Self-organized criticality in cellular automata for sandpiles

M. Frohne and P. Wolf
Helmholtz-Institut für Strahlen- und Kernphysik, Universität Bonn, Nussallee 14-16, D-53115 Bonn

(Dated: April 3, 2018)

In this paper the dynamical behavior of systems in states of self-organized criticality is studied using the example of sandpiles in two and three dimensions. Two distinct models inheriting different characteristics are implemented, one of them being the Bak-Tang-Wiesenfeld model, the other an independent approach. The scaling exponents of the three observables *size, duration* and *area* of avalanches are determined for both models. This is achieved by conducting an analysis of their moments in order to extract the scaling exponents as well as their uncertainties. Apart from slight deviations in the custom model, due to its different nature, the results stated in several papers by different authors could be reproduced in a satisfying manner.

I. INTRODUCTION

Self-organized criticality describes a special case of the very general concept of criticality. Every system exhibiting self-organized criticality has in common the occurrence of macroscopic scale invariant properties. In the picture of a real sandpile this can be seen by looking at the average slope of the sandpile, which stays constant even if the sandpile is enlarged by adding more and more sand to it. It is notable that this is the case despite the dynamics of a pile of sand are actually governed by complicated microscopic[1] interactions of its many constituents, the sand grains.

Such self-organized criticality has been observed before in simulations of sandpiles carried out by Bak, Tang and Wiesenfeld in the 1980s. It is indeed interesting to further analyze such a sandpile model because it can serve as a toy model for many other similar behaving systems such as landslides, rock falls, earth quakes etc., which all follow from different physical processes, though. In the mentioned systems signs of self-organized criticality have been observed, not only from simulations but also from real-world data [2].

We simulate the sandpiles using a cellular automaton algorithm analog to the approach of Bak, Tang and Wiesenfeld, which is commonly referred to as the "BTW model". Scaling exponents are determined and compared to the results of a customized sandpile model.

Section II starts with an overview on the theory behind self-organized criticality and in particular explains the scaling behavior governed by the scaling exponents. In Section III our implementation of the sandpile cellular automata is shown as well as the used methods for extracting the relevant parameters from the generated data. The results are presented in Section IV and discussed in Section V.

II. THEORY

A. Self-organized criticality

To understand the concept of self-organized criticality consider first a system that exhibits a critical state, like the Ising model. The critical state in the Ising model corresponds to a phase transition where vanishing magnetization evolves into spontaneous magnetization. This happens at a specific temperature, the critical temperature T_{crit}. To obtain a system in the critical state, the temperature has to be precisely tuned to the critical temperature.

The criticality of the system leads to scale invariant properties. For instance clusters of the same spin direction will form and the size distribution of these clusters follows a simple power law. A power law distribution implies scale invariance, where scale invariance means that the dynamics of the considered system do not change if the system is scaled by some global factor. In particular the observables' distribution functions keep the same form.

The crucial difference of this definition of criticality and the criticality observed in the case of the sandpile model is that sandpiles will evolve into a critical state without tuning of any critical parameter. If enough sand is deposited into a sandbox or, respectively, enough iteration steps of the computer model are performed it reaches an equilibrium state by itself with a fixed average slope. This self-organization into a critical state always happens, independently of the actual detailed model parameters[3]. Thus this phenomenon is called "self-organized criticality".

B. Scaling exponents

If there is scale invariance in a system it can be directly observed in the distribution functions of the observables of the system as indicated above. Consider some observable $\hat{Y}$ that can be measured in the present system. If this observable is measured many times and histogrammed one obtains a distribution according to the corresponding probability density function (PDF) $P^Y(y)$. If now the system is scaled by a factor λ it must hold

$$P^Y(\lambda y) = f(\lambda) \times P^Y(y) \tag{1}$$

due to the assumed scale invariance.

As scale invariance is suspected as a main property of SOC models in general, one is mainly interested in studying these distribution functions to test this hypothesis. To do this a theoretical prediction is necessary first. In the simplest approximation the distribution function of an observable $\hat{Y}$ in a scale invariant SOC model is assumed to follow a simple power law

$$P^Y(y) \sim y^{-\rho} \tag{2}$$

in order to fulfill Equation (1) as $(\lambda y)^{-\rho} \sim y^{-\rho}$. Here ρ is called the *critical exponent* of the observable $\hat{Y}$.

