Complexity and Networks Project

Rohan Kamath 01209729
Word Count 2514

February 18, 2019

Abstract

A simulation of a rice pile was completed using python by modeling it as a 1D lattice. Sizes of $L = 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024$ were simulated in order to study self organized criticality. The height of the pile and its standard deviation were studied, and scaling relations were investigated, along with corresponding correction factors for small system sizes. Avalanche sizes were also analyzed both using direct data collapse methods and indirect moment analysis methods, in order to estimate values of critical exponents, and $D = 2.168 \pm 0.006$ and $\tau_s = 1.550 \pm 0.009$ were obtained..

Introduction

This project aims to study self-organized criticality, in order to gain insight into the scale invariant behaviour of complex systems, by simulating the Oslo Model. In particular, the effect of small perturbations on a pile of rice was studied, with a new grain of rice added to the system and the system was allowed to stabilize.This was done for multiple system sizes with a large number of grains. Such a model has many real world applications.

For example, the Gutenberg - Richter Law provides an equation for earthquake size probabilities in a form similar to the result generated by the Oslo model. The model also seems to have applications in hitherto unrelated fields, as demonstrated by *Biondo, Pluchino et.al.* where a similar model of self organized criticality was used to study fluctuations in stock prices, and provided good agreement with historical data in the case of the stock price of General Electric [2].

In this report, the Oslo Model was studied, and the results of height and avalanche size probabilities fluctuation have been elucidated upon. This report follows a similar structure to that of the project specifications, with the implementation and result of each task discussed one after the other.

1 Implementation of the Oslo Model

A rice pile can be implemented as a 1D lattice, characterized by its length L, and local slopes z_i, which is the difference between the heights of the lattice on the locations of i and $i+1$. Each location also has a threshold slope, after which a grain would topple. The threshold slopes were assumed to be a random value of 1 or 2.

This was simulated using python. A class named `aval` was instantiated, with attributes of slopes, threshold slopes, height. An Boolean attribute called `crossover` was also defined to indicate whether the first grain of sand had fallen out of the system, along with an empty array called s, to append avalanche sizes into.

Methods for drive and relax were also defined, and they followed the same algorithm specified by the Oslo Model in [2]. This was the basic skeleton code which was modified in order to implement the subsequent tasks.

1.1 Testing

The following tests were devised in order to check the correct implementation of the model.

- A method called `checkrelax` was defined, which would iterate through the array again to ensure that the threshold slopes were higher than the actual slopes. This helped debug any issues with the program flow.

- A function called `ricepile` was also defined to use matplotlib to plot step-by-step bar graphs of the actual rice pile, in order to aid visualizations.
- The threshold slopes were initially changed to 1 instead of random values, to see if it reached the characteristic height and the staircase formation.
- The threshold slopes were randomized, and print statements were added in the program to print the threshold slopes and the slopes. Then, using a pen and paper, the avalanche was manually simulated, and compared to the `matplotlib` output generated by the rice pile function.
- Finally, the system size was scaled up to 16 and 32, and as a sanity check, the stable height was found and checked against the reference values given in the specification sheet, and they seem to have good correlation.

2 Height of Pile

The height of the pile was simulated and plotted as a function of time (or grains) for system sizes of $L = 4, 8, 16, 32, 64, 128, 256, 512, 1024$ and the following results were obtained:

2.1 Task 2A: Height as a function of Time

On plotting the height as a function of time, it was seen that the height rapidly increased with time, and then plateaued to a constant value with some deviations from the mean. Upon changing the axes to a log-log plot, as shown in figure 1, it was confirmed that the initial increase was exponential, as a straight line was obtained.

