Transient Dynamics and Long-Term Incoherence in Coupled Systems

Vivek Sagar

(MS11017)

A dissertation submitted for the partial fulfillment of

BS-MS dual degree in Science

Indian Institute of Science Education and Research Mohali

(April 2016)

Certificate of Examination

This is to certify that the dissertation titled "Transient Dynamics and Long-Term Incoherence in Coupled Systems " submitted by Mr Vivek Sagar (Reg. No. MS11017) for the partial fulfillment of BS-MS dual degree programme of the Institute, has been examined by the thesis committee duly appointed by the Institute. The committee finds the work done by the candidate satisfactory and recommends that the report be accepted.

Prof. Somdatta Sinha   Prof.SudeshnaSinha   Dr.AbhishekChaudhury
(Supervisor)
Date:

Declaration

The work presented in this dissertation has been carried out by me under the guidance of Prof. Somdatta Sinha at the Indian Institute of Science Education and Research Mohali. This work has not been submitted in part or in full for a degree, a diploma, or a fellowship to any other university or institute. Whenever contributions of others are involved, every effort is made to indicate this clearly, with due acknowledgment of collaborative research and discussions. This thesis is a bona-fide record of original work done by me and all sources listed within have been detailed in the bibliography.

Vivek Sagar
(MS11017)

In my capacity as the supervisor of the candidate's project work, I certify that the above statements by the candidate are true to the best of my knowledge.

Prof. Somdatta Sinha

Acknowledgments

List of Figures

Contents

Abstract

The aim of this study is to quantify and characterize the dynamic heterogeneity (transient and incoherent patterns) in spatially-extended systems. Coupled map lattices of the logistic, exponential and the host parasite map are used as model systems and the variation of transient and incoherent phenomena with respect to different nonlinear map functions and the model parameters is studied. A few quantities based on statistical measures and image analysis of the transient states in the CML is developed to characterize their occurrence for each map. We have formulated a new quantitative measure to characterize the long term spatiotemporal incoherent states, that can differentiate between the Chimeric states and the other types of heterogeneity. We show that the transient and long term spatiotemporal dynamics in a coupled system depends on the intrinsic nonlinear function, coupling strength, the number of neighbors, and initial conditions. We also observed that the spatial extent of the transient states follow a power law and incoherent states like Chimera patterns are much more frequent in the lattice of exponential maps than that of the logistic maps.

Chapter 1

Introduction

1.1 Dynamical Systems in Biology

Tools from dynamical systems are increasingly being used in modeling and analyzing biological phenomenon, e.g. population dynamics, pattern formation in tissues and activity of single neurons or their network. Stability and coherence in the long term behavior is generally expected to be functionally useful in spatially-extended dynamical systems. The long-term behavior of such systems has been the focus of several studies [1]. These models often exhibit a vast range of dynamical behavior spanning from regular spatial and/or temporal patterns to irregular/chaotic ones. An example of a periodic pattern in space and time is the propagation of spiral waves in cardiac tissues during ventricular fibrillation [2]. However, in recent times, the evolution, persistence and characterization of dynamic heterogeneity (transient or incoherent patterns) are finding importance both in physical and biological systems. These have been implicated to underlie many events in Biology, and involved in several normal and pathological phenomena. For example, several mammals exhibit unihemispheric slow wave sleep, that can be linked to spatial incoherence [3]. Sharp waves and ripples, sleep spindles and K-complexes are transient phenomenon that are identifiable through electroencephalography [4]. Dynamic heterogeneity is also possible in social systems [5]. In this study I intend to explore the transient states and the formation of long term coexistence of spatially coherent and incoherent domains (chimera-like states) in coupled systems, The aim of this study is to quantify and characterize these patterns and study their correlation with varying local and non-local coupling. Coupled Map Lattices of logistic, exponential or host-parasite maps have been used as model systems for this study.

1.2 Definitions

Various definitions of **synchronization** have been proposed in the literature. **Total synchronization** of two dynamical variables implies having identical time-series. **Phase syn-**

chronization means overlapping crests and troughs in the time series, although their amplitudes may differ. **Phase lagging** means having a constant phase difference between the two systems. **Frequency locking** is defined as having a time-invariant functional relationship between the frequencies of the two systems. Absence of any form of relationship between the frequency spectrum of the variables is called **asynchronization.** The Venn diagram, shown in Figure 1.1 describes various forms of synchronization [6].

Transience is a brief change in the state of a system. For a state variable of a dynamical system, (temporal) transient state is a brief interval (compact subset) in its time-series during which it exhibits a qualitatively different dynamical behavior from that before and after the interval, e.g. ripples in still water, sleep spindles in EEG rhythms (shown in Figure 1.2), solitary Ca2+ wave during fertilization of an egg, new gene expression due to transient increase in morphogen [7] etc. **Long term incoherence**, for a spatially-extended dynamical system, indicates states of sustained non-uniform synchronization. Example are the Chimera states [8] in which a system of identical non-locally coupled oscillators is split into coexisting regions of phase/frequency locked and drifting oscillation. These patterns have also been shown to exist in experiments [9]. These states are shown to be extremely sensitive to initial conditions.
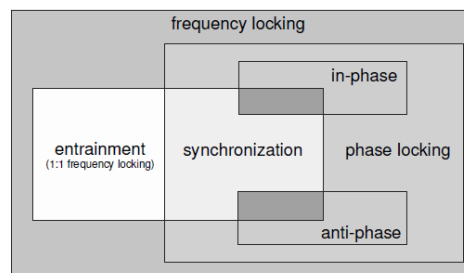
Figure 1.1: *Various forms of synchronization. The box representing synchronization implies total synchronization* [6].
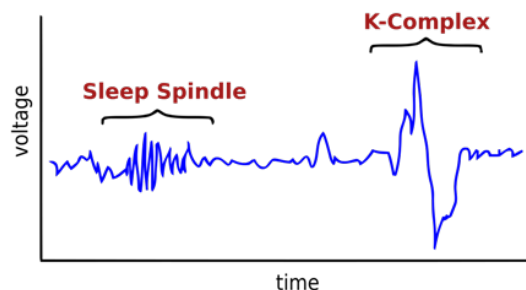
Figure 1.2: *Transient states in an EEG pattern during slow wave sleep* [10].

1.3   Coupled Map Lattice as Model System

Spatially extended systems can be broadly classified into the following four categories **[11]** depending upon whether the underlying spatial organization, temporal evolution or state variable is discrete or continuous.

1. A **cellular automaton (CA)** consists of an n-dimensional grid/lattice of cells, each in one of the possible countable number of states (e.g. binary). Given appropriate initial conditions, the system is evolved at discrete time-steps according to some time-invariant mathematical rule.

2. A **coupled map lattice (CMLs),** like CAs possess discrete underlying spatial structure and temporal dynamics. However, unlike the former, the state variables in a CML are continuous. Each of the lattice sites in an uncoupled CML is governed by a system of difference equations or maps.

3. **System of ODEs** consists of an array of ordinary differential equations, thus useful for modeling systems with discrete spatial organization but continuous system variables and temporal dynamics.

4. **Partial differential Equations** are used for modeling phenomenon with continuous space, time and state variables.

In this study, the dynamic heterogeneity exhibited in the coupled map lattice models of three biologically relevant dynamical systems (logistic, exponential and host-parasite maps) is characterized.

1.4   Organization of the Thesis

To characterize the different forms of dynamic heterogeneity, as mentioned above, and their dependence on nonlinear functions, parameters and initial conditions, I have studied three different discrete models systems, namely, the Logistic and Exponential (Ricker) 1-dimensional maps, and the Host-Parasite 2-dimensional map, in the Coupled Map Lattice systems with local and non-local coupling.

In Chapter 2, the single maps (1-d: Logistic and Exponential, and 2-d: Host-Parasite) are stated and the corresponding spatiotemporal dynamical systems (coupled map lattices) are described. The new measures developed for the characterization of the transient and long term dynamic heterogeneity are stated here.

In Chapter 3, all results for analyzing the transient and long term heterogeneous states are discussed.

In Chapter 4, the major conclusions and future directions are discussed, and the literature quoted are given in the References chapter.

Chapter 2

Models and Methods

2.1 Maps

Consider the difference equation $x_{n+1} = f(x_n)$ where $n \in \mathbb{N}$, and $x_n \in \mathbb{R}^m$ and f is suitably well behaved function. Such an equation/function is called a map or a difference equation. Maps appear naturally in various ways [12]:

1. As tools for analyzing differential equations.

2. As models of natural phenomena.

3. As simple examples of chaos.

The Logistic and Exponential Maps are used to model single-species discrete population dynamics in ecology in the presence of limited resources. The host-parasite model is used for modeling the growth of two interacting populations in which one of the populations grows at the expense of the other.

2.1.1 1D Maps

The logistic map was popularized in 1976 by Robert May [13] and is of the following form:

$$x_{n+1} = f(x_n) = rx_n(1 - x_n) \tag{2.1}$$

The map is bounded between 0 and 1, while $0 < r < 4$. It is a unimodular map that undergoes pitchfork bifurcations as r increases and follows a period doubling route to chaos at $r = 3.57$. Beyond this, the map shows chaos interspersed with patterns with periodicity of ...7, 5, 3, that ultimately cumulates the unstable cycles of all periods through an "intermittency route to chaos" at $r = 4$.

The exponential or the Ricker model [14] has the form:

$$x_{n+1} = f(x_n) = x_n e^{r(1-x_n)} \tag{2.2}$$

Like the logistic map, the exponential map is also unimodal and follows a period-doubling route to chaos with periodic windows. Unlike the logistic map, the exponential map is asymmetric around the maxima, has a point of inflection between 0.5 and 1, and has a domain that is not bounded above. Their bifurcation diagrams are shown below.
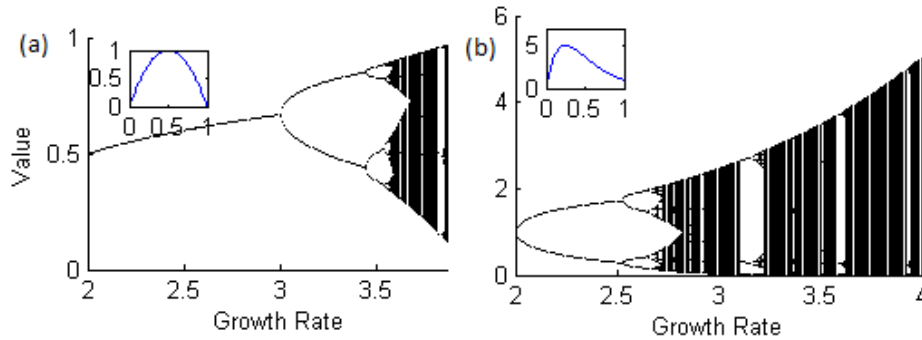
Figure 2.1: *(a) Bifurcation diagram of logistic map; (b) Bifurcation diagram of exponential map. The insets show the nonlinear functions (Return Maps).*

2.1.2 2D Map

Host-Parasite map **[15]** can be used to understand the dynamics of two interacting species in which an adult female parasite searches for a host on which to oviposit (deposit its eggs). In some cases eggs are attached to the outer surface of the host during its larval or pupal stage. In other cases the eggs are injected into the host's flesh. The larval parasites develop and grow at the expense of their host, consuming it and eventually killing it before they pupate.

Assumptions:

1. Hosts that have been parasitized will give rise to the next generation of parasites.

2. Hosts that have not been parasitized will give rise to their own progeny.

3. The fraction of hosts that are parasitized depends on the rate of encounter of the two species.

4. Encounters occur randomly. The number of encounters of hosts by parasites is therefore proportional to the product of their densities.

5. Only the first encounter between a host and a parasite is significant. (Thus the interaction between the host and parasite can be modeled as a homogeneous Poisson process)

6. In the absence of parasites, the host population grows to some limited density (determined by the carrying capacity of its environment).

Depending upon the growth function of the host population used, the HP system can take up one of the following two forms:

Logistic Host-Parasite Model

$$H_{n+1} = f\left(H_n,\ P\right) = rH_n(1 - H_n)e^{-bP_n}$$
$$P_{n+1} = g\left(H_n,\ P_n\right) = H_n(1 - e^{-bP_n})$$

$$(2.3)$$

Where P_n is the size of the parasite population, H_n is the size of the host population, r is the growth rate of the host population and b is the searching efficiency of the parasite population.