However, in a real implementation of the sandpile model or any other SOC model one has to deal with finite system sizes since infinitely large systems can neither be simulated nor do they exist in nature. A finite system size limits the validity of Equation (2) because for instance avalanche sizes on a sandpile are naturally limited to the size of the sandpile itself. Thus a more sophisticated description of scaling has to be used, so-called *finite-size scaling* as described in [4]:

$$P^Y(y) = a y^{-\rho} G^Y\left({}^{y}\!/_{y_c(L)}\right), \quad y_c(L) = bL^K \tag{3}$$

The additional scaling function G^Y accounts for limited size corrections to the simple power law scaling by modifying the probability density for values of y in the order of the limiting *characteristic scale* $y_c(L)$. This characteristic scale where limitations eventuate can itself depend on the system size L, e.g. the maximum area coverage of a two dimensional lattice is L^2. Thus the characteristic scale is approximately assumed to behave as $y_c(L) = bL^K$ for any observable, where b, K depend on $\hat{Y}$. The set of the critical exponent ρ together with the "observable's dimension" K are generally called finite-size scaling exponents or short scaling exponents.

One possibility to obtain the finite-size scaling exponents ρ and K from the measured distribution functions is via their moments $\langle y^n \rangle$. These moments follow from Equation (3) by integration and as is shown in [4] in the limit of an infinitely large lattice this yields

$$\langle y^n \rangle(L) \xrightarrow[L\to\infty]{} a g^Y(0)(bL)^{\sigma_n}, \quad \sigma_n := K(1+n-\rho), \tag{4}$$

with $g^Y(0) = \text{const.}$ a definite integral containing the scaling function G^Y. Equation (4) is as an approximation also valid for finite but large enough lattice sizes where the observable's characteristic scale is much larger than a certain lower cutoff parameter. This lower cutoff may arise from the detailed microscopic interactions and the discreteness of the lattice, i.e. $L \gg 1$ for a lattice divided into discrete cells of edge length 1 can in principle yield sufficiently good results.

Using Equation (4) the scaling exponents could now be calculated from any measured distribution function by doing simple linear regressions to log-log plots of different moments $\langle y^n \rangle(L)$ against L. Combining the parameters σ_n from the regression for different n provides ρ and K.

III. EXPERIMENTAL METHODS

A. Cellular Automata

To realize the simulation of sandpiles it is convenient to construct a cellular automaton. A cellular automaton is an algorithm that takes values on a discrete n-dimensional lattice and repeatedly performs updates on all lattice sites with each updated value only depending on the old value and the values of its nearest neighbors (and possibly random numbers). Additionally the lattice sites can be periodically perturbed, for instance increasing the value of a randomly picked lattice site.

B. Implementation of sandpile cellular automata

In general a cellular automaton for a sandpile in $\mathcal{N}$ real dimensions works on an $\mathcal{N} - 1$ dimensional lattice where each lattice site stands for a stack of sand grains in the $\mathcal{N}$-th dimension. That means for example that a real sandpile, which is three dimensional, is simulated on a two dimensional lattice reflecting the ground as x-y plane. The entry on each site then characterizes how many grains of sand are stacked in the z direction. We will denote these lattice entries by the variable $s(\vec{r})$ in the following, with the discrete lattice sites $\vec{r} = (x, y, \ldots)^\top$.

It can be either chosen to store the actual amount of grains as the entry s or the slope with respect to the vicinity of the stack. Both choices enable for sufficient characterization and simulation of the microscopic dynamics of the sandpile. Choosing the slope entries, however, has the drawback that the model will then be non-isotropic anymore, as will become clear in the following, whereas storing the actual heights can make the model implementation a bit more complex.

Within the scope of this paper we have implemented two different cellular automata that are based on these two different choices.

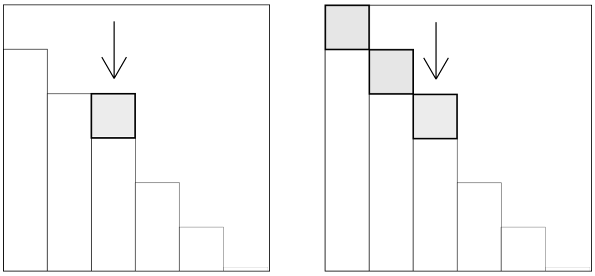
1. The BTW model

The first approach reflects the BTW model. The BTW model by Bak, Tang and Wiesenfeld [5] is the first model investigating SOC in sandpile dynamics. It uses a lattice that contains slope values and alternatingly performs a *driving* and a *relaxation* of the lattice:

Driving generally describes the active perturbation of the lattice. For a real sandpile the perturbation intuitively consists of adding a grain of sand to the pile at a random position. Since the BTW model deals with slopes the addition of one grain of sand could not be achieved by just increasing $s(\vec{r})$ about one but the neighboring entries would have to be changed, too. This type of driving is called *conservative driving* (see Figure 1a). For the BTW model, however, we decided to use *non-conservative driving* described by simply adding one slope unit to the considered lattice site:

- $s(\vec{r}) \mapsto s(\vec{r}) + 1$

The non-conservative driving can be thought of as adding grains of sand to *all* uphill sites with respect to $\vec{r}$ (Figure 1b), which is of course unphysical but converges much faster to the SOC state [6]. It can still reproduce results similar to the conservative algorithm when the same, physical, relaxation algorithm is used.



