Asynchronous updating induces order in threshold coupled systems

Manish Dev Shrimali,[1,2] Sudeshna Sinha,[3] and Kazuyuki Aihara[1,2]
[1]*Aihara Complexity Modeling Project, ERATO, JST, Tokyo 151 0064, Japan*
[2]*Institute of Industrial Science, The University of Tokyo, Tokyo 153 8505, Japan*
[3]*The Institute of Mathematical Sciences, CIT Campus, Taramani, Chennai 600 113, India*
(Received 12 February 2007; revised manuscript received 21 August 2007; published 10 October 2007)

We study a class of models incorporating threshold-activated coupling on a lattice of chaotic elements, evolving under updating rules incorporating varying degrees of synchronicity. Interestingly, we observe that asynchronous updating, both random and sequential, yields more spatiotemporal order than parallel (synchronous) updating. Further, the order induced by random asynchronous updating is very robust and occurs even for small asynchronicities in the temporal evolution of the local dynamics. So this case study suggests a very different mechanism for inducing regularity in extended systems.

DOI: 10.1103/PhysRevE.76.046212 PACS number(s): 05.45.−a

I. INTRODUCTION

A basic aim of modeling in physics is to provide suggestive conceptual frameworks for understanding complex phenomena which are generic in physical systems. The hope is to build prototypes that can yield a repertoire of dynamics reminiscent of behavior observed in systems where spatial extent is important, for instance pattern formation (such as dislocation dynamics, intermittency in space time, spatial patterns), Josephson junction arrays, neural dynamics, coupled systems of optically bistable devices, electron-hole plasmas, and even parallel computing and evolutionary biology [1]. One of the important prototypes of extended complex systems are nonlinear dynamical systems with spatially distributed degrees of freedom, or alternately spatial systems composed of large numbers of low-dimensional nonlinear systems. The basic ingredients of such systems are (i) creation of local chaos or local instability by a low-dimensional mechanism and (ii) spatial transmission of energy and information.

Now many results have been obtained for the time evolution of coupled nonlinear systems under parallel (or synchronous) updating. In such models all individual local maps of the lattice are iterated forward simultaneously. Here on the other hand, we will focus on *asynchronous evolution*, where the *updates are not concurrent*. There have been a few studies on asynchronously evolving extended dynamical systems [2–9], and there are strong reasons to revisit this fundamental issue in the light of the fact that asynchronous updating can be closer to physical reality than synchronous updates in certain situations. So in the physical situations where a coupled system is comprised of a collection of elemental dynamical units which evolve asynchronously, the lattice dynamical model must of course employ asynchronous updating schemes. It is then of importance to investigate the effects of asynchronicity (if any) in prototype models.

Specifically, in this paper we will focus on a class of systems incorporating *threshold-activated coupling* on a lattice of nonlinear dynamical elements [10]. We investigate the range of pattern formation obtained under varying degrees of synchronicity, for different threshold levels and relaxation times.

The basic motivation behind this work is to identify whether synchronicity of the local (nonlinear) dynamics makes a significant difference to the spatiotemporal behavior emerging from the extended interactive system [10]. The results from this test bed would be a strong indicator of what to expect from similar models. What we will show in this study is that *asynchronous* updating leads to more *ordered* behavior than simultaneous updating. The details of the model and our results are described in the sections below.

II. MODEL

In our model, time is discrete, labeled by n. Space (or the local node) is labeled by sets of integers, for instance, in the one-dimensional case by i, $i=1,N$, where N is the system size. The state variable $x_n(i)$ (which in physical systems could be quantities like energy, velocity, pressure or concentration) is continuous. Each individual site in the lattice evolves under a suitable nonlinear map $f(x)$. For instance, the local map $f(x)$ ($x\in[0,1]$) can be chosen to be the chaotic logistic map $x_{n+1}=\mu x_n(1-x_n)$ ($\mu=4.0$) or the tent map $x_{n+1}=1-2|x_n-\frac{1}{2}|$. Such maps have widespread relevance as prototypes of low-dimensional chaos.

Now, on this nonlinear lattice a threshold-activated coupling is incorporated [10–12]. The coupling is triggered when a site in the lattice exceeds the critical value x^*, i.e., when a certain site $x_n(i)>x^*$. The supercritical site then relaxes (or "topples") by transporting the excess $\delta=(x_n(i)-x^*)$ equally to its two neighbors,

$$x_n(i) \to x^*,$$

$$x_n(i-1) \to x_n(i-1)+\delta/2,$$

$$x_n(i+1) \to x_n(i+1)+\delta/2. \tag{1}$$

The process above occurs in parallel, i.e., all supercritical sites at any instant relax simultaneously, according to Eqs. (1), and this constitutes one relaxation time step. After r such relaxation steps, the system undergoes the next chaotic update. In some sense then, time n associated with the chaotic dynamics is measured in units of r. The relaxation of a site

may initiate an avalanche of relaxation activity, as neighboring sites can exceed the threshold limit after receiving the excess from a supercritical site, thus starting a domino effect. This induces a bidirectional transport to the two boundaries of the array. These boundaries are open so that the snowballing "excess" may be transported out of the system. After r relaxation steps the next dynamical update of the sites occurs.

The threshold mechanism is reminiscent of the Bak-Tang-Wiesenfeld cellular automata algorithm [13], or the "sandpile" model. This model gave rise to self-organized criticality (SOC), and commanded extensive research interest. The model here is however significantly different, the most important difference being that the threshold mechanism now occurs on a *nonlinearly evolving substrate*, i.e., there is an *intrinsic deterministic dynamics* at each site. So the local chaos here is like an *internal* driving or perturbation, as opposed to *external* perturbation/driving in the sandpile model, which is effected by dropping "sand" from outside.

The discrete-valued driven-from-outside "sandpile" models of self-organized criticality have been relevant in phenomena ranging from earthquakes to ecology [14]. Our modified scenario with continuous state variables and intrinsic driving from local instabilities, should also be relevant to the range of phenomena addressed by the more idealized "sandpile" model [15]. Different natural or human engineered situations will of course necessitate modifications of this basic model, in order to suit the specifics of the phenomena it attempts to capture. However, the study of the generalized abstract model can serve as a test bed for understanding the wide class of extended systems coupled by threshold responses.

In particular, this kind of threshold (pulse) coupled distributed systems are relevant for describing integrate-and-fire oscillators [16]. Also certain mechanical systems like chains of nonlinear spring blocks and the phenomena of fracture and crack propagation in materials [17] can be described by variants of this model. Further, some biological phenomena particularly in ecology, such as population migrations could be well captured by such a model. For instance, it is reasonable to model the population of an area (state at a site) as a logistic map and when this population exceeds a certain critical amount the excess population moves to a neighboring area (site). The spatiotemporal dynamics resulting from such a model mimics the movement of population among geographical regions. Last, in the context of engineering and computer science, the phenomena arising from this model is relevant to asynchronous artificial neural network models [18].

