

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE FÍSICA

**ESTADOS ASSINTÓTICOS EM
REDES DE MAPAS QUÁRTICOS
ACOPLADOS LOCALMENTE**

LUCIANO CAMARGO MARTINS

Dissertação realizada sob a orientação do Prof. Dr. Leonardo Gregory Brunnet e apresentada ao Instituto de Física da UFRGS para preenchimento parcial dos requisitos à obtenção do título de Mestre em Física.

PORTO ALEGRE

2001

Aos meus filhos, companheiros de todas as horas,
Lucas, Ana Lúcia e Maria Clara

Agradecimentos

- Ao Leonardo, pelo tempo e esforço que me dedicou durante esses dois anos em que trabalhamos juntos;
- A todos os companheiros da FEJ¹, por terem possibilitado o meu afastamento para desenvolver esse trabalho;
- Ao meu tio Welcy, pelas *charlas* e pelos mates; e à minha tia Estér, pelo desvelo com que nos recebeu tantas vezes em seu *JK*;
- Às minhas primas Rejane e Joice, pela força que me deram;
- À minha irmã Lia Rosa, por tudo!
- Aos meus pais, por terem sempre investido e acreditado nos meus estudos;
- Aos meus filhos, por todos os momentos que passamos juntos;
- À cidade e ao povo de Porto Alegre.

¹Faculdade de Engenharia de Joinville–UDESC, Joinville-SC.

Resumo

O objetivo deste trabalho é investigar redes bidimensionais de mapas acoplados localmente. A dinâmica local é dada pelo mapa quártico que possui dois atratores. Diferentes tipos de condições iniciais são usadas: valores iniciais aleatórios para cada sítio da rede, várias regiões uniformes com diferentes valores entre si. Diferentes estados assintóticos são encontrados ao se variar a intensidade ou a topologia do acoplamento local entre os sítios. Estes estados são caracterizados e interpretados ao longo do trabalho pelas distribuições dos valores dos sítios e pela rugosidade média da rede.

Abstract

We investigate a two-dimensional, locally coupled map lattice (CML) with the local dynamics driven by the multi-attractor quartic map. Different sets of initial conditions such as random initial values for each site or random initial values for regions were tested leading to the same qualitative results. The system reaches different asymptotic states as the intensity or the topology of the local coupling is varied. These states are characterized and interpreted throughout this work by the distributions of the values of the maps and by the average roughness over the lattice.

Sumário

<i>Resumo</i>	<i>iv</i>
<i>Abstract</i>	<i>v</i>
1 Introdução	1
2 Revisão	6
2.1 O Caos Clássico	6
2.2 Os Sistemas Espacialmente Extensos	8
3 O Mapa Quártico	12
3.1 O Mapa	12
3.2 O Diagrama de Bifurcação	13
3.3 A Bacia Atratora	14
3.4 A Convergência de X_t	18
3.5 O Espaço de Parâmetros	19
4 Redes de Mapas Acoplados	24
4.1 A Rede	24
4.2 A Evolução da Rede	25
4.3 O Algoritmo	25
4.4 A Rede Hexagonal	26
4.5 Caracterização da Dinâmica Assintótica da Rede	29
4.5.1 Sincronização	29
4.5.2 CCNT	30

4.6	Medidas	31
4.6.1	O Valor Médio $\langle X(t) \rangle$	31
4.6.2	O Desvio Quadrático Médio $\sigma_X^2(t)$	31
4.6.3	O Espectro $\mathcal{P}[X(t)]$	32
4.6.4	A Rugosidade $G_\epsilon(t)$	32
4.7	Exploração Preliminar da Rede	36
5	Resultados	39
5.1	Inicialização RANDOM	40
5.1.1	Rede Quadrada, Primeiros Vizinhos	41
5.1.2	Rede Quadrada, Segundos Vizinhos	48
5.1.3	Rede Hexagonal, Primeiros Vizinhos	50
5.2	Inicialização RING50%	51
5.2.1	Rede Quadrada, Primeiros Vizinhos	52
5.2.2	Rede Quadrada, Segundos Vizinhos	53
5.2.3	Rede Hexagonal, Primeiros Vizinhos	54
5.3	Inicialização PATCH3	55
5.3.1	Rede Quadrada, Primeiros Vizinhos	55
5.3.2	Rede Quadrada, Segundos Vizinhos	60
5.3.3	Rede Hexagonal, Primeiros Vizinhos	62
5.4	Inicialização PATCH5	62
5.4.1	Rede Quadrada, Primeiros Vizinhos	63
5.4.2	Rede Quadrada, Segundos Vizinhos	63
5.4.3	Rede Hexagonal, Primeiros Vizinhos	63
5.5	Quadro Geral de Resultados	65
6	Conclusões	67
6.1	Continuidade Deste Trabalho	68
	Apêndices	71
A	Índice Geral dos Estados Assintóticos	71
B	Fluxograma Básico do Programa qmcm1-2.0.0.c	84

C - Listagem do Programa <code>qmcml-2.0.0.c</code>	85
D - Listagem da Biblioteca <code>ufrgs-2.0.0.c</code>	101
E - Listagem da Biblioteca <code>mickey-2.0.0.c</code>	108
F - Lista de Figuras	115
G - Lista de Tabelas	116
H - Lista de Citações	116
I - \LaTeX Layout	118
Referências Bibliográficas	119

Capítulo 1

Introdução

Tentar descrever macroscopicamente um sistema físico composto por um grande número de partículas ou sub-sistemas similares, cuja dinâmica individual possa ser modelada, simulada e inspecionada localmente, parece ser uma estratégia natural de abordagem para muitos dos sistemas estudados na Física, especialmente em casos onde se sabe a priori que nenhum tipo de solução analítica pode ser encontrada para descrever a dinâmica global do sistema.

A idéia de investigar o comportamento coletivo de sistemas complexos a partir da modelagem computacional da dinâmica local dos elementos básicos do sistema e da definição de regras simples para descrever a forma pela qual esses elementos interagem entre si, foi primeiramente utilizada por KANEKO [KANEKO, 1983] em uma rede unidimensional de mapas logísticos acoplados globalmente, isto é, onde todos os elementos do sistema interagem entre si, e onde cada mapa evolui individualmente, segundo uma mesma regra recursiva comum a todos os mapas. O principal objetivo desse tipo de abordagem foi, inicialmente, explorar a dinâmica espaço-temporal de sistemas extensos. Do ponto de vista numérico-computacional, simulações desse tipo de sistema podem ser realizadas com relativa facilidade, hoje factíveis em microcomputadores pessoais, e têm-se mostrado uma importante ferramenta de inferência e exploração no estudo de modelos dinâmicos cada vez mais complexos, ou seja, cada vez mais realistas, do ponto de vista físico [KANEKO, 1989a, BRUNET *et al.*, 1998b].

Dos resultados iniciais obtidos através do estudo de redes de mapas acoplados, destaca-

se a descoberta do *comportamento coletivo não-trivial* nestes sistemas, isto é, mesmo em redes de mapas cujo atrator local é um ciclo periódico finito, o sistema pode apresentar globalmente um comportamento coletivo caótico, dependendo da intensidade e do alcance das interações locais entre os mapas elementares. A partir do descobrimento deste novo tipo de comportamento coletivo, uma vasta série de aplicações e novas propriedades coletivas têm sido investigadas, muitas delas sugeridas por sistemas reais provindos de várias áreas: Física, Química, Biologia [SOLÉ *et al.*, 1992], Economia, Medicina, Micro-Eletrônica, Meteorologia, Imunologia [BERNARDES *et al.*, 1997], Engenharia de Tráfego [BRUNET *et al.*, 1997], Inteligência Artificial [SINHA *et al.*, 1998] e outras.

A universalidade dos resultados obtidos pelo estudo de redes de mapas acoplados justifica e corrobora a multi-disciplinariedade imposta pelo estudo dos Sistemas Complexos, e de particular interesse dentre eles, os Sistemas Dinâmicos Caóticos Determinísticos, trazendo à tona uma nova forma de abordagem ao estudo dos sistemas de muitos corpos.

Um tipo de fenômeno observado em redes de sistemas acoplados, é o fenômeno da *sincronização*, descoberto em 1665 pelo físico holandês Cristiaan Huygens. Ele observou que osciladores fracamente acoplados tendem a sincronizar suas fases depois de algum tempo, mesmo depois de perturbados propositadamente.

Sistemas de osciladores acoplados podem ser encontrados em toda a natureza, inclusive nos seres vivos. Um exemplo pode ser visto no comportamento coletivo de algumas espécies de vaga-lumes que, na época do acasalamento, apagam e acendem suas luzes de forma sincronizada [STROGATZ *et al.*, 1993, STROGATZ, 1994]. A sincronização de osciladores biológicos pode ser encontrada também no mecanismo de geração dos ritmos cardíacos e respiratórios, e nos ritmos circadianos [GLASS *et al.*, 1997] e de locomoção e respiração.

Mais recentemente, e em várias áreas diferentes da ciência, foram descobertas evidências do mesmo fenômeno de sincronização de fases, ou em outros casos de sincronização de frequências, em vários sistemas essencialmente distintos, como por exemplo, nas cadeias de junções Josephson acopladas [WIESENFELD *et al.*, 1996, WIESENFELD *et al.*, 1989] que podem apresentar tanto estados sincronizados como não-sincronizados. Apesar da necessidade de uma teoria quântica para explicar a origem do efeito Josephson - o tunelamento que ocorre nas junções Josephson sob certas condições - a dinâmica desse dispositivo

pode ser descrita através da equação de um pêndulo amortecido e forçado, havendo assim um elo comum entre uma rede de junções Josephson e uma rede de pêndulos acoplados, onde o fenômeno da sincronização observado é mais intuitivo. Em ambos os casos, do ponto de vista matemático, os sistemas físicos são modelados por um conjunto de equações diferenciais ordinárias acopladas. A dinâmica destas redes foi extensivamente estudada, e vários aspectos foram explorados, inclusive a sincronização [KURAMOTO, 1984, PIKOVSKY *et al.*, 1997, VIEIRA *et al.*, 1994].

Do ponto de vista computacional, o tempo de processamento gasto com o estudo numérico deste tipo de rede é muito grande devido à natureza contínua da sua variável temporal, o que torna o estudo assintótico da dinâmica do sistema bastante difícil. Por outro lado, as redes de mapas acoplados não necessitam de muito tempo computacional, pois o tempo, neste caso, é discreto, possibilitando uma maior facilidade para a sua implementação computacional. Após os trabalhos iniciais de KANEKO, muitos outros surgiram utilizando as redes de mapas acoplados, dentre os quais podemos citar GRASSBERGER *et al.*, 1991, KELLER *et al.*, 1992 e TANAKA *et al.*, 1997, que estudam a transição de fase em redes de mapas acoplados. Simulações a respeito do comportamento termodinâmico de modelos de Ising, utilizando redes de mapas acoplados, são encontrados na literatura [MARCQ *et al.*, 1997, ANDRADE, 1999]. O fenômeno da sincronização também foi estudado nestes sistemas, como é visto em VIANA *et al.*, 1998, e CHATERJEE *et al.*, 1996.

Em sua generalidade, estes trabalhos utilizam-se de mapas locais com apenas um atrator para a dinâmica. Esta limitação pode ser superada pelo uso de mapas com atratores múltiplos, isto é, mapas cuja dinâmica assintótica depende das condições iniciais. Um exemplo canônico para tal mapa é o mapa quártico, introduzido por GALLAS em 1993 [GALLAS, 1993] motivado em generalizar o mapa logístico com um novo parâmetro independente. Na sua forma geral, o mapa quártico pode ser visto como uma transformação do tipo $X(t+1) = f(X(t), a, b)$ definida sobre um intervalo contínuo para a variável X , e onde a e b são dois os parâmetros. A dinâmica deste mapa depende da escolha dos dois parâmetros e da condição inicial usada, de onde será obtida uma série temporal através da fórmula recursiva dada, e que pode apresentar ciclos com período finito e órbitas abertas, ou seja, caóticas. Neste trabalho, usou-se especialmente os valores comuns $a_0 = 0.35$

e $b_0 = 1.35$ para o mapa quártico [BRUNET *et al.*, 1998b] para a construção das redes estudadas.

Os elementos de uma rede de mapas acoplados interagem segundo uma prescrição especificada, que pode ser do tipo local, ou do tipo global, onde cada elemento interage com todo o resto do sistema. Tal prescrição pode utilizar um ou mais parâmetros, permitindo que os elementos da rede interajam mais ou menos intensamente. Neste estudo usou-se apenas o acoplamento local, descrito por um parâmetro de acoplamento ε (*epsilon*), que define a intensidade das interações locais entre os elementos da rede e sua vizinhança próxima, e é responsável pelo efeito difusivo na rede. Variando-se esse parâmetro de acoplamento, o número de vizinhos e as condições iniciais da rede, investigamos os diferentes regimes finais atingidos pela dinâmica coletiva, em tempo suficientemente longo onde quantidades como valores médios, espectros de distribuições e rugosidade puderam ser definidos e medidos sobre o sistema.

O objetivo deste trabalho é estudar e caracterizar, para as diferentes regiões do espaço de parâmetro de acoplamento ε , as diferentes dinâmicas coletivas obtidas para o sistema, bem como identificar sob que condições de inicialização da rede há o surgimento espontâneo de paredes fixas, isolando o sistema em sub-redes de dinâmica mista, onde coexistem regiões sincronizadas e caóticas, isoladas por paredes fixas; e investigar qual a estabilidade destes diferentes regimes dinâmicos.

O Capítulo 2 apresenta uma revisão geral a respeito dos sistemas dinâmicos espacialmente extensos mais comuns na literatura: os autômatos celulares, as equações diferenciais parciais e as redes de mapas acoplados.

A dinâmica do mapa local usado, o mapa quártico, está detalhada no Capítulo 3, onde é apresentada a bacia fractal local de atração do mapa, e o diagrama de sua segunda iterada, que serve de base para o entendimento do comportamento da rede de mapas quárticos acoplados localmente. Os diagramas de fase obtidos a partir do expoente de Lyapunov, mostrando o ciclo do mapa local em função dos parâmetros a e b são apresentados, e inspecionados em torno do ponto (a_0, b_0) .

No Capítulo 4, definimos a rede estudada e as quantidades relevantes medidas sobre a mesma, tais como a *rugosidade*, usada como um parâmetro para a análise das transições ocorridas na mudança da dinâmica do comportamento coletivo do sistema, à medida em

que variamos a constante de acoplamento ε .

No Capítulo 5, resumimos os principais resultados obtidos através da simulação numérica da dinâmica coletiva da rede e discutimos as condições de estabilidade sob as quais pode-se encontrar a multi-estabilidade observada para os intervalos de ε em que a rede assume o comportamento misto (caos+sincronização), acompanhado pelo surgimento de *paredes rígidas* sobre o sistema [ABRAMSON *et al.*, 1998].

Por fim, no Capítulo 6 apresentamos as conclusões finais deste trabalho, onde procuramos sintetizar a partir dos resultados obtidos as propriedades dinâmicas gerais observadas. Algumas das perspectivas futuras para a continuidade e extensão deste trabalho estão listadas ao final deste capítulo.

Capítulo 2

Revisão

2.1 O Caos Clássico

O estudo de sistemas dinâmicos clássicos de baixa dimensionalidade, ou seja, sistemas que possuem poucos graus de liberdade, levou à necessidade e, ao mesmo tempo, serviu como base de uma nova classe de métodos teóricos e computacionais chamada de *Caos em Sistemas Dinâmicos Clássicos* ou simplesmente *Caos em Sistemas Dinâmicos*. Um sistema físico largamente estudado, e que pode ser adotado como um paradigma para essa classe de sistemas caóticos, é o pêndulo simples amortecido e forçado que, dependendo dos parâmetros e condições iniciais escolhidos para o seu movimento, apresenta uma extensa gama de soluções caóticas.

Por outro lado, existem sistemas físicos que apresentam caos *espaço-temporal*, como por exemplo, turbulência em fluídos, sistemas ergódicos em Mecânica Estatística, formação de padrões em fenômenos naturais, reações químicas, sólitons, dinâmica populacional, ondas químicas e redes de neurônios, possuem muitos graus de liberdade, além de serem espacialmente extensos, isto é, não podem ser reduzidos a um modelo do tipo “partícula”, como no caso do pêndulo simples. O que ocorre nesses sistemas, e o que os torna similares do ponto de vista do caos, é que cada uma das suas partes constituintes evolui no tempo segundo uma dinâmica caótica, interagindo com o resto do sistema direta ou indiretamente, de forma sincronizada ou com efeitos de retardos no tempo, o que afeta sensivelmente a dinâmica coletiva do mesmo como um todo. Assim como para os sistemas

de baixa dimensionalidade diz-se que a sensibilidade às condições iniciais é uma indicação direta de que a dinâmica do sistema é caótica, para os sistemas espaço-temporais, as condições iniciais e as dinâmicas locais assumidas pelas diferentes partes do sistema definem o seu aspecto dinâmico global.

Um caso ainda mais notável desse tipo de comportamento coletivo global, induzido pelos efeitos coletivos das interações entre as partes de um sistema espacialmente extenso juntamente com a sua evolução temporal, é o caso do *comportamento coletivo não-trivial*. Neste caso, mesmo sendo formado por elementos cuja dinâmica local não apresenta caos, isto é, possui uma órbita periódica bem definida dada por algum atrator finito, o sistema apresenta coletivamente uma dinâmica caótica complexa. Não se trata aqui de efeitos transientes sobre o sistema, pois este pode evoluir para uma dinâmica estacionária distinta daquela que se poderia supor a partir da análise da dinâmica local dos seus elementos constituintes.

Essa visão microscópica do sistema dinâmico espacialmente extenso, que o trata como um aglomerado de subsistemas aparentemente mais simples e cuja dinâmica local associada à forma com que os subsistemas trocam informação define os rumos da dinâmica global, talvez seja responsável pelo fato de não podermos prever a priori a complexidade apresentada pela dinâmica coletiva do sistema. Isto se deve ao método utilizado, mas que, como uma primeira e mais simples forma de abordagem, talvez seja o mais natural a ser seguido numa abordagem inicial. Assim, na descrição de modelos mais gerais, tanto quanto ao número de graus de liberdade, quanto à geometria, foram introduzidos os sistemas dinâmicos que apresentam extensão espacial a fim de modelar e caracterizar o comportamento dos vários diferentes sistemas extensos citados. Desta forma, muitos dos conceitos da teoria de caos desenvolvida para os sistemas clássicos foram generalizados para o caso dos sistemas espacialmente extensos que apresentam caos do tipo espaço-temporal. Numa visão mínima da sua complexidade, estes sistemas são descritos através do tempo, do espaço e de uma variável de estado local, sendo que cada uma destas três variáveis citadas pode assumir um caráter *discreto* ou *contínuo*, dependendo da forma com a dinâmica do sistema é modelada. Obtemos desse modo uma classificação dos sistemas extensos com oito tipos, dos quais apenas quatro recebem maior atenção. São eles: as equações diferenciais parciais, os autômatos celulares, as cadeias de osciladores e as redes

Modelo Tipo	Espaço	Tempo	Variável dinâmica
Sistema de EDP	contínuo	contínuo	contínua
Autômato Celular	discreto	discreto	discreta
Cadeia de Osciladores	discreto	contínuo	contínua
Rede de Mapas Acoplados	discreto	discreto	contínua

Tabela 2.1: *Os sistemas espacialmente extensos mais estudados, classificados quanto ao tipo de variáveis.*

de mapas acoplados. A Tabela 2.1 mostra a classificação dos sistemas extensos, quanto ao tipo de variáveis consideradas.

2.2 Os Sistemas Espacialmente Extensos

Os sistemas descritos por um conjunto de equações diferenciais parciais (EDP) surgem associados à modelação matemática de vários problemas físicos [BRUNET *et al.*, 1994] e geométricos, quando as suas funções relevantes dependem de duas ou mais variáveis independentes. Estas variáveis podem ser o tempo e uma ou mais coordenadas espaciais, por exemplo. Tanto as variáveis independentes, quanto as funções que representam o estado local do sistema físico num dado instante de tempo, devem variar continuamente, dentro de intervalos estabelecidos de acordo com a natureza do problema associado. Equações como a de Poisson, de Laplace, da difusão, de Helmholtz, entre outras, são exemplos bem conhecidos das EDP aplicadas a problemas físicos. Estas equações também são frequentemente estudadas como modelo para caos espaço-temporal. Um outro exemplo é a equação de Ginzburg-Landau [KANEKO, 1993]

$$\frac{\partial \phi(\vec{r}, t)}{\partial t} = \gamma \phi(\vec{r}, t) - g |\phi|^2 \phi + D \nabla^2 \phi(\vec{r}, t) , \quad (2.1)$$

que descreve processos do tipo reação-difusão; sendo $\phi(\vec{r}, t)$ a variável de estado, \vec{r} a posição e t o tempo. O parâmetro D é a constante de difusão, γ e g são os coeficientes de reação. Outras EDP são bastante utilizadas para estudar o comportamento caótico em hidrodinâmica, física de plasmas [THAM *et al.*, 1992] e outras áreas da Física. Um outro exemplo de EDP que apresenta caos espaço-temporal pode ser encontrado em sistemas

que representam ondas interagindo de forma não-linear [LOPES *et al.*, 1998].

Um *autômato celular* é essencialmente uma representação de um conjunto de muitos objetos que interagem simultaneamente, sendo que o espaço, o tempo, e a variável de estado local são todos discretos. Tal sistema dinâmico tem as seguintes propriedades:

- o espaço é discretizado em uma rede regular de dimensão um, dois, três ou N , no caso mais geral;
- a variável local de cada sítio da rede pode estar em apenas um dos finitos estados acessíveis ao sistema. Os estados são representados por S números inteiros;
- os valores de todos os sítios na rede são atualizados sincronizadamente em cada instante do tempo discreto (iteração da rede);
- os valores dos sítios são atualizados usando um conjunto de regras que levam em conta os valores do sítio e de seus vizinhos.

Grande parte dos estudos que estão relacionados a estes sistemas estão direcionados principalmente para a sincronização [MORELLI *et al.*, 1998] e para a auto-organização [MEDVEDEV *et al.*, 1998]. Na modelagem de fenômenos físicos, este tipo de sistema dinâmico é bastante usado para simular o comportamento estatístico de sistemas do tipo modelo de Ising [HEMMINGSSON *et al.*, 1993].

As cadeias de osciladores acoplados apresentam o tempo e a variável de estado contínuos e o espaço discreto. A discretização do espaço é facilmente obtida através da construção de uma rede, na qual são distribuídos N osciladores de forma que cada um esteja a uma mesma distância dos seus vizinhos adjacentes. O local da rede onde um único oscilador está é denominado de *sítio*. Os N elementos da rede interagem seguindo alguma regra estabelecida, que pode ser o acoplamento introduzido em KURAMOTO, 1984

$$\dot{\theta} = \Omega_i + \frac{K}{N} \sum_{j=1}^N \text{sen}(\theta_j - \theta_i) \quad , \quad (2.2)$$

sendo θ_i e Ω_i , respectivamente, a fase e a frequência natural do i -ésimo oscilador, e K é a constante de acoplamento. Uma aplicação das cadeias de osciladores acoplados em biologia é descrita em SOMPOLINSKI, 1991, onde é simulada uma rede de neurônios associados à região do cérebro de um gato, responsável pelo processamento visual.

Há também um grande interesse na modelagem e descrição de redes de junções Josephson [WIESENFELD *et al.*, 1989, WIESENFELD *et al.*, 1996] e *lasers* [HOHL *et al.*, 1997]. Redes de osciladores Rössler, devido à sua dinâmica bastante rica, aparecem frequentemente na literatura [OSIPOV *et al.*, 1997, BRUNET *et al.*, 1998a].

Uma rede de mapas acoplados unidimensional é construída da seguinte forma: uma variável $x_t^{(i)}$ é distribuída sobre uma rede espacial unidimensional ($d = 1$); sendo t o tempo e i a posição do sítio na rede: ($i = 1, 2, \dots, N$), onde N é o tamanho da rede. Devido ao tamanho finito N da rede, assume-se uma condição periódica de contorno, que no caso mais simples pode ser representada pela condição $x_n^{(i \pm N)} \equiv x_n^i$. A evolução da variável de estado x_t é governada pela dinâmica local, que evolui com o tempo, e pela forma de acoplamento com os outros sítios. Isto pode ser exemplificado através da seguinte rede de mapas acoplados:

$$x_{t+1}^{(i)} = f(x_t^{(i)}) + \varepsilon [f(x_t^{(i-1)}) - 2f(x_t^{(i)}) + f(x_t^{(i+1)})] , \quad (2.3)$$

sendo ε a constante de acoplamento e $f(x_t^{(i)})$ a dinâmica local da rede, ou seja, uma função não-linear que corresponde a um mapa unidimensional [LOSSON *et al.*, 1994]. Como pode ser visto em [PINTO, 1999], apêndice A, essa forma de acoplamento pode ser derivada a partir de uma EDP de segunda ordem, que descreve processos de reação-difusão.

A inspeção direta do último termo da equação (2.3), indica claramente uma equação de diferenças finitas para o cálculo numérico de uma diferencial de segunda ordem em x , pelo método de Euler. Esse último termo, define a parte difusiva da rede, já que promedia os valores de x_t sobre a sua vizinhança, em contraposição ao primeiro termo, que define a evolução individual de cada sítio. A competição definida pela importância relativa desses termos irá influir diretamente sobre que tipo de comportamento coletivo a rede irá assumir, ou seja, será fundamental para a dinâmica global da rede.

Este trabalho dedica-se ao estudo de redes de mapas acoplados, ou seja, aos sistemas nos quais o espaço e o tempo são variáveis discretas, e a variável de estado local é contínua, o que foi primeiramente introduzido por Kaneko em [KANEKO, 1983] como um modelo simples para caos espaço-temporal. Inicialmente estas redes eram unidimensionais, mais tarde, Kaneko estendeu este modelo a redes bidimensionais [KANEKO, 1989c].

A modelagem e caracterização desta classe de sistemas mostrou-se importante no

estudo da turbulência como um fenômeno geral [KANEKO, 1989a], [COSENZA *et al.*, 1996], e este fato não está restrito apenas à dinâmica de fluidos, mas ocorre em outras áreas da ciência e da ciência aplicada, como por exemplo: Física do Estado Sólido, Óptica, Química, problema de formação de padrões em sistemas biológicos, etc. [KANEKO, 1990]. Caos espaço-temporal pode ser observado na convecção de Bénard, convecção em cristais líquidos, problemas de reação-difusão em Química e cadeias de junções Josephson.

Capítulo 3

O Mapa Quártico

Neste capítulo apresentamos uma breve revisão do mapa local e as propriedades relevantes para a construção e análise da rede de mapas acoplados — o mapa quártico.

3.1 O Mapa

O mapa quártico é definido pela equação recursiva

$$X_{t+1} = (X_t^2 - a)^2 - b , \quad (3.1)$$

onde X é a variável que está sendo “mapeada”, ou *variável dinâmica*; e a e b são os dois parâmetros do mapa. Expandindo-se o quadrado na equação acima, mostra-se que o mesmo pode ser reescrito como um polinômio de quarto grau em X_t , justificando o nome “quártico” escolhido para o mapa. Apesar deste ser um mapa unidimensional, fica definido sobre um espaço de parâmetros bidimensional (co-dimensão 2), isto é, qualquer ponto $(a, b) \in \mathbf{R}^2$ define um mapeamento diferente, o que determina particularmente a evolução da variável X_t do mapa, tanto quanto a escolha do valor inicial $X_0 \equiv X_{t=0}$.

Através de uma inspeção simples da equação (3.1), pode-se observar que quando $a = b$ o mapa assim definido nada mais é do que a segunda iteração do *mapa logístico*, que é um dos mapas mais bem estudados da literatura, e que na sua forma mais simples pode ser escrito como

$$X_{t+1} = X_t^2 - a \quad (3.2)$$

que depende de um único parâmetro real a .

3.2 O Diagrama de Bifurcação

Tomando-se um valor inicial X_0 real qualquer, os mapas acima definidos permitem o cálculo recursivo de uma série infinita de pontos $\{X_1, X_2, \dots, X_t, X_{t+1}, X_{t+2}, \dots\}$, o que pode ser visto nos seus diagramas de bifurcação das Figura 3.1a para o mapa logístico, e Figura 3.1b para o mapa quártico com o parâmetro $b = 1.35$.

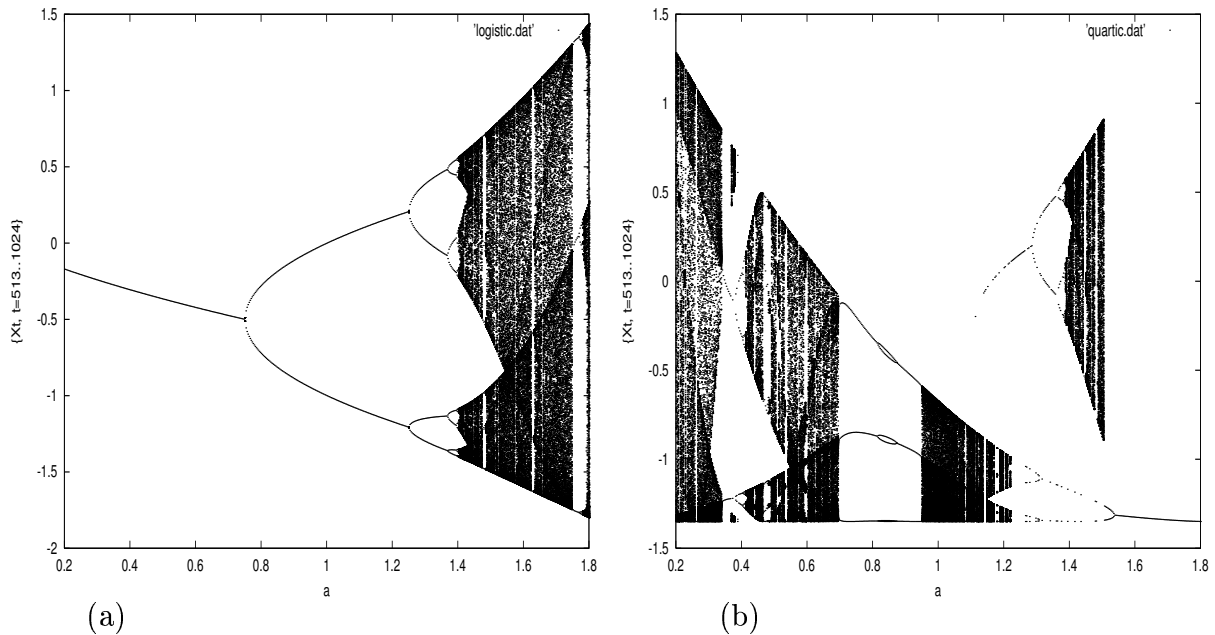


Figura 3.1: Diagramas de bifurcação para o intervalo $a \in [0.2, 1.8]$, com o valor inicial X_0 aleatório no intervalo $[0, 1]$, para o mapa logístico (a); e para o mapa quártico com $b = 1.35$ (b).

O conhecido diagrama de bifurcação do mapa logístico está mostrado na Figura 3.1a, apresenta o atrator único do mapa, para os vários valores do parâmetro a , no intervalo $[0.2, 1.8]$. A Figura 3.1b mostra o mesmo diagrama de bifurcação para o mapa quártico, com o valor do parâmetro $b = 1.35$.

Para o mapa quártico, um diagrama de bifurcação mais detalhado é mostrado na Figura 3.2, e revela o multi-atrator característico do mapa, para $b = 1.35$. Observa-se pelo seu diagrama de bifurcação que, para certos intervalos de a o mapa apresenta dois atratores distintos, como no intervalo $a \in [1.10, 1.55]$; e para outros intervalos, os atratores se tornam coincidentes, como para o intervalo $a \in [0.7, 0.9]$, por exemplo. Essa notável característica do mapa quártico, dá origem a uma nova gama de possibilidades para a dinâmica associada X_t , e será especialmente relevante quando o mapa for usado em

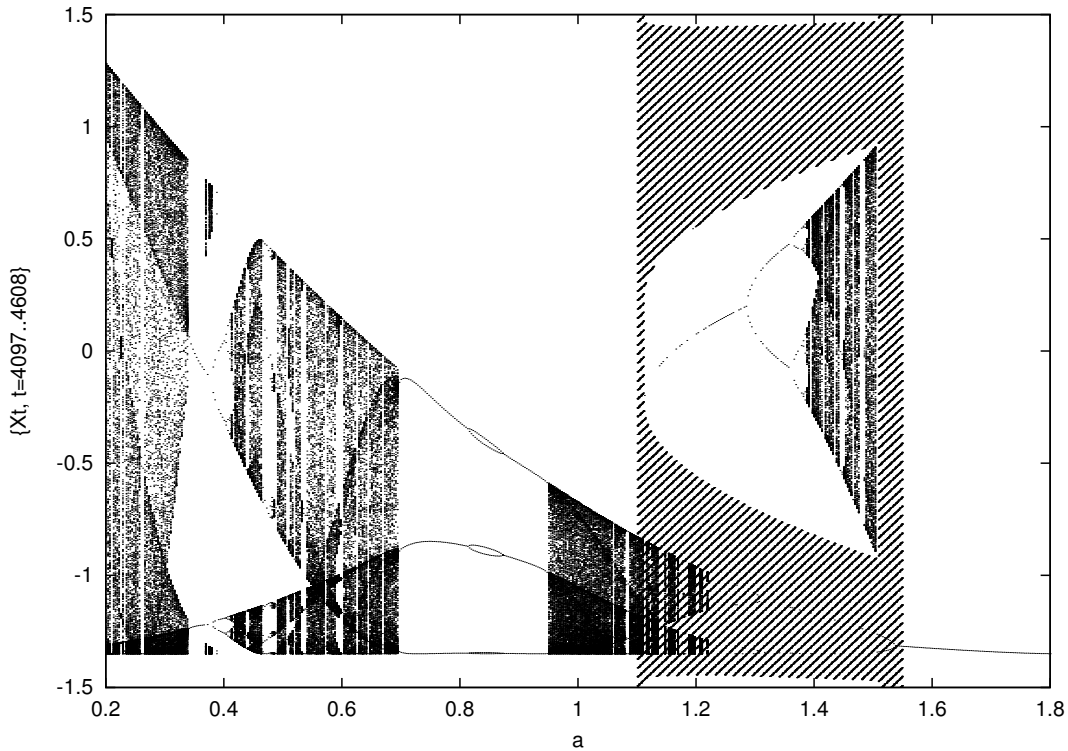


Figura 3.2: Diagrama de bifurcação do mapa quártico, construído como o da Fig. 3.1b, para $b = 1.35$. A estrutura do seu multi-atrator é revelada. Para $a = [1.10, 1.55]$, o atrator depende da condição inicial usada, e para $a = [0.7, 0.9]$, por exemplo, os diferentes atratores coincidem.

uma rede, onde o efeito de difusão introduzido pelo acoplamento de vários mapas induz flutuações na variável X_t suficientemente grandes para que a dinâmica local mude de uma bacia atratora para a outra, já que o mapa possui um *multi-atrator*.

3.3 A Bacia Atratora

Para o intervalo $a \in [1.10, 1.55]$, onde a coexistência de dois atratores distintos é evidenciada, a região hachurada da Figura 3.2 mostra a bacia atratora do mapa quártico para um de seus atratores, isto é, para valores de X_0 nesta região, a série X_t converge para o atrator visto na parte de baixo. Para a região não hachurada desta faixa (branca), a convergência é para o atrator superior, que é semelhante ao atrator do mapa logístico. Ver Figura 3.1a.

A fronteira entre essas duas bacias define uma *separatriz* para os atratores do mapa quártico, neste caso para o valor fixo de $b = 1.35$, assemelha-se à parábola com eixo horizontal mostrada na Figura 3.2.

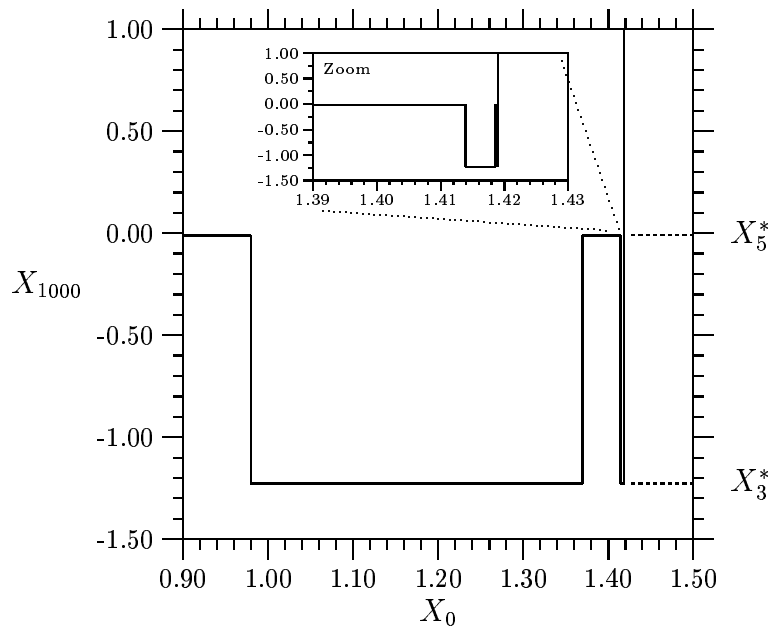


Figura 3.3: A dependência de $X_{t=1000}$ em função de X_0 para o mapa quártico, com parâmetros ($a_0 = 0.35, b_0 = 1.35$), onde um ciclo-2 é encontrado. A janela em zoom mostra detalhes da auto-similaridade encontrada para esse gráfico no limite $X_0 \rightarrow X_{0\infty} = 1.41918530$.

Tomando-se o ponto ($a_0 = 0.35, b_0 = 1.35$) do espaço de parâmetros do mapa quártico, observa-se que a série X_t assim obtida converge para um atrator periódico de ciclo-2 $\{X_3^* = -1.227591, X_5^* = -0.011399\}$, quando inicializado por valores X_0 em $[-1.419185, 1.419185]$.

Ver Tabela

reftab:qmfix. Fora desse intervalo inicial, a série X_t converge rapidamente para o atrator $+\infty$, ou seja, diverge.

O ponto (a_0, b_0) usado como exemplo para o mapa quártico, não foi escolhido por alguma razão especial, exceto porque apresenta um ciclo-2, isto é, o atrator obtido para esse ponto é periódico [BRUNET *et al.*, 1998b].

Da Figura 3.3, encontramos um conjunto infinito $\{X_{0k}\}$ de valores críticos de X_0 separando a convergência das séries X_t em direção de um ou de outro ponto fixo do atrator de ciclo-2. Esta propriedade do mapa é responsável pelas inversões (*flips*) $X_{1000} = X_3^* \leftrightarrow X_{1000} = X_5^*$ vistas na Figura 3.3, à medida que X_0 aumenta de valor. A primeira inversão ocorre em $X_{00} = 0.97924$, a segunda em $X_{01} = 1.36974$, etc. Ver Tabela 3.1. Para o conjunto $\{X_{0k}\}$, encontramos numericamente uma razão limite de convergência

$$\alpha_k \equiv (X_{0(k-1)} - X_{0(k-2)}) / (X_{0k} - X_{0(k-1)}) \quad (3.3)$$

definida no sentido da constante de Feigenbaum, com o valor limite $\alpha \simeq 9.56$.

k	X_{0k}	α_k	k	X_{0k}	α_k
0	0.979236286000	—	7	1.419185234296	9.491831
1	1.369738505500	—	8	1.419185296953	9.405084
2	1.413917597500	8.839073	9	1.419185303587	9.444718
3	1.418627302000	9.380438	10	1.419185304288	9.465116
4	1.419126235000	9.439553	11	1.419185304362	9.555557
5	1.419179051500	9.446537	:	:	:
6	1.419184645000	9.442478	∞	1.419185304388	~ 9.56

Tabela 3.1: Os conjunto dos valores iniciais críticos $\{X_{0k}\}$, para as inversões $X_3^* \rightarrow X_5^*$ para o ciclo-2 do mapa quártico com $a = 0.35$ e $b = 1.35$.

k	X_k^*	Estabilidade	k	X_k^*	Estabilidade
1	-1.33368039	I	5	-0.01139846	E
2	-1.27459667	I	6	+0.27459666	I
3	-1.22759094	E	7	+0.69119334	I
4	-0.97923629	I	8	+1.41918530	I

Tabela 3.2: Os pontos fixos para o diagrama de segundo retorno $X_{t+2} = X_t$ para o ciclo-2 do mapa quártico com $a = 0.35$ e $b = 1.35$. Destes pontos fixos, apenas dois X_3^* e X_5^* são estáveis (E), os outros são instáveis (I).

Na preparação de uma rede de mapas acoplados para simulação, talvez o procedimento mais simples para a inicialização de cada mapa seja a escolha de valores iniciais restritos a algum intervalo $[0, X_{0k}]$, para algum k especificado. Este é um procedimento “seguro”, no sentido de que podemos esperar que nenhum mapa apresente divergência nos passos iniciais da simulação, devido à evolução temporal da variável X_t . Por outro lado, é possível realizarmos a inicialização da rede com valores ligeiramente maiores do que $X_{0\infty}$, sem que haja divergência local, e portanto global, uma vez que na rede os valores assumidos pelos mapas X_t serão promediados com os valores dos mapas vizinhos.

Como o mapa quártico apresenta um ciclo-2 para o ponto (a_0, b_0) , através da análise do seu mapa de segundo retorno $X_{t+2} \times X_t$, pode-se identificar os seus pontos fixos $X_{t+2}^* = X_t^*$, que são em número de oito. Nos gráficos da Figura 3.4, os pontos fixos são os pontos em que a curva intercepta a reta de inclinação igual a 1 (linha pontilhada). Deste conjunto de pontos fixos, identificam-se apenas dois *estáveis*, isto é, intersecções da curva com derivada de módulo menor que 1, ou seja, $|dX_{t+2}/dX_t|_{X_t=X_k^*} < 1$, os outros seis pontos fixos são *instáveis*. Ver Tabela 3.2. Na mesma figura, apresentamos as curvas para outros

dois valores de b , vizinhos do ponto b_0 , que são: $b = 1.30$ e $b = 1.40$, que possuem outros conjunto de valores para os pontos fixos.

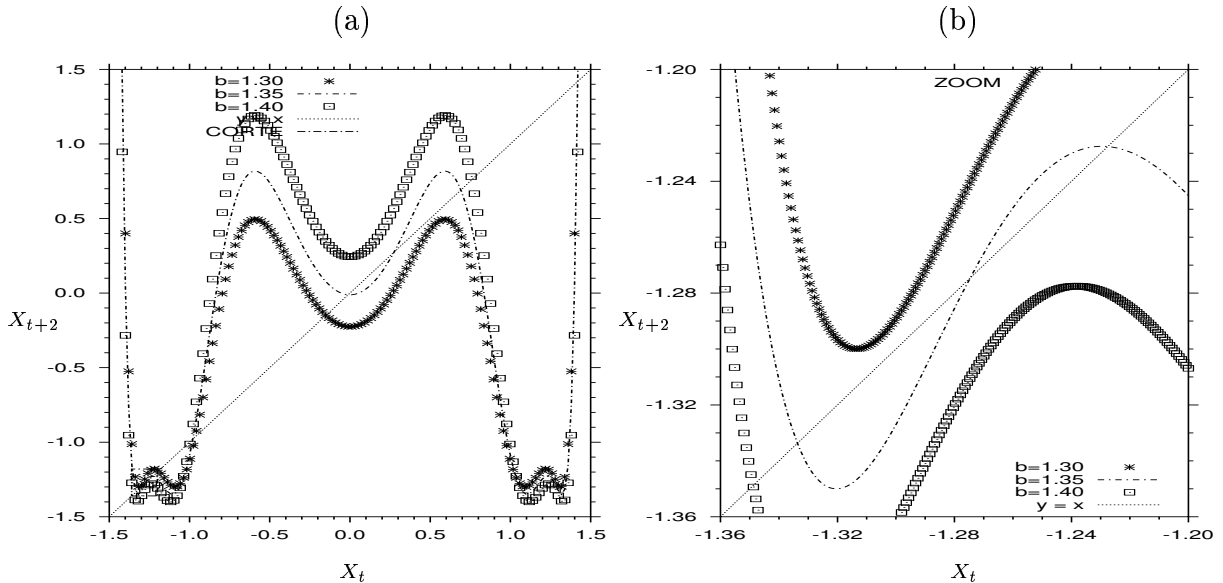


Figura 3.4: Mapa de segundo retorno $X_{t+2} \times X_t$ para o mapa quártico com $a = 0.35$ (a). Vista parcial em “zoom” (b). As intersecções da curva com a reta unitária (linha pontilhada) são os pontos fixos do mapa.