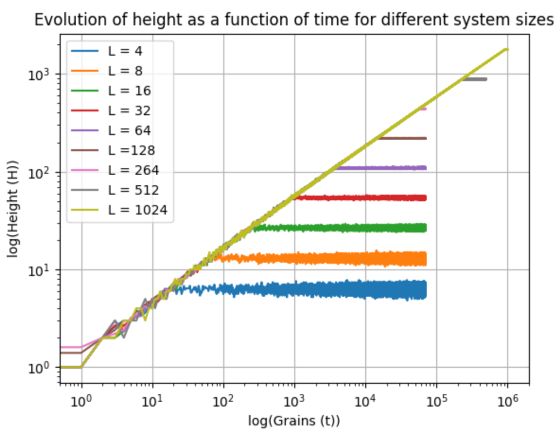

Figure 1: Log-Log plot of height vs time: The height was characterized by an exponential increase followed a stable height value once the system was saturated.

2.2 Task 2B: Theoretical Scaling Relations

In order to explain the scaling relation, some theoretical background was required to see how the height and crossover time scales with system size. For size L » 1, we can assume that that the steady state of the system is a right angled triangle of length L and a height h, and the hypotenuse having a slope of $< z >$. The crossover time t_c can also be defined as the number of grains in the system before the first grain leaves the system. Using the above approximations and assumptions, we can begin to try and understand how the system properties scale with size.

Height

Using trigonometry, we can define the slope of the hypotenuse $\tan\theta$ as $\frac{h}{L}$. Hence,

$$h = < z > \cdot L \tag{1}$$

This shows that for large system sizes, the height of the system scales linearly with system size.

Crossover Time (t_c)

The number of grains in the system before the first grain leaves is given by the area of the triangle created by the rice pile, given by:

$$t_c = \frac{1}{2} \cdot h \cdot L \tag{2}$$

Using the value of h obtained from Eq.1, we obtain

$$t_c = \frac{1}{2} \cdot < z > \cdot L^2 \tag{3}$$

Hence showing that the crossover time scales as L^2. The following derivation assumes that the height is a continuous variable and the result is obtained by integration. However, if height was assumed to be discrete and a summation was performed, extra correction terms are observed, which disappear for large L. The existence and implications of such corrections are discussed in detail in section 2.4.

2.3 Task 2C: Data Collapse of Height

The results of Task 2B demonstrate that the height scales linearly with L, and the crossover time scales with L^2. In order to use this model for arbitrarily large system sizes, the height evolution must be expressed as as a function of a variable which is independent of system size, and only depends on the inherent properties of the system. With this in mind, the Y axis was re-scaled as $\frac{h}{L}$ and the X axis was re-scaled as $\frac{t}{L^2}$. The result was plotted in Fig. 2

Barring a few fluctuations at small values of L, it was seen that all the system sizes had similar height evolution structures, with a characteristic exponential increase then a constant stable height. The scaling function can be written of the form:

$$\mathcal{F} : h(t; L) = L \cdot \mathcal{F}(\frac{t}{L^2}) \tag{4}$$

From Eq.3 it can be inferred that argument $\mathcal{F}$ will have the dimensions of $\frac{<z>}{2}$. Since the time axis is also analogous for the area of the triangle of the pile, the height can be written as:

$$h = \sqrt{2 < z > t}; \quad t < t_c \tag{5}$$

The stable height of the system also has to scale linearly with the system size. Now, $\mathcal{F}$ can be defined as

$$\mathcal{F}(x) = \begin{cases} \sqrt{2 < z > x}; & x \leq \frac{<z>}{2} \\ < z >; & \frac{<z>}{2} \leq x \end{cases}$$