The spatiotemporal behavior of the lattice under different threshold levels x^*, for the case of $r\to\infty$ (namely the situation where the array relaxes fully before the subsequent chaotic update), was investigated both numerically and analytically in [10–12]. Specifically, for the case of networks of chaotic logistic maps $[f(x)=4x(1-x)]$ there exist many *phases* in x^* space $(0<x^*<1)$ [11]. For example, for $x^* \leq 3/4$ the dynamics goes to a fixed point. When $0.75<x^*<1.0$, the dynamics is attracted to *cycles* whose periods depend on the value of x^*. By tuning x^* one thus obtains spatiotemporal cycles of varying orders.

Note however, that the dynamical outcome crucially depends the relaxation time r, i.e., on the time scales for autonomously updating each site and propagating the threshold-activated coupling between sites. As mentioned above, the limiting case of large r has been extensively studied. Since the local chaos in the maps can be thought of as "intrinsic" perturbation, this case corresponds to the "dilute" perturbation limit. When $r\to\infty$, the system is fully relaxed before the subsequent dynamical update. So the time scales of the two processes, the intrinsic chaotic dynamics of each site and the threshold-activated relaxation, are separable. Here the relaxation mechanism is much faster than the chaotic evolution, enabling the system to relax completely before the next chaotic iteration. This scenario is similar to the SOC model, where the driving force (perturbation) is very dilute, e.g., in the sandpile model the avalanche of activity, initiated by an external "sand grain" dropped on the pile, ceases before the next "sand grain" perturbs the pile.

At the other end of the spectrum is the limit of very small r where the local dynamics and the coupling take place simultaneously. It is evident that lowering r essentially allows us to move from a picture of separated relaxation processes to one where the relaxation processes can overlap, and disturbances do not die out before the subsequent chaotic update. It was observed in [19] that for short relaxation times the system is driven to spatiotemporal chaos. This is due to the fast driving rate of the system which does not provide enough time to spatially distribute the perturbations and allow the excess to propagate to the open boundaries. However large r gives the system enough time to relax, and allows the excess to be transported out of the system through the open ends. So for large r the system displays very regular behavior for a large range of threshold values.

Now in order to effectively study the influence of varying degrees of synchronicity we investigate the following dynamics: We break the lattice into subsets, and update the sites belonging to each subset simultaneously, while updating the different subsets sequentially. We denote the number of sites (nodes) updated together as n_{sync}. So the fraction $p_{\text{sync}}=n_{\text{sync}}/N$ serves as an effective parameter for synchronicity. The limiting case of $p_{\text{sync}}=n_{\text{sync}}/N=1$, namely $n_{\text{sync}}=N$, corresponds to the usual parallel updates. On the other hand, $n_{\text{sync}}=1/N$, which tends to 0 as $N\to\infty$, corresponds to the completely asynchronous update. So as n_{sync}/N takes values from 0 to 1 (i.e., n_{sync} takes values 1 to N), one has decreasing degrees of asynchronicity in the evolution.

One can also construct other interesting models of asynchronous dynamics. For instance, one can have models where the threshold response is asynchronous too, namely, the threshold response occurs *only* after a chaotic update. So effectively then, the threshold mechanism will operate at random intervals, rather than at the instant a site is above the critical value (as in our model). Such a model would be closer to the type of asynchronicity seen in [8], but the connection between it and the sandpilelike models of SOC will not be straightforward. Note that in this work we will focus only on the model given above, where the driving due to the intrinsic chaos is asynchronous, while the threshold response occurs whenever a site is above the critical threshold.

We investigate two cases: (i) *random asynchronous up-*

dating, where the subsets of n_{sync} sites are chosen randomly, and changed at each iteration; (ii) *sequential updating* (or the "typewriter mode"), where the subsets are fixed and ordered, namely the first subset runs from site index 1 to n_{sync}, the next from $n_{\text{sync}}+1$ to $2n_{\text{sync}}$ up to $(N-1)n_{\text{sync}}+1$ to Nn_{sync}. In both cases $n_{\text{sync}}=N$ corresponds to the synchronous updating limit.

In the sections below we investigate how the synchronicity parameter affects the spatiotemporal characteristics of the threshold coupled chaotic lattice, at different relaxation times r.

III. SPATIAL PATTERNS UNDER ASYNCHRONOUS UPDATING

Here we consider coupled logistic map arrays under threshold coupling. We consider thresholds in the range $0<x^*<0.75$. In this threshold range when relaxation time is very long $(r\to\infty)$ all updating schemes yield a synchronized spatiotemporal fixed point, with all sites $x_n(i)\sim x^*$ for all i and n. So in the limit of long relaxation times, when all supercritical sites in the system have the opportunity to fully relax via the threshold mechanism, both synchronous and asynchronous updating yield similar results.

The space-time density plot of the state variable $x(i)$ for threshold coupled logistic maps is shown for the case of random, sequential, and synchronous updating (Fig. 1). In these figures, the top two panels show the spatiotemporal evolution of the system updated asynchronously, one site at a time (i.e., $p_{\text{sync}}=1/N$) randomly (a) and sequentially (b). The bottom panel show the system under synchronous updating (namely all N sites are updated together, so $p_{\text{sync}}=1$). It is clearly evident that the system is more ordered and spatially synchronized under asynchronous updating as compared to its spatiotemporal state under synchronous updating.

Figures 2(a) and 2(b) show illustrative examples of the dynamical states obtained for $r=1$ under random asynchronous updating and sequential updating, with different degrees of synchronicity $p_{\text{sync}}=n_{\text{sync}}/N$. Again, clearly for low p_{sync} (more asynchronicity) the lattice is much more synchronized. Also note that random asynchronous updating yields an ordered state over a much larger range of asynchronicity as compared to the case of sequential updating. For instance, in Fig. 2 it is evident that for the case of random asynchronous updating the range $0<p_{\text{sync}}<0.3$ yields spatiotemporal order, while for sequential updating a much smaller range $(0<p_{\text{sync}}<0.05)$ yields ordered states. So the order induced by random asynchronous updating is very robust indeed, and occurs even for small asynchronicities in the temporal evolution of the local dynamics.

IV. SYNCHRONIZATION ORDER PARAMETER

We study the degree of synchronization [20] in the system quantitatively through an average error function defined as the mean square deviation of the spatial profile,

$$Z(n)=\frac{1}{N}\sum_{i=1}^{N}[x_n(i)-\langle x_n\rangle]^2, \tag{2}$$

where $\langle x_n\rangle$ is the average $[1/N\Sigma_{i=1,N}x_n(i)]$ of the spatial profile at time n. We denote this quantity averaged over time n and over different realizations as $\langle Z\rangle$. When $\langle Z\rangle=0$, we have complete synchronization. Higher values of $\langle Z\rangle$ indicate lower degree of synchronization.

