Complexity Project
The Oslo Model

John Norrie

CID: 01289535

Abstract

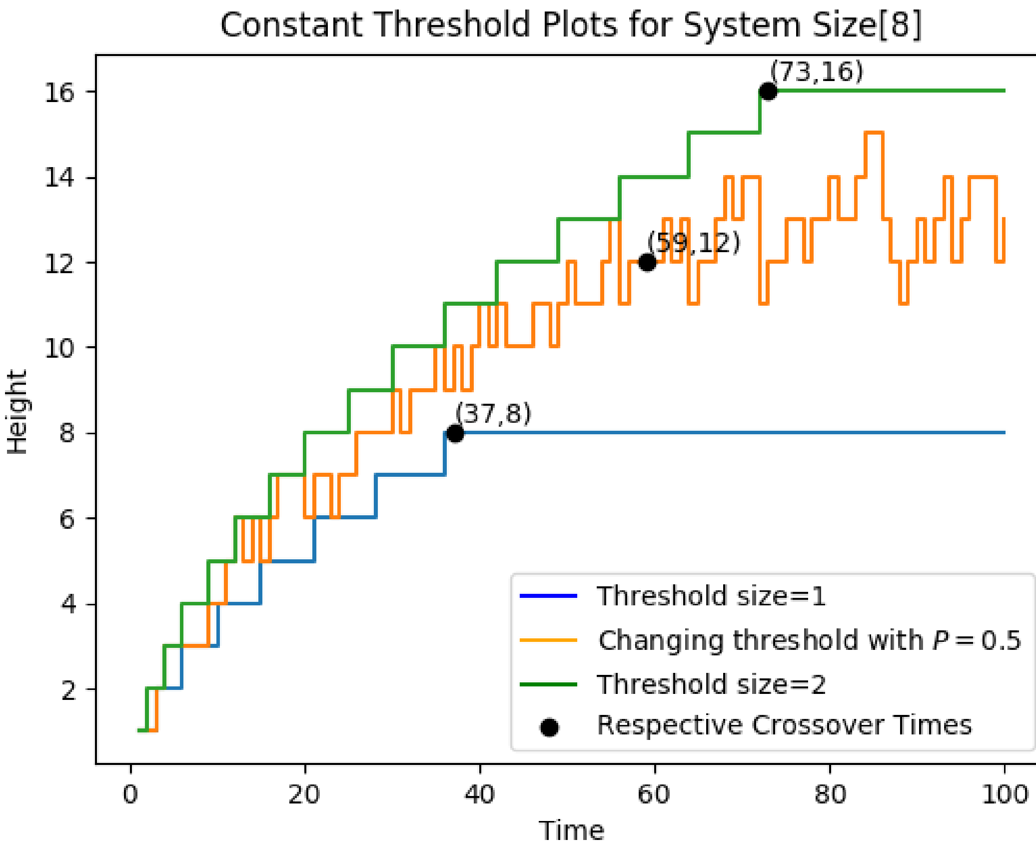
From the 1980s onwards, sand piles became a subject of great interest as the area of dynamical systems and complex systems grew. The problem of the creation of a sandpile may seem straightforward at first glance, but these systems represents a fundamental case of self-organized criticality (SOC). The principles and logic used in the analysis of SOC can be applied to a range of other systems which exhibit similar features. This includes systems such as earthquakes, rainfall, and ant colonies. One model that exhibits SOC particularly well is the Oslo model. One of the key features of the Olso model is it’s ‘slip-stick dynamics’. This is what sets it apart from the Bak–Tang–Wiesenfeld model(BTW).

Word Count 2,432

<u>Introduction</u>
During this project I aim to successfully create a program which simulates the Oslo Model. After which I shall perform a series of tests to better understand the problem and area of self-organising critical systems.

Testing

Fig. 1.0



First to test the model I concentrated on a system size of *L=8*. This was to make the system as simple as possible so any errors could be quickly spotted. First I set the probability of a threshold changing to 2 after a relaxation, *P(Thresh)=0*. From doing this we recover the Bak–Tang–Wiesenfeld model(BTW) with a threshold of 1. This can be seen on fig1.0 above. As you can see the height increases until it hits a maximum of 8, at which point it can go no higher as all slopes are equal to their threshold. After this point any grain added will roll down to the bottom slot and out of the system. The crossover time is the time at which the system changes between being in transient states(states that the system will not return to after leaving), into recurrent states(states that the system may return to). In theory, for the BTW model the time taken until we reach the recurrent states is $T_c=L(L+1)/2$. This is because the largest possible number of grains in the first row is L, L-1 in the second,…, 1 in the L^{th} row. Taking the sum of this: $\sum_{n=1}^{L} n=L(L+1)/2$,we retrieve the formula. For example, L=8, we get 36. The crossover times we actually recover however is $T_c=37$. This is because a more accurate way to measure the crossover time is to measure when the first grain leaves the system, rather than when the system fills up.

Setting *P(Thresh)=1* gives use a BTW model with step size 2. We have the same characteristics as before however the maximum height and T_c-1 have both doubled. This is because the capacity of grains in the system has doubled.
The third graph I have plotted occurs when I set *P(Thresh)=0.5*, and is the model I will be testing throughout this project. It acts similarly to the BTW models, but with added ‘noise’ due to the random changing of threshold size. Even with this noise notice that the total height always stays bounded between the other 2 curves. This is because the 2 cases before express the minimum and maximum possible values of total height respectively.

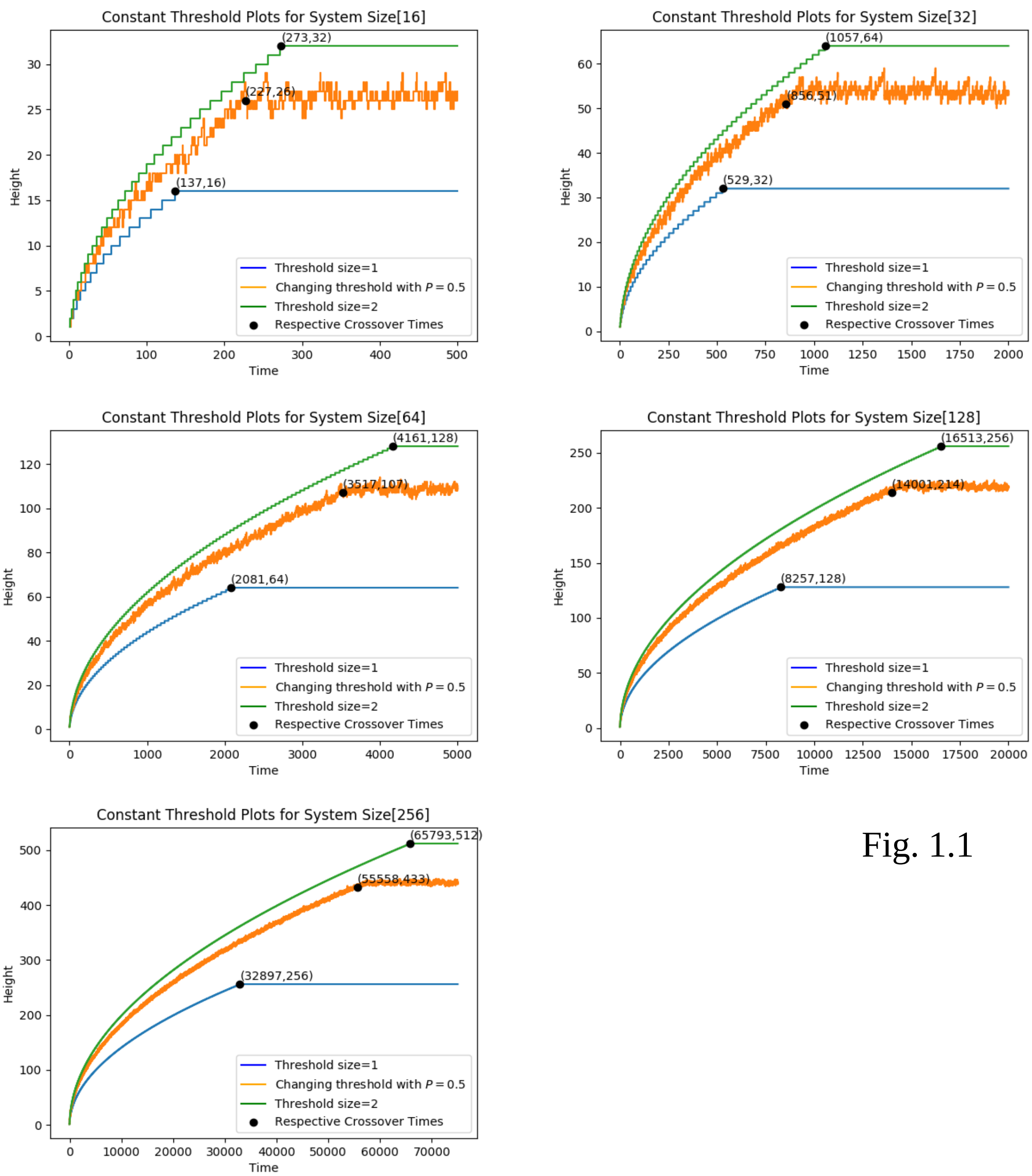


Fig. 1.1

Along with $L=8$, I shall also be testing system sizes, $L=16,32,64,128,256$. fig1.1 shows how the total height scales with time for each L, and $P(Thresh)=0, 0.5, 1$. At a glance, one might expect the model to produce a total height of exactly halfway between that of the BTW models at $h_T=1.5L$, as one would expect the average slope to be 1.5. However, since the thresholds=1 fill up quicker and therefore change more often we are more likely to find a threshold=2, meaning that the average slope is greater than expected.

