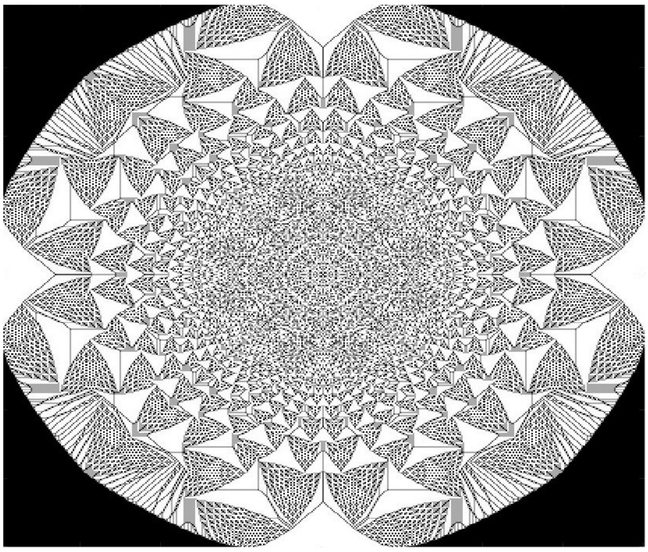
Final Report

The Art of Scientific Computing: Complex System

Author:
Qihan Zou
1174116

Subject Handbook code:
COMP90072



Faculty of Science
The University of Melbourne

1st Oct 2023

Subject co-ordinators: A/Prof. Roger Rassool & Prof. Harry Quiney

Contents

Chapter 1

Introduction

1.1 Background and introduction of this project

There are many famous cellular automatons such as the Elementary cellular automaton and Conway's Game of Life that can help us to discover the idea of self-similar and fractal. In this project, we focus on a simple cellular automaton, the sandpile model which is also called the Abelian sandpile model (ASM) or Bak–Tang–Wiesenfeld model (BTW). The sandpile model (BTW model) is the first model that displays self-organized criticality and its idea and algorithm were first introduced by Bak et al. (1987). Later, Dhar (1990) study a BTW automaton model that the toppling conditions depend on the height of the site rather than on its gradient, Dhar refers to this model as the Abelian model (AM) since this model follows Abelian dynamics. In this study, we first examine the basic sandpile model and then we build different models and perform simulations based on the basic sandpile model and analysis some properties of them.

During the work of this project, I got some inspiration from the model we built in section 3.5 namely the Rotation lighthouse model (add sand in the center). My current project for my master's thesis is closely related to astronomy, in particular, to determine the Metallicity distribution of distant galaxies (mainly spiral galaxies). Hence, I am interested in the structure of spiral galaxies, especially, the formation of the spiral arms of spiral galaxies. When I see the results of the model in section 3.5, it shows some patterns that are similar to the spiral structure. Besides that, I try to do the extension of the sandpile model in the polar coordinate system in the early stage of the project. Hence, I want to further develop the model to see whether is possible to relate the sandpile model to the galaxy formation model. Hence, I study the two famous galaxy spiral formation theories, the first one is the density wave theory proposed by Lin and Shu (1964) and the second one is the stochastic self-propagating star formation (SSPSF) model introduced by Mueller and Arnett (1976) and some related works. Since the second theory is more related to our study of the sandpile model, therefore, we will focus on the discussion of the SSPSF model and the possible relation to the sandpile model. In this project, I have not successfully built out an extended sandpile model that can generate a clear spiral structure yet in this stage. I only got some basic results for this idea and further models need to be built and examined in future studies. We will continue our discussion of them in Chapter 4 of this report.

Furthermore, there are many by-products things and interesting things have been discovered in this project besides the main project. In addition, I learned many useful techniques of computer programming.

The writing part of the project is done by using the Overleaf with Latex. The source code for the Latex should be available on the GitHub file.

The coding part of this project is mainly done by using the programming language MATLAB. We use MATLAB R2021a for the first few parts of the codes such as for the basic sandpile model and we use MATLAB R2023b for the remaining parts. We should note that all of the codes should be able to run in MATLAB version R2023b. Besides that, other languages such as PYTHON have been used to build the basic sandpile model at the early stage of this project, however, we will focus on the discussion based on the MATLAB codes.

Some packages such as Image Processing Toolbox, Statistics and Machine Learning Toolbox, and Symbolic Math Toolbox are required for running the codes. In addition, the reason behind the usage of the Parallel Computing Toolbox package is to improve the computation and simulation speed. The package is mainly used in part 2 of the extension.

We should note that some of the contents (mainly in the basic sandpile model, Chapter 2) are based on an unpublished document Complex System provided by COMP90072-The Art of Scientific Computing. This document was written by William Lawrie, Patrick Kennedy, Joshua Ellis, and Isaac Sanderson in the School of Physics of The University of Melbourne (17th Aug 2020).

In this report, the first chapter describes the coding language and packages and some important notes. Chapter 2 mainly focuses on the basic sandpile model. Chapters 3 and 4 focus on the extended models of the basic sandpile model. Chapter 5 discuss the limitation and lack of this project and some potential further studies. Chapter 6 is the summary and the conclusion of the project.

We use CXSY or CXSYPZ to name the results documents. C represents Chapter, $X \in 2, 3, 4$ is a variable representing the chapter number. S represents Section, $Y \in 1, 2, 3, ...$ is a variable representing the section number. P represents Part, $Z \in 1, 2, 3, ...$ is a variable representing the part number. For example, C2S1 represents the results document of Chapter 2 section 1. C3S9P2 represents the results document of Chapter 3 section 9 part 2. Furthermore, there exists a file called More Results which contains more results of the simulation that we do not include in this report. We should use this way of naming to check the results and further results that we included in the document that is available in zip files.

Some conventions are when we calculate N/2 and M/2 but N or/and M is/are odd numbers, then we always use the ceil of the result values if they do not state a specific new rule.

All codes that provide the results in this report are available in our zip files. All videos and plots are available in our zip files. If you have any problem or question regarding the contents, codes, results, plots, videos, pictures, and references of this report, please feel free to contact us via the following email address: qihan@student.unimelb.edu.au or qihanzou@gmail.com. We will reply to you as soon as possible.

Chapter 2

Basic BAK-TANG-WIESENFELD Model

2.1 sandpile model (random) and analysis

The first model we will discuss is the basic two-dimensional sandpile model which a new grain of sand is added to a random site in the table. Suppose we have a $N \times M$ two-dimensional table. Let $Z_t(i,j)$ indicate the number of grains on the site (i,j) at time step t where $i \in [1,...,N]$ and $j \in [1,...M]$. Suppose the time steps t are discrete and positive integers. Then, a new grain of sand is added to a random site in the table at each time step t. This makes the value of $Z_t(i,j)$ increase, that is the number of sand grows. Besides that, we assume the table is empty initially. Furthermore, if $Z_t(i,j)$, the number of sand in site (i,j) or say the height of sand reaches 4 or more, then the sand pile topples and distributes four grains of sand to its four immediate neighbours. We assume the neighbours do not include the diagonal neighbours in this case. Mathematically:

$$if \quad Z_t(i,j) >= 4$$

Then, we do the following:

$$Z_t(i,j) = Z_t(i,j) - 4$$
$$Z_t(i \pm 1, j) = Z_t(i \pm 1, j) + 1$$
$$Z_t(i, j \pm 1) = Z_t(i, j \pm 1) + 1$$

We should note that this process is instantaneous and does not increment t. In this case, we assume the table is bounded, that is, if toppling happens on the edge of the table $i \in 1, N$ and/or $j \in 1, M$, then the grains that would topple to a site of the table are deleted from the system. Based on this algorithm, the BTW model can evolve toward criticality by itself.

In the randomly adding grains cases, if we choose a table without finite boundaries, there will not be the properties that we want to investigate since the sand can drop in any site of the infinite table. In this situation, much of the sand in distant sites may never affect the sand pile in other sites, and hence we do not expect it will display the Self-organized criticality. Therefore, for the randomly adding grains cases, we will only choose to use a table that has finite boundaries.

However, dropping the sand on only a finite subset of an infinite table would display the Self-organized criticality, and hence the case of the table with infinite boundaries will be discussed in later sections (section 2.3).

At each time step, unstable sites are possible to topple and the toppling can influence other sites to be unstable and subsequently topple. We define an avalanche as a chain of toppling caused by the addition of a grain of sand. Based on the algorithm that is described above, we mainly focus on some properties of the avalanches in this study, such as the average height of the sandpile, the Topples, the Area, and the Loss. Before the analysis, we should define the definitions of above properties as follows:

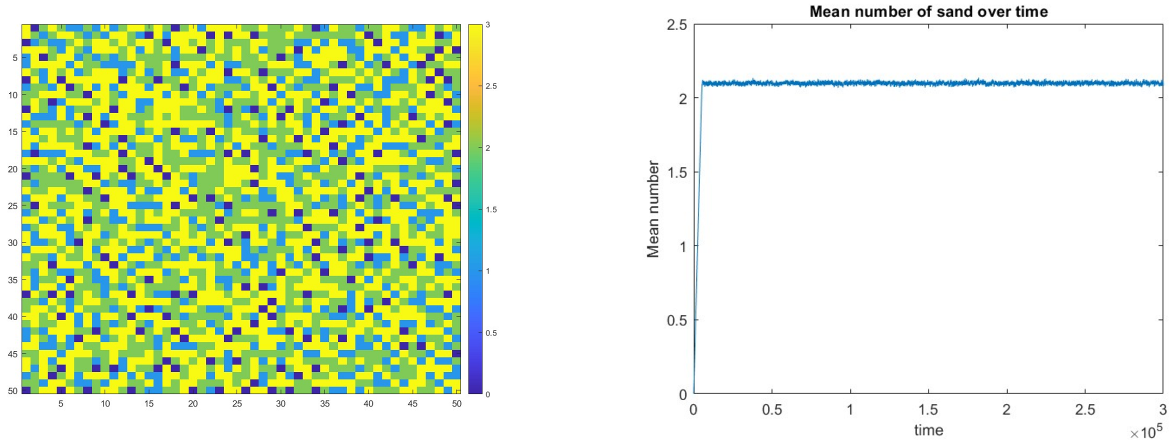
1. Average height: $E(Z_t) = \frac{1}{NM} \sum_{i,j} Z_t(i,j)$.
2. Topples: the number of topples is the topples times that are required for an unstable state to become a stable state.
3. Area: it is the number of unique sites that have been toppled.
4. Loss: the number of grains of sand that are toppled out of the table boundaries.

After the simulations, we use a linear model to fit log(topples) vs log(frequency); log(Area) vs log(frequency); and log(Loss) vs log(frequency). We choose to use the log scale because the values of the results differ widely.

We run the code three times, the first time uses a 50x50 table with 10000 time steps, the second one uses a 50x50 table with 50000 time steps and the third one uses a 50x50 table with 300000 time steps. We include the results of the first and second run in the GitHub document and also available in the "figure" of this Latex document but we do not put them in the report PDF. We should first note that all of the results of 3 different runs with different time steps are similar, therefore we only show and analyze the result of the third run.

One important thing we should mention here is the area plots of 10000 and 50000 time steps are not accurate since the code has some errors when we run them, hence we should ignore the area plots for them in the GitHub documents, the same situation for the next two sections. The third run that we will show below with a correct area plot and the new correct codes are updated in the GitHub code section for chapter 2. In addition, raw plots without fitting by the linear model are available in the GitHub results document and also available in the "figure" of this Latex document.

The following plots are the results of the simulation in a 50x50 table with 300000 time steps (sand drops).

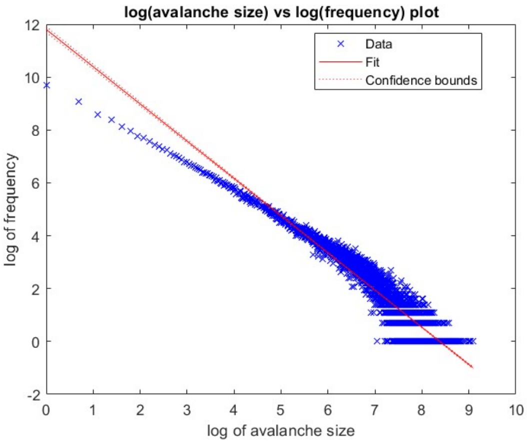


(a) 50x50 table with 300000 time steps

(b) The mean height of sand in the 50x50 table

Figure 2.1: Sandpile on the table and mean height: We can see that the sandpile looks randomly distributed in the table. The mean height of the sandpile is approximately 2.1 units. The height stays constant after the early few iterations.

From the above Figure 2.1 (a), we can see that in the sandpile model adding sand randomly results from a table with random sand. We cannot discover some regular patterns in this case. Moreover, from Figure 2.1 (b), we find that the mean height of the sandpile increases and then remains constant at around 2.1 units of height. This indicates the model reached a critical state since the number of grains on the table stays approximately constant. (The important thing we should note here is the title of the mean height plot "Mean number of sand over time" means "the mean height number" in this case rather than the mean number of sand in the table. However, we should note that if the mean height value of the table is almost constant as time steps increase, we can conclude that the mean number of sand in the table should be almost constant as time steps increase.)



(a) Log of avalanche size and Log of its frequency plot

2×4 table

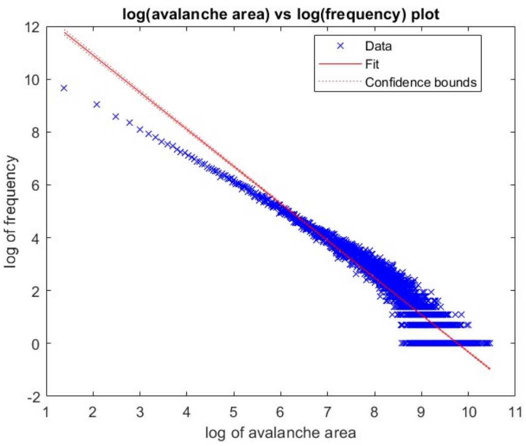
	Estimate	SE	tStat	pValue
(Intercept)	11.8	0.054973	214.65	0
x1	-1.4082	0.0075589	-186.3	0

(b) The corresponding coefficients of the Log of avalanche size and Log of its frequency plot by using linear model fitting technique

Figure 2.2: avalanche size plot and corresponding coefficients: We can see there exists a clear power law relation between the avalanche size and its frequency since there exists a linear relation between them in log plots that are shown in (a). In (b), the table is the corresponding coefficients of the linear equation line (red line in (a)) that we obtained by fitting the data points using a linear model. The "estimate" of (intercept) and (x1) are the coefficients. The linear model will be used throughout our report frequently. SE, t Statistics, and p Value are some more values that are related to the property of this linear fitting line. We will always include them in our report so if there exist readers are interested in it, are able to check them.

From the log of avalanche size versus the log of frequency plot, Figure 2.2 (a), we can see that there exists a power law distribution between the avalanche size s of the random adding sandpile model and its frequency $f(s)$ which indicates the sandpile reach the critical state. Mathematically, $f(s) = \beta x^{\alpha}$ for some β and α. By taking the log on both sides, we can get a linear fitting between the log avalanche size and its log frequency and we can denote this as $y = ax + b$ where a is the gradient of this linear equation and its value is shown in the above coefficient table. The b represents the y-intercept and its value is also shown in the above table. (We should note that the equation of the power law distribution and the linear equation will be mentioned lots of times in the rest part of this report, hence we should avoid writing them many times and hence we decide to not repeat them in this report. We will use words like "the power law distribution" to mention these two equations.)

Next, the log of the avalanche area versus the log of its frequency plot and the corresponding coefficients table are shown below. We fitted the data points by using the linear model again.



(a) the log of the avalanche area and the log of its frequency plot

2×4 table

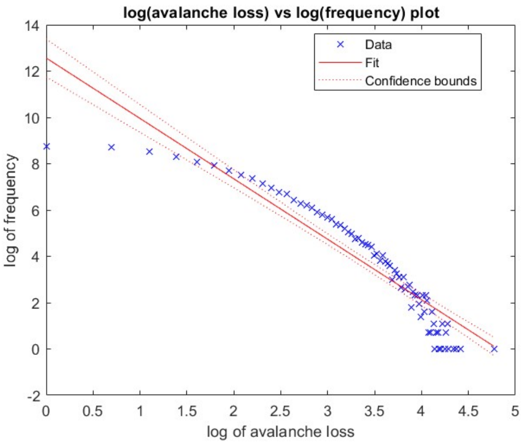
	Estimate	SE	tStat	pValue
(Intercept)	13.712	0.065089	210.66	0
x1	-1.4036	0.0075354	-186.26	0

(b) The coefficients of the log of the avalanche area and the log of its frequency plot

Figure 2.3: avalanche area plot and corresponding coefficients: clear linear relation between the log of the avalanche area and the log of its frequency implies a power law relation between the avalanche area and its frequency.

From the above plot, Figure 2.3 (a), we can see there also exists a power law distribution between the avalanche area and its frequency. The corresponding coefficients table, Figure 2.3 (b) (the coefficients of the best linear fitting line, the red line in Figure 2.3 (a)) is shown above.

Finally, the log of avalanche loss versus the log of its frequency plot is shown below.



(a) the log of the avalanche loss and the log of its frequency plot

2×4 table

	Estimate	SE	tStat	pValue
(Intercept)	12.564	0.41212	30.485	1.2296e-43
x1	-2.6095	0.11763	-22.184	1.9915e-34

(b) The coefficients of the log of the avalanche loss and the log of its frequency plot

Figure 2.4: avalanche loss plot and corresponding coefficients: We can see the shape of the raw data points in Figure 2.4 (a) is similar to the quadratic curve rather than the linear line. This may imply the avalanche loss and its frequency do not follow a power law relation. In Figure 2.4 (b), the coefficients table of the best linear fitting line (red line in Figure 2.4 (a)), we can see that the SE (Standard Error) is very large compared to the SE in Figure 2.3 and Figure 2.2. These results further imply that the linear model is unsuitable in this situation and strengthen our conclusion that there does not exist a strong power law relation between the avalanche loss and its frequency.

From the above plot, Figure 2.4, the linear relation between the log of avalanche loss and the log of frequency is not significant as we discussed above.

In summary, the basic sandpile model with randomly adding sand at each time step shows a power law distribution relation between the avalanche size and its frequency and between the avalanche area and its frequency. In addition, we can see that the log scale plots do not follow the linear relationship exactly, there are some deviations in their tails, the reason behind that is that we only use a 50 x 50 table and it is finite and relatively small.

2.2 sandpile model (middle) and analysis

To extend the model in the last section, we changed the rule for adding new grand of sand to be the new grain of sand added to the middle site of the table at each time step. All other parts of the algorithm remain the same as in section 2.1. By using these rules, many beautiful patterns can be created, one of the famous sandpile model patterns is created by using this algorithm. Mathematically speaking, at each time step t,

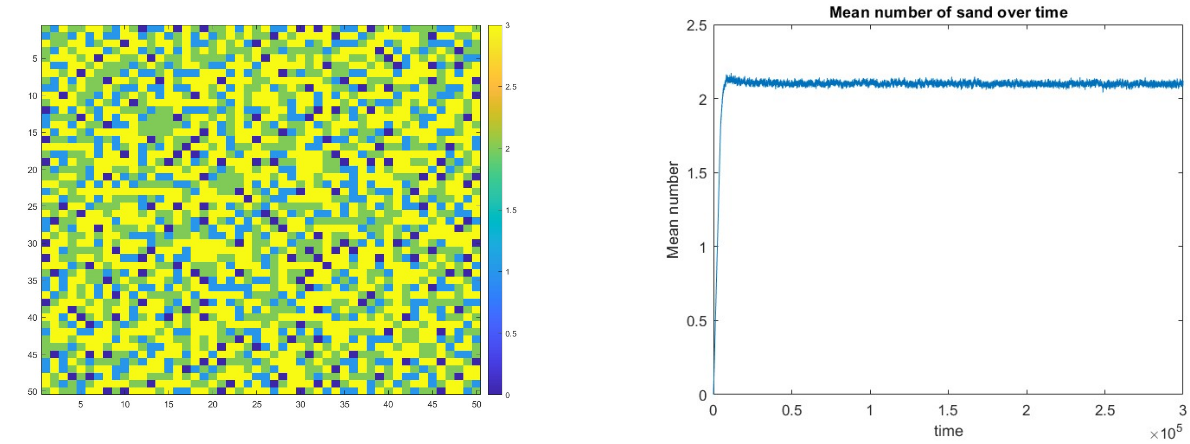
$$Z_t(N/2, M/2) = Z_t(N/2, M/2) + 1$$

and then we do the following:

$$if \quad Z_t(i,j) >= 4$$
$$Z_t(i,j) = Z_t(i,j) - 4$$
$$Z_t(i \pm 1, j) = Z_t(i \pm 1, j) + 1$$
$$Z_t(i, j \pm 1) = Z_t(i, j \pm 1) + 1$$

For a bounded $N \times M$ table, if the values of $N/2$ and/or $M/2$ is/are non-integer, by convention, we usually take the ceil of the value(s).

We run the code three times, the first time uses a 50x50 table with 10000 time steps, the second one uses a 50x50 table with 50000 time steps and the third one uses a 50x50 table with 300000 time steps. We present the results from the third run here:



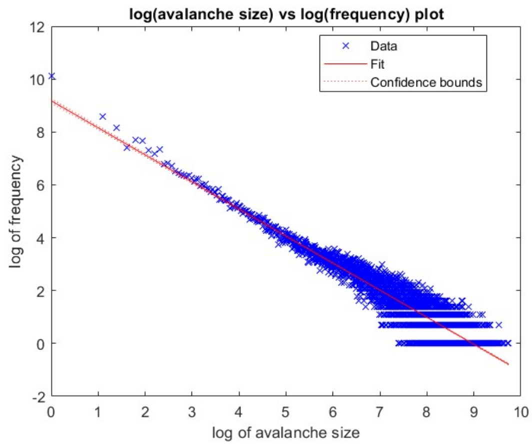
(a) 50x50 table with 300000 time steps

(b) The Mean height number of the table

Figure 2.5: The sandpile model (middle) and its mean height plots: We can see there is a clear pattern in (a), it appears some symmetry around the center middle point. In (b), we can see the mean height is stable around 2.1 similar to our result in section 2.1.

From the above plots, we can see some patterns in the sandpile plot in Figure 2.5 (a), however, since the table is a 50 x 50 table and is relatively small, we cannot see the whole pattern clearly in this case. We will present a clear picture in the next section with an unbound table. Besides that, we can see the mean height of this model is almost the same as the model in the last section and it stays almost constant indicating the model reaches the critical state.

First of all, we should begin our analysis by looking at the log avalanche size versus the log of its frequency plot and the corresponding coefficients that are shown below.



(a) The log of avalanche size and the log of its frequency plot

2×4 table

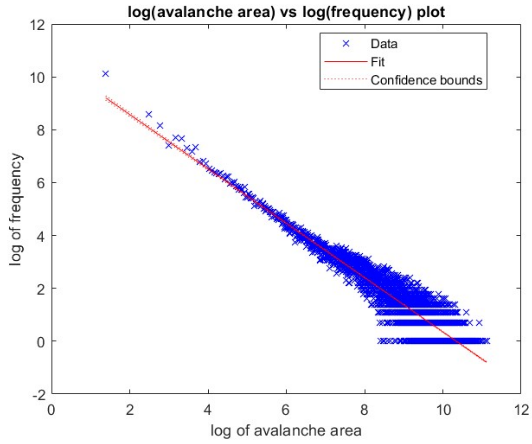
	Estimate	SE	tStat	pValue
(Intercept)	9.1787	0.041217	222.69	0
x1	-1.0234	0.0051981	-196.88	0

(b) The corresponding coefficients of the log of avalanche size and the log of its frequency plot

Figure 2.6: The log of avalanche size versus the log of its frequency plot and the corresponding coefficients: We can see that there exists a very strong linear relationship in the data points in (a) which implies the avalanche size and its frequency are follow a power law relationship. From the corresponding table, Figure 2.6 (b), the SE of the estimate parameters is very small implying the linear relation is very strong which strengthens our conclusion.