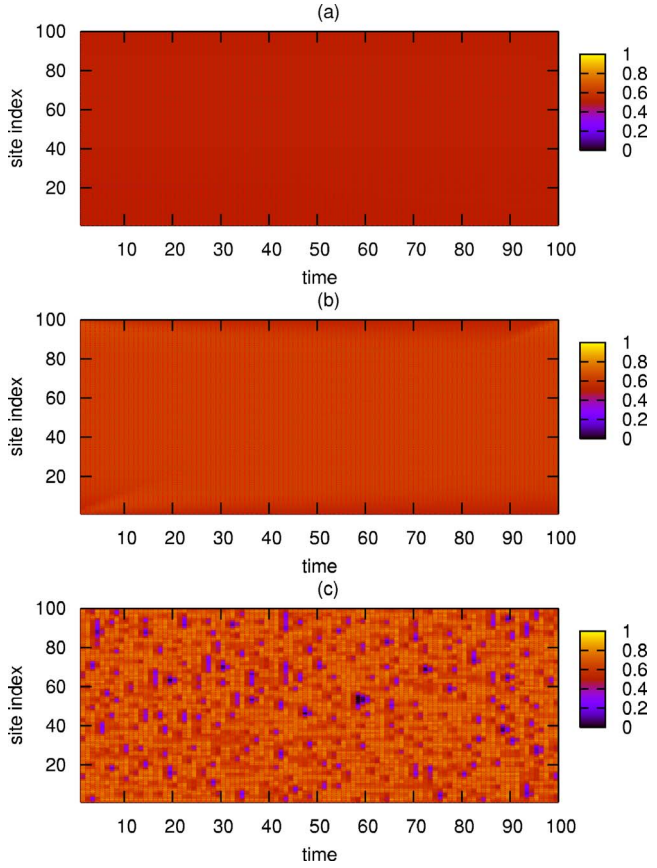

FIG. 1. (Color online) Space-time density plots of an array of threshold coupled logistic maps, with threshold value $x^*=0.5$, size $N=100$ and relaxation time $r=1$. The top two panels show the cases of (a) random and (b) sequential updating, with $n_{\text{sync}}=1$. The bottom panel (c) shows the case of synchronous updating with $n_{\text{sync}}=100$ (i.e., $p_{\text{sync}}=1$). The horizontal axis denotes the time and the vertical axis denotes the site index.

We also compute the average error function normalized by the average value of x in the array, namely,

$$\overline{Z(n)}=\frac{1}{N}\sum_{i=1}^{N}\frac{[x_n(i)-\langle x_n\rangle]^2}{\langle x_n\rangle^2}. \tag{3}$$

We denote this quantity averaged over time n and over different realizations as $\langle\bar{Z}\rangle$. Again, higher values of $\langle\bar{Z}\rangle$ indicate lower degree of synchronization.

Figure 3 displays $\langle\bar{Z}\rangle$ for very short relaxation time $r=1$ and very long relaxation time $r=1000$, under random updating and sequential updating. In the case of random updating, the synchronization error $\bar{Z}$ remains low for reasonably high values of synchronicity p_{sync}. In the case of sequential updating there is a sharp transition from the synchronous to the nonsynchronous state, signaled by a jump in the synchronization error $\bar{Z}$, at fairly low values of p_{sync}. For instance, this

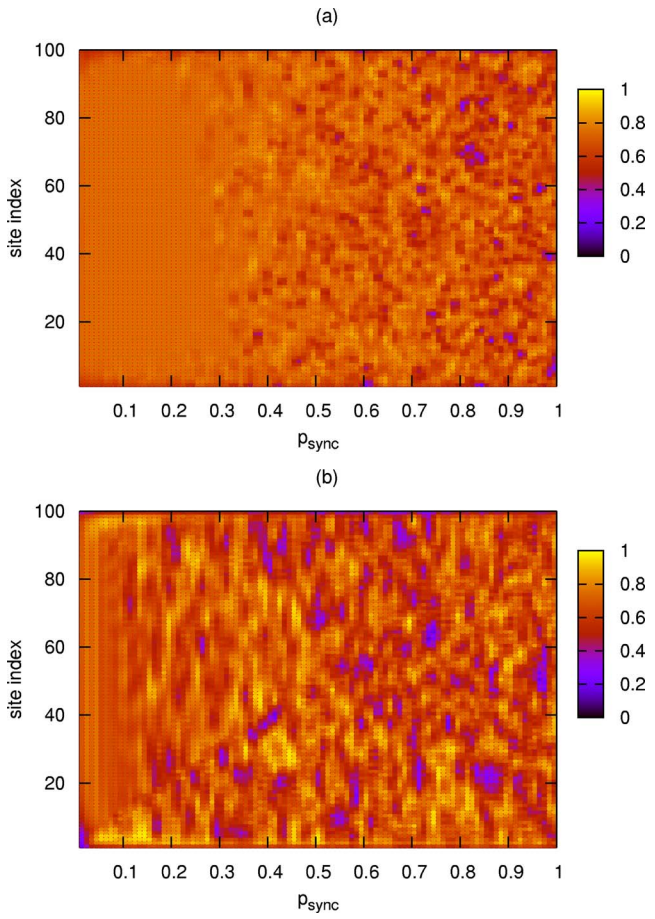

FIG. 2. (Color online) Density plots of the state of an array of threshold coupled logistic maps, at an instant of time, with threshold value $x^*=0.5$, size $N=100$ and relaxation time $r=1$. The coupled maps evolve under (a) random asynchronous and (b) sequential updating, with varying degrees of synchronicity p_{sync}. The vertical axis denotes the site index and the horizontal axis denotes the value of p_{sync} $(0<p_{\text{sync}}\leq 1)$.

occurs around $n_{\text{sync}}=5$ $(p_{\text{sync}}=n_{\text{sync}}/N)$ in the case of $r=1$ (see Fig. 3). Again this indicates that random updating yields ordered states over a wider range of p_{sync} than sequential updating does.

Further the synchronization error $\bar{Z}$ is studied for different threshold values x^* (see Figs. 4 and 5). For instance, Fig. 4 shows $\bar{Z}$ over a range of threshold values, for the coupled maps evolved under random $(n_{\text{sync}}=1)$, sequential $(n_{\text{sync}}=1)$, and synchronous $(n_{\text{sync}}=N)$ updating schemes, with relaxation time $r=1$. Figure 5 shows $\bar{Z}$ as a function of varying degrees of synchronicity p_{sync}, for random and sequential updating. Clearly all these results demonstrate that random and sequential updating yields more order, namely, lower synchronization error, than synchronous updating for all threshold values.