Task 2

Fig. 2.0

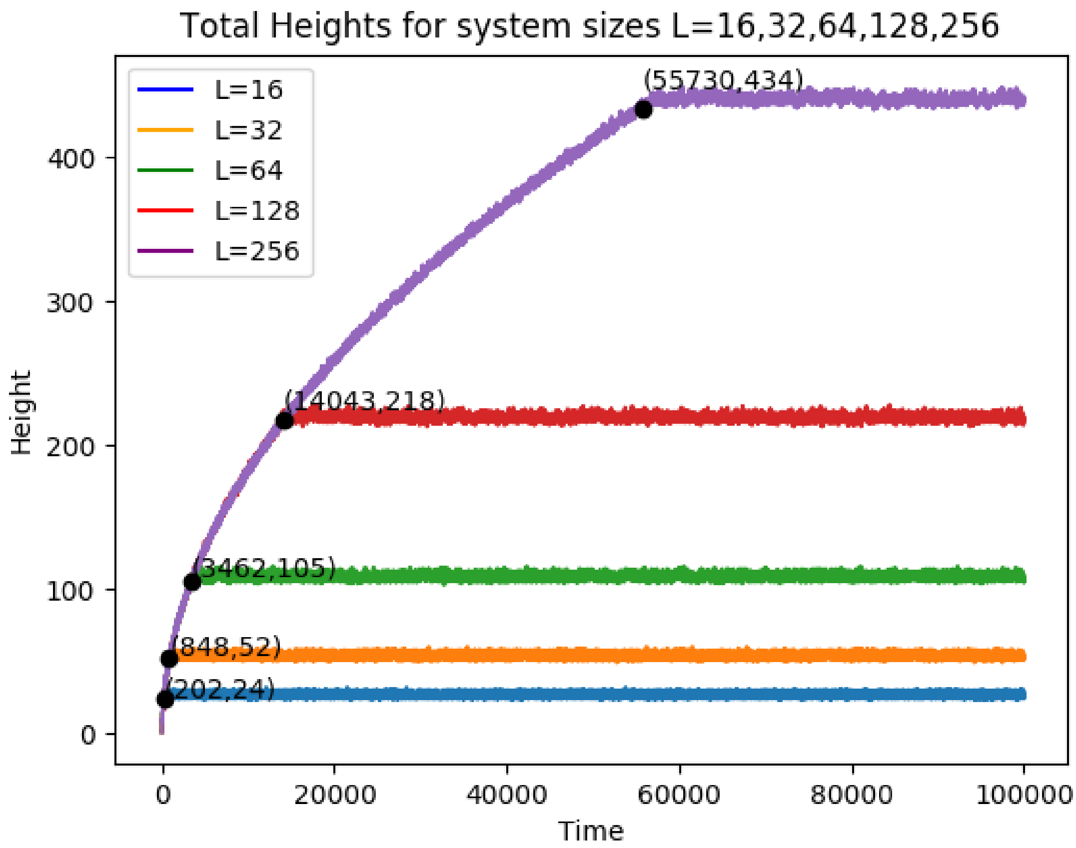


Fig.2.0 shows a plot of the heights against time for our model with *P(Thresh)=0.5*. Before the crossover time the system is going through it's transient states and the total height increases until the crossover time. At which point, the system reaches it's recurrent states. Unlike the BTW model, the system has more than one recurrent state, and fluctuates rapidly. This is caused by avalanches which take place due to the random change of threshold size after relaxation.
Physically we would expect the total height, $h_T=aL$ where a is the average slope for a given position L. This is because we have L 'slots' for the grains to settle in, and if the height is rising by a in each slot, then the total height will be aL(Fig.2.1).

Fig. 2.1

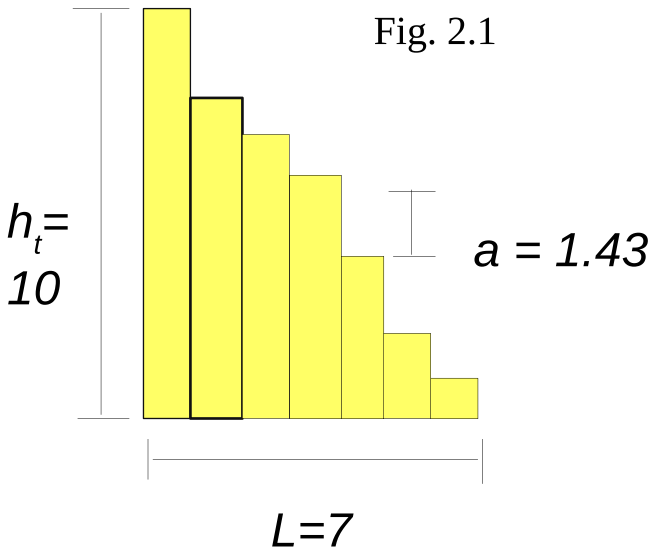
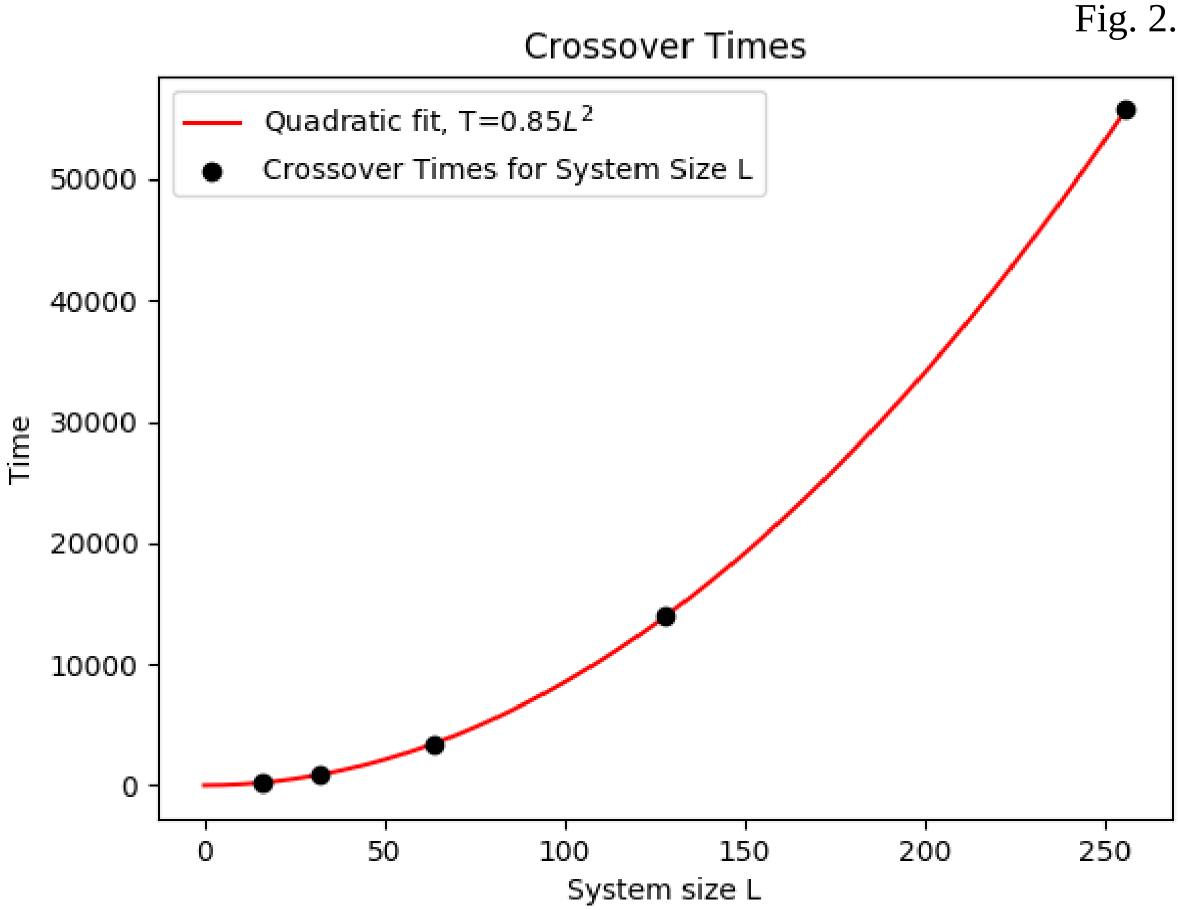
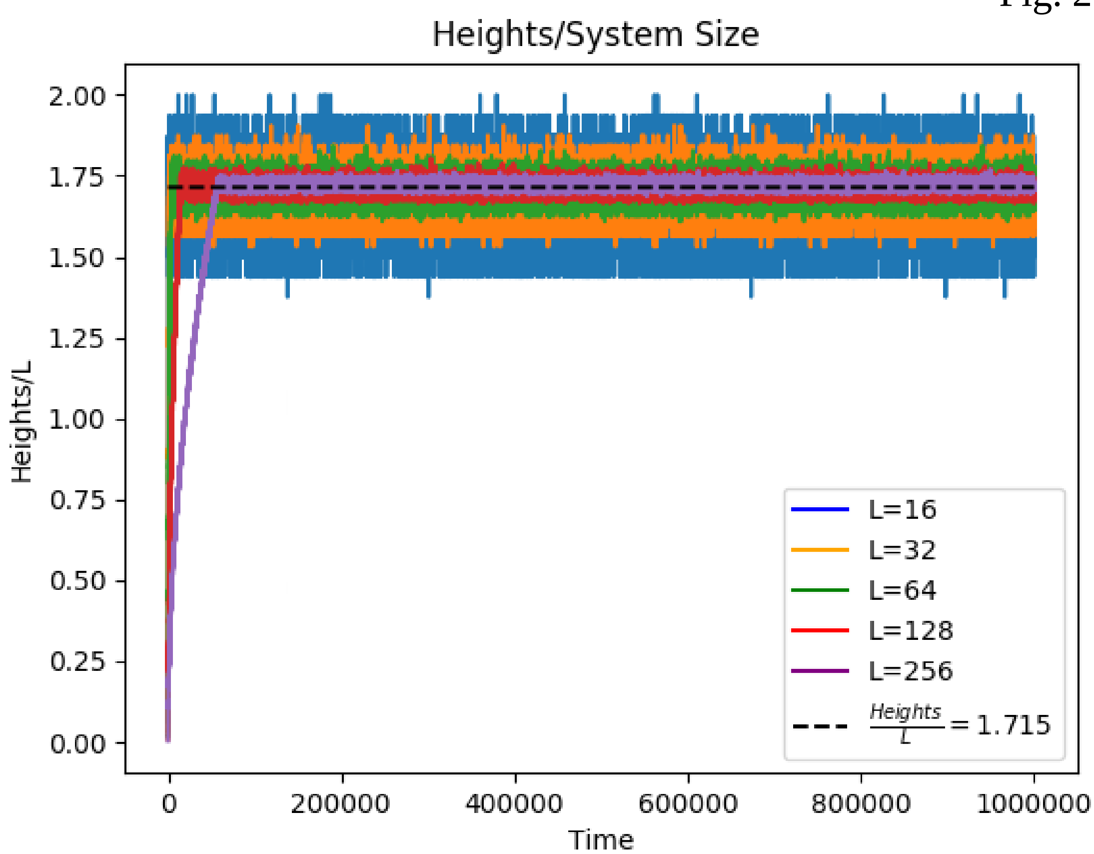