From Figure 2.6 (a), We can see there exists a power law distribution relationship between the avalanche size and its frequency from the above plots, and the best linear fitting coefficients are shown beside the plot.

Furthermore, the log of the avalanche area and the log of its frequency plots are shown below.



(a) The log of the avalanche area and the log of its frequency plot

2×4 table

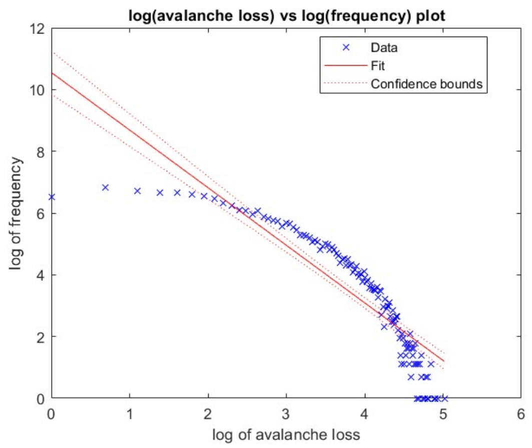
	Estimate	SE	tStat	pValue
(Intercept)	10.63	0.048804	217.8	0
x1	-1.0278	0.0052464	-195.91	0

(b) the coefficients of the log of the avalanche area and the log of its frequency plot

Figure 2.7: The log avalanche area versus the log of its frequency plot and the corresponding coefficients: we can see that there exists a strong linear relationship between the log of avalanche area and the log of its frequency. This implies there exists a strong power law relation between the avalanche area and its frequency. The SE in (b) is very small which strengthens our conclusion.

From the above plot, Figure 2.7, we can see there exists a power law distribution relation between the avalanche area and its frequency. The log of avalanche loss versus the log of its frequency plot and the corresponding coefficients table are shown below. As same as before, we cannot see a significant linear relation between them.

Finally, the log of the avalanche loss versus the log of its frequency plot and the corresponding coefficients table are shown below;



(a) The log of the avalanche loss versus the log of its frequency plots

2×4 table

	Estimate	SE	tStat	pValue
(Intercept)	10.553	0.35178	29.998	7.1089e-58
x1	-1.8656	0.088845	-20.998	3.6645e-42

(b) The corresponding coefficients of the log of the avalanche loss versus the log of its frequency plots

Figure 2.8: The log of avalanche loss versus the log of its frequency plot and the corresponding coefficients: we can see the shape of the data points in (a) is similar to the quadratic curve rather than the linear line and hence our linear model fitting may not suitable and this implies there does not exist a linear relationship between the log of the avalanche loss and the log of its frequency. Hence, there does not exist a power law relation between the avalanche loss the its frequency. The SE in (b) is relatively large, which implies the linear fitting is unsuitable again.

In summary, for the sandpile model (middle) in this section, we can see the power law distribution relationship between the avalanche size and its frequency and between the avalanche area and its frequency. In this case, we can see both the log scale plots almost follow the linear relation, the tail of them mostly lies on the fitting line.

2.3 sandpile model with open (unbound) table

We consider the tables to have boundaries before. Now, for further investigation of the sandpile model, we suppose the model in the last section is in an open table without finite boundaries. That is, the table will grow when more sand is added to it. We drop the grains of sand only in the middle of the table. The remaining part of the algorithm is the same as we discussed in the last section.

In this case, we do not include a loss plot since there should not exist any loss as our table will grow when the total area of the sandpile increases.

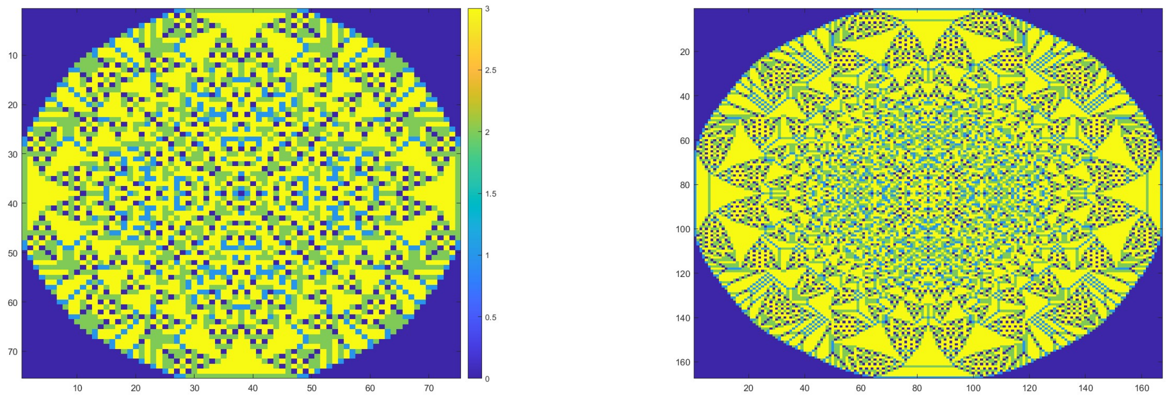
We should build an algorithm to make the table from a closed table to an open table and there is a simple way to achieve this. We should follow the idea below. Simply speaking, when there exists sand in the boundaries of the table, we make the table bigger.

```
%% set N and M both be 1 initially.
N=1;
M=1;

%% For each time step, for each topping site, we check whether the topping site is in the boundary of
    our table as below:
if x_coord_tops(j)==1 || y_coord_tops(j)==1 || x_coord_tops(j)==N || y_coord_tops(j)==M
%% If true, we add new sites around the table to become a larger table:
            sand_pile=[0, zeros(1,M), 0; zeros(N,1), sand_pile, zeros(N,1); 0, zeros(1,M), 0];
            N = N + 2;
            M = M + 2;
```

By adding the above codes to our previous codes, we can simulate the sandpile model in the unbound table now.

Now, we have the algorithm to achieve our objective, the following plots are the results of a 50x50 table with 10000-time steps and 50000-time steps. The plots are shown below:

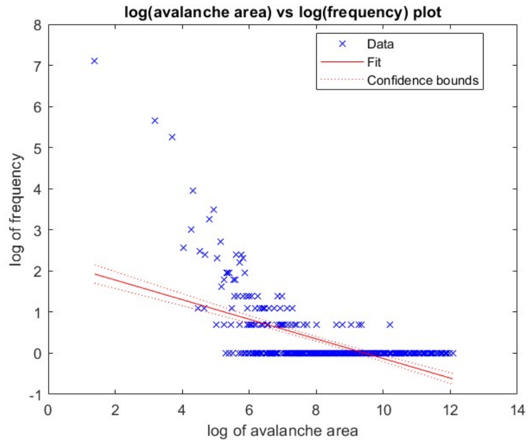


(a) The sandpile structure at 10000 time steps middle unbound

(b) The sandpile structure at 50000 time steps middle unbound

Figure 2.9: The sandpile structure at different time steps: we can see there exist very beautiful symmetry patterns appear in the result plots (a) and (b). As the time steps increased, in (b), the patterns became more and more elaborate.

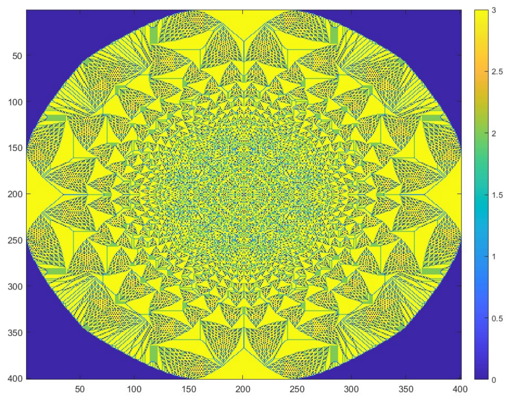
Due to the limitation of computation resources, we cannot present the area plot for 50000 and 300000 time steps cases since the computer cannot run out the result for them. Hence, we provide the area plot for 10000 time steps case as follows:



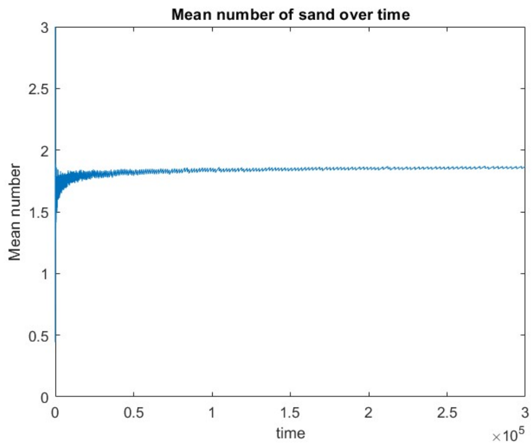
(a) 10000 time steps middle unbound table: the log of the avalanche area and the log of its frequency plot (Due to the limitation of the compute resource, we are unable to provide the area plots with more time steps.)

From the above log of the avalanche area plot, we can still see a weak linear relation between the log of the avalanche area and the log of its frequency. Since the time steps are relatively less and hence the fitting line looks a bit out of expectation. If we can plot the log of the avalanche area and the log of its frequency plots with more time steps, we should be able to obtain data points that have a stronger linear relationship.

Then, for the unbound table with 300000 time steps, the sandpile plot and the mean height plot are shown below.



(a) 300000 time steps sandpile in the middle adding, unbound table

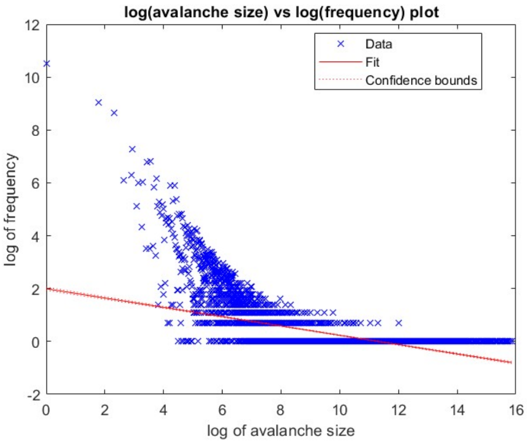


(b) 300000 time steps middle adding, unbound table sandpile mean height plot

Figure 2.11: The unbound 300000 sandpile plot and its mean height plot: The symmetry pattern is very clear now in (a). In (b), we can see the mean height of the sand is around 1.8 units and stays almost constant after the early few time steps.

We can see the unbound table with 300000 time steps can give us a very clear pattern at every time step (We can see this in the corresponding video). Then, we can see the mean height of this case is between 1.5 and 2 unit height. However, we should note here that the mean height includes the sites outside the main "circle pattern" in the table as shown above. These positions are 0 height and largely affect the mean height plot.

Furthermore, the log of avalanche size versus the log of its frequency plot and the corresponding coefficients table are shown below.



(a) 300000 time steps middle adding, unbound table: The log of avalanche size and the log of its frequency plot

2×4 table

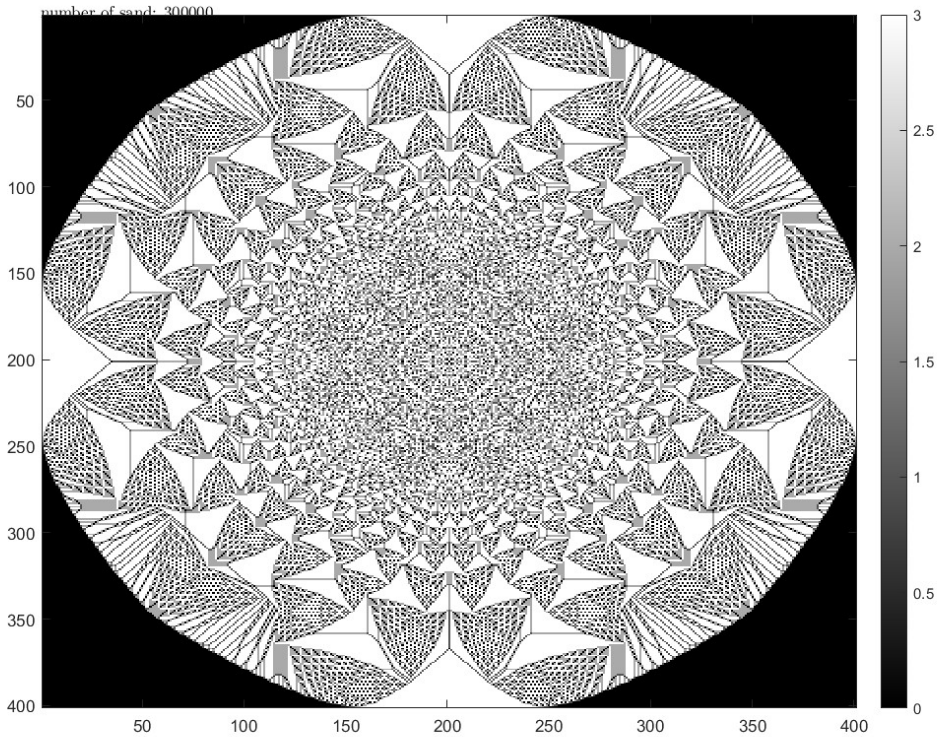
	Estimate	SE	tStat	pValue
(Intercept)	1.9946	0.029667	67.233	0
x1	-0.1761	0.0030127	-58.453	0

(b) 300000 time steps middle unbound size plot corresponding coefficients

Figure 2.12: The log of avalanche size versus the log of its frequency plot and the coefficients table

From Figure 2.12, we can see that the avalanche size and its frequency follow a power law distribution, but since they record lots of log(frequency)=0 and hence the best linear fitting line is a little bit strange. The reason is our analysis methods have some limitations in the unbounded table case, hence the best-fitting line by using the linear model is unreasonable for this case and we can fit the line manually in further study for these situations.

The beautiful picture on the front cover page of this report in greyscale is made by using this unbounded sandpile model with 300000 time steps.



(a) The cover page figure: unbounded sandpile model with 300000 time steps in greyscale. We do not include the axis in the version we used on the cover page.

In summary, we can still see there exists a power law distribution relation between avalanche size and its frequency and between avalanche area and its frequency although the best linear fitting line is a bit out of our expectation.

2.4 Technical part: How to make a video by using MATLAB

It is important to talk about how to make a video file to record each of our simulations. We will use this technique in almost every model simulation to record them. The idea is for every eg. 100 or 10 sand drops (time steps), we record a frame and then we put all of our record frames together to make a video.

Below is an example of the randomly added sandpile model that we discussed in the first section of this chapter. We record the frame for every 100 time steps and the resulting video will have the name "random added sandpile". We can change the "interval = " to be any positive integer number to record different numbers of frames, for example, if we write 1 after the equal sign, it will record every time step. We use "image()" to plot 2 Dimensional sandpile model here, and if we use "polarscatter()" here, we can plot the polar sandpile model and record the video. We do this in part 2 of the extension sandpile model for extending the sandpile model to polar coordinates. We should note that the resulting videos are in MPEG-4 format.

The following codes are the codes we used to make the video of this project, we can follow the comments that I wrote to see how to make the video successful.

```
%% before the main for loop:
interval=100; %% every 100 sand drops polt a graph.
frames(number_sand/interval)=struct('cdata',[],'colormap',[]);
bi=1; %index of images

%% before the end of the main for loop:
  if mod(i,interval)==0 %for each time satisfy the interval ,drawthe graph.
        image(sand_pile,'CDataMapping','scaled')
        colorbar
        graphsetting([0.2 0.1 0.6 0.8]); %percentage of the graph size corresponding to computer
             screen.
        str=['number of sand: ',sprintf('%d',i)];
        set(text(0,0,str),'interpreter','latex','HorizontalAlignment','left');
        drawnow;
        frames(bi) = getframe(gcf);
        bi=bi+1;
    end
%% Then,there is the end of the main for loop.

%% Then, at the end of the codes:
%% create video:
Video = VideoWriter('random added sandpile','MPEG-4');
open(Video);
writeVideo(Video,frames)
close(Video);

function graphsetting(setting)
    set(0,'units','centimeters') % we want to get unit in cm.
    computerscreensize=get(0,'screensize');
    Length = computerscreensize(3); %length of computer screen
    Height = computerscreensize(4); %height of computer screen
    position=[setting(1)*Length setting(2)*Height setting(3)*Length setting(4)*Height];
    set(gcf,'units','centimeters','position',position); %Set the position & size of graph.
end
```

This is all we should know for the technique to create a video for our project by using Matlab. We will use this structure a lot of time in the code. Every time we use this code just need to change the function for plotting figures or change the name of the variable or the model.

2.5 Chapter summary

We model three basic sandpile models in this chapter, the sandpile model with randomly adding sand, the sandpile model with adding sand at center, and the sandpile model with adding sand at center with open table. We analysis the mean sandpile height, avalanche size, avalanche area, and avalanche loss for them. We report the average height of each model in this chapter. We found there exists a power law distribution relationship between avalanche size and its frequency for all these three models and there exists a power law distribution relationship between avalanche area and its frequency for all these three models. For the avalanche loss, we cannot see there exists a significant power law distribution between it and its frequency. In addition to the resulting plots, we also provide the corresponding linear fitting coefficients of most of the plots. Furthermore, we created some beautiful figures of the open table sandpile model.

At the end of this chapter, we have a brief discussion about the important technique that we use to create videos for simulations in this chapter and we will continue to use the same technique to create videos for our simulations for the rest of our project. We should mention here, this technique can be used in any codes with iterations (frames). In the next chapter, we will extend the basic sandpile model to some extension sandpile models and analysis some properties related to them.

Chapter 3

Extending The BAK-TANG-WIESENFELD Model (Part 1)

In this chapter, we will focus on the extensions of the basic BTW model in a two-dimensional square table. We extend the model in the following ways: give an external force to force the direction of topples; introduce a rotation; make the grain of sand always add on the outer four corners or on the outer four edges. In addition to the simulation, we try to analysis some properties of these extended models, especially, we focus on the average height and the number of topples that we have given definitions in the previous chapter (chapter 2).

3.1 Lighthouse model (bounded)

The first extended model of this project, the lighthouse model is built by giving a force to force the direction of topples to be different in four regions of the table. As the result figure looks like a lighthouse, hence we call it the "lighthouse model" in this report. The algorithm for this model is at every time step we add 1 sand in the middle of the table when $Z(i,j)$, the height of a site is greater than 3 then this site topples, the value of height change to zero.

check the position of the topples coordinates x and y and if

$$x_{tops}(j) < N/2 \quad and \quad y_{tops}(j) >= M/2$$

then we will do the following action:

$$Z(x_{tops}(j)-1, y_{tops}(j)) = Z(x_{tops}(j)-1, y_{tops}(j)) + 2$$

$$Z(x_{tops}(j), y_{tops}(j)+1) = Z(x_{tops}(j), y_{tops}(j)+1) + 2$$

check the position of the topples coordinates x and y and if

$$x_{tops}(j) >= N/2 \quad and \quad y_{tops}(j) > M/2$$

then we will do the following action:

$$Z(x_{tops}(j)+1, y_{tops}(j)) = Z(x_{tops}(j)+1, y_{tops}(j)) + 2$$

$$Z(x_{tops}(j), y_{tops}(j)+1) = Z(x_{tops}(j), y_{tops}(j)+1) + 2$$

check the position of the topples coordinates x and y and if

$$x_{tops}(j) > N/2 \quad and \quad y_{tops}(j) <= M/2$$

then we will do the following action:

$$Z(x_{tops}(j)+1, y_{tops}(j)) = Z(x_{tops}(j)+1, y_{tops}(j)) + 2$$

$$Z(x_{tops}(j), y_{tops}(j)-1) = Z(x_{tops}(j), y_{tops}(j)-1) + 2$$

check the position of the topples coordinates x and y and if

$$x_{tops}(j) <= N/2 \quad and \quad y_{tops}(j) < M/2$$

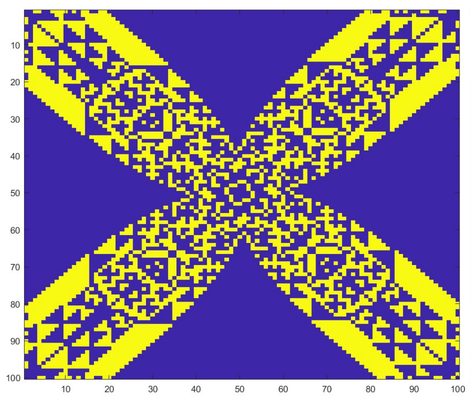
then we will do the following action:

$$Z(x_{tops}(j) - 1, y_{tops}(j)) = Z(x_{tops}(j) - 1, y_{tops}(j)) + 2$$

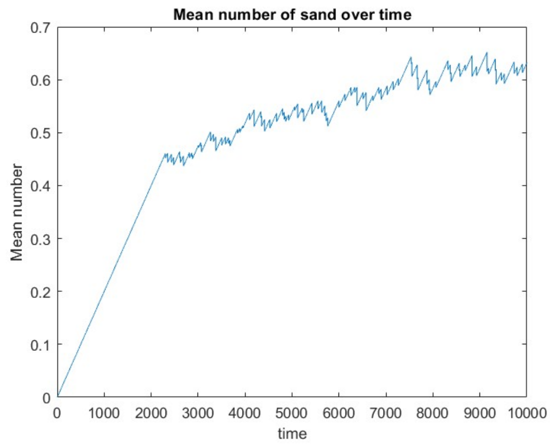
$$Z(x_{tops}(j), y_{tops}(j) - 1) = Z(x_{tops}(j), y_{tops}(j) - 1) + 2$$

where x_{tops} and y_{tops} are the position of the topple site. This is the main idea and the core of this model. The N and M represent we have an N $\times$ M two-dimensional table. for $N/2$ and $M/2$, we use "ceil" of the result value by convention. In addition, we need to note that j in x_{tops} and y_{tops} represent the index of the topple sites and it is not the same as the j in $Z(i, j)$. Furthermore, in this section, we assume the table is bounded. For the topples, we decide the neighbours of the topple sites will add 2 sand, the reason is if we only add 1 sand, the sandpile model generates too slowly.

We built a 100 x 100 table with 10000 time steps to do the simulation for this model, the results are shown below.