Exponential Host-Parasite Model

$$H_{n+1} = h\left(H_n,\ P\right) = H_n e^{r(1-H_n)-bP_n}$$
$$P_{n+1} = p\left(H_n,\ P_n\right) = H_n(1 - e^{-bP_n})$$

$$(2.4)$$

All the variables are same as in (3). Despite the fact that the logistic and exponential map belong to the same universality class, the dynamics of the two systems can be quite different as shown in the figure below (the choice of the growth rate is explained in the next section).
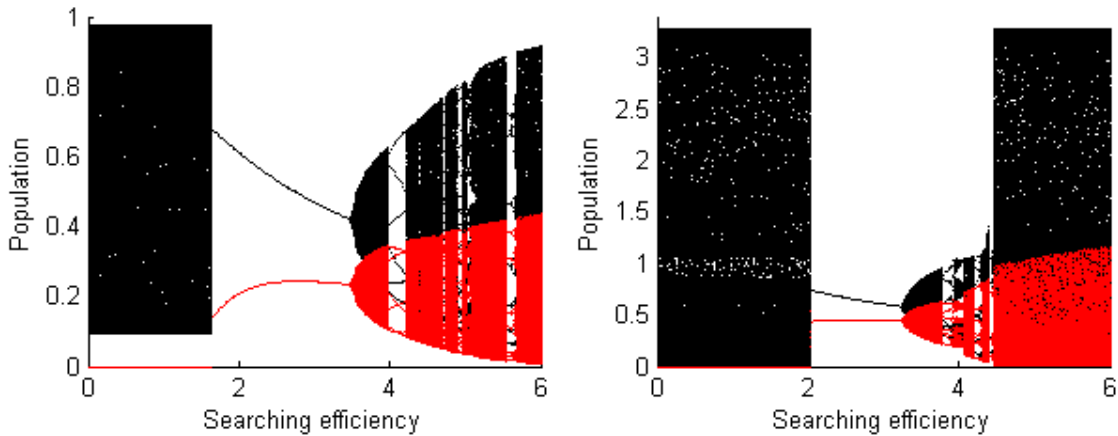
Figure 2.2: *(a) Bifurcation diagram of logistic HP map for r = 3.9; (b) Bifurcation diagram of exponential HP map for r = 3.42.*

2.2   Coupled Map Lattice

A Coupled Map Lattices (CML) is a dynamical system that has discrete time dynamics, discrete underlying spaces (lattice/networks) and real, continuous and local state variables. Here, in each lattice site the real and continuous state variable x associated with each map function evolves at discrete time steps. A CML of logistic/exponential maps can be used to

model meta-population dynamics of different segregated sub-populations coupled through migration. Evolution of population (x) at each lattice site (s) coupled to each other through inter-site migration happens in two steps as follows: First, each of the lattice sites evolves as per recursive map: $x_{n,s} \rightarrow f(x_{n,s}) = x_{n+1,s}$ (growth phase). Second, a constant fraction ϵ of this state migrate/ diffuse into the adjacent sites with zero delay (Migration phase). For a 1D lattice, the quantity $d = K/N$ is called the range of the coupling, where N is the total number of sites and K is the number of neighbors on one side. When $K = 1$, the coupling is "local" or to the nearest neighbors. For, $K > 1$, the coupling is known as "non-local". The underlying lattice can be of any dimension m in which case the range may be an $m-$ dimensional vector. In all work given here, periodic boundary condition is used for the 1-dimensional CML of size (N) 50 and 100 nodes. Thus, the evolution of the CML can be described as [16]:

$$x_{n+1,s} = (1-\epsilon)f(x_{n,s}) + \frac{\epsilon}{2K} \sum_{k}^{K} (f(x_{n,s-k}) + f(x_{n,s+k})) \tag{2.5}$$

For a 2D map, $x_{n,s} = (H_{n,s}, P_{n,s})$
$f(x_{n,s}) = F(H_{n,s}, P_{n,s}) = (h(H_{n,s}, P_{n,s}), p(H_{n,s}, P_{n,s}))$.
Several sets of random initial conditions were chosen to study the dynamic heterogeneity in CMLs.

In this study, a coupled lattice of logistic/exponential maps is called LCML/ECML and that of the logistic/exponential HP maps is called LHP-CML and EHP-CML respectively.
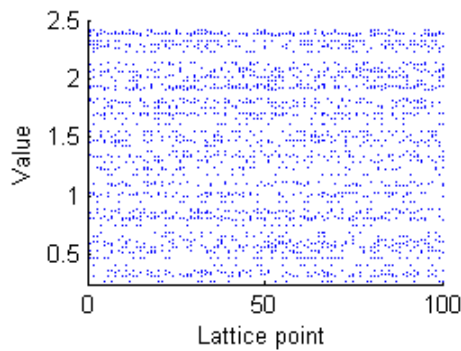
Figure 2.3: *Overlapping plots of 25 random initial conditions as a function of lattice site for the lattice of exponential maps.*

These coupled systems are capable of exhibiting a vast range of heterogeneous spatiotemporal patterns. The following space-time amplitudes show transient and incoherent states in the ECML.
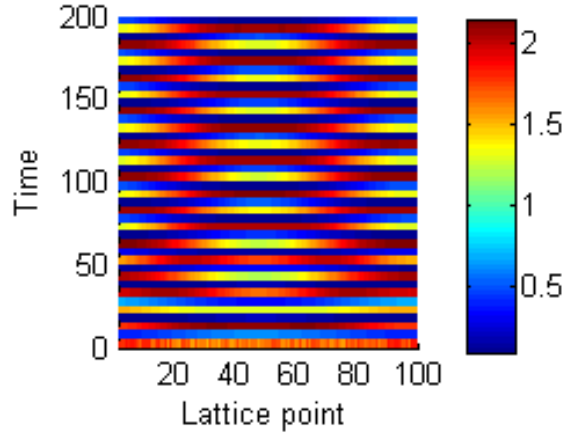
Figure 2.4: *Transient behavior in the ECML*
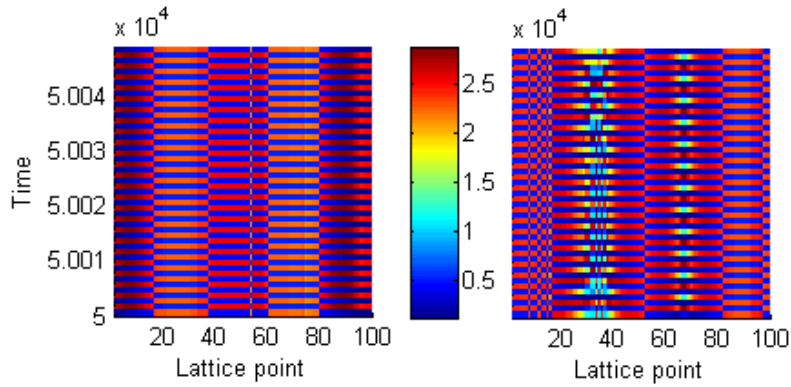
Figure 2.5: *Incoherent behavior in the ECML (left) Non-Chimera; (right) Chimera*

2.3 Characterization of Dynamic Heterogeneity

In order to completely characterize and quantify these heterogeneous states in space and time, the following measures were used.

2.3.1 Synchronization order parameter (R)

The synchronization order parameter **[17]** is a real number bounded between 0 and 1 that quantifies the degree of synchronization across the lattice sites during a time window. Mathematically,

$$R = \frac{\langle M_n^2 \rangle - \langle M_n \rangle^2}{[\langle x_{n,s}^2 \rangle - \langle x_{n,s} \rangle^2]}$$

where M_n is the spatial average at every time n, $[\,]$ denotes the average across the space while $\langle\,\rangle$ denotes the average across time. The idea behind this is that during total synchro-

nization, all the time series $\{x_{n,s}\}$ approach the average behavior $[x_{n,s}]$ of the ensemble. The variance in the time series of a single lattice site will therefore be equal to that of the time series of the average (across space). Therefore, $R = 1$ for total synchronization. During complete desynchronization, it is likely that for every possible value of the lattice sites x, there is a site s and some small distance δ such that $|x - x_{n,s}| < \delta$. In other words, roughly every possible value is taken by some lattice site or the other. In this case, the average value $[x_{n,s}]$ will be roughly constant and $R = 0$.

2.3.2 Lyapunov exponent (LE)

The Lyapunov Exponent **[18]** characterizes the rate of separation of infinitesimally close trajectories. Quantitatively, two trajectories in phase space with initial separation δZ_0 diverge (provided that the divergence can be treated within the linearized approximation) at a rate given by $e^{\lambda t}\delta Z_0$. Here λ is called the Lyapunov exponent. For a 1D map $f(x)$, LE can be calculated as:

$$\lambda(x_0) = \lim_{N\to\infty}\frac{1}{N}\sum_{n=0}^{N}\log|f'(x_n)| \tag{2.6}$$

For an n-dimensional map, there are n lyapunov exponents, and a system can exhibit chaos if and only if at least one of the lyapunov exponents is positive. Analytically, the vector of the lyapunov exponents $\Lambda = \{\lambda_i\}$ is calculated from the following matrix V :

$$V = \lim_{N\to\infty}(J^N \cdot (J^\dagger)^N)^{\frac{1}{2N}} \tag{2.7}$$

Where J^N is the jacobian matrix of the Nth iterate of the map. λ_i are the log of eigenvalues of this matrix. The actual computation of LE even for $2D$ is not straightforward. I have used *pull-back* **[19]** algorithm for the computation of LE in 2D. In this study, LE is used to compare the equivalent rate of divergence in LCML vs ECML as well as LHP-CML vs EHP-CML.

2.3.3 Sparsity/Coefficient of Variation (S/CV)

Coefficient of Variation (CV) or Sparsity (S) of the histogram of the moving window across the time-series of each lattice site is computed to characterize these transient states. The basic idea behind this measure is that for a chaotic time-series the histogram of the trace of values is almost constant. These plots can be used to estimate:

1. Number of transient states in a given time window.

2. Spatial extent of transient state (width of the smallest rectangle that circumscribe the transient patch).

3. Duration of the transient state (Area of the Sparsity patch/length).

These quantities are called transience estimates. There are several advantages of using sparsity/CV plots to characterize the transients states: CV plots do not involve averaging over the spatial variables and hence it is more accurate for making coherence estimates. For example, these plots can easily distinguish between the chimera and non-chimera patterns. It is also easy to estimate the periodicity of the lattice sites using these plots. However, one drawback of these plots is that they do not take the temporal order into account and therefore, fail to distinguish chaos from a time-series with very high periodicity.
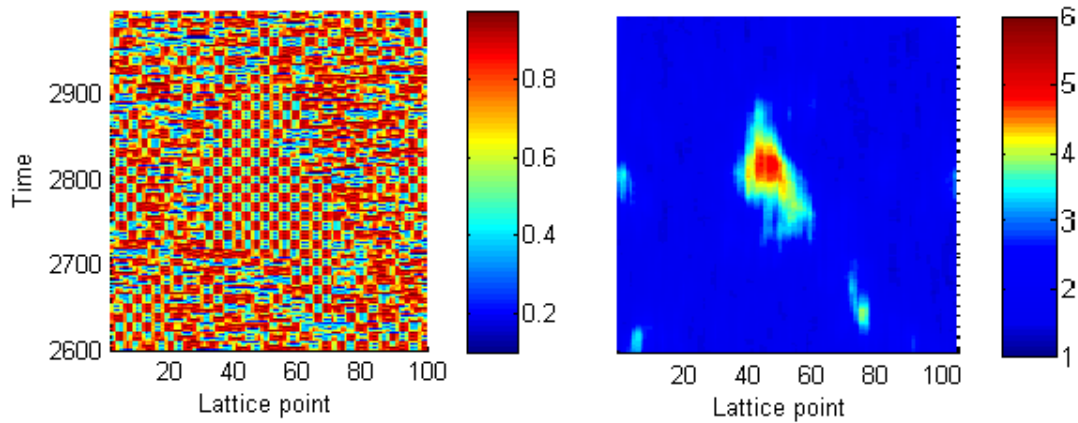
Figure 2.6: *Transient-Incoherent state in a LCML (a) Space-Time-Amplitude Map (b) Corresponding CV plot*
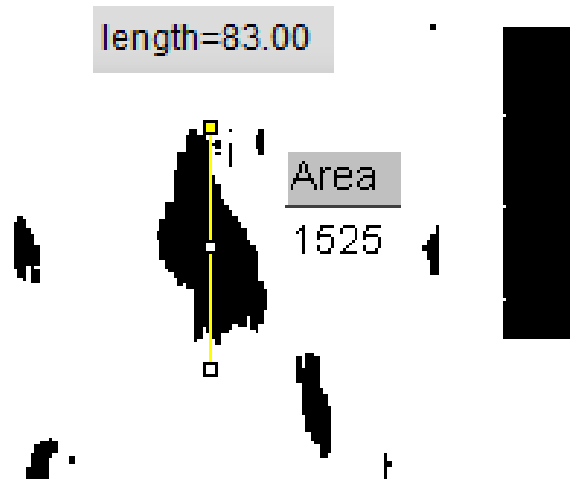
Figure 2.7: *Quantifying length, area and the number of transient patches from fig2.6(b)*