Fig. 2.2



As shown in fig2.2, the crossover times scale quadratically with system size. Thinking logically about what t_c physically means, we would expect $t_c \approx a(L)^2/2$, where a is the average slope (e.i. t_c is approximately the time taken to for a right-angled triangle of lengths aL and L to be filled by

Fig. 2.3

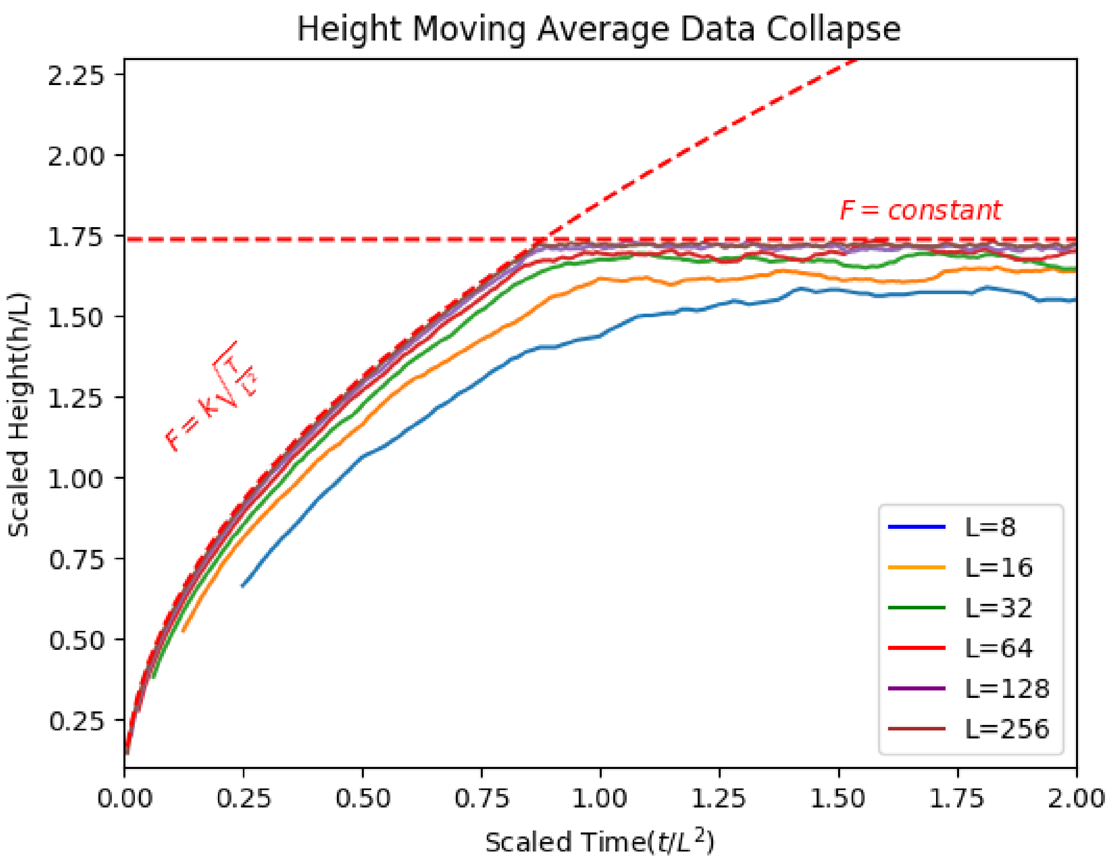


grains). This gives a first approximation of the average slope at roughly *1.7*. Fig 2.3 shows the heights of the piles divided by the system size *L*. Notice that in the steady state the heights roughly collapse around *1.715*. This tells us that the heights scale linearly with *L*(*e.i* $h_T \propto L$). It also gives us an even better approximation of the average slope.

2b

Developing on ideas discussed in **2a**, I wish to find the average slope more accurately, and a method to collapse the data. Firstly, I created a moving average for my heights, which took an average over a window of length *4L*, for each height entry. This smooths out the data to help better see any relationships. From **2a** and **2b** I already know that height scales linearly, and the crossover times should scale quadratically with system size. This gave me an idea of how one might collapse the data: I stretched the time axis by a factor of L^{-2} and the heights by a factor of L^{-1}. Figure 2.4 is the outcome.

Fig 2.4



There is some correction to scaling, however, one can clearly that as the system size increases, the transformed height tends towards a scaling function:

$$F(t/L^2) = \begin{cases} k\sqrt{t/L^2} & , t<t_c \\ constant & , t>t_c \end{cases} \qquad Where \quad h_T(t) = L * F(t/L^2)$$

The equation above shows how *F* acts for small and large arguments of *t*. Also, using this relationship we can predict that $h_t \propto \sqrt{t/L^2}$ in the transient phase($t<t_c$).

2c

In this part of the question, we shall investigate the mean height($\langle h \rangle$) and standard deviation(σ) for $t>t_c$. To obtain accurate measurements I recorded the total height over 1 million time steps and only started record steps at t_c+100. From **2b** we know that $h \propto L$ with some correction to scaling, and so we expect $\langle h \rangle$ to do the same. To account for correction to scaling, we assume that mean height follows a relationship:

$\langle h \rangle = a_0 L(1 - a_1 L^{-\omega_1} + a_2 L^{-\omega_2} + ...)$. I.e. <h> is linear(as shows in **2a** & **2b**) with some additional terms which become negligible as $L \to 0$. I approximated a_0, a_1 and ω_1 using the following method:

$$\log(u) = \log(a_1) - \omega_1 \log(L) \quad where \quad u = 1 - \frac{\langle h \rangle}{a_0 L}$$

From this one can approximate a_0. We want a linear relationship between the *log(u)* and *log(L)*, this would imply we have found a correct power law. We vary a_0 to obtain this. If a_0 is too small, *u* will become negative as $L \to \infty$ and therefore *log(u)* will become undefined. If a_0 is too large then the *log(u)* $\to$ *constant* as $L \to \infty$. Fig2.5 shows the trail and error approach I used to estimate a_0.