A curva apresentada na Figura 3.5a mostra que para $b = 1.30$ a convergência da série X_t é lenta para um ciclo-2 com os valores $\{-0.16737, -0.13102\}$, e que a derivada da curva de segundo retorno nesse ponto é próxima de $m = -1$. Como se trata aqui de um mapa de segundo retorno, isto significa que o mapa quártico apresenta um atrator finito de ciclo igual a 2 nesse caso. O expoente de Lyapunov do mapa com esses parâmetros é negativo, porém muito próximo de zero (Ver Figura 3.7).

Para $b = 1.40$ o ponto fixo X_5^* não existe mais, como no caso $b = 1.35$, pois a curva não intercepta mais a reta de inclinação unitária. Se testarmos a órbita obtida a partir do valor inicial $X_0 = -0.645$, mostrada nas Figuras 3.5 e 3.6, observamos que para $b = 1.40$ o mapa apresenta uma dinâmica caótica, e para $b = 1.35$ a órbita passa primeiramente por um transiente caótico em torno de X_7^* , mas como esse é um ponto fixo instável, depois de algumas iterações ela acaba caindo na bacia atratora para o ponto fixo X_5^* , que é estável. Observamos aqui que a bacia atratora do transiente caótico que ocorre em torno de X_7^* está fundida (superposta) com a bacia atratora do ponto fixo estável X_5^* . Este fato será fundamental para o entendimento da dinâmica coletiva observada em uma rede de mapas

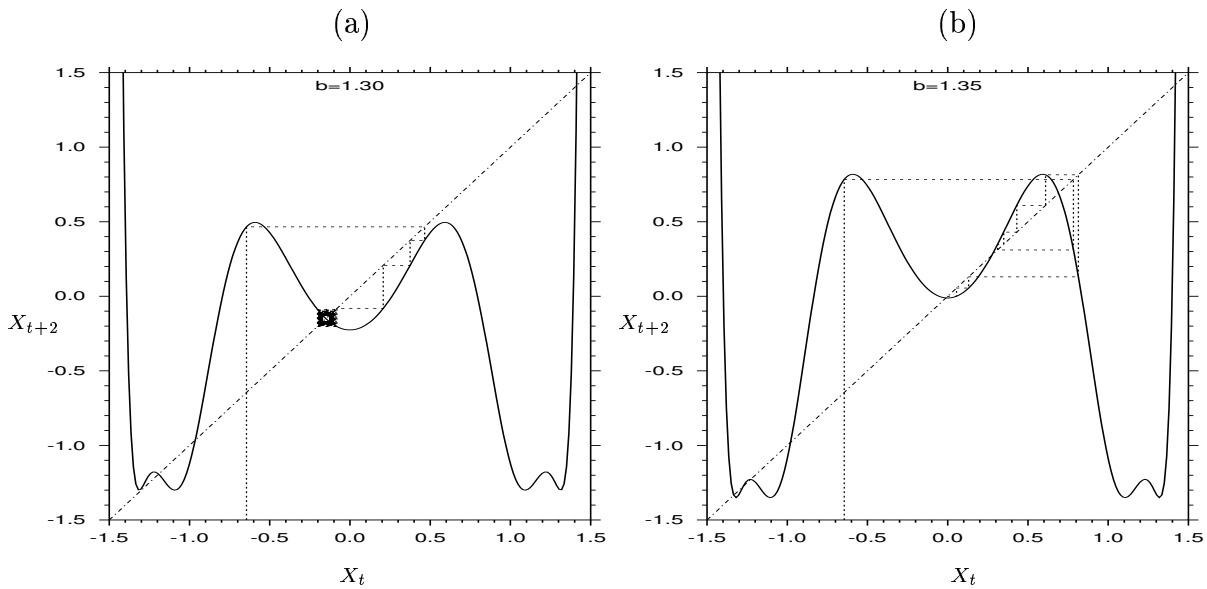


Figura 3.5: Mapa de segundo retorno $X_{t+2} \times X_t$ para o mapa quártico com $a = 0.35$, para (a) $b = 1.30$ e (b) $b = 1.35$, com órbita do ponto inicial $X_0 = -0.645$ mostrada para ambos.

quárticos, o que será discutido no Capítulo 6 deste trabalho¹.

Ainda para $b = 1.35$, a equação de primeiro retorno $X_{t+1} = X_t$, cujas soluções são os valores de X_t que serão mapeados em si mesmos (ciclo-1), obtemos numericamente dois pontos fixos instáveis com os mesmos valores X_4^* e X_8^* mostrados na Tabela 3.2. Para o ciclo-4 do mapa quártico, além dos pontos da tabela, mais quatro pontos fixos instáveis são encontrados.

3.4 A Convergência de X_t

Para o intervalo de convergência finita da série X_t , observa-se que o tempo de convergência não é uniforme para os diferentes valores iniciais X_0 possíveis. O tempo de convergência pode ser estimado pelo número de iterações necessárias para que a série X_t convirja para o atrator, a menos de um certo pequeno erro fixo $\delta = 10^{-6}$, por exemplo. O gráfico da Figura 3.8 apresenta os tempos de convergência para este erro. Como o mapa é definido por uma *função par*, a figura apresenta somente os valores positivos de X_0 .

Outra propriedade relevante do mapa quártico revelada pelo gráfico da Figura 3.8 é a

¹MARTINS L. C.; BRUNET L. G. Heterogeneous multi-state coupled map lattice *Physica A*, artigo submetido à publicação, com a síntese dos resultados dessa dissertação.

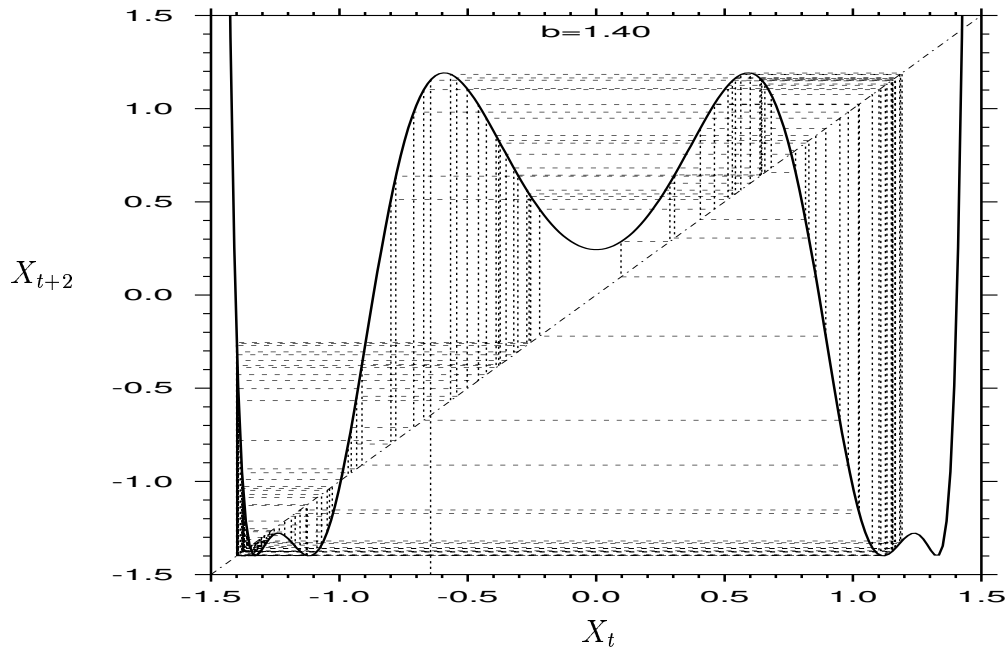


Figura 3.6: Mapa de segundo retorno $X_{t+2} \times X_t$ para o mapa quártico com $a = 0.35$ e $b = 1.40$. O ponto inicial $X_0 = -0.645$ é atraído pelo atrator caótico.

de que o ponto fixo X_5^* nunca é obtido antes do outro ponto fixo X_3^* , do ciclo-2 apresentado pelo mapa. Aqui “antes” significa “com um número de iterações menor do que”. Isto significa que o tempo de convergência para o valor X_3^* é menor ou, em alguns casos igual, ao tempo de convergência para o valor X_5^* .

Esse efeito de *retardo*, que também é encontrado em outros mapas discretos, será relevante ao se tratar uma rede de mapas acoplados, pois a convergência inicial do mapa usado dependerá sensivelmente dessa propriedade de retardo, assim como da inicialização de cada mapa da rede, como será visto também no próximo capítulo.

3.5 O Espaço de Parâmetros

Considerando-se o espaço de parâmetros do mapa quártico, o ponto (a_0, b_0) está localizado em um “braço” da região periódica central que, dependendo do valor inicial X_0 usado para inicializar cada mapa, está conectado ou não a ele, conforme mostra a Figura 3.9. A dependência do X_0 encontrada caracteriza o fato de que existem dois diferentes atratores para a dinâmica do mapa quártico.

Para testar a vizinhança do ponto (a_0, b_0) , exploramos a dinâmica do mapa para

$X_0 \in [0, 1.5]$, separadamente para $a \in [0.3, 0.4]$ e $b \in [1.3, 1.4]$. Os resultados são os mostrados na Figura 3.10, onde a cor cinza indica divergência, a branca atrator caótico e a preta atrator periódico. A Figura 3.9 evidencia as regiões de caos nas vizinhanças do ponto (a_0, b_0) , e as “pontes” para a região periódica central.

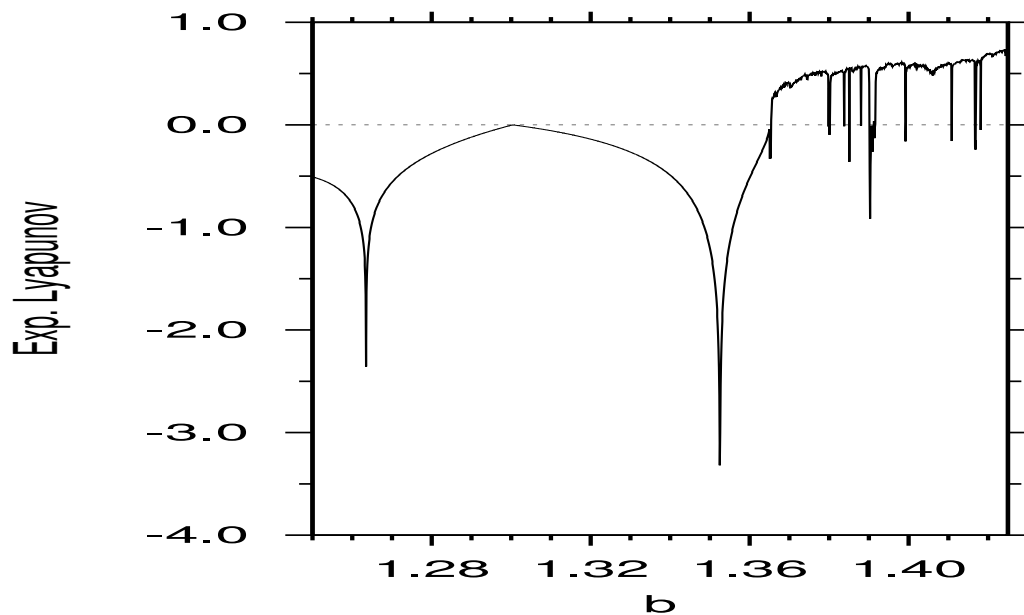


Figura 3.7: O expoente de Lyapunov do mapa quártico para $a = 0.35$.

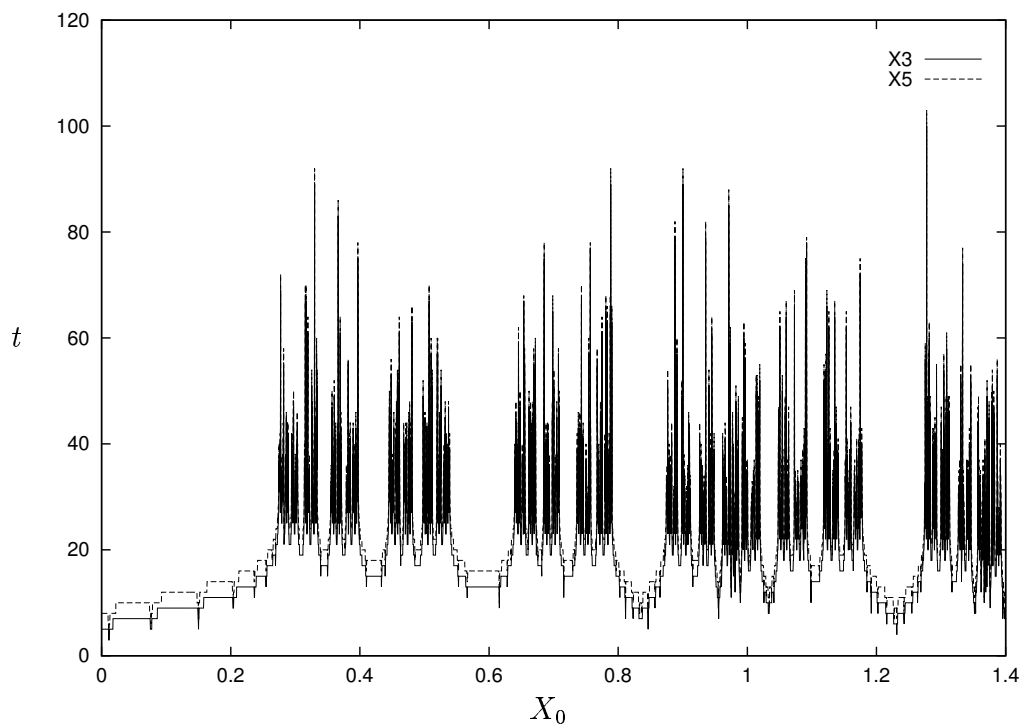


Figura 3.8: Tempo de convergência da série X_t do mapa quártico, para $(a = 0.35, b = 1.35)$, inicializado com diferentes valores X_0 em $[0, 1.4]$.

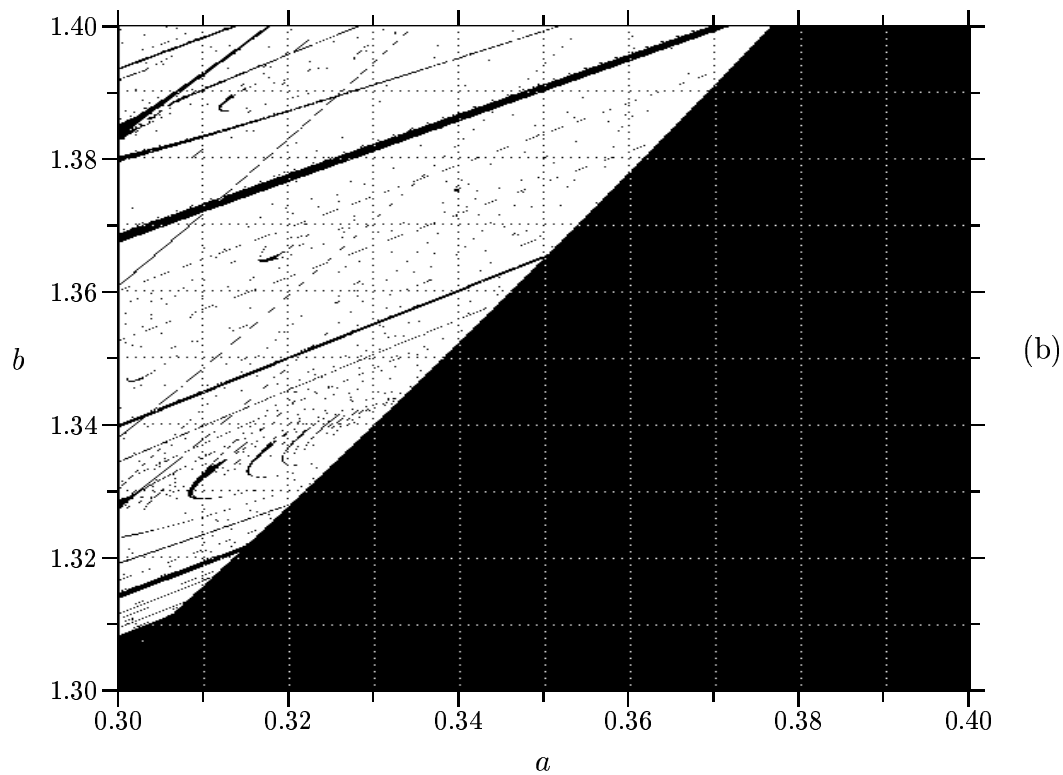
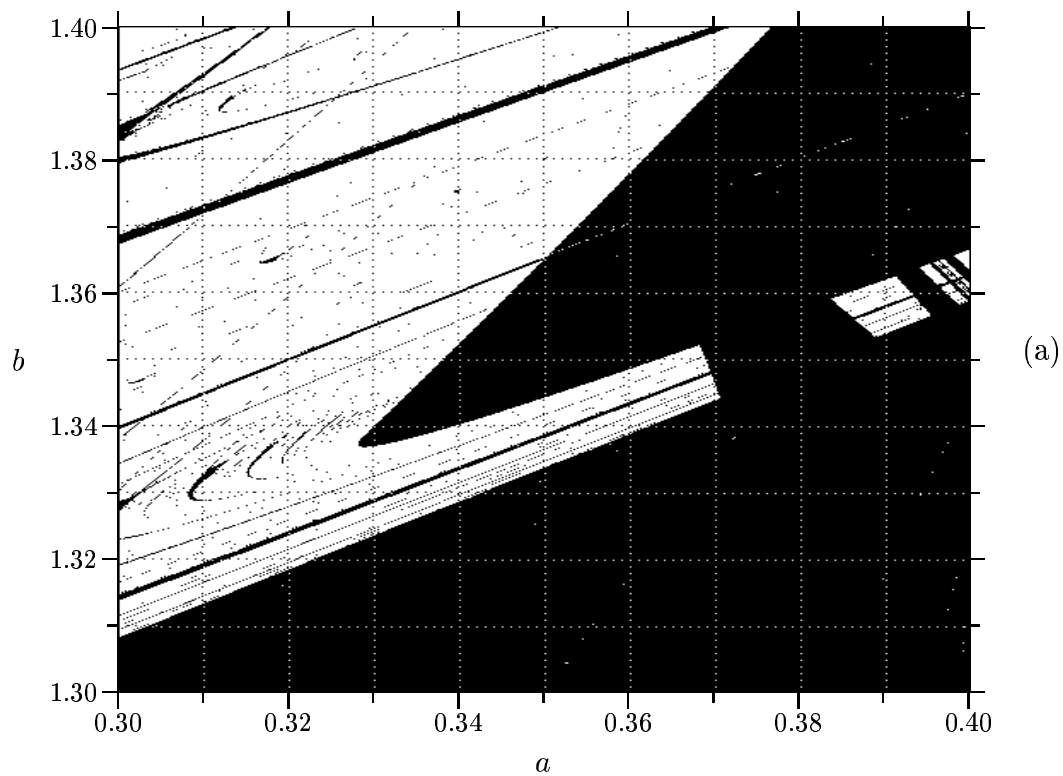


Figura 3.9: Espaço de parâmetros do mapa quártico em “zoom” centrado em $(a_0 = 0.35, b_0 = 1.35)$, para a condição inicial $X_0 = 1.0$ (a), e $X_0 = 0.0$ (b). Nas regiões pretas, o mapa é periódico de ciclo finito, e nas brancas, a dinâmica do mapa é caótica.

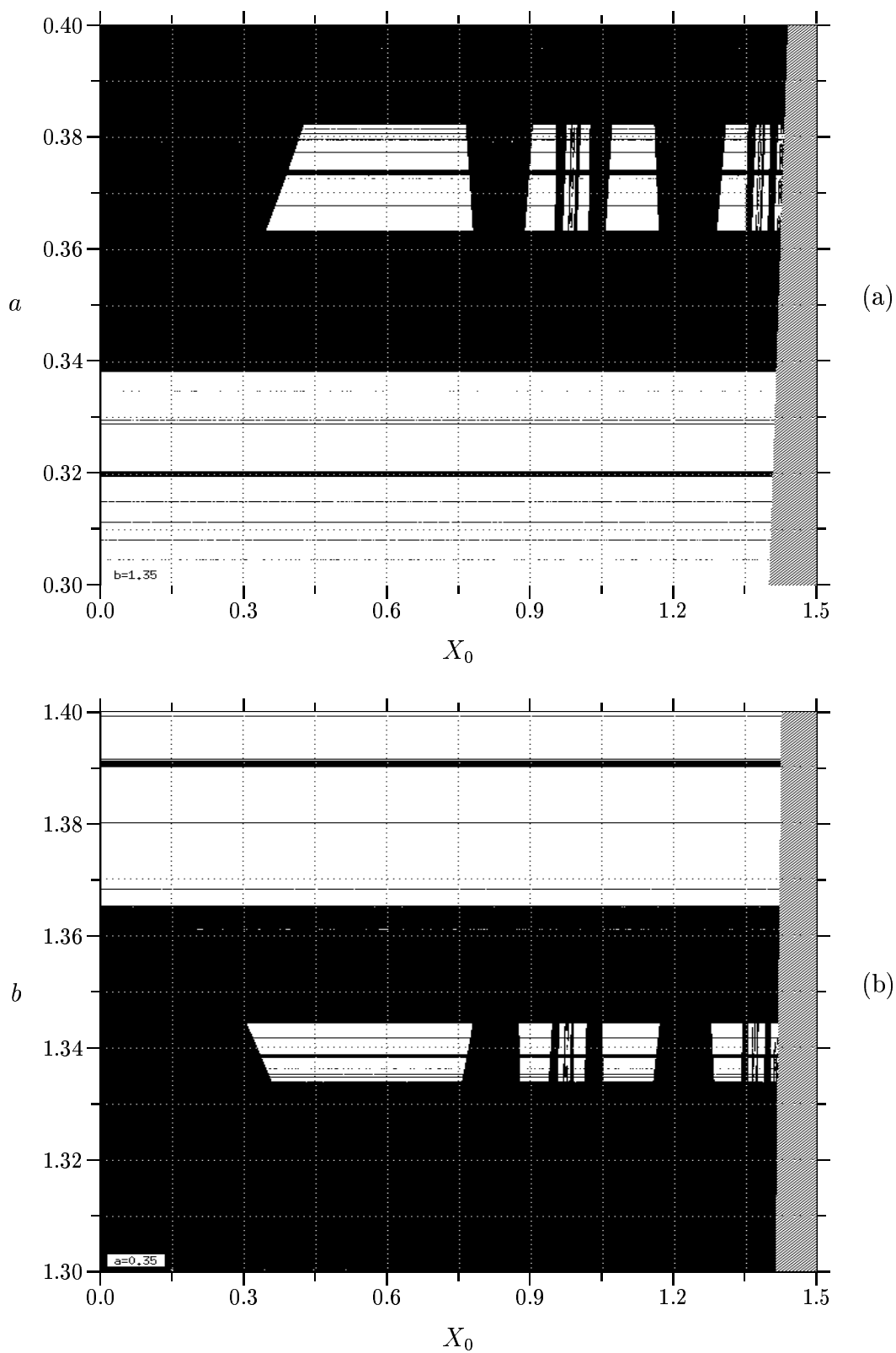


Figura 3.10: A dependência da condição inicial X_0 em torno de (a_0, b_0) , com $b_0 = 1.35$ mantido constante (a); e com $a = 0.35$, constante. Nas regiões cinza (laterais) o mapa diverge, nas pretas a dinâmica é periódica, e nas brancas, caótica.

Capítulo 4

Redes de Mapas Acoplados

4.1 A Rede

Uma rede de mapas acoplados consiste basicamente em um conjunto de mapas definidos sobre uma estrutura espacial, de forma que cada mapa tenha uma posição definida sobre a estrutura, que é chamada de *rede* [CHATÉ, 1998]. Matematicamente, essa estrutura pode ser construída usando-se uma matriz d -dimensional, de forma que cada elemento da matriz represente um mapa da rede, e dessa forma, a matriz de mapas possa ser vista como uma *estrutura espacial discreta*.

Neste trabalho, será tratado apenas o caso em que a rede é bidimensional, e a variável associada a cada mapa da rede é *real* e evolui no tempo de forma discreta, conforme a relação de recorrência que define o mapa local estudado – o mapa quártico.

A *rede de mapas acoplados* consiste, conforme as descrições dadas em [KANEKO, 1989c] e [BRUNET *et al.*, 1998b], em uma matriz bidimensional $N \times N$ de mapas quárticos $X_{t=0}^{i,j}$, com $i, j = 1, 2, \dots, N$, todos definidos num mesmo ponto (a, b) do seu espaço de parâmetros, e inicializados com valores uniformemente distribuídos sobre um intervalo real $[-R, +R]$, para cada mapa. O índice t representa o tempo discreto (iteração) para o mapa local, e os índices $\{i, j\}$ atuam como coordenadas espaciais, definindo a posição do mapa sobre a rede.

Várias condições periódicas de contorno podem ser usadas, porém a mais comum delas é a que faz com que a rede assuma a topologia de um *toróide*, isto é, $X_t^{i \pm N, j \pm N} = X_t^{i, j}$,

para todos os sítios (i, j) da rede, em qualquer instante de tempo t .

4.2 A Evolução da Rede

A rede de mapas acoplados definida na seção anterior modela um sistema espacialmente extenso, cuja dinâmica assintótica ($t \rightarrow \infty$) pode ser obtida numericamente e caracterizada por um conjunto de medidas feito sobre todo o conjunto de mapas do sistema: a média dos valores $\langle X_t \rangle$, o espectro de distribuição para X_t , o desvio quadrático médio de $\langle X_t^2 - \langle X_t \rangle^2 \rangle$, a rugosidade do sistema, a formação espontânea de paredes rígidas separando diferentes tipos de dinâmicas em sub-redes independentes, etc.

Além do fenômeno da sincronização e do movimento coletivo não trivial (*ver seção 4.5*) observado nesse tipo de rede [COSENZA, 1995, CHATÉ *et al.*, 1996], o surgimento espontâneo de sub-redes independentes apresenta uma variedade nova de solução para o estado final do sistema dinâmico. A investigação detalhada das condições para o surgimento e da estabilidade destas sub-redes define o ponto central de interesse deste trabalho, bem como a discussão da robustez e universalidade do mesmo.

4.3 O Algoritmo

O algoritmo usado na simulação da evolução temporal da rede consiste essencialmente em quatro passos executados em um *loop-sem-fim*, depois de inicializada a rede:

i) evolução temporal – cada mapa $X^{i,j}$ evolui no tempo, seguindo a prescrição recursiva de um mapa quártico simples, isto é,

$$X_{t+1}^{i,j} = ((\tilde{X}_t^{i,j})^2 - a)^2 - b \quad (4.1)$$

onde a e b são os parâmetros do mapa, comuns a toda a rede, e que são mantidos fixos para cada simulação.

Uma vez que cada mapa tenha sido atualizado, o segundo passo é realizado:

ii) condição de contorno – para os sítios na fronteira do sistema, os valores dos sítios vizinhos são determinados através de uma condição periódica de contorno, visando

minimizar os efeitos de superfície causados pelo fato do sistema ser finito;

iii) acoplamento local – o valor de cada mapa $X_t^{i,j}$ é promediado sobre o conjunto de mapas vizinhos $\{X_t^{i',j'}\}$,

$$\tilde{X}_{t+1}^{i,j} = (1 - \varepsilon)X_{t+1}^{i,j} + \frac{\varepsilon}{S} \sum_{\{i',j'\}} X_{t+1}^{i',j'} \quad (4.2)$$

onde ε é a constante de acoplamento, e S é o número de vizinhos considerados. Esse tipo de acoplamento é dito *local* e *difusivo* e desta última equação pode-se ver que para $\varepsilon \rightarrow 0$ a trajetória de cada mapa é atraída de volta para a dinâmica comum do mapa local, e os mapas se tornam *desacoplados*, o que faz com que as correlações espaciais nos valores de $X_t^{i,j}$ deixem de existir.

O último termo da equação (4.2) é uma soma sobre todos os S vizinhos $X_{t+1}^{i',j'}$ do mapa central $X_{t+1}^{i,j}$, onde S depende do tipo da vizinhança usada no acoplamento. Neste trabalho, consideram-se dois tipos de vizinhança para o acoplamento local. O primeiro tipo é o acoplamento até *primeiros vizinhos* $S = 4$, onde são incluídos na equação (4.2) os vizinhos de cima, de baixo da esquerda e da direita, para cada mapa. E o segundo tipo, além dos primeiros vizinhos, inclui-se também os outros quatro das diagonais, o que dá $S = 8$ nesse caso, onde o acoplamento é dito até *segundos vizinhos*.

Demonstra-se também que o valor médio de X_t não é alterado pelo acoplamento, isto é, que $\langle \tilde{X}_t \rangle = \langle X_t \rangle$. Assim sendo, a atenuação das irregularidades na variável X_t (campo escalar) imposta pela média tomada no acoplamento acarreta a “difusão” dos valores X_t através da rede, mantendo invariante o seu valor total, neste processo.

Este passo do algoritmo é o que exige maior esforço computacional, consumindo a maior parte o tempo gasto na execução do programa de simulação da rede.

iv) obtenção de dados – dependendo da iteração atual t em que está o sistema, tomam-se as medidas de interesse e as imagens do sistema são salvas para posterior análise.

4.4 A Rede Hexagonal

Para testar as rede de mapas acoplados construída a partir da descrição acima, usamos também um tipo alternativo de acoplamento com a vizinhança, através da introdução de

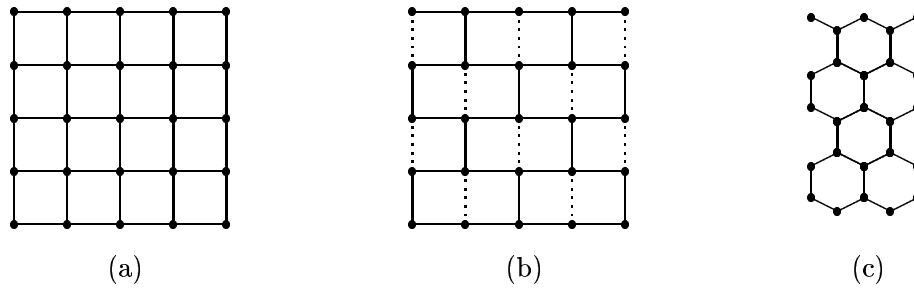


Figura 4.1: A rede hexagonal onde cada vértice possui três conexões ($S = 3$) mostrada em (c), pode ser mapeada sobre uma rede quadrada usual (a), removendo-se desta as conexões alternadas, mostradas em (b) como linhas pontilhadas.

uma rede hexagonal, isto é, no lugar da rede quadrada usual, as conexões entre mapas vizinhos da rede é feita sobre uma geometria hexagonal, até primeiros vizinhos. Com isso, cada mapa passa a ter 3 primeiros vizinhos ($S = 3$), o que torna o acoplamento estruturalmente ainda mais fraco, observando-se o sistema do ponto de vista da topologia da estrutura (rede) sobre a qual está construído. MARCQ *et al.* já fizeram uso de uma regra seletiva como a descrita acima, para a rede quadrada, para redução do número de vizinhos [MARCQ *et al.*, 1997]. Porém, como não encontramos na literatura nenhuma referência explícita a esse tipo de rede de mapas acoplados hexagonalmente, apresentamos a seguir uma descrição sumária deste tipo de rede acoplada, e como pode ser implementada computacionalmente de forma simples, já que a maioria das linguagens de programação dão suporte apenas às matrizes N -dimensionais de números, privilegiando os modelos baseados nas redes quadradas N -dimensionais (cúbicas, quadradas ou vetores unidimensionais), utilizadas com mais frequência.

Do ponto de vista computacional, o problema da construção de uma rede hexagonal, consiste no mapeamento de uma estrutura de conexões hexagonais sobre uma matriz bidimensional, já que estamos interessados nas redes planas. Para isso, observamos que é possível implementar um tal mapeamento sobre uma matriz quadrada $N \times N$ de forma a se obter uma rede hexagonal com a metade desse tamanho, ou seja, com $N \times N/2$, da forma descrita a seguir.

Observando a estrutura hexagonal da Figura 4.1c, vemos que cada vértice possui 3 conexões com a rede, e que se adicionarmos alternadamente uma quarta conexão, linha pontilhada vista na parte (b) da figura, recuperamos a estrutura quadrada usual.

Da forma inversa, podemos mapear a rede hexagonal sobre uma rede quadrada usual, removendo-se alternadamente as conexões de cima/baixo de cada vértice da rede. Nota-se também da figura, que são necessárias duas células da rede quadrada para mapear cada célula da rede hexagonal, daí a redução de tamanho pelo fator 1/2 em uma das direções, no caso, a horizontal. A estrutura assim obtida é topologicamente equivalente a uma rede quadrada usual, como pode ser visto na parte (c) da figura. Mais precisamente, considere-se o vértice (i, j) conectado apenas com o seu “vizinho de cima” $(i - 1, j)$, quando $i + j$ for *par*, caso contrário, apenas a conexão com o “vizinho de baixo” $(i + 1, j)$ é considerada. Em linguagem C, este teste¹ pode ser feito com

```

...
if( ((i+j) % 2)==0 ){                /* caso PAR   */
    ...
}
else{                                /* caso IMPAR */
    ...
}
...

```

se os índices i e j forem usados para localizar o vértice (mapa) (i, j) .

Notas:

1. no código acima, o símbolo “%” indica a operação $\text{MOD}(i+j,2)$, ou seja, a congruência modular.
2. Com esse tipo de algoritmo, a simulação numérica da rede hexagonal se torna mais rápida do que para a rede quadrada, até primeiros vizinhos, por exemplo, já que o teste seletivo “IF” é feito em menor tempo do que o acesso ao valor da variável associada a um quarto vizinho, o que nesse caso é desnecessário.

¹para maior eficiência, a rede bidimensional $X^{i,j=1..N}$ é mapeada em um vetor unidimensional $X^{k=1..N^2}$, onde $k = N * i + j$, e o teste pode ser feito de modo análogo, escrevendo-se i e j a partir de k .

4.5 Caracterização da Dinâmica Assintótica da Rede

Dois tipos distintos de dinâmicas coletivas foram observados nos estados assintóticos das redes estudadas nesse trabalho, que podem ocorrer tanto simultaneamente, separadas por paredes fixas, ou isoladamente, que são a *sincronização* e o *comportamento coletivo não-trivial* (CCNT).

4.5.1 Sincronização

Dizemos que a rede apresenta sincronização quando todos os mapas da rede apresentam o mesmo ciclo-2 do mapa local, ou seja, num dado instante t todos os valores X_t assumem o mesmo valor X_3^* , por exemplo, e no instante seguinte $(t + 1)$, todos os mapas serão levados ao outro valor X_5^* . Neste caso, verifica-se que o desvio quadrático médio $\sigma_X^2(t)$ se torna identicamente nulo, o mesmo ocorre com a rugosidade da rede.

Uma vez que a rede apresente sincronização, a sua dinâmica coletiva não será mais alterada, não sendo mais possível a rede deixar esse estado sincronizado. Neste caso pode-se dizer com certeza que a rede já atingiu o seu estado dinâmico assintótico, que apresenta sincronização ou, simplesmente, que a rede sincronizou.

Em algumas situações a rede pode também apresentar sub-regiões sincronizadas, isoladas do resto do sistema por paredes fixas, formando ilhas dentro da rede. Uma mesma rede pode apresentar várias destas ilhas sincronizadas, sendo que cada uma apresenta sincronização em um dos valores do ciclo-2 local, independentemente uma das outras. Desta forma é possível se verificar casos em que as duas “fases” sincronizadas podem ser vistas simultaneamente sobre a rede, ou seja, num mesmo instante t algumas sub-regiões apresentam-se sincronizadas no valor X_3^* , enquanto outras estão sincronizadas no valor X_5^* . A identificação da presença de sub-regiões sincronizadas pode ser feita através da inspeção direta do espectro $\mathcal{P}[X(t)]$ da rede que, apresentará picos sobre os valores X_3^* e ou X_5^* , quando as respectivas sincronizações existirem no sistema.

Na seção 5.1.1 do próximo capítulo apresentamos uma discussão sobre a dinâmica dos mapas sob uma parede fixa, dividindo uma sub-região sincronizada e outra não.

4.5.2 CCNT

O comportamento coletivo não trivial (CCNT) é o outro tipo de dinâmica observada na rede, ou sub-regiões desta, sendo esta a dinâmica de nosso maior interesse.

Fenomenologicamente, observa-se que para uma rede que apresenta CCNT cada mapa segue uma dinâmica tipicamente caótica, mas o conjunto de todos os mapas apresenta valores com uma distribuição precisa [CHATÉ *et al.*, 1992].

Observa-se também que os estados assumidos pela rede com CCNT nunca se repetem, e depois de algumas iterações da rede (15-20) eles se tornam descorrelacionados. O valor médio instantâneo $\langle X(t) \rangle$ da rede apresenta um ciclo quase periódico, porém com valores diferentes do atrator periódico do mapa local. Em geral, o valor médio pode apresentar um comportamento quase periódico do tipo ciclo- n , onde $n = 4, 5, 6, \dots$. Neste regime coletivo, domina sobre a dinâmica do sistema o transiente caótico imposto pelos pontos fixos instáveis do mapa local, que também é responsável pela perda da correlação temporal observada na rede. Durante o CCNT os valores assumidos pelos mapas rede apresentam uma correlação espacial de curto alcance, com comprimento de correlação da ordem de alguns de sítios, formando um padrão de pequenas regiões coerentes, mas que possuem uma dinâmica temporal intensa, fazendo com que esse padrão mude continuamente no tempo.

Durante o CCNT, a rugosidade da rede cai para valores muito baixos, em torno de 10^{-3} - 10^{-4} , mas nunca se anula. Esse fenômeno se deve ‘a suavização dos valores assumidos pelos mapas da rede ocasionada pelo acoplamento local difusivo.

Nos espectros da rede com CCNT, observa-se distribuições grandes de mapas em torno das regiões próximas aos pontos fixos instáveis X_1^* e X_7^* , devido ao transiente caótico que estes pontos fixos produzem, e o tamanho relativo dessas regiões do espectro indica a presença de uma ou mesmo das duas fases de CCNT simultaneamente sobre o sistema. No caso mais simples a rede alterna entre dois estados de CCNT, apresentando uma das fases de cada vez. Em outros casos, pode apresentar as duas fases ao mesmo tempo, em sub-regiões isoladas por paredes fixas.

No caso mais geral observado na rede de mapas acoplados, observou-se a coexistência de ambas as fases sincronizadas e ambas as fases de CCNT, simultaneamente sobre o

sistema. Note-se que a coexistência de mais de um tipo de fase na na dinâmica assintótica da rede só é possível mediante a presença de paredes fixas estáveis. Na Fig. 4.3 pode-se ver claramente as duas fases de CCNT sobre a rede, onde uma das fases predomina sobre a outra, e nesse exemplo, a fase minoritária irá desaparecer completamente no estado assintótico ($t \rightarrow \infty$). Neste exemplo, não está presente nenhuma sub-região sincronizada.

4.6 Medidas

Numericamente, a evolução da rede de mapas acoplados é feita através de algumas quantidades definidas para toda a rede, obtidas diretamente da matriz $X_t^{i,j}$, enumeradas a seguir.

4.6.1 O Valor Médio $\langle X(t) \rangle$

O valor médio $\langle X(t) \rangle$, definido do modo usual

$$\langle X(t) \rangle = \frac{1}{N^2} \sum_{i,j=1}^N X_t^{i,j} \quad (4.3)$$

tomada sobre os valores da rede no mesmo instante t .

Esta quantidade média, definida para a rede é importante no acompanhamento da evolução coletiva da rede. Essa medida é central para a caracterização de comportamento coletivo, pois nestes casos a dinâmica temporal da média sobre a rede não coincide com a dinâmica presente em cada sítio e as flutuações com relação à média serão não nulas.

4.6.2 O Desvio Quadrático Médio $\sigma_X^2(t)$

O desvio quadrático médio dos valores $X_t^{i,j}$ assumidos pelos mapas da rede numa mesma iteração t , é definido por

$$\sigma_X^2(t) = \frac{1}{N^2} \sum_{i,j=1}^N (X_t^{i,j} - \langle X(t) \rangle)^2, \quad (4.4)$$

é útil para a detecção, e também serve como medida, do fenômeno de sincronização dos valores X_t sobre a rede. Por definição, para uma rede sincronizada, essa quantidade

será nula. Uma vez atingido um estado sincronizado, e quando isso ocorre o fenômeno é rápido (ocorre em poucas iterações), a rede permanecerá indefinidamente nesse estado, não havendo possibilidade de atingir qualquer outro estado assintótico diferente deste.

4.6.3 O Espectro $\mathcal{P}[X(t)]$

É o espectro de distribuição dos valores X_t assumidos pelos mapas na rede, obtido pelo histograma normalizado desses valores para um dado instante (iteração) t .

4.6.4 A Rugosidade $G_\varepsilon(t)$

A *rugosidade* é definida como a média dos quadrados das diferenças dos valores de todas as 'primeiras vizinhanças' da rede, contadas uma vez cada. Para cada mapa $X_t^{i,j}$, há duas diferenças a serem computadas, que podem ser $X_t^{i+1,j}$ e $X_t^{i,j+1}$, por exemplo. Com isso, define-se:

$$G_\varepsilon(t) = \frac{1}{2N^2} \sum_{i,j=1}^N [(X_t^{i+1,j} - X_t^{i,j})^2 + (X_t^{i,j+1} - X_t^{i,j})^2] \quad (4.5)$$

que, como sugere o nome, mede a diferença quadrática média entre os valores assumidos pelos mapas ao longo da rede. Se a rede for visualizada como uma superfície tridimensional $z(i, j; t) \equiv X_t^{i,j}$, onde a altura z é o próprio valor assumido pela rede na posição (i, j) , então o conceito de *rugosidade* se torna mais intuitivo, pois quanto mais irregular a superfície, maior seriam as diferenças a serem somadas em (4.5), e como são computadas ao quadrado, sempre contribuem com valores positivos para a rugosidade. Se toda a rede *sincronizar*, ou seja, todos os mapas assumirem um mesmo valor qualquer num dado instante, a rugosidade automaticamente será zero. No limite $t \rightarrow \infty$ usaremos a notação simplificada G_ε para a rugosidade.