2.3.4   Local Coherence (L)

Local Coherence **[20]** is calculated as the absolute value of the moving average of the unit vectors with phases (arcsine of the normalized values ϕ_s) of the lattice sites. This depends upon the averaging window $N \cdot d$, and is hence computed in the thermodynamic limit.

$$L_s = \lim_{N \to \infty} \frac{1}{2dN} | \sum_{|j-k|<dN} e^{i\phi_s} | \tag{2.8}$$
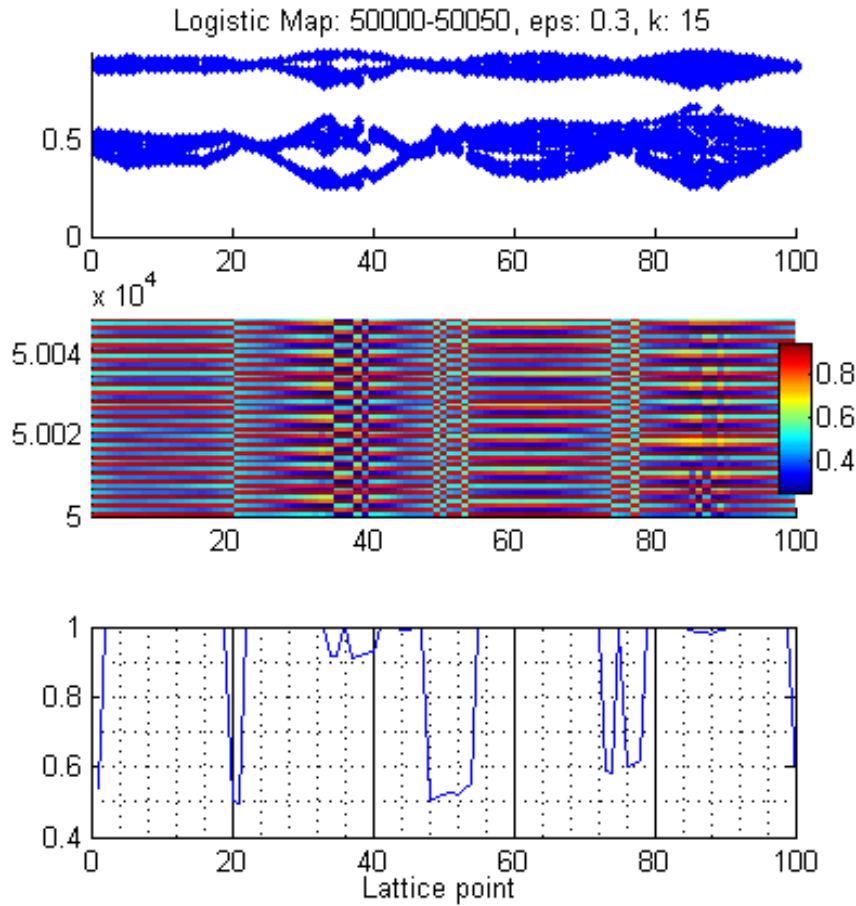
Figure 2.8: *(Top) Overlapping snapshots of the system for 50 times steps in the steady state. (Middle) Space time amplitude for those time-steps. (Bottom) Local Coherence for weak, non local coupling in the LCML (with r = 3.9).*

2.4   Statistical Analysis

All statistical analyses (Wilcoxon rank sum test, Regression analysis and Box plots) were performed in MATLAB [21]. In order to compare the median of the sample of the transience estimates from different initial conditions across different coupling strengths,

Wilcoxon rank sum test was used. Wilcoxon rank sum test is defined as **[21]**: "... a rank sum test of the null hypothesis that data in two vectors x and y are independent samples from identical continuous distributions with equal medians, against the alternative that they do not have equal medians." For each initial condition, least square linear regression of the transience estimates was performed with coupling strengths as the abscissae and the R^2 fit and the best fit slope was calculated. High number of initial conditions with positive slopes and high values of R^2 fit implied an increasing trend.

All the simulations were done on MATLAB R2012b [21] . The image processing was done using ImageJ [22].

Chapter 3

Results

The results are organized to highlight the comparative spatiotemporal collective dynamics of the transient and long term incoherent states in LCML and ECML with 25 different initial conditions. First, the general dependence of the locally coupled (K=1) LCML dynamics on different system parameters are studied to show their role in synchronisation of the collective dynamics. ECML behaviour is similar. Then the propensity and properties of the two locally coupled systems (LCML and ECML) to develop transient states are compared for different initial conditions. The long term incoherence is studied in the non-locally coupled ECML and LCML for a large number of initial conditions. Finally preliminary results of the CMLs with 2-d maps are stated.

3.1 Synchronization

1. The LCML desynchronizes as the growth rate of the logistic map is increased.
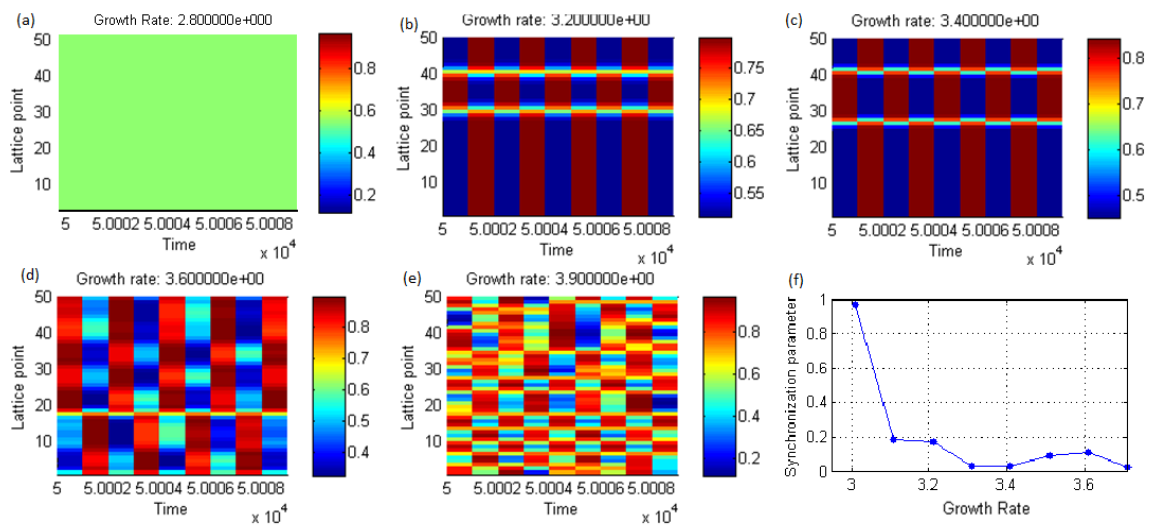
Figure 3.1: *(a)-(e) Space-time amplitude of a lattice of logistic maps with $N = 50$, $\epsilon = 0.5$, $k = 1$. (f) Synchronization order parameter as a function of coupling strength.*

2. The lattice synchronizes as the coupling strength increases at low growth rate (regular dynamics), but chaotic lattices show low synchrony even at higher coupling strengths. Periodic lattices show higher synchrony at low growth rates for different system sizes, but remain asynchronous for higher r at all N, for low coupling strength.
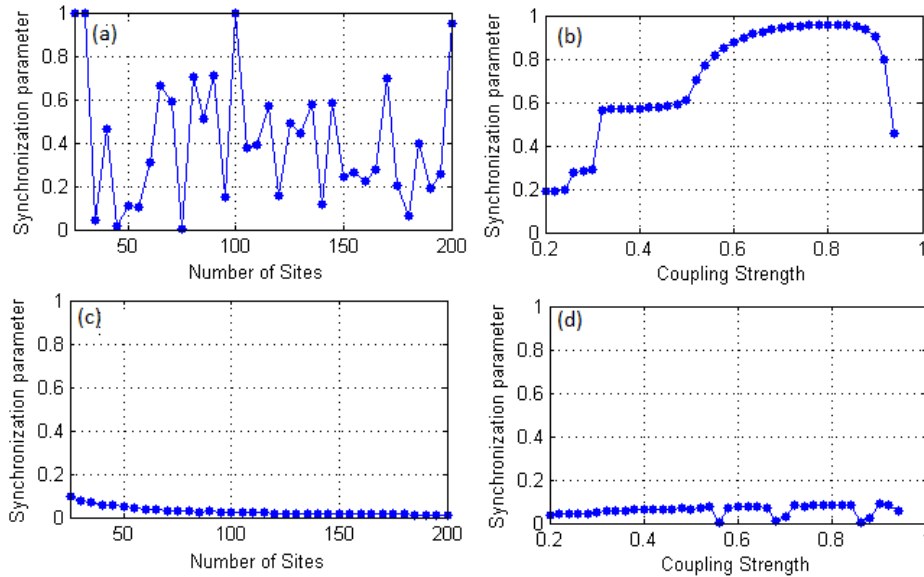
Figure 3.2: *(a) R shows no variation with N for low growth rate (r = 3.2, $\epsilon = 0.3$). (b) R increases with coupling for r = 3.2, N=50. (c) R decreases with N, for r = 3.9 $\epsilon = 0.3$ (d) R weakly increases with coupling for r = 3.9, N = 50. Notice the small dip at the end of plots (b): Higher coupling may not imply high synchronization.*

3.2   Transient Dynamics

For comparing transient dynamics in the Logistic and Exponential CMLs, local chaotic dynamics with the same Lyapunov Exponents are considered (Figure 3.3). As mentioned in section 2.3; three quantities are computed from the CV/Sparsity plots to characterize the transient states that are collectively called transience estimates.