Fig.2.5

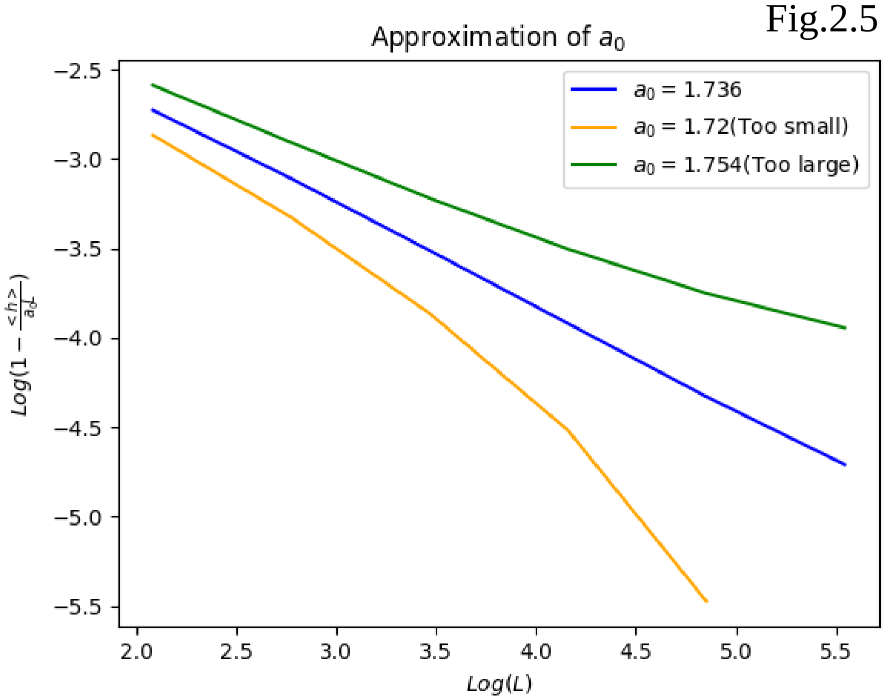
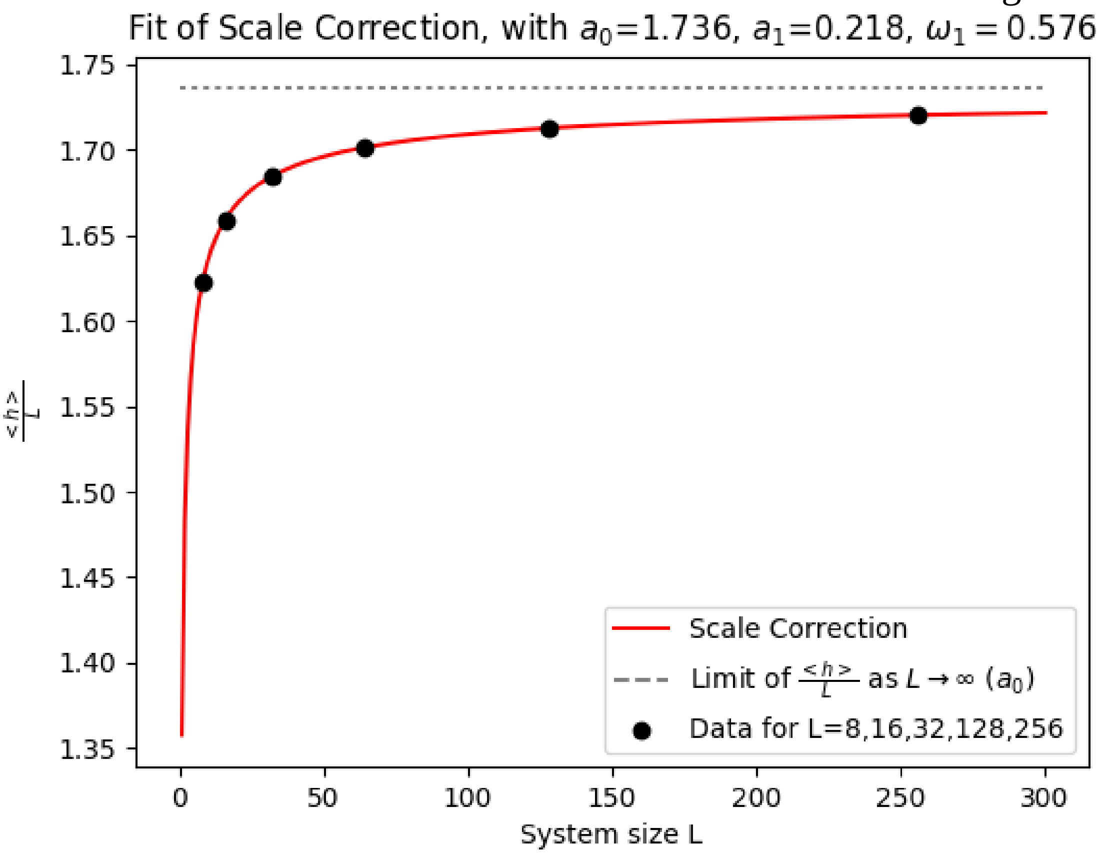
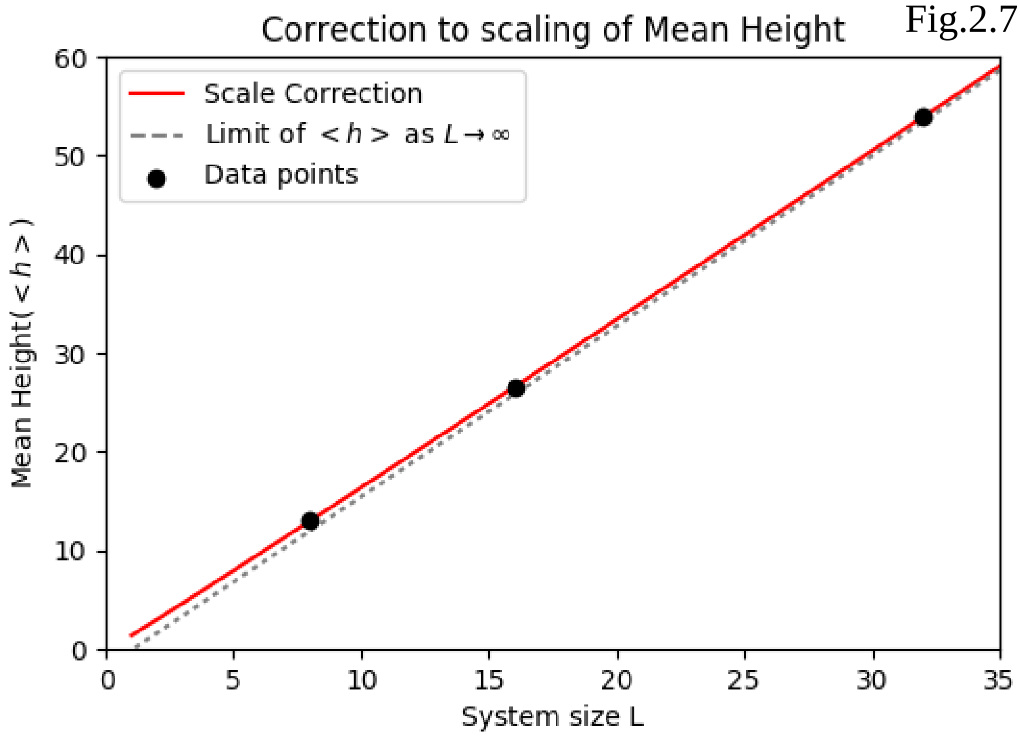


Fig. 2.6



Taking $a_0=1.736$ gives a linear fit. The gradient of this fit gives an approximation of $-\omega_1$, and the y-intercept of the graph gives $log(a_1)$. Numerically, I estimated $\omega_1=0.576$ and $a_1=0.218$. Fig2.6 shows the plot of the $<h>/L$ and the scale correction function $G(L)=a_0(1-a_1 L^{-\omega_1})$. The curve fits the data very well implying that my scaling function is particularly accurate. This is backed up by Fig.2.7 which a plot of mean height vs. system size, with my correction to scaling(red) and a naive

Fig.2.7



fit(grey). Physically a_0 represents the average slope as L→inf and so, $<h> \rightarrow a_0 L$ as $L \rightarrow \infty$.
The standard deviation does not follow a linear fit however, as shown in fig2.8. To find the exponent which relates σ to L I measured the gradient of the log-log graph of the two variables. Doing this I extracted an exponent, $\alpha=0.21$. In the log-log plot however(fig2.9), one can clearly see that the data does not follow a linear fit, this implies that correction to scaling may be involved.

Fig. 2.8

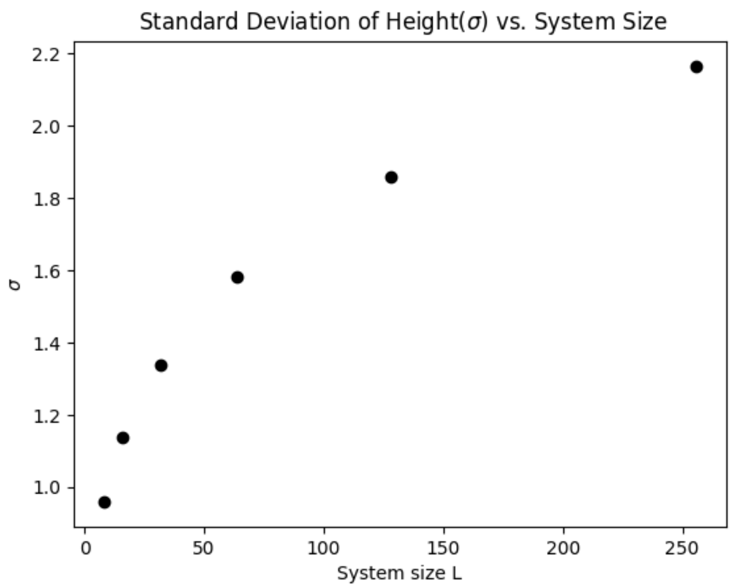
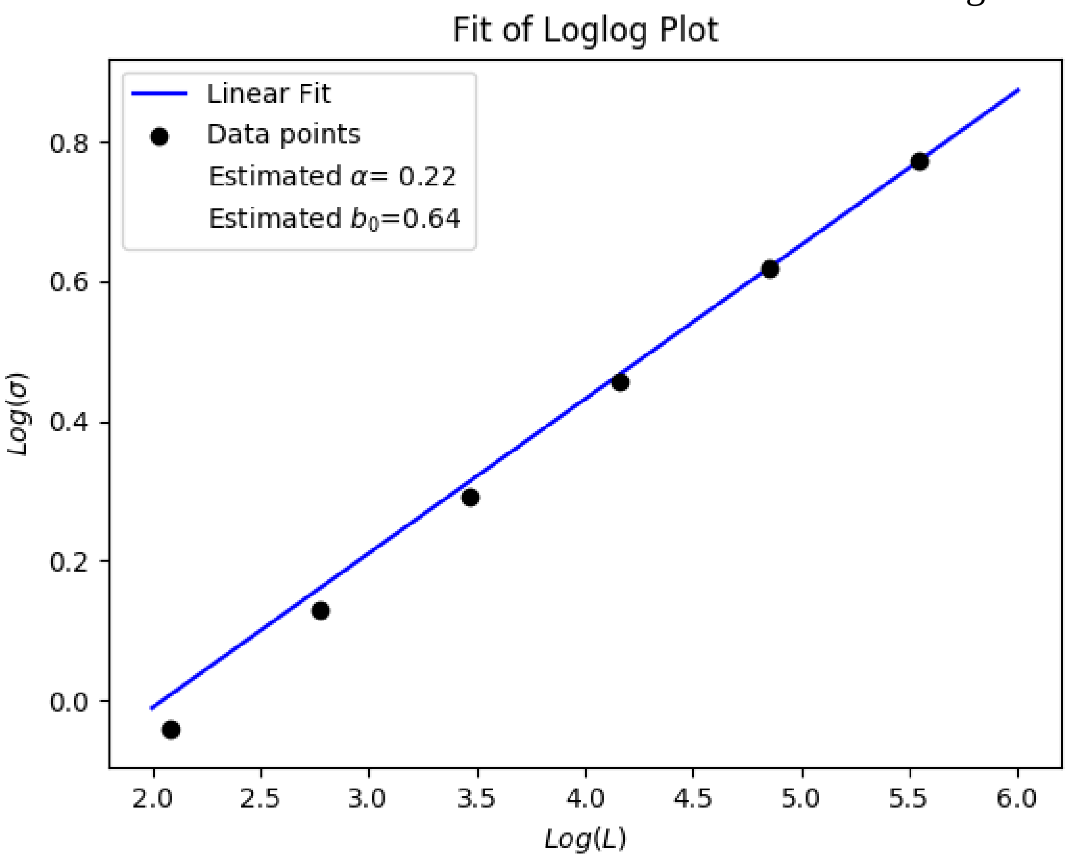
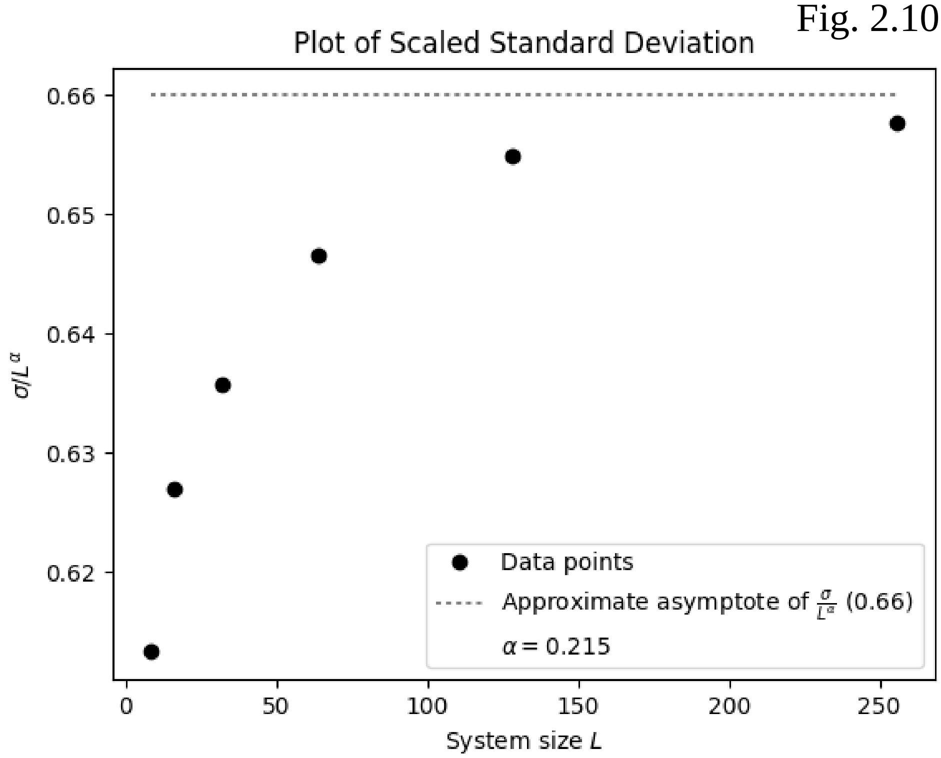