(a) Conservative driving. (b) Non-conservative driving.

Figure 1: Driving of a one dimensional sandpile where the columns in the sketch display the *actual pile heights*.

Relaxation takes place after a driving step has been performed. If the sandpile becomes too steep somewhere on the lattice, i.e. $s \geq s_{\text{crit}}$ with the *critical slope* s_{crit}, after sand has been added the pile gets instable. It locally collapses and thus creates avalanches of sand sliding down the pile until the slope recovers to a non-critical value again. In the BTW model the critical slope is usually fixed to $s_{\text{crit}} = 2 \times \text{Dim}\,[\vec{r}]$. For each site with critical slope the following relaxation algorithm is performed:

- $s(\vec{r}) \mapsto s(\vec{r}) - 2 \times \text{Dim}\,[\vec{r}]$
- $s(\vec{r} \pm \vec{\text{e}}_i) \mapsto s(\vec{r} \pm \vec{\text{e}}_i) + 1, \quad \forall i \in [1, \text{Dim}\,[\vec{r}]]$

If after the relaxation any site is still critical or if any other site has become critical in the process of relaxation the relaxation procedure is repeated until there is no critical site on the lattice anymore. Every repetition of the relaxation procedure counts as one unit time step.

To translate this relaxation routine to the picture of a real sandpile it calculates the local slope of the sandpile as the sum of the local slope (= stack height difference) in x-direction and the local slope in y-direction. If this summed slope reaches the critical slope value s_{crit}, for each dimension one grain of sand is redistributed from the site to its downhill nearest neighbors. This results in a consistent description of a lattice containing slopes and the two procedures for driving and relaxation.

The fact that the slopes s are only integer numbers naturally implies that they cannot contain any information about the direction of the slope. If one takes the real sandpile picture from above into account again it becomes clear that the slope direction is fixed to the same diagonal for every lattice point. This is an inherent feature of the BTW model as a result of the reduction to storing slopes instead of stack heights. It creates a preferred direction for avalanches and thus makes the model *non-isotropic* such that the BTW critical state must be principally thought of as a “diagonal hillside”.

2. The custom model

The second approach for the sandpile simulations uses the local height or number of sand grains as lattice entries. Thus it can circumvent the problem of the BTW model to be non-isotropic because the slopes can be directly calculated from the heights in every direction. The overall implementation of the custom model cellular automaton essentially follows the implementation of BTW model. Since in the custom model the lattice entries $s(\vec{r})$ are heights instead of slopes the relaxation procedure works differently compared to the BTW model, though. For the slope between two neighboring sites $\vec{r}_1$ and $\vec{r}_2$ the difference $s(\vec{r}_1) - s(\vec{r}_2)$ is used. If the absolute value of this slope reaches the critical value q_{crit} the site with the higher stack is called critical site. The critical slope q_{crit} is a free parameter in this custom model.

If now one site is critical with respect to at least one of its nearest neighbors the stack is considered unstable and about to collapse. All excess sand grains with respect to the neighbor with the least height will then be redistributed to all nearest neighbors to which the slope is at least critical until none of the slopes to the neighbors is critical anymore. This represents one unit time step. Since grains have been redistributed and slopes could have changed, in the next relaxation step all nearest neighbors of the critical site are checked if they have become critical. Like in the BTW model this is repeated until no site is critical anymore and the avalanche stops.

The driving procedure simply consists of increasing $s(\vec{r})$ about 1.

In the custom model it is possible to choose different boundary conditions for each edge of the lattice independently. It can be chosen between *open* and *closed* boundary conditions. The closed boundaries basically model an infinitely high wall that does not let any sand grains pass whereas the open boundaries can be thought of as the edge of a table where any excess sand can just drop from the table. This means that sand grains can fall of the lattice at open boundaries but accumulate at closed boundaries. In order to make the comparison of the BTW model and the custom model possible the configuration of Dim $[\vec{r}]$ neighboring open boundaries meeting at one corner of the lattice and respectively Dim $[\vec{r}]$ neighboring closed boundaries was chosen for the simulations, which yields a “diagonal hillside” for the custom model, too.