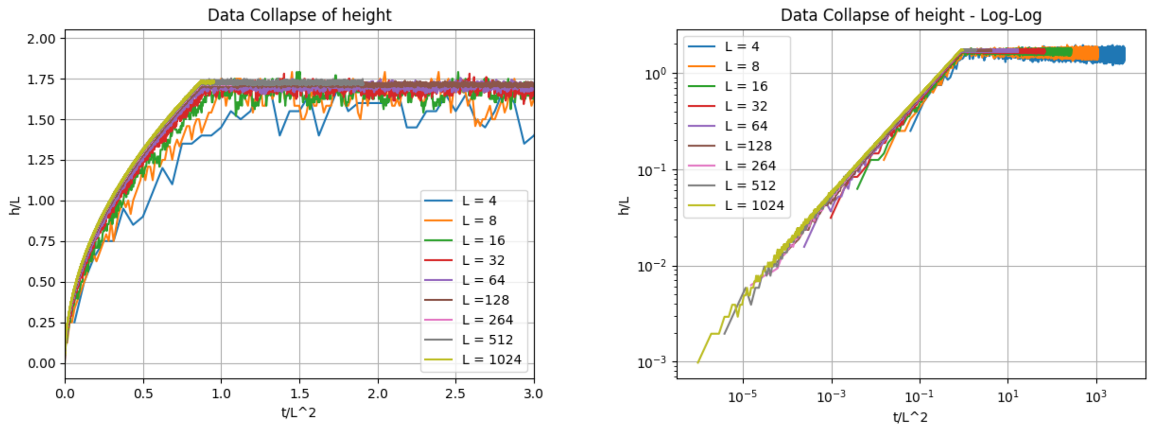

Figure 2: Data Collapse of Height: Upon Re-scaling of the axes, it was found that the height collapsed on a single curve for all system sizes, with an exponential increase followed by a saturation.

2.4 Task 2D:Corrections to Crossover Times

The form of F suggests a sharp discontinuity in the slope of $\mathcal{F}$ at t_c. However, for small system sizes, this was not observed, with errors primarily arising from the approximations made about the lattice length as a continuous variable. In order to add corrections to scaling for crossover times, the area of the pile was redefined as the sum of the heights at each location i Since the algorithm defines local slope z_i as the difference between consecutive heights, we can write t_c as:

$$t_c = \sum_{i=1}^{L} z_i \cdot i \tag{6}$$

Making the assumption that $z_i =< z >$, we obtain:

$$t_c =< z > \sum_{i=1}^{L} i \tag{7}$$

and hence,

$$
\begin{aligned}
t_c &= < z > \frac{L(L+1)}{2} \\
&= \frac{< z >}{2} L^2 (1 + \frac{1}{L})
\end{aligned}
\tag{8}
$$

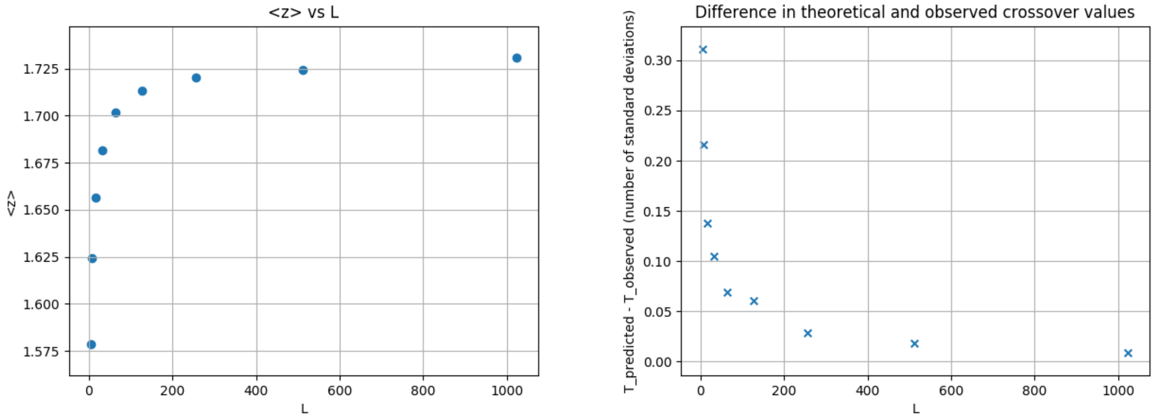

Figure 3: The difference in theoretical and observed crossover times (in terms of standard deviations) was plotted against system size L, and the result showed that the model tended to theoretical values for large system sizes

In order to verify this, python code was written to extract $< z >$ values and corresponding standard deviations for different system sizes. It was found to asymptotically tend to a value of ≈ 1.732, which was in good agreement with Christensen. et. al[2]. It was then compared against the predicted values, and the difference (in terms of standard deviations) was plotted as a function of system size, and the graph shown in Fig.3 was obtained.