As a specific illustrative example, consider the implications of these results for migrating populations. As mentioned earlier, a nonlinear map, in particular, the logistic map, is a reasonable model for the population of a geographical location and/or area (site). If this population is too

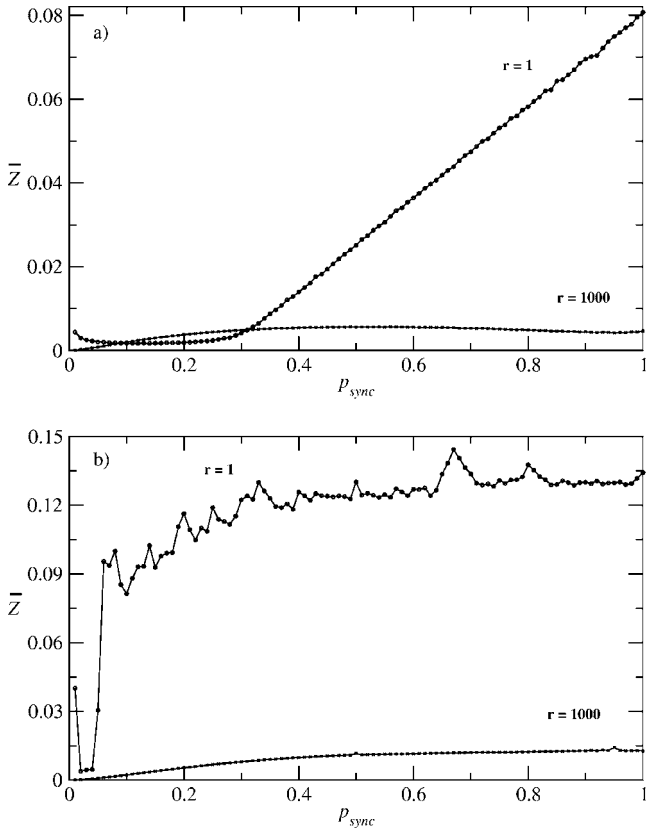

FIG. 3. The synchronization order parameter $\bar{Z}$ as defined in the text vs the degree of synchronicity p_{sync} $(0<p_{\text{sync}}\leq 1)$ for an array of threshold coupled logistic maps, with threshold value $x^*=0.5$, size $N=100$ and relaxation time $r=1$ and 1000. The coupled maps evolve under random updating in (a) and sequential updating in (b).

large, migration of the excess to neighboring locations occurs. The threshold is given by the sustainable population at the location. Now the results above would suggest that, if the growth at the locations is asynchronous the resulting population distribution among locations would be more evenly distributed. If, on the other hand, the population growth in different locations are synchronous, the population distribution of the locations will be very nonuniform and the population profile will be more irregular.

V. CRITICAL RELAXATION TIME

We observe that there is a critical value of relaxation time r after which the spatiotemporal behavior of the system is quite indistinguishable from the very long relaxation time limit. We denote this value of relaxation time as r_c. So the behavior of the system with $r\geq r_c$ is essentially the same as for $r\to\infty$. Recall that the spatiotemporal dynamics in the threshold range $0\leq x^*\leq 3/4$, in the infinite relaxation time scenario, yields a spatiotemporal fixed point with all sites at x^*. This implies that r_c is the smallest relaxation time at which the system yields nearly zero synchronization error.

Now this critical relaxation time r_c is naturally system-size dependent, and we observe that it varies as N^{ϕ} for large

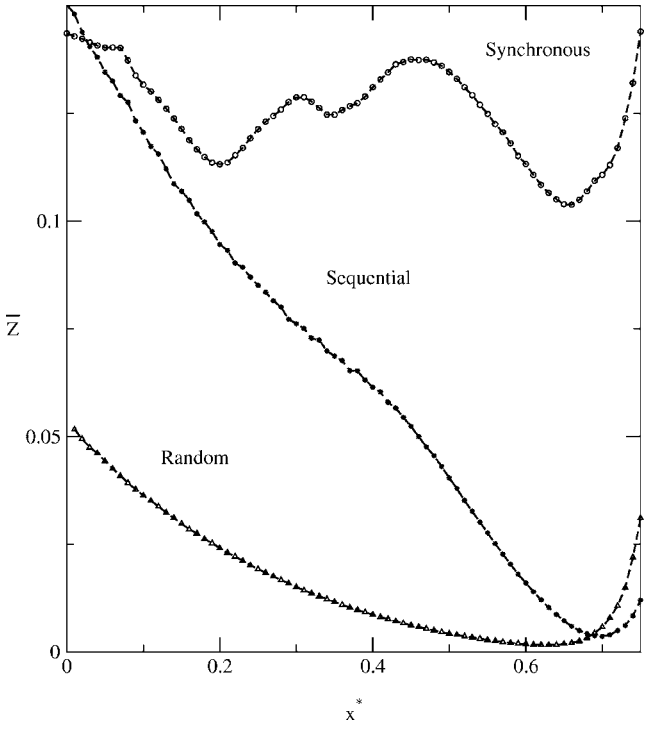

FIG. 4. The synchronization order parameter $\bar{Z}$ as defined in the text with respect to threshold value x^*, for an array of threshold coupled logistic maps, with relaxation time $r=1$. The coupled maps evolve under random ($n_{\mathrm{sync}}=1$), sequential ($n_{\mathrm{sync}}=1$), and synchronous ($n_{\mathrm{sync}}=N$) updating.

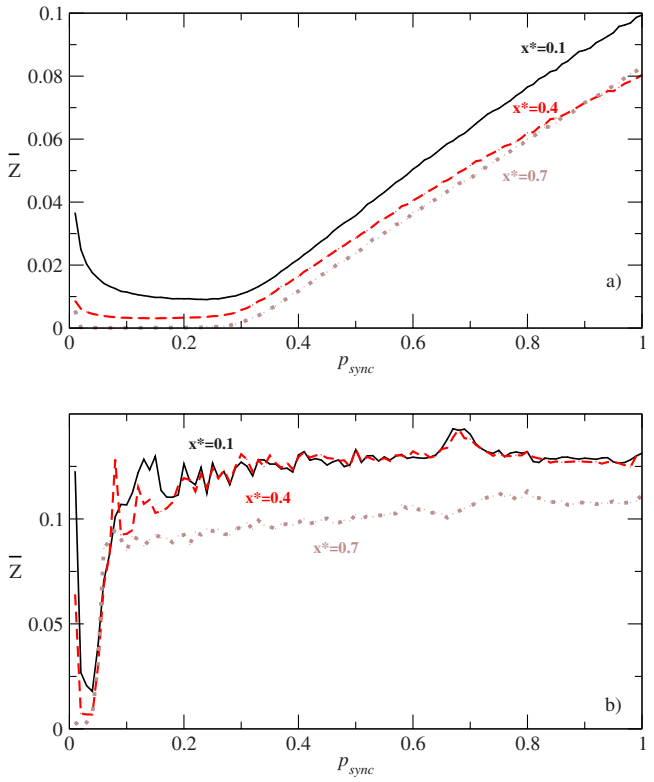

FIG. 5. (Color online) The synchronization order parameter $\bar{Z}$ as defined in the text vs. the degree of synchronicity p_{sync} ($0<p_{\mathrm{sync}}\leq 1$) for an array of threshold coupled logistic maps, with threshold value $x^*=0.1$ (solid), 0.4 (dashed), and 0.7 (dotted), size $N=100$ and relaxation time $r=1$. The coupled maps evolve under random updating in (a) and sequential updating in (b).