(a) 100x100 10000 time steps lighthouse model plot

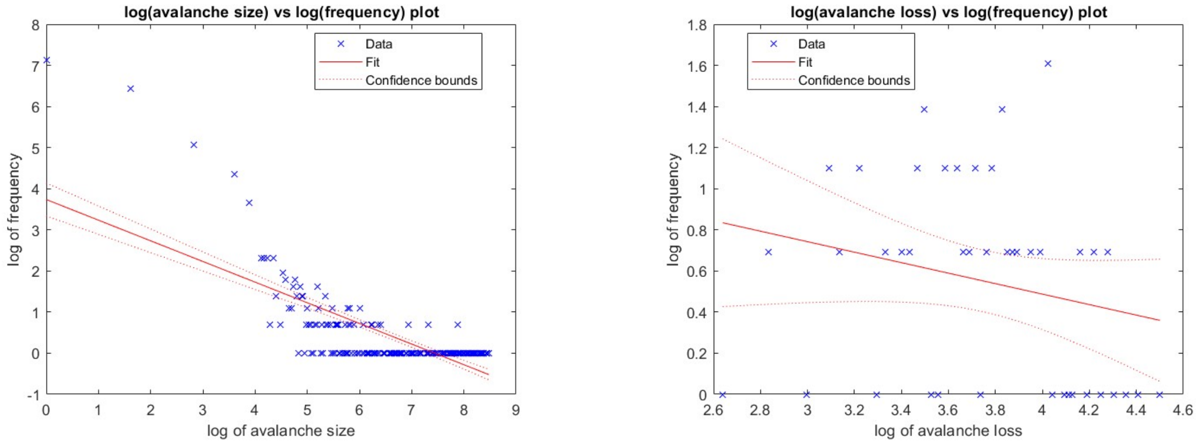


(b) 100x100 10000 lighthouse mean height plot

Figure 3.1: result plot and mean height plot of the 100x100 10000 time steps lighthouse model: we can see that from (a) the result sandpile looks like a symmetry "X" or a "Lighthouse", this is the origin of the name of this model. From (b), we can see the mean height has a wavelike rise and then becomes more stable after 7000-time steps although the main trend still fluctuates.

From the above plots, Figure 3.1 (a), we can see the sandpile that appears in our 100x100 bounded table looks like a lighthouse, like a flower, or an X chromosome. From the mean height plot, we can see the mean height of the sandpile is around 0.6 from Figure 3.1 (b), it is relatively low since the huge blue parts of the table contributed many 0 heights and in this case, since we force the topple directions as shown above, the heights of four huge blue parts will never grow (our table is bounded and sand add on the edge by toppling will loss). However, we can still see that the mean height is becoming constant throughout the time steps, and hence the mean number of sand in this table is becoming constant, this indicating that the sandpile model is in a critical state.

We present the log of the avalanche size versus the log of its frequency plot and the log of the avalanche loss versus the log of its frequency plot as follows:



(a) log of the avalanche size vs log of the frequency plot

(b) log of the avalanche loss vs log of the frequency plot

Figure 3.2: log of the avalanche size versus the log of its frequency plot and the log of the avalanche loss versus the log of its frequency plot: from (a), there exists a linear relationship in the data points implying the power law relation between the avalanche size and its frequency. However, from (b), there does not exist any recognizable relation between the log of the avalanche loss and the log of its frequency. Therefore, there does not exist a power law relation between the avalanche loss and its frequency.

From Figure 3.2 (a), we can see the avalanche size and its frequency still follow a power law distribution as same as the basic bounded sandpile model although the tail of the plot has some deviation. This may be due to the force we apply to the model or the time steps we used for this model being relatively less. For the loss plot, Figure 3.2 (b), we cannot see a significant pattern or relationship between the avalanche loss and its frequency. The corresponding coefficients tables determined by the linear model fitting are shown below.

2×4 table

	Estimate	SE	tStat	pValue
(Intercept)	3.7375	0.20347	18.369	3.9621e-47
x1	-0.50219	0.029533	-17.005	1.1835e-42

(a) Corresponding 100x100 10000 lighthouse size coefficients table for the best linear fitting line

2×4 table

	Estimate	SE	tStat	pValue
(Intercept)	1.5076	0.63818	2.3623	0.023113
x1	-0.25507	0.1691	-1.5084	0.1393

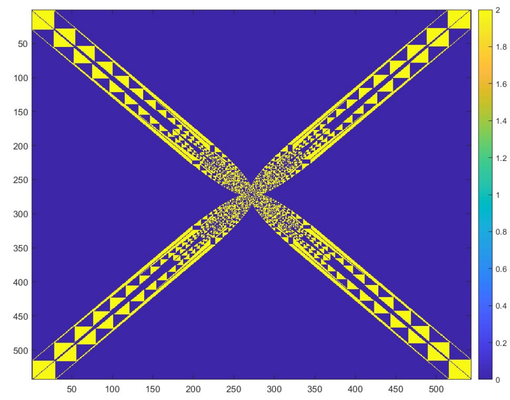
(b) Corresponding 100x100 10000 lighthouse loss coefficients table for the best linear fitting line

Figure 3.3: The coefficients of the best linear fitting line by using linear model

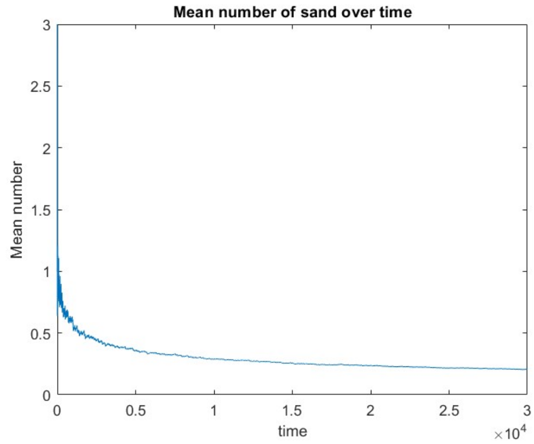
In summary, we built an extended sandpile model, the lighthouse model in this section and we found that there exists a power law distribution relation between avalanche size and the frequency of this model. Based on this model, we will develop further extended models in the next few sections.

3.2 Lighthouse model (unbounded)

In this section, we develop the lighthouse model with an unbounded table, that is, at the beginning of the simulation, the table is 1 x 1, and if there are any sites add some grains of sand including topples then the table will grow in both horizontal direction and vertical direction. The algorithm for extending the table is the same as the algorithm we discussed and shown in the last chapter for the unbounded sandpile model. Other parts of the algorithm are the same as we described in section 3.1. We simulate this model with 30000 time steps and record a plot for every 100 time steps to generate a video. The results are shown below.



(a) unbound table 30000 time steps lighthouse model plot



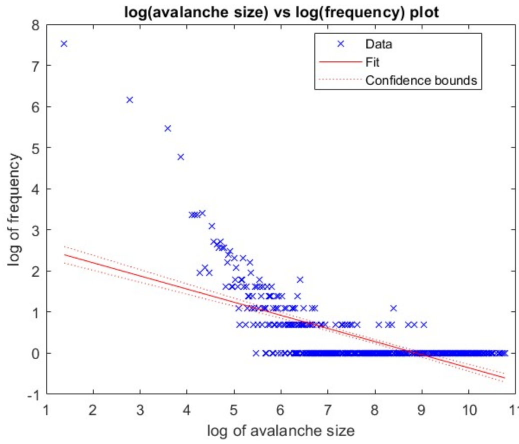
(b) unbound table 30000 time steps lighthouse mean height plot

Figure 3.4: resulting sandpile plot and the mean height plot: we can see the "X" become longer and thinner than the "X" in the previous section. The general pattern is very similar and still symmetry. From (b), we can see the mean height plot is very different compared to the mean height plot of the bounded lighthouse model in the previous section. In the unbounded table case, the mean height starts from above 1 and then decreases very fast in the interval within around 0 to 0.5×10^4 time steps. Then, the rate of decrease becomes slower after 0.5×10^4 time steps but still continuously decreases. The x-axis (y=0, the axis of the time step) should be an asymptotic line of the mean height since when the time steps go to infinity, there will be many positions with 0 height except the very thin "X" (The blue area in (a) will be very large). Therefore, the mean height will tend to be 0 when the time steps go to infinity or equivalent, the table tends to be infinitely large. This is the reason behind the difference between the mean height plot in Figure 3.4 for the unbounded lighthouse model and the mean height plot for the bounded lighthouse model.

From the above Figure 3.4 (a), we can see the sandpile still has a similar pattern to the model we built in the last section. It looks like a lighthouse that emits four light beams in four directions in the background of a blue ocean. The color of the higher sites is yellow by default and the color of the lower sites is blue. We might think, if we give this pattern a rotation, what pattern will created? We will keep this idea and develop it in the later section of this chapter.

From the mean height plot, Figure 3.4 (b), we can see the mean height of the sandpile decrease as time steps increase, this meets our expectation since as time steps increase, the table will increase as it is unbounded. However, since we restricted the direction of topples and hence most of the regions of this table will never be reached and keep zero height. Furthermore, these zero-height regions will become larger as time steps increase, and therefore, the mean height will continuously decrease. There should be an asymptotic line at $x = 0$ as the mean height will never reach zero.

Moreover, the log of avalanche size versus the log of its frequency plot is shown below.



(a) The log of the avalanche size vs log of its frequency plot

2×4 table

	Estimate	SE	tStat	pValue
(Intercept)	2.8363	0.12269	23.118	4.8197e-83
x1	-0.3193	0.015355	-20.795	3.0839e-71

(b) The log of the avalanche size vs log of its frequency plot coefficients

Figure 3.5: The log of the avalanche size vs log of its frequency plot and the corresponding coefficients: we can see from the (a), that there exists a linear relation between the data points implying the power law relation between the avalanche size and its frequency. From (b), we can see the corresponding SE is a little bit high since our fitting line by using a linear model is largely affected by the data points in the log of frequency equal to 0. However, we can still see the strong linear relation in (a), hence we should conclude that although the fitting line has some deviation from the main trend by using a linear model, there still exists a power law relation between the avalanche size and its frequency.

From the above plot, Figure 3.5 (a) and (b), we can see that there exists a power law relation between avalanche size and its frequency, and the corresponding coefficients table is shown above.

3.3 Lighthouse model with add sand in five sites in middle (unbounded)

In this section, we further extend the last unbounded lighthouse model to a new model. For each time step, we add 5 grains of sand in 5 different sites around the middle of the table after the first time step. For the first time step, we only add 1 sand in the middle of the table. Mathematically:

for the first time step (t=1 or the first sand drop), we let:

$$Z(N/2, M/2) = Z(N/2, M/2) + 1$$

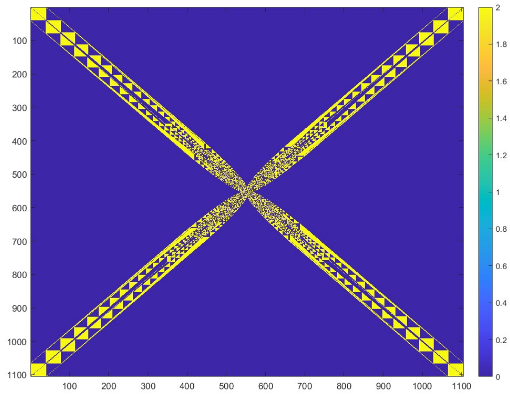
then, for the remaining time steps, that is for

$$N > 1 \quad and \quad M > 1,$$

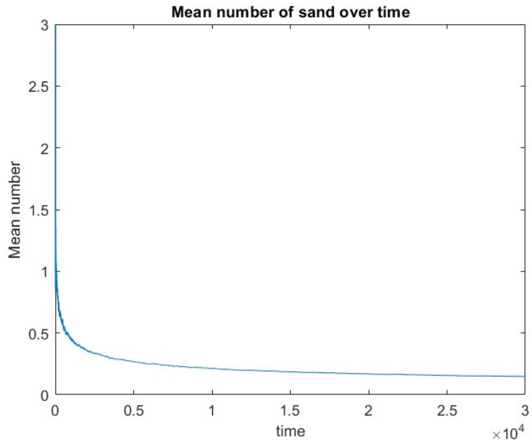
we let:

$$\begin{aligned} Z((N/2)-1,(M/2)-1) &= Z((N/2)-1,(M/2)-1)+1 \\ Z((N/2)+1,(M/2)+1) &= Z((N/2)+1,(M/2)+1)+1 \\ Z((N/2)+1,(M/2)-1) &= Z((N/2)+1,(M/2)-1)+1 \\ Z((N/2)-1,(M/2)+1) &= Z((N/2)-1,(M/2)+1)+1 \end{aligned}$$

for every time step, where Z is the height of the sandpile, the definition is the same as we used before. The other parts of the algorithm are the same as the lighthouse model (unbounded) described above. We want to see does the initial condition affects the model a lot by doing this. We simulate this model by running 30000 time steps.



(a) unbound table 30000 time steps modify lighthouse plot

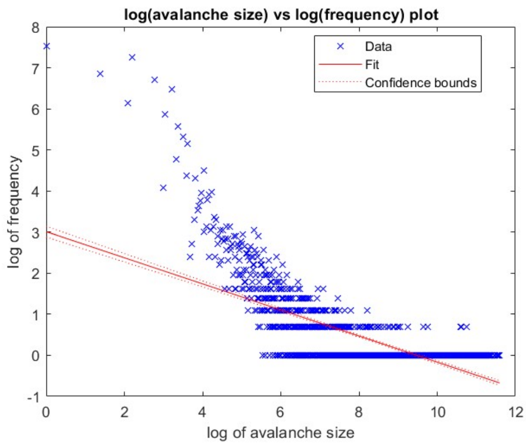


(b) unbound table 30000 time steps modify lighthouse mean height plot

Figure 3.6: result plot of the unbound table 30000 time steps modify lighthouse and the corresponding mean height plot: From (a), the only main difference is the table is much larger than the original unbounded lighthouse model in the previous section. The other part of the result plot is almost the same as before. Form (b), the behavior of the mean height of the modified lighthouse sandpile model is the same as before, the time axis is the asymptotic line and the mean height will approach 0 when the time goes to infinity.

Clearly, we can see the table is much larger than the model with only adding sand at the center in the last section from Figure 3.6 (a). However, the overall pattern is similar. For the mean height plot, Figure 3.6 (b), we can see this model still behaves like the model in the last section. Hence, we can conclude that there is not a huge difference when we change some initial conditions around the adding rules at the center of the table of the lighthouse model.

Similarly, the avalanche size versus frequency plot in the log-log scale shown below is very similar to the last model with different initial conditions.



(a) The log of the avalanche size vs the log of its frequency plot

2×4 table

	Estimate	SE	tStat	pValue
(Intercept)	3.0107	0.067587	44.546	3.8833e-290
x1	-0.31769	0.0079601	-39.91	4.458e-248

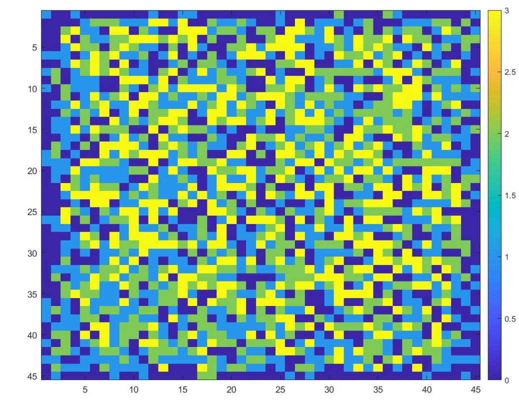
(b) The log of avalanche size vs the log of its frequency plot coefficients table

Figure 3.7: The log of the avalanche size vs the log of its frequency plot and the corresponding coefficients table: Clearly we can see the linear relation between the log of the avalanche size and the log of its frequency form (a) which implies there exists a power law relation between the avalanche size and its frequency. The corresponding coefficients of the fitting line by using linear model estimation is shown in (b).

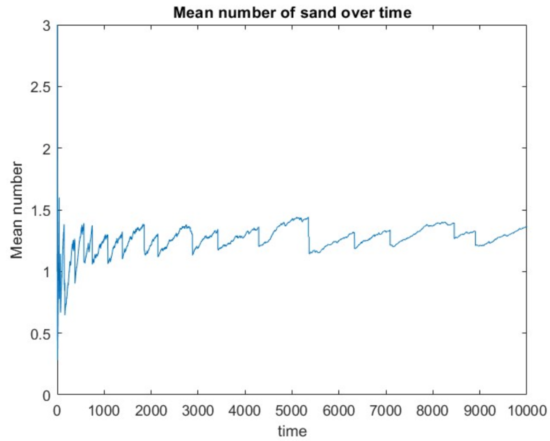
From Figure 3.7 (a), we can see that there exists a power law distribution relation between the avalanche size and its frequency. Overall, we can only see one difference: the resulting table and the area of this model are larger than the lighthouse model (only adds one sand in the middle for each time step) in the last section.

3.4 Lighthouse model (unbounded) with random add sand in table

In this section, we build a lighthouse model that every time step, we add sand to the table randomly rather than always adding sand in the center of the table. We still use an unbounded table in this case, therefore, for the first step, the sand is added in the center. Then as the time step increases, the table will grow. We ran this model twice times, the first one with 10000 time steps, and the second one with 300000 time steps. In addition, one important thing that we should mention here is when a topple happens, the height of the neighbours of the topple site will only increase by one instead of two as we discussed above. Other parts of the algorithm are the same as we describe above for the Lighthouse model (unbounded). The results are shown below.

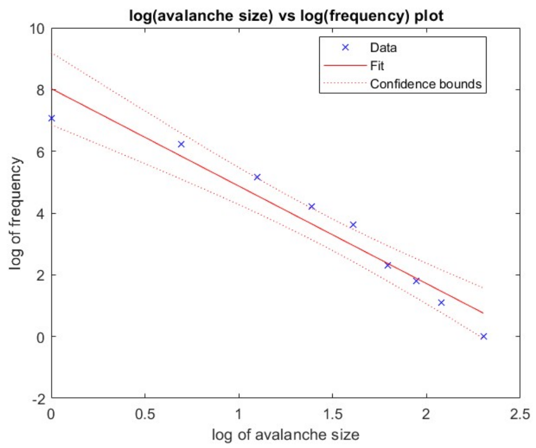


(a) The unbound table with 10000 time steps, random adding lighthouse model plot



(b) The unbound table with 10000 time steps, random adding lighthouse model mean height plot

Figure 3.8: resulting sandpile plot and the corresponding mean height plot



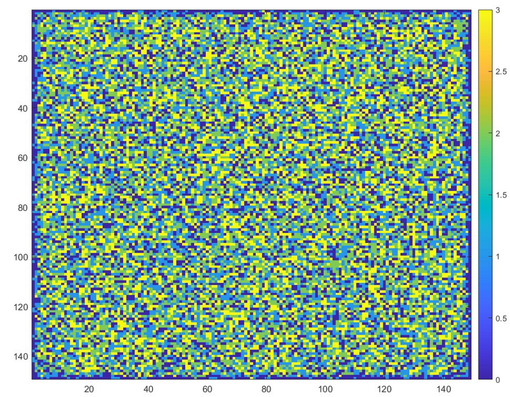
(a) The log of avalanche size vs the log of its frequency plot

2×4 table

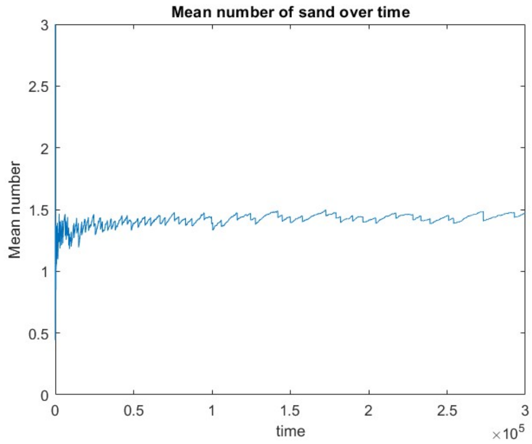
	Estimate	SE	tStat	pValue
(Intercept)	8.0291	0.49526	16.212	8.2682e-07
x1	-3.1581	0.31101	-10.154	1.9336e-05

(b) The log of avalanche size vs the log of its frequency plot coefficients

Figure 3.9: The log of avalanche size vs the log of its frequency plot and the corresponding coefficients table

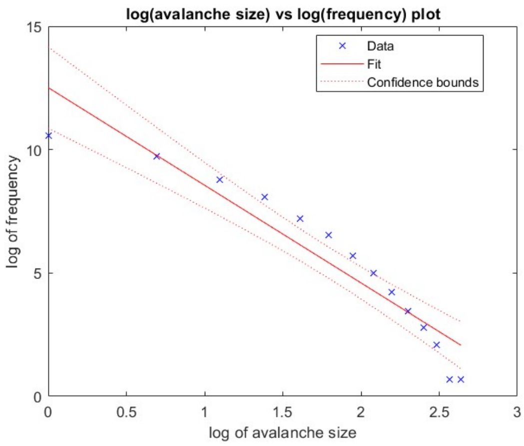


(a) The unbound table with 300000 time steps, random adding lighthouse model plot



(b) The unbound table with 300000 time steps, random adding lighthouse model mean height plot

Figure 3.10: resulting sandpile plot and the corresponding mean height plot



(a) The log of the avalanche size vs the log of its frequency plot

2×4 table

	Estimate	SE	tStat	pValue
(Intercept)	12.513	0.75688	16.532	1.2718e-09
x1	-3.9575	0.38857	-10.185	2.9374e-07

(b) The log of the avalanche size vs the log of its frequency plot coefficients table

Figure 3.11: The log of the avalanche size vs the log of its frequency plot and the corresponding coefficients table

Overall, from the above plots, Figure 3.8 (a) and Figure 3.10 (a), we can see that the sandpile in the table looks random, however, when we look at them closer and carefully, we can see there exists some pattern that is different than the random adding sandpile model that we discuss in the last chapter. The mean height of the sandpile is around 1.5 unit height constantly in both Figure 3.8 (b) and Figure 3.10 (b). From Figure 3.9 (a) and Figure 3.11 (a), there seems also to exist a power law distribution between the avalanche size and its frequency but the data points collected by our algorithm are not enough and our algorithm for avalanche size that is used in this model does not perform well. However, the resulting plot still shows a linear relation as shown above. We should do further simulations with advanced algorithms to further examine our conclusion here in the future study.

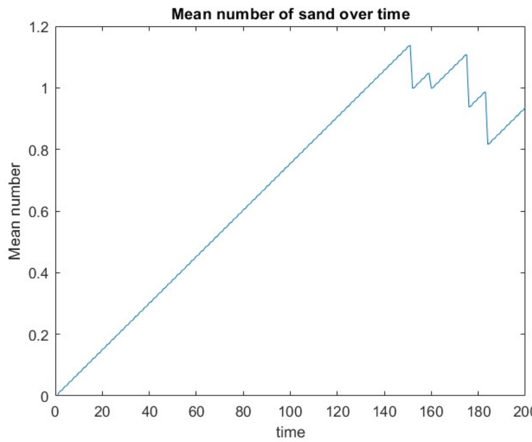
3.5 Rotation lighthouse model (add sand in the center)

In this section, we develop a rotation-bounded lighthouse model, this is one of the main parts of our extended model. The algorithm is the same as the algorithm used for the lighthouse mode (bounded) in section 3.1. The difference is we add a rotation process at the end of each time step. The rotation direction is clockwise and the details will be discussed in the last section of chapter 3 "A new Matrix rotation algorithm". We should note that for every topple, the height of neighbours of the topple site will add 2 as same as in section 3.1.

We first simulate this model in a 23 x 23 table with 200 time steps and then simulate this model in a 33 x 33 table with 200 time steps. We should mention that the reasons for choosing 23 x 23 and 33 x 33 tables and only 200 time steps are due to the limitation of our computation resources. Although we use both CPU and GPU to do the simulation by using the Parallel Computing Toolbox package, the computer that we can use is unable to perform a simulation with the larger table or more time steps. We will do the simulation with a larger table and more time steps in the future once we get more computation resources. However, we should note that the results with 200 time steps can still uncover some properties of this model.

The following is the plot of the sandpile in the table after 200 time steps and besides that is the mean height plot of the model.