Para o caso de uma rede desacoplada, isto é, para o caso trivial onde $\varepsilon = 0$, a rugosidade pode ser calculada analiticamente. Para o ponto usual (a_0, b_0) , por exemplo, como o mapa quártico apresenta um atrator finito de ciclo-2, todos os mapas da rede acabarão convergindo independentemente para um dos valores $\{X_3^*, X_5^*\}$ do atrator, em tempo suficientemente longo ($t \rightarrow \infty$). Tendo sido todos os mapas inicializados aleatoriamente,

podem-se efetuar as somas da equação (4.5), que neste caso fica reduzida é seguinte forma:

$$G_0 = \frac{1}{2N^2} \sum_{i',j'} [(X_3^* - X_5^*)^2 + (X_5^* - X_3^*)^2] \quad (4.6)$$

$$= \frac{1}{N^2} \sum_{i',j'} \Delta X_{35}^2 \quad (4.7)$$

$$= \frac{1}{N^2} n_{35} \Delta X_{35}^2 \quad (4.8)$$

onde $\Delta X_{35} \equiv |X_5^* - X_3^*|$, e a soma se reduz aos pares vizinhos $\{i', j'\}$ que estão em ciclos diferentes, para que as diferenças não se anulem. Como assumimos que os mapas são inicializados independentemente, e evoluem sem acoplamento, a probabilidade p_{35} de ocorrência de um par de mapas em ciclos diferentes, é a probabilidade conjunta

$$p_{35} = p_3 \times p_5 \quad (4.9)$$

devido à independência dos valores iniciais assumidos pelos mapas, e devido ao ciclo-2 do mapa, $p_3 + p_5 = 1$.

Como existe na rede um total de N^2 mapas, o valor esperado para o número de pares de vizinhos diferentes será

$$n_{35} = p_{35} \times 2N^2 \quad (4.10)$$

e usando a propriedade (4.9), a rugosidade pode então ser escrita como:

$$G_0 = 2p_3(1 - p_3)\Delta X_{35}^2. \quad (4.11)$$

Avaliando-se p_3 numericamente, a expressão acima pode ser usada para o cálculo direto da rugosidade. Para uma rede com $N = 256$, o valor obtido foi $p_3 = 0.279325$, o que dá para a rugosidade o valor $G_0 = 0.595502$, quando $\varepsilon = 0$, para valores iniciais dos mapas foram escolhidos com distribuição uniforme no intervalo $[-1.4, 1.4]$. Esse tipo de inicialização foi chamada de RANDOM, neste trabalho. Para redes maiores, veja a Tabela 4.1.

De forma análoga, o valor médio $\langle X(t \rightarrow \infty) \rangle \equiv \langle X \rangle$ para a rede desacoplada será

$$\langle X \rangle = p_3 X_3^* + p_5 X_5^*, \quad (4.12)$$

N	p_3	G_0
256	$0.28005981 \pm 0.00045738$	$0.59646070 \pm 0.00059685$
512	$0.28049850 \pm 0.00026404$	$0.59703099 \pm 0.00034527$
1024	$0.27897263 \pm 0.00008636$	$0.59504247 \pm 0.00011290$
2048	$0.27889276 \pm 0.00005533$	$0.59493800 \pm 0.00007239$
4096	$0.27894998 \pm 0.00001950$	$0.59501285 \pm 0.00002550$

Tabela 4.1: Valores numéricos para a probabilidade p_3 e a rugosidade g_0 , para redes de vários tamanhos, com distribuição uniforme no intervalo $[-1.4, 1.4]$.

e na iteração seguinte da rede, após a inversão dos valores dos mapas imposta pelo ciclo-2,

$$\langle X \rangle = p_3 X_5^* + p_5 X_3^* . \quad (4.13)$$

Para a inicialização RANDOM, substituindo os valores correspondentes nas equações acima obtemos o ciclo-2 para a média $\langle X \rangle = \{-0.887878, -0.351111\}$.

A rugosidade definida pela equação (4.5) depende implicitamente da constante de acoplamento ε , através da difusão imposta aos valores $X_t^{i,j}$ pelas médias locais, e também da amplitude da distribuição uniforme inicial usada para inicializar a rede, isto é, do intervalo $[-RAD, RAD]$ usado. Dependendo da amplitude usada, as bacias atratoras dos pontos fixos X_3^* e X_5^* serão diferentemente populadas no início da simulação, o que será determinante no tipo de dinâmica final atingida pela rede. Assim sendo, é imediato que $G(t \rightarrow \infty)$ dependa diretamente do valor da amplitude (RAD) usada para a inicialização da rede. Devido à forma da dependência da bacia atratora do mapa quártico mostrada na Figura 3.3, conclui-se que a competição entre os atratores X_3^* e X_5^* deve-se à distribuição inicial dos valores assumidos pelos mapas da rede, e também do valor da constante de acoplamento ε , uma vez que é o acoplamento difusivo o responsável pelo fato de que um mapa mude de atrator durante a evolução da rede, o que define toda a complexidade de padrões obtidos para valores grandes desse parâmetro. Para valores grandes de RAD (≥ 1.2) e $\varepsilon > 0$ espera-se que a população inicial dos atratores do mapa local seja responsável pela definição da dinâmica final da rede com um todo, uma vez que a bacia atratora local tem a estrutura auto-similar mostrada na Figura 3.3, e os mapas iniciados em valores próximos a RAD serão localizados em estreitas bandas atratoras da bacia local, tornando-os bons candidatos a mudarem várias vezes de atrator, antes que o transiente global da

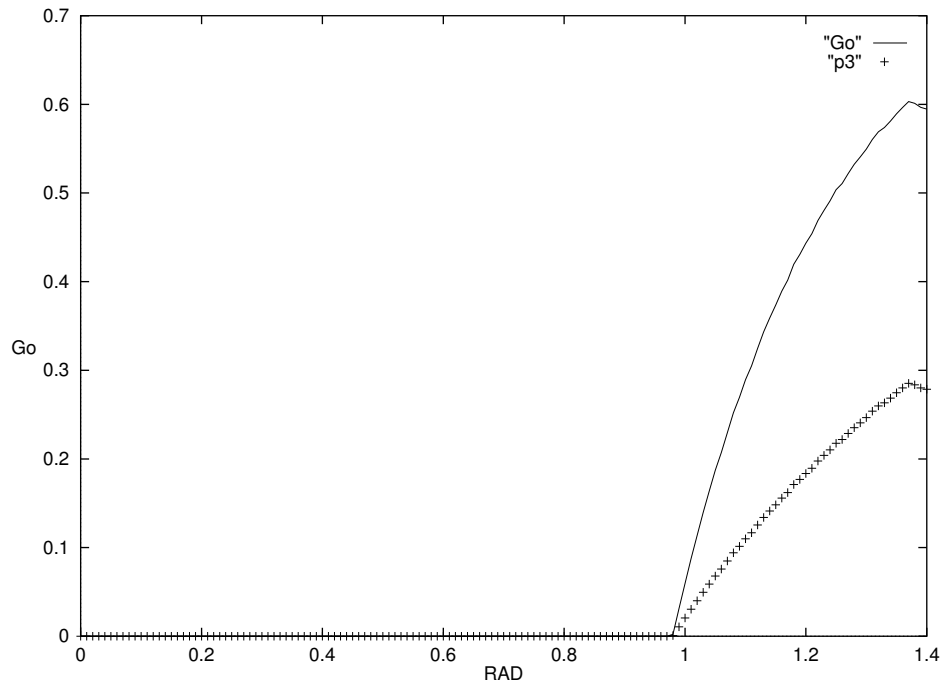


Figura 4.2: Dependência da rugosidade e p_3 , para a rede desacoplada ($\varepsilon = 0$) de tamanho 512×512 , iterada 256 vezes ($t = 256$), em função da amplitude RAD usada para a distribuição inicial da rede. Note-se que p_3 é proporcional à G_0 , nesse caso.

rede termine, e a dinâmica global tome conta do sistema.

Como um caso particular, a Figura 4.2 mostra a dependência da rugosidade da rede desacoplada ($\varepsilon = 0$) de tamanho 512×512 , iterada 256 vezes ($t = 256$), em função da amplitude RAD usada para a distribuição inicial da rede. A probabilidade p_3 também é mostrada no mesmo gráfico, de onde se pode ver que esta é proporcional à rugosidade G_0 .

Outra forma de inicialização da rede que utilizamos neste trabalho, que chamamos de RING50%, foi a distribuição equiprovável dos valores iniciais $X_{t=0}^{i,j}$, de forma que 50% dos mapas são inicializados com valores da bacia atratora para X_3^* , e os outros 50% com valores atraídos para X_5^* . Para esse tipo de inicialização, a equação (4.11) resulta em $G(t \rightarrow \infty)_{50\%} = \Delta X_{35}^2/2$, já que $p_3 = p_5 = 1/2$. Para este caso, temos numericamente o resultado $G_0 = 0.7395615$, para a rede sem acoplamento ($\varepsilon = 0$), e as médias assumem um mesmo valor final $\langle X \rangle = 0.619495$, que é a média aritmética dos valores dos pontos fixos X_3^* e X_5^* .

4.7 Exploração Preliminar da Rede

Não menos importante que a obtenção dos dados numéricos para a análise quantitativa da rede a que nos propomos a estudar nesse trabalho, optamos inicialmente pela exploração qualitativa das propriedades gerais da rede, onde pode-se identificar visualmente os diferentes padrões dinâmicos assumidos coletivamente pelo sistema, para as diferentes geometrias, vizinhanças e intensidades de acoplamento usados.

O acompanhamento da evolução coletiva da rede é feito visualmente, através da discretização e mapeamento linear dos valores $X_t^{i,j}$, sobre uma paleta suave de 256 cores, que possa ser visualizada durante a execução do programa. Com isso, a matriz $N \times N$ usada para a simulação da rede passa a ser visualizada como uma imagem digital formada por um quadrado de $N \times N$ *pixels* (ou pontos gráficos), onde cada mapa é representado por um *pixel*, e *vice-versa*, numa relação bi-unívoca.

A inspeção visual da rede é muito importante, pois fornece uma visão geral de como ela se comporta globalmente, e permite verificar se existem regiões sincronizadas, regiões caóticas ou se as duas situações ocorrem simultaneamente, em diferentes partes do sistema.

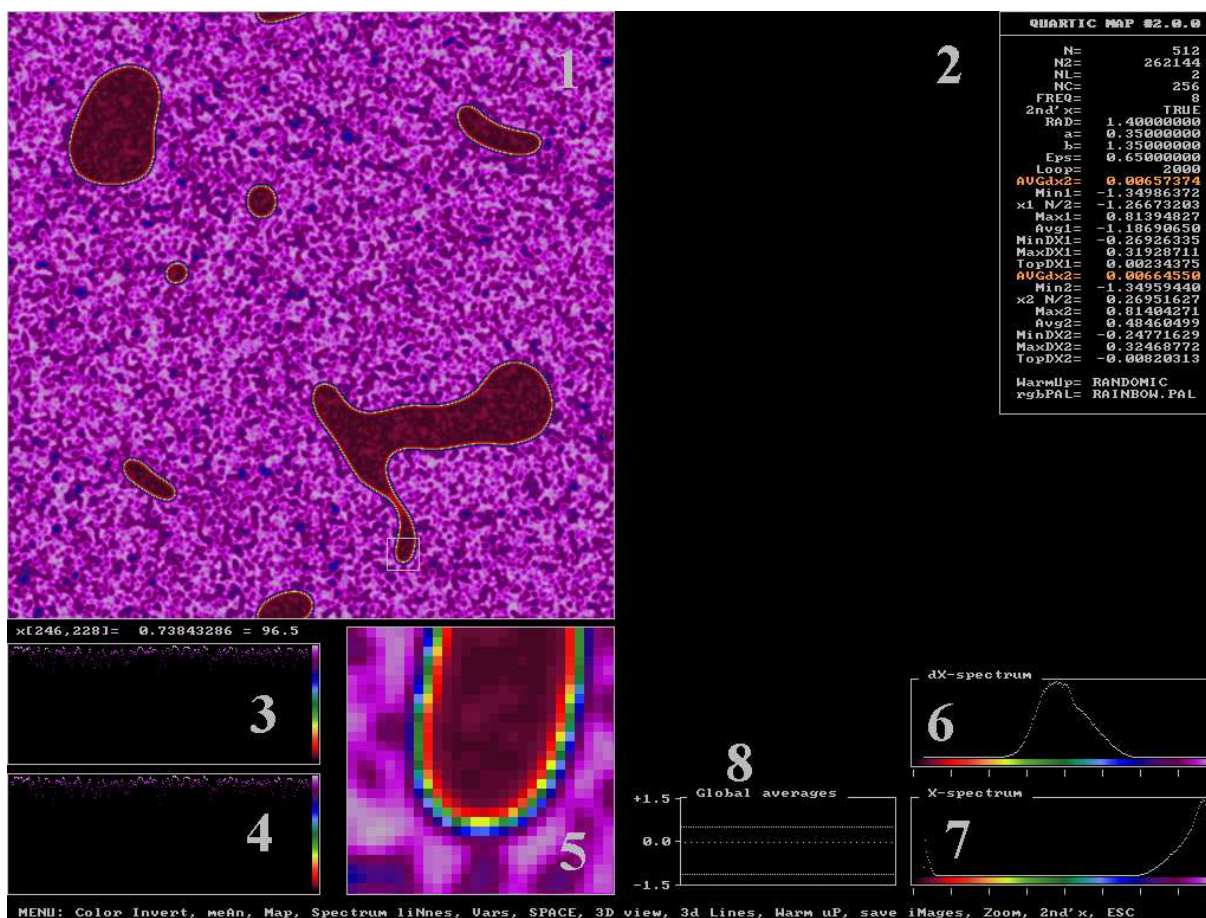
Para essa abordagem inicial, desenvolvemos o software `qmcm1-2.0.0.c` que é totalmente interativo, e possui uma interface gráfica bastante intuitiva, capaz de permitir a variação de praticamente todos os parâmetros relevantes para a simulação da rede, inclusive o tamanho da rede.

A Figura 4.3a mostra a tela básica do programa `qmcm1-2.0.0.c`, onde identificamos as principais informações que ele nos apresenta, como:

1. Mapeamento gráfico da rede, isto é, os elementos da matriz $X_t^{i,j}$, são coloridos através da paleta de cores arco-íris (`rainbow.pal`). O valor mínimo assumido por $X_t^{i,j}$ é mapeado na cor preta (0), e o máximo na cor branca (255). Os valores intermediários são mapeados linearmente através do espectro de cores da paleta. Nessa mesma parte da tela, o programa permite também a visualização tridimensional da rede, o que é mostrado no detalhe da Figura 4.4;
2. Os valores numéricos dos parâmetros fixos (constantes) e variáveis da simulação;
3. Vista em corte transversal da parte central da rede ($i = N/2$), para a iteração atual

(t);

4. Idem ao anterior, só que para a iteração anterior ($t - 1$);
5. Vista da rede ampliada, em detalhe (zoom= $8X$);
6. Espectro de distribuição dos valores de $\Delta X_t^{i,j} \equiv X_t^{i,j} - X_{t-1}^{i,j}$, com escala X arbitrária, usado para visualização da convergência do espectro de X . Ver próximo item.
7. Espectro de distribuição dos valores de $X_t^{i,j}$. Espera-se que este espectro assumira um perfil invariante, à medida que a rede converge para o seu estado final no limite ($t \rightarrow \infty$);
8. Gráfico da evolução temporal do valor médio $\langle X_t \rangle$. A convergência assintótica para um ciclo-2 é aqui observada como um fenômeno coletivo resultante do ciclo-2 apresentado pelo mapa local, apesar dos valores médios serem em geral diferentes de X_3^* e X_5^* , este comportamento final da média sempre é observado.



A Paleta rainbow.pal:



Figura 4.3: Interface gráfica do programa qmcml-2.0.0.c, usado para a exploração interativa da rede (a). A paleta rainbow.pal de 256 cores usada para a visualização da rede.

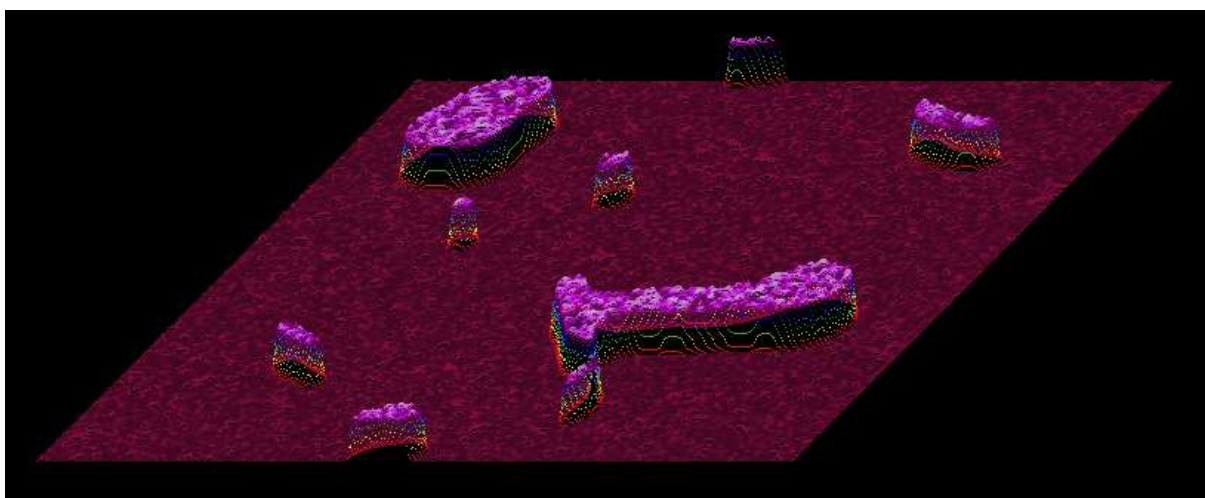


Figura 4.4: Visualização tridimensional da rede na iteração seguinte à mostrada na Figura 4.3.

Capítulo 5

Resultados

Apresentamos a seguir os resultados obtidos para a dinâmica assintótica da rede de mapas quárticos acoplados, obtido a partir do programa `qmcml-2.0.0.c`. Foram testados quatro tipos diferentes de inicialização para a rede, que foram chamadas: RANDOM, RING50%, PATCH3 e PATCH5. Essas inicializações foram usadas para os acoplamentos até segundos vizinhos ($S = 4$ e $S = 8$); para a rede hexagonal, testamos apenas o acoplamento até primeiros vizinhos ($S = 3$). Em todos os casos, foram usadas redes de tamanho 512×512 , e foram feitas 10.000 iterações para cada rede. Estas simulações foram feitas para valores da constante de acoplamento no intervalo $\varepsilon \in [0, 1]$, assumindo valores a cada $\Delta\varepsilon = 0.01$.

Especial ênfase foi dada na análise do primeiro dos casos descritos abaixo, o caso da rede quadrada inicializada pelo método RANDOM e considerando acoplamento até primeiros vizinhos, já que muitos dos resultados obtidos para os outros casos são comuns ou similares aos deste, e então procuramos ressaltar as diferenças encontradas, quando for o caso, e comparar as situações similares.

Ao final deste capítulo, os principais resultados são apresentados de forma resumida na Tabela 5.3, facilitando a comparação dos diferentes estados dinâmicos assintóticos obtidos para as diferentes combinações de inicialização, geometria de rede, ordem de acoplamento e valores de acoplamento usados.

5.1 Inicialização RANDOM

Para esse tipo de inicialização, usamos uma distribuição uniforme de valores iniciais $X_{t=0}^{i,j}$ escolhidos aleatoriamente no intervalo $[0, 1.4]$. Inicialmente, para a rede desacoplada ($\varepsilon = 0$), observamos que a configuração final da rede depende apenas da sua configuração inicial, já que o mapa local é periódico de ciclo-2. Além disso, como a inicialização RANDOM privilegia a bacia atratora local para o ponto fixo X_5^* , o que pode ser visto na Figura 3.3, cerca de 72% dos mapas são inicializados na bacia de atração para X_5^* e esse percentual será mantido para a rede ao longo do tempo, já que não há acoplamento. Nas iterações pares $t = 0, 2, 4, \dots$ esse percentual de mapas será encontrado convergindo para o valor X_5^* , e o resto dos mapas convergindo para X_3^* . Nas iterações ímpares $t = 1, 3, 5, \dots$, os percentuais de mapas que convergem para X_3^* e X_5^* serão intercambiados um pelo outro. Coletivamente, a rede assume um comportamento assintótico do tipo ciclo-2, já que dois estados fixos serão visitados alternadamente.

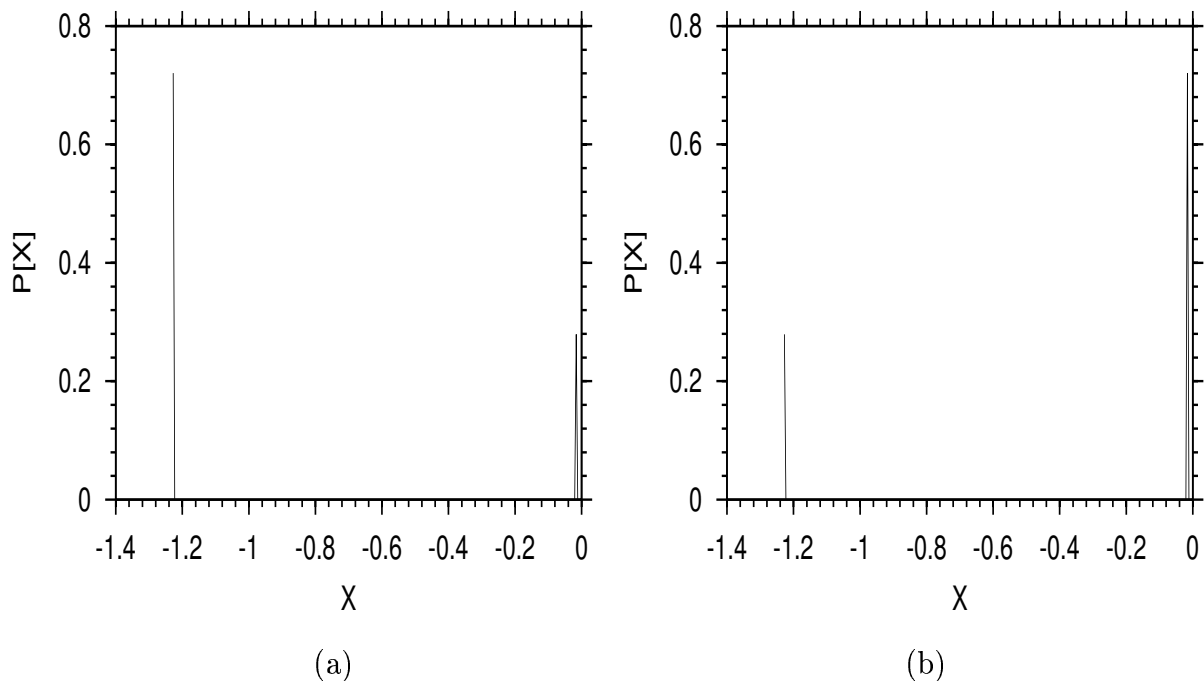


Figura 5.1: O espectro $\mathcal{P}[X(t)]$ para ($\varepsilon = 0$), nos instantes $t = 9999$ (a), e $t = 10000$ (b). Os picos observados estão sobre os valores fixos X_3^* e X_5^* do mapa local.

Como a convergência é rápida para o mapa quártico, apesar das irregularidades (picos) apresentadas na Figura 3.8, após cerca de 100 iterações a configuração final da rede se

torna praticamente invariante, e os estados finais obtidos são como os mostrados abaixo:

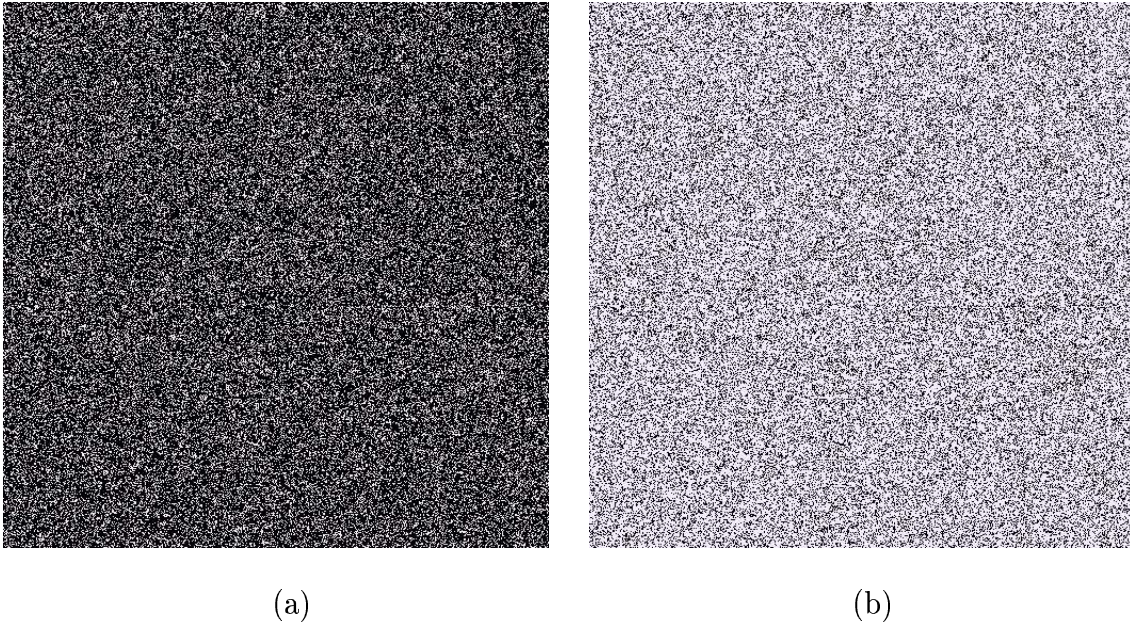


Figura 5.2: *Estados finais da rede de mapas quárticos desacoplada ($\varepsilon = 0$), em duas iterações sucessivas $t = 9999$ (a), e $t = 10000$ (b).*

Para esse caso, o valor obtido para rugosidade final da rede desacoplada foi a mesma calculada em 4.6.4, isto é, $G_0 = 0.595502$. Mesmo para valores de ε ligeiramente acima de zero, no intervalo $\varepsilon \leq \varepsilon'$, com $\varepsilon' = 0.0027$, o comportamento da rede é o mesmo do caso $\varepsilon = 0$, independente da geometria e ou ordem de acoplamento, e não apresenta nenhuma diferença para o seu estado assintótico. Neste estreito intervalo da constante ε o efeito do acoplamento é irrelevante, se comparado com o efeito do atrator local do mapa quártico, que domina as órbitas individuais dos mapas da rede, determinando por si só a dinâmica coletiva da rede.

5.1.1 Rede Quadrada, Primeiros Vizinhos

Para valores do acoplamento no intervalo $\varepsilon' \leq \varepsilon \leq 0.40$, a rede apresenta um estado assintótico *bloqueado*, isto é, existem pequenas sub-regiões da rede que apresentam movimento coletivo não trivial isoladas de outras sub-regiões sincronizadas por uma estrutura de paredes fixas. As irregularidades iniciais em X_0 induzidas pela escolha aleatória dos valores dos mapas da rede, agora que o acoplamento não pode mais ser desprezado, induzem a uma competição entre as diferentes fases possíveis rede. Nas primeiras iterações

da rede, o acoplamento de sítios vizinhos com valores próximos cria núcleos de formação de sub-regiões em comportamento coletivo e sub-regiões sincronizadas separadas por um intrincado padrão de paredes fixas, como os da Figura 5.3.

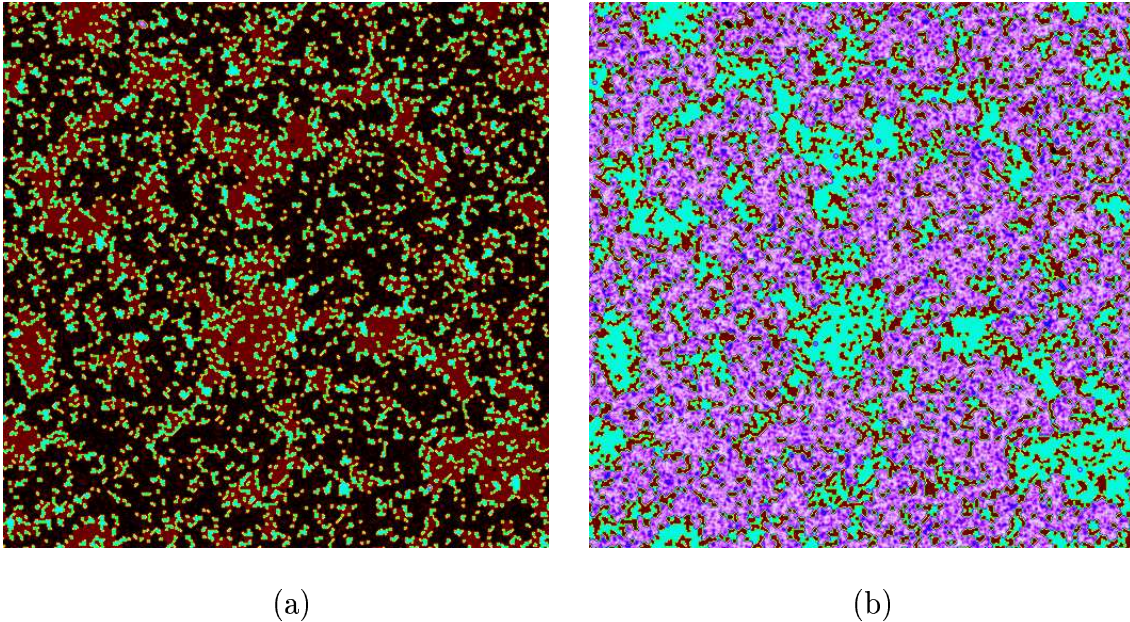


Figura 5.3: Os instantâneos da rede em $t = 9999$ (a), e $t = 10000$ (b), para $\varepsilon = 0.40$.

Para os estados mostrados na Figura 5.3 os espectros de distribuições dos valores de X_t correspondentes são, respectivamente, os mostrados na Figura 5.4.

Para valores de ε nos intervalos $[0.40, 0.50)$ e $[0.69, 1.00]$ observamos que as pequenas sub-regiões de comportamento coletivo juntam-se em regiões maiores separadas por paredes fixas bem definidas onde as duas fases do ciclo coletivo aparecem simultaneamente em diferentes partes do sistema, coexistindo com regiões de sincronização nos pontos fixos X_3^* e X_5^* . No caso mais geral, os quatro diferentes tipos de regiões podem ser encontradas simultaneamente na rede. Para esse intervalo de acoplamento, observamos que a redução no número de paredes resultante do aumento de tamanho das sub-regiões que compõe o sistema faz com que a rugosidade da rede varie de forma linear com o acoplamento ε , conforme pode ser visto na Figura 5.5.

Acima dessa região de acoplamento, para $0.50 \leq \varepsilon \leq 0.60$, as paredes formadas não são mais estáveis e as diferentes sub-regiões acabam se fundindo em uma única. Não há mais regiões sincronizadas e a fase majoritária do ciclo coletivo acaba se impondo sobre a outra. Nesse caso, a rugosidade cai a valores bem pequenos, dando origem ao patamar

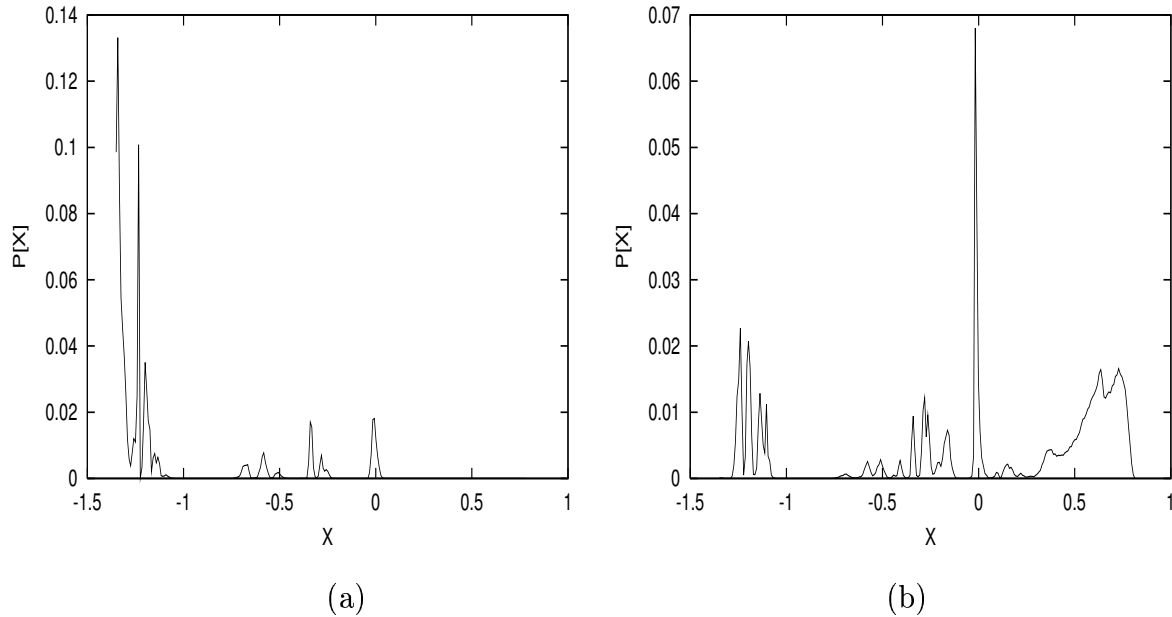


Figura 5.4: As distribuições dos valores de X_t , para $t = 9999$ (a), e $t = 10000$ (b), para $\varepsilon = 0.40$.

em torno de $G \simeq 2 \times 10^{-3}$ para este intervalo.

Para o ponto particular $\varepsilon = 0.54$, a rede apresentou uma dinâmica coletiva diferente, com uma região sincronizada e outra com comportamento coletivo não trivial, associada ao outro ponto fixo do mapa local. Na interface das regiões nenhuma parede é identificada visualmente, apesar da estabilidade desse padrão assumido pelo sistema. *Ver Figura 5.6a.* É possível também que presença da região sincronizada indique que o estado assintótico do sistema não tenha sido atingido, pois estamos próximos ao limiar de coexistência dos atratores para o comportamento coletivo e para o regime sincronizado.

Essa situação particular pode ser acompanhada também pelo espectro mostrado na Figura 5.6b, onde pode-se identificar claramente o pico associado ao ponto fixo X_5^* (região sincronizada) e a distribuição em torno de X_7^* (região caótica).

Finalmente, para o intervalo $0.61 \leq \varepsilon \leq 0.68$ a rede volta a apresentar paredes fixas bem definidas, separando diferentes fases do mesmo comportamento coletivo ciclo 2; o predomínio de uma fase sobre a outra deve-se à inicialização que privilegia um dos atratores (pontos fixos) do mapa local. Para o ponto $\varepsilon = 0.65$ deste intervalo, a Figura 5.8 apresenta o estado da rede (a) e o espectro correspondente para X_t (b).

O fato dos espectros de X_t modificarem-se de forma suave, ao longo da direção ε (ver

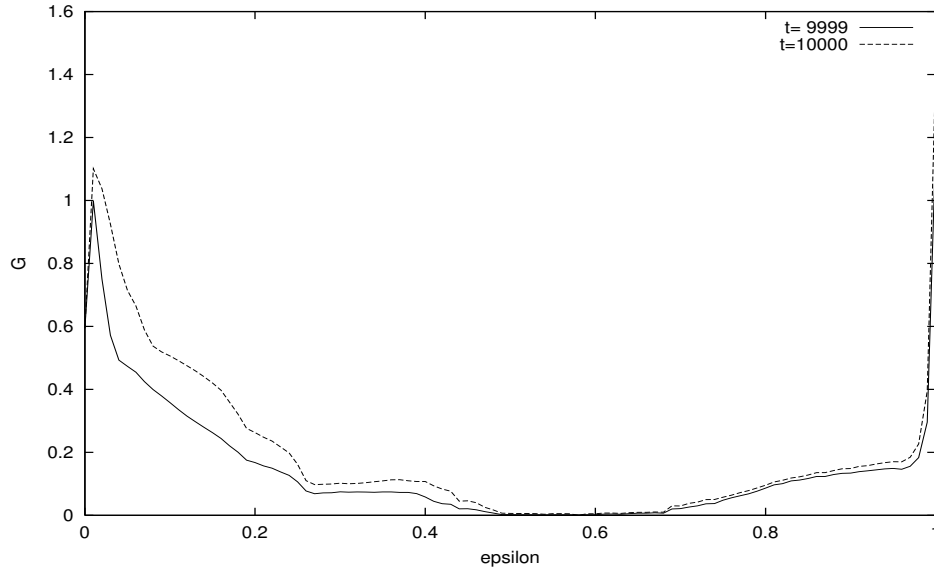


Figura 5.5: Gráfico da rugosidade $G \times \varepsilon$ para a rede quadrada, primeiros vizinhos e inicialização *RANDOM*.

Figura 5.9) mostra que a dinâmica assintótica apresentada pela rede é *robusta* no sentido de que não depende da distribuição inicial usada, mas sim do método de inicialização. Esta idéia será reforçada pelos resultados obtidos para os outros tipos de inicialização estudados, o que será apresentado nas seções seguintes.

Para valores grandes da constante de acoplamento, no intervalo $\varepsilon > 0.9$, o enfraquecimento do primeiro termo do lado direito da equação (4.2), que é proporcional a $1 - \varepsilon$, faz com que o segundo termo desta equação tenha uma importância relativa maior tornando-se dominante no cálculo da média. Desta forma, o valor $\tilde{X}_t^{i,j}$ acaba se tornando independente do valor local $X_t^{i,j}$ no limite $\varepsilon \rightarrow 1$. Nesse limite, o novo valor da variável local, obtido pelo acoplamento, depende apenas dos valores dos mapas vizinhos.

Se escrevermos a equação (4.2) para três mapas vizinhos, digamos $X_t^{i,j}$, $X_t^{i,j+1}$ e $X_t^{i+1,j+1}$, obteremos respectivamente:

$$\tilde{X}_t^{i,j} = (1 - \varepsilon)X_t^{i,j} + \frac{\varepsilon}{4}(X_t^{i-1,j} + \underbrace{X_t^{i+1,j}}_2 + X_t^{i,j-1} + \underbrace{X_t^{i,j+1}}_1) \quad (5.1)$$

$$\tilde{X}_t^{i,j+1} = (1 - \varepsilon)X_t^{i,j+1} + \frac{\varepsilon}{4}(X_t^{i-1,j+1} + X_t^{i+1,j+1} + X_t^{i,j} + X_t^{i,j+2}) \quad (5.2)$$

$$\tilde{X}_t^{i+1,j+1} = (1 - \varepsilon)X_t^{i+1,j+1} + \frac{\varepsilon}{4}(\underbrace{X_t^{i,j+1}}_{1'} + X_t^{i+2,j+1} + \underbrace{X_t^{i+1,j}}_{2'} + X_t^{i+1,j+2}), \quad (5.3)$$

com os termos idênticos $1 = 1'$ e $2 = 2'$ indicados.

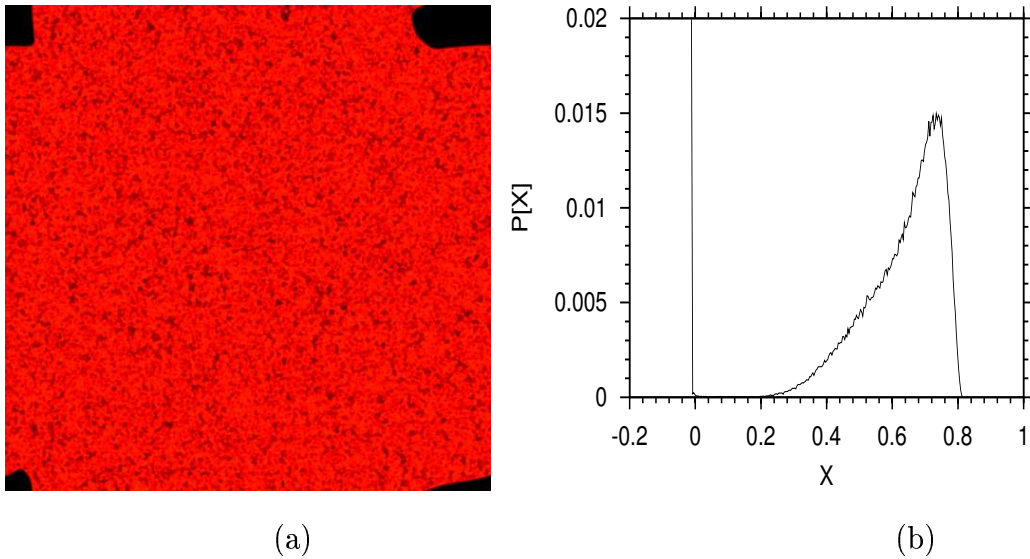


Figura 5.6: Uma solução diferente para $\varepsilon = 0.54$, a rede apresenta uma região sincronizada e outra caótica, para $t = 10000$ (a). O espectro correspondente para X_t (b). Devido à condição periódica de contorno, a única região sincronizada apresenta-se fragmentada nos vértices da rede, mas são partes de uma mesma região contígua.

A comparação das duas primeiras expressões acima mostra que, mapas adjacentes na direção horizontal (e, por simetria, também na direção vertical) possuem apenas um primeiro vizinho em comum. Mas se considerarmos dois mapas diagonalmente adjacentes, a primeira e última expressão acima mostram que o número de primeiros vizinhos comuns é igual a dois, o dobro do primeiro caso. Essa anisotropia causada pela restrição do acoplamento com primeiros vizinhos, tem duas implicações importantes e imediatas:

1. Como no limite $\varepsilon \rightarrow 1$ apenas os últimos termos das expressões acima contribuem para o acoplamento, os mapas vizinhos nas direções horizontal e vertical ficarão “descorrelacionados”, no sentido de que um não contribui no cálculo do novo valor do outro, dado pelas expressões acima, e *vice-versa*. Da função de correlação, definida por

$$C_t^{m,n} = \langle \tilde{X}_t^{i,j} \tilde{X}_t^{i+m,j+n} \rangle, \quad (5.4)$$

pode-se mostrar que, para a rede não sincronizada, ao longo das direções horizontal e diagonal teremos a desigualdade $C_t^{0,1} < C_t^{1,1}$.