N, with $1\leq\phi\leq 2$. The value of $\phi\sim 2$ for synchronous updating schemes, while $\phi\sim 3/2$ for sequential updating and $\phi\sim 1$ for random asynchronous updating (see Fig. 6). If the movement of the "excess" via threshold-activated transport from the interior to the open edges is like a random walk, one would expect the time taken for the excess from the bulk to reach the open ends, which determines the critical relaxation time r_c, to scale with system size as N^2. So it appears that for synchronous updating the transport is effectively random walklike, while it is super diffusive for asynchronous updating, with the sequential case being Levy flightlike and the random updating effectively behaving as ballistic transport.

We also find that the critical relaxation time r_c decreases with increasing asynchronicity in the evolution (see Fig. 7), i.e., the spatiotemporal dynamics is quite indistinguishable from the infinite relaxation time limit (when synchronization error is close to zero) at smaller relaxation times for more asynchronous updating. Namely, the system yields synchronization sooner, as the asynchronicity in the evolution increases.

The average critical relaxation time r_c, where the synchronization error becomes zero, is plotted in Fig. 8 as a function of threshold value x^*, with the coupled maps evolving under random, sequential, and synchronous updating. Clearly, asynchronous updating has lower r_c for all values of threshold.

In order to check the generics of our results, we have also studied systems with the local dynamics given by different one-dimensional maps, as well as the case of higher dimensional lattices. We find that these models also show similar results, namely, asynchronous updating yields more spatial order than synchronous updating in all cases. This suggests that the regularizing effect of asynchronicity may be quite general and broadly applicable.

VI. ANALYSIS

A. Local dynamics

First consider the effect of the threshold response on a single element evolving under a chaotic map $f(x)$. The threshold condition implies that when the dynamical state of the element exceeds a threshold x^*, it relaxes down to x^*, namely, the evolution of the element is governed by the function $x_{n+1}=f(x_n)$ when $x_{n+1}<x^*$, but $x_{n+1}=x^*$ when $x_{n+1}>x^*$. For the single element the excess is "transported" out of the element when it exceeds threshold (as it is an effective open edge for transport in either direction).

The effective map $\overline{f(x)}$ of the threshold chaotic dynamics is a "beheaded" or "flat-top" nonlinear map, with $x_{n+1}=x^*$ when $f(x_n)>x^*$. This effective map supports a stable fixed point $x=x^*$ for thresholds x^* lower than x_{fixed}, where x_{fixed} is the (unstable) fixed point solution of the chaotic map $f(x)$. The fixed point $x=x^*$ can be obtained graphically by the

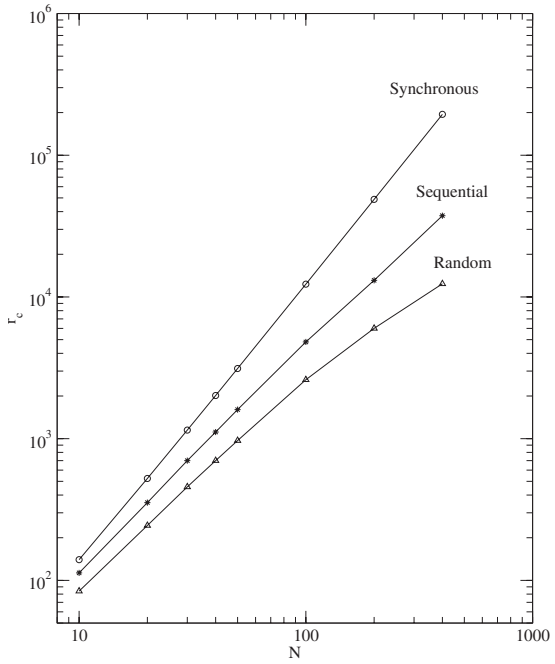

FIG. 6. Critical relaxation time r_c vs. system size N, for an array of threshold coupled logistic maps. The case of random and sequential updating (with $n_{\text{sync}}=1$), as well as synchronous updating ($n_{\text{sync}}=N$) is shown here.

intersection of the $x_{n+1}=x_n$ line with the flat portion of the threshold chaotic map $f(x)$, and this fixed point is super stable as the slope at the intersection is zero. For instance, for the case of the logistic map, where $x_{\text{fixed}}=3/4$, the range of thresholds under which a single chaotic element yields a

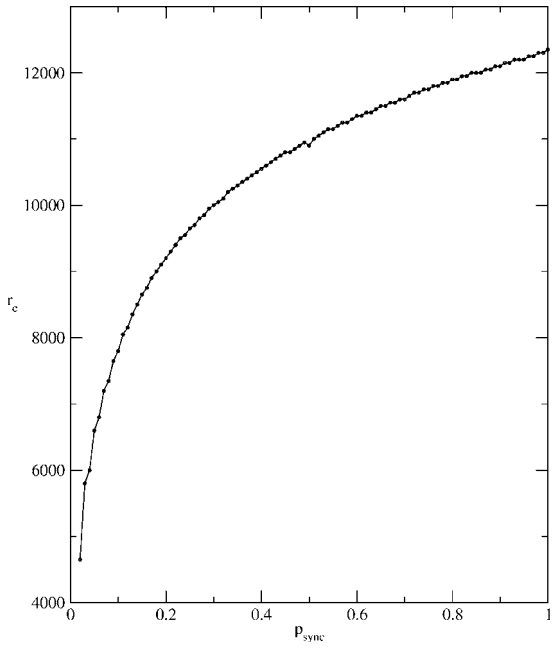

FIG. 7. Critical relaxation time r_c vs the degree of synchronicity p_{sync}, for an array of threshold coupled logistic maps. The case of sequential updating is shown here.