This was found to be in agreement with the theoretical argument, with the model getting better for larger system sizes. It must also be noted that the standard deviation also decreased as the system size increased (discussed later), and hence, the fact that the difference still keeps getting smaller means that the theoretical argument works for large system sizes. However, for small system sizes, extra correction terms must be still be added, to possibly account for the quantization of slopes, as the magnitude of the slope is comparable to system size (and consequently height).

2.5 Task 2E: Corrections to Height Scaling

As the height of the pile is intimately linked with the crossover time, corrections to scaling of crossover time imply the existence of similar correction terms in the scaling of height as well. These terms were expressed as arbitrary exponentials of L , each weighted with an arbitrary constant, as follows:

$$< h(t; L) >= a_0 L(1 - a_1 L^{-w_1} + a_2 L^{-w_2} + ...) \tag{9}$$

Parameter	Linear Fit	L - M Fit
a_0	1.735	1.7334 ± 0.0007
a_1	0.579	0.25 ± 0.04
w_1	0.6575	0.63 ± 0.04

Table 1: The parameters a_0 and w_1 agreed with each other from both methods. Discrepancies were observed in the value of a_1, arising due to the base of the log used in the linear fit.

Terms with index 2 and above were neglected for the subsequent analysis, primarily owing to lack of data points for satisfactory analysis.

From the equation, it can be inferred that for large values of L, the height does scales linearly with height. From equation Eq.1 it has also been proven, that given certain approximations, $h \approx < z > L$, and the value of $< z >$ was asymptotically tend to 1.732.

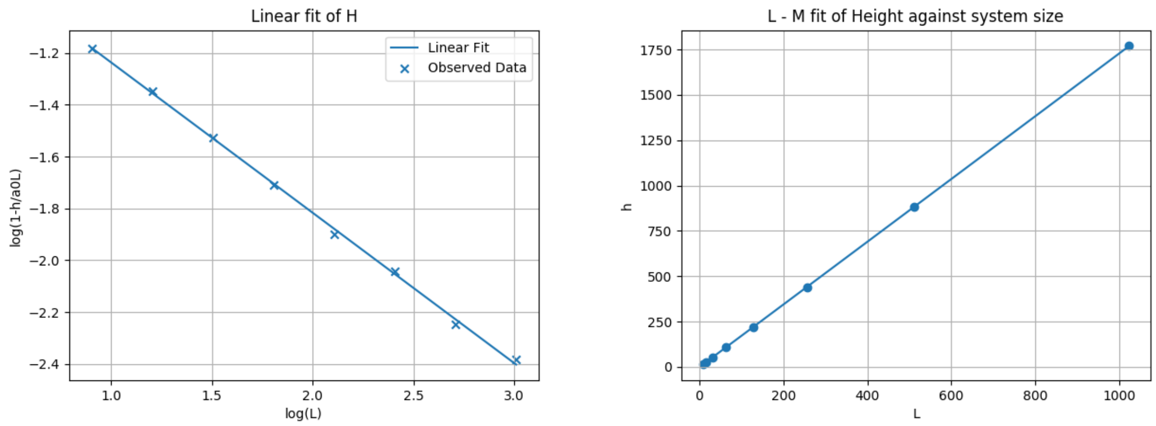

Figure 4: Log-Log plot of height vs time: There was an initial exponential increase and then stayed constant

[!h]

This proved to be a good starting point and analysis was started with an initial guess of 1.732 as the value of a_0. The above equation was further manipulated, and $log(1-\frac{h}{a_0 L})$ was plotted against $log(L)$. The value of a_0 was varied until a fairly linear curve was obtained, and a linear fit was performed. However, since the setting of a_0 was done by eye, error analysis could not be satisfactorily completed, as it was possible to offset extremely low errors in a_0 with extremely high errors in w_1. Hence, as a sanity check, a damped least squares fitting was also performed on the original curve, using the Levenberg-Marquardt (L-M) algorithm. This was done using the `scipy.optimize.curve_fit` library, and no initial assumptions or guesses were made about any of the values. The second analysis also provided corresponding errors.