Pela inspeção direta dos resultados gráficos obtidos nesse limite, vemos que coexistem efetivamente duas sub-redes independentes: uma para mapas com $(i + j)$ pares

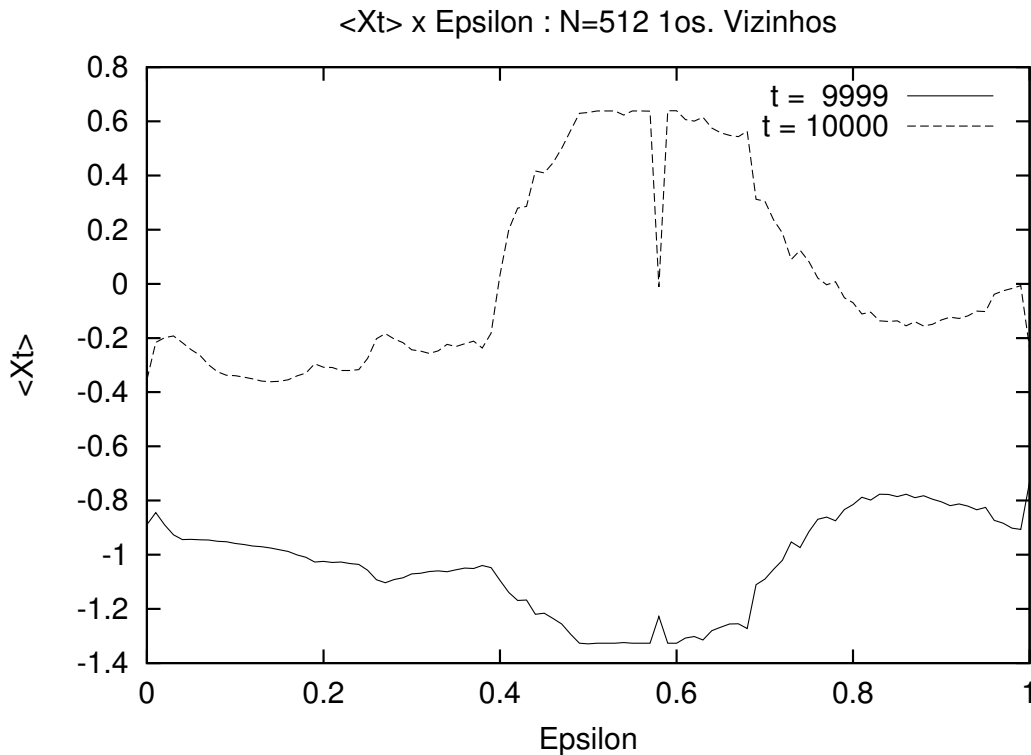
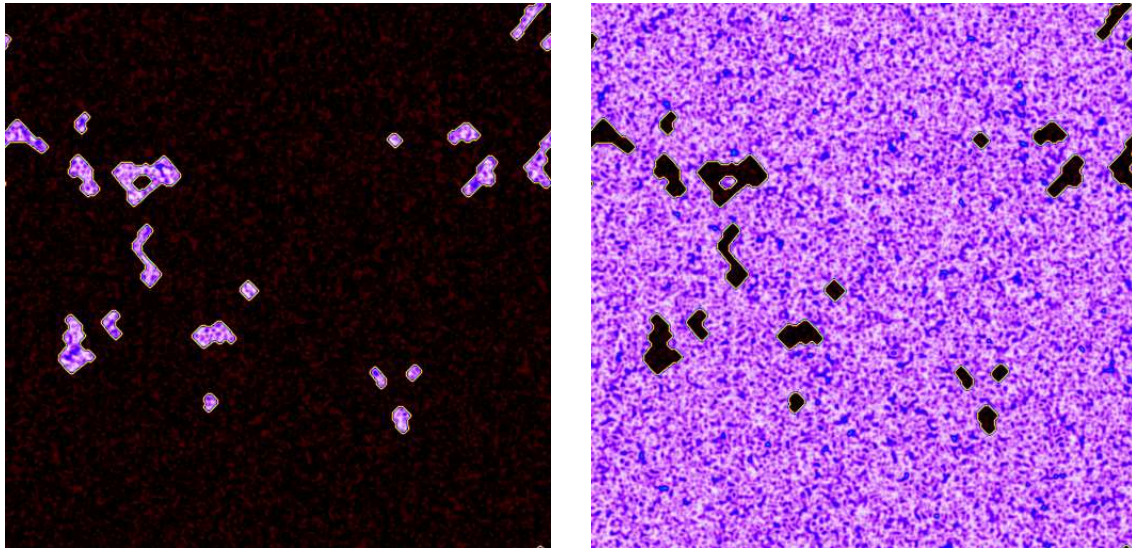


Figura 5.7: Gráfico do valor médio $\langle X(t) \rangle \times \epsilon$ para a rede quadrada, primeiros vizinhos e inicialização *RANDOM*.

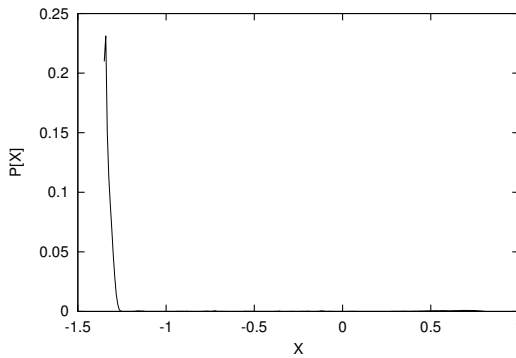
e outra para os ímpares, como num tabuleiro de xadrez. Esse fenômeno ocorre justamente pela desigualdade encontrada acima. Para o acoplamento feito até os segundos vizinhos o surgimento desse “efeito colateral” não será mais observado. A Figura 5.10 mostra o estado final original do sistema para $\epsilon = 1$, e as sub-redes separadas, para melhor observação das sub-redes independentes. Sobre cada uma dessas sub-redes a dinâmica observada é semelhante, embora independente, pois cada mapa possui efetivamente 4 vizinhos diagonais.

2. Mesmo para valores intermediários de ϵ , a desigualdade acima ainda se verifica para redes não sincronizadas, e isso privilegia a formação de paredes nas direções diagonais (a 45 graus). Ao longo de uma parede, a rede apresenta sincronização, ou seja, se os mapas sob a parede forem observados em detalhes, teremos uma linha de sincronização dentro do sistema, e isto ocorre preferencialmente sobre as diagonais, para o acoplamento de primeiros vizinhos. Como exemplo, apresentamos o estado

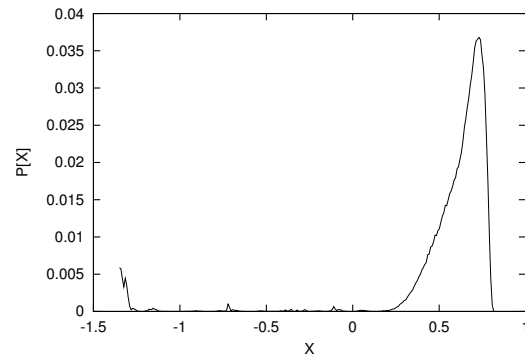


(a)

(b)



(a)



(b)

Figura 5.8: *Paredes fixas são encontradas separando diferentes regiões caóticas na rede, para $\varepsilon = 0.65$. A figura mostra a rede nos instantes $t=9999$ (a) e $t=10000$ (b), e os espectros correspondentes de X_t em (c) e (d).*

da rede para o valor *democrático* da constante de acoplamento:

$$\varepsilon_d = S/(S + 1) , \quad (5.5)$$

que é o valor para o qual a equação (4.2) se reduz à média aritmética dos valores de X_t envolvidos, com isso todos os mapas tem o mesmo peso relativo. Para o caso em questão temos $S = 4$ e portanto $\varepsilon_d = 4/5 = 0.80$. No detalhe mostrado na Figura 5.11b pode-se ver a estrutura detalhada de uma parede fixa da rede.

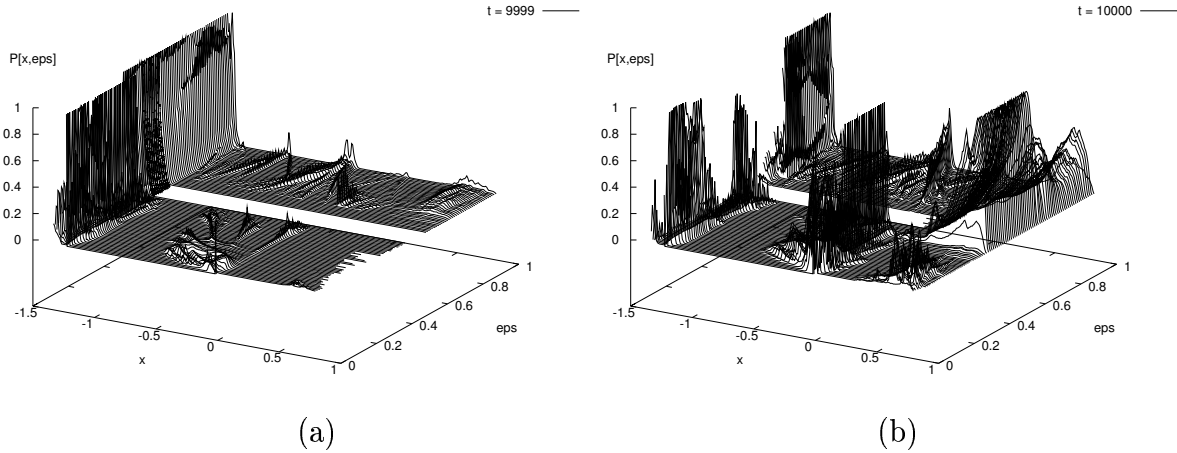


Figura 5.9: A superfície espectral para todos os valores de $\varepsilon \in [0, 1]$ usados, avaliados nos instantes $t=9999$ (a) e $t=10000$ (b).

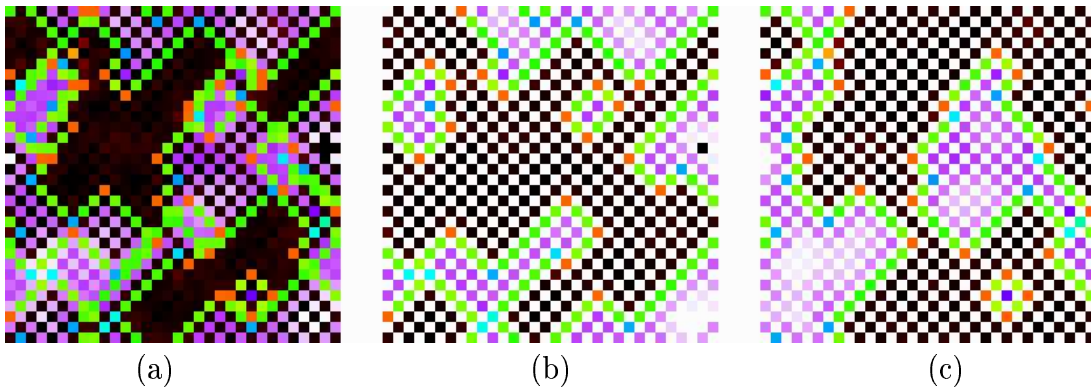


Figura 5.10: O efeito de desdobramento da rede completa para $\varepsilon = 1$ (a), em duas sub-redes independentes (b) e (c).

5.1.2 Rede Quadrada, Segundos Vizinhos

Considerando-se agora, para o acoplamento local, a inclusão dos segundos vizinhos de cada mapa da rede ($S = 8$), os resultados obtidos para a faixa inicial do parâmetro de acoplamento $\varepsilon' \leq \varepsilon < 0.26$ mostram que a rede apresenta o mesmo estado bloqueado do caso anterior.

Para valores no intervalo $0.26 \leq \varepsilon \leq 0.47$ a rede apresenta regiões bem definidas de atividade coletiva e regiões sincronizadas, separadas por paredes fixas, o que também se verificou para os valores particulares $\varepsilon = \{0.67, 0.81, 0.89\}$.

Como no caso de primeiros vizinhos, para o intervalo $0.48 \leq \varepsilon \leq 0.63$ o estado final da rede oscila entre duas fases caóticas diferentes, apresentando coletivamente um

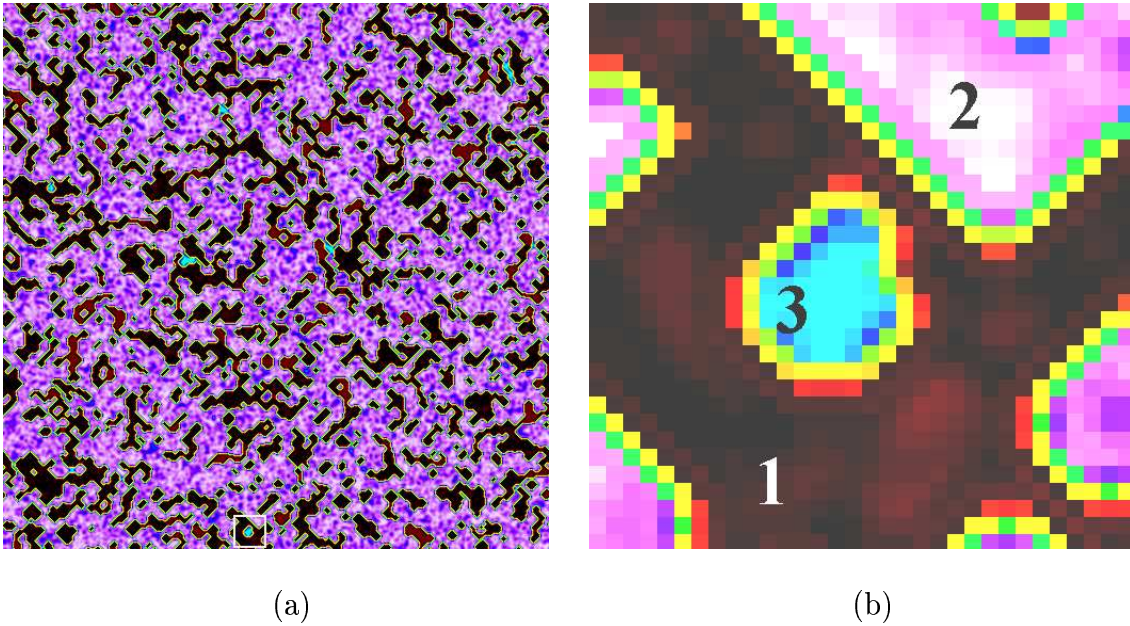


Figura 5.11: O estado final da rede para o acoplamento democrático $\varepsilon_d = 0.80$ (a), e um detalhe (b) centrado em uma pequena região sincronizada (3), cercada por regiões de caos (1-2), evidenciando as paredes fixas diagonais. A região ampliada tem tamanho igual a 32×32 mapas, e está marcada em (a).

ciclo-2 entre duas distribuições de valores de X_t bem definidas. Neste caso, se verifica que as paredes formadas no sistema nas primeiras iterações da rede são instáveis, e logo desaparecem devido à competição e domínio de uma das fases sobre a outra.

Aumentando-se um pouco mais o valor de ε , observamos que o sistema inicial de paredes evolui para uma configuração estável de paredes fixas, sem exceção. Isto ocorre para todo o intervalo

$0.65 \leq \varepsilon \leq 1.00$, sem que o efeito de desdobramento da rede em sub-redes independentes seja mais observado, mesmo para $\varepsilon = 1.00$. O que se observa pela inspeção das imagens da rede é que nessa região as duas fases de comportamento coletivo coexistem separadas por paredes fixas definidas por contornos suaves, e que não seguem mais a direção preferencial diagonal, como no caso anterior. A Figura 5.12 mostra os estados finais da rede para os extremos desse intervalo, no instante $t = 10000$.

Estados finais com sincronização completa foram obtidos apenas para os valores $\varepsilon = \{0.56, 0.64, 0.83\}$ do acoplamento.

A rugosidade obtida para a rede, neste caso, está mostrada na Figura 5.13a, e a evolução dos valores médios finais $\langle X(t) \rangle$ é mostrada na Figura 5.13b.

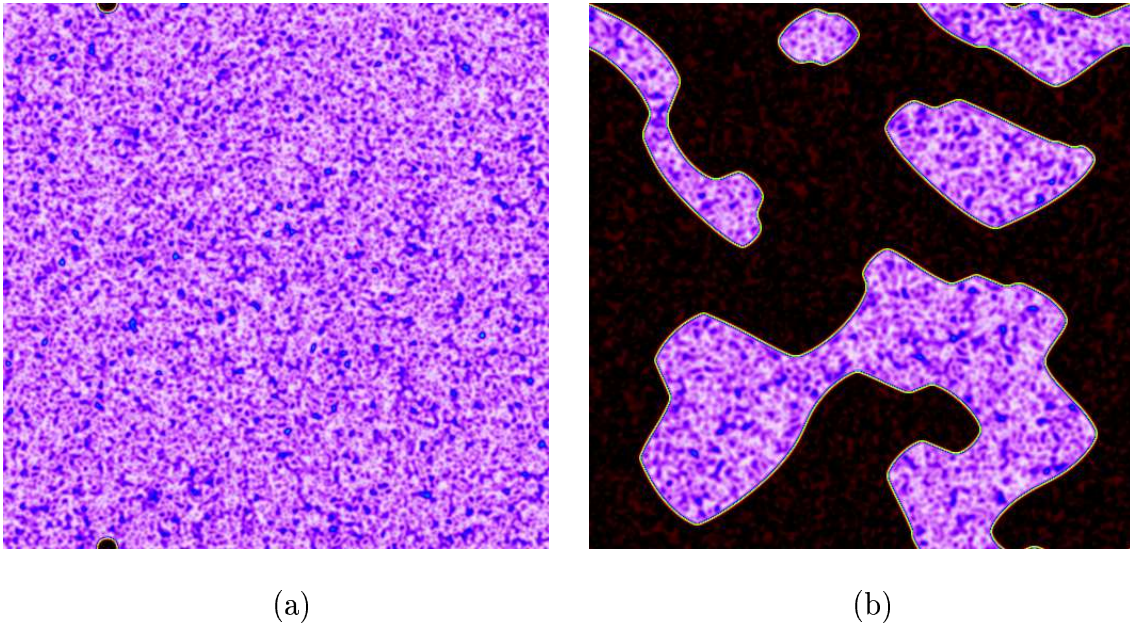


Figura 5.12: O estado da rede no instante $t = 10000$, para $\varepsilon = 0.65$ (a), e $\varepsilon = 1.00$ (b), mostrando o sistema de paredes fixas.

A evolução dos espectros dos valores de X_t pode ser visualizada pelas superfícies mostradas na Figura 5.14, e a persistência das duas fases para valores do acoplamento $\varepsilon \geq 0.65$ pode ser identificada pela superposição dos espectros que ocorre nessa região, para as iterações pares e ímpares. Já na região anterior a essa ($0.48 \leq \varepsilon \leq 0.63$), onde não existe essa mistura de fases caóticas, os espectros não se superpõem, dando origem à lacuna (faixa branca) observada na superfície.

5.1.3 Rede Hexagonal, Primeiros Vizinhos

Para a rede hexagonal, devido ao enfraquecimento estrutural imposto pela geometria da rede, já que cada mapa possui agora apenas três primeiros vizinhos ($S = 3$), observamos que para qualquer valor $0.0027 < \varepsilon \leq 1.00$ a rede apresenta estados finais bloqueados.

Neste tipo de rede, dois sítios adjacentes não possuem nenhum mapa vizinho em comum, a não ser eles mesmos. Com isso, não há nenhuma direção preferencial para a formação de paredes sobre a rede.

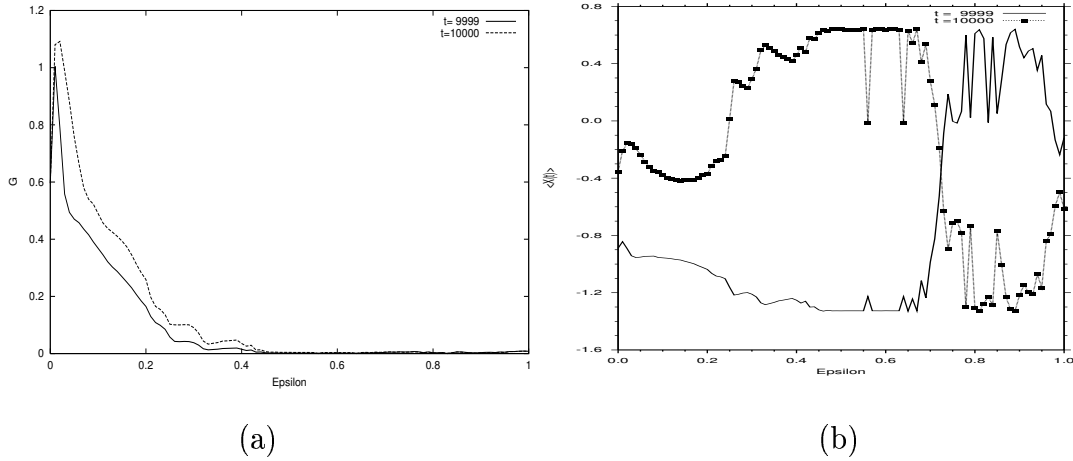


Figura 5.13: Gráfico da rugosidade G para a rede quadrada, segundos vizinhos e inicialização *RANDOM* (a). A evolução de $\langle X(t) \rangle$ para os diferentes valores de ε (b).

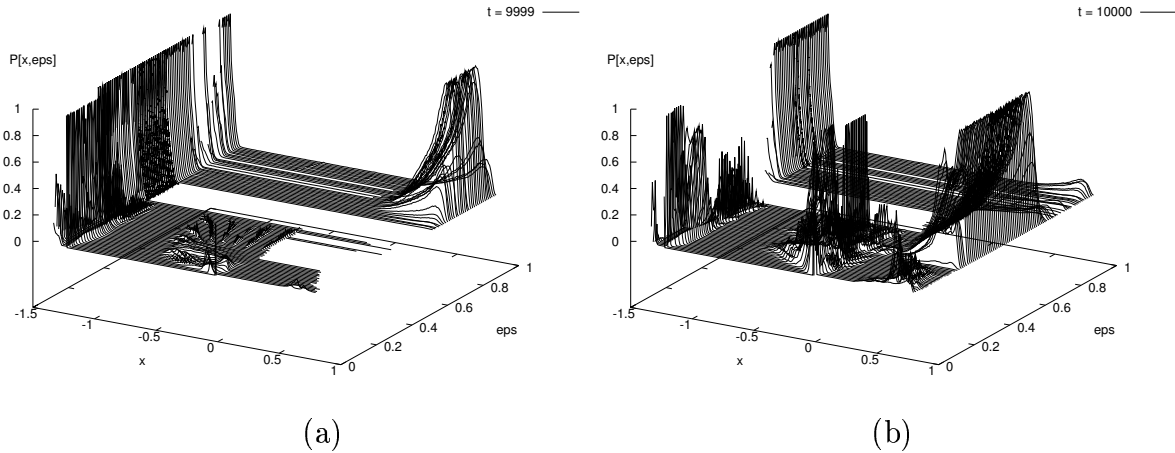


Figura 5.14: A superfície de espectros $\mathcal{P}[X(t)]$ para o acoplamento de segundos vizinhos, rede quadrada e inicialização *RANDOM*, para $t=9999$ (a) e $t=10000$ (b).

5.2 Inicialização RING50%

Para a inicialização RING50%, utilizamos uma distribuição uniforme de valores iniciais $X_{t=0}^{i,j}$ sorteados aleatoriamente em torno do ponto X_{00} . Este valor separa as trajetórias de forma que os pontos iniciados à esquerda dele estarão em uma fase do ciclo periódico após um transitório, e os iniciados à direita estarão na fase oposta. Lançamos então as condições iniciais no intervalo $[X_{00} - \xi, X_{00} + \xi]$, com $\xi = 0.05$. Com isso, esperamos que estatisticamente 50% dos mapas sejam inicializados dentro da bacia atratora para o ponto fixo X_3^* do mapa local, e os outros 50% dentro da bacia atratora para o outro ponto fixo X_5^* .

Para a rede desacoplada ($0 \leq \varepsilon \leq \varepsilon'$, com $\varepsilon' = 2.93 \times 10^{-3}$), a rede apresenta uma dinâmica assintótica do tipo ciclo-2, como para a inicialização RANDOM, só que agora, com metade dos mapas em cada ponto fixo do mapa local.

5.2.1 Rede Quadrada, Primeiros Vizinhos

Para valores do acoplamento no intervalo $\varepsilon' \leq \varepsilon < 0.45$, a rede evolui para estados bloqueados, similares aos encontrados para a inicialização RANDOM.

Aumentando-se o acoplamento para valores de ε , nos intervalos $[0.45, 0.50]$ ou $[0.69, 1.00]$, verificamos o surgimento de paredes fixas bem definidas, onde coexistem regiões de comportamento coletivo com regiões sincronizadas. Entre estes intervalos, para $0.51 \leq \varepsilon \leq 0.68$, observamos que sobrevivem apenas grandes regiões de CCNT (Comportamento Coletivo Não Trivial), separadas por paredes. Nesse caso, a rugosidade cai a valores bem pequenos, dando origem ao patamar em torno de $G \simeq 2 \times 10^{-3}$ para este intervalo.

Para este tipo de inicialização não foi observado nenhum caso de sincronização total da rede, e também nenhum estado final onde apenas uma fase caótica fosse encontrada a cada iteração, portanto sem paredes. Mas como para esta inicialização espera-se uma competição mais longa entre as fases, uma simulação mais longa ($t = 40000$) mostra que assintoticamente uma das fases predomina. Ver Tabela 5.3, colunas CAOS 2F e CAOS 1F, onde 1F significa “uma fase”, etc. A evolução da média $\langle X(t) \rangle$ e da rugosidade da rede são as mostradas na Figura 5.15.

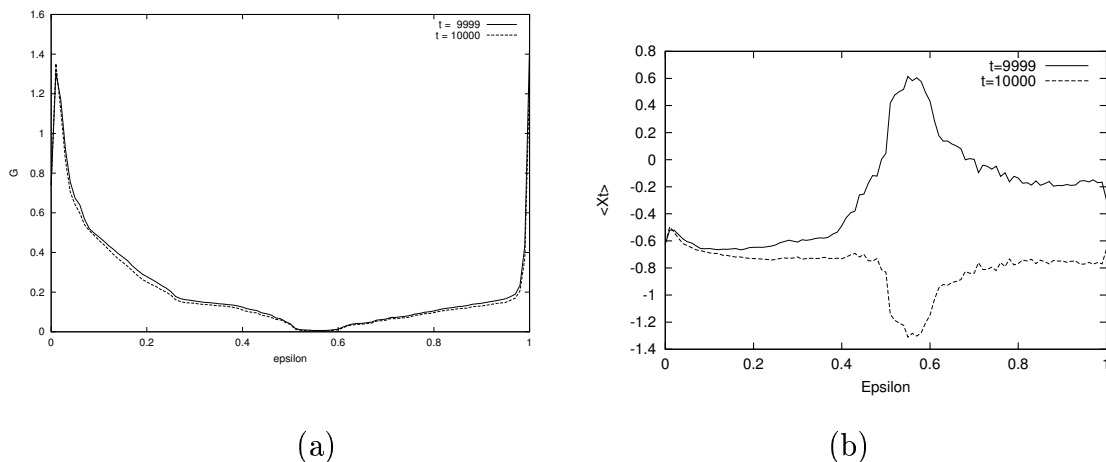


Figura 5.15: Gráfico da rugosidade $G \times \varepsilon$ para a rede quadrada, segundos vizinhos e inicialização RING50 (a). A evolução de $\langle X(t) \rangle$ para os diferentes valores de ε (b).

Novamente, a continuidade da superfície espectral para a rede, obtida para os diferentes valores de ε testados, sugere a existência de um atrator global para a rede, o que torna a dinâmica assintótica do sistema da rede *robusta*. De outra forma, não haveria como explicar essa característica observada nesta superfície de espectros, pois se a evolução do sistema fosse puramente estocástica, não poderíamos esperar, a priori, que a distribuição final dos valores $\mathcal{P}[X(t)]$ fosse invariante frente à mudança da configuração inicial da rede, como ocorre na rede. Ver Figura 5.16.

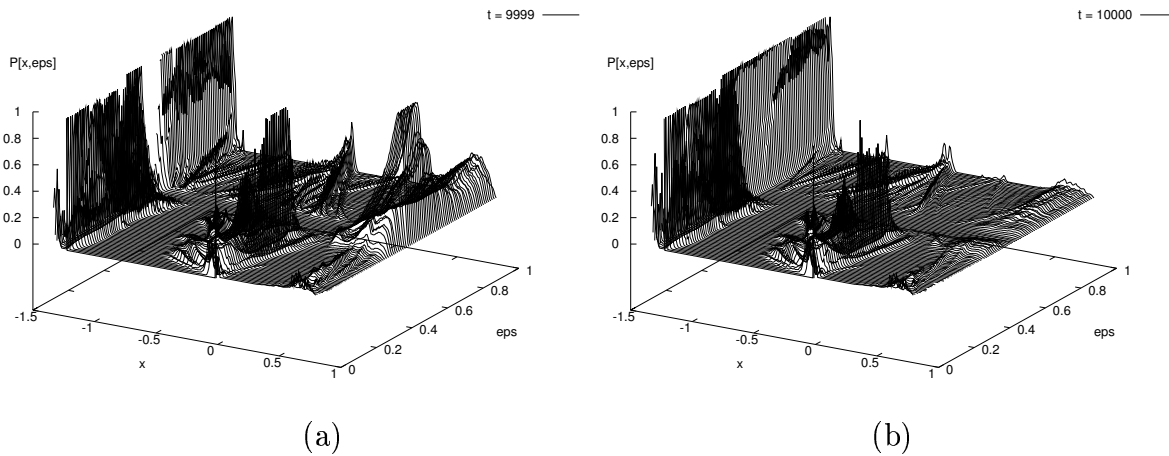


Figura 5.16: A evolução dos espectros $\mathcal{P}[X(t)]$ para a inicialização RING50%, nos instantes consecutivos $t = 9999$ (a) e $t = 10000$ (b).

Para valores grandes da constante de acoplamento, no intervalo $\varepsilon > 0.9$, a rede apresenta o mesmo comportamento de separação em sub-redes observado na inicialização RANDOM, primeiros vizinhos.

5.2.2 Rede Quadrada, Segundos Vizinhos

Para o acoplamento até segundos vizinhos, a inicialização em questão resulta em estados bloqueados para $\varepsilon' \leq \varepsilon < 0.43$. Para a estreita faixa $0.43 \leq \varepsilon \leq 0.47$ sub-regiões sincronizadas e de CCNT convivem na mesma iteração do sistema.

Para os valores particulares $\varepsilon = \{0.70, 0.83\}$, voltamos a Ver Figura 5.17.

Para o intervalo $0.48 \leq \varepsilon \leq 0.52$ a rede apresenta as duas fases caóticas simultaneamente, e portanto, paredes fixas. Nesta região, uma fase é predominante sobre a outra, conforme pode ser visto na Figura 5.18.

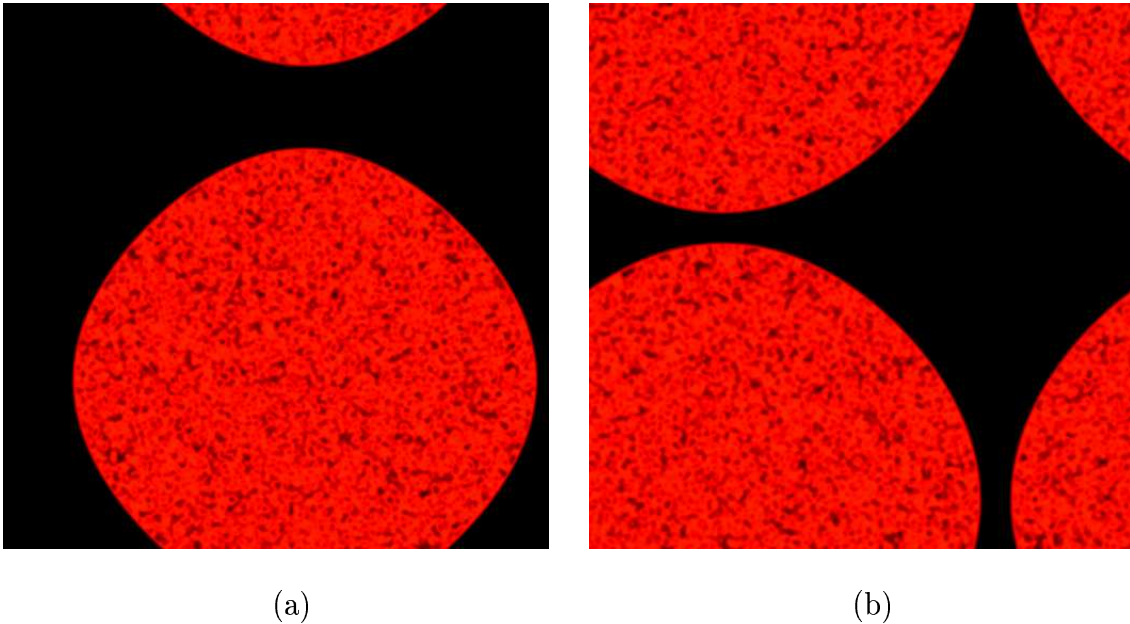


Figura 5.17: *Dois estados singulares da rede no instante $t = 9999$, para $\varepsilon = 0.70$ (a), e $\varepsilon = 0.83$ (b), mostrando uma fase caótica (circular) sobre a rede sincronizada.*

Finalmente, para todo o intervalo $0.53 \leq \varepsilon \leq 1.00$, a rede apresenta, em alguns intervalos menores apenas uma fase caótica e, para um grande número de pontos isolados, sincronização completa. Este panorama pode ser melhor visualizado pelo gráfico da média $\langle X(t) \rangle$ mostrado na Figura 5.19. Nos pontos de sincronização da rede, o valor médio apresenta o ciclo-2 sobre os pontos fixos do mapa local (descontinuidades). A figura também mostra a rugosidade da rede. A diferença na rugosidade de estados completamente sincronizados e estados de CCNT homogêneos não pode ser observada nos gráficos lineares apresentados pois para os primeiros a rugosidade é nula, e para os segundos, é da ordem de 10^{-5} .

5.2.3 Rede Hexagonal, Primeiros Vizinhos

Para a rede hexagonal, a inicialização RING50% apresenta os mesmos resultados obtidos para a inicialização RANDOM. *Ver subsecção 5.1.3.*

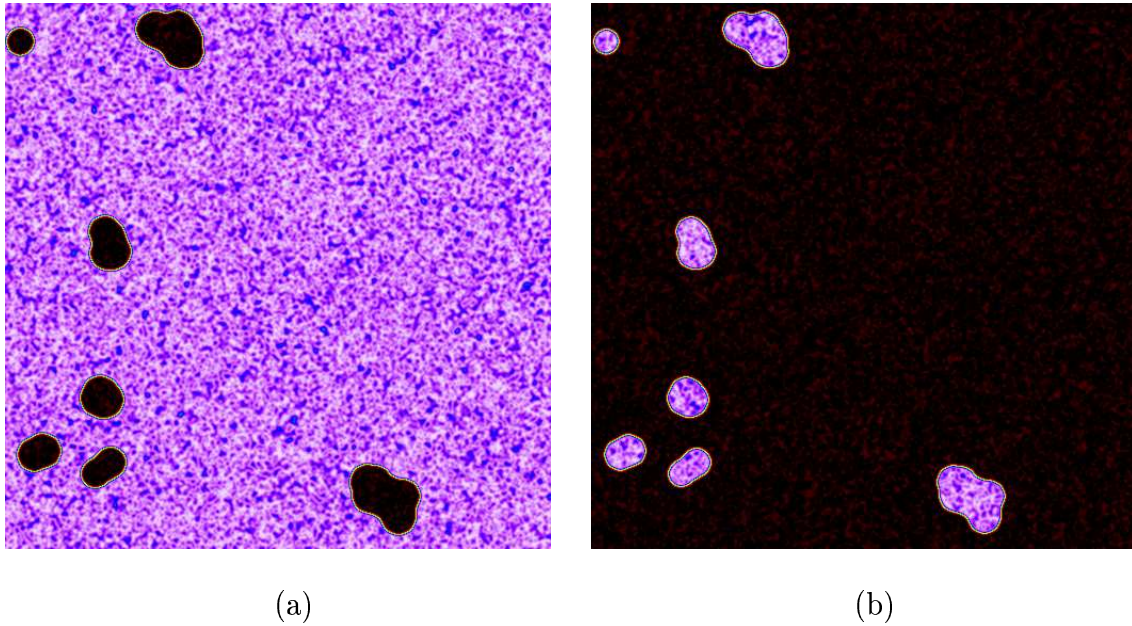


Figura 5.18: Um estado característico da rede para $\varepsilon = 0.51$ nos instantes $t = 9999$ (a), e $t = 10000$ (b), mostrando as fases caóticas isoladas por paredes fixas.

5.3 Inicialização PATCH3

Mais duas formas de inicialização foram testadas, especialmente preparadas para verificação da estabilidade de uma região sincronizada sobre a fase caótica, e conseqüentemente, das paredes fixas que aparecem em sua interface. Para isso inicializamos uma região circular de raio $N/4$, previamente sincronizada com uma dos valores X_3^* ou X_5^* , e fora dessa região central circular, usou-se a inicialização RANDOM, conforme a descrição dada na secção 5.1, com a amplitude $RAD = 1.0$. Com isso, garantimos que nenhuma parede nova será formada na rede, e a evolução da parede circular inicial pode ser acompanhada em detalhes. Essas inicializações foram chamadas PATCH3 e PATCH5, respectivamente, e inicializam cerca de 19.6% dos mapas da rede já sincronizados na região circular central.

Nesta secção descrevemos os principais resultados da inicialização PATCH3, e os resultados para a inicialização PATCH5 são os apresentados na secção 5.4.

5.3.1 Rede Quadrada, Primeiros Vizinhos

Para valores da constante de acoplamento no intervalo $0.00 \leq \varepsilon \leq 0.47$ a região circular central (sincronizada no valor X_3^*) permanece inalterada após 10000 iterações da rede, sendo que o resto da rede passa do estado bloqueado para uma fase de CCNT apenas, em

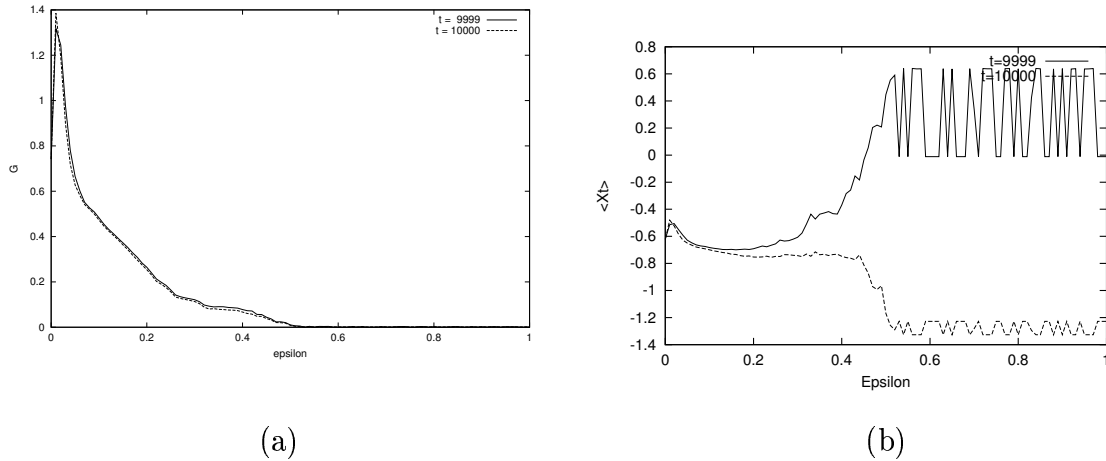


Figura 5.19: Gráfico da rugosidade $G \times \varepsilon$ para a rede quadrada, segundos vizinhos e inicialização RING50% (a). E o gráfico de $\langle X(t) \rangle \times \varepsilon$ (b).

torno de $\varepsilon = 0.15$.

Acima deste intervalo, para $0.48 \leq \varepsilon \leq 0.80$ a parede fixa circular dá lugar a formas que lembram polígonos regulares, com o predomínio de paredes diagonais: losango equilátero para $0.48 \leq \varepsilon \leq 0.61$ (instável para $t \rightarrow \infty$), hexágono para $0.62 \leq \varepsilon \leq 0.68$ e octógono para $0.69 \leq \varepsilon \leq 0.80$. A partir do valor democrático $\varepsilon_d = 0.80$, a parede circular inicial volta a ser obtida nos estados finais da rede. Ver Figura 5.20.

Em todos os casos estudados para essa inicialização, sempre encontramos para $t = 10000$, uma região central sincronizada, separada do resto da rede por paredes. Para certos intervalos de ε a área da região sincronizada diminuiu, indicando um estado assintótico sem a referida região, o que é compatível com os resultados anteriormente obtidos para estas condições de acoplamento. A forma geométrica assumida pela região central sincronizada, sugere a existência de intervalos bem definidos onde a competição imposta pelo acoplamento local privilegia a formação de paredes em diferentes ângulos. Próximo ao valor central $\varepsilon = 0.5$, no qual os dois termos do acoplamento (Ver Eq. (4.2)) assumem a mesma importância no cálculo da média local, vemos claramente que as paredes diagonais dominam as linhas de sincronização encontradas na rede, dando origem, pela simetria imposta nesse caso, à figura do losango obtido. Já para valores em torno de $\varepsilon = 0.70$, a forma octogonal bem definida indica a competição de paredes nas direções vertical e horizontal, contrabalançando a preferência anterior pelas direções diagonais.

$i \setminus j$	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
45	0.7346	0.7808	0.7173	0.5574	0.4349	0.3630	0.3999	0.4916	0.6458	0.6883	0.6312	0.5870	0.2909	-0.4390	-1.0993	-1.2592
46	0.7679	0.7852	0.7015	0.5603	0.4705	0.3791	0.3518	0.4055	0.5723	0.6993	0.6251	0.2918	-0.4429	-1.1007	-1.2591	-1.2287
47	0.7569	0.7627	0.7193	0.6931	0.6586	0.5223	0.3682	0.3663	0.5064	0.6034	0.3066	-0.4397	-1.1010	-1.2591	-1.2287	-1.2278
48	0.6790	0.6788	0.6956	0.7598	0.7545	0.6273	0.4600	0.4644	0.5295	0.3002	-0.4344	-1.1005	-1.2591	-1.2287	-1.2278	-1.2276
49	0.6748	0.6151	0.6506	0.7532	0.7405	0.6271	0.5354	0.5355	0.2901	-0.4328	-1.0998	-1.2591	-1.2287	-1.2278	-1.2276	-1.2276
50	0.7063	0.6201	0.6278	0.7040	0.6798	0.5974	0.5540	0.2944	-0.4322	-1.0994	-1.2591	-1.2287	-1.2278	-1.2276	-1.2276	-1.2276
51	0.6899	0.7073	0.7279	0.7547	0.7024	0.6006	0.3017	-0.4309	-1.0992	-1.2591	-1.2287	-1.2278	-1.2276	-1.2276	-1.2276	-1.2276
52	0.5519	0.6737	0.7651	0.7684	0.6595	0.3132	-0.4317	-1.0994	-1.2591	-1.2287	-1.2278	-1.2276	-1.2276	-1.2276	-1.2276	-1.2276
53	0.4263	0.6044	0.7227	0.6651	0.3156	-0.4351	-1.1001	-1.2591	-1.2287	-1.2278	-1.2276	-1.2276	-1.2276	-1.2276	-1.2276	-1.2276
54	0.4766	0.6204	0.6331	0.3087	-0.4390	-1.1010	-1.2591	-1.2287	-1.2278	-1.2276	-1.2276	-1.2276	-1.2276	-1.2276	-1.2276	-1.2276
55	0.6108	0.6053	0.2994	-0.4416	-1.1016	-1.2591	-1.2287	-1.2278	-1.2276	-1.2276	-1.2276	-1.2276	-1.2276	-1.2276	-1.2276	-1.2276
56	0.6305	0.2999	-0.4421	-1.1018	-1.2590	-1.2286	-1.2278	-1.2276	-1.2276	-1.2276	-1.2276	-1.2276	-1.2276	-1.2276	-1.2276	-1.2276
57	0.3116	-0.4393	-1.1013	-1.2591	-1.2286	-1.2278	-1.2276	-1.2276	-1.2276	-1.2276	-1.2276	-1.2276	-1.2276	-1.2276	-1.2276	-1.2276
58	-0.4324	-1.0998	-1.2591	-1.2287	-1.2278	-1.2276	-1.2276	-1.2276	-1.2276	-1.2276	-1.2276	-1.2276	-1.2276	-1.2276	-1.2276	-1.2276
59	-1.0965	-1.2592	-1.2287	-1.2278	-1.2276	-1.2276	-1.2276	-1.2276	-1.2276	-1.2276	-1.2276	-1.2276	-1.2276	-1.2276	-1.2276	-1.2276
60	-1.2593	-1.2288	-1.2278	-1.2276	-1.2276	-1.2276	-1.2276	-1.2276	-1.2276	-1.2276	-1.2276	-1.2276	-1.2276	-1.2276	-1.2276	-1.2276

Tabela 5.1: Os valores numéricos de $X_{t=10000}^{i,j}$ para $i, j = 45, 46, \dots, 60$ para a região marcada na Fig. 5.20c, na fronteira da parte central sincronizada (cinza). O brilho e o contraste da imagem da rede foram aumentados para facilitar a visualização dos números.

