(xi+1,yi)=A(xi+1,yi) + 1;
            A(xi,yi+1)=A(xi,yi+1) + 1;
            A(xi-1,yi)=A(xi-1,yi) + 1;
            A(xi,yi-1)=A(xi,yi-1) + 1;
            end
        end
        end
end
imagesc(A)
colormap(cool);
pause(.0001)
end
end
```

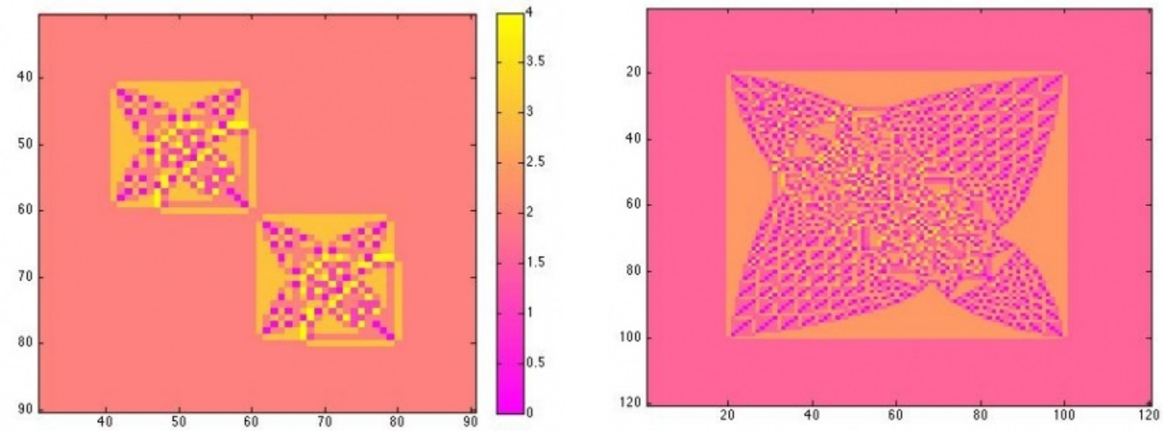

Figure 4: The figure on the left has initial height of two with twelve iterations. The colorbar on the side ranges from 1 to 4, as in the height at each site of the model. The symmetrical pattern produced by the ASM is still clear in this image. The figure on the right has initial height of two with twelve iterations. The colorbar on the side ranges from 1 to 4, as in the height at each site of the model. The symmetrical pattern produced by the ASM is still clear in this image

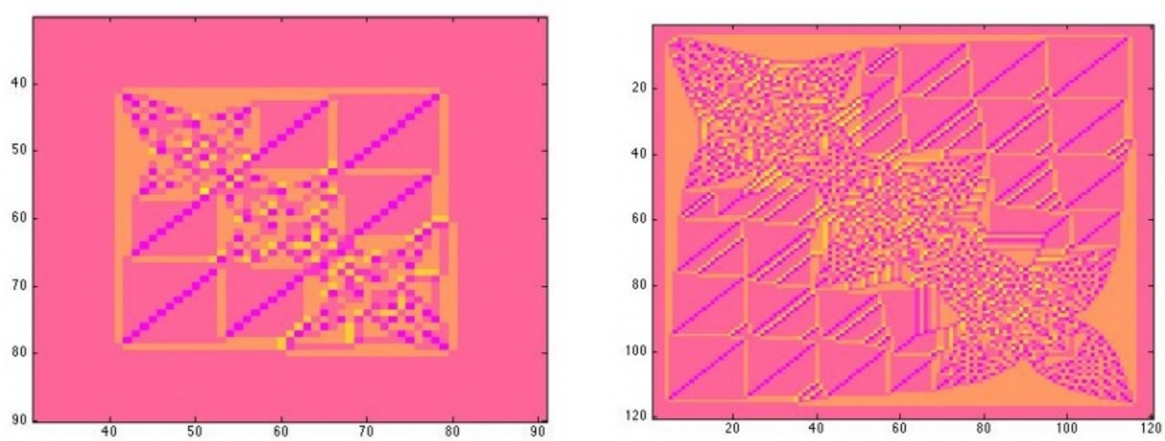

Figure 5: Both figures have an initial height of two. The figure on the right is the same model as the figure on the left after more iterations have occurred.

7 Next Steps

There are some topics of this model that I was unable to research closely. The next thing I would like to accomplish is to prove that the ASM displays power law behavior. I would like to examine the topics of power law behavior as well as 1/f noise, and relate these to the distribution of avalanches in the model and the amount of sand that leaves the system at each time step. I would like to alter the MATLAB code I wrote for the model so it will calculate the amount of sand leaving the system as well as the size of the avalanche for each time step. I can then use that information in a histogram as well as a log plot to show the power law behavior.

8 Conclusion

The topic of cellular automata is one that has been getting more and more attention in recent years. It can be used to explain complex behavior in many natural systems, as well as the self-organization of the sandpile model. Two main sandpile models are the Bak-Tang-Wiesenfeld model and the Abelian Sandpile model. Both of these models exhibit power law behavior. The Abelian Sandpile model can be coded using MATLAB, showing that the model expresses symmetrical behavior.

9 Acknowledgements

I would like to thank Professor Gary Davis and Professor Alfa Heryudono for their help in supporting this project, as well as the National Science Foundation for funding this research.

References

[1] Samarth Chandra Deepak Dhar, Tridib Sadhu. Pattern formation in growing sandpiles. Technical report, Tata Institute of Fundamental Research, 2008.

[2] Claudius Gros. *Complex and Adaptive Dynamical Systems*. Springer, 2008.

[3] Elka Korutcheva. *Advances in condensed matter and statistical physics*. Nova Publishers, 2003.

[4] N.H. Packard and S. Wolfram. Two-dimensional cellular automata. *Journal of Statistical Physics*, 1985.

[5] Frank Redig Ronald Meester and Smitri Znamenski. The abelian sandpile; a mathematical introduction. Technical report, Vrije Universiteit Amsterdam Vrije Universiteit Amsterdam, Department of Mathematics, 2001.