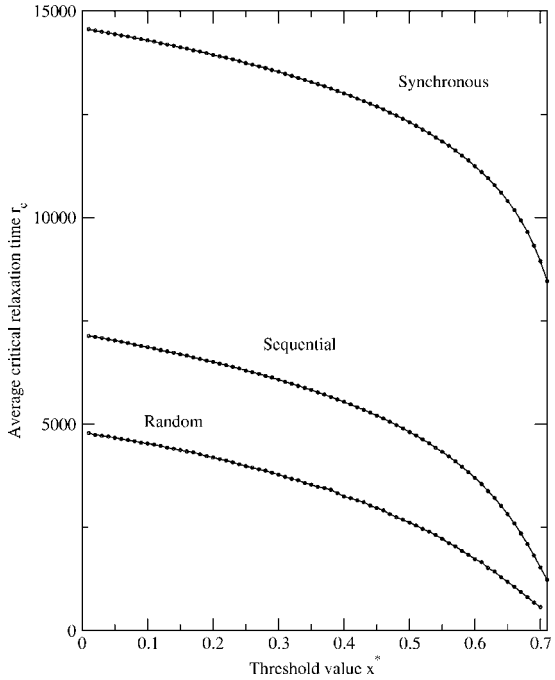

FIG. 8. The average critical relaxation time r_c, where the synchronization error becomes zero, is plotted as a function of threshold value x^*, for an array of threshold coupled logistic maps. The cases of asynchronous updating (random updating and sequential updating), and synchronous updating are displayed.

stable fixed point at x^* is $0 \leq x^* \leq 3/4$ [11]. This is in marked contrast to the dynamics without threshold which is strongly chaotic.

The fact that all thresholds $x^* \leq x_{\text{fixed}}$ will yield a fixed point for a single element can also be seen from the fact that the map $x_{n+1}=f(x_n)$ is above the line $x_{n+1}=x_n$ when $x_n < x_{\text{fixed}}$. Since the system is ergodic, x is guaranteed to exceed x^*. Now starting from $x_n=x^*$ one immediately obtains that the subsequent chaotic update will yield a state greater than x^*. This supercritical state will be clipped back, under the threshold mechanism, to x^*: so x will always be equal to x^*.

Now in an array of maps with thresholds in the fixed point regime, the threshold-activated transport leads to the individual elements relaxing toward the fixed point at x^* of the individual threshold maps $f(x)$. So in the limit of long relaxation times, namely $r \to \infty$, asymptotically $x(i) \to x^*$. But when the relaxation time between chaotic updates is small all the "excess" of the system will not have the time to move out to the open boundaries, and all the sites will not be at $x=x^*$. For synchronous local dynamics, for low r this is what happens, and one obtains very irregular spatial profiles with complete lack of spatial coherence. The remarkable thing however is that, when the elements evolve asynchronously, even for very low r (for instance, $r=1$) the system attains a great degree of spatial coherence.

In the following sections we will specifically consider the robustness of a synchronized spatial profile, where all sites i are at the critical value x^*. Now after a chaotic update, i.e., just at the onset of the relaxation process ($r=0$) the updated

TABLE I. Evolution of the state of an array of N elements with an initial uniform profile, under synchronous chaotic updates. Here the transport is in one direction, namely if site i is above threshold it will transfer the excess to its neighbor $i+1$. The boundary at $i=N$ is open. There is one relaxation step $(r=1)$ between the chaotic updates. Here, $\delta_1=f(x^*)-x^*$, $\delta_2=f(x^*+\delta_1)-x^*\sim\delta_1[1+f'(x^*)]$, …, $\delta_N=f(x^*+\delta_{N-1})-x^*\sim\delta_1[1+f'(x^*)+f'^2(x^*)+f'^3(x^*)+\cdots]$ under the assumption that $\delta_1\ll 1$.

Initial state	$x(1)=x^*$	$x(2)=x^*$	$x(3)=x^*$	$x(N-1)=x^*$	$x(N)=x^*$
Chaotic update step 1	$f(x^*)$	$f(x^*)$	$f(x^*)$	$f(x^*)$	$f(x^*)$
Relaxation step $r=1$	x^*	$x^*+\delta_1$	$x^*+\delta_1$	$x^*+\delta_1$	$x^*+\delta_1$
Chaotic update step 2	$f(x^*)$	$f(x^*+\delta_1)$	$f(x^*+\delta_1)$	$f(x^*+\delta_1)$	$f(x^*+\delta_1)$
Relaxation step $r=1$	x^*	$x^*+\delta_1$	$x^*+\delta_2$	$x^*+\delta_2$	$x^*+\delta_2$
Chaotic update step $N-1$	$f(x^*)$	$f(x^*+\delta_1)$	$f(x^*+\delta_2)$	$f(x^*+\delta_{N-2})$	$f(x^*+\delta_{N-2})$
Relaxation step $r=1$	x^*	$x^*+\delta_1$	$x^*+\delta_2$	$x^*+\delta_{N-2}$	$x^*+\delta_{N-1}$
Chaotic update step N	$f(x^*)$	$f(x^*+\delta_1)$	$f(x^*+\delta_2)$	$f(x^*+\delta_{N-2})$	$f(x^*+\delta_{N-1})$
Relaxation step $r=1$	x^*	$x^*+\delta_1$	$x^*+\delta_2$	$x^*+\delta_{N-2}$	$x^*+\delta_{N-1}$

sites will be at the state $f(x^*)$. In the fixed point region of threshold parameter space $(0<x^*<0.75)$, this implies that all the updated sites are above threshold, as $f(x^*)>x^*$. So all updated sites under the threshold response will transport the excess amount $[f(x^*)-x^*]$ on to the neighboring sites. Now the pertinent question here is how much and how quickly the uniform profile is distorted in the case of synchronous and asynchronous updating.

Qualitatively, in the case of synchronous updating the entire lattice updates together. When r is large the system has enough time to relax and this deviation is evened out as the excess is transported out of the open edges. So the controlling effect of the threshold-activated relaxation removes the deviation by moving the perturbing amount out of the array from the open ends. However, when r is small, say $r=1$, then the deviation is magnified from update to update, as the deviation does not have the time to be transported out of the open boundaries. So deviations grow rapidly and destroy coherence in space. For completely asynchronous updating on the other hand, for the case of $r=1$, after any one site undergoes a chaotic update, there is a threshold-activated relaxation across the lattice. So the perturbing influence of the chaotic evolution is dilute in space, and the deviation has time to move toward the open edges and be transported out of the system. In the following sections we will analyze more exactly this qualitative argument.

B. Analysis of threshold coupled elements under unidirectional transport

First consider the simpler situation of a threshold coupled lattice of N elements under unidirectional transport, namely, a one-way coupled lattice. Here a supercritical site $x_n(i)$ relaxes by transporting the excess $\delta=[x_n(i)-x^*]$ in one direction to one neighbor (say $i+1$). We start with a uniform initial state where all the elements initially are at values $x(i)=x^*$, $i=1,\ldots,N$.

It is clear from Table I that under synchronous updating all the elements have states different from x^* after very first chaotic updates. The spatial nonuniformity increases in time with each chaotic update, and after N chaotic updates no two sites of the initially uniform lattice have the same state even if $|f(x^*)-x^*|$ is very small. So the system is completely desynchronized.