$t \setminus j$	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
9985	0.7659	0.7668	0.7501	0.7383	0.6771	0.6825	0.6468	0.3113	-0.4340	-1.1001	-1.2591	-1.2286	-1.2278	-1.2276	-1.2276	-1.2276
9986	-1.3036	-1.2992	-1.3059	-1.3125	-1.3333	-1.3381	-1.3265	-1.3105	-1.1389	-0.5924	-0.0612	0.0378	-0.0093	-0.0111	-0.0114	-0.0114
9987	0.5184	0.4741	0.4981	0.5436	0.6666	0.6905	0.6226	0.3130	-0.4366	-1.1012	-1.2591	-1.2287	-1.2278	-1.2276	-1.2276	-1.2276
9988	-1.3428	-1.3394	-1.3416	-1.3452	-1.3421	-1.3397	-1.3303	-1.3108	-1.1390	-0.5906	-0.0598	0.0379	-0.0093	-0.0111	-0.0114	-0.0114
9989	0.7434	0.7493	0.7593	0.7748	0.7631	0.7263	0.6483	0.3167	-0.4349	-1.1011	-1.2590	-1.2286	-1.2278	-1.2276	-1.2276	-1.2276
9990	-1.3155	-1.3043	-1.2970	-1.2890	-1.2977	-1.3189	-1.3249	-1.3109	-1.1393	-0.5908	-0.0602	0.0377	-0.0094	-0.0111	-0.0114	-0.0114
9991	0.5801	0.4868	0.4264	0.3866	0.4464	0.5537	0.5948	0.3121	-0.4334	-1.1005	-1.2590	-1.2286	-1.2278	-1.2276	-1.2276	-1.2276
9992	-1.3453	-1.3385	-1.3241	-1.3155	-1.3282	-1.3444	-1.3338	-1.3108	-1.1393	-0.5916	-0.0608	0.0378	-0.0094	-0.0111	-0.0114	-0.0114
9993	0.7548	0.7246	0.6362	0.5889	0.6591	0.7301	0.6727	0.3239	-0.4335	-1.1007	-1.2591	-1.2287	-1.2278	-1.2276	-1.2276	-1.2276
9994	-1.3120	-1.3210	-1.3422	-1.3474	-1.3394	-1.3263	-1.3193	-1.3105	-1.1400	-0.5913	-0.0604	0.0378	-0.0093	-0.0111	-0.0114	-0.0114
9995	0.5725	0.6147	0.7339	0.7733	0.7282	0.6326	0.5647	0.2970	-0.4322	-1.0998	-1.2591	-1.2286	-1.2278	-1.2276	-1.2276	-1.2276
9996	-1.3482	-1.3440	-1.3178	-1.3009	-1.3208	-1.3445	-1.3317	-1.3079	-1.1386	-0.5929	-0.0613	0.0378	-0.0093	-0.0111	-0.0114	-0.0114
9997	0.7838	0.7457	0.5941	0.5110	0.6184	0.7273	0.6542	0.3063	-0.4405	-1.1013	-1.2591	-1.2287	-1.2278	-1.2276	-1.2276	-1.2276
9998	-1.2936	-1.3116	-1.3433	-1.3457	-1.3428	-1.3278	-1.3250	-1.3096	-1.1389	-0.5905	-0.0599	0.0379	-0.0093	-0.0111	-0.0114	-0.0114
9999	0.4641	0.5599	0.7387	0.7788	0.7497	0.6550	0.6107	0.3098	-0.4367	-1.1010	-1.2590	-1.2286	-1.2278	-1.2276	-1.2276	-1.2276
10000	-1.3383	-1.3408	-1.3114	-1.2917	-1.3088	-1.3399	-1.3314	-1.3101	-1.1395	-0.5909	-0.0604	0.0378	-0.0093	-0.0111	-0.0114	-0.0114

Tabela 5.2: A evolução temporal dos valores numéricos para a linha 55 da rede mostrada na Tabela 5.1, isto é, $X_t^{i=55,j}$ para $j = 45, \dots, 60$, e $t = 9985, \dots, 10000$. As colunas 52 – 54 (em negrito) correspondem, respectivamente, às diagonais azul, amarela e vermelha daquela tabela.

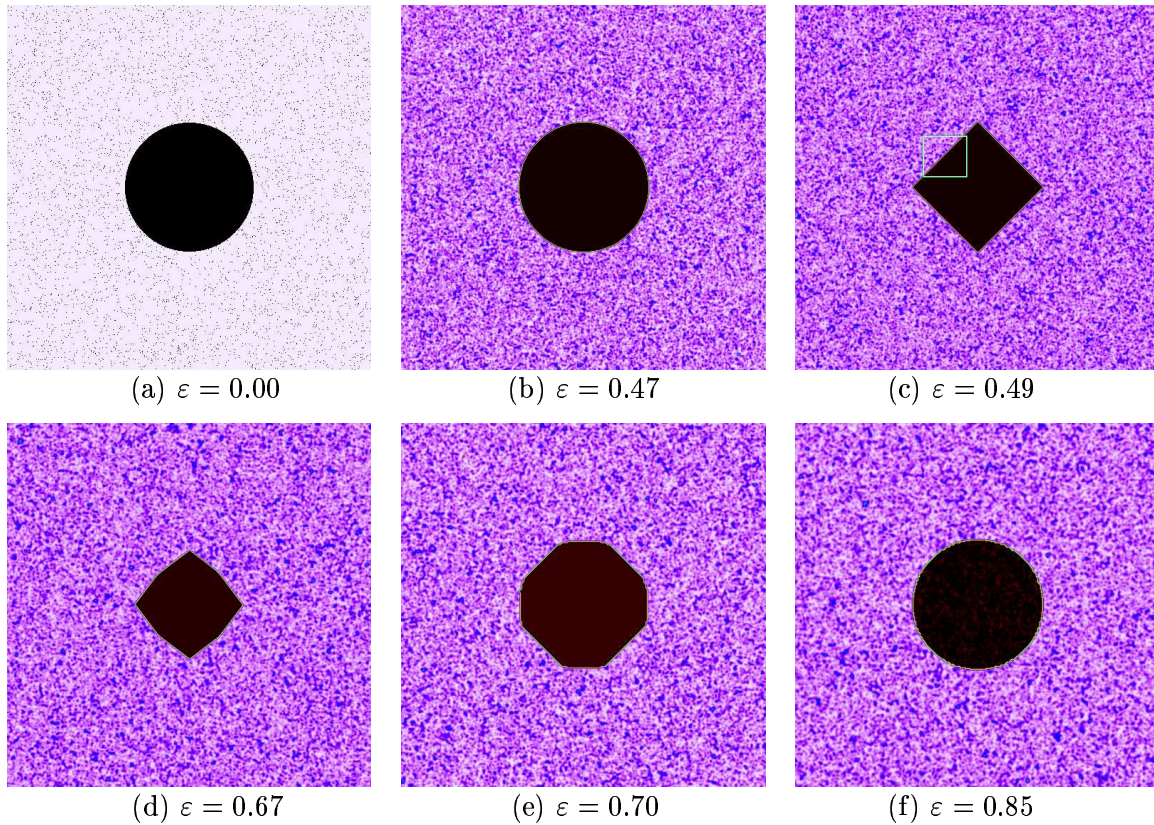


Figura 5.20: Alguns estados finais da rede em $t = 10000$ para a inicialização *PATCH3*, para $\varepsilon = 0.00$ (a), $\varepsilon = 0.47$ (b) e $\varepsilon = 0.49$ (c), em cima. E para $\varepsilon = 0.67$ (instável para $t \rightarrow \infty$) (d), $\varepsilon = 0.70$ (e) e $\varepsilon = 0.85$ (f), em baixo. A pequena área de 16×16 mapas marcada em (c) tem seus valores numéricos apresentados na Tabela 5.1.

A inspeção direta dos valores numéricos assumidos pelos mapas da rede, como no exemplo mostrado na Tabela 5.1 para $\varepsilon = 0.49$, revela a estrutura microscópica da parede observada na Figura 5.20, estrutura essa que é comum às paredes observadas em outras situações. Ao fundo da tabela está colocada a imagem ampliada (*zoom*) da pequena região da rede, com 16×16 mapas, usada na tabela. Identifica-se na tabela as linhas de quase-sincronização (nas colunas 52–54 que correspondem às diagonais azul, amarela, vermelha, etc. da Tab. 5.1) que fazem a transição da parte central sincronizada (nesse caso) até a parte caótica, onde as flutuações da variável local estão sempre presentes. Fixando-se a atenção em uma única linha dessa tabela, a linha 50 por exemplo, verificamos que para outras iterações da rede, cada mapa da parede apresenta uma dinâmica quase-periódica de período-2, e ao longo parede, variações muito pequenas em X_t são observadas, devido ao acoplamento com a parte caótica. Pode-se ver também da Tabela 5.1, que a “espessura” da parede é de pelo menos 5 sítios, apesar de ser difícil a definição precisa de uma tal

medida. Os valores numéricos obtidos para a linha 50 pode ser visto na Tabela 5.2, para as 16 iterações $t = 9985, \dots, 10000$ da rede, e as linhas diagonais centrais da parede (azul, amarela e vermelha) estão destacadas em negrito.

Observação: os valores numéricos mostrados nas Tabelas 5.1 e 5.2 foram obtidos em duas diferentes execuções de um mesmo programa de simulação, que utilizaram duas seqüências diferentes de números pseudo-aleatórios para a inicialização da rede, donde a não coincidência exata dos valores para a iteração final $t = 10000$. Isto foi feito de propósito, para evidenciar o fato de que a dinâmica da rede é robusta.

A Figura 5.21 mostra o comportamento da rugosidade e da média obtida para a inicialização PATCH3, primeiros vizinhos, sobre a rede quadrada.

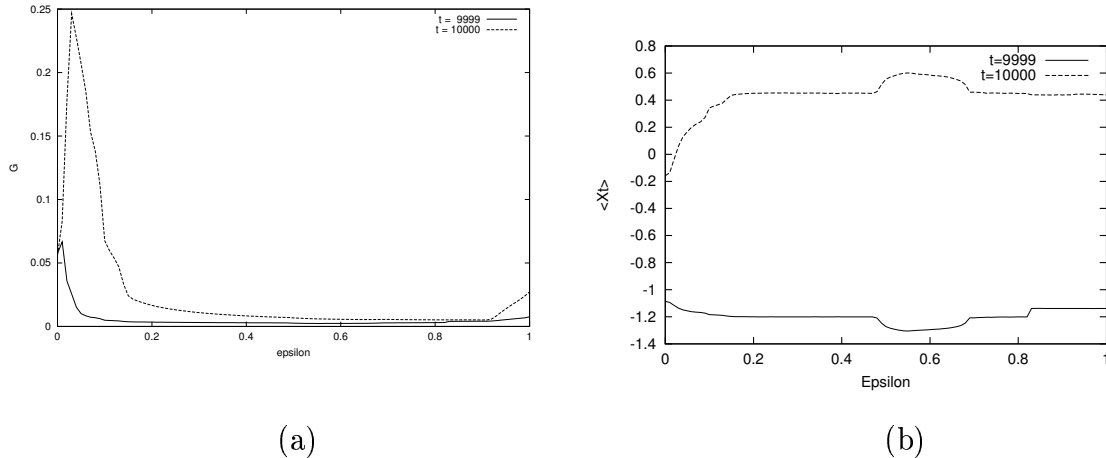


Figura 5.21: Gráfico da rugosidade $G \times \epsilon$ para a rede quadrada, primeiros vizinhos e inicialização PATCH3 (a). E o gráfico de $\langle X(t) \rangle \times \epsilon$ (b).

O efeito da região central sincronizada também pode ser observado na superfície dos espectros da rede, mostrado na Figura 5.22.

5.3.2 Rede Quadrada, Segundos Vizinhos

Considerando o acoplamento de segundos vizinhos, observamos que a parede circular inicial se mantém inalterada para $0.00 \leq \epsilon \leq 0.48$, mantendo-se estável após 10000 iterações da rede.

Para o intervalo seguinte, $0.49 \leq \epsilon < 0.57$, a parede circular evolui para uma forma final octogonal, e com aresta cada vez menor, até desaparecer completamente, para $\epsilon =$

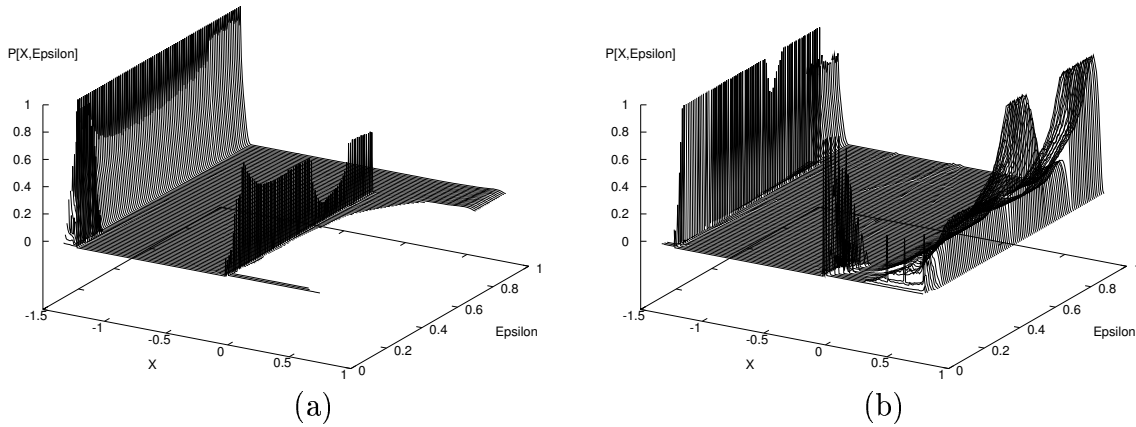


Figura 5.22: A evolução dos espectros $\mathcal{P}[X(t)]$ para a inicialização *PATCH3*, para $t = 9999$ (a), e $t = 10000$ (b).

0.58. A partir desse valor até $\varepsilon = 1.00$, a rede apresenta em geral uma fase de CCNT, voltando a sincronização completa nos pontos isolados $\varepsilon = 0.63$ e $\varepsilon = 0.67$.

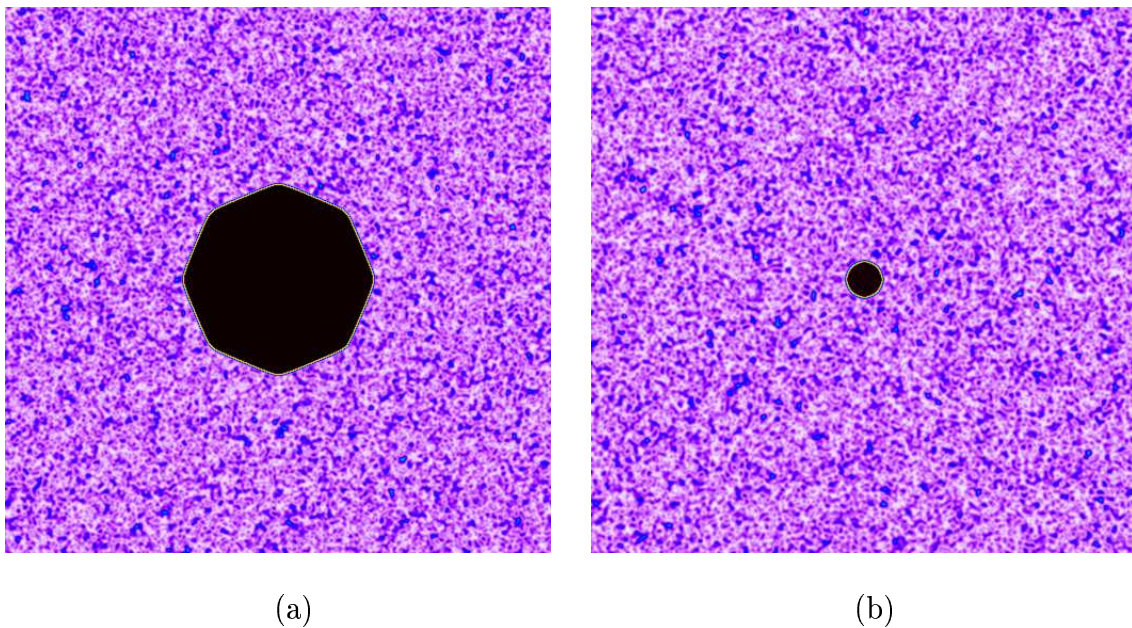


Figura 5.23: A rede quadrada com inicialização *PATCH3*, segundos vizinhos, para $\varepsilon = 0.49$ (a), e $\varepsilon = 0.56$ (b).

A comparação dos estados finais mostrados em 5.20c e 5.23a sugere que a forte anisotropia encontrada no primeiro caso ($S = 3$) é atenuada para esse acoplamento ($S = 8$), havendo maior igualdade na competição entre paredes, em qualquer direção da rede.

5.3.3 Rede Hexagonal, Primeiros Vizinhos

A rede hexagonal apresenta os mesmos resultados para a inicialização PATCH3 observados para as outras inicializações já vistas. A parede circular inicial imposta por esta inicialização, permanece sempre inalterada até a iteração $t = 10000$, mostrando que é possível a existência de paredes estáveis em qualquer direção para esse acoplamento. Microscopicamente, há um pequeno rearranjo dos valores dos mapas em torno da parede inicial devido à geometria singular da rede hexagonal, o que desaparece depois de algumas iterações da rede.

A Figura 5.24 mostra a rugosidade e média obtida para a inicialização PATCH3, primeiros vizinhos, agora sobre a rede hexagonal.

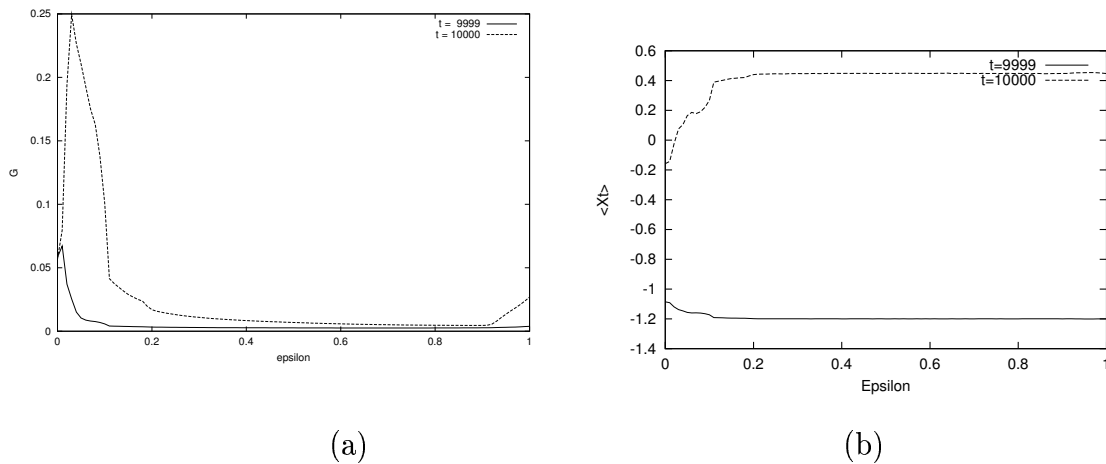


Figura 5.24: A rugosidade da rede quadrada, primeiros vizinhos e inicialização PATCH3 (a), e o valor médio de $X_t(\epsilon)$ (b).

5.4 Inicialização PATCH5

Neste caso, tanto a região central como o resto da rede é inicializada com valores pertencentes à mesma bacia atratora local para o ponto fixo X_5^* , não havendo inicialmente portanto competição entre estas bacias locais. Desta forma se espera testar a estabilidade da parte sincronizada associada ao ciclo-2 do atrator periódico local sobre a possível fase de CCNT associada ao transiente caótico do mapa.

5.4.1 Rede Quadrada, Primeiros Vizinhos

Na ampla faixa de acoplamento $0.12 \leq \varepsilon \leq 1.00$ a rede apresenta apenas uma fase CCNT, ou sincronização completa. Dentro desse intervalo, apenas para $\varepsilon = 0.65$ a rede apresentou um estado final semelhante ao casos mostrados na Figura 5.6.

Para $\varepsilon = 0.11$ a rede apresenta o estado final mostrado na Figura 5.25. Pelo espectro da variável X_t para esse acoplamento, pode-se identificar alguns picos em torno dos pontos fixos estáveis do mapa local, e também outros, um deles associado a ponto fixo instável X_7^* ($t = 9999$) associado ao pico em torno do valor $X = 0.7$ ($t = 10000$).

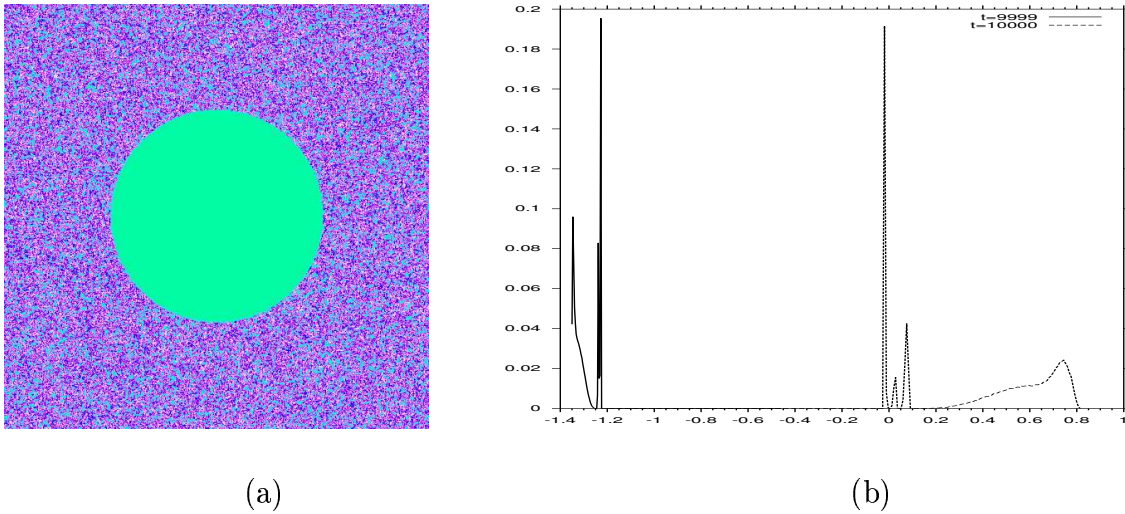


Figura 5.25: Para $\varepsilon = 0.11$ a rede apresenta a região central sincronizada e outra bloqueada, para $t = 10000$ (a). O espectro correspondente (b).

5.4.2 Rede Quadrada, Segundos Vizinhos

Para o acoplamento local de segundos vizinhos a rede apresenta praticamente os mesmos resultados para o caso de primeiros vizinhos. A figura 5.26 apresenta a rugosidade e a média sobre a rede, para esse tipo de acoplamento.

5.4.3 Rede Hexagonal, Primeiros Vizinhos

Também para a rede hexagonal, a inicialização PATCH5 apresentou os mesmos resultados básicos obtidos para a rede quadrada, primeiros vizinhos. Ver subsecção 5.4.1.

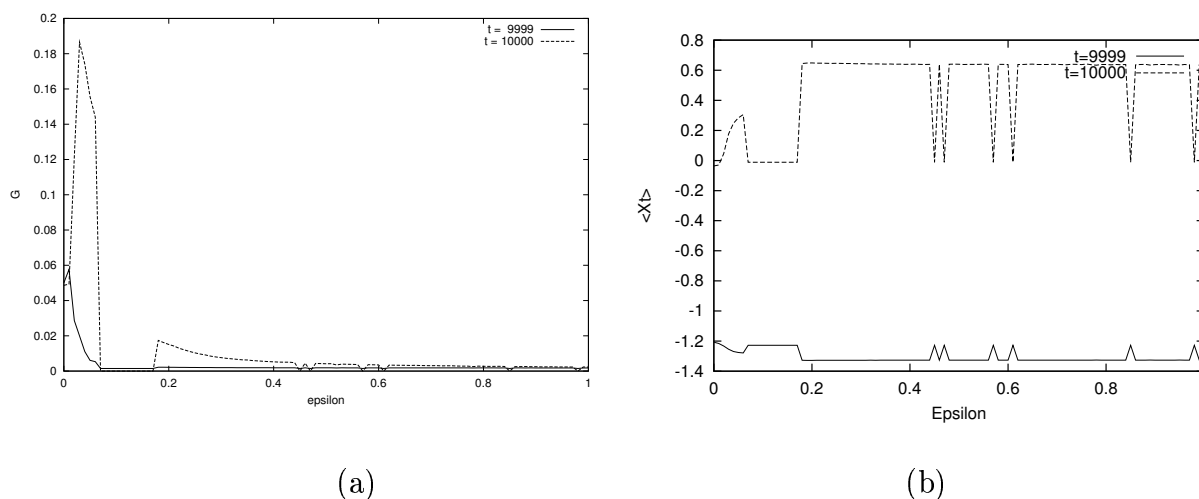


Figura 5.26: Gráfico da rugosidade $G \times \epsilon$ para a rede quadrada, primeiros vizinhos e inicialização PATCH3 (a). E o gráfico de $\langle X(t) \rangle \times \epsilon$ (b).

Para um valor particular do acoplamento $\epsilon = 0.45$, a rede hexagonal apresentou um estado do tipo observado na Figura 5.6, só que com as regiões de CCNT assumindo formas hexagonais, conforme mostra a figura 5.27.

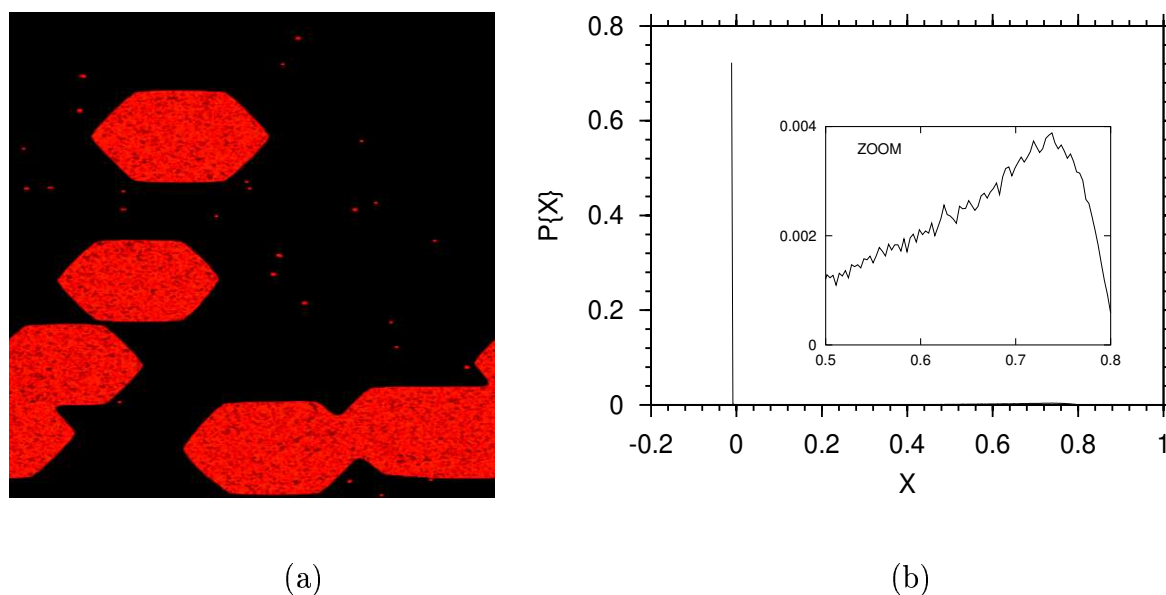


Figura 5.27: Para $\epsilon = 0.45$, paredes hexagonais separam regiões de CCNT sobre a rede sincronizada (a); o espectro de X_t correspondente (b).

Finalmente, para a superfície espectral da rede, obtemos os gráfico mostrados na Figura 5.28.

Pode-se observar na Figura 5.28 a existência de três regiões bem distintas, que corre-

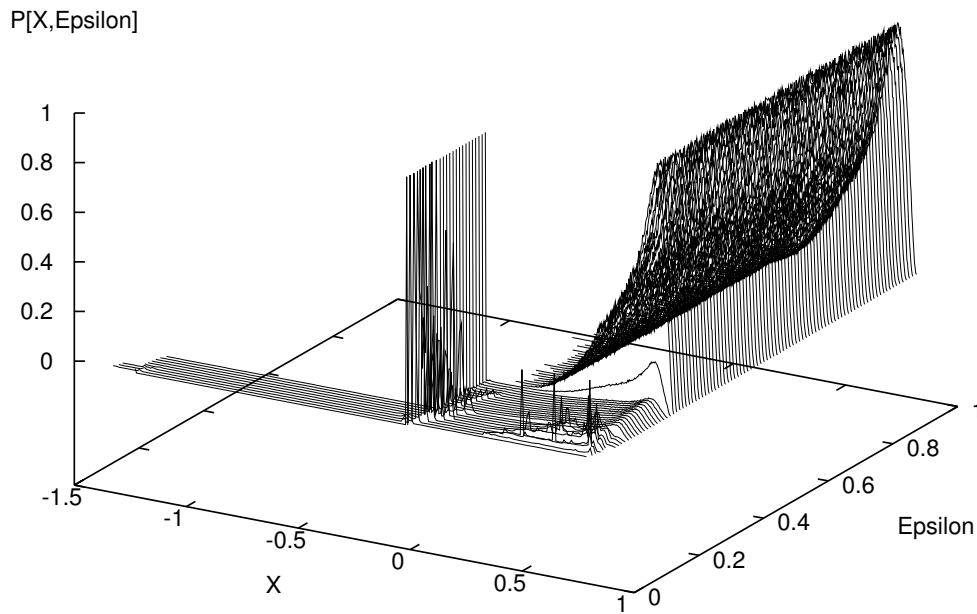


Figura 5.28: A evolução dos espectros $\mathcal{P}[X(t)]$ para a inicialização *PATCH3*, para $t = 10000$.

spondem ao comportamento observado na rede. Na primeira faixa estreita ($0.00 \leq \varepsilon \leq 0.09$) corresponde aos estados finais bloqueados, a segunda ($0.10 \leq \varepsilon \leq 0.24$) corresponde aos estados finais sincronizados. Por fim, a faixa de trás ($0.24 < \varepsilon \leq 1.00$) é formada pelos espectros dos estados finais que apresentam comportamento coletivo não trivial – caos.

5.5 Quadro Geral de Resultados

Para finalizar a análise feita caso a caso nas secções anteriores, resumimos em uma só tabela as principais informações obtidas para os estados finais das redes estudadas. O Apêndice 6.1 serve como referência básica para os estados finais obtidos graficamente, apresentando todas as imagens finais obtidas para $0.00 \leq \varepsilon \leq 0.99$, obtidos a cada $\Delta\varepsilon = 0.01$, para $t = 9999$ e $t = 10000$.

Rede	Acoplamento	Rede de Mapas Quárticos Acoplados – Dinâmica Assintótica para $N = 512$ e $t = 10000$					Inicialização
		Bloqueado	Caos+Sincro*	Caos 1F	Caos 2F*	Sincro 1F	
Quadrada	1 ^{os.} Viz.	$0 \leq \varepsilon < 0.40$	$0.40 \leq \varepsilon \leq 0.50$ e $0.69 \leq \varepsilon \leq 1.00!$ e $\varepsilon = 0.54!!$	$0.50 < \varepsilon \leq 0.60$	$0.61 \leq \varepsilon \leq 0.68$	$e = 0.58$	RANDOM
	S=4	$0 \leq \varepsilon < 0.45$	$0.45 \leq \varepsilon \leq 0.50$ e $0.69 \leq \varepsilon \leq 1.00!$	—	$0.51 \leq \varepsilon \leq 0.68$	—	RING50%
	$\varepsilon_d = 4/5$	$0 \leq \varepsilon < 0.15$	$0.15 \leq \varepsilon \leq 0.47$ (circ.) e $0.48 \leq \varepsilon \leq 0.80$ (hex.)	—	—	—	PATCH3
		$0 \leq \varepsilon < 0.11$	$\varepsilon = 0.65$	$0.11 < \varepsilon \leq 1.00?$	—	$0.11 < \varepsilon \leq 1.00?$	PATCH5
	2 ^{os.} Viz.	$0 \leq \varepsilon < 0.26$	$0.26 \leq \varepsilon \leq 0.47$ e $\varepsilon \in \{0.67, 0.81, 0.89\}$	$0.48 \leq \varepsilon \leq 0.63$	$0.65 \leq \varepsilon \leq 1.00$	$\varepsilon \in \{0.56, 0.64, 0.83\}$	RANDOM
	S=8	$0 \leq \varepsilon < 0.43$	$0.43 \leq \varepsilon \leq 0.47$ e $\varepsilon \in \{0.70, 0.83\}!!$	$0.53 \leq \varepsilon \leq 1.00?$	$0.48 \leq \varepsilon \leq 0.52$	$0.53 \leq \varepsilon \leq 1.00?$	RING50%
	$\varepsilon_d = 8/9$	$0 \leq \varepsilon < 0.12$	$0.12 \leq \varepsilon \leq 0.48$ (circ.) e $0.49 \leq \varepsilon < 0.57$ (oct.)	$0.57 \leq \varepsilon \leq 1.00$	—	$\varepsilon \in \{0.63, 0.67\}$	PATCH3
		$0 \leq \varepsilon < 0.07$	—	$0.18 \leq \varepsilon \leq 1.00$	—	$0.07 \leq \varepsilon \leq 0.17$ $0.18 \leq \varepsilon \leq 1.00?$	PATCH5
Hexagonal	1 ^{os.} Viz.	$0 \leq \varepsilon \leq 1.00$	—	—	—	—	RANDOM
	S=3	$0 \leq \varepsilon \leq 1.00$	—	—	—	—	RING50%
	$\varepsilon_d = 3/4$	$0 \leq \varepsilon < 0.11$	$0.11 \leq \varepsilon \leq 1.00$	—	—	—	PATCH3
		$0 \leq \varepsilon < 0.10$	$\varepsilon = 0.45!!$ (hex.)	$0.24 < \varepsilon \leq 1.00$	—	$0.10 < \varepsilon \leq 0.24$	PATCH5

Tabela 5.3: Classificação geral das diferentes dinâmicas assintóticas para as redes de mapas quárticos acoplados, com parâmetros ($a_0 = 0.35, b_0 = 1.35$), para as inicializações e geometrias testadas. Nos tipos marcados com ‘*’ a rede apresenta paredes rígidas bem definidas, com ‘!’ apresenta pequenas regiões sincronizadas, com ‘!!’ apresenta regiões de caos sobre fundo sincronizado e com ‘?’ apresenta alternância dos regimes marcados, para diversos valores de ε no intervalo.

Capítulo 6

Conclusões

Neste trabalho estudamos uma rede bidimensional de mapas quárticos acoplados localmente. Todos os mapas da rede foram definidos sobre o mesmo ponto ($a_0 = 0.35, b_0 = 1.35$) do espaço de parâmetros do mapa local – o mapa quártico. Este mapa possui dois atratores cujas bacias de atração, como no caso do ponto a_0, b_0 acima mencionado, por vezes se fundem. Foram testadas várias formas de inicialização dos mapas da rede e diferentes números de coordenação (ordens de acoplamento), isto sobre uma rede quadrada e também sobre uma rede hexagonal.

A análise da dinâmica do mapa local revelou a existência de fusão (superposição) entre as bacias de atração associadas a dois de seus pontos fixos, um periódico X_5^* (estável) e outro caótico X_7^* , que é instável. O ponto fixo caótico é responsável pelo surgimento do comportamento coletivo não-trivial (CCNT) observado na rede. Este comportamento foi identificado nas várias combinações de diferentes geometrias, ordens e intensidades de acoplamento. Uma evidência disso é o fato de que os valores médios sobre cada fase do ciclo de CCNT coincidem aproximadamente com os valores dos pontos fixos caóticos.

As simulações das redes feitas com 10.000 iterações, para cada valor da intensidades de acoplamento (ε), mostraram que existem intervalos de coexistência de estados sincronizados e de CCNT. Para alguns valores específicos foram feitas simulações mais longas, mesmo nestes casos manteve-se a coexistência o que torna extremamente provável sua existência para os estados assintóticos do sistema.

De maneira geral os estados assintóticos obtidos dependem fortemente das distribuições

iniciais e da topologia da rede utilizada.

A estabilidade das paredes que separam diferentes fases coexistentes nos estados assintóticos depende do valor assumido pela constante de acoplamento. Em particular, na vizinhança do valor $\varepsilon = 0.5$ as paredes são assintoticamente instáveis. A medida da rugosidade apresenta um patamar mínimo em torno desse ponto, independentemente do número de coordenação, o que significa também um número mínimo de paredes fixas estáveis.

Para $\varepsilon > 0.7$ existem regiões de medida não nula (extensas) para as quais a rede apresenta sincronização completa nos valores do ciclo-2 estável do mapa local $\{X_3^*, X_5^*\}$.

Para valores abaixo de $\varepsilon = 0.5$ a rede apresentou a formação inicial de um fundo sincronizado, sobre o qual se encontram várias “ilhas” de CCNT, sendo que essas últimas crescem sobre a parte sincronizada resultando em estados assintóticos com CCNT. Dito de outra forma, neste intervalo de acoplamento as paredes entre as regiões sincronizadas e de CCNT são instáveis.

A regularidade observada em todas as superfícies espectrais obtidas, ao longo da direção do eixo ε apresenta-se como um forte argumento a favor da hipótese da existência de um “atrator global” para os estados assintóticos da rede, além de caracterizar a dinâmica coletiva observada como um fenômeno robusto. Por outro lado, o fato de se encontrar estados completamente sincronizados para certos valores (altos) de acoplamento indica que este atrator não é único e que atratores para uma dinâmica do tipo CCNT e sincronizada podem conviver lado a lado no espaço de parâmetros ou ainda coexistir para o mesmo valor de acoplamento, como ocorre para valores de acoplamento baixos, i.e., $0.35 \leq \varepsilon \leq 0.5$.

6.1 Continuidade Deste Trabalho

Algumas das idéias que foram surgindo ao longo desse trabalho, principalmente na sua fase final, e que portanto não puderam ser devidamente desenvolvidas, pois estenderiam essa dissertação muito além do previsto inicialmente, mas que são de nosso interesse imediato, estão enumeradas abaixo:

1. A inspeção analítica da função usada para efetuar o acoplamento local dos mapas da rede revelou a existência de um conjunto de transformações lineares que permite a construção de um autômato celular equivalente à rede de mapas acoplados estudada. Essa transformação que preserva a dinâmica do sistema original, que é robusta, e as equações obtidas são invariantes na forma. A principal vantagem dessa transformação é que, para o autômato celular a variável dinâmica é discreta, e todas as operações realizadas dentro do *loop* principal do programa de simulação passam a ser sobre números inteiros, o que aumenta a velocidade de processamento por um fator próximo a 16.
2. Até o presente momento, o mecanismo de formação das paredes fixas ainda não é bem compreendido do ponto de vista teórico, e portanto carece de melhor investigação [CHATÉ *et al.*, 1997, KANEKO, 1989b, LEMAÎTRE *et al.*, 1998]. Modelos macroscópicos para a descrição da dinâmica coletiva observada nas redes são objetos de pesquisa atualmente [LEMAÎTRE *et al.*, 1999, LEMAÎTRE *et al.*, 1999b].
3. Como uma extensão natural deste trabalho, seria de interesse investigar o comportamento de uma rede tridimensional, nos moldes em que foi feito este estudo para o caso bidimensional. Neste caso, seria especialmente interessante a transformação da rede em um autômato celular, devido ao ganho na velocidade de processamento.
4. Outra possibilidade de estudo posterior, seria o acoplamento de duas ou mais redes definidas em diferentes pontos do espaço de parâmetros (a, b) .
5. É possível também definir-se uma rede de mapas acoplados, com um número variável $S_{i,j} \in \{3, 4, \dots, 8\}$ de mapas vizinhos para cada mapa da rede, podendo-se escolher várias distribuições para o conjunto de valores $\{S_{i,j}\}$. Com isso, todas as situações de vizinhanças encontradas para a rede cúbica e hexagonal poderiam ser encontradas em um só rede, e o efeito combinado desses diferentes níveis de acoplamento poderia ser estudado [JÁNOSI *et al.*, 1999].
6. Numa abordagem mais geral, pode-se remover completamente o conjunto de mapas da rede, associando a cada mapa X_t^k uma posição $\vec{r}_k = (x_k, y_k)$ sobre uma superfície contínua qualquer, por exemplo um toróide, uma esfera ou um plano, de modo

que cada mapa esteja acoplado com todos os mapas “vizinhos” $\{X_t^{k'}\}$, tais que $|\vec{r}_{k'} - \vec{r}_k| \leq \xi$, para algum raio ξ especificado. Diferentes densidades de mapas podem ser testadas sobre a superfície, constantes ou variando periodicamente ao longo do sistema, admitindo que este obedeça a uma condição periódica de contorno (se for finito). A generalização dessa abordagem para o espaço real tridimensional é imediata, apesar do enorme custo computacional adicionado pela adição de mais uma dimensão ao sistema.

7. Finalmente, outros tipos de mapas podem ser testados sobre a rede estudada, e a universalidade [MARCQ *et al.*, 1996, MARCQ *et al.*, 1998] dos resultados obtidos pode ser verificada, quanto a sua validade.

Apêndices

A - Índice Geral dos Estados Assintóticos

Apresentamos a seguir todas as imagens obtidas para os estados assintóticos da rede de mapas quárticos acoplados, para a rede quadrada e hexagonal, primeiros e segundos vizinhos, para as inicializações RANDOM, RING50%, PATCH3 e PATCH5, nesta ordem.

As imagens individuais foram obtidas para $\varepsilon \in [0.00, 0.99]$, tomadas a cada $\Delta\varepsilon = 0.01$ nos instantes $t = 9999$ e $t = 10000$, e foram condensadas em uma só tela, facilitando a visualização geral dos resultados gráficos, como também sua análise comparativa. As imagens de cada tela, estão dispostas em colunas, na ordem crescente dos valores do acoplamento, de modo que ε cresce em 0.01 a cada imagem, de cima para baixo, e em 0.10, da esquerda para a direita.

As imagens originais de tamanho 512×512 foram inicialmente reduzidas a $1/8$ do seu tamanho original, sem reamostragem (resampling) dos *pixels*, e posteriormente coladas uma a uma na imagem final, de modo que alguns detalhes menores podem ter-se perdido no processo de redução, ou mesmo no processo de impressão. As 256 cores da paleta original `rainbow.pal` foram mantidas sem alteração, e estão reproduzidas em todas as telas para referência.