Table 1 compares the results. It was interesting to note that the values of a_1 did not agree in both cases, and can attributed to the fact that its value depends on the base of the logarithm taken. However, both methods provided us with fairly significant non zero terms. This strongly points to the existence and need for correction terms.

2.6 Task 2F: Standard Deviation of Height

After studying how height scales with length and subsequent corrections with scaling, deviations from mean height was studied to see the extent to which systems deviated from the mean in the steady state. This was done by plotting the standard deviation σ_h against system size. A linear plot was obtained on the log-log axis, suggesting an exponential increase. It was assumed to be of the form $\sigma_h = aL^b$ and subsequent fitting provided values of $a = 0.25 \pm 0.01$ and $b = 0.251 \pm 0.006$.

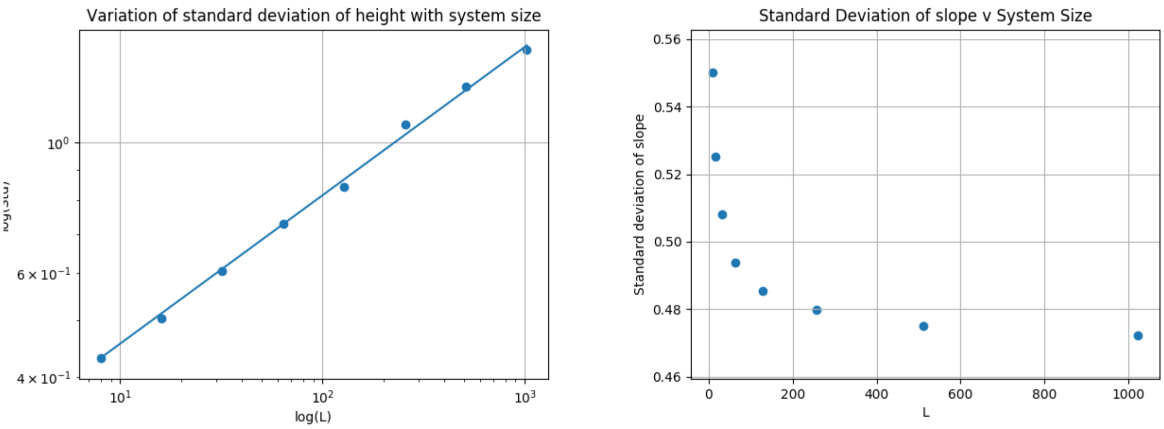

Figure 5: The standard deviation increased exponentially with system size, translating to an exponential error in the standard deviation of slope.

As has been shown earlier, the height scales linearly with system size for large L, and is of the form

$$h(L) =< z > L \tag{10}$$

upon partially differentiating both sides in order to obtain errors, we get,

$$\sigma_h = \sigma_z L \tag{11}$$

As shown in the previous paragraph, σ_h scales exponentially with system size. Hence, for large L,

$$\sigma_h = 0.25 \cdot L^{0.251} \tag{12}$$

on substituting this in Eq. 5, we get

$$\sigma_z = 4 \cdot L^{-0.749} \tag{13}$$

However, the above equation could not be satisfactorily fit with the obtained values, as for smaller values of L, the error was dominated by quantization of z and other scaling correction terms. However, the exponentially decaying errors for large values of L did prove reassuring.

2.7 Task 2G: Data Collapse of Height Probability

According to the central limit theorem, if z_i are indeed independent random variables, a Gaussian should be obtained for the height of the pile. In order to verify this, the generated height data was analyzed using `pandas.series` library, and probability distributions were generated. As noted in the previous sections, for each system size, there were different mean heights, and each of these distributions had their means and standard deviations. Note: The graph of the height probability distributions was not included as it did not provide any extra information.