Fig.2.9



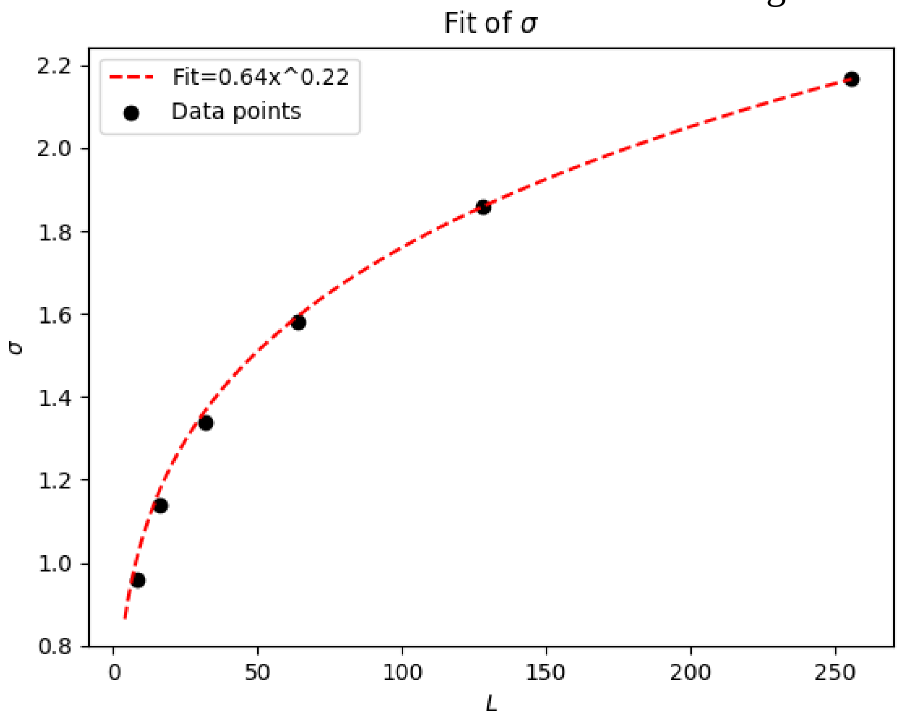
Using our intuition from the mean correction to scaling, and assuming that the standard deviation follows a similar formula, $\sigma = b_0L^{\alpha}(1\text{-}b_1L^{-w}+\ldots)$., which allows us to assume that as $L\rightarrow\infty$, $\sigma\rightarrow b_0 L^{\alpha}$. Figure 2.10 shows how this can be recovered. I have given an example of a possible convergence value(greyline)(b_0) and I possible value for α .However with no real insight into how standard deviation should scale physically, it would be very hard to estimate these values accurately.

Fig. 2.10



Even without this correction to scaling figure 2.11 shows that σ=0.62*$L^{0.22}$ is a fairly good fit. Notice though that the fit and data points deviate slightly as we go closer to origin. This is due to correction to scaling.

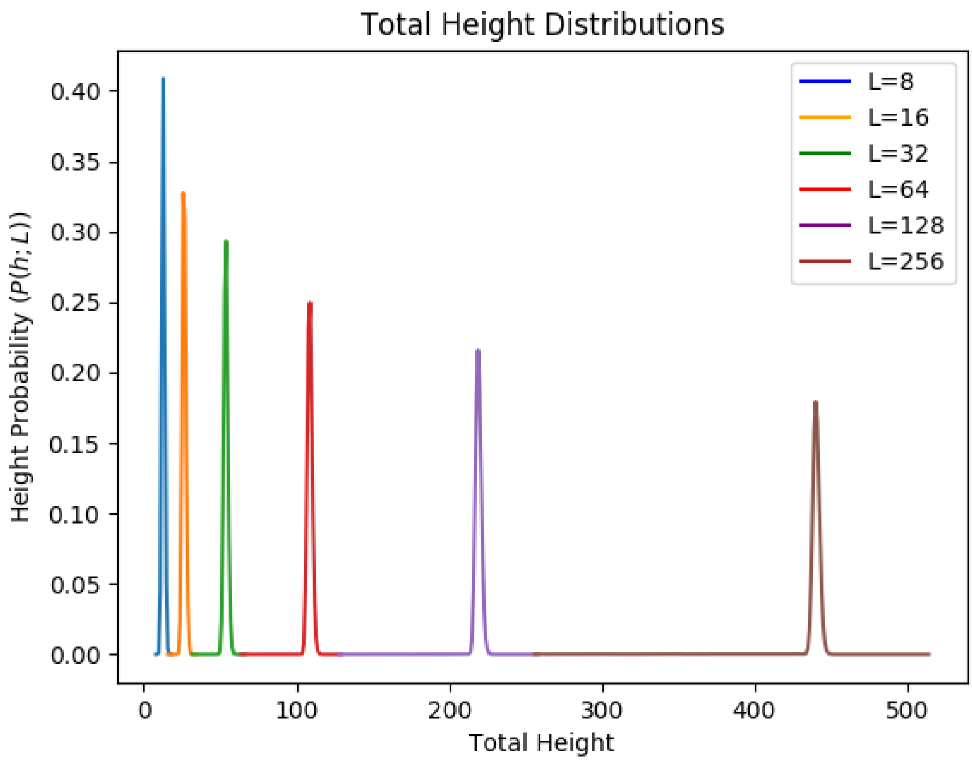
Fig. 2.11



2d

Now we wish to consider the probability that a given total height occurs at a time $t>t_c$ in the system. To do so I sorted the data for each system size, counting the frequency at which each total height occurred, and normalising it by dividing through by the number of observed configurations. Fig 2.12 shows this probability versus the total height for the system sizes stated.

Fig. 2.12



As we can see there produce 6 separate peaks, centred around the mean height of the respective system height. Note that the peaks grow higher closer to the origin, implying that the standard deviation is less for smaller system sizes, which is in accordance with **2c**. A first hint of how to collapse this data is to observe that the peaks look to be Gaussian. If this proposition is true then with the correct transformation this distributions will collapse onto the pdf of the N(1,0) distribution. To explore this we do the following:

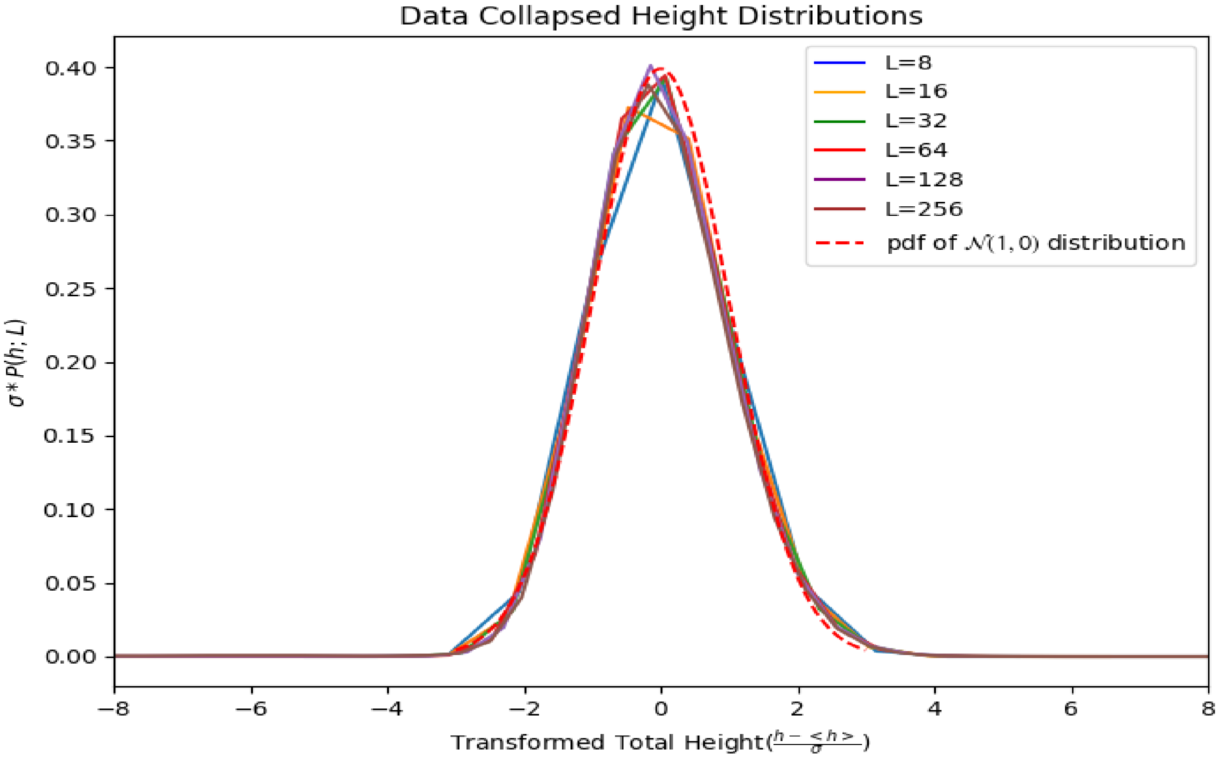
- Transform the total heights thusly: $h \rightarrow \frac{h-\langle h_t\rangle}{\sigma} = \tilde{h}$
- Scale the Probability by: *P(h)* → *σP(h)*