C. Extraction of scaling exponents

Since we want to study the SOC scaling behavior of sandpile avalanches we need to extract scaling exponents of different observables. Besides the two most intuitive observables that characterize an avalanche, the *area*, the *time duration* and the *linear size/radius*, there is another quantity that is typically considered in SOC studies, which is slightly misleadingly called *size* of an avalanche. The size quasi describes the overall instability released in the avalanche or also the total mass-, momentum- or energy dissipation of the avalanche. For the automaton algorithms we precisely define all four observables as follows:

Size: The number of critical sites accumulated over all repetitions of the relaxation procedure.

Duration: The number of unit time steps from the triggering of the avalanche until it stops. The unit time step corresponds to one repetition of the relaxation routine.

Area: The Dim $[\vec{r}]$ dimensional volume that contains the avalanche, i.e. the number of lattice sites that took part in the avalanche.

Linear size: The maximum (euclidean) distance between any pair of sites that took part in the avalanche.

The calculation of the linear size gets very computing-intensive and memory consuming for large lattice sizes L and has thus been disabled during our full simulations. In this paper we concentrate on investigating the size, duration and area.

To each of the three considered observables needs to be assigned a set of scaling exponents (ρ, K) according to Equation (3):

Size: critical exponent τ, avalanche dimension D.

Duration: critical exponent α, dynamical exponent Z.

Area: critical exponent κ, avalanche area dimension T.

As pointed out in Section II B the exponents can be determined from the corresponding measured distributions. To obtain these distributions one full simulation run needs to be performed, which alternatingly performs a driving and triggering of a relaxation of the lattice as described in Section III B. This happens until a certain given number of drives is reached. Each relaxation between two drives corresponds to an avalanche[7]. The desired observables are calculated within the relaxation procedure for each avalanche and the values for all avalanches are in the end stored by the main program. The stored data then represent samples of the underlying distribution functions.

Given a sample of one of these distributions, the distribution’s first and second moment is estimated via

$$\langle y^n \rangle(L) := \int \mathrm{d}y\, y^n P^Y(y) \simeq \sum_i y_i^n \Big/ \sum_i y_i^0, \quad n = 1, 2, \tag{5}$$

where L is the lattice size used in the simulation and y_i the measured value of $\hat{Y}$ in the i-th avalanche. According to Section II B we plot $\ln(\langle y^n \rangle(L))$ against $\ln(L)$ using simulations for different values of L and apply a linear regression to the plot. This yields a straight line with slope σ_n, such that ρ, K follow from:

$$\rho = \frac{2\sigma_2 - 3\sigma_1}{\sigma_2 - \sigma_1}, \qquad K = (\sigma_2 - \sigma_1) \tag{6}$$

To enable for estimating uncertainties via bootstrapping, each simulation run must be applied multiple times such that multiple samples of the distribution lead to multiple measurement values of $\ln(\langle y^n \rangle(L))$ for constant L. We use a double bootstrap procedure in order to do a complete linear regression including the covariance matrix of the measurement points $\langle \ln(\langle y^n \rangle(L_i)), \ln(\langle y^n \rangle(L_j)) \rangle$.

IV. SIMULATION RESULTS

To achieve sufficiently good statistics also for the larger, i.e. rarer, avalanches each simulation was set up to do 100000 random drives and was repeated ten times for doing the bootstrap. Simulations were done for both models in two and three dimensions for lattice sizes ranging from 10 to 100.

All the sandpile lattices used for the simulations have been filled and driven to a self-organized critical state beforehand using the same two cellular automata. For this purpose these were kept running until the averaged slope of the piles did not change significantly anymore, measured across a certain amount of drives in between.

1. BTW model

In Figure 2 one exemplary two-dimensional critical sandbox for the BTW model is shown, which represents one of the sandboxes that were used as initial sandbox for the later simulations.

Avalanches of different size and shape occurred in the simulations. Figure 3 shows one example of the avalanches recorded for a two-dimensional sandbox.

In Figure 4 the bootstrapped linear regression of the first and second moment of the avalanche size distribution is shown for a two-dimensional sandbox. The data show only minor deviation from the straight line fit. The fits for avalanche area and duration yield similar results, also for three-dimensional lattices. The resulting scaling exponents for two- and three-dimensional lattices are written down in the summarizing Table I on page 6.