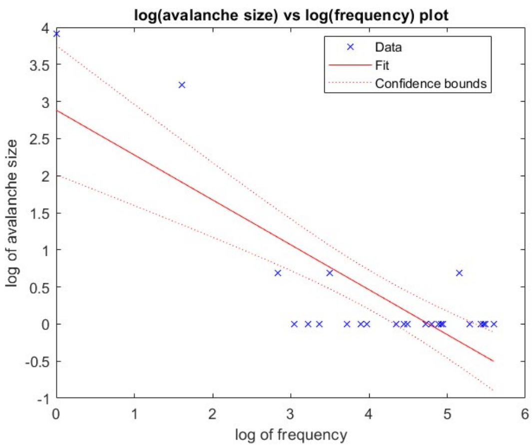
(a) 23x23 table with 200 time steps rotation lighthouse model plot



(b) 23x23 table with 200 time steps rotation lighthouse model mean height plot

Figure 3.12: The rotation lighthouse result plot and the corresponding mean height plot

From the above rotation lighthouse plot, Figure 3.12 (a), we can see a symmetry and self-similar pattern for the sandpile. From the mean height plot, Figure 3.12 (b), we can see the mean height stays almost constant after 140 time steps but it still has a big fluctuation. The fluctuation is due to the special pattern that this model made, it changes a lot but the pattern will become similar periodically. The log of the avalanche size versus the log of its frequency plot with the corresponding coefficients table is shown below.



(a) The log of the avalanche size vs the log of its frequency plot

2×4 table

	Estimate	SE	tStat	pValue
(Intercept)	2.8839	0.41989	6.8682	6.7339e-07
x1	-0.60535	0.096915	-6.2462	2.7547e-06

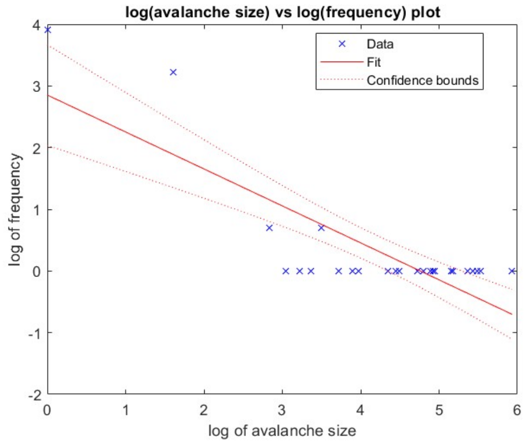
(b) The log of the avalanche size vs the log of its frequency plot coefficients table

Figure 3.13: The log of the avalanche size vs the log of its frequency plot and the coefficients table: we can see there does not exist a linear relationship between data points in (a), which implies that no power law relation between the avalanche size and its frequency in this rotational lighthouse sandpile model.

In this case, from Figure 3.13 (a), we cannot see there is a significant linear relationship between the log of avalanche size and the log of its frequency. This may be due to the rotation that we are involved. However, it may also be due to the time steps we used for this model is really very small compared to what we used for other models. For this stage, we concluded that there does not exist a power law distribution relation between the avalanche size and its frequency in this rotation lighthouse model.

As we want to see the whole pattern of the sandpile in the table for this model, we changed our table be 33x33 to simulate this model again. The reason that we chose a 33x33 table is because this is the maximum size of the table that our computer can do the simulation for this model.

(a) 33x33 table with 200 time steps rotation lighthouse model plot



(b) 33x33 table with 200 time steps rotation lighthouse model mean height plot

Figure 3.14: result rotation lighthouse sandpile plot and the log of the avalanche size versus the log of its frequency plot: From (a), we can see the whole pattern now, and from (b), there does not exist a linear relation between the log of the avalanche size and the log of its frequency. The result is the same as the above discussion which confirms our conclusion that there does not exist a power law relationship.

From the above result sandpile plot, Figure 3.14 (a), we can see the clear whole pattern of this rotation lighthouse model. We also provided the log of avalanche size and the log of its frequency plot, Figure 3.14 (b), as we discussed before, we cannot see a significant power law distribution relation between them.

In this section, we only provide the video of the 23 x 23 200 cases but for 33 x 33 200 cases, the codes for the simulation can also generate video for it. Furthermore, if we have more computation recuses, we can generate other videos for a larger table and more time steps.

We will show some more captures of the result plots below during the simulation to see how the pattern grows, is generated, and becomes the pattern that we have shown above.

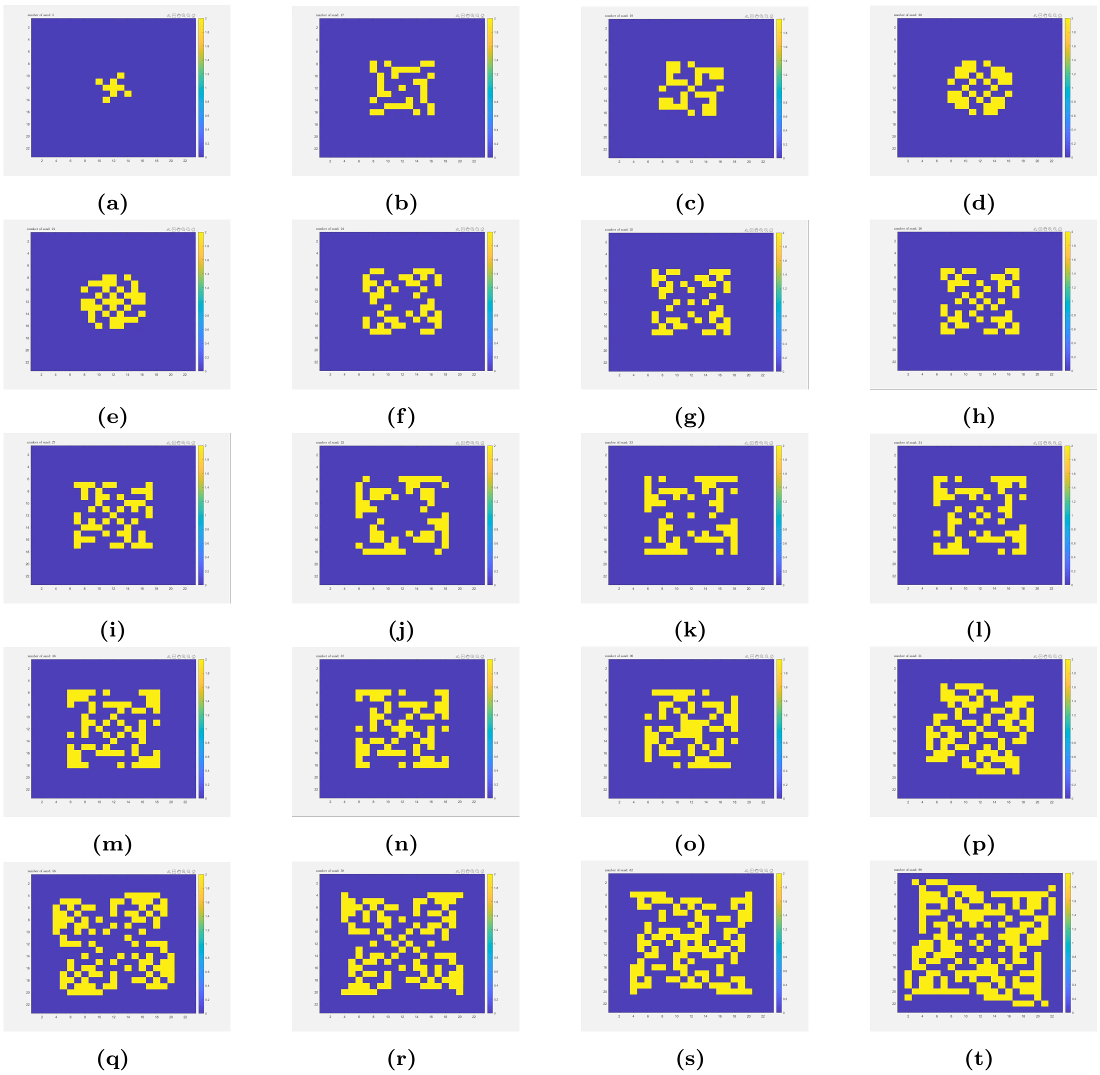


Figure 3.15: More plots of the rotation lighthouse sandpile model show the process of growth of it. From (a), the very early stage of the model to (t), the very late stage of the model. We can see as the time steps increase, the pattern of the sandpile changes as the rotation. It is better to look at the video to see the continuous process of this model.

Clearly, we can see the generation of our rotation lighthouse model in the above plots, Figure 3.15 (a) to (t). We can see that the pattern of these plots is like a rotation galaxy with a spiral, the spiral galaxy. The process of growth of the sandpile with different structures looks like the galaxy in different lifetimes.

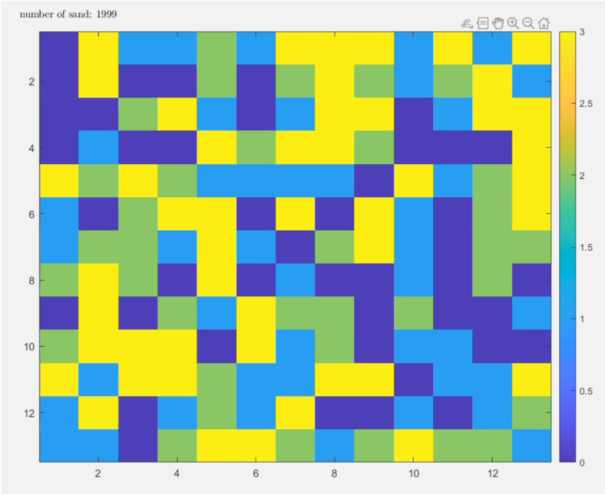
At this point, I started to think about whether is possible to relate our sandpile model to the star formation model or if is it possible to build up a star formation model based on the sandpile model. More about this will be discussed in the next chapter (chapter 4, the second part of our extended models which are in the polar coordinate table).

For now, we should continue our extension model for the 2-dimensional sandpile model.

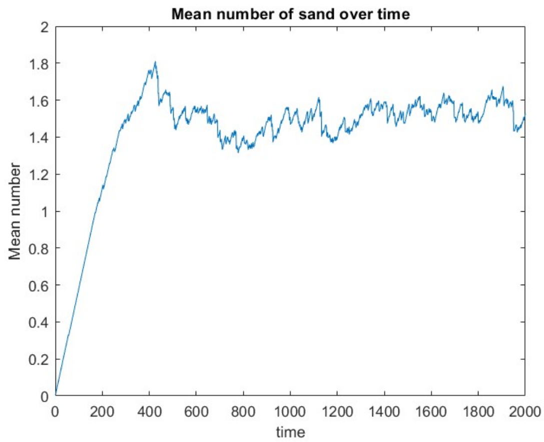
3.6 Rotation lighthouse model (random adding)

Now, we build a lighthouse model with rotation but we add sand in the table for every time step randomly rather than always adding sand in the center of the table. In this new situation, we add one sand to the neighbours of the topple site. The other part of the algorithm is the same as in section 3.5.

We simulate a 13 x 13 table with 2000 time steps in this section, the results are shown below.



(a) 13x13 table with 2000 time steps random rotation lighthouse sandpile plot

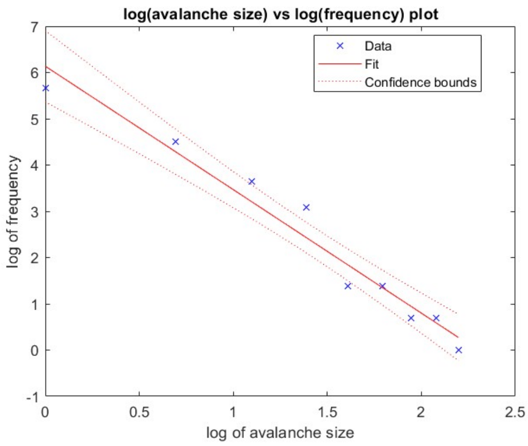


(b) 13x13 table with 2000 time steps random rotation lighthouse mean height plot

Figure 3.16: resulting sandpile plot and the mean height plot: From (a), we can see there is not a recognizable symmetry pattern now since our adding sand randomly to the table. In (b), we can see the mean height is going to be more stable after 1000 time steps and around 1.5 units of height.

From the above plot, we cannot see a clear pattern again in this randomly added model. However, for the mean height plot, we can see it is relatively stable around 1.5 mean height compared to the model in the last section.

Then, the log of the avalanche size versus the log of its frequency plot and the corresponding coefficients table of the best-fitting line by using linear model are shown below.



(a) The log of the avalanche size vs the log of its frequency plot

2×4 table

	Estimate	SE	tStat	pValue
(Intercept)	6.1363	0.32534	18.861	2.9272e-07
x1	-2.6691	0.20646	-12.928	3.8512e-06

(b) The log of the avalanche size vs the log of its frequency plot coefficients table

Figure 3.17: The log of the avalanche size vs the log of its frequency plot and the corresponding coefficients table: from (a), the resulting plot shows that there exists a linear relationship between the log of the avalanche size and the log of its frequency, which implies there exist a power law relation between the avalanche size and its frequency. However, from (b) we can see that the SE of the coefficients of the best-fitting line is relatively large. This is due to the table size we used being small and the time steps we used being small, hence we do not record enough data points for the plot (a), and hence the linear relation between the avalanche size and its frequency is not very significant in this case.

From the above Figure 3.17 (a), the log of avalanche size versus the log of its frequency plot, we can see there is evidence to exist a power law distribution relation between them. However, since the table is only 13x13 and the time steps are less, we cannot see many data points in the plots. The data points in the plots almost follow the linear relation exactly and the corresponding coefficients are shown in the above table, Figure 3.17 (b). Next section, we will discuss a new sandpile model.

3.7 Model with add on outer 4 corners

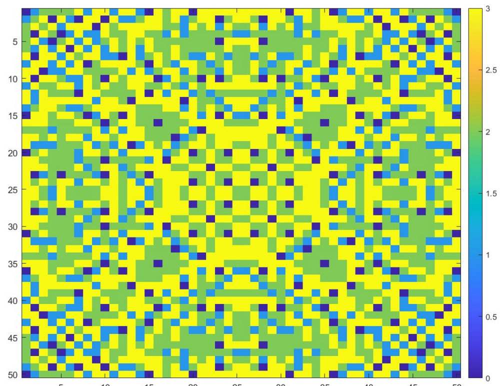
From this section, we start to build a new extended sandpile model that at each time step, we add 2 sand in each outer corner of the table. This model is completely different from the model we built before. That is, we will let:

$$Z(1,1) = Z(1,1) + 2$$
$$Z(1,N) = Z(1,N) + 2$$
$$Z(N,1) = Z(N,1) + 2$$
$$Z(N,N) = Z(N,N) + 2$$

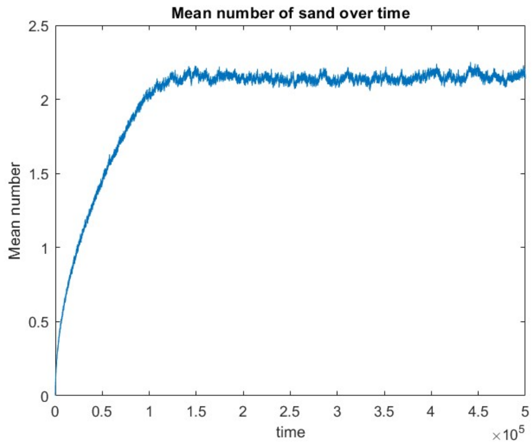
for every time step.

The reason for adding 2 sands is we want the sandpile at four corners to grow faster and interact quickly. This model is a bounded model. For topples, we decide to let the neighbours of the topple site only add 1 sand. The other parts of this algorithm are the same as the basic sandpile model (bounded) that we discussed in Chapter 2. We can change to add 1 sand at each time step and we can see if the table is large, the four regions of sandpile are difficult to meet each other, they are hard to interact.

We simulate this model with a 50 x 50 table and 500000 time steps as shown below.



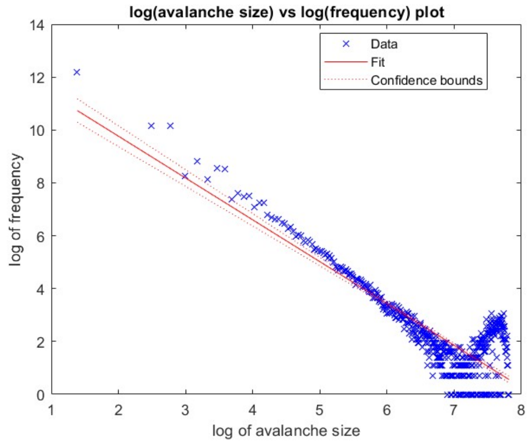
(a) 50x50 table with 500000 time steps outer four corners sandpile plot



(b) 50x50 table with 500000 time steps outer four corners sandpile mean height plot

Figure 3.18: resulting sandpile plot and the mean height plot: from (a), we can see a symmetry pattern that symmetry about the middle of the table. From (b), we can see that the mean height increases from 0 to around 100000 time steps and then it becomes stable around 2 units height.

From Figure 3.18 (a), we can see a special pattern for this model. The result pattern behavior symmetry about the center. From the mean height plot, Figure 3.18 (b), we can see the mean height is stable at around 2 units height, this indicates this model is in a critical state after about 100000 time steps. Then, the log of the avalanche size versus the log of its frequency plot and the corresponding coefficients table of the best-fitting line using the linear model are shown below.



(a) The log of the avalanche size vs the log of its frequency plot

2×4 table

	Estimate	SE	tStat	pValue
(Intercept)	12.931	0.28291	45.705	1.4172e-181
x1	-1.5828	0.041937	-37.743	4.1711e-149

(b) The log of the avalanche size vs the log of its frequency plot coefficients table

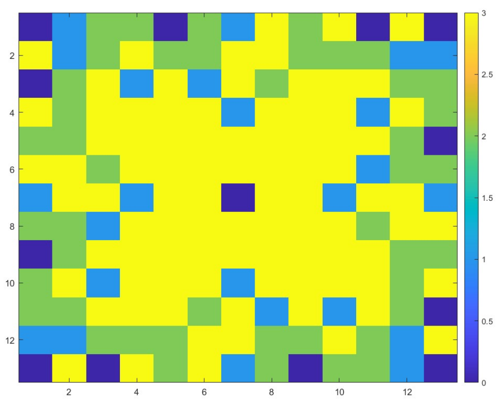
Figure 3.19: The log of the avalanche size vs the log of its frequency plot and the corresponding coefficients table: we can see that most of the data points in (a) lie linearly, which indicates that there exists a linear relationship between the log of the avalanche size and the log of its frequency. This further indicates there exists a power law relation between the avalanche size and its frequency in this new model.

From the above log of avalanche size and the log of its frequency plot, Figure 3.19 (a), we can see there exists a power law distribution relationship between them with strong evidence for this model. The corresponding coefficients are shown in the table, Figure 3.19 (b).

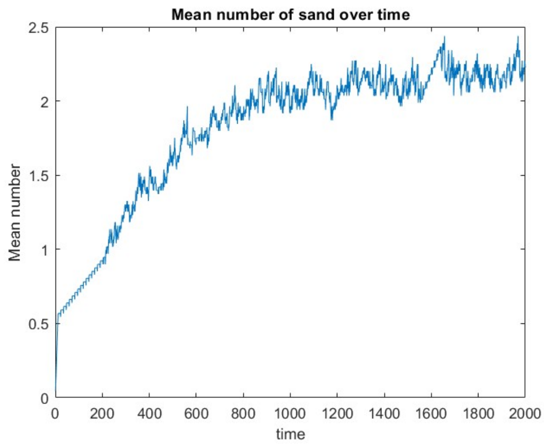
One thing we should mention here is we can see the growth of 4 clusters of the sandpile and their interactions in the corresponding video in the file.

3.8 Model with add on outer 4 corners with rotation

In this section, we add a rotation of the model in the last section. We can see from the resulting video, that this model provides a similar plot to the plots we shown in section 3.5, Figure 3.15. This model is like a "complement set" of the model in section 3.5 which is a very interesting result. We can see this in the corresponding video and a more beautiful pattern can be shown if the table used to simulate the model becomes larger and the number of time steps becomes larger.



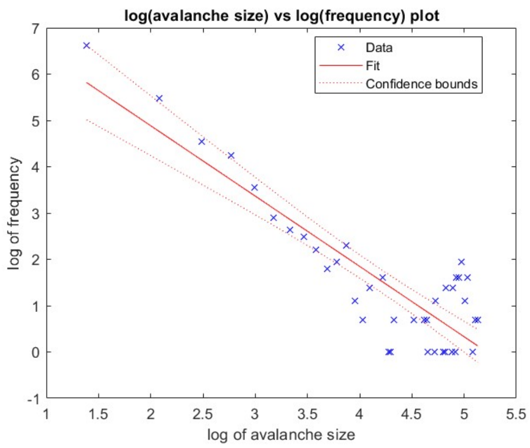
(a) 13x13 table with 2000 time steps outer four corners rotation sandpile plot



(b) 13x13 table with 2000 time steps outer four corners rotation sandpile mean height plot

Figure 3.20: resulting sandpile plot and the mean height plot: we can see from (a), that the pattern looks like the "complement set" of the model in section 3.5 has been shown, the rotation lighthouse model. From (b), we can see the mean height is going to be more stable after 1600 time steps, the mean height of the sandpile should become more stable if we use more time steps.

From the above plot, Figure 3.20 (a), a pattern like the "complement set" of the model in section 3.5 has been shown. Figure 3.20 (b), the mean height plot of this model is shown above and we can see the mean height of this model is around 2 when the model is in a critical state. Then, we will provide the log of the avalanche size versus the log of its frequency plot and the corresponding coefficients table as shown below.



(a) The log of avalanche size vs the log of its frequency plot

2×4 table

	Estimate	SE	tStat	pValue
(Intercept)	7.9213	0.57768	13.712	4.4387e-16
x1	-1.5185	0.13511	-11.239	1.7304e-13

(b) The log of avalanche size vs the log of its frequency plot coefficients table

Figure 3.21: The log of avalanche size vs the log of its frequency plot and the corresponding coefficients table: from (a), similar to the case in section 3.7, we can see most of data points shown a linear relation between the log of the avalanche size and the log of its frequency which indicate there exist a power law relation between the avalanche size and its frequency.

Although the table is relatively small and the time steps are less, from Figure 3.21 (a), we can see a power law distribution relationship still exists between the avalanche size and its frequency. The corresponding coefficients table is shown above in Figure 3.21 (b).

3.9 Model with adding sand on outer edges

Now, in this section, we build a sandpile model that adds sand on outer edges. That is, we will let:

$$Z(1, 1:N) = Z(1, 1:N) + 1$$

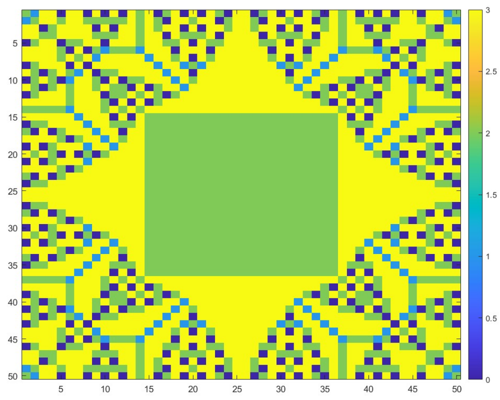
$$Z(1:N, 1) = Z(1:N, 1) + 1$$

$$Z(N, 1:N) = Z(N, 1:N) + 1$$

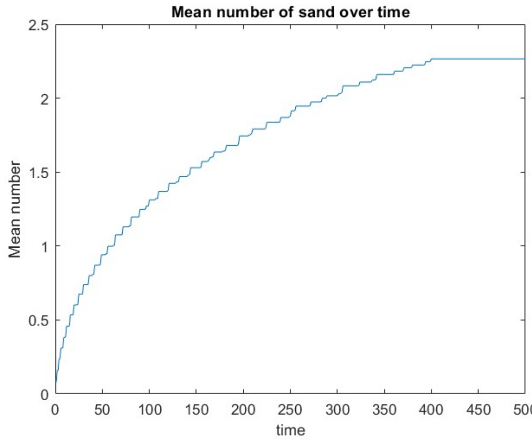
$$Z(1:N, N) = Z(1:N, N) + 1$$

for every time steps. The other parts of this model are the same as the model in section 3.7, the model with add an outer 4 corners.