However under sequential updating (see Table II), starting from a uniform profile, the number of sites in the lattice that are different from x^* is always one, as the sites get updated one by one. So after N updates the system has only one site

TABLE II. Evolution of the state of an array of N elements, starting with a uniform spatial profile, under sequential updating. Here the transport is in one direction, namely if site i is above threshold it will transfer the excess to its neighbor $i+1$. The boundary at $i=N$ is open. There is one relaxation step $(r=1)$ between the chaotic updates. Here, $f(x^*)=x^*+\delta_1$, $f(x^*+\delta_1)=x^*+\delta_2$, …, $f(x^*+\delta_{N-1})=x^*+\delta_N$.

Initial state	$x(1)=x^*$	$x(2)=x^*$	$x(3)=x^*$	$x(N-1)$	$x(N)=x^*$
Chaotic update 1	$f(x^*)$	x^*	x^*	x^*	x^*
Relaxation	x^*	$x^*+\delta_1$	x^*	x^*	x^*
Chaotic update 2	x^*	$f(x^*+\delta_1)$	x^*	x^*	x^*
Relaxation	x^*	x^*	$x^*+\delta_2$	x^*	x^*
Chaotic update $N-1$	x^*	x^*	x^*	$f(x^*+\delta_{N-2})$	x^*
Relaxation	x^*	x^*	x^*	x^*	$x^*+\delta_{N-1}$
Chaotic update N	x^*	x^*	x^*	x^*	$f(x^*+\delta_{N-1})$
Relaxation	x^*	x^*	x^*	x^*	x^*

TABLE III. State of an array of N elements with a uniform spatial profile, after synchronous chaotic updates. Here the transport is in both directions, namely if site i is above threshold it will transfer the excess equally to its two neighbors $i-1$ and $i+1$. The boundaries at $i=1$ and $i=N$ are open. There is one relaxation step ($r=1$) between the chaotic updates, and $\delta=f(x^*)-x^*$, $\delta'=f(x^*+\delta)-x^*$, $\delta''=f(x^*+\delta/2)-x^*$. Due to the isotropy of the transport, one has the symmetry: $x(i)=x(N+1-i)$. It is also evident that after the very first chaotic update (and the consequent relaxation) all elements become different from x^*, with the boundary elements differing from the bulk. Subsequently, at every chaotic update two additional elements become dissimilar.

Initial state	$x(1)=x^*$	$x(2)=x^*$	$x(3)=x^*$	$x(N-2)=x^*$	$x(N-1)=x^*$	$x(N)=x^*$
Chaotic update step 1	$f(x^*)$	$f(x^*)$	$f(x^*)$	$f(x^*)$	$f(x^*)$	$f(x^*)$
Relaxation step $r=1$	$x^*+\frac{\delta}{2}$	$x^*+\delta$	$x^*+\delta$	$x^*+\delta$	$x^*+\delta$	$x^*+\frac{\delta}{2}$
Chaotic update step 2	$f(x^*+\frac{\delta}{2})$	$f(x^*+\delta)$	$f(x^*+\delta)$	$f(x^*+\delta)$	$f(x^*+\delta)$	$f(x^*+\frac{\delta}{2})$
Relaxation step $r=1$	$x^*+\frac{\delta'}{2}$	$x^*+\frac{\delta'}{2}+\frac{\delta''}{2}$	$x^*+\delta'$	$x^*+\delta'$	$x^*+\frac{\delta'}{2}+\frac{\delta''}{2}$	$x^*+\frac{\delta'}{2}$

different from x^*, and all other sites at x^*. So clearly sequential updating manages to preserve the uniformity of the initial spatial profile.

From another point of view, for the case of synchronous updating, on an average, excess of the order of $N\delta$ is created after every chaotic update, while only that of the order of δ is transported out every relaxation step. So the perturbation cannot die down when $r\sim 1$. So at the end of N chaotic updates the total excess remaining in the system is $=\Sigma_{i=1}^{N-1}\delta_i$, while the excess transported out is only δ_N. So the excess remaining in the system (which is a source of local instability) is far larger than the excess transported out (which allows the system to relax). This imbalance is the source of spatial disorder in synchronously evolving systems.

For asynchronous updating (random or sequential) on the other hand, the total excess created is of the order of δ every chaotic update. Specifically for sequential updating, after N chaotic updates, the excess in the lattice is δ_N and the excess transported out is also δ_N (see Table II). This leaves the spatial profile completely uniform again, in sharp contrast to the situation under synchronous updates. So what allows the system to be more ordered under asynchronous updating is the steady state between the generation of disturbance and the transport of the perturbing amount from the open system. For sequential updating this balance is exact, and for random updating this holds true statistically.

C. Analysis of threshold coupled elements under bidirectional transport

Now we will analyze in some more detail how the spatial profile changes under bidirectional transport (extensive results from the numerical simulation of this situation are presented in Secs. III–V). Consider a lattice of N elements, with all the elements initially at $x(i)=x^*$, $i=1,\ldots N$. Tables III and IV show how the elements evolve for the cases of synchronous and sequential updating with relaxation time $r=1$. It is clear that under synchronous updating all the elements have states different from x^* after the very first chaotic update. However, under sequential updating, starting from a uniform profile, the number of sites in the lattice that are different from x^* is 1, 2, 2, 3, 3, 4, 4, … (see Table IV), as the sites get updated one by one. So after N updates the system differs at around $N/2$ sites from x^*. So clearly, synchronous updating moves the spatial profile away from the fixed point faster than asynchronous updating.

Again another way of understanding this phenomenon is to consider the total disturbance in the system. This is the

TABLE IV. State of an array of N elements, starting with a uniform spatial profile, after sequential updating. Here the transport is in both directions, namely if site i is above threshold it will transfer the excess equally to its two neighbors $i-1$ and $i+1$. The boundaries at $i=1$ and $i=N$ are open. There is one relaxation step ($r=1$) between the chaotic updates. Here, $\delta=f(x^*)-x^*$, $\delta'=f(x^*+\delta/2)-x^*$, $\delta''=f(x^*+\delta'/2)-x^*$, and $\delta''=f(x^*+\delta''/2)-x^*$.

