Forest-Fire Model

Denisa Apetroaei

Abstract

Scaling law behaviour can be seen in forest fires, earth quakes, stock market crashes. These can be described by Self Organized Criticality and the purpose of this investigation is to understand what parameters influence them, criticality exponents and how analysis can be improved for such models.
A forest-fire simulation was successfully created, and the behaviour did match the literature, phase transition models fitting the experimental results.

1 Introduction

Self Organised Criticality (SOC) is the property of a dynamical system to, independently of spatial scales, criticaly organize its behaviour. Such systems will build up energy over time, before suddenly releasing it at once.The concept was introduced by P. Bak et al. [1] in 1987. An example of such a system is a forest fire model, where new trees are growing and lightning striking individual trees occurs randomly, with a fixed probability per iteration. This investigation aims to create a forest fire simulation and demonstrate any criticality it may exhibit. Cases of SOC in nature include financial systems, avalanches on snowy terrain, brain activity, and the fires in wooded areas. Inspection into this field could allow for control of such events in nature, preventing them from entering potentially dangerous critical states. An example of such an application is the controlled burn of forest fires.

2 Theory

As previously mentioned, a forest fire model exhibits SOC. Trees slowly growing next to each other, forming increasingly large clusters, are analogous to the system's energy increasing. When lightning strikes and sets a single tree on fire, this energy is release as the clusters of trees burn up, where one tree causes all adjacent trees to catch on fire on the following iteration.
Due to the fact that it takes longer for large tree clusters to randomly form than smaller clusters, the occurrence of large burning tree clusters is considered less likely than the occurrence of smaller burning clusters. For a given burning cluster size, s, the frequency of these events is expected to be proportional to s^{-a} [2], where a is some positive constant.
The above property is said to be scale invariant. Above a certain area, for a large forest, the probability for events of a certain size will stay the same.

2.1 Sandpile Model

Given a finite grid, sand particles constantly fall on it at a given rate. Due to friction between particles, the system takes some time to automatically equilibrate. Immediately after a grain is added, everything stays in the same place. A certain height is necessary for the system to adjust and sand to fall to the grid cells around it in space. This creates an avalanche. This states represents the criticality, when the dynamics can not be described by local laws [3].
For a bi-dimensional system, if $z(x, y)$ represents the height, adding a grain:

$$z(x, y) \mapsto z(x, y) + 1 \tag{1}$$

now if the new height reaches the maximum point, the criticality, it will spread ,as an example, in 4 directions in the grid elements around it so it becomes:

$$z(x, y) \mapsto z(x, y) - 4 \tag{2}$$

The total number of grains that can fall around the central point represents s, the magnitude of the event.
In analogy to the forest-fire, s represents the cluster size, the maximum number of trees that can burn if a fire in the group would be lit.

2.2 Previous Models

One of the best known models is the Drossel-Schwabl-Model [2].It considers a forest with density ρ covered in live trees and empty slots. There is a factor θ such that $\theta = \frac{p}{f}$, p being the probability trees grow and f being trees would be hit by lightning. When trees are hit by lightning they are considered to burn completely until the next step. Any trees from the same cluster will be affected by the fire as well.
For p tending to 0, the lightning is modelled as a Poisson process, with $\theta 1$. The critical exponents of the model can be written as:

$$R(s) \sim s^{\frac{1}{\mu}} \tag{3}$$

where $= d$ and 1_{max} with s_{max} an effective cut-off. Distribution of cluster size can be written as:

$$N(s) \sim s^{-\alpha} \tag{4}$$

which indicates clearly how the probability of events with big magnitude decreases linearly.

3 Current Implementation

The code written in Python follows the rules described by the Drossel-Schwabl-Model [2] with the difference of defining independently the probabilities p and f, of growing trees and burning them, unlike defining the factor $\theta = \frac{p}{f}$.
A class is created for the object fire that updates the whole matrix synchronously after every burn considering vecinity to burning trees.