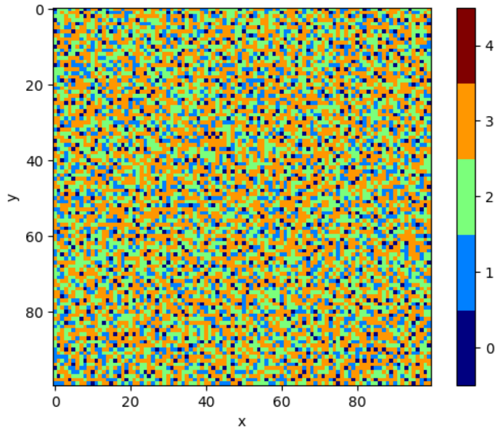


Figure 2: Critical two-dimensional sandbox of the BTW model with lattice size $L = 100$.

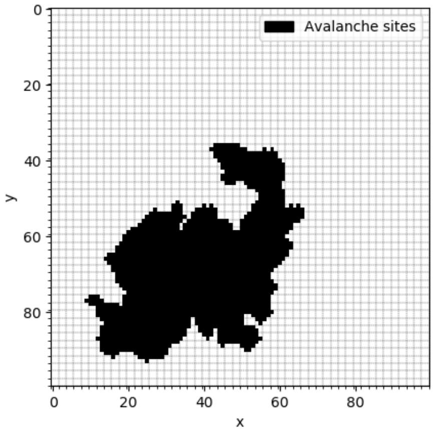


Figure 3: Avalanche of the BTW model in a two-dimensional sandbox with lattice size $L = 100$. Every black entry stands for a site that took part in the avalanche by relaxing at least once during the whole avalanche.

2. *Custom model*

Figure 5 shows a sandbox obtained from the custom model simulations with $q_{\text{crit}} = 7$, open boundary conditions at the bottom and left edge and closed boundary conditions at the top and right edge.

Note again that this sandbox displays heights and not slopes and hence looks completely different than the BTW sandbox in Figure 2. To roughly compare both sandboxes one can have a look at Figure 6, which shows how this sandbox would appear in the BTW model, i.e. shows the corresponding BTW-like slopes, which are a superposition of the slopes in x- and y-direction.

The avalanches observed in the custom model simulations are typically smaller and of a more elongated shape compared to the avalanches of the BTW model. Figure 7 shows one example of the avalanches recorded for a two-dimensional sandbox. Traces of the most recent of these avalanches can be even seen in the sandbox itself (Figure 5).

Analog to the case of the BTW model we show the linear fits of the first and second moment of the avalanche size distribution for a two-dimensional sandbox in Figure 8. The straight lines do not fit to the data as good as the ones for the BTW model but for different runs of the bootstrap fitting the slope and fit quality changed noticeable. This seems to be caused by the limited amount

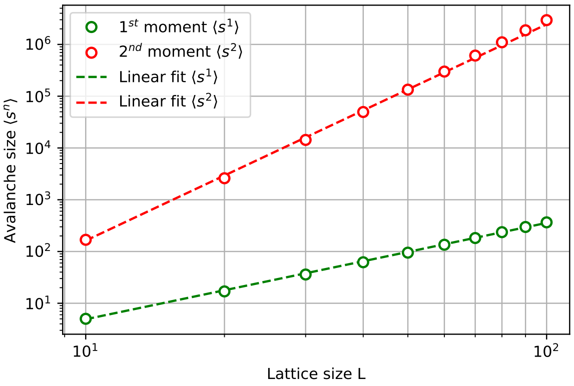


Figure 4: Log-log plot of the first and second moments of the avalanche size distribution of two-dimensional sandboxes (BTW model) against L including bootstrapped straight line fits. The displayed data points are mean values over ten repeated simulations. Error bars are too small to be visible.

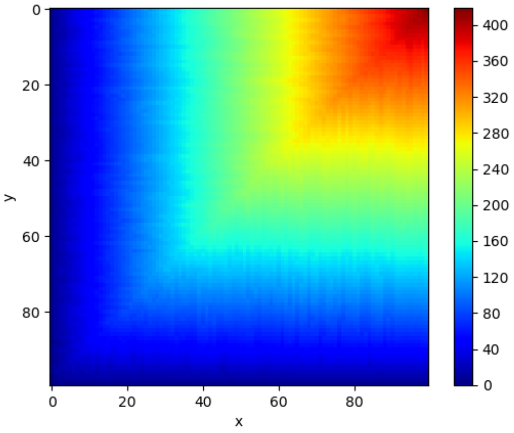


Figure 5: Critical two-dimensional sandbox of the custom model with lattice size $L = 100$ and critical slope $q_{\text{crit}} = 7$. The boundary conditions are open at the bottom and left edge and closed otherwise.