To explain these transformations we need to look at the probability density functions for normal distributions:

$$f(h;\mu,\sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(h-\mu)^2}{2\sigma^2}} \rightarrow F(\tilde{h}) = \frac{\sigma}{\sigma\sqrt{2\pi}} e^{-(\tilde{h})^2} = \frac{1}{\sqrt{2\pi}} e^{-\tilde{h}^2} \quad \text{which is the pdf of } N(0,1)$$

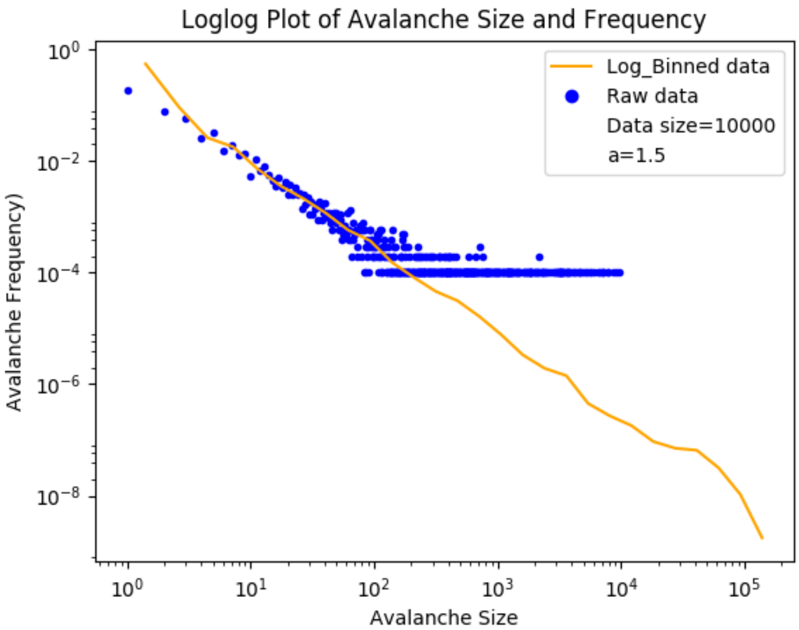
Figure 2.13 shows that using these transformations, the peaks can be collapse onto the pdf of N(1,0) and therefore the total heights follow a normal distributions with means and standard deviations discussed in 2c. This implies that the heights are normally distributed.

Fig. 2.13



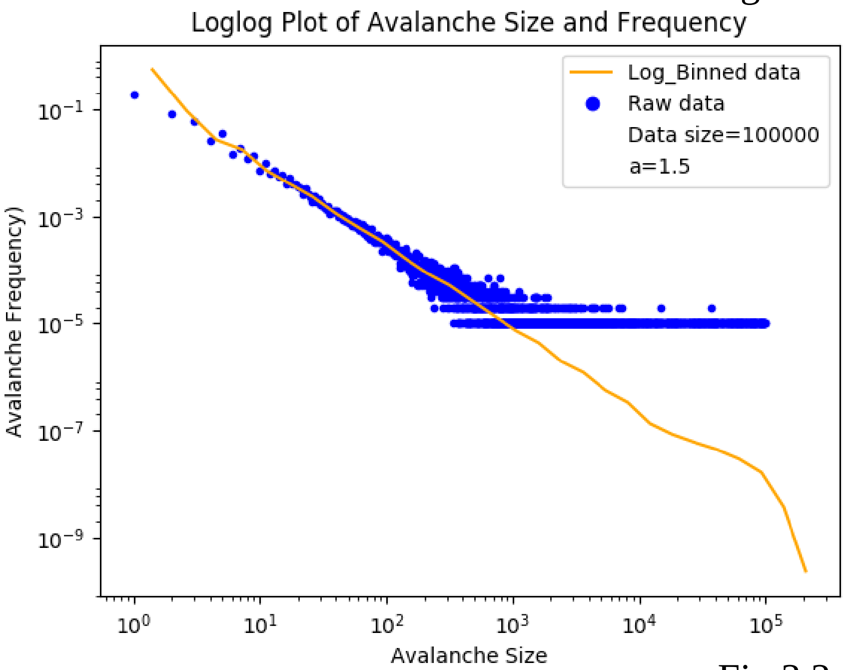
Question 3

I wish to measure the frequency at which avalanches of certain sizes occur. For this we use a function called log_bin. Unlike the frequency plots, which puts the data into equal sized bins, the log_bin function creates bins which grow by a factor of '*a*'. For example if our first bin is size *2*, then our second one is width *2*a*, etc. So at small avalanche sizes, where we have a lot of data, log-bin sorts then very finely into small bins, to give us as much information as possible. Whereas for large avalanche sizes, where data is sparse, the data is put into big bins. What is then plotted is the geometric mean of each bin. Figures 3.0, 3.1and *3.2* show the affect of log-binning the data against the non-processed data for data sizes 10^4, 10^5, 10^6. Again we record data for $t > t_c + 100$. The log_binned data is a lot cleaner, and we are more easily able to infer the relationship involved in avalanche size frequency. The 'bump' that features in the graphs towards the cut-off point s_L is due to having slightly more counts in the last bin, because the is so big compared to the system size. After which there is very little data, and so we get this steep cut off, s_L. The shape of the log_binned data becomes more defined for greater data size, this is due to the fact that there is more avalanche data available. Therefore for the rest of my experiments, I shall data size 10^6, as it is the largest size available to me. An 'a' value of 1.5 worked particularly well in sorting this data.



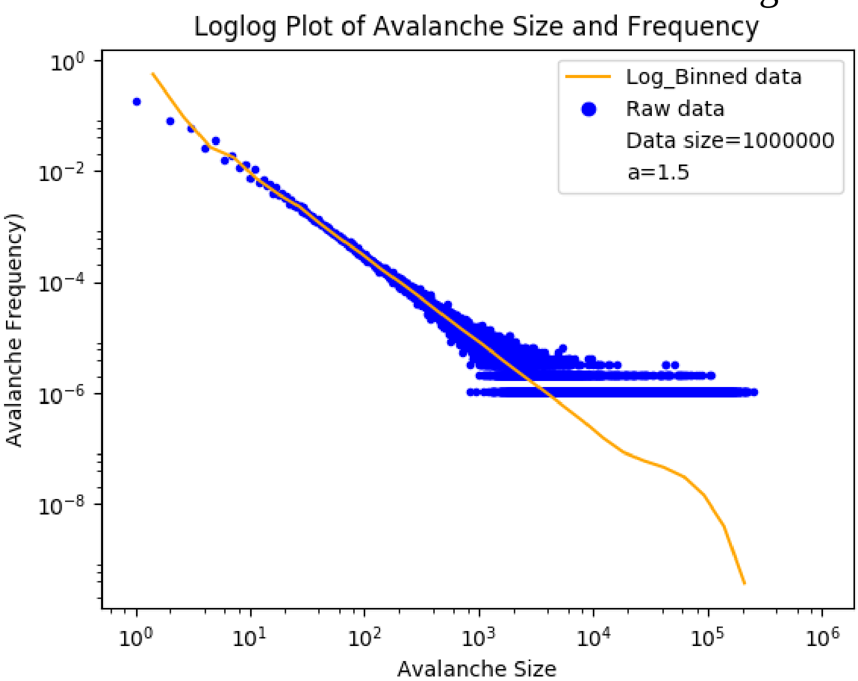
Loglog Plot of Avalanche Size and Frequency
Log_Binned data
Raw data
Data size=10000
a=1.5
Avalanche Frequency)
Avalanche Size

Fig.3.1



Loglog Plot of Avalanche Size and Frequency
Log_Binned data
Raw data
Data size=100000
a=1.5
Avalanche Frequency)
Avalanche Size

Fig.3.2



Loglog Plot of Avalanche Size and Frequency
Log_Binned data
Raw data
Data size=1000000
a=1.5
Avalanche Frequency)
Avalanche Size

To show the effect of changing *a*, fig3.3 shows a plot of $a=1$. As you can see the log-binned data is identical to the original data, as the original bin length is conserved. Conversely fig3.4 shows what happens when *a* is too large, all the data is put into few big bans, not showing us any of the graph's features.