We simulate this model two times in order to see what will happen to the sandpile pattern, the first one with a 50 x 50 table with 500 time steps. The second one with a 100 x 100 table with 2000 time steps.



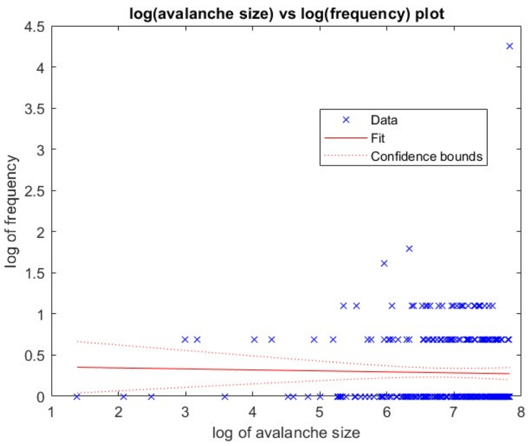
(a) 50x50 table with 500 time steps outer edges rotation sandpile plot



(b) 50x50 table with 500 time step outer edges rotation sandpile mean height plot

Figure 3.22: result sandpile plot and the mean height plot: We can see from (a) the pattern is symmetry and there exist a square in the middle of the sandpile pattern. From (b), the mean height become stable after 400 time steps and the mean height is around 2.7 units.

From the resulting plot, Figure 3.22 (a), we can see there is a rectangle or say, square in the middle of the plot and a symmetry pattern abound it. From the mean height plot, Figure 3.22 (b), we can see the mean height becomes stable after 400 time steps which implies the sandpile is in a critical state. Then, the log of the avalanche size versus the log of its frequency plot with the corresponding coefficients table is shown below.



(a) the log of the avalanche size versus the log of its frequency plot

2×4 table

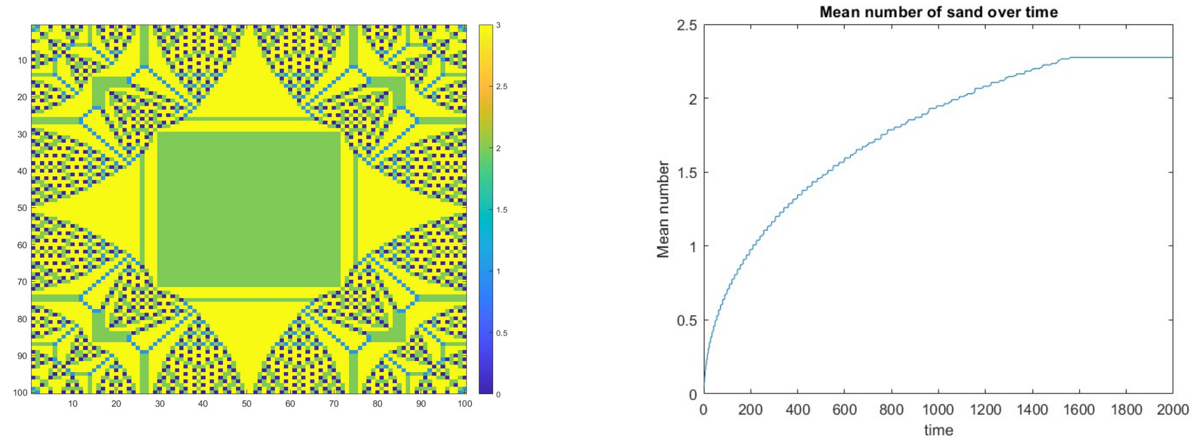
	Estimate	SE	tStat	pValue
(Intercept)	0.36867	0.19744	1.8673	0.062853
x1	-0.012039	0.028371	-0.42434	0.67162

(b) the log of the avalanche size versus the log of its frequency plot coefficients table

Figure 3.23: the log of the avalanche size versus the log of its frequency plot and the corresponding coefficients table: from (a), we can see there does not exist any linear relation between the log of the avalanche size and the log of its frequency, therefore, there do not exist a power law relation between the avalanche size and its frequency in this model.

However, we cannot see any linear relationship between the log of avalanche size and the log of its frequency from Figure 3.23 (a). This implies there does not exist a power law distribution relationship between the avalanche size and its frequency.

Now, let us look at another simulation with a larger table and more time steps.

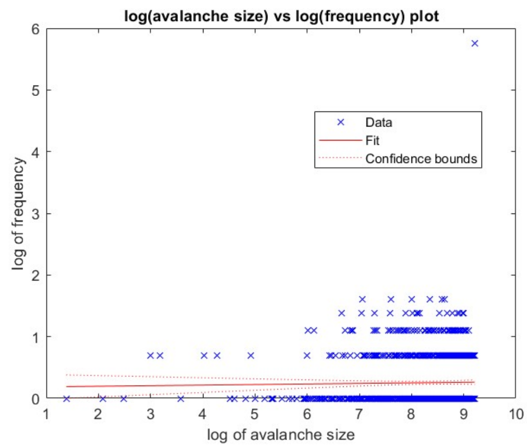


(a) 100x100 table with 2000 time steps outer edges rotation sandpile plot

(b) 100x100 table with 2000 time steps outer edges rotation sandpile mean height plot

Figure 3.24: resulting sandpile plot and the mean height plot: similar to Figure 3.22 (a), a symmetry sandpile with a big square in the middle of the table from (a). From (b), the mean height becomes stable after about 1500 time steps and the mean height is around 2.3.

Then, the log of the avalanche size versus the log of its frequency plot and the corresponding coefficients table are shown below.



2×4 table

	Estimate	SE	tStat	pValue
(Intercept)	0.17747	0.11577	1.533	0.12554
x1	0.0092414	0.0139	0.66486	0.50627

(a) log of the avalanche size versus the log of its frequency plot

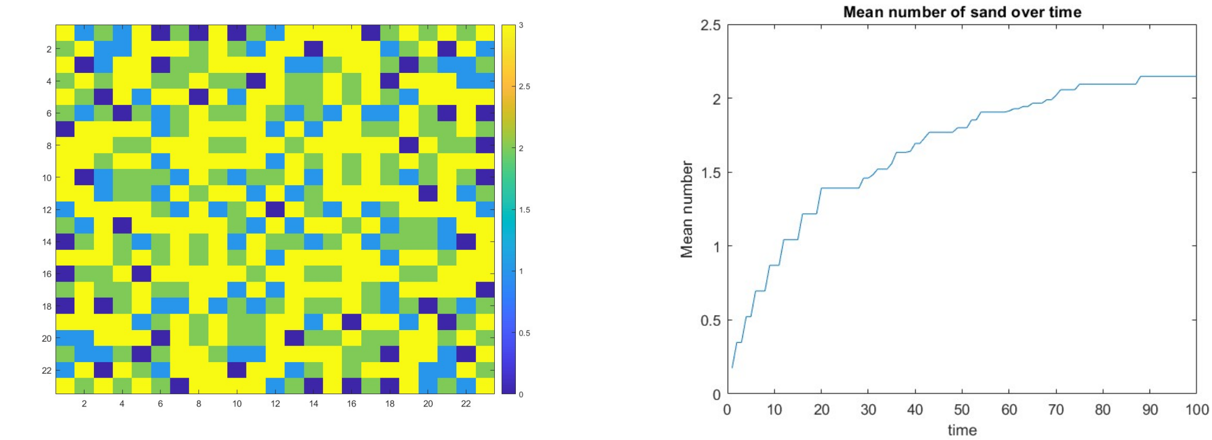
(b) log of the avalanche size versus the log of its frequency plot coefficients table

Figure 3.25: log of the avalanche size versus the log of its frequency plot and the corresponding coefficients table: from (a), data points do not show a linear relationship which implies that there does not exist a power law relation between the avalanche size and its frequency.

Similarly, from the above plots, Figure 3.25 (a), we can see a similar pattern and there does not exist a power law distribution relationship in this model between the avalanche size and its frequency.

3.10 Model with add on outer all edges with rotation

In this section, we build a new model that is the model in the last section with rotation. This model will give us information about what the symmetry pattern with a big square in the middle will be like in rotation. The resulting plot of the sandpile and the mean height plot are shown below.

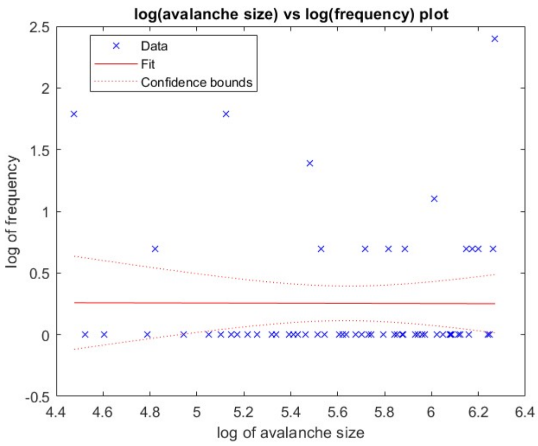


(a) 23x23 table with 100 time steps outer edges rotation sandpile plot

(b) 23x23 table with 100 time steps outer edges rotation sandpile mean height plot

Figure 3.26: resulting sandpile plot and the mean height plot: we can see the pattern is similar to Figure 3.20 (a) from (a). From (b), we can see the mean height of the sandpile becomes stable after about 90 time steps and the mean height is 2.1 units when stable.

From the above plot, Figure 3.26 (a), we can see a there exists a regular pattern of the sandpile but is different from the pattern in the results in the last section. For the mean height plot, Figure 3.26 (b), we can see its increase rate slow after 80 time steps and it should be stable after more time steps. Next, the log of the avalanche size versus the log of its frequency plot and the corresponding coefficients table are shown below.



2×4 table

	Estimate	SE	tStat	pValue
(Intercept)	0.27787	0.85063	0.32666	0.74514
x1	-0.0043159	0.15033	-0.028709	0.9772

(a) The log of the avalanche size versus the log of its frequency plot

(b) The log of the avalanche size versus the log of its frequency plot coefficients table

Figure 3.27: The log of the avalanche size versus the log of its frequency plot and the corresponding coefficients table: clearly, there does not exist a linear relation between the log of the avalanche size and the log of its frequency in this model which indicates that there does not exist a power law relation between the avalanche size and its frequency in this model.

From the above log of avalanche size versus the log of its frequency plot, Figure 3.27 (a), we cannot see there is a power law relationship between them.

3.11 Further Models that with changes in middle site

There are some further extensions we did for the extended sandpile model. Firstly, add sand in add sites of outer edges with the following new topple rule (all other parts are same as the Model with add on outer all edges with rotation in section 3.10), that is:

We check the position of the topping, and if

$$x_{tops}(j) + 1 = N/2 \quad and \quad x_{tops}(j) - 1 = N/2$$

Then, we should let:

$$Z(x_{tops}(j) + 1, y_{tops}(j)) = Z(x_{tops}(j) + 1, y_{tops}(j)) + 0$$

$$Z(x_{tops}(j)-1, y_{tops}(j)) = Z(x_{tops}(j)-1, y_{tops}(j)) + 0$$

We check the position of the topping, and if

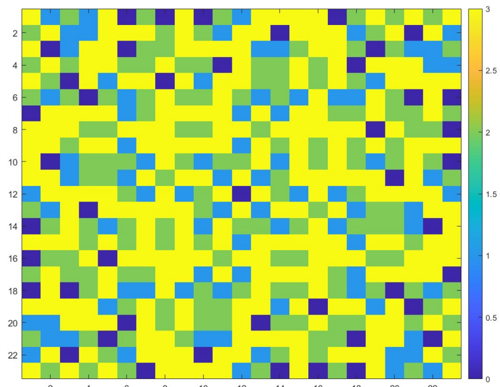
$$y_{tops}(j)+1 = M/2 \quad and \quad y_{tops}(j)-1 = M/2$$

Then, we should let:

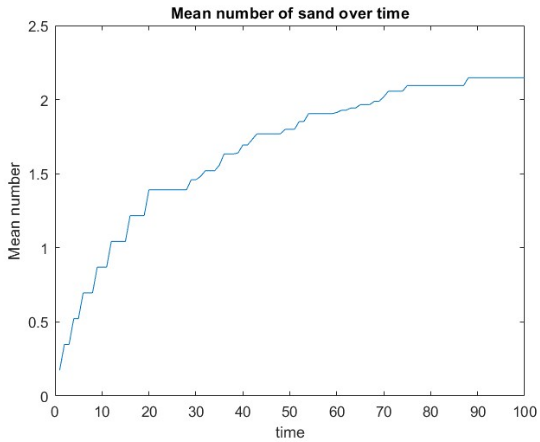
$$Z(x_{tops}(j), y_{tops}(j)+1) = Z(x_{tops}(j), y_{tops}(j)+1) + 0$$
$$Z(x_{tops}(j), y_{tops}(j)-1) = Z(x_{tops}(j), y_{tops}(j)-1) + 0$$

These are the new things we will introduce in this model.

The resulting sandpile plot and the mean height plot are shown below.



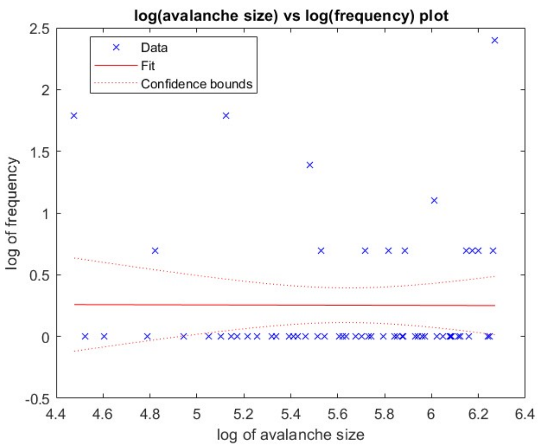
(a) 23x23 table with 100 time steps outer edges rotation sandpile plot



(b) 23x23 table with 100 time steps outer edges rotation sandpile mean height plot

Figure 3.28: resulting sandpile plot and the mean height plot

The log of the avalanche size versus the log of its frequency plot and the corresponding coefficients table are shown below.



(a) The log of the avalanche size versus the log of its frequency plot

2×4 table

	Estimate	SE	tStat	pValue
(Intercept)	0.27787	0.85063	0.32666	0.74514
x1	-0.0043159	0.15033	-0.028709	0.9772

(b) The log of the avalanche size versus the log of its frequency plot coefficients table

Figure 3.29: The log of the avalanche size versus the log of its frequency plot and the corresponding coefficients table

Secondly, we changed our rule slightly. We add sand in add sites of outer edges with the following new topple rule:

We check the toppling position and if:

$$x_{tops}(j)+1 = N/2$$

Then we will let:

$$Z(x_{tops}(j)+1, y_{tops}(j)) = Z(x_{tops}(j)+1, y_{tops}(j)) + 0$$

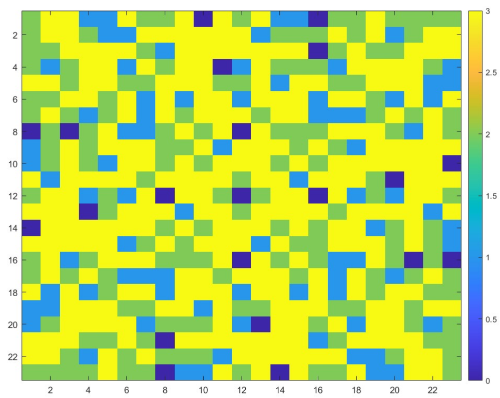
We check the toppling position and if:

$$y_{tops}(j) + 1 = M/2$$

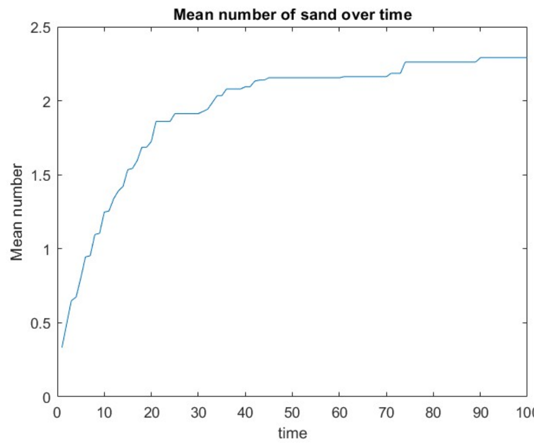
Then we will let:

$$Z(x_{tops}(j), y_{tops}(j) + 1) = Z(x_{tops}(j), y_{tops}(j) + 1) + 0$$

Other parts of the codes will remain the same as before. The resulting sandpile plot and the mean height plot are shown below.



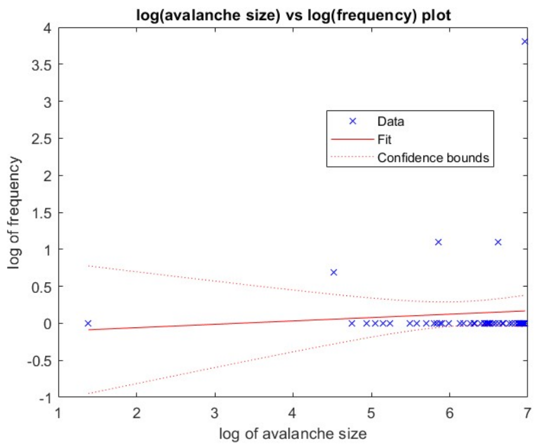
(a) 23x23 table with 100 time steps outer edges rotation sandpile plot



(b) 23x23 table with 100 time steps outer edges rotation sandpile mean height plot

Figure 3.30: resulting sandpile plot and the mean height plot

The log of the avalanche size versus the log of its frequency plot and the corresponding coefficients table are shown below.



(a) The log of the avalanche size versus the log of its frequency plot

2×4 table

	Estimate	SE	tStat	pValue
(Intercept)	-0.15011	0.54886	-0.27349	0.78563
x1	0.045553	0.087869	0.51842	0.6065

(b) The log of the avalanche size versus the log of its frequency plot coefficients table

Figure 3.31: The log of the avalanche size versus the log of its frequency plot and the corresponding coefficients table

In the above two cases, from Figure 3.28 (a) and Figure 3.30 (a), we can see there exist some patterns for the sandpile but again, there does not exist a power law relationship between the avalanche size and its frequency from Figure 3.29 (a) and Figure 3.31 (a).

This is the end of our extended models in this chapter, we still have many trial models with codes but not shown here, we included them in the resources file.

3.12 A new Matlab Matrix rotation algorithm

For the purpose of model building, we need an algorithm for matrix rotation. We do lots of searches on this but we do not find any existing algorithm or package by using Matlab that can do this. There are some related algorithms, however, they are not what we want for this project. Hence, we develop a new algorithm for matrix rotation as shown below.

```
empty_sandpile=zeros(N,M);

for t=1:((((N)+1)/2)-1)
    M2 = sand_pile(t:(N-(t-1)),t:(M-(t-1)));
    Length_matrix=length(M2);
    M1=zeros(length(M2),length(M2));
    [x_coord,y_coord]= find(M2>=0);

    for k=1:length(x_coord)
        if (x_coord(k) == 1) && (y_coord(k) == 1)
            M1(x_coord(k),y_coord(k)+1) = M2(x_coord(k),y_coord(k));
        elseif (x_coord(k) == 1) && (y_coord(k) == Length_matrix)
            M1(x_coord(k)+1,y_coord(k)) = M2(x_coord(k),y_coord(k));
        elseif (x_coord(k) == Length_matrix) && (y_coord(k) == Length_matrix)
            M1(x_coord(k),y_coord(k)-1) = M2(x_coord(k),y_coord(k));
        elseif (x_coord(k) == Length_matrix) && (y_coord(k) == 1)
            M1(x_coord(k)-1,y_coord(k)) = M2(x_coord(k),y_coord(k));
        elseif (x_coord(k) == 1) && (y_coord(k) ~= 1) && (y_coord(k) ~= Length_matrix)
            M1(x_coord(k),y_coord(k)+1) = M2(x_coord(k),y_coord(k));
        elseif (y_coord(k) == Length_matrix) && (x_coord(k)~=1) && (x_coord(k)~=Length_matrix)
            M1(x_coord(k)+1,y_coord(k)) = M2(x_coord(k),y_coord(k));
        elseif (x_coord(k) == Length_matrix) && (y_coord(k) ~= Length_matrix) && (y_coord(k) ~= 1)
            M1(x_coord(k),y_coord(k)-1) = M2(x_coord(k),y_coord(k));
        elseif (y_coord(k) == 1) && (x_coord(k)~=1) && (x_coord(k)~=Length_matrix)
            M1(x_coord(k)-1,y_coord(k)) = M2(x_coord(k),y_coord(k));
        end
    end
empty_sandpile(t:(N-(t-1)),t:(N-(t-1)))=M1;
empty_sandpile((N+1)/2,(N+1)/2)=sand_pile((N+1)/2,(N+1)/2);
end

sand_pile=empty_sandpile;
```

The description of this algorithm:

1. we build an empty table with the same width and length as the table that we used to perform the simulation.
2. Then, we consider every ”lap” of the matrix one by one.
3. We move every single site in a lap of the matrix one site clockwise.
4. We combine all the laps after moving together to become a matrix again. We store it in our empty matrix.
5. we define our sandpile to be the ”empty matrix” now. (The empty matrix is not empty at this time, it is actually the sandpile matrix that moves all of the sites one site clockwise.)
6. Now, we get a new sandpile after the rotation.

We should note that, in this algorithm, the N and M must be integer numbers like 3,5,7,9,11,13... It cannot be even a number otherwise it will not work. We did build a different algorithm for an even number, however, we want the table to have a center in the middle, hence we will always use odd numbers to be our table length and table width when we involve rotation. In all the above models that involved rotation, we rotate the sandpile in the table every single time step, however, it is easy to define different time intervals of rotation. For example, we can change to rotate the sandpile every 10-time steps by adding a checking condition ”mod(timesteps,10)”, if this is equal to 0, then we rotate the sandpile. Overall, by using this general algorithm, we can always add rotation to the sandpile model to see what will be different.

3.13 Chapter summary

In summary, we developed several different models in this chapter with different rules, such as the lighthouse model, and the model with adding sand at the edges of the table. We analysis some properties of them, mainly the avalanche size and its frequency. We found that in some of the model, the avalanche size and its frequency follows a power law distribution while some of them does not follow this relation. Finally, we define an algorithm for matrix rotation and we apply this algorithm to our models to see how they behave when there exists a rotation of the sandpile. We will discuss the sandpile model in the polar coordinates system table in the next chapter.

Chapter 4

Extending The BAK-TANG-WIESENFELD Model (Part 2)

In this chapter of the report, we will mainly discuss the models built by extending the Rotation lighthouse model (adding sand in the center) that we discussed in the last chapter. We were inspired by this model and the Matrix rotation algorithm, therefore, we further built the rotation models in this chapter. Firstly, we will try to extend the 2-dimensional sandpile model to the polar coordinate system. After that, we will try to develop a rotational polar sandpile model. Then, we will try to extend the polar coordinate system sandpile model to build a model for star formation of spiral galaxies.