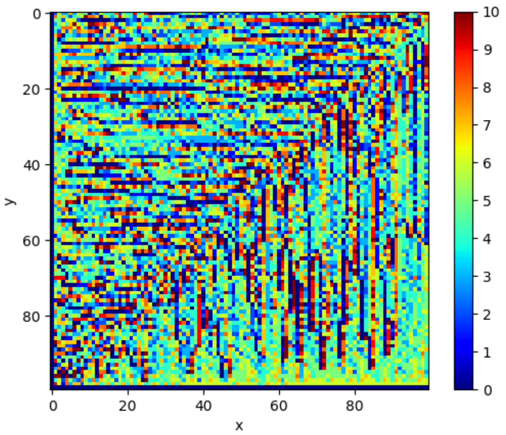


Figure 6: Critical custom model sandbox from Fig. 5 that has been transformed from heights to BTW-compatible summed slopes. Traces of recent avalanches are clearly visible.

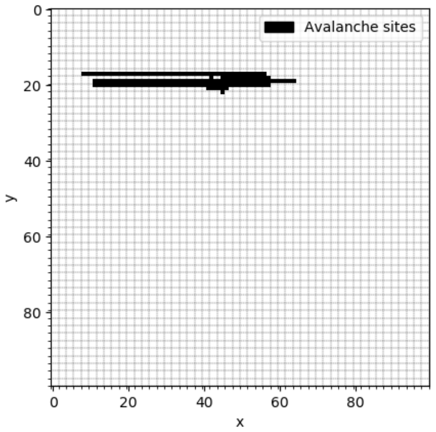


Figure 7: Avalanche of the custom model in a two-dimensional sandbox with lattice size $L = 100$.

of only ten measurements per lattice size L, which might not represent an ideal base sample for the bootstrap. The figure shows the fit run that yielded the smallest error on the resulting exponents.

The resulting scaling exponents for two- and three-dimensional lattices are summarized in Table I together with the results for the BTW model. Results for different values of the critical slope parameter of the custom model are given.

V. DISCUSSION

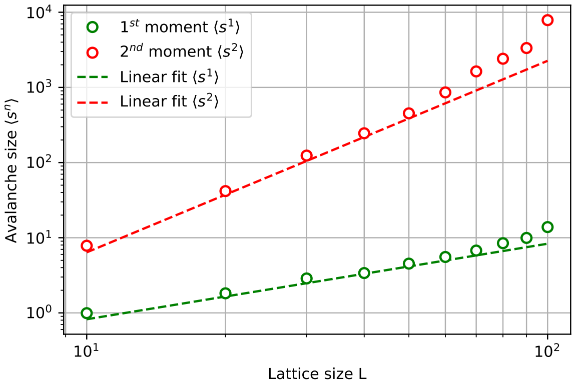


Figure 8: Log-log plot of the first and second moments of the avalanche size distribution of two-dimensional sandboxes (custom model) against L including bootstrapped straight line fits. The displayed data points are mean values over ten repeated simulations. Error bars have been omitted for reasons of clarity.

Table I: Scaling exponents for avalanche size, duration and area for two- and three-dimensional (2D/3D) simulations of the BTW model (BTW) and the custom model (CST). The free parameter critical slope q_{crit} of the custom model is denoted by the index C q_{crit}.

Model	Size		Duration		Area		
	τ	D	α	Z	κ	T	
BTW	1.19(9)	2.3(2)	1.21(2)	1.34(3)	1.23(7)	1.9(2)	2D
CST$_{\text{C5}}$	1.3(2)	1.6(2)	1.2(2)	0.7(2)	1.3(2)	0.8(2)	
CST$_{\text{C7}}$	1.4(2)	1.6(4)	1.3(1)	0.81(8)	1.4(1)	1.0(1)	
BTW	1.3(1)	2.5(3)	1.45(3)	1.41(6)	1.4(3)	2.6(7)	3D
CST$_{\text{C5}}$[a]	1.23(1)	1.99(2)	1.29(1)	1.17(1)	1.29(1)	1.43(1)	

[a] Values only based on simulations for lattice sizes $10, 20, 30, 40$ due to limited computational resources. Hence bootstrapped fitting failed and external fit library was used.

The sandboxes of both models clearly self-organized into some equilibrium state with temporally constant average slope and overall appearance. The extraction of scaling exponents based on the finite-size scaling assumption (Equation (3)) was possible and even naive power law fits to the histogrammed distributions indicate scale invariance (see Appendix 1). This confirms the occurrence of SOC in the BTW model as already previously observed by many other researches and it shows that SOC also appears in our custom sandpile model.