1. Number of transient states in a given time window.

2. Spatial extent of transient state (width of the smallest rectangle that circumscribe the transient patch).

3. Duration of the transient state (Area of the Sparsity patch/length).

3. Transient patterns are more commonly present in LCML in contrast to ECML.
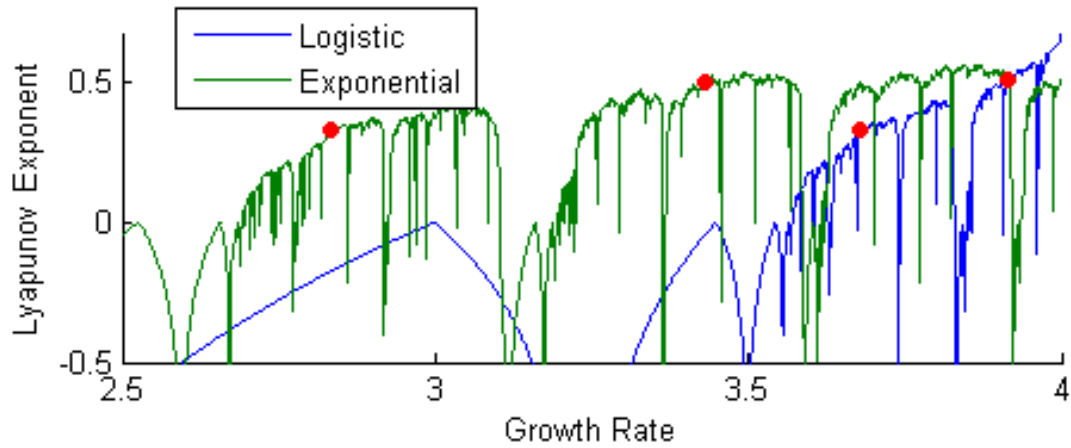
Figure 3.3: *Red dots at 3.42 (exponential map) and 3.9 (logistic map) imply strong chaos.*
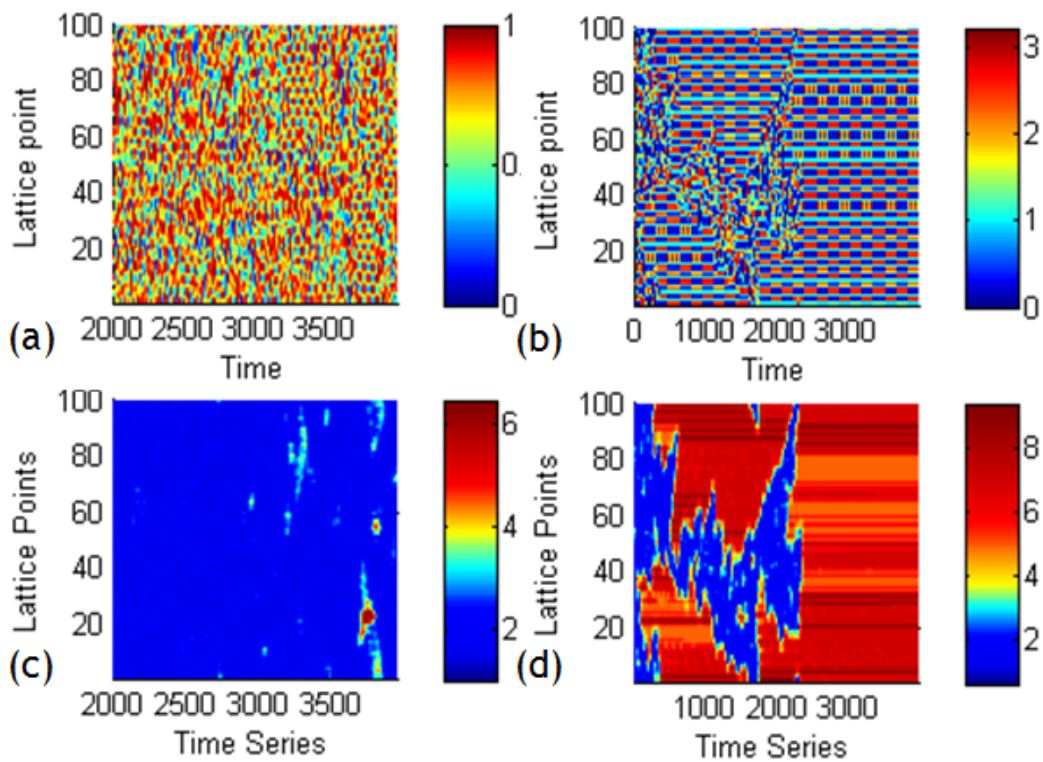
Figure 3.4: *Space-time amplitudes and CV plots of transient states in – LCML (a) & (c)* $(r = 3.9, \ k = 1, \ \epsilon = 0.4)$: *ECML; (b) & (d)* : $(r = 3.42, \ k = 1, \ \epsilon = 0.4)$.

4. The number of transient states, spatial extent and the area of the transient patches increase with the coupling strength in LCML.

The number of transient states and area and spatial extent (length) of the largest patch during 2000-5000 time steps of the locally coupled LCML is studied as a function of coupling strength ϵ for 25 different initial conditions. The standard deviation of these quantities stabilizes after 25 samples (Figure 3.5). For each quantity (area of the largest patch, number of transient states and spatial extent) the analysis is done for 25 initial conditions with increasing coupling strengths. The variation is statistically fitted using linear correlation analysis and the goodness of fit R^2 is calculated. Of the 25 best fit lines, the ones showing positive slopes with $R^2 > 0.6$ indicate increasing trends in the transient state parameters with coupling strength (results shown in Figure 3.6).
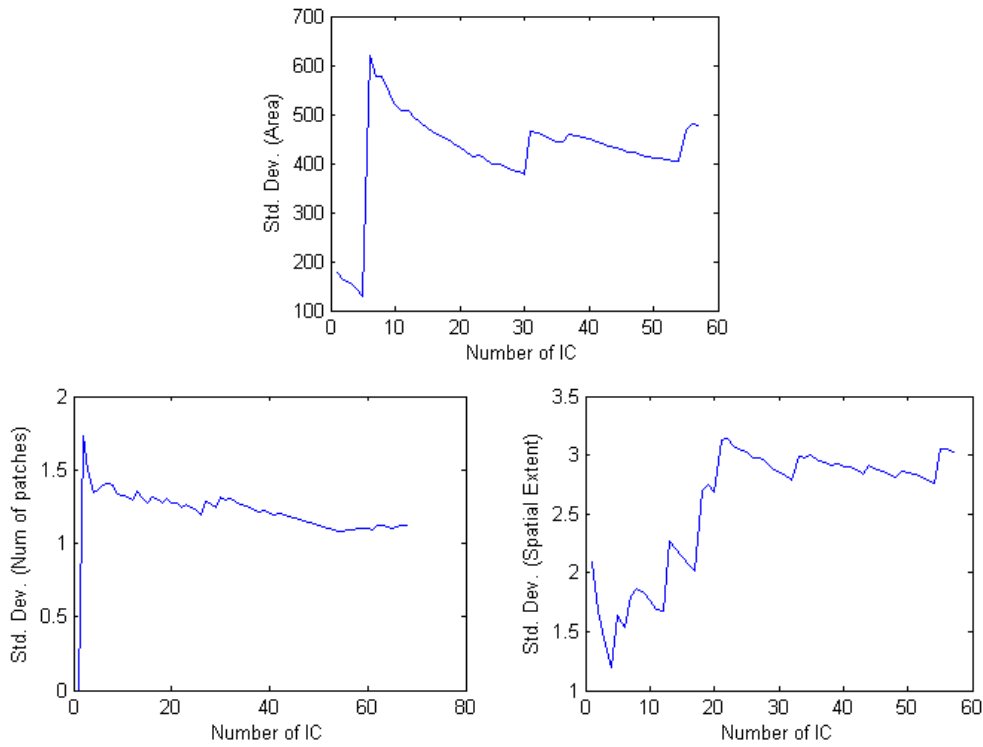
Figure 3.5: Standard deviation in transience estimates as a function of number of initial conditions.
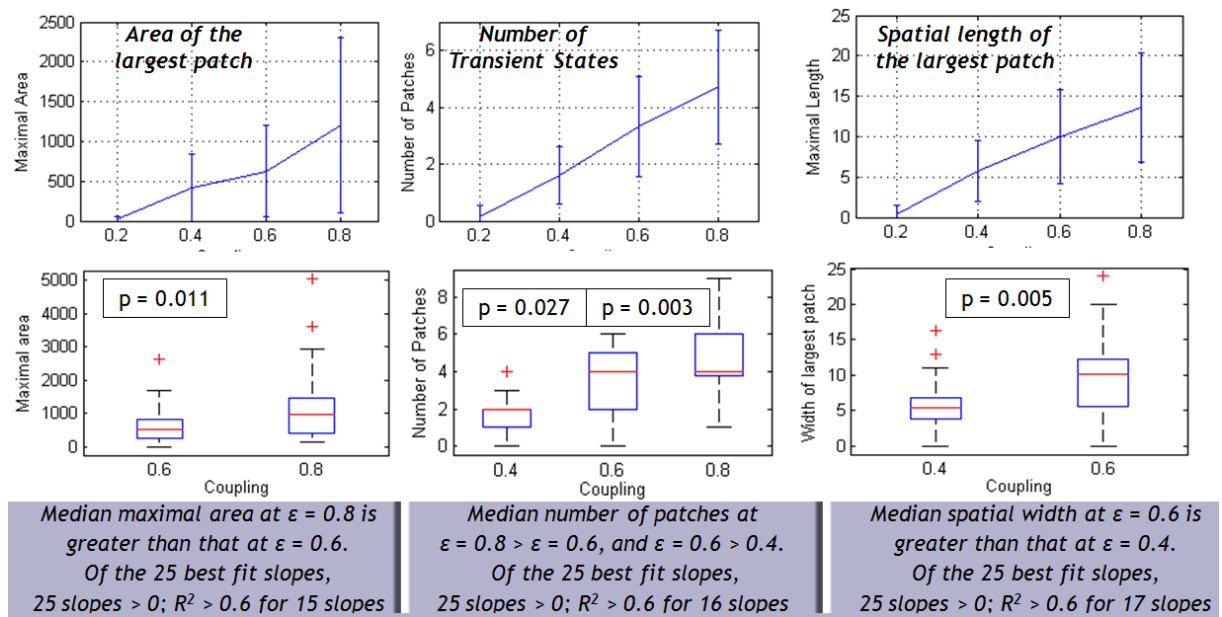
Figure 3.6: *Increase of the area of the largest patch, the number of transient states and the spatial extent with coupling strength.*

5. The spatial width of largest transient patch follows a power law distribution with a scaling of 2.67.
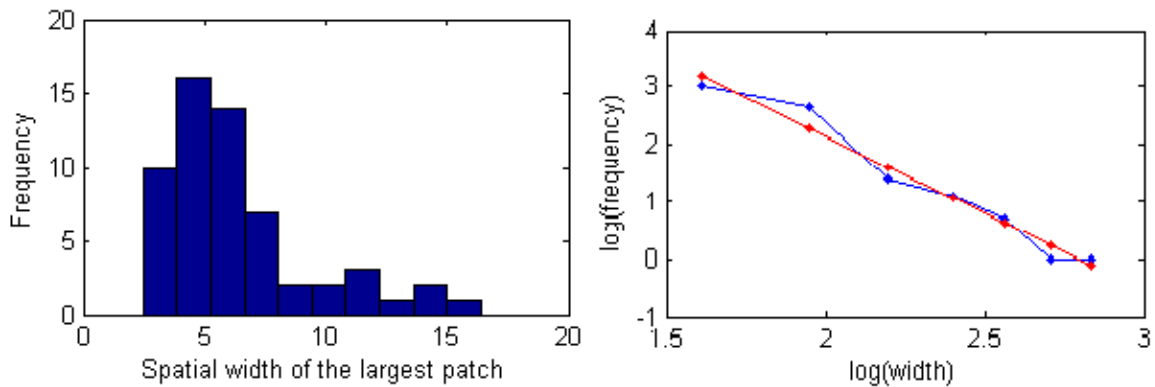
Figure 3.7: *(left) Histogram of spatial width of the largest patch for 75 random initial conditions for coupling strength ($\varepsilon = 0.4$); (right) Log-log plot with R^2 fit of 0.98 (Red-line). Scaling coeff. is 2.67.*

3.3   Long-Term Incoherence

Long term incoherence is formed in both LCML and ECML.