Fig.3.4

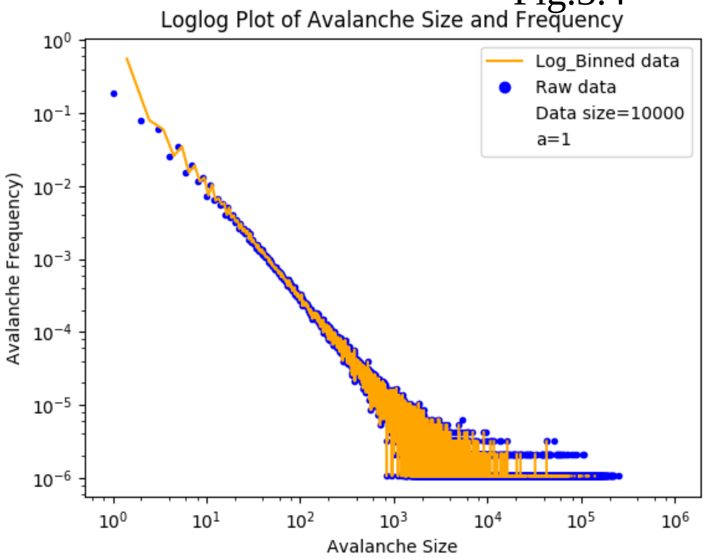


Fig3.5

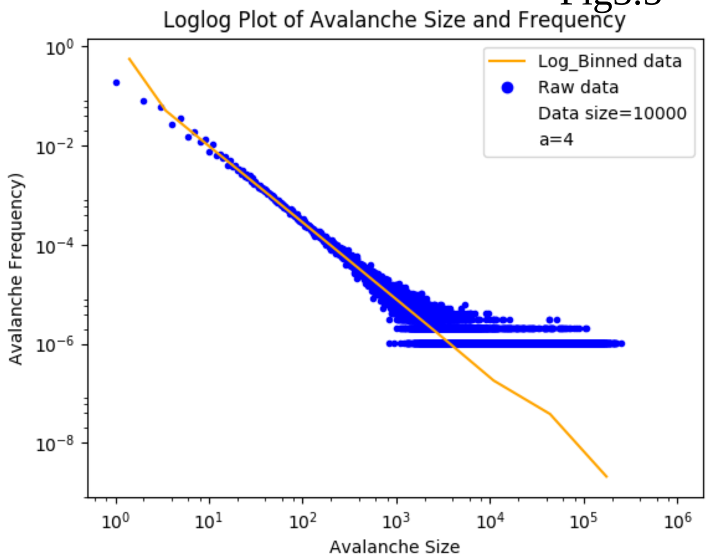
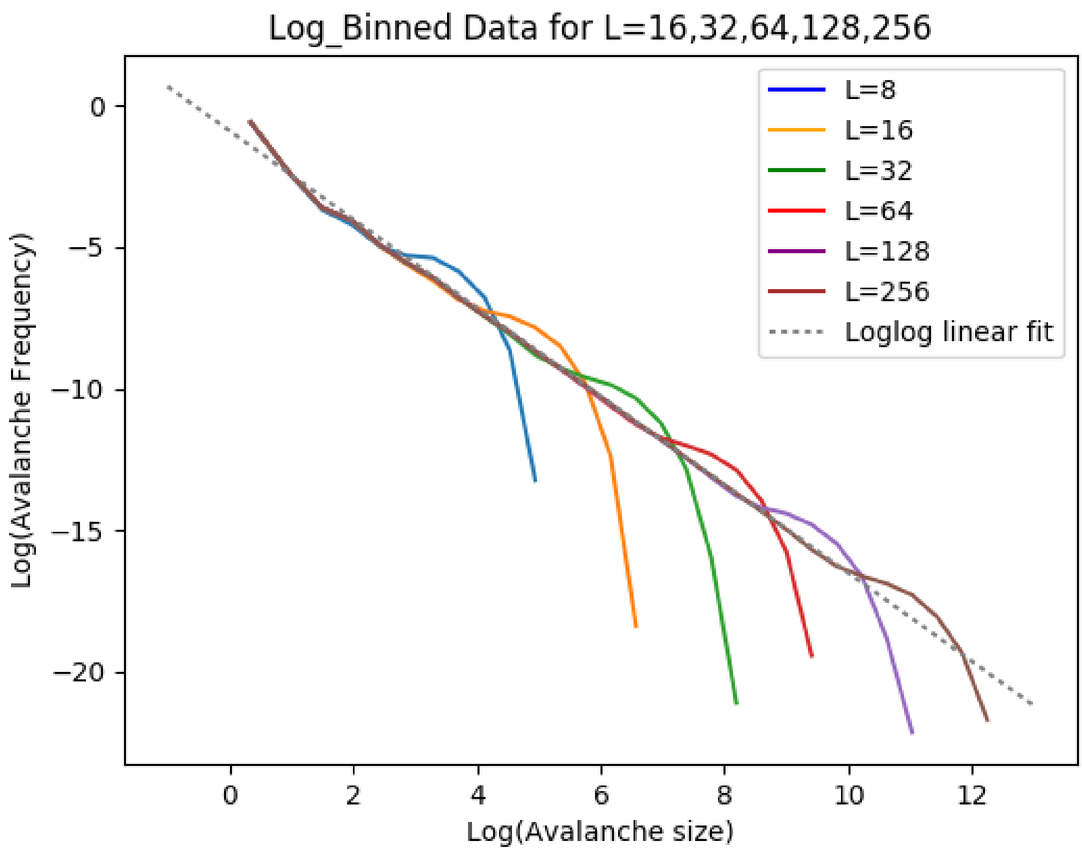


Fig3.6 shows the log-binned data for system sizes L=8,…,256 . The plots follow a common gradient before a critical point s_L when the avalanche size falls dramatically. One can see that these appear to happen very regularly to depending on system size, with $L=8$ dropping off first etc. We would expect this as intuitively $<s>\propto L$. The intuition comes from the fact that $\langle No.\ grains\ added\rangle \propto L$ and $\langle No.\ grains\ leaving\ the\ system\rangle \propto \langle s\rangle$. In the recurrent phase however, we have that $\langle No.\ grains\ added\rangle = \langle No.\ grains\ leaving\ the\ sytem\rangle$, and so $<s>\propto L$

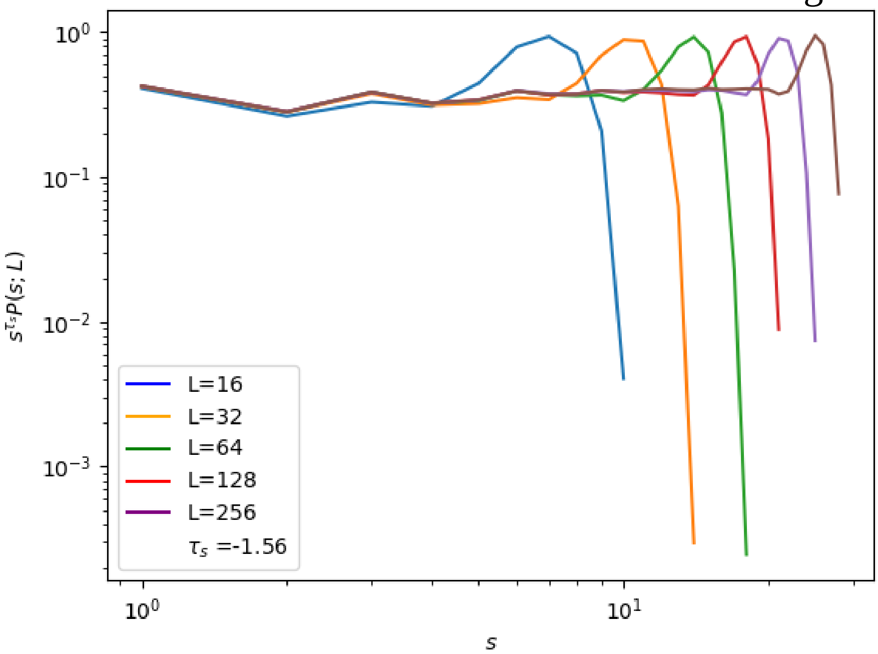
3C

Fig.3.6



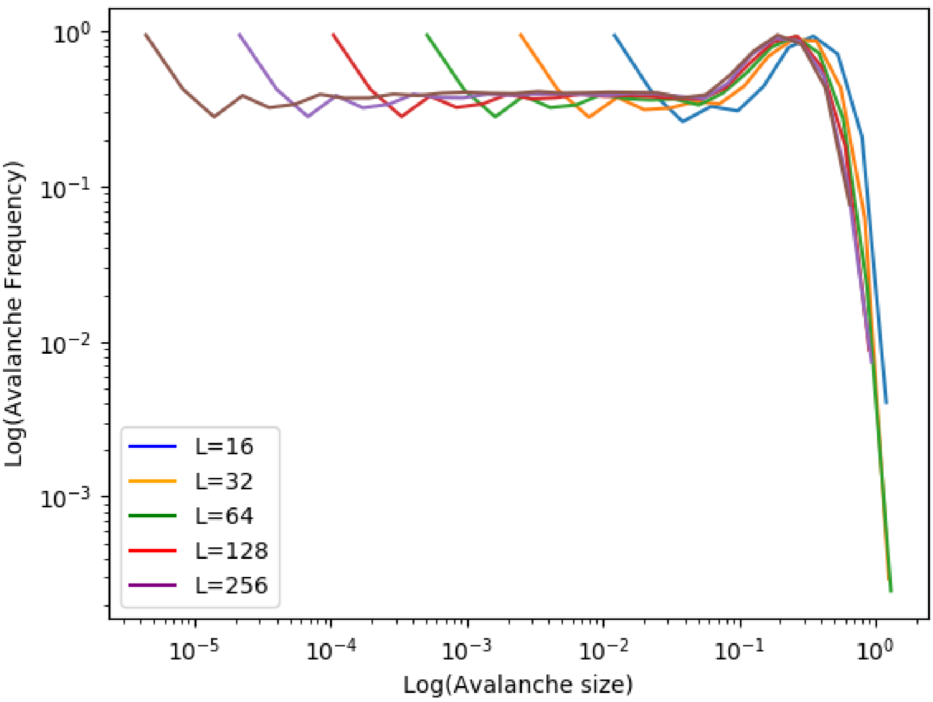
Now we shall test whether the log_binned data P˜*(s;L)* is consistent with the finite_size scaling ansatz: $\widetilde{P}(s;L) \propto s^{-\tau_s} G\left(\frac{s}{L^D}\right)$ .First we approximate τ_s by measuring the gradient of the log_binned data before the cut off s_L. From my data I predicted $\tau_s = 1.56$. Fig,3.7 is the effect of plotting $s^{\tau_s} \widetilde{P}(s;L)$ *vs.* s . As one can see, the plots collapse vertically to form a flat plateux. This implies that my prediction of τ_s is good.