A SOC state seems to develop in the custom model independently of the value of the critical slope parameter. Since we only simulated for $q_{\text{crit}} = 5$ and $q_{\text{crit}} = 7$ we suggest, however, to check this again for a much broader range of critical slopes.

The avalanches look quite different for the different models. The BTW avalanches appear much more spread over the lattice than the custom model ones. As one can see in Figure 5 the choice of boundary conditions in the custom model did not completely yield the desired "diagonal hillside" but rather two split off hillsides towards the x- and y-axes. We believe that a further modification of the custom model, namely to allow for dropping grains to the diagonal nearest neighbors, would yield improved results. This would also describe the physics more accurately.

There were virtually no large avalanches observed for the three-dimensional custom model. This is possibly due to a too small critical slope as we have only simulated for $q_{\text{crit}} = 5$ in three dimensions due to limited computational resources. The critical slope in the BTW model is the fixed parameter of $s_{\text{crit}} = 2 \times \text{Dim}\,[\vec{r}]$ and is thus automatically increased for larger dimensions. Simulations in three dimensions with $q_{\text{crit}} \gg 5$ are advisable to achieve improved results.

The bootstrap fits for the extraction of the scaling exponents turned out to be not always reliable. Multiple calls of the fit routine yielded varying fit qualities and varying scaling exponents so we decided to take the best quality fits with the smallest errors on the scaling epxonents out of many fit runs. However, the errors are still somewhat large and we suggest to increase the number of simulation samples for each moment $\langle y^n \rangle(L)$ to a value considerably larger than only ten.

We obtained particularly bad fitting to the measurement points for the three-dimensional BTW model at large lattice sizes (see Appendix 2). It is yet unclear if the scaling breaks down at large lattice sizes or if the 100000 drives were just too few drives compared to the number of lattice entries equal to L^3. We suspect that this could be resolved by increasing the number of drives to the order of 1000000. But care must be taken here since the simulations with less drives and less runs can already take several hours for each lattice size.

Large lattice sizes in three dimensions for the custom model could not be investigated due to limited computational resources.

The results for the scaling exponents (see Table I) of the BTW model can be compared to many different literature values from various papers [4], which are essentially in accordance with our measurements for the two-dimensional case. For the three-dimensional measurements we observe slight deviations to smaller values compared to the literature. It is not always given an appropriate measurement uncertainty in the literature, though, and also values from different papers are not necessarily compatible to each other within the uncertainties. Thus we believe that an overall good agreement to the literature is given in two as well as in three dimensions. Note also that we suppose that too few drives were performed for the three-dimensional lattices as pointed out before.

The scaling exponents of the custom model are compared to our results of the BTW model to investigate if the different approaches of scalar slopes vs. heights yield a similar scaling behavior, which we would expect because both models are supposed to simulate the same system. We still don't necessarily expect exactly matching values due to the non-isotropic relaxation in the BTW model whereas in the custom model the sand grains are redistributed in a more isotropic way.

From Table I one finds the tendency to smaller K. Most notably the value for the avalanche area dimension is about 2 in the BTW model and about 1 in the custom model. This corresponds to the other, visual observations of narrow, less spreading avalanches of the custom model. The avalanche area dimension indeed seems to reflect the dimensionality of the avalanches. We suggest again to try out incorporating of dropping to the diagonal nearest neighbors to see if it impacts these observations.

The values for ρ tend to be a bit larger, which would mean a slight suppression of large avalanches compared to the BTW model, but they are actually compatible with the BTW values within the uncertainties. To resolve this the uncertainties have to be narrowed down by increasing the number of drives and simulation samples as described above.

All scaling exponents are independent of the chosen critical slope within uncertainties. This should be checked again with improved resolution, too, as well as it should be checked with a broader range of critical slopes. In three dimensions we have only simulations for lattice sizes of $L = 10, 20, 30, 40$ available due to limited computational resources. Also the bootstrapped fitting could not be reliably applied such that the values of the three-dimensional custom model exponents cannot really be considered reliable.