   6. Chimera-like states are far more frequent in ECML than in the LCML. Studies
[20] have shown the existence of different chimera-like patterns (as in fig 3.8 (left) and
3.9 (left)), but since the incoherent regions in such patterns are phase locked, they are not
considered as chimera states here.   Here a chimera state is the one which involves both
incoherence and phase drift (chaos), have a sparsity value > 0.8, and local coherence < 0.95.
As examples of different types of long term incoherent patterns, Figures 3.8 and 3.9 show
Chimera and non-Chimera states for the ECML and LCML dynamics for non-local coupling
k=15. For each sub-figure with four panels, the top shows the overlapping plots of the lattice
for 50 consecutive time steps, the second shows the space-time and amplitude map, the third
is the sparsity plot at the 500K time-step while the bottom corresponds to the local coherence
at 500K time-step.
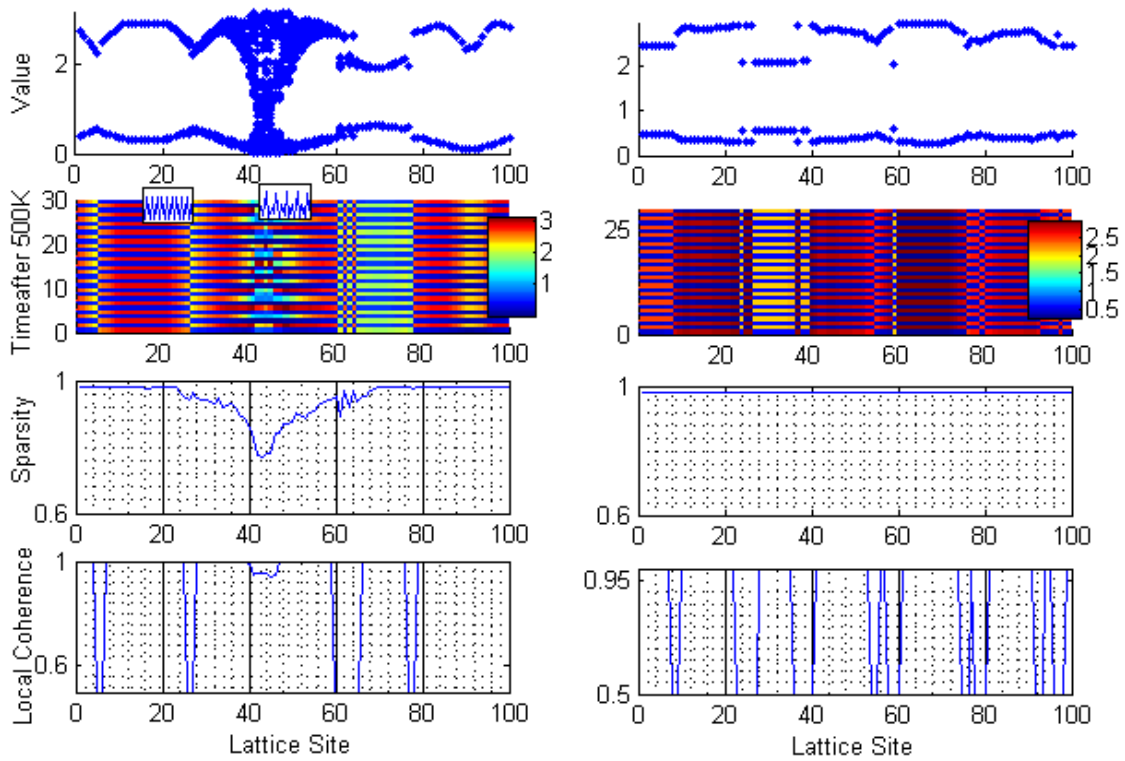
Figure 3.8:   *(left)* **Chimera state***: ECML (r = 3.42, ε = 0.32). 15/20 initial conditions
exhibited chimera states for 0.23 <ε < 0.33; (right) Stable incoherent (non-chimera) pattern
in the ECML (r = 3.42, ε = 0.28) for a different initial condition.*
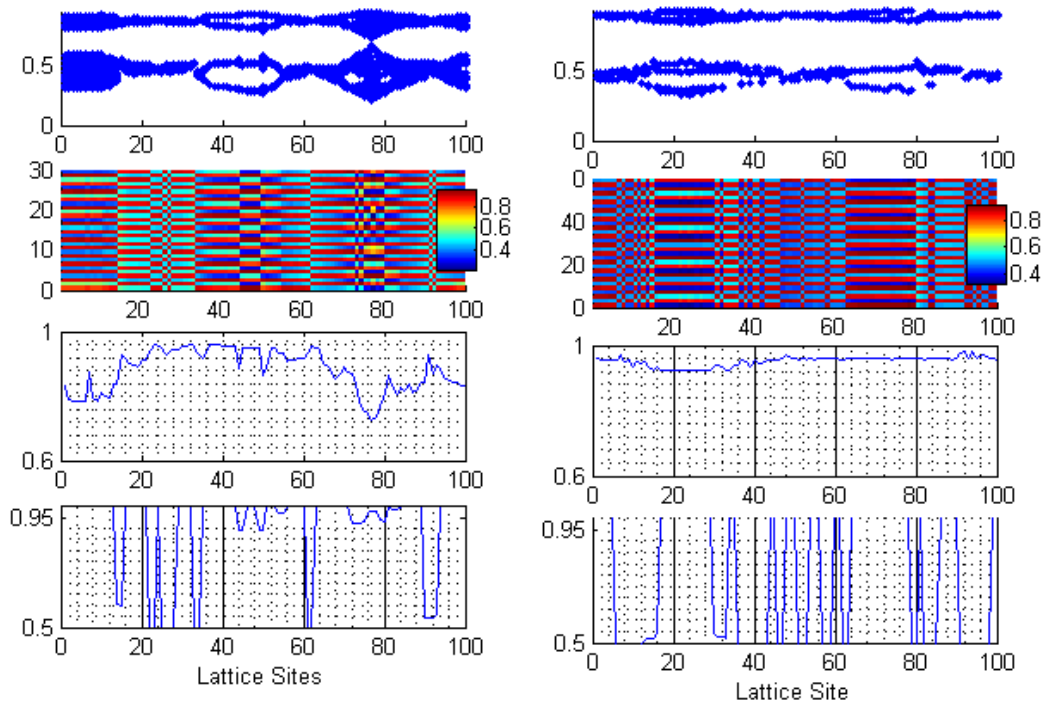
Figure 3.9: *(left)* **Chimera state***: Logistic CML (r = 3.9, ε = 0.29). 3/20 initial conditions exhibited chimera states for 0.23 < ε < 0.33 ; (right) Stable incoherent (non-chimera) pattern in the logistic CML (r = 3.9, ε = 0.28).*

7. Chimera states are stationery in space.
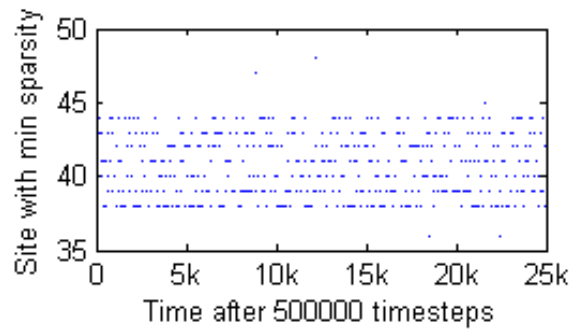
Figure 3.10: *Robustness of the chimera patterns is tested by plotting the site of minimum sparsity in (3.8 a) as a function of time*

8. Chimera states are highly sensitive to initial conditions, coupling strength and coupling range.
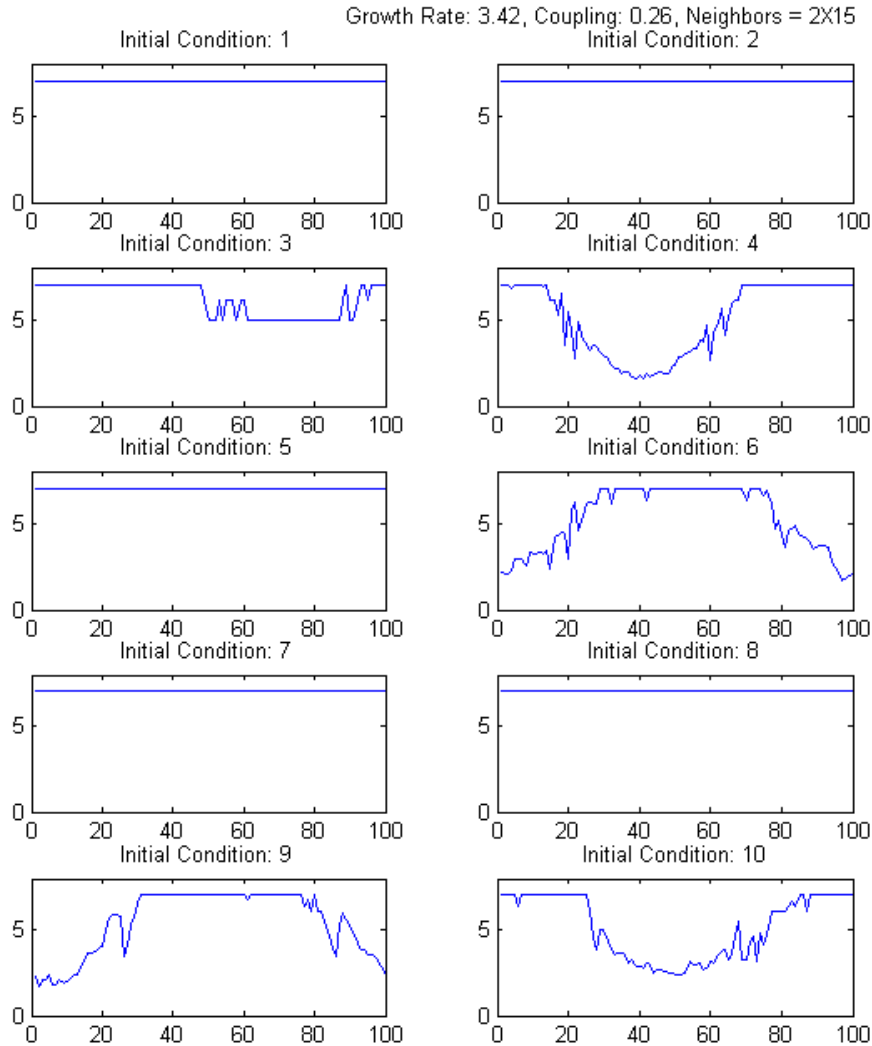
Figure 3.11: *Coefficient of variation (CV) in ECML (N=100) shown for 10 initial conditions for $\varepsilon = 0.26$, $K = 15$. Chimera patterns exist only for 4 Initial conditions as is clearly delineated from the CV value.*

3.4   Host-Parasite CMLs

In order to compare the two types of the HP systems **[23]**, the 2D lyapunov spectrum was computed and the maximum lyapunov exponent was plotted as a function of the searching efficiency and the growth rate. For the same values of the lyapunov exponent, both the lattices show similar dynamics (Fig 3.12). The computed spectrum is not dependent on initial conditions.
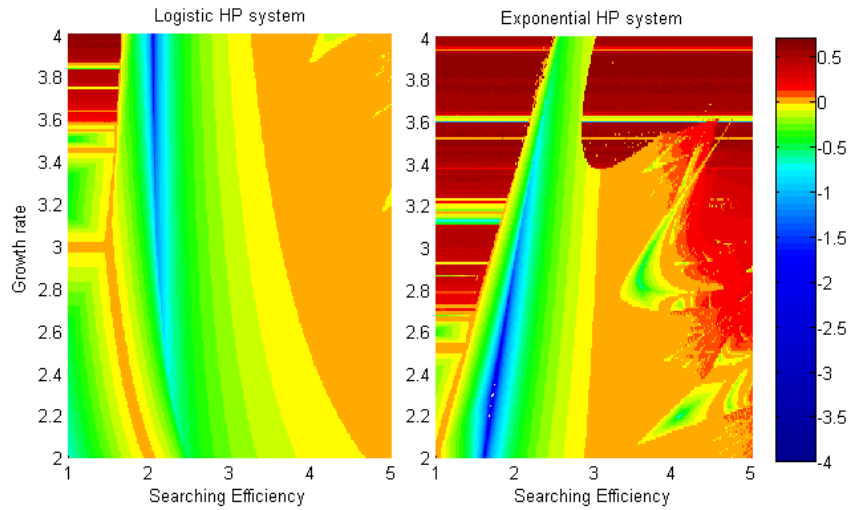
Figure 3.12: *Maximal lyapunov exponent as a function of searching efficiency and growth rate. Red values imply strong chaos.*
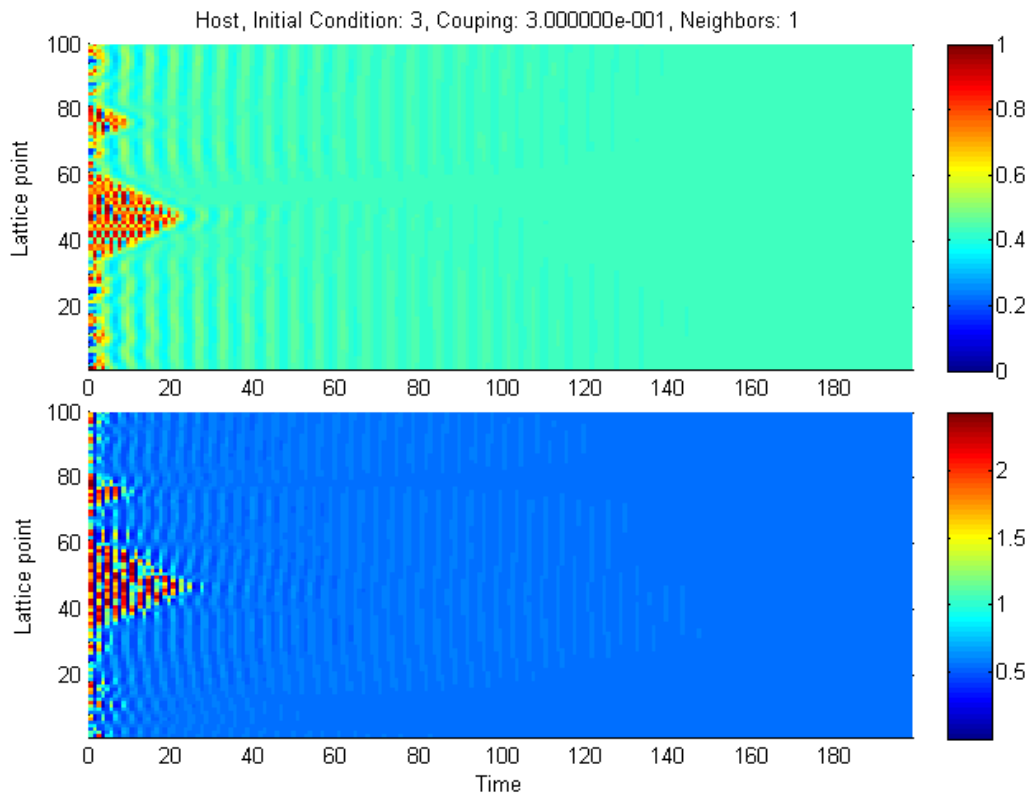
Figure 3.13: *(Top) Host population in the LHP-CML with LE -0.08 ($r = 3.9, b = 3, \epsilon = 0.3$). (Bottom) Host population in the EHP-CML with the same LE ($r = 3.3, b = 3, \epsilon = 0.3$).*