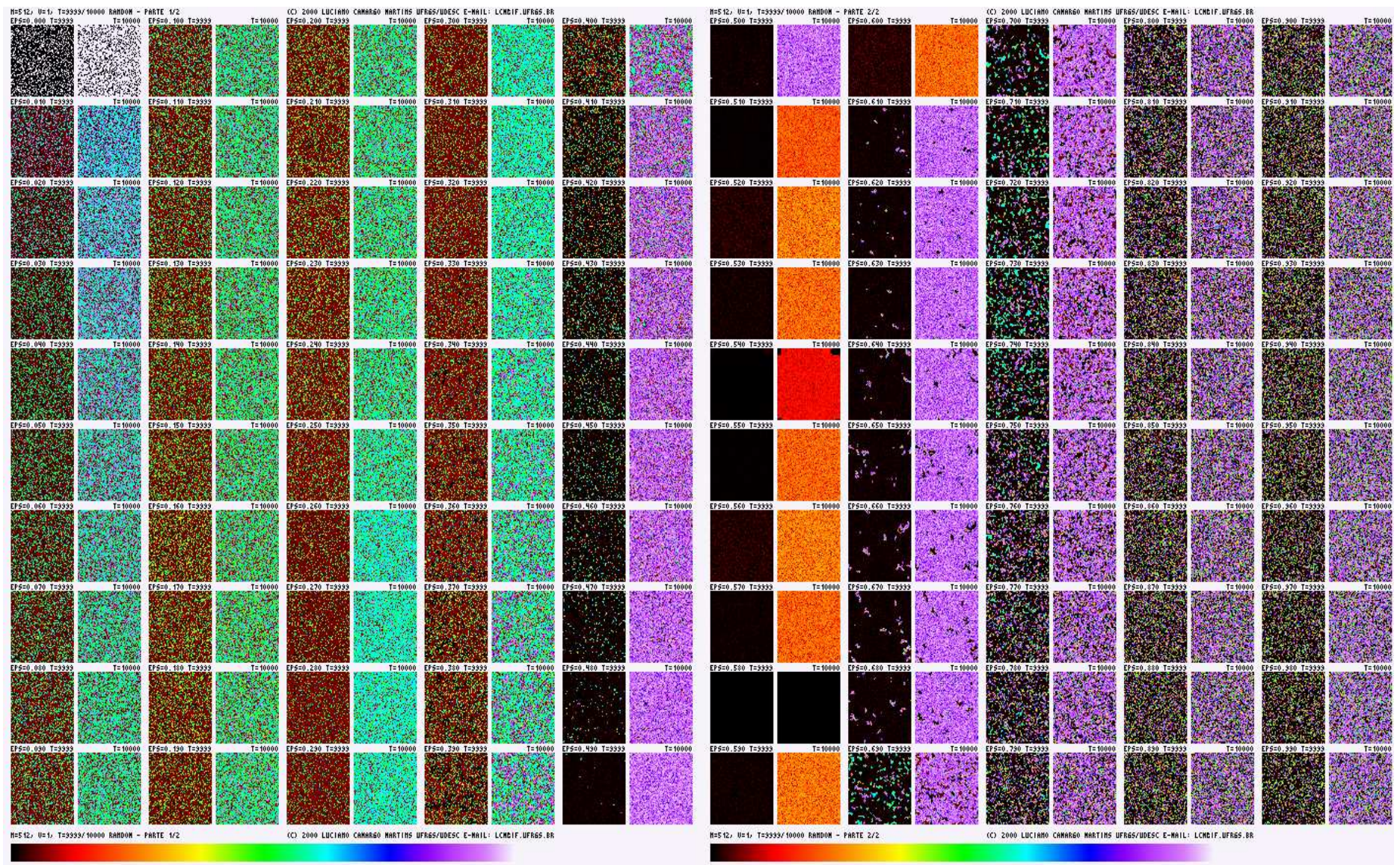


Figura 1: Rede QUADRADA, inicialização RANDOM, PRIMEIROS vizinhos, $t = 9999$ e $t = 10000$.

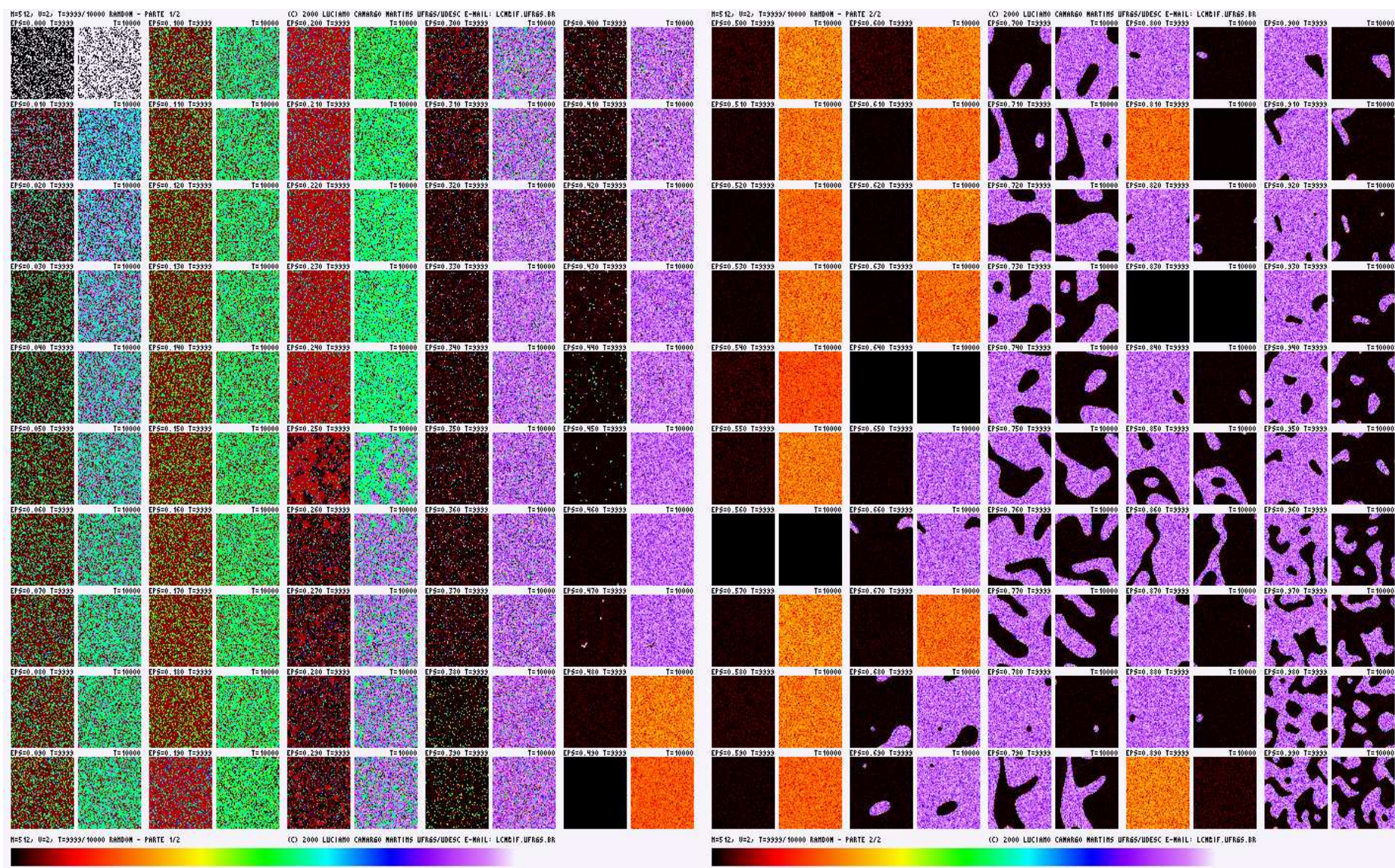


Figura 2: Rede QUADRADA, inicialização RANDOM, SEGUNDOS vizinhos, $t = 9999$ e $t = 10000$.

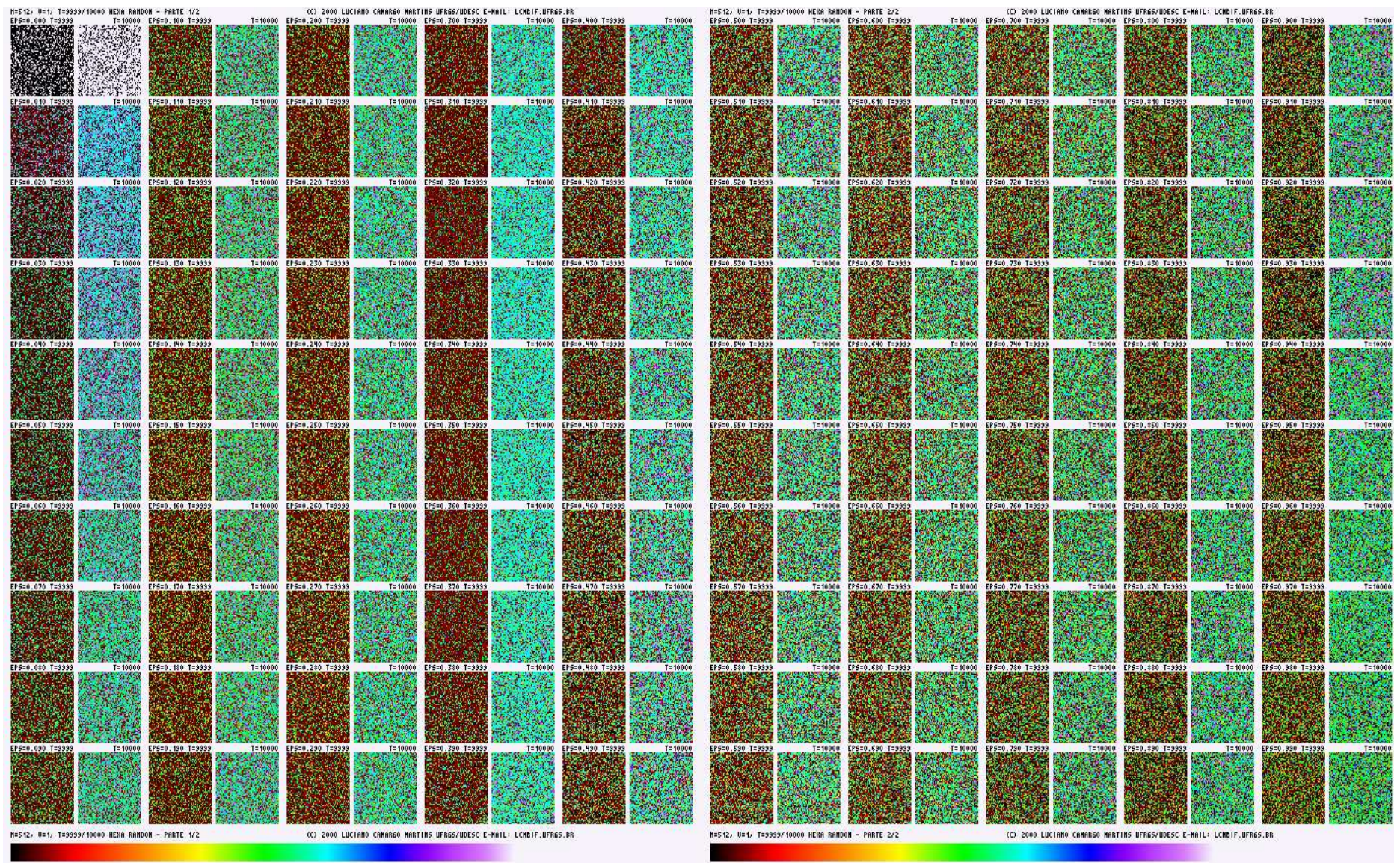


Figura 3: Rede HEXAGONAL, inicialização RANDOM, PRIMEIROS vizinhos, $t = 999$ e $t = 1000$.

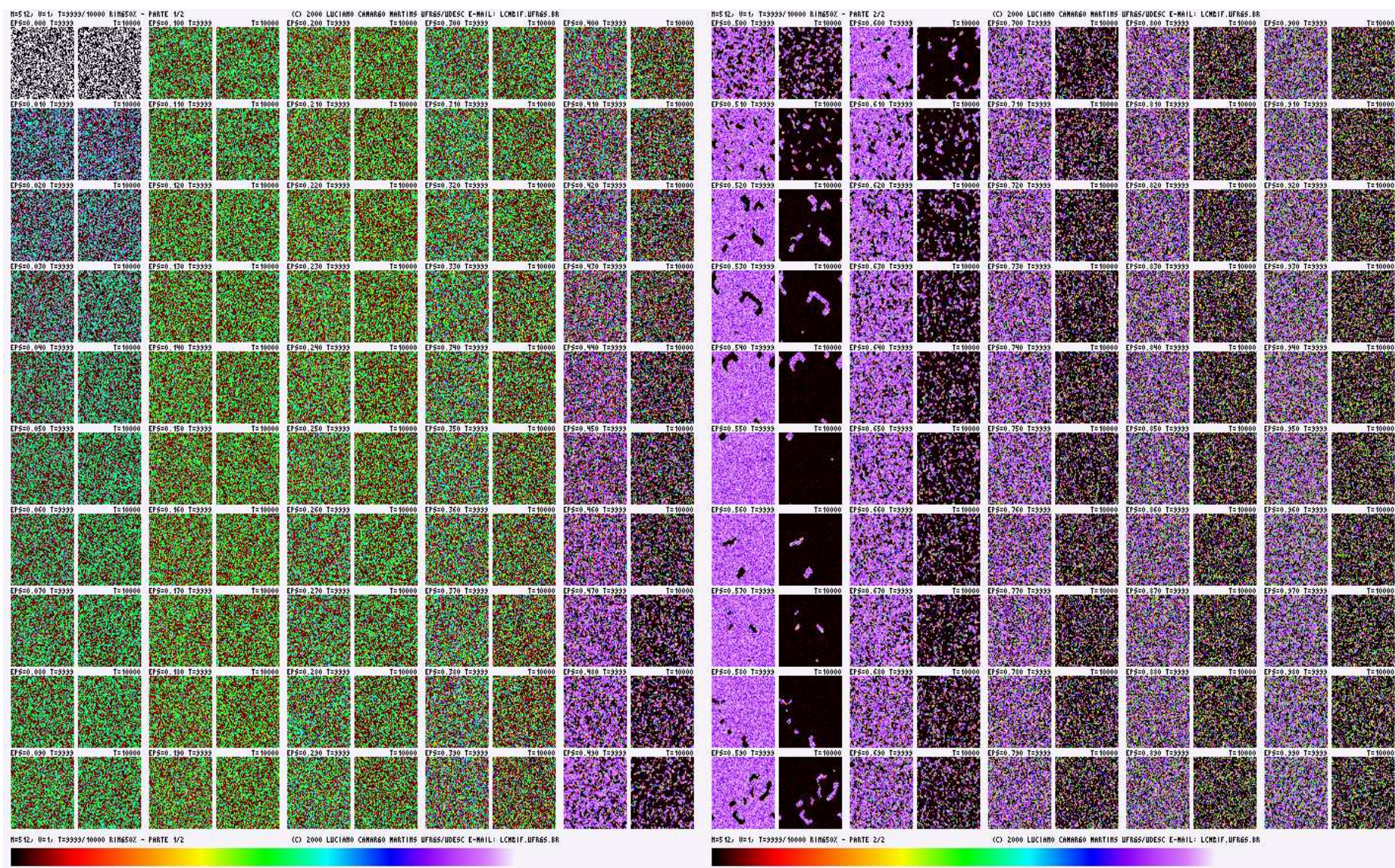


Figura 4: Rede QUADRADA, inicialização RING%50, PRIMEIROS vizinhos, $t = 9999$ e $t = 10000$.

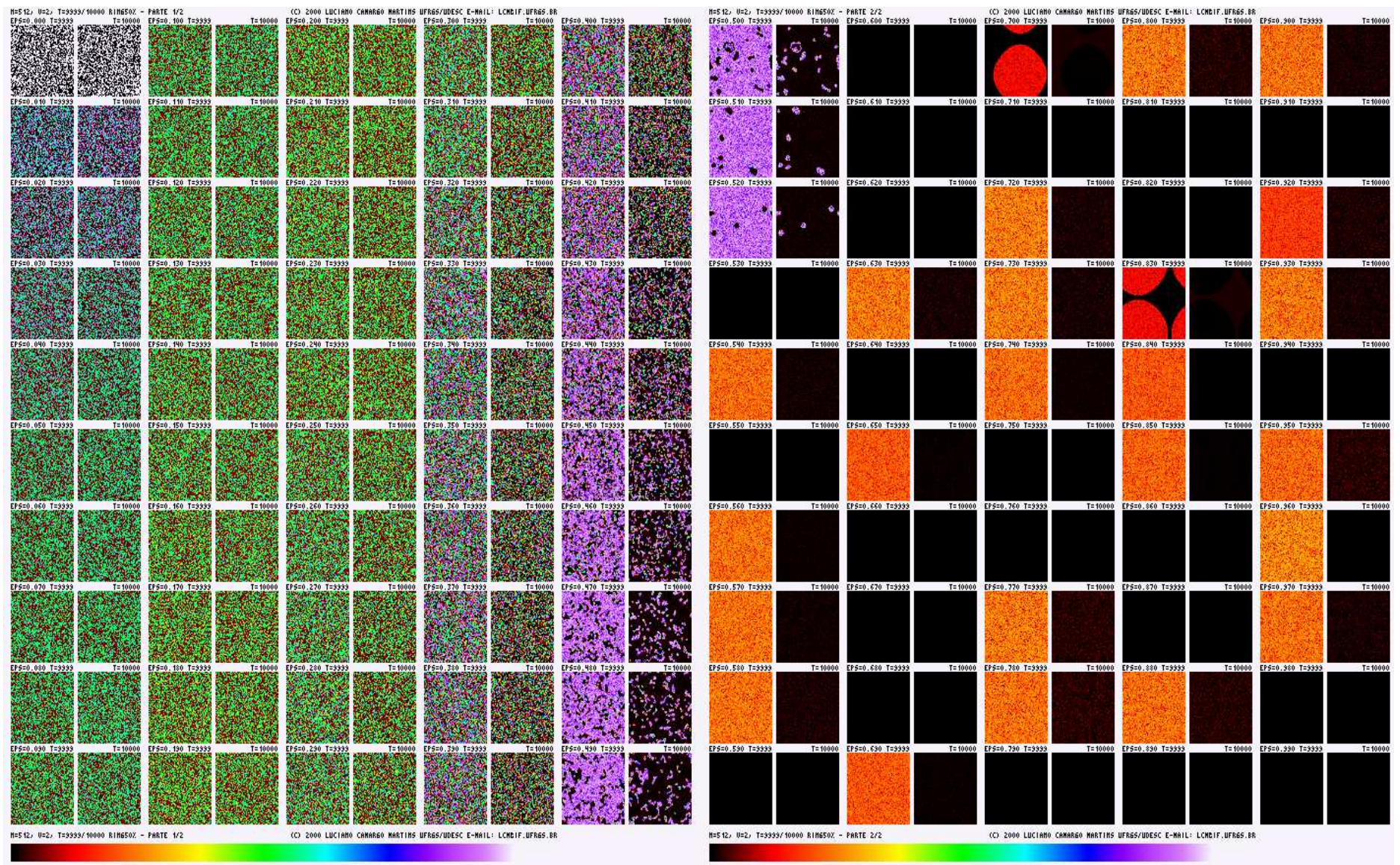


Figura 5: Rede QUADRADA, inicialização RING%50, SEGUNDOS vizinhos, $t = 9999$ e $t = 10000$.

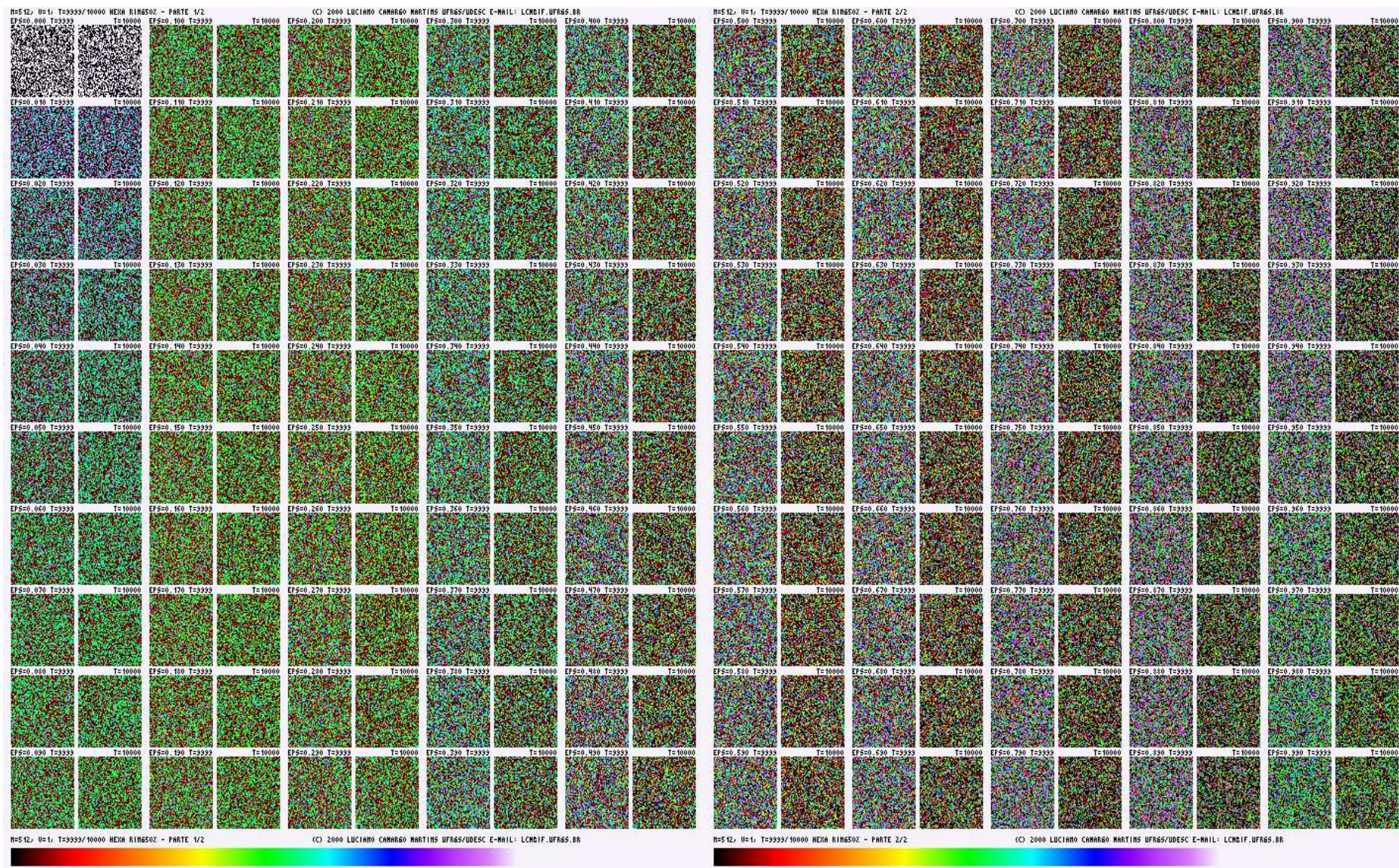


Figura 6: Rede HEXAGONAL, inicialização RING50%, PRIMEIROS vizinhos, $t = 9999$ e $t = 10000$.

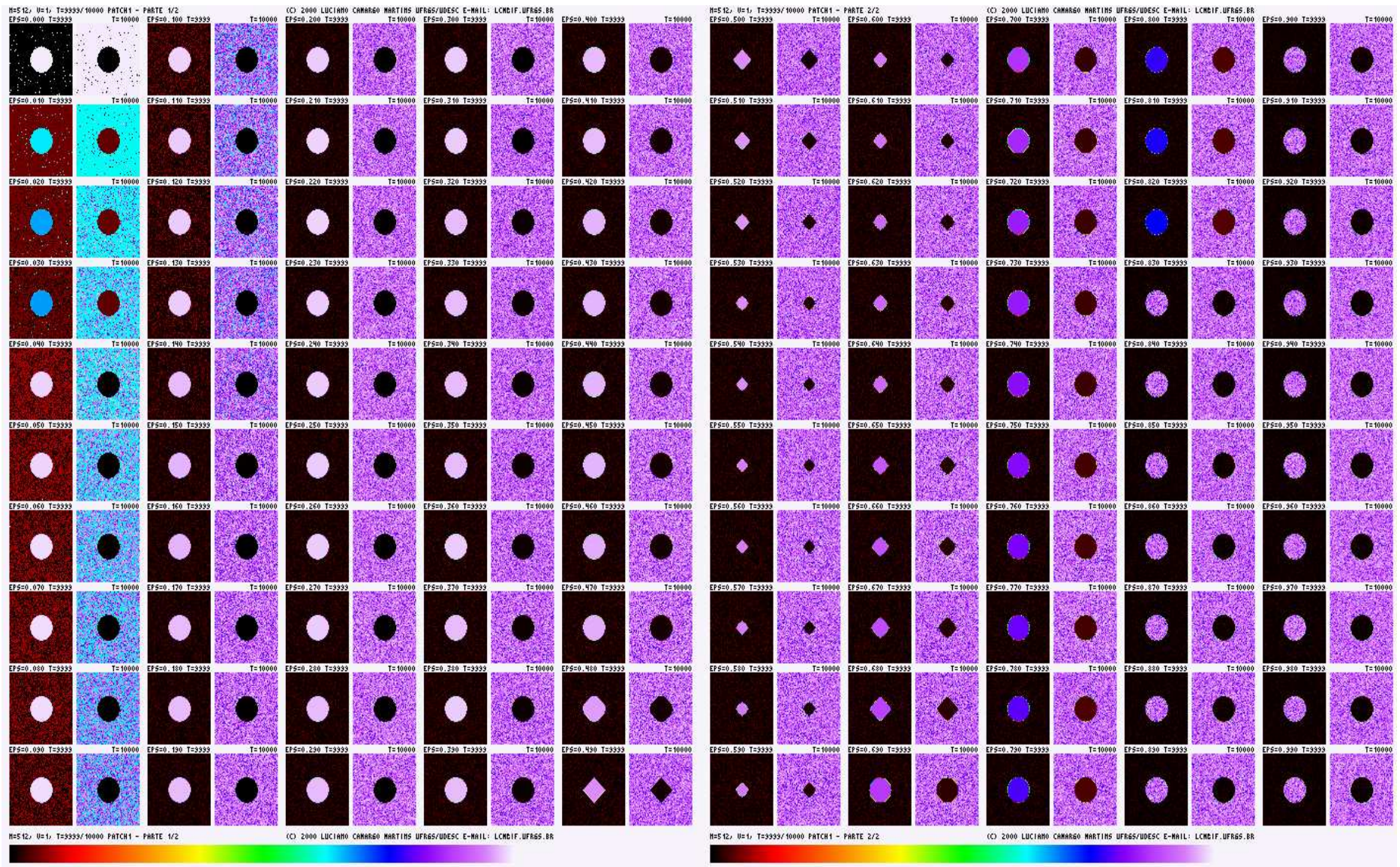


Figura 7: Rede QUADRADA, inicialização PATCH3, PRIMEIROS vizinhos, $t = 9999$ e $t = 10000$.

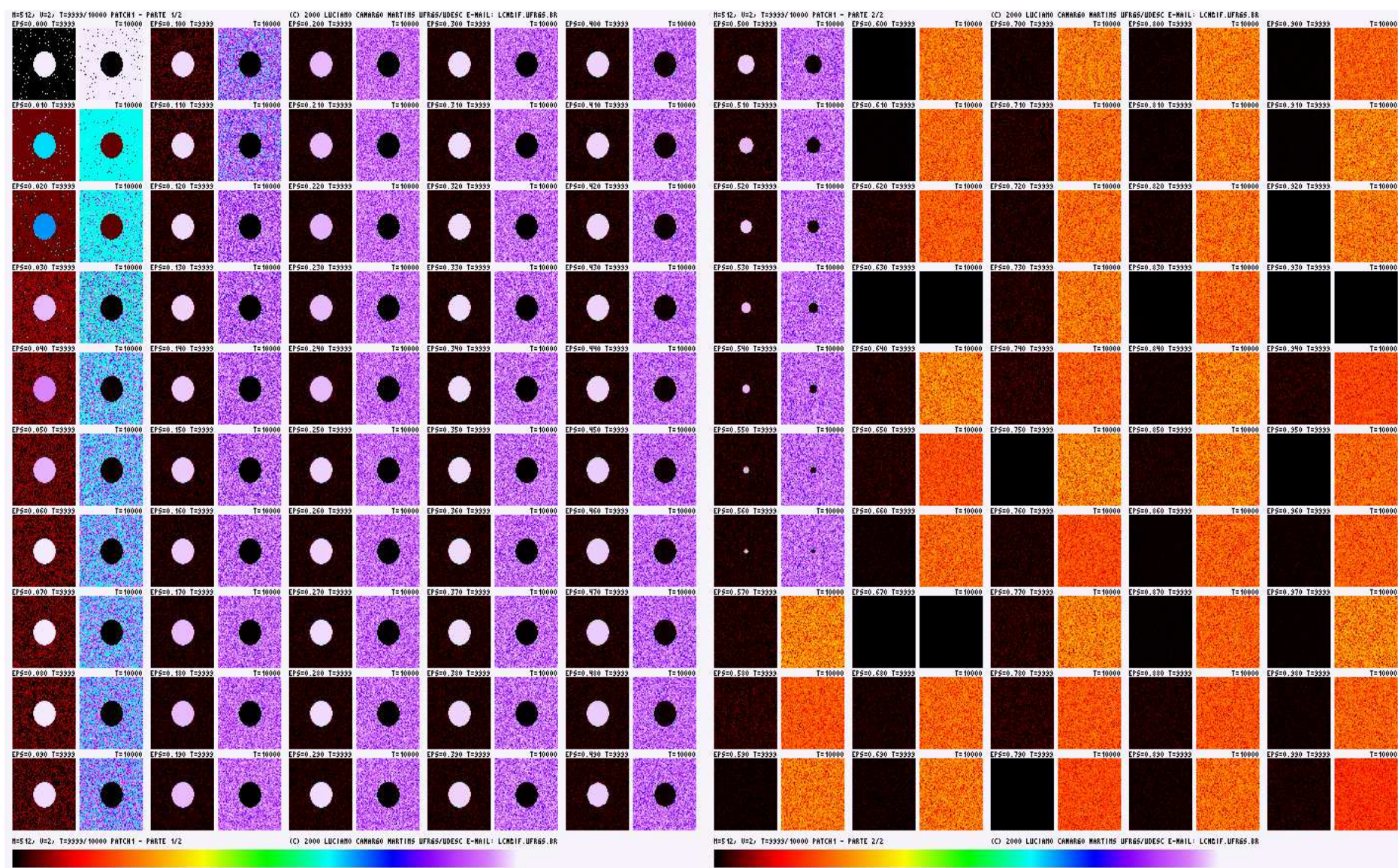


Figura 8: Rede QUADRADA, inicialização PATCH3, SEGUNDOS vizinhos, $t = 9999$ e $t = 10000$.

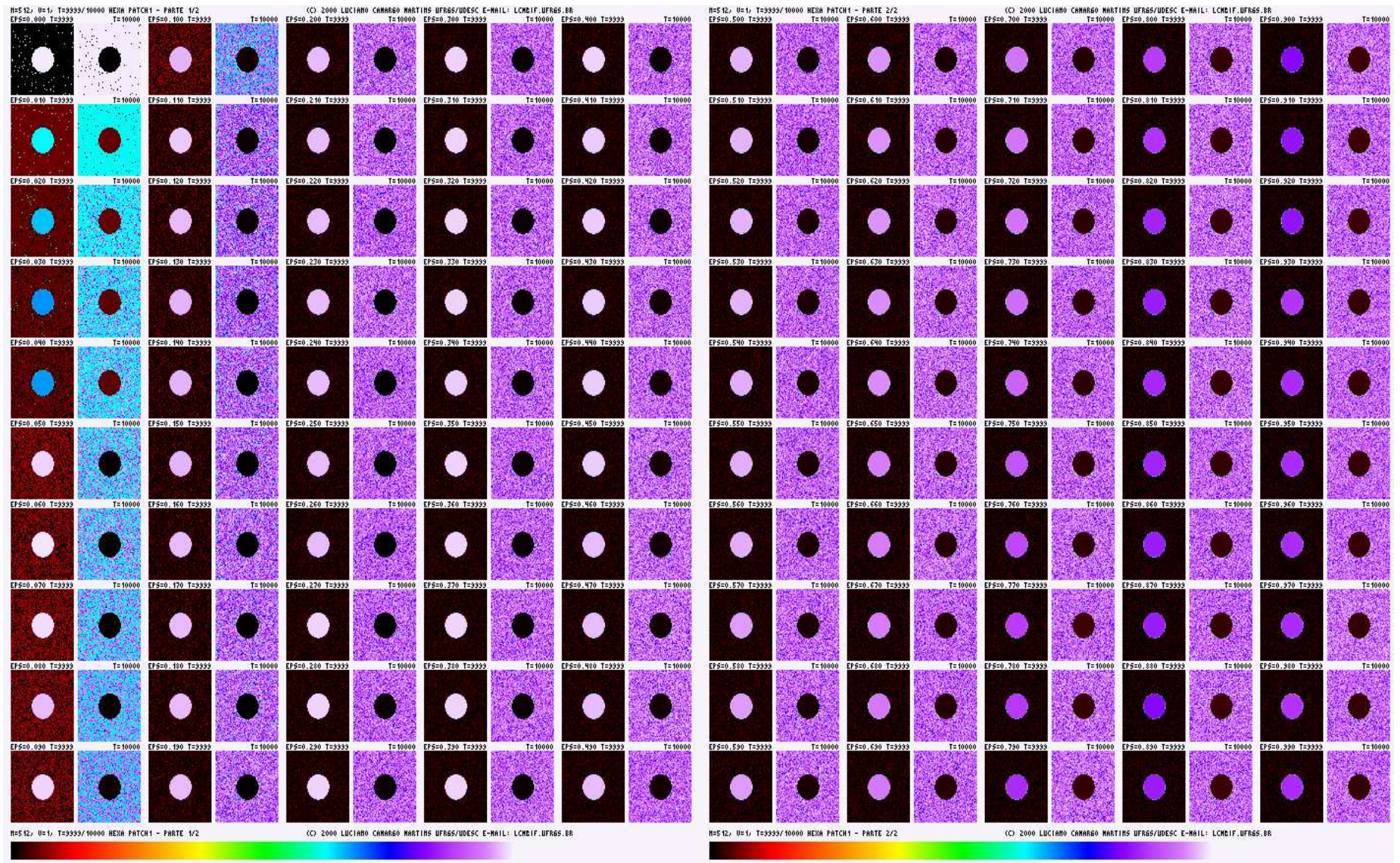


Figura 9: Rede HEXAGONAL, inicialização PATCH3, PRIMEIROS vizinhos, $t = 9999$ e $t = 10000$.

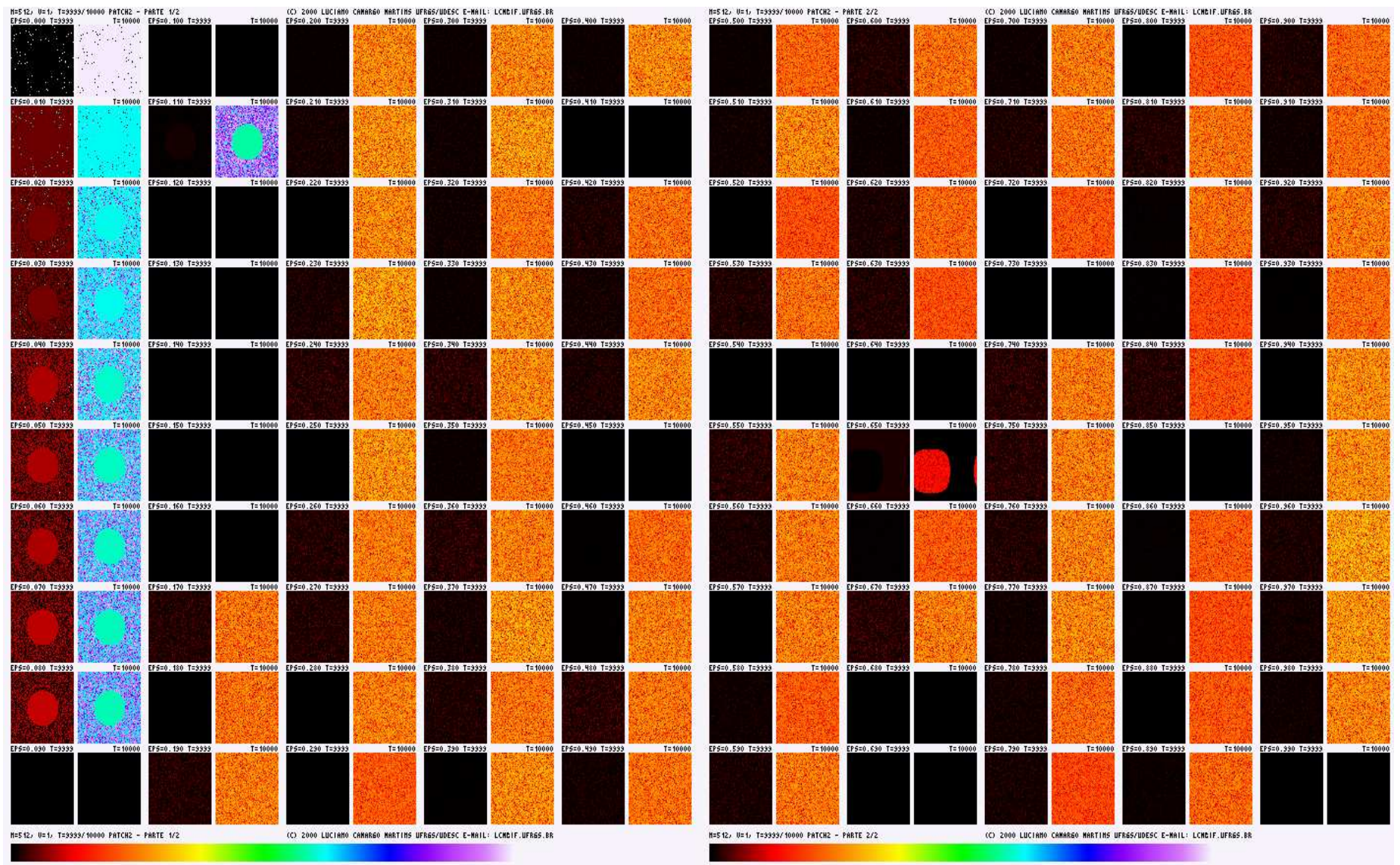


Figura 10: Rede QUADRADA, inicialização PATCH5, PRIMEIROS vizinhos, $t = 9999$ e $t = 10000$.

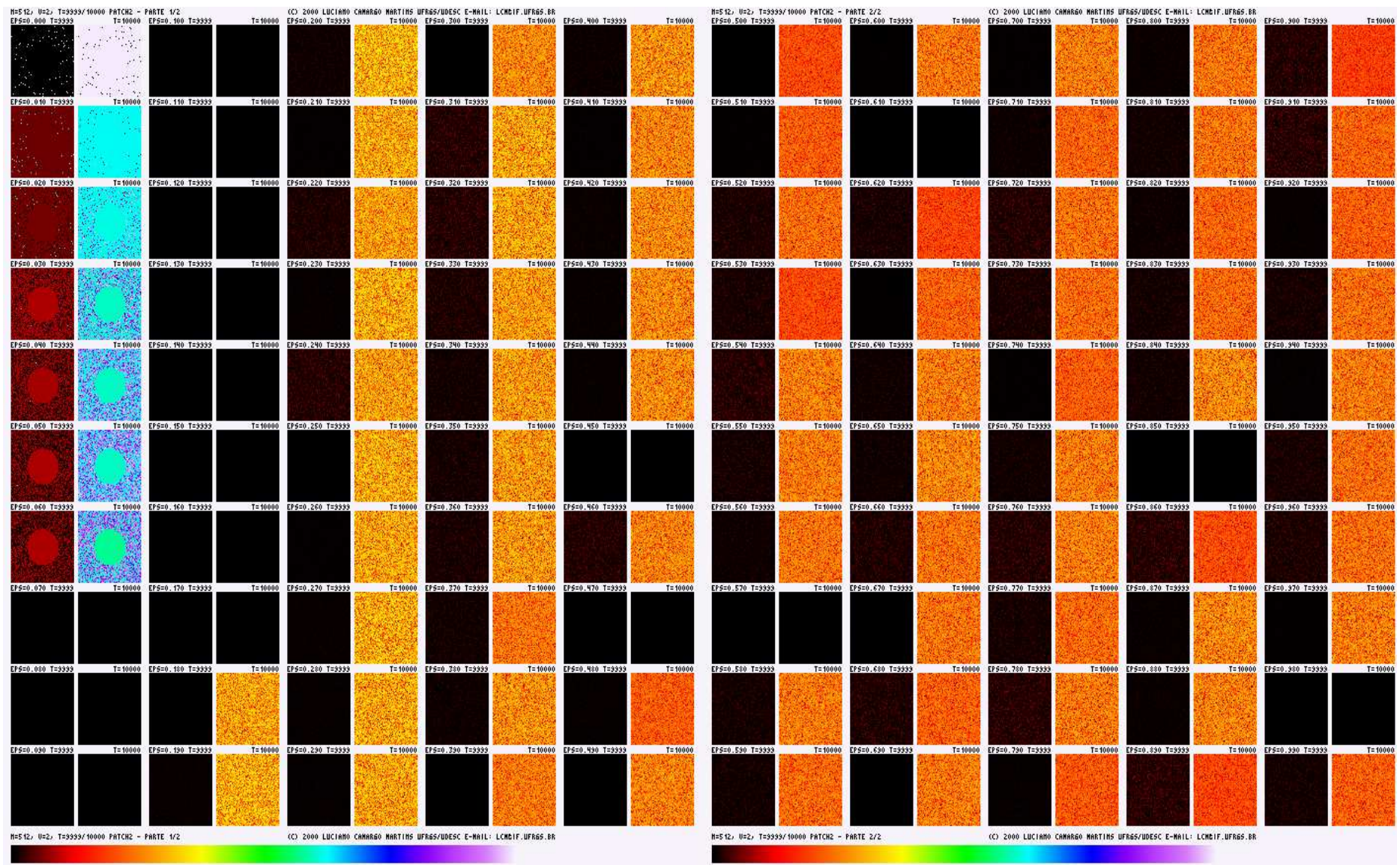


Figura 11: Rede QUADRADA, inicialização PATCH5, SEGUNDOS vizinhos, $t = 9999$ e $t = 10000$.

