SELF-ORGANIZED CRITICALITY BY EXAMPLE OF A FOREST FIRE MODEL

NISHANT DOGRA, MICHAEL WILD

Abstract. In the scope of the course “Modelling and Simulating Social Systems with Matlab” at ETH, we set out to find if a traditional implementation of the forest fire model shows typical SOC-behavior or not. We test out different implementations of the same problem in a more realistic way and analize the results obtained from the simulations.

Part 1. Introduction

1. What is Self-Organized Criticality?

Self-Organized Criticality (SOC) is the idea that the dynamics of a physical system tend to converge on a critical point which commonly is not the traditional equilibrium state, independant of the initial conditions imposed upon. At this critical point, a small disturbance can propagate through the whole system. The concept is best understood by example, so we want to bring up the very educative, although 1D, example first proposed by Bak et al.

Imagine a line which is discretisized into n intervals. We shall denote them with $z_1, \ldots, z_n$. There are n rules in this example:

- Every interval or cell has a number of particles in it, which is directly proportional to the height $h(z_i)$.
- If the difference $\Delta h = h(z_i) - h(z_{i+1})$ between two neighbouring cells exceeds a predefined value d, the cell containing more particles will drop particles onto the other until the difference is less than this value.
- Each timestep, a particle is dropped on z_1.
- We consider closed boundary conditions, meaning that z_1 and z_n may contain any number of particles as long as they comply with the rules of height difference in respect to z_2 and z_{n-1}.

Now an equilibrium state in the traditional, physical sense to this example would of course be a flat surface. But what happens in the simulation, and what is easy to see, is, that a well defined slope is starting to form. This slope will steepen until it reaches a certain point, the critical point, to be specific. At this point, the height difference between any two neighbouring cells will be exactly d. Now what will happen if we drop the next particle on z_1? Of course, $\Delta h_{12} > d$, so a particle will drop to z_2. This leads to $\Delta h_{23} > d$. The process repeats itself until the particle reaches z_n, where the propagation stops because of the proposed boundary conditions.

This example neatly shows how the system, at the critical point, will allow a yet so small disturbance to propagate through the whole system. It is also easy to see that the system, with any initial conditions, will alway converge to the critical point.

Now this is essentially a 1D-Model. It is not easily possible to transform the results on higher dimension problems, but the basic idea should be explained.

Another key idea of the SOC-idea is the stability of the state in respect to disturbances and boundary conditions. This will cause some headaches when the idea is applied to a forest fire model, but will be explained later.

2. What is a Forest Fire Model?

A Forest Fire Model (FFM) is basically a cellular automata with 3 states and 4 rules. The cell states are:

- 0: Empty
- 1: Alive
- 2: Burning

The 4 rules are defined as follows:

- An empty site will turn into an alive site with probability p.
- An alive site will turn into a burning site with probabiltiy f.
- If any neighbor of an alive site is burning, so will this site.
- A burning site becomes an empty site in the next timestep.

There are several ways to implement the aforementioned rules. This can have a large effect on the results of the simulation and will be treated later. It is, however, important to note that for a realistic model, $\frac{f}{p} \ll 1$ and therefore $f \ll p$ should be valid.

It is also important to note that an FFM is not only related to Forest fires but has applications that go well beyond that. An example would be the spreading pattern of a desease, which follows the same, basic rules.

3. Power laws

Power laws are functions of the form $P(s) \propto s^{-\tau}$. These functions are found in many data sets, such as the size distribution of cities in a certain area. Take any Country, one will find one very large metropolis, a few large cities, many medium sized towns and a huge lot of small towns. Power laws imply that the frequency of a general data point is inversely proportional to its magnitude.

In context to the FFM, it is proposed as a way of quantifying the frequency of Fires with respect to their cluster size.

Power laws have also prominently been found in data sets of earthquakes, where they apply almost perfectly, meaning that small earthquakes happen very often in respect to the frequency of big ones.

4. Key Parameters in SOC

To find out wether a basic FFM exhibits SOC behavior, we need to define a set of parameters which serves as indicators for the desired analysis. A first indicator is the cluster size distribution. To understand the meaning of the cluster size, we need to have a short look at network theory. Consider an instantaneous fire which burns every tree directly or indirectly connected to the ignitor. One can look at this network as a graph of nodes which are connected if they are neighbors on the grid. Now we define the cluster size as the network diameter, meaning the longest shortest connection between any two nodes. Since this network illustation is well connected to the grid reality, we can take that as a good indicator for the distance the fire can travel on the grid. This is equivalent to the criterium for SOC that was mentioned before, which stated that a disturbance can travel through the whole system. We shall denote the cluster size with the parameter D.

Part 2. Modelling the problem

5. Cellular automata

A cellular automata is a very powerful way of simulating problems which are defined by a set of rules. It is usually simulated on a grid, but not necessarily restricted to those geometrical constraints. The main characteristics are:

- Every grid point has a state
- Grid points change their state depending on the neighbor states according to a set of rules
- Random actions may be introduced

In the example of the FFM, every grid point has three states (empty, alive and burning) and four rules, the most obvious of which is “change state to burning if you are alive and your neighbor is burning”.

Cellular Automata can exhibit very complicated behavior when fed with very simple rules, which is the main reason why they are so powerful. There are similarities to agent-based models and networks, but it is not to be mistaken for one of these.