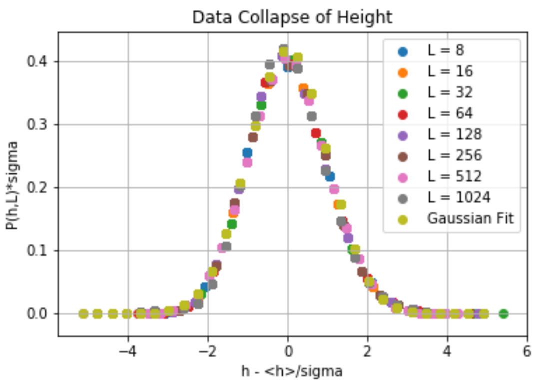

Figure 6: The standard deviation increased exponentially with system size

However, if they were indeed Gaussian as expected, they would have the form

$$P(h;L) : \mathcal{G}(h(L)) = \frac{1}{\sigma_h\sqrt{2\pi}}e^{(\frac{h-<h>}{\sigma_h})^2} \tag{14}$$

where $< h >$ is the mean height and σ_h is the standard deviation. Hence, in order to obtain a data collapse, the X axis was shifted for all system sizes to have a mean of 0, and was also divided by their standard deviations. The Y axis was multiplied by σ_h and the following graph was obtained. At first sight, it seems quite obvious that the Gaussian approximation works fairly well. However, upon careful examination, it can be seen that the collapsed data is not symmetric, as can be expected from a Gaussian. This suggests that the values of z are not truly independent of each other, and requires further investigation.

3 Avalanche-Size Probabilities

3.1 Task 3A: Log - Binning of Data

The analysis was now shifted toward analyzing the size of the avalanches generated after relaxation. On plotting avalanche sizes against their probability distribution, a linear plot was obtained on a log-log scale. It however had a very noisy tail and no further analysis could be done. Hence, in order to gain more insight into the data, log binning was performed. Similar to histograms, log binning groups data but in exponentially increasing bin sizes.

For example, the j^{th} bin covers the interval $[a^j, a^{j+1}[$. If s^j_{max} and s^j_{min} are the maximum and minimum avalanche size values of each bin, them the following bin characteristics can be defined:

$$\tilde{P}_N(s^j) = \frac{Avalanches\ in\ bin\ j}{N\Delta s^j} \tag{15}$$

and

$$s^j = \sqrt{s^j_{max} s^j_{min}} \tag{16}$$

where s^j is the geometric mean of the maximum and minimum avalanche values of the bin and $\tilde{P}_N(s^j)$ is the corresponding probability.

Upon log binning of the data, the noise was removed and characteristics could be studied. It was observed that every system had a characteristic cut-off for avalanche sizes,

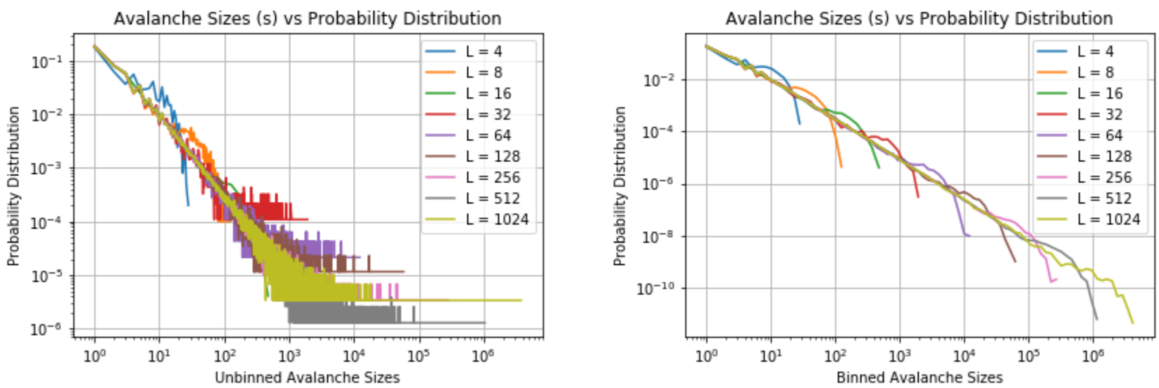

Figure 7: Unbinned and Binned avalanche data: The noise was reduced and finer characteristics could be studied upon binning the data.