For this Chapter, the results will be presented according to the order in which these models are built. We did a lot of work for this chapter, hence we will present them in different series and we will skip most of them and only focus on the important ones.

Since I am interested in construction models based on self-organization and related them to the formation of the galaxy and the structure of the galaxy, I found there already exist a few self-organization star formation models that are mentioned in the articles Aschwanden et al. (2016), Aschwanden (2014) and Aschwanden (2011), mainly the stochastic self-propagating star formation (SSPSF) model that first introduced in the work of Mueller and Arnett (1976) and further constructed by the work of Gerola and Seiden (1978) and Gerola et al. (1980).

A few years later, Seiden and Schulman (1990) built a Percolation model of galactic structure based on the SSPSF model and discussed the properties of the SSPSF model in detail. The SSPSF model, percolation, and the sandpile model are strongly related to each other, therefore, we believe that we can build a new sandpile galaxy formation model based on previous works in this area that can be used to simulate the star (or say, the star cluster since we can think one ”point” in our polar table as a single star or a cluster of stars.) formation process in the universe. Algorithms in the later of this chapter (mainly section 4.2) will follow the idea of Gerola and Seiden (1978).

4.1 Sandpile model (and the rotational sandpile model) in the polar system by the first algorithm

4.1.1 Algorithm part

At the beginning of this chapter, we will first introduce the sandpile model in the polar coordinate system using the first algorithm. We built 9 models (and many trial models) for this section, we called them ”galaxy1” to ”galaxy9”. These works are done in the early stage of the project.

The main idea is, that we use a polar table with a center constructed by using many concentric circles around the center. We denoted each concentric circle be a ring. Each ring is divided into different amount of sites, the number of sites in each ring depends on the radius of the ring to the center. For the first ring, the radius is set to 1 unit and the number of sites is 6. For the second ring, the radius is set to 2 units and the number of sites is 12. For the third ring, the radius is set to 3 units and the number of sites is 18, and so on. Then, we use the radius and the angle theta to the positive x-axis to determine the position of each site. The 0 degree is the positive x-axis and the degree increase from 0 degree to 360 degree (which is 0 degree again) anticlockwise. That is, for each ring i where i denotes the radius of the ring to the center from 0 to N (the total rings we considered in the model, this depends on the capacity of the computer used to simulate the model) and it also denotes the name of the ring, we define $N_i = i$ and vector for theta th_i from $(2\pi)/(N_i \cdot 6)$ to 2π

with interval $(2\pi)/(N_i \cdot 6)$. The length of the vector of theta is the number of sites this ring contained. Then, we define another vector for the radius associated with each site of each ring, the radius is the same for sites in the same ring.

Then, we define a height $Z(r,\theta)$ vector that records the height of each site in our polar table, initially, all heights are 0. Then, in this algorithm, we define a big matrix that combines the radius vector, the theta vector, and the height vector.

$$sandpile = \begin{bmatrix} theta \\ radius \\ height \end{bmatrix}$$

Then, for each iteration, we randomly choose a site in the ring and add a grain of sand to it, $Z(r,\theta) = Z(r,\theta) + 1$. Then, we check whether there exists any site with a height greater than critical value 3 that is $Z(r,\theta) > 3$. If exist, we denoted them as topples sites, then for each of them, $Z(r_{top},\theta_{top}) = Z(r_{top},\theta_{top}) - 4$ and the height of the neighbours of the topples site will plus 1 (grain of sand). The definition of neighbours in this chapter is different from the definition from previous chapters since we are using a polar table now. We cannot simply say the sites around the topples site, hence, we use a formula to calculate the distance from the topples site to all other sites in the table.

$$d = \sqrt{r_i^2 + r_{top}^2 - 2r_i^2 r_{top}^2 \cos(\theta_i - \theta_{top})}$$

Then, we compare all the distances and then check which of them less than a critical distance value, and these sites will increase their height by 1. Note here that we need to convert the angle from radians to degrees first in order to use the formula.

We find that comparing all distances is wasting our computation resources after the simulations, hence, we can use "mink(d,u)" to give the closer "u" sites with their index, and d is the distance. This will be used for most of our models in this chapter.

We should note here that we will use a more reasonable algorithm later that separates all of our rings and only combines them when plotting the figure. This will increase the response speed and be more reasonable when we consider rotation. We will discuss this algorithm in section 4.2.

For introducing the rotation to our models, we use the function of Matlab "$th_i = circshift(th_i, [0, v])$". This means we shift all of theta for each ring by v unit. Then we combine thetas of each ring to a new big matrix "sandpile" again. By using this method, we can add the rotation into our models. Introducing rotation in a polar system table is much easier than in a 2D square table which we discussed in chapter 2 and chapter 3, we only need to consider moving the theta clockwise or anticlockwise of sites now.

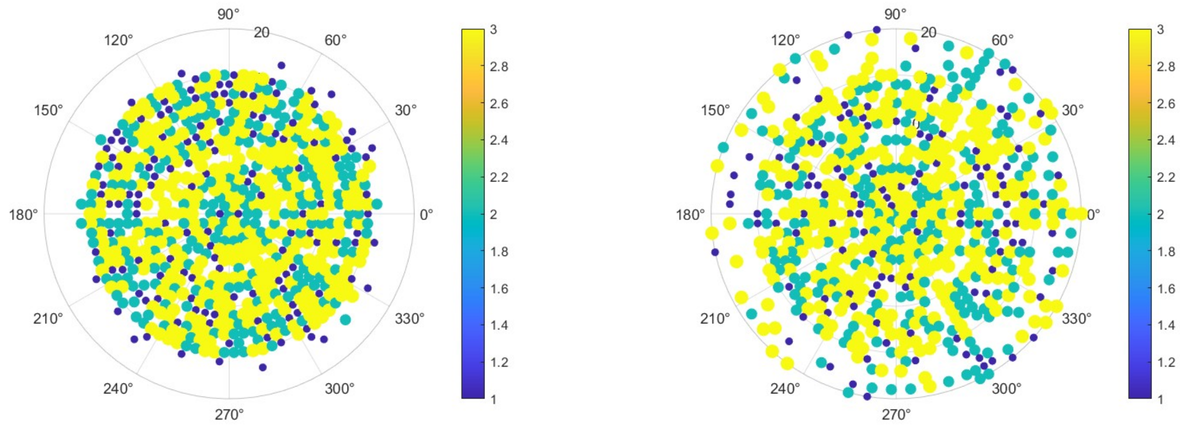
4.1.2 Results part

For the result section in this section and the next sections, we only focus on the resulting plots of different models. We will not discuss the properties of these models in this report and we will not provide videos of most of them. This is due to the time limit of this project and also we do not want our report to become so long, however, we should mention here, that adding the analysis part to all of the models we will talk about is possible and simple, we already construct codes for analysis, they can be used here in the same way we did before for previous chapters. Besides that, we can use the same code for making videos here as before but again, due to the same reasons, we did not do this in this report. We will only focus on the phenomena approach in this report.

we will ignore the first two models "galaxy1" and "galaxy2" here since they are the first two basic models we built and are not very interesting. In addition, the plotting method used in these two models is not suitable. Therefore, we will begin by "galaxy3".

Galaxy3 model (The Basic Polar Sandpile Model):

The galaxy3 model has 20 rings (plus the center site (the "0" ring), we will always have the center site for most cases) and we set 1500 time steps now. However, we only add sand inside the first 10 rings randomly in this model rather than randomly adding sand in all sites. The reason is we want to see topples earlier as our computer cannot do more time steps but we do not want the sand only added in the center of the table. Therefore, we chose to add sand to the first 10 rings, the inner half of the polar table. Rotation for this model is v=[0 1 3 6 8] for the first 5 rings and then v=10 for all other rings. As we talked about before, v denotes how many sites it shifted in the ring. Furthermore, we also did a simulation without rotation to check the differences. The results are shown below.



(a) The galaxy3 model with rotation

(b) The galaxy3 model without rotation

Figure 4.1: The result plots of galaxy3 model with and without rotation. We can see in (a) that the polar sandpile model with rotation, looks more like a regular circle. If we remove the rotation, see (b), we can see the pattern looks more irregular, with many "outlier" sites. It's like a lot of points have been thrown out of the main structure.

In the plot of this model, we use both the size and the colours of the sites to denote the height of the site. Yellow colour represents the highest and the size of the site bigger means higher.

From Figure 4.1, we can clearly see a sandpile structure that is similar to the basic sandpile model in the 2D table we discussed before in Chapter 2 and Chapter 3.

One remarkable phenomenon is that the galaxy3 model with rotation seems to restrict the sites to be a pattern similar to a circle, however, when it is without rotation, there are a lot of points that seem to have been "thrown" out. The reason we only use 1500 time steps is if we use more time steps, the response of the computer becomes very slow when the time steps increase. The reader can change the code to see what happens when the time steps increase and we randomly drop sand at all sites without any restriction or only drop sand in the center of the polar table exactly. We forecast the results will be similar in our chosen case and similar to the basic sandpile model. Furthermore, if we change the time steps to be larger or allow adding sand randomly, then we should be able to see a pattern similar to the basic sandpile since we already do some simulations for fewer rings with the same adding rule and same time steps and we can see a pattern similar to basic sandpile model.

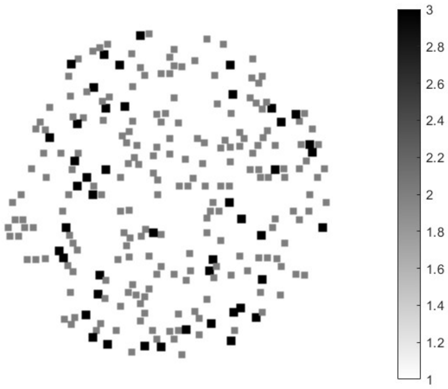
We conclude that we extended the basic sandpile model from the 2D table to the polar coordinate table successfully by the galaxy3 model. We can call it the basic polar table sandpile model.

Galaxy4 model:

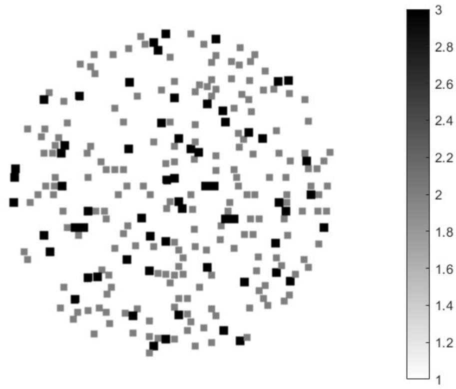
Based on the galaxy3 model, we built a new model (the galaxy4 model), and we introduced a new vector (active vector) to record the active or inactive of each site. Following the works of Gerola and Seiden (1978) and Seiden and Schulman (1990), we want to extend their models and build a self-organization stochastic star formation sandpile model. In their models, active or inactive is a useful parameter therefore we will include them here.

If a site has increased the height then it will be marked as an inactive site by giving a regeneration time to the corresponding position of the active vector. For every time step, we check which sites are inactive, then all inactive sites' regeneration time minus 1 time step. If the regeneration time is less or equal to 0, then this site becomes active again. Then, we check which sites are active, and we randomly add a grain of sand to one of the active sites, then mark it inactive. For the rules about topples, the induced sites caused by topples sites must be active to add 1 height. the height of inactive neighbours will not increase. The remaining parts of the algorithm are the same as the galaxy3 model.

The parameters we have chosen in the below results are 1000 time steps, 20 rings (and the center site). The neighbours are the closest 4 sites (do not consider the topple site itself). The regeneration time is 100 time steps. The critical value for topples is the height greater than 3. For the rotation one v1=0; v2=1; v3=2; v4=3; v5=4; v6=5; v7=6; v8=7; v9=8; v10=9; v11=11; v12=12; v13=13; v14=13; v15=14; v16=14; v17=14; v18=15; v19=15; v20=15; v21=15. The definition of v are same as the previous one. The results of the simulations are shown below.



(a) The galaxy4 model with rotation



(b) The galaxy4 model without rotation

Figure 4.2: The result plots of galaxy4 model with and without rotation. We can see that after adding the new parameters (active and inactive), the sandpile appears more sparse. Rotation doesn't seem to make a huge difference in this model.

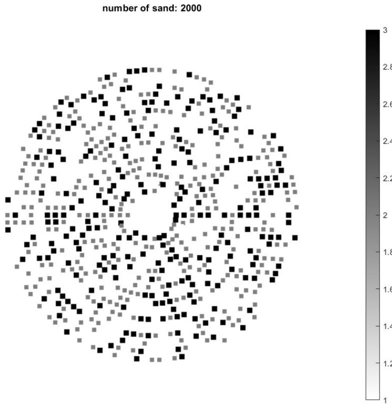
We use the square markers to denote the sites here, and we use greyscale here. The colour and the size of the sites are depended on their height. From Figure 4.2, We can see that the space between the sites gets a lot bigger than the results of the galaxy3 model (Figure 4.1). This is due to the introduction of the regeneration time. Since we randomly add sand in active sites, the position added is completely determined by active or not, therefore, the rotation or not does not affect the results significantly. The introduction of new parameters to make our sandpiles appear spaced is in line with the idea of galaxy-generating models that there should be a lot of space between stars, rather than the entire galaxy being filled with stars everywhere.

Galaxy5 model:

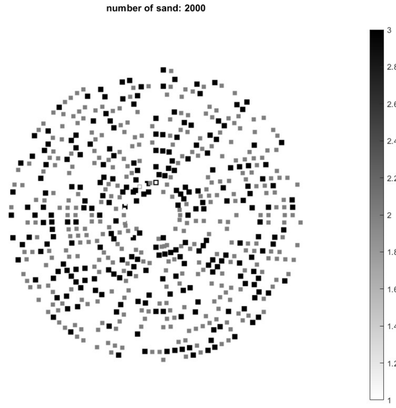
In this model, we further explore the galaxy4 model and built the galaxy5 model. We choose the dropping regeneration time to be 0 time steps and the topping regeneration time to be 200 time steps here. This is because we wanted to experiment with how much and how different combinations of parameters affect our sandpile. The number of time steps is 2000 for this simulation. We only consider rings from ring 4 now (since we want to build a model of the galaxy and the physics of the center of the galaxy is not very clear now followed Gerola and Seiden (1978). However, we can also not consider any of this, only think these models are different extended sandpile models with a space in the center of the polar table).

For the sites shift, we set v4=1; v5=1; v6=1; v7=1; v8=1; v9=1; v10=1; v11=1; v12=1; v13=1; v14=1; v15=1; v16=1; v17=1; v18=1; v19=1; v20=1; v21=1. There are some redundancy codes in the code but we do not use them in this model, hence, please ignore them.

Other parts of the algorithm are the same as before. Furthermore, we add the process of making videos in this model, the result videos can be found in the C4 results file. For the most of later models, we provided videos of them and also stored them in the C4 results file.



(a) The galaxy5 model with rotation



(b) The galaxy5 model without rotation

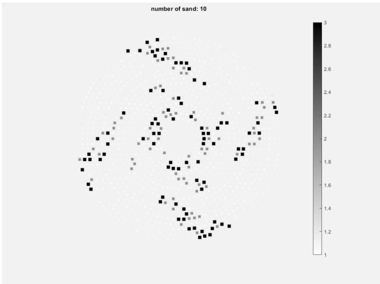
Figure 4.3: The result plots of galaxy5 model with and without rotation. It looks similar to the galaxy4 model
.

It is not very different from the galaxy4 model, except that the distance between the points is smaller. This is due to the dropping regeneration time being set to 0 time steps, although the topples regeneration time is double in this model. We can conclude that the sand that falls each time steps dominates the sandpile in this model.

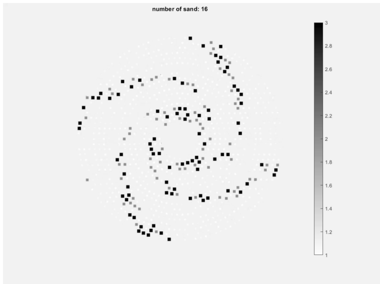
Galaxy6 model:

Based on the galaxy5 model, we construct a new model by introducing the distribution of sand before the simulation and restricting the positions that can be chosen to add sand during the simulation.

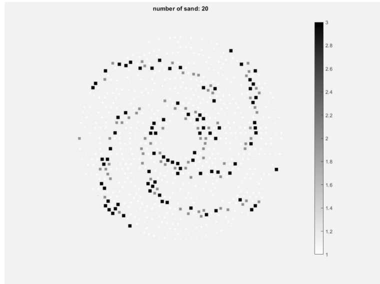
Before the simulation, we distributed 2000 sand in the polar table by the following rule. Firstly, randomly choose a site in the table, then change the theta corresponding to this site from radian to degree. Then, we check whether the radius of this site is inside the range 3 to 6. We check whether the theta is inside the intervals [65,85] degrees, [155,175] degrees, [245,265] degrees, [335,355] degrees. If both of them is true, then the height of this site will increase by 3 unit. For the topples, we will check the criteria of the theta of the neighbour sites again and if true, then it will be induced. This is a model without physical meaning since we control the position that which sand can be dropped, it's not a random self-organizing model but as we can see from the results, the model is very successful in showing four main spirals.



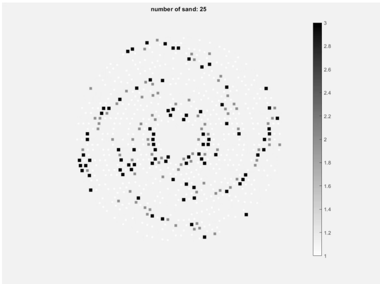
(a) 10 time steps plot



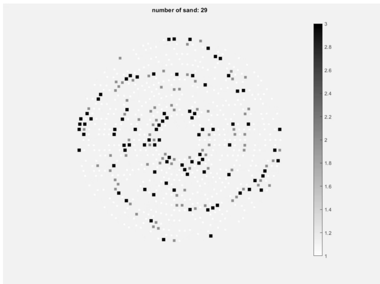
(b) 16 time steps plot



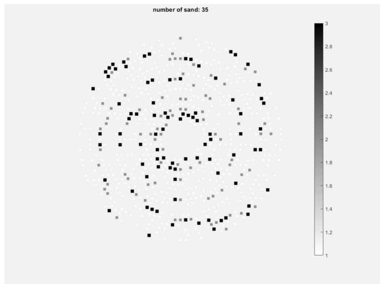
(c) 20 time steps plot



(d) 25 time steps plot



(e) 29 time steps plot

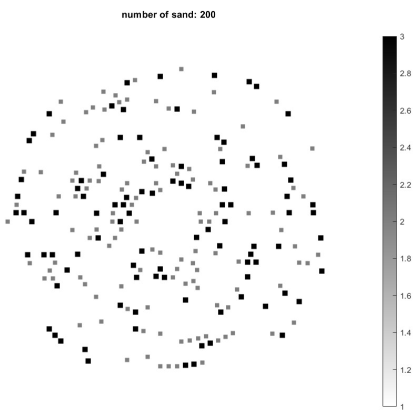


(f) 35 time steps plot

Figure 4.4: The galaxy6 model with rotation in different time steps, we can see it shows 5 clear spirals until 29 time steps. After this time, The spiral arms begin to intertwine and lose their distinctive features. This was the common problem with the galaxy model before the rotation curve was introduced. This problem may arise because we also do not follow the galaxy rotation curve in the rotation speed of this model.

From Figure 4.4, we can see the result has 4 clear spirals and how it changes in different time steps. The spiral structure will be unclear after 35 time steps. However, the structure is controlled by us, not the self-organization, hence, this is not the model that we want in this project.

However, we can still learn from these results, that is, if we change this model a little bit, let the allowed added sand intervals rotate as same as how the sandpile rotates (since the intervals are always in the same positions in this model and sand are only able to add in this region, however previous sand will rotate and hence this will destroy the galaxy spiral arm structure). In addition, remove the distribution of the sand before the simulation. Then, we can create a very stable spiral structure. We can combine the sandpile model with the famous density wave theory for the spiral structure that was first introduced by Lin and Shu (1964) to build a very clear and stable galaxy spiral arm structure. Meanwhile, since we introduce the sandpile model and the idea of self-organization, our model can also able to predict and simulate the flocculent spirals as same as the SSPSF model did. In this case, our model will have physical meaning and can serve as an extension of the SSPSF model. In addition, we need to let the model follow the galaxy rotation curve to give the correct spiral structure of the galaxy.



(a) The galaxy6 model with rotation (200 time steps)

Figure 4.5: plot at the end of 200 time steps of the simulation

Figure 4.5 shows the appearance of the sandpile after 200 time steps, the parameters that this model used are: dropping regeneration time is 0, topping regeneration time is 1. Neighbours are the closest 4 sites of the topples site. The number we add randomly or induced by topples is 3. The critical value is larger than 3. v4=1; v5=1; v6=1; v7=1; v8=1; v9=1; v10=1; v11=1; v12=2; v13=2; v14=2; v15=2; v16=2; v17=2; v18=2; v19=2; v20=2. Furthermore, we make videos for the galaxy6 model. In addition, the number of spirals depends on the number of allowed intervals we choose. In this case, we choose to use 4 intervals and hence there are 4 spirals. If we choose different intervals, such as 2, then, there should be two spirals. As we talk above, this model is just a basic trial model for now, and we can further extend this model to a real model that can simulate and predict the spiral structure of a real galaxy. However, we should continue our discussion more related to the sandpile model in this project now,

We should note here that the galaxy7 model is almost the same as the galaxy6 model, we continuously develop the codes that we used in galaxy6 and we change some values of parameters. The results of the galaxy7 model are similar to Figure 4.4 and Figure 4.5. The galaxy8 and the galaxy9 models are also similar to the galaxy6 model, we only change parameters and improve the codes. Therefore, we will not discuss these models in the report, however, their codes are still useful for further investigation of the galaxy formation model.

4.1.3 Further algorithm 1

After the "galaxy series" of codes, we developed a new series called the "morering series" in our files, models in this series are called morering plus a number such as the morering1 model. As the name said, we extended the polar model from 20 rings to 30 rings, 40 rings, 50 rings, and so on.

We introduce two new parameters, spontaneous probability P_{sp} and induced probability P_{st} that followed the idea from Gerola and Seiden (1978) and Gerola et al. (1980). In addition, as we discovered in galaxy series models, the spontaneous addition of sand in each time step dominant the structure of the sandpile and topples also have a significant effect on the structure of the sandpile. Hence, we want to introduce these two probabilities to control and likelihood of a site being added sand or induced to add sand by topple site.

In this case, for every time step, there is not always added sand in the active position but follows the spontaneous probability. There is a spontaneous probability to add sand and then we check the other conditions we discussed above. In addition, for the induced sites by topples sites, there is an induced probability that each site be induced rather than it will always be induced. This is a more complex sandpile model in the polar table and other parts are still as same as we talked about in the galaxy6 model.

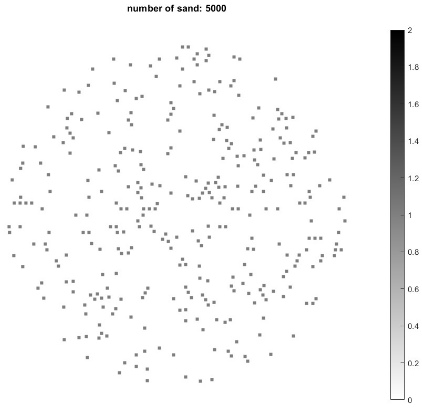
4.1.4 Further results 1

In most of the models of this series, we will only consider models with rotation and ignore the models without rotation unless specified until the end of this chapter. Now, we will present the results by separating them into different parts.