Initial state	$x(1)=x^*$	$x(2)=x^*$	$x(3)=x^*$	$x(4)=x^*$	$x(5)=x^*$	$x(N)=x^*$
Chaotic update 1	$f(x^*)$	x^*	x^*	x^*	x^*	x^*
Relaxation	x^*	$x^*+\frac{\delta}{2}$	x^*	x^*	x^*	x^*
Chaotic update 2	x^*	$f(x^*+\frac{\delta}{2})$	x^*	x^*	x^*	x^*
Relaxation	$x^*+\frac{\delta'}{2}$	x^*	$x^*+\frac{\delta'}{2}$	x^*	x^*	x^*
Chaotic update 3	$x^*+\frac{\delta'}{2}$	x^*	$f(x^*+\frac{\delta'}{2})$	x^*	x^*	x^*
Relaxation	x^*	$x^*+\frac{\delta'}{4}+\frac{\delta''}{2}$	x^*	$x^*+\frac{\delta''}{2}$	x^*	x^*
Chaotic update 4	x^*	$x^*+\frac{\delta'}{4}+\frac{\delta''}{2}$	x^*	$f(x^*+\frac{\delta''}{2})$	x^*	x^*
Relaxation	$x^*+\frac{\delta'}{8}+\frac{\delta'}{4}$	x^*	$x^*+\frac{\delta'}{8}+\frac{\delta''}{4}+\frac{\delta'''}{2}$	x^*	$x^*+\frac{\delta'''}{2}$	x^*

total excess above the fixed point that is generated by the chaotic updates. The dissipation of this disturbance is achieved by transporting the excess from the open edges. For the case of synchronous updating, on an average, excess of the order of $N\delta$ is created after every chaotic update, while only that of the order of δ (i.e., $\delta/2$ from each edge) is transported out every relaxation step. So the perturbation cannot die down when $r \sim 1$. For asynchronous updating (random or sequential) on the other hand, the total excess created is of the order of δ every chaotic update and typically a similar amount gets transported out of the open edges after each chaotic update. So statistically speaking there is a steady state between the generation of disturbance and the transport of the perturbing amount from the open system. This allows the system to be much more ordered than the synchronously evolving system.

Qualitatively speaking then, asynchronous updating is akin to the *slow driving limit* of the perturbation due to the intrinsic local chaos. Asynchronous evolution is similar to the driving in the "sandpile model" where the "grains of sand" are sprinkled one by one, rather than being poured down on the entire pile at once, i.e., the extrinsic driving is dilute. In our case, the local instabilities induced by the internal chaos, have an effectively slower time scale when they occur asynchronously, as asynchronous local chaos triggers threshold-activated interactions between the elements more sparsely in the lattice than synchronous local chaos, where all sites can go critical together. So asynchronous evolution gives the relaxation process more time to bring order and organization.

VII. CONCLUSIONS

We have studied a general class of models incorporating threshold-activated coupling on a lattice of chaotic elements, under varying degrees of asynchronicity in the updating rules. In fact in the absence of a global clock, asynchronicity is expected to be very natural in the evolution of large interactive systems. From our study it is evident that the degree of asynchronicity in the evolution of the local dynamical components of the extended system crucially determines the spatiotemporal dynamics of the system. So, though it has been studied little so far (compared to the extensive literature on models implementing parallel updating), our results provide indications that the updating mechanism is a very important issue, and needs to be understood.

The central observation, borne out by our extensive simulations, is that *asynchronous updating favors synchronization*. Further the order induced by random asynchronous updating is very robust, and occurs even for small asynchronicities in the temporal evolution of the local dynamics. So this suggests a very different mechanism for inducing regularity in physical and biological systems.

[1] *Theory and Applications of Coupled Map Lattices*, edited by K. Kaneko (Wiley, New York, 1993), and references therein.

[2] There are cases where synchronous updates can lead to complicated spatial structures in cellular automata [M. Nowak and R. M. May, Nature **359**, 826 (1992)] which disappear under asynchronous updating [B. A. Huberman and N. Glance, Proc. Natl. Acad. Sci. U.S.A. **90**, 7716 (1993)] and are therefore considered unphysical in the absence of an external clock.

[3] E. D. Lumer and G. Nicolis, Physica D **71**, 440 (1994).

[4] M. Mehta and S. Sinha, Chaos **10**, 350 (2000).

[5] In the context of Monte Carlo simulations the importance of different kinds of updating has been appreciated. For instance, R. H. Swendsen and J.-S. Wang, Phys. Rev. Lett. **58**, 86 (1987) studied the differences induced by varied asynchronous updating schemes in Monte Carlo simulations of equilibrium systems. In fact, updating in two parallel clusters bipartite lattice is often done in order to speed up simulations on parallel machines.

[6] J. Hertz, A. Krogh, and R. G. Palmer, *Introduction to the Theory of Neural Computation* (Addison-Wesley, Reading, 1991).

[7] P. Marcq, H. Chaté, and P. Manneville, Phys. Rev. Lett. **77**, 4003 (1996); Phys. Rev. E **55**, 2606 (1997).

[8] J. Rolf, T. Bohr, and M. H. Jensen, Phys. Rev. E **57**, R2503 (1998).

[9] In Ref. [8] the significance of asynchronous updating was argued on the basis of the connection between CML and physical spatially extended systems (which is not straightforward at all). Heuristically, the closest physical parallel is probably a collection of coupled elements each evolving chaotically. In order to derive a discrete local map for the sites, one may apply a Poincaré section. In the absence of an external periodic forcing the time intervals between consecutive crossings of the section will generically not be constant and in fact vary from unit to unit in space. The state of the extended system at fixed time intervals can then be roughly mimicked by using asynchronous updates, in which the elements experience slightly different update times.

[10] S. Sinha and D. Biswas, Phys. Rev. Lett. **71**, 2010 (1993). Certain models in this class display self-organized critical SOC-type behavior.

[11] S. Sinha, Phys. Rev. E **49**, 4832 (1994); Int. J. Mod. Phys. B **9**, 875 (1995).

[12] S. Sinha, Phys. Lett. A **199**, 365 (1995).

[13] P. Bak, C. Tang, and K. Wiesenfeld, Phys. Rev. Lett. **59**, 381 (1987).

[14] *How Nature Works: The Science of Self-Organised Criticality*, edited by Per Bak (Copernicus, New York, 1996); *Self-Organized Criticality: Emergent Complex Behavior in Physical and Biological Systems*, edited by H. J. Jensen (Cambridge University Press, Cambridge, 1998).

[15] In the context of control, such threshold mechanisms have been realized on (continuous valued) voltages in analog circuit experiments. See, for example, K. Murali and S. Sinha, Phys. Rev. E **68**, 016210 (2003).

[16] A. Corral, C. J. Pérez, and A. Díaz-Guilera, Phys. Rev. Lett.

78, 1492 (1997).
[17] K. Chen, P. Bak, and S. P. Obukhov, Phys. Rev. A **43**, 625 (1991); Z. Olami, Hans Jacob S. Feder, and K. Christensen, Phys. Rev. Lett. **68**, 1244 (1992); F. Jansen and S. Hergarten, Phys. Rev. E **73**, 026124 (2006).
[18] H. Nakae, Neurosci. Res. (N. Y.) **21**, 297 (1997); R. Van Rullen and S. J. Thorpe, Neurocomputing **26–27**, 911 (1999); L. Perrinet, M. Samuelides, and S. J. Thorpe, *ibid.* **57**, 125 (2004).
[19] A. Mondal and S. Sinha, Phys. Rev. E **73**, 026215 (2006).
[20] A. Pikovsky, M. Rosenblum, and J. Kurths, *Synchronization: A Universal Concept in Nonlinear Sciences* (Cambridge University Press, Cambridge, 2001), and references therein.