Topic 5: Interdisciplinary Problems and Python Scripting

Lecture 5-3: Sandpile Automaton in 2-D

Monday, April 5, 2010

Contents

1 Two dimensional sandpile automaton **1**

1.1 Model and local update rule 2

1.2 A typical steady state 4

1.3 Self-organized criticality 4

1.4 C++ OpenGL program 5

1 Two dimensional sandpile automaton

The one-dimensional sandpile automaton has a very simple and rather uninteresting steady state behavior: the pile assumes the critical slope and any grain added to the pile slides off the edge without disturbing the pile.

Bak, Tang and Wiesenfeld [Bak-1987] introduced a very simple two-dimensional sandpile automaton model which has complex and interesting behavior.

A realistic two-dimensional sandpile model would have a two-dimensional array of columns of sand grains. At each column, one would define a *vector* slope with components in the x and y directions.

1.1 Model and local update rule

The Bak-Tang-Wiesenfeld model has a single *scalar* slope, which can take 8 discrete values

$$s_{i,j} = 0, 1, 2, 3, 4, 5, 6, 7,$$

at each column position i, j. The column i, j is *unstable* if $s_{i,j} > 3$. The sandpile is unstable if any column is unstable. The sandpile is updated with the following local rule:

- if $s_{i,j} > 3$, then remove 4 units of slope from column i, j and add one unit of slope to each of the 4 neighboring columns.

- This rule is applied synchronously to all columns in the pile, and repeated until a steady state is reached.

To make the model well defined, *boundary conditions* need to be specified at the edges of of the sandpile. One simple possibility is to use *open* boundary conditions: if a grain (strictly a unit of slope) falls off the edge of the pile, it disappears. Other possibilites are: *periodic* boundary conditions—if a grain falls off one end of the pile it appears at the opposite end; and *flow* boundary conditions—grains are injected into the pile at each time step at one or more boundary points.