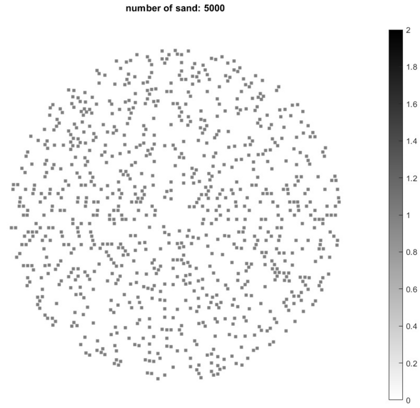
Morering models part 1:

In the morering1 model, the algorithm is the same as the galaxy6 model except we introduce $P_{sp} = 0.1$ and $P_{st} = 0.28$ and we did not distribute any sand in the table before the simulation. However, the restrictions are the same as the galaxy6 model. The time steps is 5000, there are 30 rings. Regeneration time is 200 time steps, critical value is 1. Then, we set v4=1; v5=1; v6=1; v7=1; v8=1; v9=2; v10=2; v11=2; v12=2; v13=2; v14=2; v15=3; v16=3; v17=3; v18=3; v19=3; v20=3; v21=3; v22=3; v23=3; v24=3; v25=3; v26=3; v27=3; v28=3; v29=3; v30=3 for rotation of the sandpile.

In the morering2 model, we extend the number of rings to 40 with v31=4; v32=4; v33=4; v34=4; v35=4; v36=4; v37=4; v38=4; v39=4; v40=4 for rotation. Other parameters are exactly the same as the morering1 model. However, we removed the restriction that we can only add sand in the allowed intervals of the table. The sand can be added randomly in the polar table in this model now.



(a) The morering1 model with restriction of allow adding region



(b) The morering2 model without restriction of allow adding region

Figure 4.6: The result plots of morering1-2 model with rotation

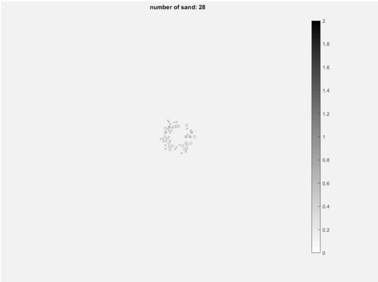
We can see from Figure 4.6 (b) and compare to (a), that if we did not restrict the intervals that can add sand, the sand would be distributed more evenly. The distance between sites gets a lot more crowded. This is because we are not limited to where we can put sand, and the area where we can put sand has become much larger. The reason the sand in Figure 4.6 (a) is not crowded into a few areas where it can be placed is because we introduced rotation, and the sand is therefore distributed throughout the polar table.

In the morering3 and morering4 models, we further extend the number of rings to 50. We change some parameters to see what is different. The results are similar to Figure 4.6. Hence, we will skip them in this report. From changing different values of parameters, we can see many different distributed patterns of the sandpile from sparse to dense. Obviously, these parameters are very important to our model, and we need to carefully study different parameter combinations to build a reasonable model. In addition, the morering5 model will be discussed later and morering6 is an incorrect model so we skip it.

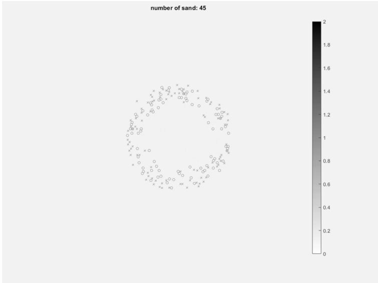
Morering models part 2 (Ripple diffusion models):

Now, let us continue our journey, we will talk about the morering7 model first. We will call this model the "Ripple diffusion model".

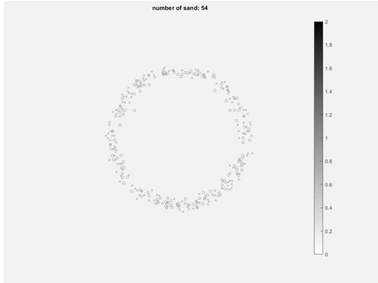
We introduce two new vectors, the lifetime vector, and the history vector. We give each site that is non-zero height a lifetime 100 time steps, and we introduce a new probability $P_{stc} = 0.1$, the probability a topples sites can induce sites in the same ring of it be very low and keep the probability of the topples site induced the sites not in the same ring of it high. The history records the position of the sites from the last iteration, and we will plot both the position of sites in this iteration and the position of the sites in the last iteration to get a clearer view of the track of the sites. History uses the "o" marker and we use "x" for the current position.



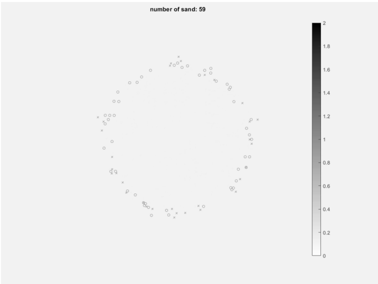
(a) 28 time steps plot



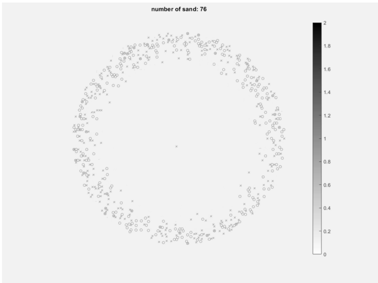
(b) 45 time steps plot



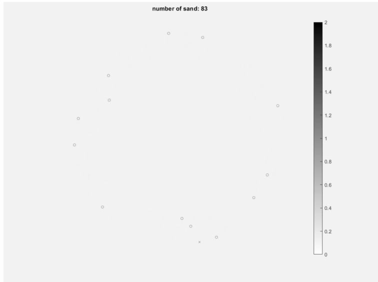
(c) 54 time steps plot



(d) 59 time steps plot



(e) 76 time steps plot

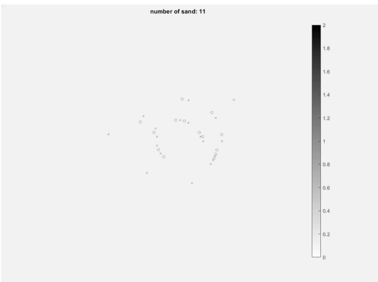


(f) 83 time steps plot

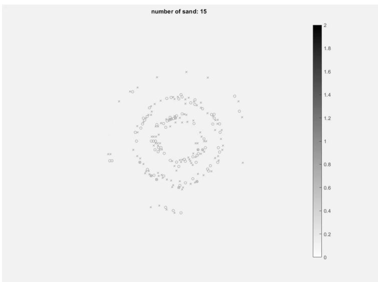
Figure 4.7: The morering7 model with rotation with different time steps

Figure 4.7 and the video show that there is a small ring of sites that appears in the beginning then it diffuses and becomes larger and larger, then, finally, it will disappear again after a few time steps. It's a cycle where the same things come and go again and again. This is very similar to the diffusion of ripples on water's surface. Therefore, we have successfully simulated the ripple diffusion model in the morering7 model. Further research can be carried out in the future, and we believe that we can build a random self-organizing ripple diffusion model based on this model.

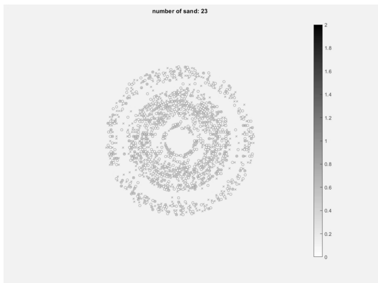
For the morering8 model, we change some parameters of the morering7 model, especially the $P_{stc} = 0.5$ and $P_{st} = 0.5$. Then, we see different behavior of our ripple diffusion model as shown below.



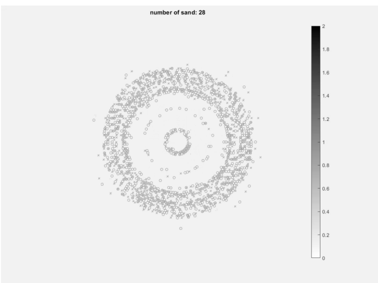
(a) 11 time steps plot



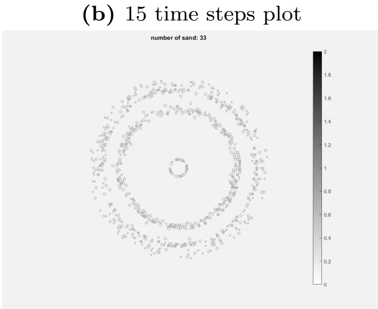
(b) 15 time steps plot



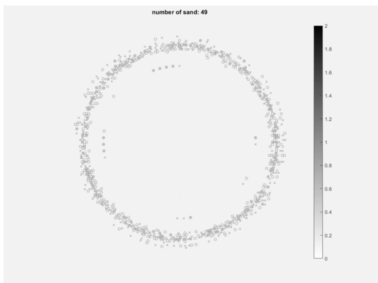
(c) 23 time steps plot



(d) 28 time steps plot



(e) 33 time steps plot



(f) 49 time steps plot

Figure 4.8: The morering8 model with rotation with different time steps

From Figure 4.8 and the corresponding video, we can see the rings become more clear but the diffusion did not look like the diffusion of ripples on the water's surface now. This shows that our model has the potential to simulate many different and quite complex ripple structures, which is a model worthy of further study. We should note that both of these two models have videos available in the C4 results file.

Morering models part 3:

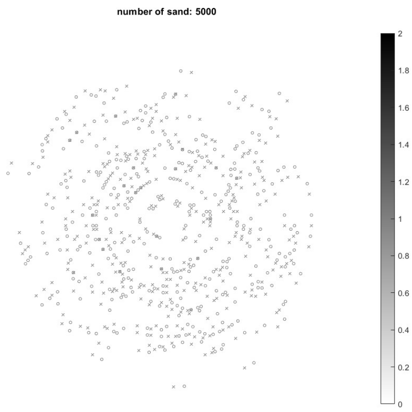
The morering9-11 will be talked about later at the end of this chapter. In this subsection, we will continuously discuss the morering12-18 models. These three models are the same except using different values for some parameters. We further explore the roles of probability that we are involved in these models.

For the morering12 model, we defined $p_{sp} = 0.8; p_{st1} = 0.28; p_{st2} = 0.8; p_{st3} = 0.28$. p_{st1} is the probability that the site is not in the same ring as the topples site and theta of this site is less than the theta of the topples site induced. p_{st2} is the probability that the site is in the same ring of the topples site be induced. p_{st3} is the probability that the site is not in the same ring as the topples site and the theta of this site is larger than the theta of the topples site induced. For other parameters, please refer to the codes.

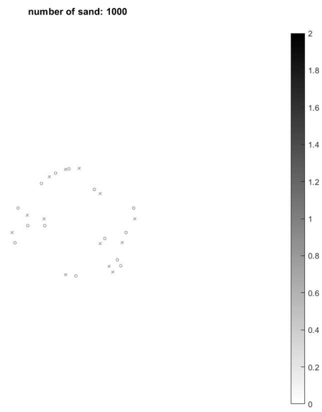
For the morering13 model, we defined $p_{sp} = 0.8; p_{st1} = 1; p_{st2} = 0.28; p_{st3} = 1$. However, the lifetime be seated to 10 in this case.

For the morering14 model, we defined $p_{sp} = 0.8; p_{st} = 0.28; p_{stc} = 0.28$. We define probability as in Morering models part 2. The lifetime is seated to 30 in this case. In this case, the "o" marker is for the active vector instead of the history record here.

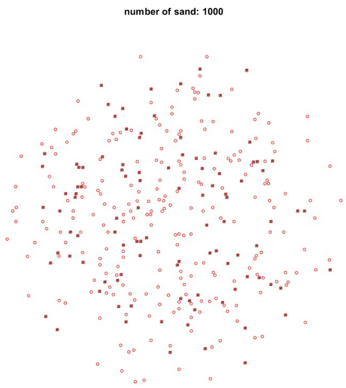
The morering16-18 do the similar things we discussed in the morering12-14. The models are the same but only change some important parameters to see what happens. The morering16 model removed the history record. We will only provide the result plots here but the parameters need to refer to the codes since we want to save some space in this report. We will not include the results of morering17 and morering18 here. graphical results are shown below.



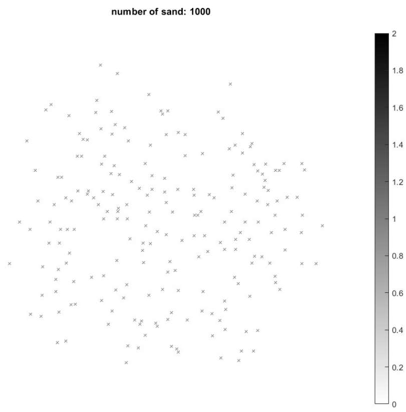
(a) The morering12 model with 5000 time steps



(b) The morering13 model with 1000 time steps



(c) The morering14 model with 1000 time steps

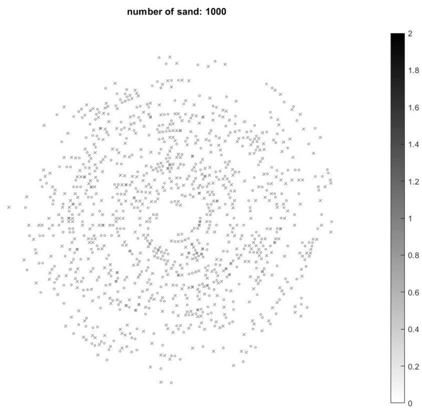


(d) The morering16 model with 1000 time steps

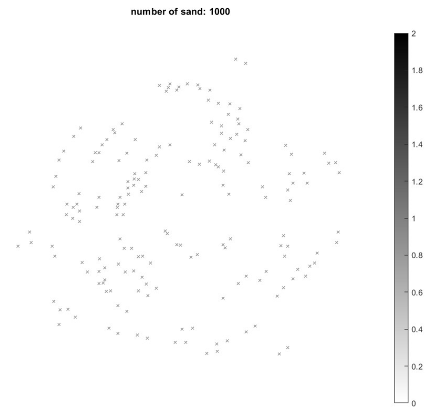
Figure 4.9: The morering12-14, 16-18 model with rotation

From Figure 4.9 (a), we can see the plot of the model starts to appear in some pattern similar to the galaxy, however, the spiral arms structure is still unclear. From Figure 4.9 (b) and the corresponding video, we can see the sandpile in this model cannot form a stable structure. The structure will appear and then disappear and its structure is confined to a very small area, and it cannot be spread over the entire table through the collapse (induce) of sites. From Figure 4.9 (c), we can see how the sites induce their neighbour sites in a graphical way. We will further discuss this method and the idea behind it in the next section (section 4.2) when we use another algorithm of the sandpile model.

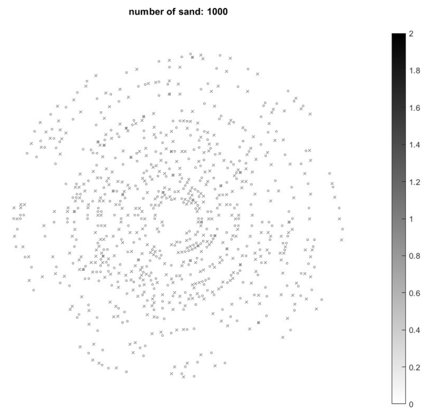
After some further developments, we will discuss the morering5 model, the morering9-11 model, and the morering15 model here since we have not talked about them yet. Since the results of the morering5 and morering10 models do not meet our expectations, we skip them at this stage in our report. We changed some parameters here, please refer to the values that in codes of the same name of this model to check the values.



(a) The morering9 model with 1000 time steps



(b) The morering11 model with 1000 time steps



(c) The morering15 model with 1000 time steps

Figure 4.10: The morering9,11,15 models with rotation with 1000 time steps

From Figure 4.10 (b), we can see the sites are too small that the parameters we have chosen in this model are not useful. From Figures 4.10 (a) and (c), we can see the morering9 model and morering15 model give a better distribution of stars in the galaxy than before. The result of the morering15 model is the best result in the moreover series. However, there still exists the problem that spiral structure is unclear in these models. The code of our model needs to be further developed to create a stable and clear spiral arm structure.

4.1.5 Further algorithm 2

Now, let us talk about a new series of our model called the "new series" and each model is called new plus a number such as new1, new2. In this series, we will only briefly discuss some results and will not talk too much about the new algorithm we used in this series since this series is a transition series and their results are not the main result of our project, we will apply the algorithm here to our further (most current works) models. When we talk about them, we will talk about the algorithm deeply. For now, we just provided some plots and comments.

We further extended our algorithm, firstly, for new1, we added 3 sand in different regions of the table every time step if the position is active and followed the P_{st} rather than we only added 1 sand before. The adding regions are different most of the time controlled by a new parameter but we did not change it a lot most of the time.

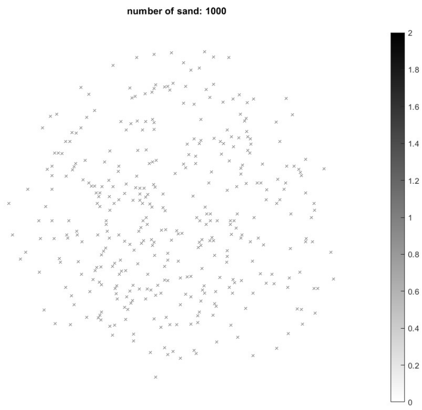
The new1 model is similar to the models we discussed in the last subsection, we used P_{st} and P_{stc} to control whether the sites are induced by the topples site.

The new2 model tries some new codes and it becomes a ring that rotates around the center constantly (The ring will stay in the outer few positions most of the time and become wider or thinner). The new3 model continues this work but creates an unstable ring, the ring will change very often. Since they are not interesting, we will skip them.

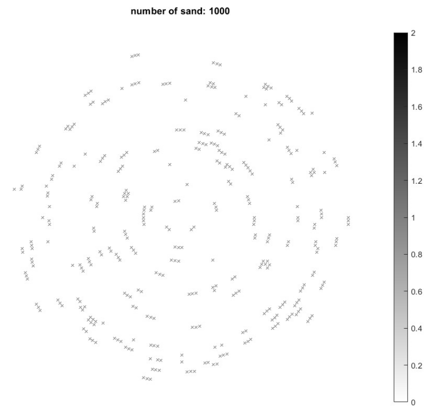
For the new4, new5, new6, and new7 models, we use a different method to check the position of the neighbours of the topples sites. In this stage, this method is relatively complicated, and hence as we said before, we will not discuss them here. The result of new7 is similar to new6, hence, we will only present new4 results. Note that, videos are available in C4 results files.

4.1.6 Further results 2

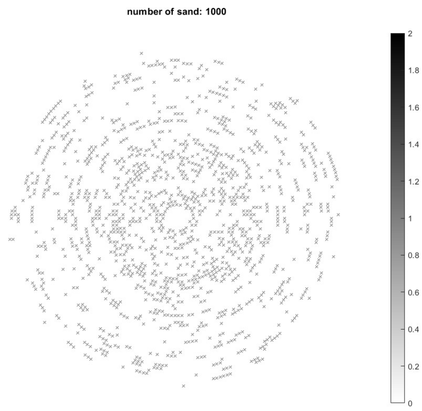
In this subsection, we will provide some results of the new series as shown below.



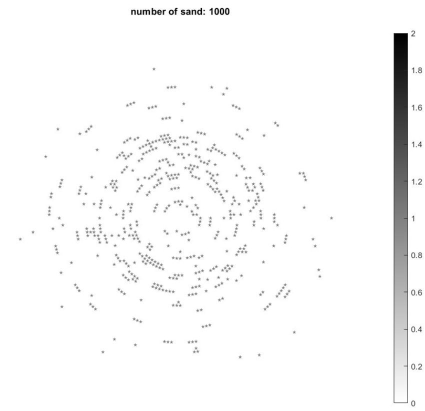
(a) The new1 model with 1000 time steps



(b) The new4 model with 1000 time steps



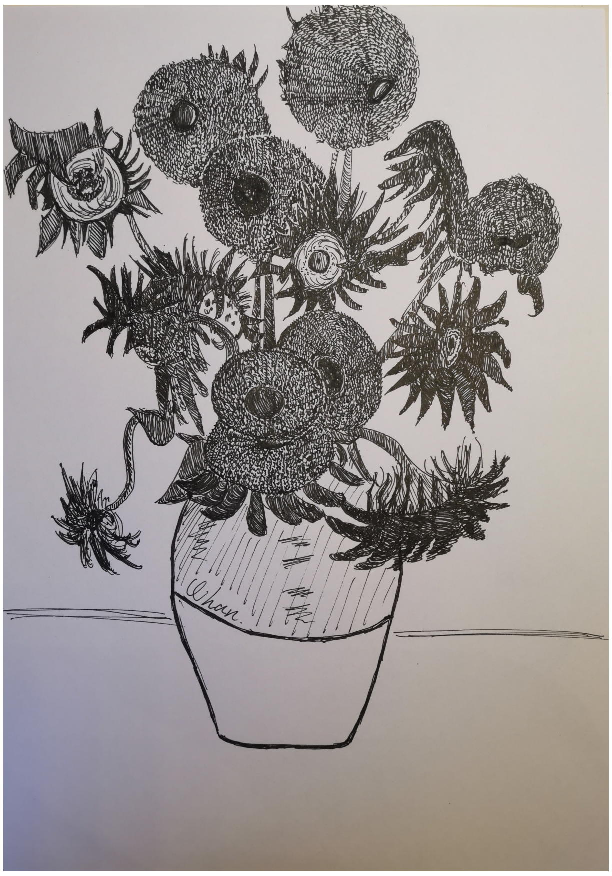
(c) The new5 model with 1000 time steps



(d) The new6 model with 1000 time steps

Figure 4.11: The new1, 4-6 model with rotation

From Figure 4.11 (a) and (b), we can see the results are too sparse, and from Figure 4.11 (c) and (d), the result plots look like the one of the flower of The "Sunflower" drawn by the famous painter Van Gogh. I will provide my sketch of the "Sunflower" that imitates Van Gogh's "Sunflower" here. My sunflowers are only white and black and are drawn with a pen instead of an oil brush, hence it will look more similar to Figure 4.11 (c). We will use this figure to end section 4.1.



(a) Qihan Zou's (The author of this report) Sunflower (2019)

Figure 4.12: Qihan's sketch of the Sunflower that imitates Van Gogh's Sunflower, we can see the result of the new5 model (Figure 4.11 (c) and Figure 4.11 (d)) is similar to the flower in the sketch. The colors in the Figure are much lighter than those in the painting, but the structure is similar.

Although our models in the new series do not perform well and it is far away from the aim of this section (build a star formation model), however, it is still very interesting since we can use these codes to draw different flower pictures.

As the name of this subject says, the Art of Scientific Computation, I think drawing by using computer code is also art. Today, the development of AI makes it possible to make beautiful, vivid, and complex drawings with computers, which I think is the trend of the future of art, and we should embrace it.

4.2 Sandpile model (and the rotational sandpile model) in the polar system by the second algorithm

Now, the series "moreover series" will be discussed here. The moreover1 and moreover2 models are basic models, we apply the new algorithm to construct the whole model here. The moreover3 model is incorrect in some places hence we will ignore it. The moreover4 model is the fundamental model that we will use. We built moreover41, moreover42 and moreover43 based on moreover4 model. The moreover5-10 models are some trial models and did not perform well and hence we ignored them.