9. However, only the EHP-CML is able to give rise to heterogeneous patterns for $b > 2$ **as shown in the figure below.**
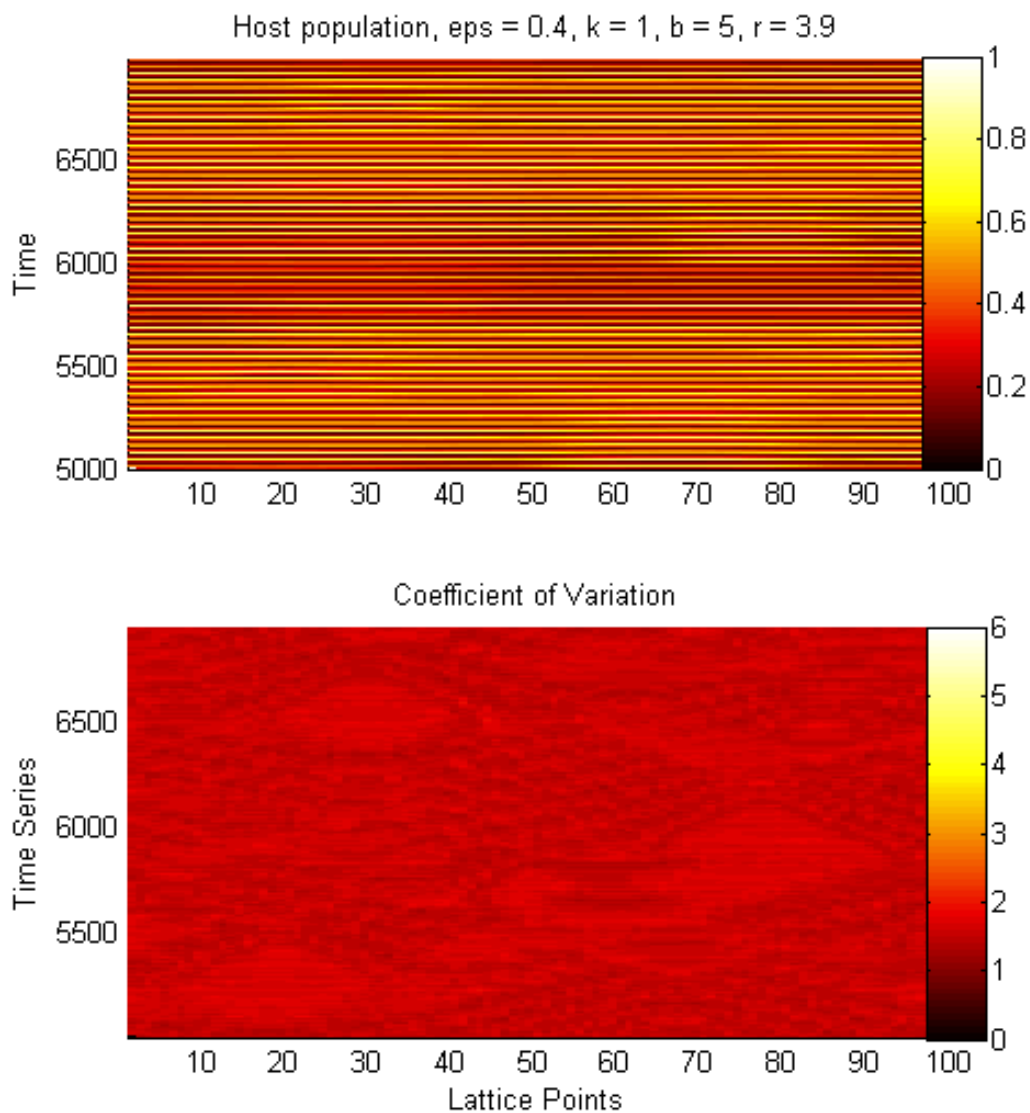
Figure 3.14: *(Top) Host population in the LHP-CML with* $(r = 3.9, \; b = 5, \; \epsilon = 0.3)$. *(Bottom) Coefficient of variation plot showing no transience.*
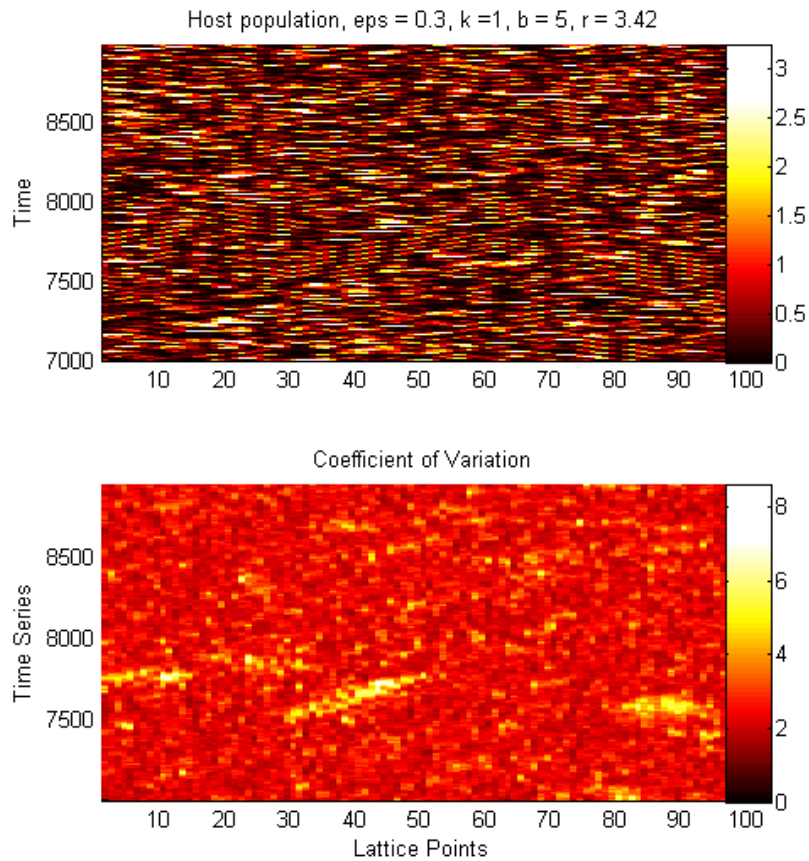
Figure 3.15: *(Top) Host population in the EHP-CML with* $(r = 3.42, b = 5, \epsilon = 0.3)$. *(Bottom) Coefficient of variation showing three transient patches.*

10. For a 2D lattice of LHP-CML, if a single parasite is introduced at a location, the speed of infection increases with the coupling strength and range.
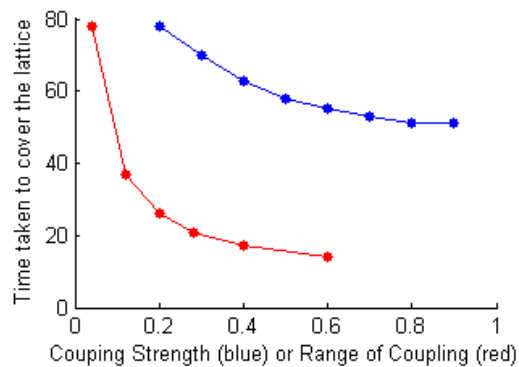
Figure 3.16: *Time taken to cover the 50X50 lattice as a function of* k *and* ϵ *(periodic boundary conditions).*

The work on HP-CMLs is currently in progress.

Chapter 4

Conclusions and Future directions

This study is an attempt to quantify and characterize the dynamic heterogeneity that arise in spatially-extended systems at short and long time scales (transient and long term incoherent patterns) after coupling is introduced amongst the dynamical systems in a Coupled Map Lattice system. Both one-dimensional (Logistic and Exponential models) and two-dimensional (Host-Parasite model) maps are used as dynamical systems and their collective spatiotemporal dynamics studied for a large range of system size, coupling strength, coupling range, and initial conditions.

4.1   Conclusions

1. Transient and long term spatiotemporal dynamics in a coupled system depends on the intrinsic nonlinear function, coupling strength, and the number of neighbors.

2. Similar maps (Logistic & Exponential in chaotic regime with same LE) show different dynamical trends in properties of transients and chimera-like dynamical states.

3. The synchronization increases with decreasing chaos (growth rate) and lattice size and increasing coupling strength.

4. The width, area and number of transient patches increase with coupling strength in the locally coupled logistic CML, and the spatial width of transient states follows a power law.

5. Chimera states are more commonly found in the exponential CMLs than in the logistic CML.

6. Chimera states in this CML are stationary in space and are highly sensitive to coupling strength, range and initial conditions.

7. The lattice of logistic and exponential map based HP systems exhibit qualitatively different dynamics even though the constituent 1D maps belong to the same universality class.

4.2   Future directions

1. The physical interpretation of power law distribution in the spatial extent of the transient states in the locally coupled logistic lattice needs further elaboration for other coupling strengths.

2. The distribution of the transient states in non-locally coupled maps needs to be investigated.

3. It is not clear why the chimera states are more probable in the exponential lattice as compared to the logistic ones - even for the same values of the LE.

4. Some more work is required to investigate the differences in the behavior of Logistic and exponential based HP systems.

5. The stability and dependence of these states under heterogeneous/dynamic coupling and in lattices in higher dimensions need to be investigated.

References

1. Murray, James D. "Mathematical Biology I: An Introduction, vol. 17 of Interdisciplinary Applied Mathematics." (2002)..

2. Panfilov, Alexandre V. "Spiral breakup as a model of ventricular fibrillation." Chaos: An Interdisciplinary Journal of Nonlinear Science 8.1 (1998): 57-64.

3. Ma, Rubao, Jianxiong Wang, and Zonghua Liu. "Robust features of chimera states and the implementation of alternating chimera states." EPL (Europhysics Letters) 91.4 (2010): 40006..

4. Maier, N., et al. "Fast network oscillations in the hippocampus." e-Neuroforum 4.1 (2013): 1-10.

5. González-Avella, Juan C., Mario G. Cosenza, and Maxi San Miguel. "Localized coherence in two interacting populations of social agents." Physica A: Statistical Mechanics and its Applications 399 (2014): 24-30.

6. Izhikevich, Eugene M. Dynamical systems in neuroscience. MIT press, 2007.

7. Whitaker, Michael. "Calcium at fertilization and in early development." Physiological reviews 86.1 (2006): 25-88.

8. Panaggio, Mark J., and Daniel M. Abrams. "Chimera states: coexistence of coherence and incoherence in networks of coupled oscillators." Nonlinearity 28.3 (2015): R67.