VI. CONCLUSION

In this project two different approaches of cellular automata for sandpile dynamics have been implemented and their dynamic properties investigated. The phenomenon of self-organized criticality and the corresponding scale invariance of the dynamics were observed for simulations in two and three dimensions for both models which supports the general concept of scale invariance in self-organized critical systems. With regard to the rather frequently studied Bak-Tang-Wiesenfeld model, our resulting sets of scaling exponents for avalanche *size* and *duration* mostly reside within a wide spread of many different reference values corresponding to various authors over several decades. Some of these references are displayed in Table II. Much less frequently, the exponents of the avalanche *area* κ, T are stated in literature which limits the references to only values for κ. For 3 dimensions a general deviation to smaller values is most likely due to the insufficient amount of drives for lattice sizes $L \geq 50$ where the number of drives was smaller than the total number of sites on the lattice. Due to the lack of errors on some of the literature values, we are not able to verify exact accordance, though, the deviations in two dimensions are generally within one σ. By conducting a moment analysis rather than a simple linear regression to the PDF of the respective observable, we were able to extract the scaling exponents while giving a reasonable estimation on their uncertainty as well as taking the influence of the finite size of the lattice on the PDF (*finite-size effects*) into account. Here, we want to point out that a number greater than ten simulation samples per lattice size is very likely to improve the moment analysis and ten should be regarded as a lower limit rather than a desired value. Generally, an increase of samples, drives, lattice sizes and finally dimensions for both models is likely to improve the obtained results and narrow down the uncertainties.

Appendix

1. Simple power law scaling

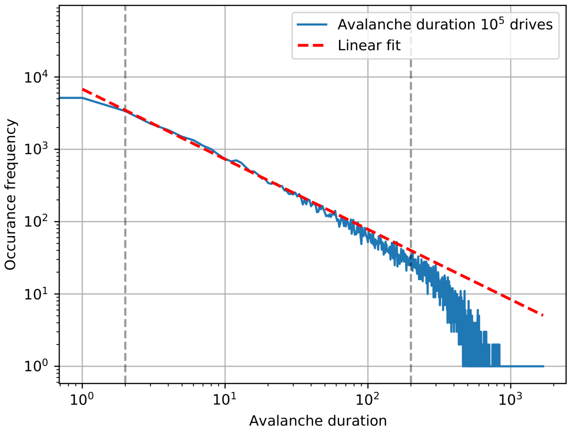


Figure 9: Naive straight line fit to the raw log-log plot of the avalanche duration distribution (BTW model, 100 × 100 lattice) for obtaining the critical exponent α.

2. Moment fitting in three dimensions (BTW)

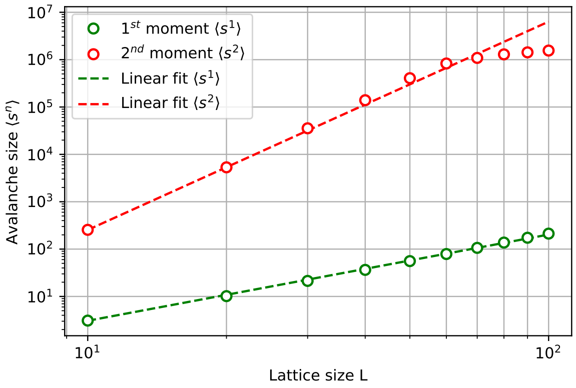


Figure 10: Fitting of first and second avalanche size moments for a three-dimensional sandbox (BTW model). The second moments show noticeable deviations from the straight line fit for large lattice sizes. Error bars have been omitted for reasons of clarity.

3. Literature values of scaling exponents

Table II: Example literature values of scaling exponents for avalanche size and duration of the two- and three-dimensional BTW model, taken over from [4].

Dim.	Size		Duration		Area
	τ	D	α	Z	κ
2D	1.27(1)[8]	2.73(2)[8]	1.16(3)[9]	1.02(5)[10]	1.333 [11]
3D	1.333 [11]	3.004 [12]	1.6 [11]	1.618 [12]	1.333 [11]

[1] Microscopic in the sense of the size of a sand grain compared to the whole pile.
[2] S. Hergarten, *Natural Hazards and Earth System Science*, **3** (2003).
[3] Note that the slope can still depend on these parameters.
[4] G. Pruessner, *Self-Organized Criticality; Theory, Models and Characterisation* (2012).
[5] P. Bak, C. Tang, and K. Wiesenfeld, Physical Review Letters **59** (1987).
[6] K. Christensen, H. C. Fogedby, and H. Jeldtoft Jensen, Journal of Statistical Physics **63**, 653 (1991).
[7] Here we count no triggered avalanches as zero-size avalanches.
[8] A. Chessa, H. E. Stanley, A. Vespignani, and S. Zapperi, Phys. Rev. E **59**, R12 (1999).
[9] J. A. Bonachela, (2008).
[10] M. De Menech and A. L. Stella, Phys. Rev. E **62**, R4528 (2000).
[11] S. Lübeck and K. D. Usadel, Phys. Rev. E **56**, 5138 (1997), cond-mat/9708055.
[12] S. Lübeck, Phys. Rev. E **61**, 204 (2000), cond-mat/9910374.