which made intuitive sense as there cannot be infinitely large avalanches for finitely large system sizes. It could be seen that the probability of avalanche sizes reduced exponentially with size, apart from a characteristic bump before the cut off. This could be attributed to system spanning avalanches.[2]

3.2 Task 3B: Data Collapse of Avalanche Size Probability

The Finite Size Scaling Ansatz dictates that for sufficiently large system sizes,

$$\tilde{P}_N(s, L) \propto s^{-\tau_s}\mathcal{G}(\frac{s}{L^D}) \tag{17}$$

Data collapse was attempted by plotting $\tilde{P}_N(s, L) \cdot s^{-\tau_s}$ against $\frac{s}{L^D}$. Initially, the value of D was set to 0 and only the Y axis was collapsed. Upon satisfactory collapse of the Y axis, τ_s was kept constant and the value of D was changed until collapse was achieved on the X axis.

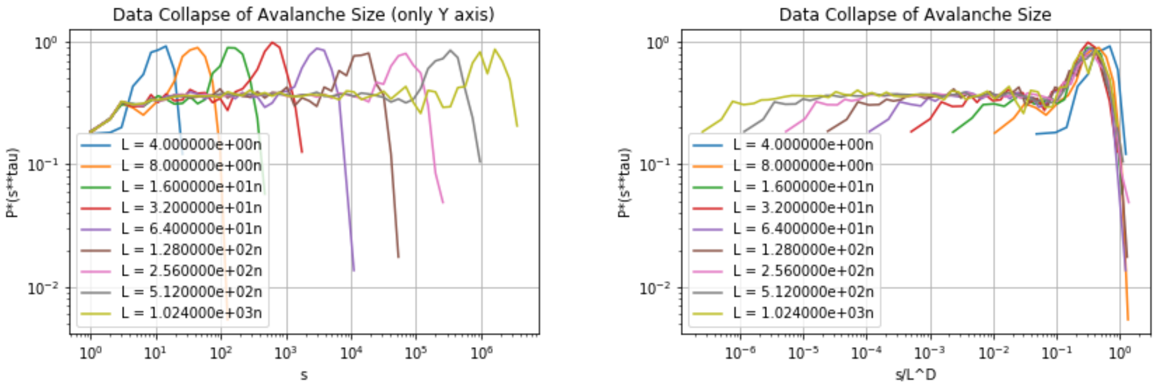

Figure 8: The Y axis was first collapsed in order to obtain a value for τ_s followed by subsequent collapse of the X axis to obtain a value for D

The value of τ_s was found to be 1.55 ± 0.02 and the value of D was found to be 2.19 ± 0.03.

3.3 Task 3C: Moment Analysis

The k^{th} moment of an avalanche can defined as follows:

$$< s^k >= \lim_{T\to\infty} \frac{1}{T} \sum_{t=t_0+1}^{t_0+T} s_t^k \tag{18}$$

where s_t is the measured avalanche size at time t.

Theoretically,

$$< s^k >= \sum_{s=1}^{\infty} s^k \cdot P(s, L) \tag{19}$$

Using the FSS, this can be written as

$$< s^k >= \sum_{s=1}^{\infty} s^{k-\tau_s} \mathcal{G}(\frac{s}{L^D}) \tag{20}$$

Re-scaling $\frac{s}{L^D}$ as u, and approximating the sum as an integral, we get

$$< s^k >= L^{D(1+k-\tau_s)} \int_{u=1}^{\infty} u^{k-\tau_s} \mathcal{G}(u) du \tag{21}$$

The integral drops out as a constant number. On plotting the avalanche moments against system size L for $k = 1, 2, 3, 4, 5$, the following plot was obtained. For each plot of s^k and L the slope obtained is given by $D(1 + k - \tau_s)$. Hence, the slopes of each of these graphs