9. Hart, Joseph D., et al. "Experimental observation of chimeras in a minimal globally-coupled network." arXiv preprint arXiv:1512.01624 (2015).

10. "Stage2Sleep" by Neocadre, Wikipedia (used under creative commons license).

11. Kaneko, Kunihiko, and Ichiro Tsuda. Complex systems: chaos and beyond: a constructive approach with applications in life sciences. Springer Science & Business Media, 2011.

12. Strogatz, Steven H. Nonlinear dynamics and chaos: with applications to physics, biology, chemistry, and engineering. Westview press, 2014.

13. May R. Simple mathematical models with very complicated dynamics, Nature 261. 459-67 (1976).

14. Ricker, W. E. Stock and Recruitment Journal of the Fisheries Research Board of Canada, 11(5): 559–623 (1954).

15. Leah Edelstein-Keshet, Mathematical Models in Biology, SIAM Classics in Applied Mathematics (2004).

16. N. Parekh, and Somdatta Sinha. "Controlling dynamics in spatially extended systems." Physical Review E 65.3 (2002): 036227.

17. S. Rajesh, Sudeshna Sinha, Somdatta Sinha. Synchronization in coupled cells with activator-inhibitor pathways, Physical Review E 75, 011906 (2007) .

18. M. Cencini, Fabio Cecconi, Angela Vulpiani. World Scientific, Series on Advances in Statistical Mechanics. Chaos From Simple models to complex systems (2010).

19. Michael T., James J. Collins, and Carlo J. De Luca. A practical method for calculating largest Lyapunov exponents from small data sets." Physica D: Nonlinear Phenomena 65.1 (1993): 117-134. (1992)

20. Omelchenko, Iryna, et al. "Loss of coherence in dynamical networks: spatial chaos and chimera states." Physical review letters 106.23 (2011): 234102.

21. MATLAB and Statistics Toolbox Release 2012b, The MathWorks, Inc., Natick, Massachusetts, United States.

22. Schneider, C. A.; Rasband, W. S. & Eliceiri, K. W., "NIH Image to ImageJ: 25 years of image analysis" (2012).

23. B.K.Singh, J.Subba Rao, R. Ramaswamy & Somdatta Sinha. The role of heterogeneity on the spatiotemporal dynamics of host parasite metapopulation. Ecological Modelling 180, 435-443, (2004)

APPENDIX: PROGRAMS

1. Bifurcation diagram of logistic and exponential maps

```
kk = 1;
r2 =  2:0.01:4;
for r = r2;
    x = rand;
    for t=1:5000
        x2(t) = x.*(exp(r.*(1-x)));
%        x2(t) = (r)*x*(1-x);
        x = x2(t);
    end
    y(:,kk)=x2(1:end);
    kk = kk+1;
end
y(1:500,:)=[];
figure('units','normalized','Position',[0.5 0.5 0.19 0.23])
hold on
plot(r2,y, 'k.', 'MarkerSize', 1)
xlabel('Growth Rate')
ylabel('Value')
axes('Position',[.2 .2 .15 .17])
box on
x1  = 0:0.01:1;
% y1 = 4.*x1.*(1-x1);
y1 = x1.*(exp((4).*(1-x1)));
plot(x1,y1)
ylim([0 7])
```

2. Bifurcation diagram of HP system

```
kk = 1;
r = 3.42;
b = 0:0.02:6;
time_steps = 2000;
for b = b
    h = 0.5;
    p = 0.5;
    for t=1:time_steps;
%        h2(t) = (r)*h*(1-h)*exp(-b*p);
        h2(t) = h*(exp(r*(1-h)))*exp(-b*p);
        p2(t) = h*(1-exp(-b*p));
        h = h2(t);
        p = p2(t);
    end
    yh(:,kk)=h2(500:end);
    yp(:,kk)=p2(500:end);
    kk = kk+1;
end
figure('units','normalized','Position',[0.5 0.5 0.19 0.23])
% subplot(2,1,1)
hold on
for ii = 1:length(yh);
```

```matlab
    plot(b,yh(ii,:), 'k.', 'MarkerSize', 4)
end
hold on
for ii = 1:length(yp);
    plot(b,yp(ii,:), 'r.', 'MarkerSize', 4)
end
xlabel('Searching efficiency')
ylabel('Population')
xlim([0 6])
ylim([0 3.4])
```

3. 2D Lyapunov Exponent using pull-back algorithm

```matlab
function[v1 v2] =  LogMap(r,b,x,y)
    v1 = x*r*(1-x)*exp(-b*y);
    v2 = x*(1-exp(-b*y));
end


function[dv1 dv2] = LogMapTangent(r,b,x,y,dx,dy)
    dv1 = r*(1-x)*exp(-b*y)*dx-r*x*exp(-b*y)*dx-b*x*r*(1-x)*exp(-b*y)*dy;
    dv2 = (1-exp(-b*y))*dx+x*dy*b*exp(-b*y);
end


function[maxLCE minLCE] = Log_lyapunov(b,r)
nTransients = 200;
nIterates = 10000;
% Initial condition
xState = 0.5;
yState = 0.3;
% Initial tangent vectors
e1x = 1.0;
e1y = 0.0;
e2x = 0.0;
e2y = 1.0;
for n = 1:nTransients;
    [xState yState] = LogMap(r,b,xState,yState);
    %     Evolve tangent vector for maxLCE
    [e1x e1y] = LogMapTangent(r,b,xState,yState,e1x,e1y);
    %     Normalize the tangent vector's length
    d = sqrt(e1x*e1x + e1y*e1y);
    e1x = e1x / d;
    e1y = e1y / d;
    %     Evolve tangent vector for minLCE
    [e2x e2y] = LogMapTangent(r,b,xState,yState,e2x,e2y);
    %     Pull-back: Remove any e1 component from e2
    dote1e2 = e1x * e2x + e1y * e2y;
    e2x = e2x - dote1e2 * e1x;
    e2y = e2y - dote1e2 * e1y;
    %     Normalize second tangent vector
    d = sqrt(e2x*e2x + e2y*e2y);
    e2x = e2x / d;
    e2y = e2y / d;
end
maxLCE = 0.0;
minLCE = 0.0;
for n = 1:nIterates;
```

```matlab
    [xState yState] = LogMap(r,b,xState,yState);
    %       Evolve tangent vector for maxLCE
    [e1x e1y] = LogMapTangent(r,b,xState,yState,e1x,e1y);
    %       Normalize the tangent vector's length
    d = sqrt(e1x*e1x + e1y*e1y);
    e1x = e1x / d;
    e1y = e1y / d;
    %       Accumulate the stretching factor (tangent vector's length)
    maxLCE = maxLCE + log(d);
    %       Evolve tangent vector for minLCE
    [e2x e2y] = LogMapTangent(r,b,xState,yState,e2x,e2y);
    %       Pull-back: Remove any e1 component from e2
    dote1e2 = e1x * e2x + e1y * e2y;
    e2x = e2x - dote1e2 * e1x;
    e2y = e2y - dote1e2 * e1y;
    %       Normalize second tangent vector
    d = sqrt(e2x*e2x + e2y*e2y);
    e2x = e2x / d;
    e2y = e2y / d;
    %       Accumulate the shrinking factor (tangent vector's length)
    minLCE = minLCE + log(d);
end
maxLCE = maxLCE / nIterates;
minLCE = minLCE / nIterates;
end
```

4. LCML/ECML

For non-local coupling

```matlab
function [run_sum] = runningsum(vector, window)
% Moving sum of a column vector (periodic conditions) with a given window >1.
if nargin <2
    window = 3; % Local-coupling
end
half_length = 1/2*(window-1); % Choose the window as an odd number
vec = ones(1, window);
long_vector = [vector(end-half_length+1:end); vector; vector(1:half_length)];
run_sum = conv(long_vector, vec);
run_sum = run_sum(window:window+length(vector)-1);
run_sum = run_sum-vector;
end
```

Sparsity/Coeff. of Variation

```matlab
function [rstdv] = coeff_var(Theta, t_plot, win_spar)
if nargin < 3
    win_spar = 50;
end
% Theta = vec;
% win_spar = 50;
% t_plot = 1000;
Theta(:, 1:end-t_plot)=[];
spar_y = size(Theta,2)-win_spar+1;
rstdv = zeros(size(Theta,1),spar_y);
```

```matlab
for ii=1:size(Theta,1)
%     edge = (0:0.01:max(Theta(ii,:)))';
    edge = linspace(0,max(Theta(ii,:)),100);
    for jj = 1:spar_y
        M1 = Theta(ii,jj:jj-1+win_spar);
        M2 = hist(M1, edge);
        rstdv(ii,jj) = var(M2).^0.5/(mean(M2));
    end
    clear edge
end
end
```

Local Coherence

```matlab
function [loc_coh] = local_coherence(Theta)
window = 3;
min_t = min(Theta,[], 2);
min_t = repmat(min_t,1,size(Theta,2));
max_t = max(Theta, [], 2);
max_t = repmat(max_t,1,size(Theta,2));
Sin_theta = (2*Theta-max_t-min_t)./(max_t-min_t);
Theta_rad = asin(Sin_theta);
exp_Theta = exp(1j*Theta_rad);
loc_coh = zeros(size(exp_Theta));
for jj = 1:size(Theta,2)
    loc_coh(:,jj) = abs(runningsum_total(exp_Theta(:,jj),window))./(window);
end
end
```