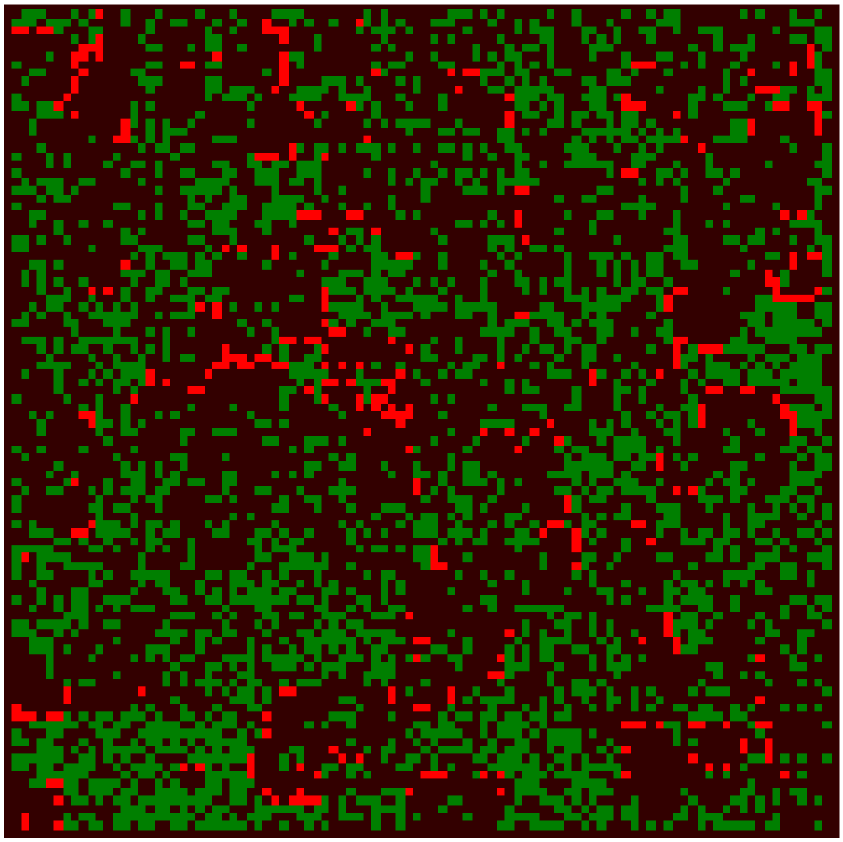

Figure 1: Image showing the chaotic behaviour of the forest-fire in the form of a fractal;with dimensions high enough the fire would show spirals and therefore self-similarity

In order to do the analysis, the Scipy.ndimage library was used. It labels all the clusters, with the structure of matrix given, showing all eight trees around a given one as part of a cluster. This way all the clusters with the same size can be counted.
In order to calculate the radius, the centre of mass function of this library is used. By getting all the coordinates of the trees in the cluster, the biggest distance is searched and that is considered to be the maximum radius a fire can go to from a given tree with coordinates (x,y).
Multiple methods were tried,such as calculating the radius as an average of all the distances from the centre of mass to each point that is part of the group of trees. This misbehaved, and the plot s vs R would give non numerical points or

infinite values.

4 Results

While generating every new matrix after a tree burns or a new tree grows, the clusters get labeled again,counted and measured. This data is saved and then plotted. Fig.3 show the evolution of the numbers of alive trees,burning trees and empty slots in time. The system reaches some stability after the first approximately 200 steps. Phase-shifts can be observed in this plot, the system dynamically oscillating between periods of higher number of trees/higher number of burnings and intervals of higher number of blank slots.