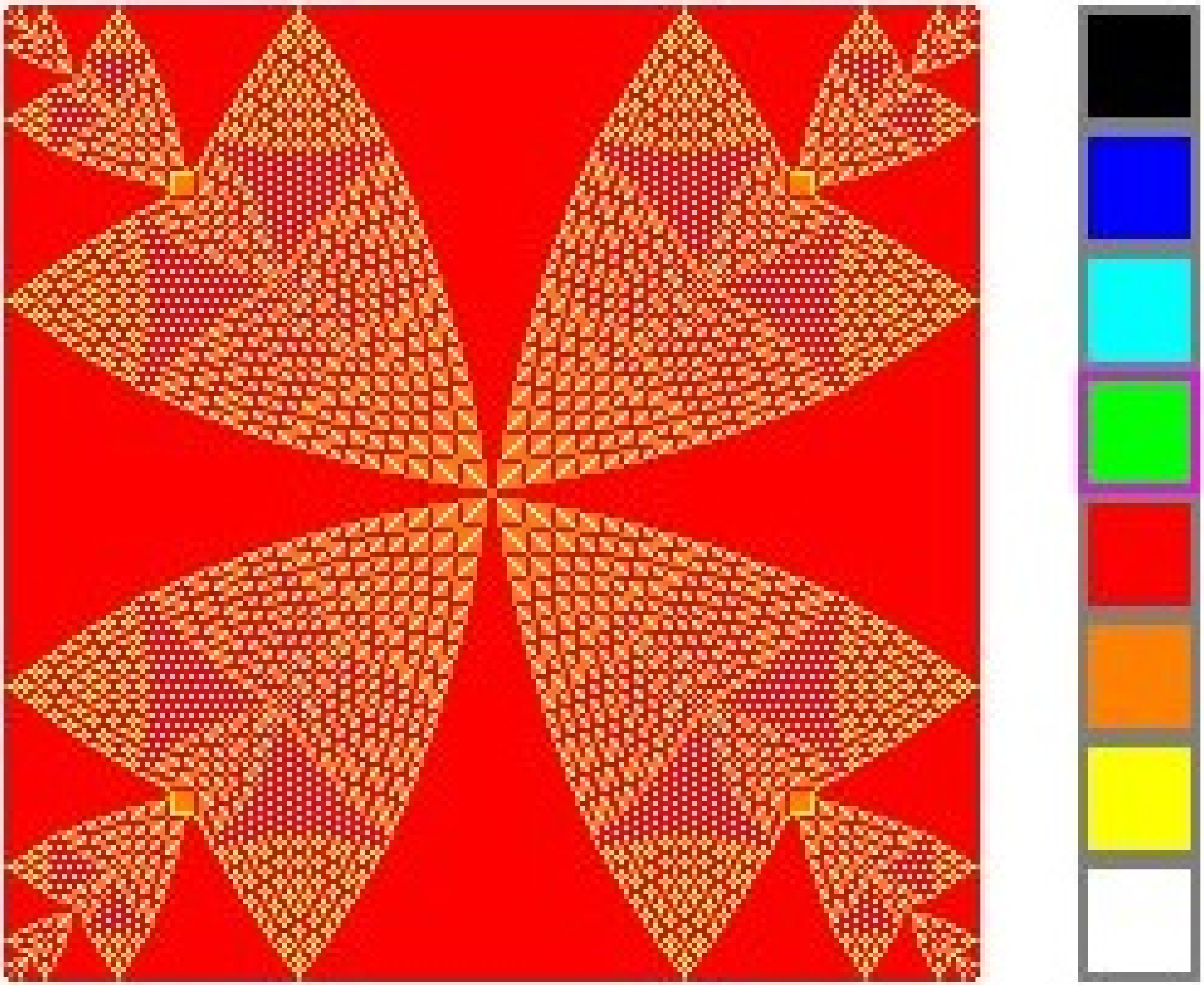
1.2 A typical steady state

The figure shows the steady state of a pile with 200×200 columns starting from an initial unstable state with $s_{i,j} = 7$.

1.3 Self-organized criticality

From the figure, it is clear that a steady state of this model can be extremely complex and intricate. The state is *critical*, which means that if a single grain of sand is added to it, an avalanche of toppling columns is triggered, and the system settles down to a different critical state. These events are so complicated that it is impossible to predict what will happen without actually running the simulation.

Let n_t be the number of topplings in an avalanche. By generating a large number of avalanches, one can



measure the distribution of numbers $N(n_t)$ of avalanches as a function of avalanche size n_t. For the sandpile model, the distribution has a *power law* behavior:

$$N(n_t) \sim \frac{1}{n_t^b} \; ,$$

where b is the *exponent* of the power law. Power-law behavior implies that the avalanches have no natural scale or size: events of all sizes can occur in the distribution. If the exponent b is very small, then events of all sizes are equally probable. If $b > 0$ then larger events are less likely than smaller events. The special characteristic of a power law is that the *ratio* of events which differ in magnitude by a fixed multiplicative factor is *independent* of the size of the events. If the factor is 10, for example, then

$$\frac{N(10n_t)}{N(n_t)} = \frac{1}{10^b} \; .$$

For the sandpile models the exponent is found to be of order one: $b \simeq 1$.

Bak, Tang and Wiesenfeld called this behavior *Self-Organized Criticality*. The sandpile evolves or *organizes itself* without any external influence into a complex *critical* state where events of all sizes can occur.

1.4 C++ OpenGL program

The code sand2.cpp implements the 2-D sandpile automaton model.

References

[Bak-1987] P. Bak, C. Tang and K. Wiesenfeld, "Self-organized criticality: An explanation of the $1/f$ noise", Phys. Rev. Lett. **59**, 381 (1987), `http://doi.org/10.1103/PhysRevLett.59.381`.

[W-CA] Wikipedia: Cellular automaton, `http://en.wikipedia.org/wiki/Cellular_automaton`.