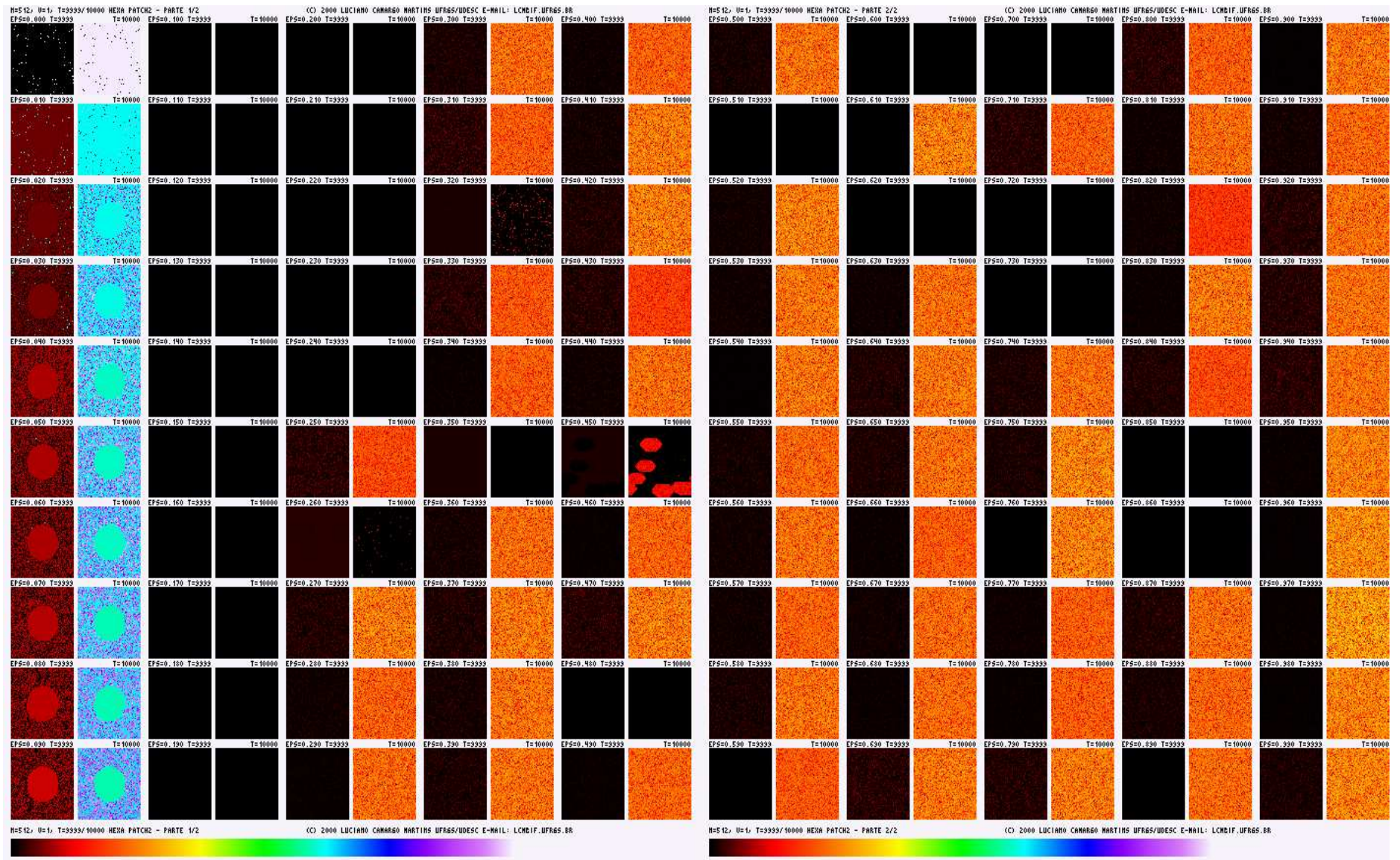


Figura 12: Rede HEXAGONAL, inicialização PATCH5, PRIMEIROS vizinhos, $t = 9999$ e $t = 10000$.

B - Fluxograma Básico do Programa qmcm1-2.0.0.c

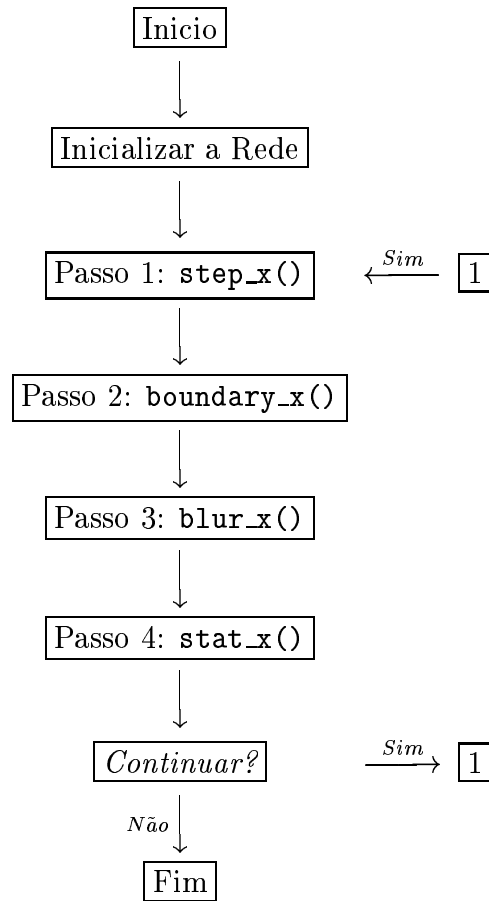


Figura 13: Fluxograma simplificado para o programa qmcm1-2.0.0.c .

C - Listagem do Programa qmcm1-2.0.0.c

```
/*
*****
* qmcm1-2.0.0.c - QUARTIC MAP 2D Coupled Map Lattices (CML)
* (C) 2000 de Luciano C. Martins (fonte C) & Leonardo G. Brunnet, Orient. *
* UDESC/CCT/FEJ/DFIS, Joinville-SC, Brasil e-mail: lcm@dcc.fej.udesc.br *
* Instituto de Fisica-UFRGS, P. Alegre-RS, Brasil E-mail: lcm@if.ufrgs.br *
*****

NOTA DO AUTOR
-----

Este programa foi escrito na linguagem C, para o compilador GNU C distribu-
ido com o Linux, e usa o suporte grafico da biblioteca SVGALib versao 1.3+.

Para poder abrir o terminal grafico usado pela biblioteca SVGALib, eh neces-
s'ario um console proprietario, ou senha de root da maquina/estacao Linux! E
o terminal grafico *NAO* pode ser exportado (terminal virtual), mas pode ser
aberto sobre a interface X11.

Compilar o programa com a seguinte linha de comando, num shell Linux:

gcc -lm -lvga1 -lvga -O2 qmcm1-2.0.0.c <enter>

Executar o programa, chamando o arquivo executavel gerado:

./a.out <enter>

Durante a execucao do programa, a tecla ESC permite fecha-lo e retornar ao
shell do sistema operacional.

As paletas de cores RGB usadas serao carregadas do diretorio ./rgb, os
arquivos de imagens gerados serao salvos no diretorio ./pic e os arquivos
de dados serao gravados em ./dat .

Porto Alegre-RS, 31/07/2000.
LCM
-----
*/

#include <stdio.h> /* suporte para as bibliotecas */
#include <vga.h> /* do sistema, a serem usadas */
#include <vgagl.h>
#include <stdlib.h>
#include <math.h>
#include <float.h>
#include <vgakeyboard.h>
#include <vgamouse.h>
#include <time.h>
#include <string.h>

#include "ufrgs-2.0.0.c" /* biblioteca matematica */
#include "mickey-2.0.0.c" /* suporte para o mouse */

#define NMAX 512 /* tamanho maximo da rede NxN */
#define NMIN 32 /* tamanho minimo da rede NxN */
#define NMAXX (NMAX+2) /* tamanho da rede + contorno */
#define NMAX2 (NMAX*NMAX) /* constantes... */
#define NMAXH (NMAX/2)
#define NL 2
#define NC 256 /* num. de cores do modo SVGA */
#define NCM1 (NC-1)
#define Nchess 16 /* resolucao do zoom da rede */
#define MOUSE_DELAY 16
#define ESC 27
#define SPACE 32
#define ENTER 13
#define TRUE 1
#define FALSE 0
#define TRANSIENT 1000 /* num. iteracoes descartadas */
#define SAMPLE 1000 /* num. iteracoes computadas */

/* os 13 tipos de inicializacao (condicao inicial-CI) pre-definidas: */
typedef enum {
    randomic, /* distrib. inicial uniforme (-R,+R) dos mapas (default) */
    pie1, /* circulo, atrator ciclo-2 x2 dentro, x1 fora */
    pie2, /* circulo, atrator ciclo-2 x1 dentro, x2 fora */
    pie3, /* circulo, x2 dentro, aleatorio uniforme fora */

```

```

pie4,          /* circulo, aleat[orio uniforme dentro, x2 fora */
chess,        /* tabuleiro de xadrez, um valor aleat. p/ cada quadro */
gaussian,     /* distribuicao gaussiana com media zero */
constant,    /* uma valor aleatorio igual para toda a rede */
randlocked,  /* distribuicao uniforme discreta: 50% de x1, 50% de x2 */
chesslocked, /* xadrez com cada quadro inicializado em x1 ou x2 (50%) */
ring,        /* aleat. uniforme para valores entre em [Rmin,Rmax] */
mixed,       /* meia rede fixo, meia rede aleatorio uniforme */
squared     /* quadrado, dentro aleatorio unif., fora valor fixo */
} twarmup;

/* os tipos de paletas de 256 cores disponiveis para visualizar os graficos*/
typedef enum {
gray,        /* do preto ao branco */
rainbow,     /* tipo arco-iris (default) */
bluered1,   /* do azul ao vermelho 1 */
bluered2,   /* do azul ao vermelho 2 */
red,        /* do preto ao vermelho */
green,      /* do preto ao verde */
blue,       /* do preto ao azul */
copper,     /* do preto ao cobre metalico */
random_     /* cores aleatorias, unica em tons decontínuos */
} tvalidpal;

twarmup
warmup=randomic; /* inicializacao padrao (default)

tvalidpal
pal=rainbow; /* paleta de cores padrao (default)

double
x[NMAXX][NMAXX], /* matriz principal da para os mapas da rede NMAXxNMAX */
y[NMAXX][NMAXX], /* matriz auxiliar para o processo difusivo */
EPS=0.0,         /* constante de acoplamento dos mapas da rede - EPSILON */
RAD,            /* raio maximo para os valores aleatorios iniciais (CI) */
a =0.3500,      /* valor do parametro "a", para todos os mapas da rede */
b =1.3500,      /* valor do parametro "b", para todos os mapas da rede */
delta_a =0.0100, /* incremento/decremento para "a" durante a execucao */
delta_b =0.0100, /* incremento/decremento para "b" durante a execucao */
delta_eps=0.010, /* incremento/decremento para a constante epsilon (EPS) */
x1 = -1.227591, /* menor valor do atrator do ciclo-2 para (a0,b0) */
x2 = -0.011399, /* maior valor do atrator do ciclo-2 para (a0,b0)

err1, err2,     /* contadores para as medias usadas na "rugosidade"
sum1=0.0,
sum2=0.0,
sum12=0.0,
sum22=0.0,
mean1, mean2,
avg,

c1,c2,c3,      /* coeficientes auxiliares para a expressao acoplamento */

min_avg,       /* valores m;aximos e minimos das variaveis globais
max_avg,
min_x,
max_x,
min_x_global,
max_x_global,
min_dx,
max_dx,
max_ddx,
most_probable_dx;

long
N = NMAX/2,    /* tamanho atual da rede NxN
N2 = NMAX2/4,
NP1 = NMAX/2+1,
NM1 = NMAX/2-1,
NH = NMAX/4,

main_k,        /* contador para o loop-sem-fim central do programa
total_k=0,     /* contador acumulativo, zerado so no inicio do prog.
start,         /* instante inicial do programa
stop,          /* final do programa
f[NC],         /* vetor de frequencia para o espectro de x[][]
fdx[NC],       /* vetor de freq. para o espectro de delta_x[][]
max_f,         /* freq. maxima do espectro da rede
max_f_global;  /* freq. maxima global, para todas as iteracoes

char
vers_id[] = /* identificadores do software
date_id[] = "31/07/2000",

/* mascaras para os nomes dos arquivos de dados
psfile_mask[] = "pic/quartic-map-cml-%s-%dx%d-%d.eps",
bmpfile_mask[] = "pic/quartic-map-cml-%s-%dx%d-%d.bmp",
datfile_mask[] = "dat/quartic-map-cml-%s-%dx%d-%d.dat",

```

```

dirdata[]      = "dat";

int            /* PARAMETROS PRINCIPAIS DA SIMULACAO *****/
FREQ=50,      /* frequencia de atualizacao das saidas graficas SVGA */
S=4,         /* numero de primeiros vizinhos em d=2 */
get_2nd=FALSE, /* se TRUE, considera vizinhos ate segunda ordem */
lock_mm=FALSE, /* se TRUE, usa os extremos globais da rede {min,max} */
fake_ovrflw=TRUE, /* se TRUE, reinicia a rede se houver overflow de x[] */
go_hexa=FALSE, /* se TRUE, usa a rede hexagonal */
grd_ok=FALSE, /* se TRUE, plota as medias globais de x[] da rede */
hen_ok=FALSE,
shift=TRUE,   /* inverte a fase visualizada na tela */
auto_eps=TRUE, /* se TRUE, usa sempre o epsilon democratico, */

/* FLAGS DE CONTROLE DA INTERFACE GRAFICA *****/
plot_m=TRUE,  /* se TRUE, plota o mapa da rede, a cada FREQ iteradas */
plot_s=TRUE,  /* se TRUE, plota os espectros de x[] e delta_x[]s */
plot_a=TRUE,  /* se TRUE, plota as medias de x[] sobre a rede */
plot_3=FALSE, /* se TRUE, plota o mapa em 3D */
plot_p=FALSE, /* se TRUE, plota o mapa 3D com pontos apenas (+rapido) */
plot_n=FALSE, /* se TRUE, plota os espectros com linhas verticais */
plot_zm=TRUE, /* se TRUE, plota o zoom da tela apontada pelo mouse */
plot_v=TRUE,  /* se TRUE, plota as variaveis principais da rede */
save_all=FALSE, /* if TRUE, salva os dados a cada save_freq loops */
z_deep=6,     /* 0 1 2 3 4 5 6 7 alturas 3D */
z[8]=        {0xff,0x80,0x40,0x20,0x0f,0x08,0x04,0x02},
zoom=0,       /* NMAX = N*2^zoom, a altura "z" no plot 3Dthe */
saved=0,      /* numero total de arquivos salvos */
save_freq=500,
save_discard=0, /* comeca a salvar os dados apos save_discard loops */

plotted_sample1=FALSE, /* flags de controle do status das janelas... */
plotted_sample2=FALSE,
plotted_zoom=FALSE,
plotted_spectrum=FALSE,
plotted_dx_spectrum=FALSE,
plotted_avg=FALSE,
inverted_pal=FALSE; /* se TRUE, a paleta foi invertida (negativo) */

/* Estatistica para a rugosidade e o erro normal esperado associado */
double mean_dx2(main_k)
long main_k;
{
    int i,j;
    double mean, dx1, dx2, sum=0.0;

    for(j=1;j<=N;j++){
for(i=1;i<=N;i++){
        dx1 = x[i][j]-x[i][j-1];
        dx2 = x[i][j]-x[i-1][j];
        sum = sum + dx1*dx1 + dx2*dx2;
    }
    }
    mean= (double) sum/(N2<<1);
    if(main_k>TRANSIENT){
if((main_k % 2)==0){
        sum1=sum1+mean;
        sum12=sum12+mean*mean;
    }
    else {
        sum2=sum2+mean;
        sum22=sum22+mean*mean;
    }
    }
    return mean;
}

/* mostra a posicao do mouse na tela, ou sobre a rede */
void plot_site_info(void)
{
    char site_info[40];
    int mx,my;
    double percent;

    mouse_update();
    mx=mouse_getx();
    my=mouse_gety();
    percent=100.0*(x[mx][my]-min_x)/(max_x-min_x);

    if(mouse_inbox(1,1,N,N)){
        sprintf(site_info,"x[%3d,%3d]=%12.8f  =%5.1f%",mx,my,x[mx][my],percent);
        strxy(8,520,site_info,NCM1);
        setcolor(NCM1);
        if(plot_zm && !plot_3) rectangle(mx-min(mx,25),my-min(mx,25),mx+1,my+1);
    }
    else {
        sprintf(site_info,"Mouse is at (%3d,%3d)          ",mx,my);
        strxy(8,520,site_info,NCM1);
    }
}

```

```

}
}
/* mostra uma barra com a paleta de 256 cores */
void make_pal(x_,y1,y2)
int x_,y1,y2;
{
    int x,j;
    for (j=0;j<NC;j++)
    {
        setcolor(j);
        x=x_+j;
        line(x,y1,x,y2);
        line(x-1,y1,x-1,y2);
        if(((j % 32)==0) || (j==255)) {
            setcolor(NCM1);
            line(x,y2+5,x,y2+10);
        }
    }
}

/* plota a janela com o espectro de distribuicao dos de x[][] da rede */
void plot_spectrum(x,y1,y2,flag)
int x,y1,y2,*flag;
{
    int j,s,tmp=*flag;
    long max_f=0;

    if(!tmp){
        setcolor(NCM1);
        rectangle(x-2,y1-3,x+257,y2+8);
        strxy(x+5,y1-10," X-spectrum ",NCM1);
        make_pal(x,y2+2,y2+6);
        *flag=TRUE;
    }
    for(j=0;j<NC;j++) if(f[j]>max_f) max_f=f[j];

    for(j=0;j<NC;j++)
    {
        setcolor(0);
        line(x+j,y1-1,x+j,y2);
        s=(int) (y2-y1+1)*f[j]/max_f;
        setcolor(NCM1);
        if (f[j]>0){
            if (plot_n) line(x+j,y2,x+j,y2-s);
            else vga_drawpixel(x+j,y2-s);
        }
    }
}

/* salva as freq. normalizadas do espectro de x[][] num arquivo de dados */
void save_spectrum()
{
    int j;
    double s, freq; /* 0.0 - 1.0 */
    FILE *sf;
    char datfile[128];

    sprintf(datfile,"%s/qmcm1-sx-e=%5.3f-N=%d-i=%d.dat",dirdata,EPS,N,main_k);
    if((sf=fopen(datfile,"wb"))==NULL) return;

    if(!lock_mm) for(j=0;j<NC;j++) if(f[j]>max_f) max_f=f[j];
    else max_f=max_f_global;

    fprintf(sf,"# N      = %8d\n",N);
    fprintf(sf,"# main_k = %8d\n",main_k);
    fprintf(sf,"# EPS    = %8.5f\n",EPS);
    fprintf(sf,"# RAD    = %8.5f\n",RAD);
    fprintf(sf,"# a      = %8.5f\n",a);
    fprintf(sf,"# b      = %8.5f\n",b);
    if(get_2nd) fprintf(sf,"# 2nd    =      TRUE\n");
    else{
        fprintf(sf,"# 2nd    =      FALSE\n");
        if(go_hexa) fprintf(sf,"# Rede  HEXAGONAL\n");
    }

    for(j=0;j<NC;j++){
        freq=(double) f[j]/max_f;
        fprintf(sf,"%15.12f %15.12f\n",
            dequantize(j,0,NC-1,min_x_global,max_x_global,xRIGHT),freq);
    }
    fclose(sf);
}

/* plota a janela com o espectro das transicoes delta_x[][] da rede */
void plot_dx_spectrum(x,y1,y2,flag)
int x,y1,y2,*flag;
{

```



```

int j,s,tmp=*flag;
long max_f_dx=-10000;
for(j=0;j<NC;j++)
{
  if(fdx[j]>max_f_dx) max_f_dx=fdx[j];
}
if(!tmp){
  setcolor(NCM1);
  rectangle(x-2,y1-3,x+257,y2+8);
  strxy(x+5,y1-10," dX-spectrum ",NCM1);
  make_pal(x,y2+2,y2+6);
  *flag=TRUE;
}
for(j=0;j<NC;j++)
{
  setcolor(0);
  line(x+j,y1-1,x+j,y2);
  s=(int) (y2-y1+1)*fdx[j]/max_f_dx;
  setcolor(NCM1);
  if (f[j]>0){
    if (plot_n) line(x+j,y2,x+j,y2-s);
    else vga_drawpixel(x+j,y2-s);
  }
}
}

/* plota amostras dos valores do centro da rede (i=1..N,j=N/2) em corte */
void plot_sample(x1,y1,x2,y2,flag)
int x1,y1,x2,y2, *flag;
{
  int tmp, x_,y_,i,j,m,c,y1p1=y1+1,y2m1=y2-1,x1p1=x1+1,x2m1=x2-1,
  x3=x2+2,x4=x3+4;

  m=N>>1;

  setcolor(0);

  tmp = *flag;
  for(j=y1p1;j<y2;j++) line(x1p1,j,x2m1,j);

  if(!tmp){
    for(j=y1p1;j<y2;j++){
      c=quantize((double) j, (double) y1p1, (double) y2m1, 0, NCM1, FALSE);
      setcolor(c);
      line(x3,j,x4,j);
    }
    setcolor(NCM1);
    rectangle(x1-1,y1-1,x3+6,y2+1);
    *flag=TRUE;
  }

  setcolor(NCM1);

  for(i=1;i<=N;i++){
    x_ = (1+x1+(x2-x1-2)*(1.0*(i-1)/(N-1)));
    y_ = (y1+1+(y2-y1-2)*(max_x-x[i][m])/(max_x-min_x));
    c = quant(x[i][m],min_x,max_x,0,NC-1);
    putpixel(x_,y_,c);
  }
}

/* plota as medias de x[][] sobre toda a rede */
void plot_avg(x1,y1,x2,y2,flag)
int x1,y1,x2,y2,*flag;
{
  int j,x_,y_,tmp=*flag;

  if(!tmp){
    setcolor(NCM1);
    rectangle(x1-1,y1-1,x2+1,y2+1);
    strxy(x1+5,y1-8," Global averages ",NCM1);
    for(j=0;j<=2;j++){
      setcolor(NCM1);
      vga_drawline(x1-5,y1+j*(y2-y1+1)/2,x1-1,y1+j*(y2-y1+1)/2);
      switch(j){
        case 0: strxy(x1-40,y1+j*(y2-y1+1)/2-3,"+1.5",NCM1); break;
        case 1: strxy(x1-40,y1+j*(y2-y1+1)/2-3," 0.0",NCM1); break;
        case 2: strxy(x1-40,y1+j*(y2-y1+1)/2-3,"-1.5",NCM1); break;
      }
    }
    *flag=TRUE;
  }
}

x_=x1+((main_k-NL) % (x2-x1));
y_=quantize(avg,-1.5,1.5,y1,y2,FALSE);

```

```

    setcolor(0);
    for(j=y1;j<=y2;j++) vga_drawpixel(x_,j);

    putpixel(x_,y_,NCM1);
    if(main_k % 5 == 0) putpixel(x_,1+(y2+y1)/2,NCM1);
}

/* plota a janela com o zoom da regio apontada pelo mouse */
void plot_zoom(x1,y1,x2,y2,nx,ny)
int x1,y1,x2,y2,nx,ny;
{
    int mx,my,i,j,ii,jj,dx=(x2-x1+1)/nx,dy=(y2-y1+1)/ny,c,cx=x1+dx-1,cy=y1+dy-1;

    if(!plotted_zoom){
        setcolor(NCM1);
        rectangle(x1-1,y1-1,x2+1,y2+1);
        plotted_zoom=TRUE;
    }
    mouse_update();
    if(!mouse_inbox(x1,y1,x2,y2)){
        mx=mouse_getx();
        my=mouse_gety();
        if((mx>nx) && (my>ny)){
            for(j=0;j<ny;j++){
                for(i=0;i<nx;i++){
                    ii=i*dx;
                    jj=j*dy;
                    c=vga_getpixel(mx-nx+i,my-ny+j);
                    rectfill(x1+ii,y1+jj,cx+ii,cy+jj,c);
                }
            }
        }
    }
}

/*
*****
plot 2D da rede, com a paleta escolhida. Os valores de x[] [] sao
mapeados discretamente de 0 a 255, sobre as cores da paleta.
*IMPORTANTE* Atualiza o vetor de freq. f[]
*****
*/
void plot_map(k,min_x,max_x)
long k;
double min_x,max_x;
{
    register int i,j;
    unsigned char c[N], *cptr;

    for(i=0;i<NC;i++) f[i]=0;

    cptr=&c[0];
    for(j=1;j<=N;j++){
        if(j % MOUSE_DELAY) mouse_move();
        for(i=0;i<N;i++){
            if(lock_mm) c[i]=(unsigned char) quantize(x[i+1][j],
min_x_global,max_x_global,0,NCM1);
            else c[i]=(unsigned char) quantize(x[i+1][j],min_x,
max_x,0,NCM1);
            f[c[i]]++;
        }
        vga_drawscansegment(cptr,1,j,N);
    }
    for(i=0;i<NC;i++){
        if(f[i]>max_f) max_f=f[i];
    }
    if((main_k>TRANSIENT) && (max_f>max_f_global)) max_f_global=max_f;
}

/*
*****
plot 3D da rede, com a paleta escolhida. Os valores de x[] [] sao
mapeados discretamente de 0 a 255, sobre as cores da paleta. Atualiza o
vetor de freq. f[]
*****
*/
void plot3d_map(k,min_x,max_x)
long k;
double min_x,max_x;
{
    unsigned int i,j,xx,yy,color,nh,jj;

    for(i=0;i<NC;i++) f[i]=0;

    setcolor(0);
    for(j=(N>>1)-z[z_deep];j<=N;j++){
        for(i=15;i<=(3*N/2)+25;i++){
            vga_drawpixel(i,j);
        }
    }
    nh=N>>1;
}

```

```

    if (plot_p){
    for(j=1;j<=N;j++){
        if(j % MOUSE_DELAY) mouse_move();
        jj=j>>1;
        for(i=1;i<=N;i++){
            if(lock_mm) color=quantize(x[i][j],min_x_global,
max_x_global,0,NCM1);
            else color=quantize(x[i][j],min_x,max_x,0,NCM1);
            f[color]++;
            xx=nh+20+i-jj;
            yy=nh+jj;
            setcolor(color);
            vga_drawpixel(xx,yy-(color>>z_deep));
        }
    }
    }
    else{
    for(j=1;j<=N;j=j+2){
        jj=j>>1;
        if(j % MOUSE_DELAY) mouse_move();
        for(i=1;i<=N;i++){
            if(lock_mm) color=quantize(x[i][j],min_x_global,max_x_global,0,NCM1);
            else color=quantize(x[i][j],min_x,max_x,0,NCM1);
            f[color]++;
            xx=nh+20+i-jj;
            yy=nh+jj;
            setcolor(color);
            line(xx,yy,xx,yy-(color>>z_deep));
        }
    }
    }
    }
    for(i=0;i<=NC;i++){
    if(f[i]>max_f) max_f=f[i];
    }
    if((main_k>TRANSIENT) && (max_f>max_f_global)) max_f_global=max_f;
}

/* plota os parametros constantes da simulacao */
void plot_const(l) /* != */
int l;
{
    int w1=170, w2=110, gmx=vga_getxdim();
    char bstr[20];

    sprintf(bstr,"%12s",bool2str(get_2nd));

    strxyb(gmx-w1-5, 1-3, " QUARTIC MAP #2.0.0",NCM1);
    if(go_hexa) strxy(gmx-w1, 20+1," (h) N=",NCM1);
    else strxy(gmx-w1, 20+1," N=",NCM1);
    strxy(gmx-w1, 30+1," N2=",NCM1);
    strxy(gmx-w1, 40+1," NL=",NCM1);
    strxy(gmx-w1, 50+1," NC=",NCM1);
    strxy(gmx-w1, 60+1," FREQ=",NCM1);
    strxy(gmx-w1, 70+1," 2nd'x=",NCM1);
    strxy(gmx-w1, 80+1," RAD=",NCM1);
    strxy(gmx-w1, 90+1," a=",NCM1);
    strxy(gmx-w1,100+1," b=",NCM1);
    strxy(gmx-w1,110+1," Eps=",NCM1);

    longxy(gmx-w2,1 + 20, N,NCM1);
    longxy(gmx-w2,1 + 30, N2,NCM1);
    longxy(gmx-w2,1 + 40, NL,NCM1);
    longxy(gmx-w2,1 + 50, NC,NCM1);
    longxy(gmx-w2,1 + 60,FREQ,NCM1);
    strxy(gmx-w2,1 + 70,bstr,NCM1);
    doublexy(gmx-w2,1 + 80, RAD,NCM1);
    doublexy(gmx-w2,1 + 90, a,NCM1);
    doublexy(gmx-w2,1 +100, b,NCM1);
    doublexy(gmx-w2,1 +110, EPS,NCM1);

    setcolor(NCM1);
    rectangle(gmx-185,0,gmx-1,340);
    line(gmx-185,20,gmx-1,20);
    if(plot_3) rectangle(0,0,gmx-195,513);
    else rectangle(0,0,NP1,NP1);

    switch(warmup){
    case random: strxy(gmx-w1,310,"WarmUp= RANDOMIC",NCM1); break;
    case pie1: strxy(gmx-w1,310,"WarmUp= PIE #1 ",NCM1); break;
    case pie2: strxy(gmx-w1,310,"WarmUp= PIE #2 ",NCM1); break;
    case pie3: strxy(gmx-w1,310,"WarmUp= PIE #3 ",NCM1); break;
    case pie4: strxy(gmx-w1,310,"WarmUp= PIE #4 ",NCM1); break;
    case gaussian: strxy(gmx-w1,310,"WarmUp= GAUSSIAN",NCM1); break;
    case ring: strxy(gmx-w1,310,"WarmUp= RING ",NCM1); break;
    case chess: strxy(gmx-w1,310,"WarmUp= CHESS ",NCM1); break;
    }
}

```

```

    case constant: strxy(gmx-w1,310,"WarmUp= CONSTANT",NCM1); break;
    case mixed: strxy(gmx-w1,310,"WarmUp= MIXED ",NCM1); break;
    case squared: strxy(gmx-w1,310,"WarmUp= SQUARED ",NCM1); break;
    case randlocked: strxy(gmx-w1,310,"WarmUp= RDLOCKED",NCM1); break;
    case chesslocked: strxy(gmx-w1,310,"WarmUp= CHLOCKED",NCM1); break;
}
switch(pal){
case gray: strxy(gmx-w1,320,"rgbPAL= GRAY.PAL ",NCM1); break;
case rainbow: strxy(gmx-w1,320,"rgbPAL= RAINBOW.PAL ",NCM1); break;
case red: strxy(gmx-w1,320,"rgbPAL= RED.PAL ",NCM1); break;
case green: strxy(gmx-w1,320,"rgbPAL= GREEN.PAL ",NCM1); break;
case blue: strxy(gmx-w1,320,"rgbPAL= BLUE.PAL ",NCM1); break;
case bluered1: strxy(gmx-w1,320,"rgbPAL= BLUERED1.PAL",NCM1); break;
case bluered2: strxy(gmx-w1,320,"rgbPAL= BLUERED2.PAL",NCM1); break;
case copper: strxy(gmx-w1,320,"rgbPAL= COPPER.PAL ",NCM1); break;
case random_: strxy(gmx-w1,320,"rgbPAL= RANDOM.PAL ",NCM1); break;
default: strxy(gmx-w1,320,"rgbPAL= GRAY ",NCM1);
}
if(inverted_pal) strxy(gmx-w1+55, 320,"*",NCM1);
else strxy(gmx-w1+55, 320," ",NCM1);

strxy(10,vga_getydim()-10,"MENU: Color Invert, meAn, Map, Spectrum linnes, Vars,
SPACE, 3D view, 3d Lines, Warm uP, save iMages, Zoom, 2nd'x, ESC",NCM1);
}

/* plota os parametros variaveis para o controle da simulacao */
void plot_vars(k,l)
long k;
int l;
{
    int lin=10, w1=170, w2=110, gmx=vga_getxdim();
    double n;

    if(main_k<FREQ){
strxy(gmx-w1, 10+l," Loop=",NCM1);
strxy(gmx-w1, 20+l,"AVGdx2=",0100);

        strxy(gmx-w1, 30+l," Mini=",NCM1);
strxy(gmx-w1, 40+l,"x1 N/2=",NCM1);
        strxy(gmx-w1, 50+l," Max1=",NCM1);
strxy(gmx-w1, 60+l," Avg1=",NCM1);
        strxy(gmx-w1, 70+l,"MinDX1=",NCM1);
strxy(gmx-w1, 80+l,"MaxDX1=",NCM1);
strxy(gmx-w1, 90+l,"TopDX1=",NCM1);
strxy(gmx-w1,100+l,"AVGdx2=",0100);

        strxy(gmx-w1,110+l," Min2=",NCM1);
strxy(gmx-w1,120+l,"x2 N/2=",NCM1);
strxy(gmx-w1,130+l," Max2=",NCM1);
strxy(gmx-w1,140+l," Avg2=",NCM1);
strxy(gmx-w1,150+l,"MinDX2=",NCM1);
strxy(gmx-w1,160+l,"MaxDX2=",NCM1);
strxy(gmx-w1,170+l,"TopDX2=",NCM1);
    }

    if((k & 1)==1) lin=30; else lin=110;

        longxy(gmx-w2,l+10,k,NCM1);
doublexy(gmx-w2,l+lin-10,mean_dx2(main_k),0100);
doublexy(gmx-w2,l+lin+00,min_x,NCM1);
doublexy(gmx-w2,l+lin+10,x[N]>>1][N]>>1,NCM1);
doublexy(gmx-w2,l+lin+20,max_x,NCM1);
doublexy(gmx-w2,l+lin+30,avg,NCM1);
doublexy(gmx-w2,l+lin+40,min_dx,NCM1);
doublexy(gmx-w2,l+lin+50,max_dx,NCM1);
doublexy(gmx-w2,l+lin+60,most_probable_dx,NCM1);

    if(main_k>TRANSIENT){
n = (double) main_k-TRANSIENT;
mean1 = 2.0*sum1/n;
mean2 = 2.0*sum2/n;
err1 = sqrt(sum12/n-mean1*mean1)/n;
err2 = sqrt(sum22/n-mean2*mean2)/n;
doublexy(gmx-w2-100,l+300,mean1,150);
doublexy(gmx-w2-100,l+310,mean2,150);
doublexy(gmx-w2+000,l+300, err1,150);
doublexy(gmx-w2+000,l+310, err2,150);
    }
}

/* limpa a tela parcialmente */
void cleartop(){
    int i,j;

    setcolor(0);
    for(j=0;j<=NMAX+2;j++) line(0,j,vga_getxdim()-190,j);
}

```



```

        if ( ((i-hN)*(i-hN)+(j-hN)*(j-hN))<=qN2 ) x[i][j]=x2;
    else x[i][j]=x1;
        break;
    case pie2:
        if ( ((i-hN)*(i-hN)+(j-hN)*(j-hN))<=qN2 ) x[i][j]=x1;
    else x[i][j]=x2;
        break;
    case pie3:
        if ( ((i-hN)*(i-hN)+(j-hN)*(j-hN))<=(qN2>>1) ) x[i][j]=x2;
    else x[i][j]=randin(-RAD,RAD);
        break;
    case pie4:
        if ( ((i-hN)*(i-hN)+(j-hN)*(j-hN))<=qN2 ) x[i][j]=randin(-RAD,RAD);
    else x[i][j]=x2;
        break;
    case ring:
        x[i][j] = randin(1.370,1.420);
        break;
    case chess:
        x[i][j] = chess_color[(i-1)*Nchess/N][(j-1)*Nchess/N];
        break;
    case chesslocked:
        x[i][j] = chess_color[(i-1)*Nchess/N][(j-1)*Nchess/N];
        break;
    case gaussian:
        gauss=0.0;
        for(k=0;k<ng;k++) gauss+=randin(-RAD,+RAD);
        x[i][j] = (double) gauss/ng;
        break;
    case constant:
        x[i][j] = warmup_constant;
        break;
    case mixed:
        if(j<hN) x[i][j]=randin(-RAD,+RAD);
        else x[i][j]=warmup_constant;
        break;
    case squared:
        if((j<qN)||(j>3*qN)||(i<qN)||(i>3*qN)) x[i][j]=randin(-RAD,+RAD);
        else x[i][j]=-warmup_constant;
        break;
    case randlocked:
        if((rand() % 2)==0) x[i][j]=x1;
        else x[i][j]=x2;
        break;
    }
    y[i][j] = x[i][j];
}
}

if (plot_3) plot3d_map(0,-RAD,+RAD); else plot_map(0,-RAD,+RAD);

main_k      =      0;          /* resets the main step counter */
sum1        =      0.0;
sum2        =      0.0;
min_avg     = +1000.0;
max_avg     = -1000.0;
min_x       = +1000.0;
max_x       = -1000.0;
min_x_global = +1000.0;
max_x_global = -1000.0;
min_dx      = +1000.0;
max_dx      = -1000.0;
max_ddx     = +0.0000;
max_f_global =      0;

plotted_spectrum=FALSE;
plotted_dx_spectrum=FALSE;
plot_const(10);
}

/* salva as imagens PS da rede e BMP da tela */
void make_pics(bmpfile_mask,psfile_mask)
char bmpfile_mask[], psfile_mask[];
{
    char fname[128];

    sprintf(fname,psfile_mask,vers_id,N,N,main_k);
    make_ps(fname,N,500,0,0);
    sprintf(fname,bmpfile_mask,vers_id,N,N,main_k);
    make_bmp(fname);
    saved++;
}

/* *****
aplica a condicao periodica de contorno sobre a rede: toroidal.
apenas a matriz y[][] e usada, pois e sobre ela que sera feita
a difusao (acoplamento entre vizinhos) para recalculer x[][].
*****
void boundary_x()          /* Torus Periodic Boundary

```

```

                                Condition (only for y) */
{
    register int k;
    for(k=1;k<=N;k++){
y[NP1][k] = y[1][k];
    y[0][k] = y[N][k];
y[k][NP1] = y[k][1];
y[k][0] = y[k][N];
    }
    if(get_2nd){
y[0][0] = y[N][N]; /* full range 0..NP1 to fix */
y[NP1][NP1] = y[1][1]; /* 2nd neighbor corners bug */
y[0][NP1] = y[N][1];
y[NP1][0] = y[1][N];
    }
}

/* controla o display dos espectros na tela */
void plot_spectra()
{
    plot_dx_spectrum(vga_getxdim()-258,630-63,630,&plotted_dx_spectrum);
    plot_spectrum(vga_getxdim()-258,730-63,730,&plotted_spectrum);
    plot_sample(1,645,256,745,&plotted_sample1);
}

/*
-----
Essa rotina verifica a cada loop se alguma tecla foi acionada pelo usuario
o que permite a interatividade do programa. Quase todos os parametros e
variaveis principais do programa pode ser alteradas durante a execucao,
o que as vezes torna necessaria a reinicializacao da rede, o que e feito
automaticamente, quando necessario.

As teclas ativas e seus efeitos sobre o programa sao os seguintes:
Teclas Funcao
-----
0      reseta os valores de a e b para o ponto usual (a0,b0)
2      inverte o status do acoplamento: liga-desliga os segundos vizinhos
6      inverte o status da geometria: liga-desliga a rede hexagonal
      OBS: quando e ligada a rede hexagonal, automaticamente desliga-se
      o acoplamento de segundos vizinhos. Opcao nao implementada!
X,x    dobra a frequencia de atualizacao de tela (FREQ) ciclicamente
E,e    decrementa a constante de acoplamento epsilon em delta_eps
R,r    aumenta a constante de acoplamento epsilon em delta_eps
F,f    decrementa o parametro "a" em delta_a
G,g    aumenta o parametro "a" em delta_a
H,h    decrementa o parametro "b" em delta_a
J,j    aumenta o parametro "b" em delta_a
H,h    decrementa a constante de acoplamento epsilon em delta_eps
J,j    aumenta a constante de acoplamento epsilon em delta_eps
SPC    inverte a fase visualizada (par/impar)
-/+    diminui/aumenta a escala "z", no modo de vizualizacao 3D apenas
O,o    salva as figuras PS e BMP da rede e tela
M,m    inverte o status do plot da rede
K,k    inverte o status do uso dos extremos globais (min,max)
I,i    inverte o status da paleta: inverte a paleta (negativo da imagem)
Z,z    inverte o status do zoom: liga-desliga a atualizacao do zoom (plot)
V,v    inverte o status das variaveis: liga-desliga a atualizacao
S,s    inverte o status dos espectros: liga-desliga a atualizacao (plot)
A,a    inverte o status das medias: liga-desliga a atualizacao das medias
D,d    inverte o tipo de plot 2D-3D
L,l    no plot 3D, inverte o plot com pontos-linhas cheias
C,c    muda ciclicamente a paleta de cores usada, mantendo o status de
      inversao da paleta
N,n    plota os espectros com linhas verticais (mais visiveis se TRUE)
W,w    reinicializa a rede, e zera o contador de loops
P,p    seleciona outro modo de inicializacao da rede, ciclicamente
>      dobra o tamanho da rede, sem reinicializa-la!
<      divide a rede ao meio, sem reinicializa-la!
-----
Importante: - a tecla *NumLock* deve ser mantida sempre DESLIGADA!
            - teclas ainda nao usadas: "BTQUY"
*/
void switch_key(ch)
char ch;
{
    int i,j,k;
switch (ch){
    case 'X':
    case 'x':
        if(FREQ<=256) FREQ<<=1;
        else FREQ=2;
        plot_const(10);
        break;
    case '0':

```

```

a=0.35;
b=1.35;
plot_const(10);
break;
case 'E':
case 'e':
EPS-=0.01;
plot_const(10);
break;
case 'R':
case 'r':
EPS+=0.01;
plot_const(10);
break;
case 'F':
case 'f':
a-=delta_a;
plot_const(10);
break;
case 'G':
case 'g':
a+=delta_a;
plot_const(10);
break;
case 'H':
case 'h':
b-=delta_b;
plot_const(10);
break;
case 'J':
case 'j':
b+=delta_b;
plot_const(10);
break;
case ',':
shift=(shift+1) % 2;
plot_spectra();
break;
case '-':
if (plot_3){
if (z_deep<7){
z_deep++;
clear_top();
}
}
break;
case '+':
if (plot_3){
if (z_deep>1){
z_deep--;
clear_top();
}
}
break;
case '0':
case 'o':
make_pics(bmpfile_mask,psfile_mask);
break;
case 'M':
case 'm':
plot_m=!plot_m;
break;
case 'K':
case 'k':
lock_mm=!lock_mm;
break;
case 'I':
case 'i':
invertpal();
inverted_pal=!inverted_pal;
plot_const(10);
break;
case '2':
get_2nd=!get_2nd;
go_hexa=FALSE;
if(auto_eps){
if(get_2nd) S=8; else S=4;
EPS=(double) S/(S+1.0);
}
plot_const(10);
break;
case '6':
go_hexa=TRUE;
if(auto_eps){
S=3;
EPS=(double) S/(S+1.0);
}
plot_const(10);
break;
case 'Z':
case 'z':
plot_zm=!plot_zm;
break;
case 'V':

```



```

    case 'v':
    plot_v=!plot_v;
    break;
    case 'S':
    case 's':
    plot_s=!plot_s;
    break;
    case 'A':
    case 'a':
    plot_a=!plot_a;
    break;
    case 'D':
    case 'd':
    plot_3=!plot_3;
    cleartop();
break;
    case 'L':
    case 'l':
    if(plot_3){
    plot_p=!plot_p;
    cleartop();
    }
break;
    case 'C':
    case 'c':
    if(pal<random_) pal++; else pal=gray;
    switch(pal){
    case gray: loadpal("rgb/gray.pal"); break;
    case rainbow: loadpal("rgb/rainbow.pal"); break;
    case red: loadpal("rgb/red.pal"); break;
    case green: loadpal("rgb/green.pal"); break;
    case blue: loadpal("rgb/blue.pal"); break;
    case bluered1: loadpal("rgb/bluered1.pal"); break;
    case bluered2: loadpal("rgb/bluered2.pal"); break;
    case copper: loadpal("rgb/copper.pal"); break;
    case random: loadpal("rgb/random.pal"); break;
    default: loadpal("rgb/gray.pal");
    }
    if(inverted_pal) invertpal();
    plot_const(10);
break;
    case 'N':
    case 'n':
    plot_n=!plot_n;
break;
    case 'W':
    case 'w':
    init_x();
    plotted_avg=FALSE;
break;
    case '>>': /* make it grow up */
    if(N<NMAX){
    for(j=1;j<=N;j++){
    for(i=1;i<=N;i++){
    x[i+N][j]=x[N-i][j];
    x[i][j+N]=x[i][N-j];
    x[i+N][j+N]=x[N-i][N-j];
    y[i+N][j]=x[i+N][j];
    y[i][j+N]=x[i][j+N];
    y[i+N][j+N]=x[i+N][j+N];
    }
    }
    k=N+(N>>1);
    for(j=1;j<=2*N;j++){
    x[j][k]=randin(-RAD,+RAD);
    x[k][j]=randin(-RAD,+RAD);
    }
    N<<=1;
    N2=N*N;
    NP1=N+1;
    NH=N>>1;
    NM1=N-1;
    boundary_x();
    cleartop();
    plot_const(10);
    mouse_hide();
    mouse_goto(NP1,NP1);
    mouse_move();
    mouse_show();
    }
break;
    case '<': /* use a smaller one */
    if(N>NMIN){
    N>>=1;
    N2=N*N;
    NP1=N+1;
    NM1=N-1;
    NH=N>>1;
    setcolor(0);
    for(j=0;j<=N;j++) line(NP1,j,N<<1+1,j);