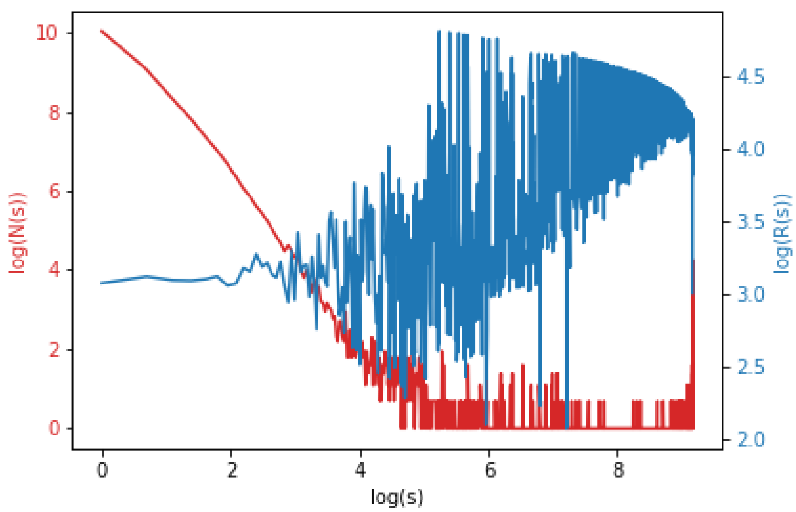

Figure 2: Average Number of clusters and average radius as a function of cluster sizes

Fig.4 shows a linear decrease in the logarithm of number of clusters of size s and logarithm of size, with a critical exponent between -1 and -2, just as expected.

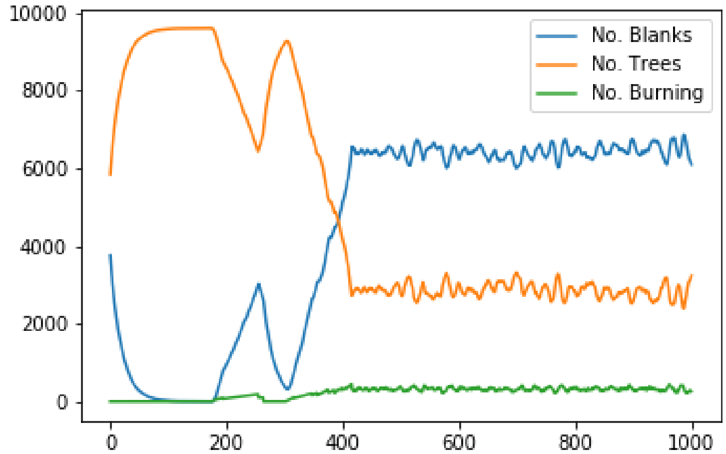

Figure 3: Evolution of number of trees, fires and empty slots as a function of time

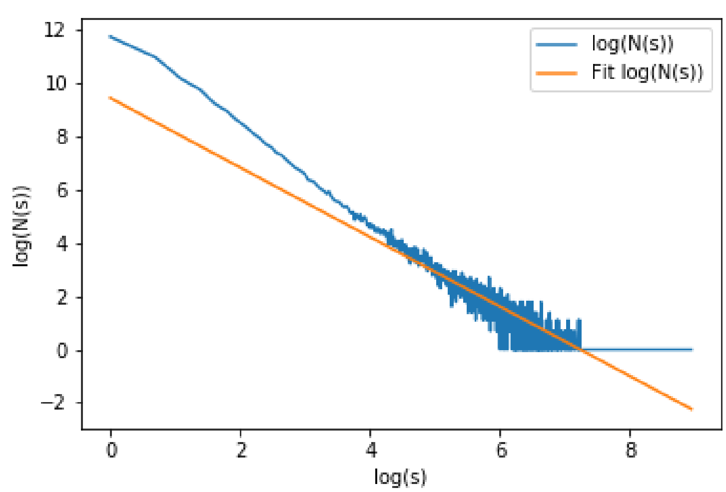

Figure 4: Linear fit of log(N) as a function of magnitude of events, where N is the number of clusters with magnitude s