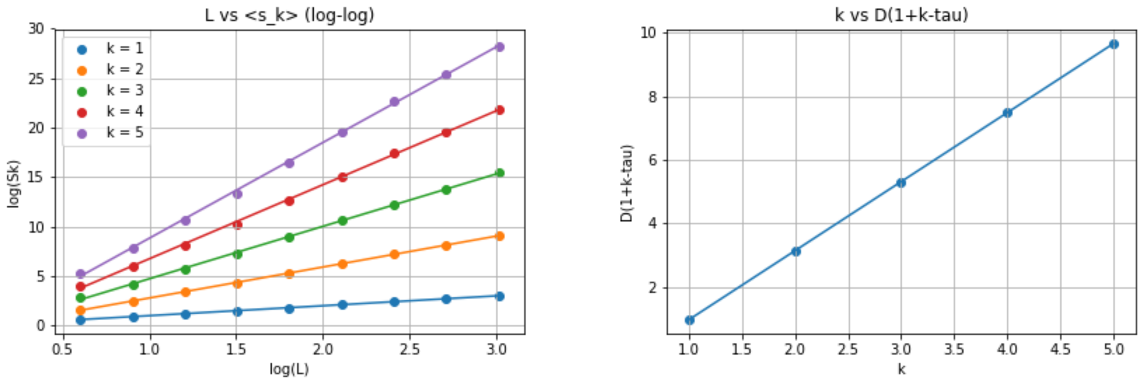

Figure 9: In the left panel, the moment was plotted against system size for different values of k. The slope of each of the lines was extracted and plotted against k to extract D and τ_s

was extracted and plotted as a function of k. A linear relation was obtained, with length a slope of D and an intercept of $D(1-\tau_s)$. This yielded a value of $D = 2.168 \pm 0.006$ and $\tau_s = 1.550 \pm 0.009$.

The theoretical and observed data were also compared with each other to check whether there any corrections to scaling. The following graph was obtained: Discrepancies were observed for small system sizes, but as $L \to \infty$, all moments plateaued out to a constant value. For k= 1, a straight line was expected as both of the methods should point to the mean avalanche size, without any scaling corrections. However, this was not the case, and the errors possibly arose from the the errors in estimation of D and τ_s. For all other moments, the ratio tended to a constant value (which depends on the integral term in Eq.21, showing that theoretical and predicted values agree with each other.

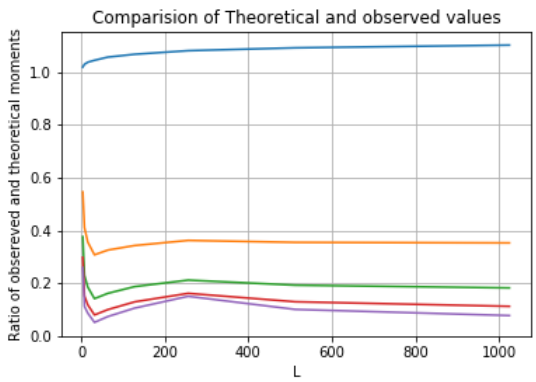

Figure 10: The ratio of theoretical and predicted values of moments were plotted against system size. For large system size, the ratio were in good agreement, but had large fluctuations at small sizes

Conclusion

Self-Organized criticality was successfully studied by simulating a rice pile for different system sizes, providing understanding of how height and avalanche sizes scale with system size, and their consequent corrections to scaling and deviations from ideal behaviour for system sizes of $L = 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024$. Scaling relations for height and corrections were obtained. Upon direct probability collapse and indirect moment analysis of avalanche size data, the values of $D = 2.168 \pm 0.006$ and $\tau_s = 1.550 \pm 0.009$ were obtained.
Future prospects involve rewriting the simulating code in a compiler based language such as C++ in order to aid quick simulations and obtain more data for analysis. Extra corrections to scaling could be investigated theoretically and computationally in order to provide a more accurate model. However, this provided valuable insight into self organized criticality, and the inner workings of some abstract physical, social and even biological systems.

References

References

[1]Biondo, A., Pluchino, A., Rapisarda, A. (2015). *Modeling financial markets by self-organized criticality.* Physical Review E, 92(4).
[2] Christensen, K., Moloney, N. (2005). *Complexity and Criticality.* London: Imperial College Press.