LCML/ECML Script

```matlab
tic
for d =1:1 % Initial conditions

    %% Parameters
    num_osc = 200; % Number of oscillators.
    time_transient = 50000;
    t_plot = 10; % keep it a multiple of win_spar
    time_steps = time_transient+t_plot;
    type = 0; % Type of initial condition
    win_spar = 50;
    row_osc = (1:1:num_osc)';
        IC = initiate_logistic(num_osc, type);
%        load('IC_mat.mat')
%        IC = IC_mat(:,d);
    sync_par = 0;
    for growth_rate = 3.2:0.1:3.2
    for neighbors = 1:7:1
        for eps = 0.3:0.02:0.3
            %          eps = 0.2; % Coupling strength.
            %          neighbors = 1;
            window = 2*neighbors+1;
            %% Time Evolution
            Theta_i = IC; % Initial phases
            logistic = zeros(size(Theta_i)); % Logistic
            Theta_c = logistic; % Size of coupling contribution
```

```matlab
Theta_f = logistic; % Size of final coupling

Theta = zeros(num_osc, time_steps+1);
Theta(:,1) = Theta_i;

for jj=1:time_steps
    logistic = growth_rate.*Theta_i.*(ones(num_osc,1)-Theta_i);
    Theta_c = runningsum(logistic, window);
    Theta_f = (1-eps).*logistic+(eps/(2*neighbors)).*Theta_c;
    Theta_i = Theta_f;
    Theta(:,jj+1)=Theta_i;
end
Theta_plot = Theta(:,time_transient+2:end);

%% Synchronization parameter

sync_par = sync_par+1;
Sync_R(sync_par) = (1/mean(var(Theta,0,2)))*var(mean(Theta,1));
eps_x(sync_par) = growth_rate;
sparsity = coeff_var(Theta, t_plot, win_spar);
loc_coherence  = local_coherence(Theta_plot);
loc_coherence2 = mean(loc_coherence,2);
loc_min(sync_par) = min(loc_coherence(:));
clear loc_coherence
Spar{sync_par} = sparsity;


%% Plotting
figure('units','normalized','Position',[0.5 0.5 0.18 0.20])
edgex_spar = 1:1:num_osc;
edgey_spar = time_transient+2:1:time_transient+t_plot+2-win_spar;
sparsity_blur = smooth_gaussian_2d(edgey_spar, edgex_spar,
sparsity, 0.5, 0.5);
sparsity_blur(1,1)=1;
sparsity_blur(1,end) = 6;
surf(edgex_spar, edgey_spar,
sparsity_blur','FaceLighting','phong','EdgeColor','none')
%             surf(edgex_spar, edgey_spar,
sparsity','FaceLighting','phong','EdgeColor','none')
            load('mycmap')
            ylabel('Lattice Points')
            xlabel('Time Series')
            str_title = sprintf('Sparsity');
            title(str_title)
            view(2)
            axis tight
            colormap(mycmap)
            colorbar
            str = sprintf('sparsity_%d_%d', floor(100*eps), d);
%% DotPlot
                figure
                subplot(4,1,1)
                hold on
                for ii=1:t_plot+1
                    plot(Theta(:,ii+time_transient), '.');
```

```matlab
                                      str = sprintf('Times %d-%d, eps = %f, nbr =
%d, r = %d',time_transient, time_transient+t_plot, eps, neighbors,
growth_rate);
                                      ylabel('Value')
                                      ylim([0 inf])
                                      title(str)
                              end
                              subplot(1,2,2)
                              hold on
                              for jj = 1:num_osc
                                      xaxis =
time_transient+2:1:time_transient+t_plot+1;
                                      plot(xaxis, Theta(jj,time_transient+2:end),
'.');
                                      str = sprintf('Time series for lattice
points, Coupling: %f', eps);
                                      xlabel('Time')
                                      ylabel('Value')
                                      ylim([0 1])
                                      title(str)
                              end
                              str2 = sprintf('Trace and Time %d-
%d',round(100*eps), neighbors);
                              print(str2, '-dpng', '-r0');
                              clearvars -except IC num_osc time_transient
t_plot time_steps growth_rate row_osc sync_par neighbors eps Sync_R

          %% Dynamics

                              figure('units','normalized','Position',[0 0 1 1])
                              subplot(4,4,2)
                              hold on
                              edge_time =
time_transient:1:time_transient+t_plot-1; % Dynamics
          %                 Theta_plot2 = Theta_plot;
          %                 Theta_plot = Theta_plot2(:,1:5:end);
          %                 edge_time = edge_time(1:5:end);
          %                 Theta_plot(1,1)=1;
          %                 Theta_plot(1,end) = 0;
                              surf(edge_time, row_osc,
Theta_plot,'FaceLighting','phong','EdgeColor','none')
                              colormap('jet')
          %                 colorbar('location', 'west')
                              colorbar
                              view(2)
                              ylabel('Lattice point')
                              xlabel('Time')
                              xlim([time_transient time_transient+t_plot-1])
                              str2 = sprintf('Initial Condition: %d, Growth
Rate: %d', d, growth_rate);
          %                 title(str2)
                              axis tight
                              fig = gcf;
          %                 set(fig, 'PaperPositionMode', 'auto')
```

```matlab
                        subplot(4,1,3)
                        plot(sparsity)
                        ylabel('Sparsity (S)')
                        ylim([0.6 1])
                        grid minor

                        subplot(4,1,4)
                        plot(loc_coherence2)
                        ylabel('Local Coherence (L)')
                        ylim([0.5 1])
                        grid minor
            %
                        str2 = sprintf('Dynamics %d-%d-%d',d,
    round(100*eps), neighbors);
                        print(str2, '-dpng', '-r0');

        figure('units','normalized','outerposition',[0 0 1 1])
        subplot(4,4,1)
        kk = 75;
        h1 = plot(Theta(:,kk+time_transient));
        set(h1, 'Marker', '.', 'MarkerSize', 5)
        str = sprintf('Timestep: %d', kk+time_transient-1);
        xlabel('Lattice point')
        ylabel('Value')
        ylim([0 1])
        title(str)

        figure('units','normalized','outerposition',[0 0 1 1])
        subplot(4,4,1)
        xaxis = time_transient+2:1:time_transient+t_plot+1;
        jj = 30;
        h1 = plot(xaxis, Theta(jj,time_transient+2:end), '.');
        set(h1, 'Marker', '.', 'MarkerSize', 5)
        str = sprintf('Time series for lattice points, Node: %d', jj);
        xlabel('Time')
        ylabel('Value')
        ylim([0 1])
        title(str)
        axis tight


            nframe=t_plot+1; % Movie
            mov(1:nframe)= struct('cdata',[],'colormap',[]);
            set(gca,'nextplot','replacechildren')
            for k=1:nframe
                h1 = plot(Theta(:,k+time_transient));
                set(h1, 'Marker', '.', 'MarkerSize', 5)
                str = sprintf('Timestep: %d', k+time_transient-1);
                xlabel('Lattice point')
                ylabel('Value')
                ylim([0 1])
                title(str)
                mov(k)=getframe(gcf);
            end
            str3 = sprintf('%d-%d.avi', 100*eps, neighbors);
            movie2avi(mov, str3, 'compression', 'None');
```

```matlab
            clear mov

%

            n_frame2 = num_osc; % Movie: Time-series
            mov2(1:n_frame2)= struct('cdata',[],'colormap',[]);
            set(gca,'nextplot','replacechildren')
            xaxis = time_transient+2:1:time_transient+t_plot+1;
            for jj = 1:num_osc
                h1 = plot(xaxis, Theta(jj,time_transient+2:end));
                set(h1, 'Marker', '.', 'MarkerSize', 5)
                str = sprintf('Time series for lattice points, Node: %d',
jj);
                xlabel('Time')
                ylabel('Value')
                ylim([0 1])
                title(str)
                mov2(jj)=getframe(gcf);
            end
            str3 = sprintf('T%d-%d.avi', 100*eps, neighbors);
            movie2avi(mov2, str3, 'compression', 'None');
            %}
        end
    end
    end
    end

figure('units','normalized','outerposition',[0 0 1 1])
h1 = plot(eps_x,Sync_R);
grid on
xlabel('Coupling strength')
ylabel('Synchronization parameter')
set(h1,'Marker','.','MarkerSize',15)
ylim([0 1])
xlim([2.8 3.45])
```

4. 1D HP

```matlab
tic

for d = 9:9 % Initial conditions
% d =1;
ee = 1;
nn = 1;
    %% Parameters
    num_osc = 100; % Number of oscillators.
    time_transient = 5000;
    t_plot = 2000; % keep it a multiple of win_spar
    time_steps = time_transient+t_plot;
    type = 0; % Type of initial condition
    growth_rate = 3.9;
    win_spar = 50;
    beta = 5;

    row_osc = (1:1:num_osc)';
```

```matlab
    load('IC_mat.mat')
    IC = H(:,1);
    IC2 = P(:,d);
% IC = IC_mat(:,d);
% IC2 = IC_mat(:,d+1);



    for neighbors = 1:7:1
        for eps = 0.4:0.05:0.4
            window = 2*neighbors+1;

            Theta_i = IC; % Initial value of host
            parasite_i = IC2; % Initial value of parasite
            logistic = zeros(size(Theta_i)); % Logistic
            Theta_c = logistic; % Size of coupling contribution
            Theta_f = logistic; % Size of final coupling
            parasite = zeros(size(Theta_i));
            parasite_c = parasite;
            parasite_f = parasite;
            Theta = zeros(num_osc, time_steps+1);
            Para = zeros(num_osc, time_steps+1);
            Theta(:,1) = Theta_i;
            Para(:,1) = parasite_i;

            for jj=1:time_steps
                logistic = growth_rate.*Theta_i.*(ones(num_osc,1)-
Theta_i).*exp(-beta*parasite_i);
                parasite = Theta_i.*(1-exp(-beta*parasite_i));
                Theta_c = runningsum(logistic, window);
                parasite_c = runningsum(parasite, window);
                Theta_f = (1-eps).*logistic+(eps/(2*neighbors)).*Theta_c;
                parasite_f = (1-
eps).*parasite+(eps/(2*neighbors)).*parasite_c;
                Theta_i = Theta_f;
                parasite_i = parasite_f;
                Theta(:,jj+1)=Theta_i;
                Para(:,jj+1) = parasite_f;
            end

%           figure
%               plot_phasespaceHP(Theta, Para, time_transient, eps, neighbors,
d) %   Phase-space plot
%               Loc = Theta(:,time_transient:end);
%               loc_coherence  = local_coherence(Loc);
%
%               figure('units','normalized','outerposition',[0 0 1 1])
%               hold on
%               subplot(3,1,3)
%               edgex_spar = 1:1:num_osc;
%               edgey_spar = time_transient:1:time_transient+t_plot+1;
%               loc_coherence(1,1)=1;
%               loc_coherence(1,end) = 0;
%               surf(edgey_spar, edgex_spar,
loc_coherence,'FaceLighting','phong','EdgeColor','none')
%               xlabel('Lattice Points')
```

```matlab
%            ylabel('Time Series')
%            str_title = sprintf('Local Order Parameter');
%            title(str_title)
%            view(2)
%            axis tight
%            colorbar
%            str = sprintf('order_parameter_%d_%d', floor(100*eps), d);
%            print(str, '-dpng', '-r0')




%            %{
            CV =coeff_var(Theta, t_plot, win_spar);%
%            CV2 = coeff_var(Para, t_plot, win_spar);

            figure('units','normalized','outerposition',[0 0 1 1])
            hold on
            subplot(2,1,2)
            edgex_spar = 1:1:num_osc;
            edgey_spar = time_transient+2:1:time_transient+t_plot+2-win_spar;
            sparsity_blur = CV;
            sparsity_blur(1,1)=0;
            sparsity_blur(1,end) = 6;
            surf(edgex_spar, edgey_spar,
sparsity_blur','FaceLighting','phong','EdgeColor','none')
            load('mycmap')
            xlabel('Lattice Points')
            ylabel('Time Series')
            str_title = sprintf('CV Host, Coupling %d', eps);
            title(str_title)
            view(2)
            axis tight
            colormap(mycmap)
            colorbar

%            figure
            subplot(2,1,1)
            edge_time = time_transient:1:time_transient+t_plot-1; % Dynamics
            Theta_plot = Theta(:,time_transient+2:end);
%            Theta_plot2 = Theta_plot;
%            Theta_plot = Theta_plot2(:,1:100:end);
%            edge_time = edge_time(1:100:end);
%            Theta_plot(1,1)=1;
%            Theta_plot(1,end) = 0;
            Theta_plot(1,1)=1;
            Theta_plot(1,end) = 0;
            surf(row_osc, edge_time,
Theta_plot','FaceLighting','phong','EdgeColor','none')
            colormap(jet)
            colorbar
            view(2)
%            xlabel('Lattice point')
            ylabel('Time')
            xlim([time_transient time_transient+t_plot-1])
            str2 = sprintf('Host, Initial Condition: %d, Couping: %d,
Neighbors: %d', d, eps, neighbors);
```

```
                title(str2)
                axis tight
            end
%            nn=nn+1;
    end
end


```

6. 2D HP

```
num_x = 50;
num_y = 50;
time_steps = 70;
transience = 1;
growth_rate = 4;
beta = 4;


load('IC.mat')


for neighbors = 10:1:10
    for eps = 0.2:0.1:0.2
        window = 2*neighbors+1;

        fH = zeros(size(H));
        fH_cpld = fH;
        fH_final = fH;
        gP = zeros(size(P));
        gP_cpld = gP;
        gP_final = gP;
        Host = zeros(num_x, num_y, time_steps);
        Para = zeros(num_x, num_y, time_steps);
        Host(:,:,1) = H;
        Para(:,:, 1) = P;

        for jj=1:time_steps
            fH = growth_rate.*H.*(ones(num_x,num_y)-H).*exp(-beta*P);
            gP = H.*(1-exp(-beta*P));
            fH_cpld = runningsum2D(fH, window);
            gP_cpld = runningsum2D(gP, window);
            fH_final = (1-eps).*fH+(eps/(4*neighbors)).*fH_cpld;
            gP_final = (1-eps).*gP+(eps/(4*neighbors)).*gP_cpld;
            H = fH_final;
            P = gP_final;
            Host(:,:,jj+1)=H;
            Para(:,:,jj+1) = P;
        end

        figure
        n_frame2 = time_steps-transience;
        mov2(1:n_frame2)= struct('cdata',[],'colormap',[]);
        set(gca,'nextplot','replacechildren')
        xaxis = 1:1:num_x;
        yaxis = 1:1:num_y;
        for jj = 1:time_steps-transience
            M = Host(:,:,jj+transience)';
```

```matlab
            M(1,1) = 0;
            M(1,end)=1;
            h1 = surf(xaxis, yaxis,
M,'FaceLighting','phong','EdgeColor','none');
            str = sprintf('Time: %d', jj);
            colorbar
            view(2)
            axis tight
            xlabel('X')
            ylabel('Y')
            title(str)
            mov2(jj)=getframe(gcf);
        end
        str3 = sprintf('Movie_%d_%d', 100*eps, neighbors);
        movie2avi(mov2, str3, 'compression', 'None');
        save(str3)
    end
end
```