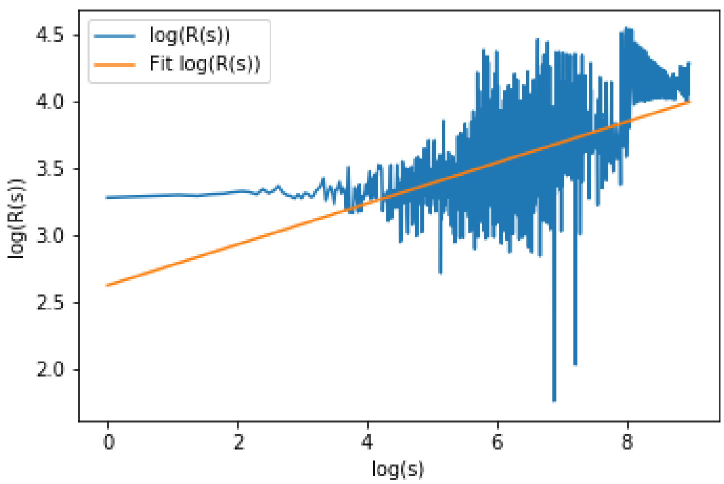

Figure 5: Linear fit of log(R) as a function of magnitude of events, where R is the average radius of clusters with magnitude s

Fig.5 shows a linear increase in the logarithm of radius of clusters of size s and logarithm of size, with a critical exponent between 0 and 1, just as expected. The slope is very close to one for log(s) in between 0 and 4, after which it increases,but it also becomes very noisy which makes it hard to fit properly.

The errors for the number of clusters with size s calculated with the Scipy library show values of 0.01817891 for the gradient and 0.11331952 for the constant, if the function would be modelled as $mx + c$.
For the Radius of clusters with size s as a function of size the values of 0.00683312 for the slope and 0.04259471 for the constant are obtained.
Even if the values are not bad, both functions can be fit better if values for s bigger than 4 would be discarded. The area after that is very noisy and it doesn't allow for a good fit.

4.1 Challenges

The tests ran were done on a grid of 100x100. The computing time for a 1000 steps would be very high, approximately 5 minutes, most of it because of the generation of the new matrix by iteration.
Given the SOC nature of the system, the dimension of the grid should not affect the results after the critical point, which is where most of the noise would manifest as well. However, with a higher number of steps the statistics becomes better as well. For 10 or 100 steps the function would not fit the log(N) vs log(s) at all sometimes, or the fit would be very poor.

The calculation of the radius excluded the situations when a cluster would be formed only by 1 element. This would be seen as a distance of 0 and as a value of infinity when calculating the logarithm for the whole set of data.

5 Conclusion

Further investigation with the simulation constructed in this paper might include assessing the most effective but minimal controlled burn of a forest in order to prevent it from reaching criticality, lessening the risk of a large scale forest fire.
Implementation of better searches through the matrix, such as radius calculation and cluster sizes can be done using graph theory instead of in-built python libraries in order to reduce the time of execution.This would allow for better analysis which would lead in itself to a better understanding of the phenomena.
The results did match the expected behaviour, but the radius measurements gave bigger errors than what would be accepted as a good simulation.
The plots are good for values of log(s) less than 4. For the bigger values analysis on the noise should be done in further work.

References

[1] P. Grassberger and H. Kantz, "On a forest fire model with supposed self-organized criticality," *Journal of Statistical Physics*, vol. 63, no. 3, p. :685–700, 1991.

[2] B. Drossel and F. Schwabl, "Self-organized critical forest-fire model," *Phys.Rev. Lett.*, vol. 69, no. 11, p. 1629–1632, 1992.

[3] B. D. W.K. Mossner and F. Schwabl, "Computer simulations of the forest-fire model," *Physica A: Statistical Mechanics and its Applications*, vol. 190, no. 3-4, p. 205–217, 1992.