As we discussed above, the algorithm used to generate the polar table is inconvenient. In addition, we consider all the rings together as a whole as a very big matrix may cause some problems when we consider rotation. Moreover, the algorithm we used before wasted lots of computation resources. We said we would present a more convenient method in previous sections, now is the time to discuss this method.

In this algorithm, we use some trick to consider each ring as a matrix rather than put all rings together. Then, we do the simulation for each ring one by one. That is,

$$ring_i = \begin{bmatrix} theta_i \\ radius_i \\ active_i \\ life_i \\ height_i \end{bmatrix}$$

where in some model, we with set all $active_i$ or $life_i$ be 0 and ignored them. Life vector is unused in most cases if we do not say we use it. We perform our simulation for each ring one by one follow the algorithm and then combine them together when we plot them.

For each time step, we check which site of a ring is inactive, and we decrease the recovery time by 1 unit. Then, we check which site of a ring is active, and if a site is active, it will have a spontaneous probability P_{sp} to spontaneously add 1 sand. Then, we plot a figure of the sandpile now and keep it (*). When plotting, we put all the matrix of rings together and plot the big matrix. Then, we check which site has 1 sand, if a site has 1 sand, then it will topple. That is the critical value of this model 0 if the height of the sandpile is not 0, then it will topple.

Then, we use a new method to check the neighbours of the topple site. We know the neighbours of the topple site must exist in the same ring of the topple site or in the n+1 th ring or n-1 th ring if these rings exist. Therefore, we can only compare which sites of the n th ring, n+1 th ring, and n-1 th ring are the closest 6 sites of the topple site in the n th ring since we define matrices for each ring. Then, for each neighbour site have an induced probability P_{st} to be induced. The method to check which sites in the n,n-1, and n+1 rings have the smallest distance to the topple site is using the same formula we use in section 4.1. $d = \sqrt{r_i^2 + r_{top}^2 - 2r_i^2 r_{top}^2 \cos(\theta_i - \theta_{top})}$.

Then we plot the figure of the sandpile in the polar table again. (In moreover3 and later models, we keep the position of the sandpile (*) and then plot both it and the sandpile after induced happens here).

For each site induced by topples and spontaneously induced site, we will record it be inactive and have a regeneration time to be active again.

At the end of each iteration, we rotate the sandpile following specified parameters (only for moreover4 and later models. We did not include rotation before the moreover4 model).

For moreover1 model, number of sand is 100, $p_{sp} = 0.002$, $p_{st} = 0.28$, regeneration time is 11 time steps.

For moreover4 model, number of sand is 500, $p_{sp} = 0.002$, $p_{st} = 0.28$, regeneration time is 11 time steps. V=[1 1 2 2 3], that is for ring 1-10, v=1; for ring 11-20, v=1, for ring 21-30,v=2; for ring 31-40, v=2; for ring 41-50, v=3. We will use this notation for moreover41-43 models. The definition of v is the same as we defined in previous sections.

For moreover41 model and moreover42 model, number of sand is 200, $p_{sp} = 0.0002$, $p_{st} = 0.2$, $p_{sth} = 0.3$, regeneration time is 11 time steps. V=[1 2 3 4 5]. p_{sth} is the probability for the neighbour sites in the same ring of the topples site. We will change the value of it to see what is different.

For moreover43 model, number of sand is 150, $p_{sp} = 0.0002$, $p_{st} = 0.28$, $p_{sth} = 0.28$, regeneration time is 11 time steps. V=[1 3 5 7 9].

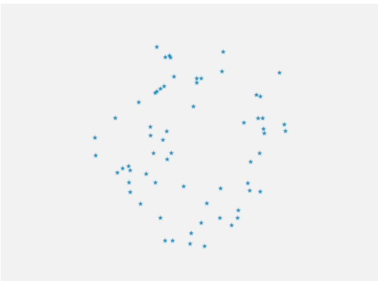
The above algorithm follows the idea of the works of Gerola et al. (1980), Gerola and Seiden (1978) and Seiden and Schulman (1990). The idea of our algorithm is mainly to follow the algorithm of the stochastic self-propagating star formation (SSPSF) model. However, we still change lots of things to fit in our sandpile model content since our project is focused on the sandpile model and the extended sandpile model and we also want to build a new star formation model based on the sandpile model.

The following description complements our algorithm above.

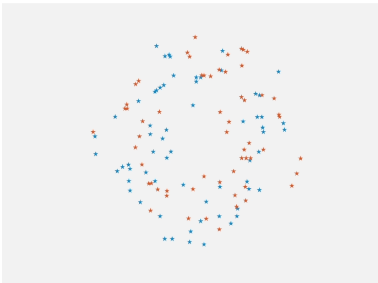
1. Firstly, build 50 rings, the first one has 6 sites with a radius of 1, and the second one has 12 sites with a radius of 2. The third one has 18 sites with a radius of 3,..., and the 50th one has 300 sites with a radius of 50. For each ring, we use a matrix to represent the theta, and radius for them. For each site, we use a vector to denote is the site active or not. Initially, all sites are inactive.
2. For every iteration, for each ring: firstly, check if there are any sites that are active, if active, we minus 1 of their lifetime (which is equal to the regeneration time).
3. Then, we check which sites are inactive, the value of the active vector is less or equal to 0. For every inactive site, there is a probability that Psp to become active. Store the current value here.
4. After that, we check which sites are active for every ring. For every active site, they will topple. We set the critical value be 0 here, if the height if more than 0 (is 1) then the site will topple.
5. for every toppling site, change it from active to inactive. Then, the sits between the toppling site that belongs to the same ring n have a probability Pst to be induced to active if they are inactive before. For rings n+1 and n-1, the closest 2 sites to the toppling site have a probability Pst to be induced to active if they are inactive before. Hence, we need to check which sites are closest. If a site is active, give it a regeneration time. The site needs regeneration time to become inactive again.
6. Then We plot a figure for all rings on the same axis. Both the value we recorded before and the value for now. We combine all rings to be a larger matrix to plot using "polarscatter".
7. Rotate the sites. We did not follow the rotation curve for the galaxy in this report.
8. number sand = 150; Psp=0.0002; Pst=0.28; regeneration time = 11; follows the common values of papers in most situations (and models) in our report. We also try some other values to check the differences.

By using this new algorithm, we reduce the difficulty of computation that we suffered before in section 4.1. Because we only need to consider the front and back two rings of the current topple ring, the amount of computation for the neighbor drops. Using loops to generate tables eliminates the need to manually write the parameters of different loops one by one, this greatly reduces the amount of code and effort by the code writer.

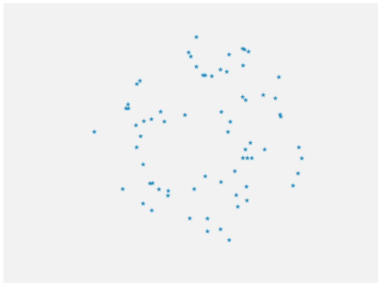
With this approach, after our trials, our same computer was not only able to simulate 50 rings very quickly but also could simulate 100 rings even more rings very quickly.



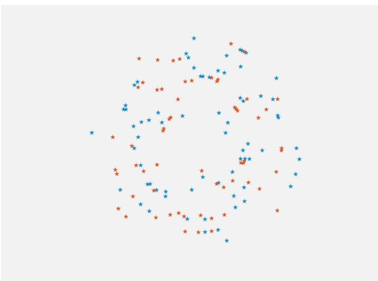
(a) The moreover1 model original sites



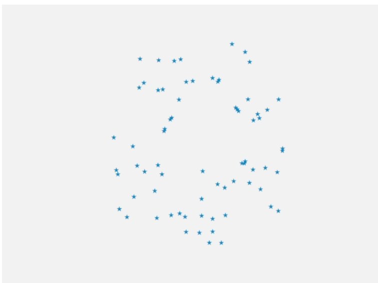
(b) The moreover1 model original sites and induced sites



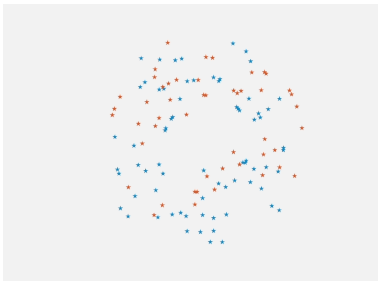
(c) The moreover1 model original sites



(d) The moreover1 model original sites and induced sites



(e) The moreover1 model original sites



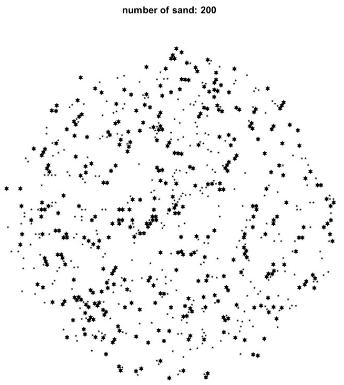
(f) The moreover1 model original sites and induced sites

Figure 4.13: The results of the moreover1 model give us an insight into how we plot two figures (one is the previous sites and one is the induced sites) together. For example, (a) is the sandpile now (blue), then it will induce some sites to be the sandpile, red sits in (b). Then the blue sites will disappear and the induced red sites become blue to be the sandpile now in (c). This process will repeat in our simulation. We will use different shapes of markers to represent the original sandpile sites and the induced sandpile sites later rather than use different colours.

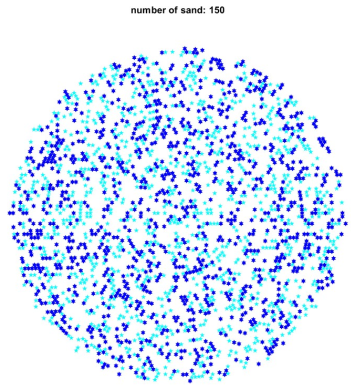
Now, let us look at some results from our simulation for moreover series models. They are the moreover4 model and further models built based on it, the moreover41 model, the moreover42 model, and the moreover43 model. The results are shown below.

(a) moreover4 model

(b) moreover41 model



(c) moreover42 model



(d) moreover43 model

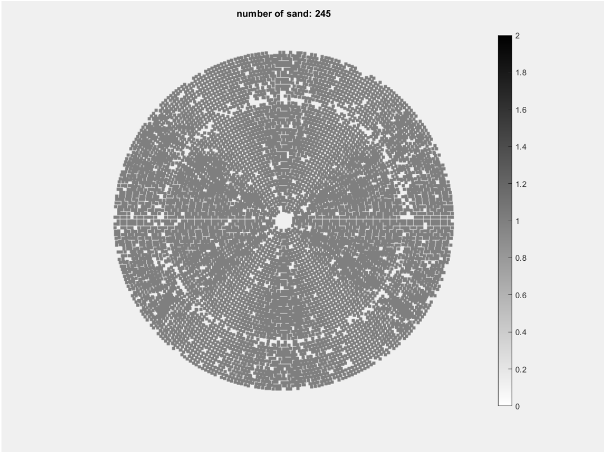
Figure 4.14: The moreover4, 41-43 model with rotation in different time steps.

From Figure 4.14 (a), we can see that there are just a few sites scattered across the disk, but there are some preliminary patterns of galaxy star distribution. From Figure 4.14 (b), we change the parameters of probability and hence result in a very dense distribution of sites in the polar disk. From Figure 4.14 (c), this result seems to be the best one given by the model so far, it further gives a distribution more similar to that of the matter distribution in a real galaxy. It is similar to the pictures from different emission lines that are captured by the telescopes. From Figure 4.14 (d), we use different colours to represent the results, it is similar to Figure 4.14 (b) which is too dense and we did not think it was a good result.

However, as same as the results we discussed in the last section, all results in this section do not contain a clear and stable spiral structure. Most of the time, the sites just look like randomly fill in the disk. There exists another problem in this model, as Aschwanden et al. (2016) discussed, the SSPSF and similar models for galaxy formation need many parameters to control the patterns of the results. The model we built basically follows the algorithm of the SSPSF model and hence we introduce lots of parameters such as active or inactive, lifetime, probabilities, and so on to build a very complex model. However, such an algorithm deviates from the original intention of introducing the sandpile model. We introduced the sandpile model and the idea behind it in order to create a more concise model that does not require human manipulation and can generate the patterns we need entirely through random self-organizing behavior as the basic sandpile model. Obviously, we have not yet accomplished this goal.

Although we have not yet come up with a more simple and effective method, we can then try to incorporate the new ideas we discussed in the galaxy6 model into the current best model (moreover42 model). Merge the idea of the density wave model and the sandpile model we discussed a lot in this model in the current simple code form, there is a great possibility that we can create a model that is better than all the models we have built so far.

With some regret, because of the time constraints of this semester, we have not yet managed to come up with a workable new theory of star formation model in this project and we have not performed a mathematical analysis of the models we already have. Let us end this chapter with another funny stuff that was created in this project.



(a) Funny stuff: The "Death Star" created by our model

Figure 4.15: The pattern (especially the complex and regular pattern above) of one of our models that looks like the front view of the "Death Star". The Death Star is a space station and super-weapon in the Star Wars series movies. Many of the complex, precise lines in our Figure are like fine mechanical structures.

Chapter 5

Conclusion

In conclusion, we first built the basic sandpile model at the beginning of this project by using open and closed tables (adding sand randomly or adding sand at the center). We discussed these basic sandpile models in Chapter 2 of our report. We then analysis some properties of these models, mainly the mean height of the sandpile, avalanche size, avalanche loss, and avalanche area. The main result We found is that in the basic sandpile model, the avalanche size and its frequency follow a power law distribution. The avalanche area and its frequency also follow a power law distribution relation. We also provided the best linear fitting of the avalanche size, the avalanche area, and the avalanche loss by using the linear model and we recorded the corresponding coefficients in the coefficients tables. We also developed a method to create videos for our simulation and this method can be used in many other simulations. The results of this chapter are stored in the C2 results file and the codes are stored in the C2 codes file.

Furthermore, we built several different extended 2D sandpile models in the squared table, and we discussed them in Chapter 3 of our report. The first important model is the lighthouse sandpile model and the second important model is the sandpile model with adding sand at the edges or outer four corners. We also changed some rules of these models to check the differences and provided a few more models in our report. One of the most important and innovative points is we introduce rotation to our sandpile models and we create a new algorithm of matrix rotation in Matlab. The interesting result we should mention again is when we add rotation in the lighthouse sandpile model, we can see patterns similar to a rotating galaxy. This is the cornerstone and starting point of chapter 4. In addition to the construction of the models, we also examined important properties of the models that we built. We mainly investigated the avalanche size and its frequency and the mean height of the table since they are most important for the sandpile model, we found that in most of these extended models, there exists a power law relationship between the avalanche size and its frequency and for most of extended models, the mean height will become constant after different time steps that mean they are going into the critical state. In Chapter 3, we also did fitting for all of our plots (except for the mean height plots) by using the linear model fitting and we provided the corresponding estimated coefficients that were included in the coefficient tables. In addition, we provided videos for all of the models in this chapter. The results of this chapter are stored in the C3 results file and the codes are stored in the C3 codes file.

In Chapter 4, we first extended our sandpile model to a polar system table and we named it the basic polar sandpile model. Then, we built many further models based on the basic polar sandpile model, the first series is the galaxy series, the second series is the morerings series, the third series is the new series and the fourth series is the moreover series. The purpose of all of the works in this part is to construct a new stochastic self-organization star formation model based on previous works in this area and combine the idea of the sandpile model in our project. The "galaxy6 model" and the "moreover42 model" are particularly noteworthy, as we discussed in the descriptions of these two models, if we combine these two models together, we should be able to construct a more reasonable and workable sandpile star formation model. Moreover, the "morering7 (the ripple diffusion) model" is also noteworthy, each site can be thought of as some molecule or atom, this model shows the potential to simulate complex ripple diffusion phenomena or complex molecular motion through stochastic self-organizing behavior. Therefore, although this model is a by-product of our main research, it is well worth exploring in more depth in the following research. In this part, we have only carried out a phenomenon approach study on the model we have built and have not carried out any mathematical analysis. Furthermore, in the codes of this chapter, we focus on how to use more elegant algorithmic logic to speed up the computations so that our computer can perform more computationally intensive simulations under the same conditions. Overall, although we have not yet fully succeeded in building a model of galaxy formation based on the sandpile model, we have successfully established a framework for the target model, we believe that in the following research, we can successfully establish our expected model. In this chapter, we also included all our results and videos of our simulations in the corresponding C4 results file. All codes are also available in the C4 codes file.

In addition to the content discussed in the main chapters, we also learned a lot of useful knowledge. Firstly, I learned how to use Latex to write reports. Before this project, I thought Latex was very hard to use since it was like a computer language, then, I successfully learned how to use it to finish my report's writing, including how to insert figures, how to add chapters, and appendix into the Latex; how to use packages of Latex; how to write mathematical formula by using Latex; How to use it for typesetting and so on. Now, I think it's a great writing tool, as easy to use as Microsoft Word. It is also very convenient, its typesetting and its way to input mathematical formulas are very convenient for science and engineering students and researchers like me.

Secondly, I learned a lot of useful programming knowledge and skills in this project, and the most important thing is the ideas for designing various algorithms. I also learned a lot of things that made my code run faster. Although in this report and the code mentioned in the report, I mainly use Matlab for modeling and calculation, however, I also learned Python during this subject and I already use the knowledge I learned in this subject in my research project for my master thesis now. Before this course, I had almost no contact with any programming knowledge. I only learned how to use Matlab for scientific calculation in one subject in my undergraduate. Besides that, I learned how to use R to fit various models in statistical subjects such as fitting linear models, generalized linear models, etc (In this report, for the basic sandpile model and part 1 of the extended sandpile model, we used a linear model to fit the curve, some of them followed the linear relationship and other not. Hence, we can try to use different models to fit these curves in further studies. However, we should mention that if we use other models to fit the curve, it will not follow the power law relation as we discussed). However, I have never really been in touch with programming. Now, through this course, I have come into contact with some real basic programming content, and I have learned a lot, and I no longer have too much resistance to programming (I used to be afraid of programming because I found it very difficult to learn new "languages" before).

Thirdly, I feel the "art" of scientific computing in this project. The area I chose to work in is the complex system, the results plots of various complex models (basic and extended models that I created) are really very beautiful and it's amazing how self-organizing it is. At the same time, I also drew a lot of interesting things as by-products of this project, such as the flower core (Figure 4.11) that looks like a sunflower of Van Gogh and the figure (Figure 4.15) that looks like the Death Star in the Star War.

I really like and enjoy this subject and want to put most of the work of this semester into my report, hence the length of my report is relatively long. I am very grateful to the readers who read my report.

At the end of the conclusion and at the end of this project (and also at the end of this semester), I am very grateful to Professor Roger Rassool, my tutor Balu Sreedhar, and my classmates. They helped me a lot with my projects and my studies and this subject is very interesting and helpful. Thank you so much.

Appendix A

Program flowchart

We will provide a sample flowchart of the basic sandpile model that adds sand randomly at each time step. For other basic sandpile models or the extended sandpile models, the fundamental ideas are the same. All the details of the algorithms are discussed in the report and the codes are available in the code files.

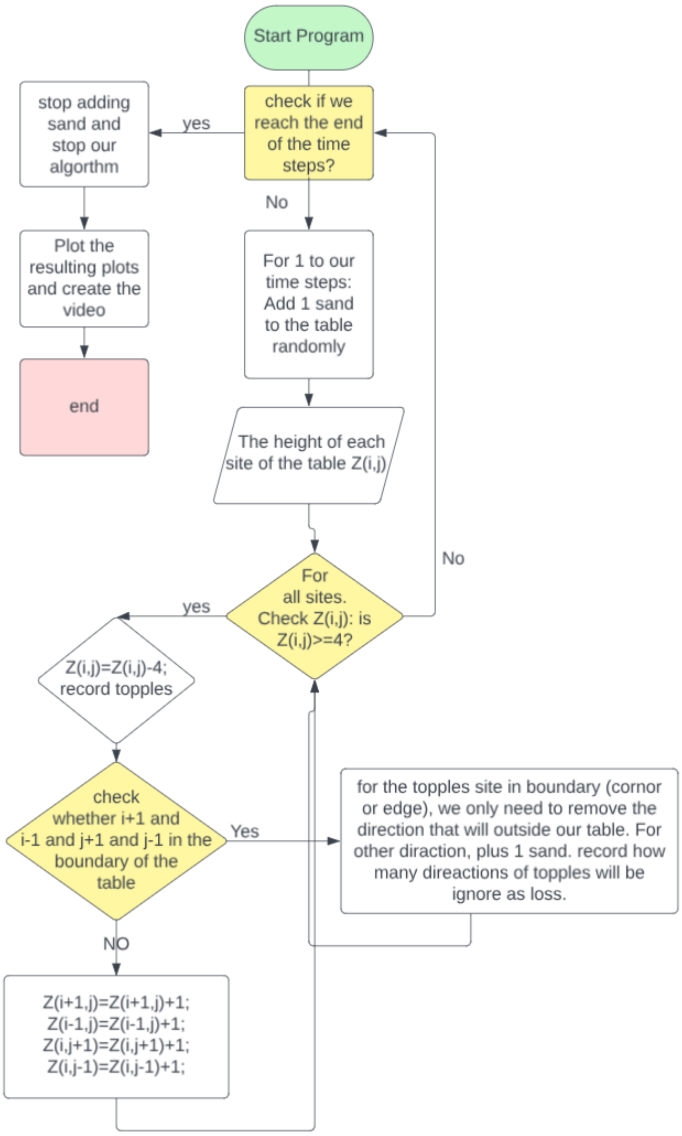


Figure A.1: A flowchart for the basic sandpile model (randomly adding sand at each time step).

Bibliography

Aschwanden, M. (2011). Self-organized criticality in astrophysics: The statistics of nonlinear processes in the universe. Springer Science & Business Media.

Aschwanden, M. J. (2014). A macroscopic description of a generalized self-organized criticality system: Astrophysical applications. The Astrophysical Journal 782(1), 54.

Aschwanden, M. J., N. B. Crosby, M. Dimitropoulou, M. K. Georgoulis, S. Hergarten, J. McAteer, A. V. Milovanov, S. Mineshige, L. Morales, N. Nishizuka, et al. (2016). 25 years of self-organized criticality: solar and astrophysics. Space Science Reviews 198, 47–166.

Bak, P., C. Tang, and K. Wiesenfeld (1987). Self-organized criticality: an explanation of 1/f noise, 1987. Phys. Rev. Lett 59, 381.

Dhar, D. (1990). Self-organized critical state of sandpile automaton models. Physical Review Letters 64(14), 1613.

Gerola, H. and P. E. Seiden (1978). Stochastic star formation and spiral structure of galaxies. Astrophysical Journal, Part 1, vol. 223, July 1, 1978, p. 129-135, 137, 139. 223, 129–135.

Gerola, H., P. E. Seiden, and L. S. Schulman (1980). Theory of dwarf galaxies. Astrophysical Journal, Part 1, vol. 242, Dec. 1, 1980, p. 517-527. 242, 517–527.

Lin, C. C. and F. H. Shu (1964, August). On the Spiral Structure of Disk Galaxies. 140, 646.

Mueller, M. W. and W. D. Arnett (1976). Propagating star formation and irregular structure in spiral galaxies. Astrophysical Journal, vol. 210, Dec. 15, 1976, pt. 1, p. 670-678. 210, 670–678.

Seiden, P. E. and L. S. Schulman (1990). Percolation model of galactic structure. Advances in Physics 39(1), 1–54.