Fig. 3.7



Now we approximate D to collapse the plot further. Note that the ansatz above implies that $\langle s^k \rangle \propto L^{D(1+k-\tau_s)}$ (Proof of this can be found in *Complexity and Criticality*, K. Christensen & Nicholas R. Moloney, pg 275), more specifically $\langle s \rangle \propto L^{D(2-\tau_s)}$. Using the intuition I explained in **2b)**, I derive that $D(2-\tau_s)= 1$. Using $\tau_s = 1.56$, I approximate $D=2.27$. Fig3.8 shows the plots,. The data has collapsed, but could be improved upon by generating more accurate predictions of τ_s and *D*.

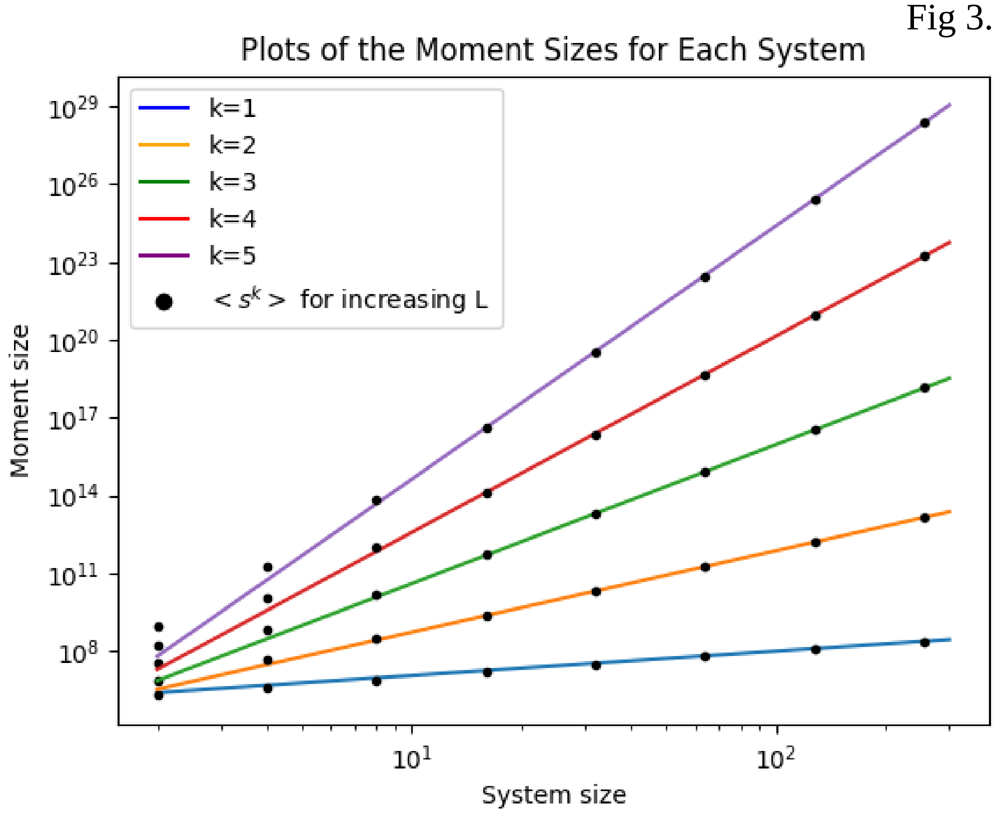
Fig. 3.8



3D

Now we assume that the scaling ansatz is true. If so we would expect the kth moment to scale like: $\langle s\rangle \propto L^{D(1+k-\tau_s)}$. So in a log plot of s, we would expect to see linear relations, with the gradients increasing systematically with moment size. This is verified by by figure 3.9.

Fig 3.9



The plot shows moment sizes, $<s^k>$ for $k=1,\ldots,5$. I have included system sizes 2 and 4 to show that the log-log data is not linear. This implies there is correction to scaling in the data. I took the gradients(α) for large L in the hope of minimising the affects of the correction to scaling.

Fig 3.10

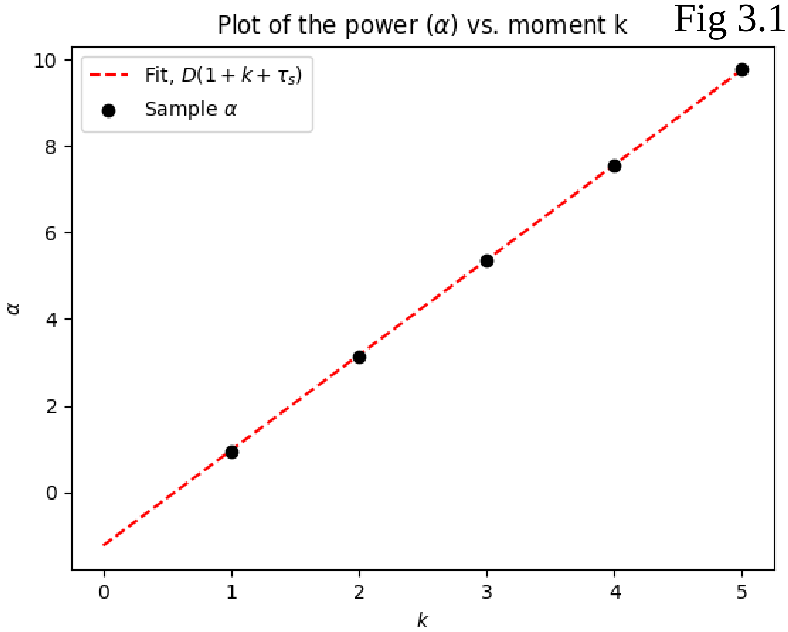
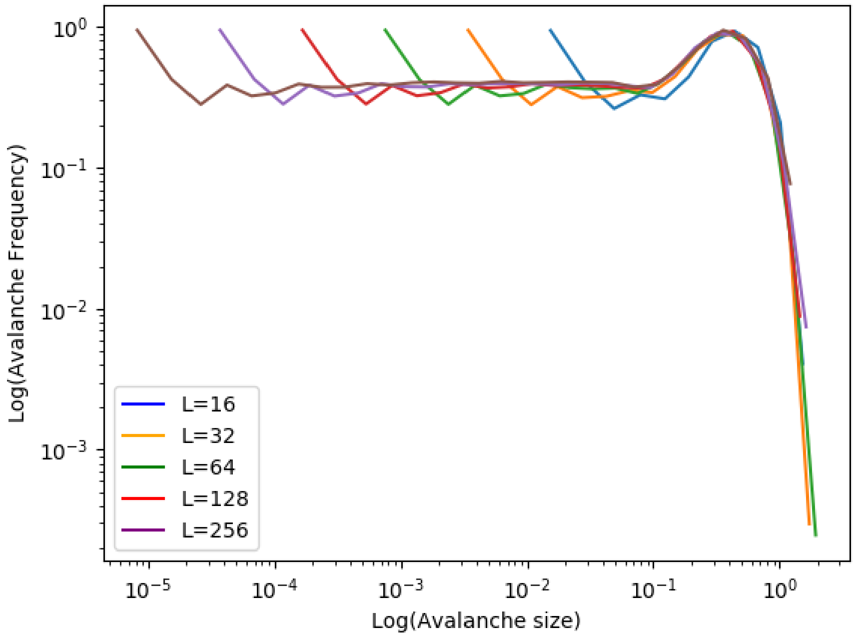


Figure 3.10 shows the α from the previous plots plotted against the moment size k. In theory, the $\alpha=D(1+k-\tau_s)$, *as experimentally* $\langle s\rangle \propto L^{\alpha}$ *but theoretically* $\langle s\rangle \propto L^{D(1+k-\tau_s)}$. Therefore D can be approximated as the gradient of this graph, and $1-\tau_s$ as the x intercept. Using this we derive that

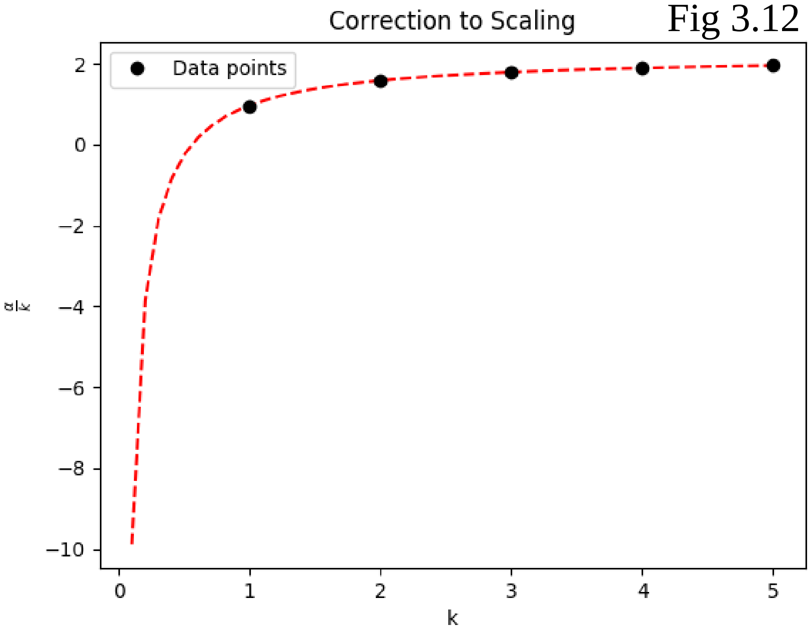
D=2.19 and τ_s=*-1.56*. This is similar to our approximated values for **3c)**. Figure 3.11 is a refit of the data collapse, as one can see the data is more neatly collapsed, implying the new *D* and τ_s are better approximated.

Fig. 3.11



As mentioned before, there is correction to scaling in our data, this is highlighted by Fig 3.10. Like in the cases of the standard deviation and mean explored before, one can clearly see that *α/k* tends towards a constant as *L* grows large.

Fig 3.12



<u>Conclusion</u>

I have explored many of the aspects of the Oslo Model which make it fundamental to the study of self-organising systems. I have shown that the features which define these systems (for example transient and recurrent steps) can be explained physically, and are therefore more than mere mathematical intrigue. This model and models like it can be used to describe complex real world problems.

References

Christensen, K. and Maloney, N. 2005. *Complexity and Criticality*, Imperial College Press, London.