```

```

        for(j=NP1;j<=(N<<1)+1;j++) line(0,j,N<<1+1,j);
        boundary_x();
        cleartop();
        plot_const(10);
        mouse_hide();
        mouse_goto(NP1,NP1);
        mouse_move();
        mouse_show();
    }
break;
    case 'p':          /* select the next init method */
    case 'p':
        if(warmup<squared) warmup++; else warmup=randomic;
        init_x();
break;
}
}

/*****
essa funcao faz a evolucao temporal, de todos os mapas da rede. cada mapa
e recalculado a partir de seu valor anterior, e os parametros "a"e "b"
*****/
void step_x() /* != Quartic Map Step ... */
{
    register int i,j;
    double tmp, infty=1.0e4;

    for(j=1;j<=N;j++){
        if(j % MOUSE_DELAY) mouse_move();
        for(i=1;i<=N;i++){
            tmp=x[i][j];
            tmp=tmp*tmp-a;
            y[i][j]=tmp*tmp-b;
            if(fabs(y[i][j])>infty){
                if(fake_ovrflw) init_x();
                else {
                    vga_setmode(TEXT);
                    printf("\nOVERFLOW: a=%5.3f b=%5.3f 2nd=%s rad=%5.3f\n\n",a,
                    b,bool2str(get_2nd),RAD,i,j,x[i][j]);
                    exit(1);
                }
            }
        }
    }
}

/*****
essa funcao faz o acoplamento local difusivo dos mapas da rede, promedian-
do os valores de x[][] com seus vizinhos locais.
*****/
void blur_x(k) /* != */
long k;
{
    register int i,j,idx;
    double sx,sum=0.0,xij,dx;
    int max_j;
    long max_f=0;

    min_x = +10000.0; max_x = -10000.0;
    min_dx = +10000.0; max_dx = -10000.0;

    for(i=0;i<NC;i++) fdx[i]=0;

    /* EPS democratic = S/(S+1), where S = # of neighbors */
    if(go_hexa) S=3; else if(get_2nd) S=8; else S=4;

    c1=1.0-EPS;
    c2=EPS/(double) S;
    c3=1.0/((double) S+1.0);

    for(j=1;j<=N;j++){
        if(j % MOUSE_DELAY) mouse_move();
        for(i=1;i<=N;i++){
            xij=y[i][j];
            if(go_hexa){
                if(odd(N+i-j)) sx=y[i-1][j]+y[i+1][j]+y[i][j+1];
                else sx=y[i-1][j]+y[i+1][j]+y[i][j-1];
            }
            else sx=y[i-1][j]+y[i+1][j]+y[i][j-1]+y[i][j+1];
            if(get_2nd) sx+=y[i-1][j-1]+y[i+1][j-1]+y[i-1][j+1]+y[i+1][j+1];
            if(auto_eps) x[i][j]=c3*(y[i][j]+sx); else x[i][j]=c1*y[i][j]+c2*sx;
            dx=x[i][j]-xij;
            idx=NC*(dx+0.15)/0.3;
            fdx[idx]++;
            sum=sum+x[i][j];
            if(x[i][j]<min_x) min_x=x[i][j];
            if(x[i][j]>max_x) max_x=x[i][j];
            if(dx<min_dx) min_dx=dx;

```

```

if(dx>max_dx) max_dx=dx;
}
}
max_ddx=max_dx-min_dx;
if(main_k>NL){
if(min_x<min_x_global) min_x_global=min_x;
if(max_x>max_x_global) max_x_global=max_x;
}

avg = (double) sum/N2;
if (k<=NL) {
if (avg<min_avg) min_avg=avg;
if (avg>max_avg) max_avg=avg;
}

for(j=0;j<NC;j++){
if(fdx[j]>max_f){
max_f=fdx[j];
max_j=j;
}
}

most_probable_dx=(double) (0.3*max_j/NC)-0.15;
}

/* plota todos os graficos e janelas escolhidos */
void plot_all(k)
long k;
{
if((k % FREQ)==shift){
plot_site_info();
if (plot_m){
if (plot_3) plot3d_map(k,min_x,max_x);
else plot_map(k,min_x,max_x);
}
if (plot_s) plot_spectra();
if(plot_zm) plot_zoom(512-225+1,745-225+1,512,745,25,25);
}
if ((k>NL) && (((k+1) % FREQ)==shift) && plot_s)
plot_sample(1,535,256,635,&plotted_sample2);
if ((k>NL) && (plot_a)) plot_avg(530+40,737-73+1,750,737,&plotted_avg);
if (plot_v) plot_vars(k,120);
}

/* se ESC for precionado, confirma o termino do programa */
int confirm_exit(x1,y1,x2,y2,c1,c2,msg)
int x1,y1,x2,y2,c1,c2;
char msg[];
{
char ch;
int confirmed=FALSE;

rectfill(x1,y1,x2,y2,c2);
setcolor(c1);
rectangle(x1,y1,x2,y2);
strxyb((x1+x2-8*strlen(msg))/2,(y2+y1)/2-4,msg,c1);
putchar('\007');
while(!confirmed){
ch=vga_getkey();
switch(ch){
case 'Y':
case 'y':
rectfill(x1,y1,x2,y2,c2);
return (TRUE);
break;
case 'N':
case 'n':
rectfill(x1,y1,x2,y2,c2);
return (FALSE);
break;
}
mouse_update();
mouse_move();
confirmed=mouse_right();
}
rectfill(vga_getxdim()-185,350,vga_getxdim()-1,369,0);
}

/*****
essa parte junta todos os passos necessarios da simulacao da rede
EPS e variado e a rede e simulada muitas vezes, a fim de se obter
amostras das quantidades relevantes, para valore de epsilin de 0 a 1
*****/
void run_map()
{
char ch=0xFF, image[128];
int done=FALSE;
char datfile[128];
FILE *sf;

```

```

sprintf(datfile,"%s/gxe-N=%d.dat",dirdata,N);
if((sf=fopen(datfile,"ab"))==NULL) return;

fprintf(sf,"# N      = %8d\n",N);
fprintf(sf,"# LIXO   = %8d\n",TRANSIENT);
fprintf(sf,"# CALC    = %8d\n",SAMPLE);
fprintf(sf,"# RAD     = %8.5f\n",RAD);
fprintf(sf,"# a      = %8.5f\n",a);
fprintf(sf,"# b      = %8.5f\n",b);
if(get_2nd) fprintf(sf,"# 2nd    = TRUE\n");
else{
    fprintf(sf,"# 2nd    = FALSE\n");
    if(go_hexa) fprintf(sf,"# Rede HEXAGONAL\n");
}

loadpal("rgb/rainbow.pal");

main_k=0;
EPS=0.0;

/* plota tudo */
plot_s=TRUE;
plot_a=TRUE;
plot_zm=TRUE;
plot_m=TRUE;
z_deep=5;
lock_mm=TRUE;

while(EPS<=1.1){
    init_x();
    start=time(NULL);
    done=false;
    while(!done){
main_k++;
total_k++;
switch_key(ch);
        step_x();
        boundary_x();
        blur_x(main_k);
plot_all(main_k);
done=(((ch=vga_getkey()) == ESC) || mouse_right() ||
(main_k==(TRANSIENT+SAMPLE)));
if(main_k>=(TRANSIENT+SAMPLE-1)){
    {/*****/}
    plot_site_info();
    if (plot_m){
if (plot_3) plot3d_map(main_k,min_x,max_x);
        else plot_map(main_k,min_x,max_x);
        save_spectrum();
    }
    if (plot_s) plot_spectra();
    if (plot_zm) plot_zoom(512-225+1,745-225+1,512,745,25,25);
    if ((main_k>NL) && ((main_k+1) % FREQ)==shift) && plot_s
        plot_sample(1,535,256,635,&plotted_sample2);
        if ((main_k>NL) && (plot_a))
            plot_avg(530+40,737-73+1,750,737,&plotted_avg);
    if (plot_v) plot_vars(main_k,120);
    {/*****/}

    sprintf(image,"%s/qmcm1-2d-e=%5.3f-N=%d-i=%d.ps",dirdata,EPS,N,main_k);
        make_ps(image,N,500,0,0);

    plot_3=false;
    cleartop();
        plot_map(main_k,min_x,max_x);
        sprintf(image,"%s/qmcm1-2d-e=%5.3f-N=%d-i=%d.bmp",dirdata,EPS,N,main_k);
        make_bmp(image);

    plot_3=true;
    cleartop();
        plot3d_map(main_k,min_x,max_x);
        sprintf(image,"%s/qmcm1-3d-e=%5.3f-N=%d-i=%d.bmp",dirdata,EPS,N,main_k);
        make_bmp(image);

    plot_3=false;
    cleartop();
}
}
fprintf(sf,"%12.8f %12.8f %12.8f %12.8f %12.8f\n",EPS, mean1,err1,mean2,err2);
EPS+=0.1;
}

fclose(sf);
mouse_free();
stop=time(NULL);
}

```

```

/* Programa Principal */
main(argc, argv)
int argc;
char *argv[];
{
    unsigned int mousetype;
    randomize();

#ifdef debug
    get_2nd=false;
    RAD=1.4;
    auto_eps=false;
    mousetype=0;
    warmup=randomic;
    lock_mm=FALSE;
    plot_n=TRUE;
    plot_p=FALSE;
#else
    printf("\nQUARTIC MAP 2D Coupled Map Lattices (CML) ver. #s (%s)\n\n",vers_id,date_id);
    printf("    Entre o raio RAD in [0.00,1.45] : "); scanf("%lf",&RAD);
    printf("    EPS Auto democratico EPS (0/1) : "); scanf("%d",&auto_eps);
    auto_eps &= 1;
    if(!auto_eps)
    {
        printf("    Entre acoplam. EPS em [0.0,1.0] : "); scanf("%lf",&EPS);
    }
    printf("    MOUSE 0-PS/2 1-Micro$oft : "); scanf("%d",&mousetype);
    printf("\n\n ");
    mousetype=mousetype % 2;
#endif

    if(vga_init()==0){
        if(argc==2){
            if(strcmp(argv[1],"-1024x768")==0) vga_setmode(G1024x768x256);
            else if(strcmp(argv[1],"-800x600")==0) vga_setmode(G800x600x256);
            else if(strcmp(argv[1],"-640x480")==0) vga_setmode(G640x480x256);
            else if(strcmp(argv[1],"-320x200")==0) vga_setmode(G320x200x256);
            else{
                vga_setmode(TEXT);
                printf(" ***** Modo grafico invalido ou nao suportado ***** ");
                exit(1);
            }
        }
        else vga_setmode(G1024x768x256);
        if(mousetype) mouse_open(MOUSE_MICROSOFT); else mouse_open(MOUSE_PS2);
        mouse_setscale(2);
        mouse_hide();
        mouse_setcolor(NCM1);
        mouse_goto(N,N);
        mouse_move();
        mouse_show();
        run_map();
        vga_setmode(TEXT);
    }
    else printf("SVGA Erro ao iniciar o modo grafico.\n");

    printf("\n\n >>> Velocidade media = %f loops/s\n",total_k/(stop-start+0.001));
    printf("\n >>> Gravadas %d imagens %dx%dx256C (BMP & Postscript)\n\n",2*saved,
        vga_getxdim(),vga_getydim());
    printf("*****\n");
    printf(" * QUARTIC MAP 2D Coupled Map Lattices (CML)          ver. #s %s * \n",
        vers_id,date_id);
    printf(" * (C) 1999-2000 by L. C. Martins (fonte C) & Leonardo G. Brunnet, Orient. * \n");
    printf(" * UDESC/CCT/FEJ/DFIS, Joinville-SC, Brazil e-mail: lcm@dcc.fej.udesc.br * \n");
    printf(" * Instituto de Fisica-UFRGS, P. Alegre-RS, Brazil e-mail: lcm@if.ufrgs.br * \n");
    printf("*****\n\n");
    printf("mindx=%f maxdx=%f maxddx=%f mprobdx=%f\n",min_dx,max_dx,max_ddx,most_probable_dx);
    printf("max_f=%d max_f_global=%d\n\n",max_f,max_f_global);
    exit(0);
}

```

D - Listagem da Biblioteca ufrgs-2.0.0.c

```

/* -----
ufrgs #2.0.0 - funcoes matematicas e de uso geral
(C) 1999-2000 by L. C. Martins e-mail: lcm@if.ufrgs.br
Instituto de Fisica, UFRGS, Porto Alegre-RS, Brazil (lcm@if.ufrgs.br)
Depto. de Fisica, UDESC/FEJ, Joinville-SC, Brazil (dfi2lcm@dcc.fej.udesc.br)
-----*/
#define GNU
#ifdef _UFRGS_
/* UFRGS Package signature ----- */

```

```

#define _UFRGS_
#endif

#ifdef GNU          /* BEGIN Specific GNU C/Linux Libraries Support ----- */
#ifdef VGAGL_H
#include <vgagl.h>
#endif

#ifdef VGA_H
#include <vga.h>
#endif
/* END ----- */

#ifdef FLOAT_H     /* BEGIN COMMON ANSI C Libraries Support ----- */
#include <float.h>
#endif

#ifdef MATH_H
#include <math.h>
#endif

#ifdef STDLIB_H
#include <stdlib.h>
#endif

#ifdef TIME_H
#include <time.h>
#endif

#ifdef STDIO_H
#include <stdio.h>
/* END ----- */

#define TRUE      1
#define FALSE     0
#define true      1
#define false     0
#define maxplotwin 16
#define strsiz    16
#define bigticksize 8
#define ticksize  4
#define yUP       false
#define xRIGHT    true
#define yDOWN     true
#define xLEFT     false
#define gridon    true
#define gridoff   false
#define pi        3.1415926

struct
tplotwin {
    int    x1, y1, x2, y2, xold, yold ;
    double xmin, ymin, xmax, ymax;
    int    active, color, border, xaxis, yaxis;
    double xscale, yscale; /* pixels per unit length */
} pw[maxplotwin];

struct
tplotaxis{int blocks, digits, decimals, tickness, axiscolor, ok, gridlines;
char title[strsiz], axislabel[strsiz], units[strsiz];
} pa[maxplotwin][2];

#ifdef GNU          /* BEGIN Specific GNU C/Linux functions ----- */
int default_color=15, default_bkcolor=0;

int getmaxx(void)
{
return vga_getxdim()-1;
}

int getmaxy(void)
{
return vga_getydim()-1;
}

void putpixel(x,y,color)
int x,y,color;
{
vga_setcolor(color);
vga_drawpixel(x,y);
}

void setcolor(color)
int color;
{
vga_setcolor(color);
default_color=color;
}

void setbkcolor(color)
int color;
{
default_bkcolor=color;
}

```

```

void opengraph(vgamode)
int vgamode;
{
    vga_setmode(vgamode);
}

void line(x1,y1,x2,y2)
int x1,y1,x2,y2;
{
    vga_drawline(x1,y1,x2,y2);
}

void rectangle(x1,y1,x2,y2)
int x1,y1,x2,y2;
{
    vga_drawline(x1,y1,x2,y1);
    vga_drawline(x2,y1,x2,y2);
    vga_drawline(x2,y2,x1,y2);
    vga_drawline(x1,y2,x1,y1);
}

void circle(xc,yc,r)
int xc,yc,r;
{
    double x, y, x_, y_, theta;
    int k;

    x_=xc;
    y_=yc+r;
    for(k=1;k<=72;k++){
        theta=k*pi/36.0;
        x = xc + r * sin(theta);
        y = yc + r * cos(theta);
        vga_drawline(x_,y_,x,y);
        x_ = x;
        y_ = y;
    }
}

void ellipse(xc, yc, rx, ry)
int xc,yc,rx,ry;
{
    double x, y, x_, y_, theta;
    int k;

    x_=xc;
    y_=yc+ry;
    for(k=1;k<=72;k++){
        theta=k*pi/36.0;
        x = xc + rx * sin(theta);
        y = yc + ry * cos(theta);
        vga_drawline(x_,y_,x,y);
        x_ = x;
        y_ = y;
    }
}

void savepal(fname)
char fname[];
{
    struct recpal {unsigned char r,g,b;} pal[256];
    int i, red, green, blue;
    FILE *f;

    for(i=0;i<256;i++){
        vga_getpalette(i,&red,&green,&blue);
        pal[i].r=(unsigned char) red;
        pal[i].g=(unsigned char) green;
        pal[i].b=(unsigned char) blue;
    }

    if((f=fopen(fname,"wb"))==NULL) return;
    else{
        fwrite(pal, sizeof(pal),1,f);
        fclose(f);
    }
}

void loadpal(fname)
char fname[];
{
    struct recpal {unsigned char r,g,b;} pal[256];
    int i, red, green, blue;
    FILE *f;

    if((f=fopen(fname,"rb"))==NULL) return;
    else{
        fread(pal,sizeof(pal),1,f);
        fclose(f);
    }
}

```

```

        for(i=0;i<256;i++){
            red = (int) pal[i].r;
            green = (int) pal[i].g;
            blue = (int) pal[i].b;
            vga_setpalette(i,red,green,blue);
        }
    }
}

void invertpal(void)
{
    int i, red, green, blue;
    for(i=0;i<256;i++){
        vga_getpalette(i,&red,&green,&blue);
        red = 63-red;
        green = 63-green;
        blue = 63-blue;
        vga_setpalette(i,red,green,blue);
    }
}

void make_bmp(fname)
char fname[];
{
    unsigned char
        h200[32]={0x42,0x4D,0x36,0xFE,0x00,0x00,0x00,0x00,
                0x00,0x00,0x36,0x04,0x00,0x00,0x28,0x00,
                0x00,0x00,0x40,0x01,0x00,0x00,0xC8,0x00,
                0x00,0x00,0x01,0x00,0x08,0x00,0x00,0x00},
        h400[32]={0x42,0x4D,0x36,0xEC,0x03,0x00,0x00,0x00,
                0x00,0x00,0x36,0x04,0x00,0x00,0x28,0x00,
                0x00,0x00,0x80,0x02,0x00,0x00,0x90,0x01,
                0x00,0x00,0x01,0x00,0x08,0x00,0x00,0x00},
        h480[32]={0x42,0x4D,0x36,0xB4,0x04,0x00,0x00,0x00,
                0x00,0x00,0x36,0x04,0x00,0x00,0x28,0x00,
                0x00,0x00,0x80,0x02,0x00,0x00,0xE0,0x01,
                0x00,0x00,0x01,0x00,0x08,0x00,0x00,0x00},
        h600[32]={0x42,0x4D,0x36,0x57,0x07,0x00,0x00,0x00,
                0x00,0x00,0x36,0x04,0x00,0x00,0x28,0x00,
                0x00,0x00,0x20,0x03,0x00,0x00,0x58,0x02,
                0x00,0x00,0x01,0x00,0x08,0x00,0x00,0x00},
        h768[32]={0x42,0x4D,0x36,0x04,0x0C,0x00,0x00,0x00,
                0x00,0x00,0x36,0x04,0x00,0x00,0x28,0x00,
                0x00,0x00,0x00,0x04,0x00,0x00,0x00,0x03,
                0x00,0x00,0x01,0x00,0x08,0x00,0x00,0x00},

        b[1024],c;
    struct recpal {unsigned char b,g,r,c;} pal[256];
    int i, j, red, green, blue;
    FILE *f;

    for(i=0;i<256;i++){
        vga_getpalette(i,&red,&green,&blue);
        pal[i].r=(unsigned char) red<<2;
        pal[i].g=(unsigned char) green<<2;
        pal[i].b=(unsigned char) blue<<2;
        pal[i].c=0x00;
    }

    if((f=fopen(fname,"wb"))==NULL) return;
    else
    {
        switch(vga_getydim()){
            case 200: fwrite(h200, sizeof(h200),1,f); break;
            case 400: fwrite(h400, sizeof(h200),1,f); break;
            case 480: fwrite(h480, sizeof(h200),1,f); break;
            case 600: fwrite(h600, sizeof(h200),1,f); break;
            case 768: fwrite(h768, sizeof(h200),1,f); break;
        }

        for(i=0;i<=21;i++) b[i]=0x00; fwrite(b,22,1,f);

        fwrite(pal,sizeof(pal),1,f);

        for(j=vga_getydim()-1;j>=0;j--){
            for(i=0;i<vga_getxdim();i++) b[i]=(unsigned char) vga_getpixel(i,j);
            fwrite(b,sizeof(b),1,f);
        }
        fclose(f);
    }
}

void saveBMP(fname) /* Synonymous of make_bmp() */
char fname[];
{
    make_bmp(fname);
}

```



```

#endif          /* END Specific GNU C/Linux functions ----- */
/* BEGIN COMMON ANSI C functions - Part I: Utils ----- */

int odd(any)
int any;
{
  if ((any & 0x01)==1) return TRUE; else return FALSE;
}

int min(a,b)
int a,b;
{
  if(a<=b) return a; else return b;
}

int max(a,b)
int a,b;
{
  if(a>=b) return a; else return b;
}

char *bool2str(flag)
int flag;
{
  if(flag==FALSE) return (char *) "FALSE";
  if(flag==TRUE) return (char *) "TRUE"; else return (char *) "UNDEF";
}

int quantize(x,xmin,xmax,x1,x2,usual)
double x,xmin,xmax;
int x1,x2,usual;
{
  if (!usual) return (int) (x1 + (x2 - x1) * (xmax - x) / (xmax - xmin));
  else return (int) (x1 + (x2 - x1) * (x - xmin) / (xmax - xmin));
}

double dequantize(x,x1,x2,xmin,xmax,usual)
int x,x1,x2;
double xmin,xmax;
int usual;
{
  if (!usual) return (double) xmin + (xmax - xmin) * (x2 - x) / (x2 - x1);
  else return (double) xmin + (xmax - xmin) * (x - x1) / (x2 - x1);
}

char quantizeh(x,xmin,xmax,usual)
double x,xmin,xmax;
int usual;
{
  char hex_digit[16]="0123456789ABCDEF";
  return (char) hex_digit[quantize(x,xmin,xmax,0,15,usual)];
}

int requant(x,xmin,xmax,x1,x2)
int x,xmin,xmax,x1,x2;
{
  return (int) (x1 + 1.0*(x2 - x1) * (x - xmin) / (xmax - xmin));
}

int quant(x,xmin,xmax,x1,x2)
double x,xmin,xmax;
int x1,x2;
{
  return (int) (x1 + (x2 - x1) * (x - xmin) / (xmax - xmin));
}

double dequant(x,x1,x2,xmin,xmax)
int x,x1,x2;
double xmin,xmax;
{
  return (double) xmin + (xmax - xmin) * (x - x1) / (x2 - x1);
}

void delay(n)
long n;
{
  long stop=time(NULL)+n;
  while(time(NULL)<=stop){ /* nop */;}
}

void randomize(void)
{
  srand((unsigned int) time(NULL));
}

double rand01()
{
  return (double) rand()/RAND_MAX;
}

double randin(a,b)

```

```

double a,b;
{
  return (double) a+(b-a)*rand01();
}

int inrange(x,x1,x2)
int x, x1, x2;
{
  return ((x>=x1) && (x<=x2));
}

int finrange(x,x1,x2)
double x, x1, x2;
{
  return ((x>=x1) && (x<=x2));
}

int inwin(x,y,x1,y1,x2,y2)
int x,y,x1,y1,x2,y2;
{
  return inrange(x, x1, x2) && inrange(y, y1, y2);
}

int finwin(x,y,x1,y1,x2,y2)
double x,y,x1,y1,x2,y2;
{
  return finrange(x, x1, x2) && finrange(y, y1, y2);
}

void rectfill(x1,y1,x2,y2,color)
int x1,y1,x2,y2,color;
{
  int j;
  setcolor(color); for(j=y1;j<=y2;j++) line(x1,j,x2,j);
}

double quartic_infinity = 1.0e5, quartic_a = 0.35, quartic_b = 1.35,
      quartic_xn, quartic_xnm1, quartic_x0, quartic_lyap;
int quartic_ovrflw;
long quartic_n;

double quartic_init(a,b,x0,n)
double a,b,x0;
long n;
{
  long k;
  double tmp,sum=0.0;

  quartic_xn = x0;
  quartic_a = a;
  quartic_b = b;
  quartic_n = n;

  for(k=1;k<=n;k++){
    quartic_xnm1=quartic_xn;
    tmp=quartic_xn*quartic_xn-quartic_a;
    quartic_xn=tmp*tmp-b;
    if(fabs(quartic_xn)>quartic_infinity){
      quartic_ovrflw=TRUE;
      return (pi);
      break;
    }
    if(abs(quartic_xn)>1.0e-8)
      sum+=log(fabs(quartic_xn*(quartic_xn*quartic_xn-quartic_a)));
  }
  quartic_ovrflw=FALSE;
  quartic_lyap=log(4.0)+sum/n;
  return quartic_xn;
}

double quartic_lyapunov(a,b,x0,n,ntrash)
double a,b,x0;
long n,ntrash;
{
  register long k;
  double tmp,xn=x0,sum=0.0;
  long ntot=n+ntrash;

  for(k=1;k<=ntot;k++){
    tmp=xn*xn-a;
    xn=tmp*tmp-b;
    if(fabs(xn)>quartic_infinity){
      quartic_ovrflw=TRUE;
      return (pi);
      break;
    }
    if((k>ntrash) && (fabs(xn)>1.0e-8)) sum+=log(fabs(4.0*xn*(xn*xn-a)));
  }
  return sum / (double) n;
}

```

```

double quartic_x(void)
{
    double tmp;
    quartic_n++;
    quartic_xnml=quartic_xn;
    tmp=quartic_xn*quartic_xn-quartic_a;
    quartic_xn=tmp*tmp-quartic_b;
    if(fabs(quartic_xn)>quartic_infinity){
        quartic_ovrflw=TRUE;
        return (pi);
    }
    else{
        quartic_ovrflw=FALSE;
        return quartic_xn;
    }
}

typedef enum {horiz,vert} tdir;

int default_masktype = 0;
tdir default_textdir = horiz;

unsigned short font[128][8] = {0,0,0,0,0,0,0,0,126,129,165,129,189,153,129,126,126,255,219,255,195,231, 255,
126,108,254,254,254,124,56,16,0,16,56,124,254,124,56,16,0,56,124,56,254,254,124,56,124,16,16, 56, 124, 254,
124,56,124,0,0,24,60,60,24,0,0,255,255,231,195,195,231,255,255,0,60,102,66,66,102,60,0,255,195,153,189,189,
153,195,255,15,7,15,125,204,204,204,120,60,102,102,102,60,24,126,24,63,51,63,48,48,112,240,224,127,99, 127,
99,99,103,230,192,153,90,60,231,231,60,90,153,128,224,248,254,248,224,128,0,2,14,62,254,62,14,2,0,24,60,
126,24,24,126,60,24,102,102,102,102,102,0,102,0,127,219,219,123,27,27,27,0,62,99,56,108,108,56,204,120,0,
0,0,0,126,126,126,0,24,60,126,24,126,60,24,255,24,60,126,24,24,24,24,0,24,24,24,24,126,60,24,0,0,24,12,254,
12,24,0,0,0,48,96,254,96,48,0,0,0,192,192,192,254,0,0,0,36,102,255,102,36,0,0,0,24,60,126,255,255,0,0,0,
255,255,126,60,24,0,0,0,0,0,0,0,48,120,120,120,48,0,48,0,108,108,108,0,0,0,0,0,108,108,254,108,254,
108,108,0,48,124,192,120,12,248,48,0,0,198,204,24,48,102,198,0,56,108,56,118,220,204,118,0,96,96,192,0,0,0,
0,0,24,48,96,96,96,48,24,0,96,48,24,24,24,48,96,0,0,102,60,255,60,102,0,0,0,48,48,252,48,48,0,0,0,0,0,0,
48,48,96,0,0,0,252,0,0,0,0,0,0,0,0,48,48,0,6,12,24,48,96,192,128,0,124,198,206,222,246,230,124,0,48,112,
48,48,48,48,252,0,120,204,12,56,96,204,252,0,120,204,12,56,12,204,120,0,28,60,108,204,254,12,30,0,252,192,
248,12,12,204,120,0,56,96,192,248,204,204,120,0,252,204,12,24,48,48,48,0,120,204,204,120,204,204,120,0,120,
204,204,124,12,24,112,0,0,48,48,0,0,48,48,0,0,48,48,0,0,48,48,96,24,48,96,192,96,48,24,0,0,0,252,0,0,252,0,
0,96,48,124,12,24,48,96,0,120,204,12,24,48,0,48,0,124,198,222,222,222,192,120,0,48,120,204,204,252,204,204,
0,252,102,102,124,102,102,252,0,60,102,192,192,192,102,60,0,248,108,102,102,102,108,248,0,126,96,96,120,96,
96,126,0,126,96,96,120,96,96,96,0,60,102,192,192,206,102,62,0,204,204,204,252,204,204,0,120,48,48,48,48,
48,48,120,0,30,12,12,12,204,204,120,0,230,102,108,120,108,102,230,0,96,96,96,96,96,96,126,0,198,238,254,
254,214,198,198,0,198,230,246,222,206,198,198,0,56,108,198,198,198,108,56,0,252,102,102,124,96,96,240,0,0,
120,204,204,204,220,120,28,0,252,102,102,124,108,102,230,0,120,204,224,112,28,204,120,0,252,48,48,48,48,48,
48,0,204,204,204,204,204,204,252,0,204,204,204,204,204,120,48,0,198,198,198,214,254,238,198,0,198,198,108,
56,56,108,198,0,204,204,204,120,48,48,120,0,254,6,12,24,48,96,254,0,120,96,96,96,96,96,120,0,192,96,48,24,
12,6,2,0,120,24,24,24,24,120,0,16,56,108,198,0,0,0,0,0,0,0,0,0,0,255,48,48,24,0,0,0,0,0,0,120,12,
124,204,118,0,224,96,96,124,102,102,220,0,0,0,120,204,192,204,120,0,28,12,12,124,204,204,118,0,0,0,120,204,
252,192,120,0,56,108,96,240,96,96,240,0,0,0,118,204,204,124,12,248,224,96,108,118,102,102,230,0,48,0,112,
48,48,48,120,0,12,0,12,12,12,204,204,120,224,96,102,108,120,108,230,0,112,48,48,48,48,120,0,0,0,204,254,
254,214,198,0,0,0,248,204,204,204,204,0,0,0,120,204,204,204,120,0,0,0,220,102,102,124,96,240,0,0,118,204,
204,124,12,30,0,0,220,118,102,96,240,0,0,0,124,192,120,12,248,0,16,48,124,48,48,52,24,0,0,0,204,204,204,
204,118,0,0,0,204,204,120,48,0,0,0,198,214,254,108,0,0,0,198,108,56,108,198,0,0,0,204,204,204,124,
12,248,0,0,252,152,48,100,252,0,28,48,48,224,48,48,28,0,24,24,24,0,24,24,24,0,224,48,48,28,48,48,224,0,118,
220,0,0,0,0,0,0,
0,16,56,108,198,198,254,0};

void echoxy(x, y, code, color)
int x,y;
unsigned char code;
unsigned int color;
{
    char i,j;

    setcolor(color);

    switch(default_textdir)
    {
        case horiz: for(j=0;j<=7;j++){
            for(i=0;i<=7;i++){
                switch(default_masktype){
                    case 1: if(odd(font[code][j] >> i)) putpixel(7+x-i,y+j,default_color);
                    break;
                    case 0: if(odd(font[code][j] >> i)) putpixel(7+x-i,y+j,default_color);
                    else putpixel(7+x-i,y+j,default_bkcolor);
                    break;
                }
            }
        }
        break;
        case vert: for(j=0;j<=7;j++){
            for(i=0;i<=7;i++){
                switch(default_masktype){
                    case 1: if(odd(font[code][j] >> i)) putpixel(x+j-7,y+i-8,default_color);
                    break;
                    case 0: if(odd(font[code][j] >> i)) putpixel(x+j-7,y+i-8,default_color);
                    else putpixel(x+j-7,y+i-8,default_bkcolor);
                    break;
                }
            }
        }
        break;
    }
}
}

```

```

void strxy(x,y,s,color)
int x,y;
unsigned char s[];
unsigned short color;
{
int l=0,i=0,j=0;
default_color=color;
while(s[l]!=0x00){
echoxy(x+1,y+j,s[l],color);
l++;
switch(default_textdir){
case 0: i+=8;
break;
case 90: j-=8;
break;
}
}
}

void echostrxy(x,y,s)
int x,y;
char s[];
{
strxy(x,y,s,default_color);
}

void strxyb(x,y,s,color)
int x,y;
unsigned char s[];
unsigned short color;
{
int saved=default_masktype;
default_masktype=0;
switch(default_textdir){
case 0: strxy(x,y,s,color);
default_masktype=1;
strxy(x+1,y,s,color);
break;
case 1: strxy(x,y,s,color);
default_masktype=1;
strxy(x,y-1,s,color);
break;
}
default_masktype=saved;
}

void doublexy(x,y,d,color)
int x,y;
double d;
int color;
{
char str[15];
int k;

sprintf(str,"%12.8f",d);
strxy(x,y,str,color);
}

void longxy(x,y,l,color)
int x,y,color;
long l;
{
char str[15];
int k;

sprintf(str,"%12d",l);
strxy(x,y,str,color);
}

```

E - Listagem da Biblioteca mickey-2.0.0.c

```

/*-----
mickey #2.0.0 - funcoes de suporte ao mouse
(C) 1999-2000 by L. C. Martins e-mail: lcm@if.ufrgs.br
Instituto de Fisica, UFRGS, Porto Alegre-RS, Brazil (lcm@if.ufrgs.br)
Depto de Fisica, UDESC/FEJ, Joinville-SC, Brazil (dfi2lcm@dcc.fej.udesc.br)

Esta biblioteca de funcoes do mouse foi escrita para uso um programa "c/c++",
especialmente desenvolvida para o sistema operacional Linux / GNU C (gcc), e
deve ser chamada com:

#include "mickey-2.0.0.c"

no inicio do programa fonte principal, ou dentro de um arquivo incluido por
este.

```

IMPORTANTE: - As funcoes graficas implementadas aqui usam o suporte grafico da biblioteca SVGALib, que deve estar pre-instalada no sistema;
 - Esta biblioteca foi testada com a SVGALib 1.3.0 no Red Hat Linux 5.1, para a arquitetura x86 (Intel);
 - Para o correto funcionamento das funcoes do mouse, este deve ser inicializado com cuidado pela funcao mouse_open();

Porto Alegre-RS, Brasil
 31/07/2000

#1.0.0 19/06/1999 Primeira tentativa
 #1.0.1 20/06/1999 Adicionado ponteiros graficos
 #1.0.3 21/06/1999 Adicionado o sistema de mascaras virtuais
 Comentarios e cabecalhos de funcoes
 #2.0.0 31/07/2000 Versao final revisada.

```

-----*/
#ifndef STDIO_H
#include <stdio.h>
#endif

#ifndef STDLIB_H
#include <stdlib.h>
#endif

#ifndef VGAGL_H
#include <vgagl.h>
#endif

#ifndef VGA_H
#include <vga.h>
#endif

#ifndef VGAKEYBOARD_H
#include <vgakeyboard.h>
#endif

#ifndef VGAMOUSE_H
#include <vgamouse.h>
#endif

#define FALSE      0
#define TRUE       1
#define false      0
#define true       1
#define ESC        27

typedef struct {
    unsigned short Mask[2][16];
    int HorzHotSpot,
        VertHotSpot;
} GraphCursMaskType;

static GraphCursMaskType
    StandardShapeCurs, mouse_icon;

static int
    mouse_ID      = 0,
    mouse_color   = 15,
    mouse_bkcolor = 0,
    mouse_status  = FALSE,
    mouse_started = FALSE,
    mouse_moved   = FALSE,
    mouse_x,
    mouse_y,
    mouse_xold,
    mouse_yold;

static unsigned short
    bit[16]= {0x0001,0x0002,0x0004,0x0008,
0x0010,0x0020,0x0040,0x0080,
             0x0100,0x0200,0x0400,0x0800,0x1000,0x2000,0x4000,0x8000};

static int
    mouse_saved[16][16];

void mouse_setcolor(color)
int color;
{
    mouse_color=color;
}

void mouse_setbkcolor(bkcolor)
int bkcolor;
{
    mouse_bkcolor=bkcolor;
}

unsigned short BinToWorld(s)
char s[];
{
    unsigned short w=0;
    int i;

    for (i=0;i<=15;i++){
        if(s[i]=='1') w |= bit[15-i];
    }
}

```

```

return w;
}

void DefineStandardShape(void)
{
    StandardShapeCurs.Mask[0][0] = BinToWorld("0011111111111111");
    StandardShapeCurs.Mask[0][1] = BinToWorld("0001111111111111");
    StandardShapeCurs.Mask[0][2] = BinToWorld("0000111111111111");
    StandardShapeCurs.Mask[0][3] = BinToWorld("0000011111111111");
    StandardShapeCurs.Mask[0][4] = BinToWorld("0000001111111111");
    StandardShapeCurs.Mask[0][5] = BinToWorld("0000000111111111");
    StandardShapeCurs.Mask[0][6] = BinToWorld("0000000011111111");
    StandardShapeCurs.Mask[0][7] = BinToWorld("0000000001111111");
    StandardShapeCurs.Mask[0][8] = BinToWorld("0000000000111111");
    StandardShapeCurs.Mask[0][9] = BinToWorld("0000000000011111");
    StandardShapeCurs.Mask[0][10] = BinToWorld("0000000111111111");
    StandardShapeCurs.Mask[0][11] = BinToWorld("0001000011111111");
    StandardShapeCurs.Mask[0][12] = BinToWorld("0011000011111111");
    StandardShapeCurs.Mask[0][13] = BinToWorld("1111000011111111");
    StandardShapeCurs.Mask[0][14] = BinToWorld("1111000011111111");
    StandardShapeCurs.Mask[0][15] = BinToWorld("1111100001111111");

    StandardShapeCurs.Mask[1][0] = BinToWorld("0000000000000000");
    StandardShapeCurs.Mask[1][1] = BinToWorld("0100000000000000");
    StandardShapeCurs.Mask[1][2] = BinToWorld("0110000000000000");
    StandardShapeCurs.Mask[1][3] = BinToWorld("0111000000000000");
    StandardShapeCurs.Mask[1][4] = BinToWorld("0111100000000000");
    StandardShapeCurs.Mask[1][5] = BinToWorld("0111110000000000");
    StandardShapeCurs.Mask[1][6] = BinToWorld("0111111000000000");
    StandardShapeCurs.Mask[1][7] = BinToWorld("0111111100000000");
    StandardShapeCurs.Mask[1][8] = BinToWorld("0111111100000000");
    StandardShapeCurs.Mask[1][9] = BinToWorld("0111111110000000");
    StandardShapeCurs.Mask[1][10] = BinToWorld("0111110000000000");
    StandardShapeCurs.Mask[1][11] = BinToWorld("0100011000000000");
    StandardShapeCurs.Mask[1][12] = BinToWorld("0000011000000000");
    StandardShapeCurs.Mask[1][13] = BinToWorld("0000001100000000");
    StandardShapeCurs.Mask[1][14] = BinToWorld("0000001100000000");
    StandardShapeCurs.Mask[1][15] = BinToWorld("0000001100000000");

    StandardShapeCurs.HorzHotSpot = -1;
    StandardShapeCurs.VertHotSpot = -1;
}

void mouse_setpointers(void)
{
    DefineStandardShape();
}

int mouse_inbox(x1,y1,x2,y2)
int x1,y1,x2,y2;
{
    mouse_update();
    mouse_x=mouse_getx();
    mouse_y=mouse_gety();
    if((mouse_x<x1) || (mouse_x>x2) || (mouse_y<y1) || (mouse_y>y2)) return 0;
    else return 1;
}

int mouse_left(void){
    mouse_update();
    return ((mouse_getbutton() & MOUSE_LEFTBUTTON)==MOUSE_LEFTBUTTON)
;
}

int mouse_right(void){
    mouse_update();
    return ((mouse_getbutton() & MOUSE_RIGHTBUTTON)==MOUSE_RIGHTBUTTON)
;
}

int mouse_both(void){
    if(mouse_left() && mouse_right()) return 1
; else return 0;
}

void mouse_push(x,y)
int x,y;
{
    int bit_,byte_,dx=mouse_icon.HorzHotSpot,dy=mouse_icon.VertHotSpot;
    for(byte_=0;byte_<=15;byte_++){
        for(bit_=0;bit_<=15;bit_++){
            mouse_saved[bit_][byte_]=vga_getpixel(x+bit_-dx,y+byte_-dy);
        }
    }
    mouse_started=TRUE;
}

```

```

}

void mouse_pop(x,y)
int x,y;
{
  int bit_,byte_,dx=mouse_icon.HorzHotSpot,dy=mouse_icon.VertHotSpot;
  if(mouse_started){
    for(byte_=0;byte_<=15;byte_++){
      for(bit_=0;bit_<=15;bit_++){
        vga_setcolor(mouse_saved[bit_][byte_]);
        vga_drawpixel(x+bit_-dx,y+byte_-dy);
      }
    }
  }
}

void mouse_fix(x,y)
int x,y;
{
  int i,j,bit_,byte_,dx=mouse_icon.HorzHotSpot,dy=mouse_icon.VertHotSpot;
  for(byte_=0;byte_<=15;byte_++){
    for(bit_=0;bit_<=15;bit_++){
      if((mouse_icon.Mask[0][byte_] & bit[bit_])==bit[bit_]){
        i=x+15-bit_-dx;
        j=y+byte_-dy;
        if((i<vga_getxdim()) && (j<vga_getydim())){
          vga_setcolor(mouse_saved[15-bit_][byte_]);
          vga_drawpixel(i,j);
        }
      }
    }
  }
}

void mouse_lock(x1,y1,x2,y2)
int x1,y1,x2,y2;
{
  mouse_setxrange(x1,x2);
  mouse_setyrange(y1,y2);
  if(mouse_status && mouse_started) mouse_pop(mouse_xold,mouse_yold);
  mouse_setposition((x2+x1)>>1,(y2+y1)>>1);
}

void mouse_free(void)
{
  mouse_setxrange(0,vga_getxdim());
  mouse_setyrange(0,vga_getydim());
}

void mouse_show(void)
{
  mouse_status=TRUE;
}

void mouse_hide(void)
{
  mouse_status=FALSE;
}

void mouse_draw(x,y)
int x,y;
{
  int i,j,bit_,byte_,dx=mouse_icon.HorzHotSpot,dy=mouse_icon.VertHotSpot;
  for(byte_=0;byte_<=15;byte_++){
    for(bit_=0;bit_<=15;bit_++){
      if((mouse_icon.Mask[1][byte_] & bit[bit_])==bit[bit_])
        vga_setcolor(mouse_color);
      else vga_setcolor(mouse_bkcolor);
      i=x+15-bit_-dx;
      j=y+byte_-dy;
      if((i<vga_getxdim()) && (j<vga_getydim())) vga_drawpixel(i,j);
    }
  }
}

void mouse_point(x,y)
int x,y;
{
  int bit_,byte_,dx=mouse_icon.HorzHotSpot,dy=mouse_icon.VertHotSpot;
  if(mouse_status){
    mouse_pop(mouse_xold,mouse_yold);
    mouse_push(x,y);
    mouse_draw(x,y);
    mouse_fix(x,y);
    mouse_xold=x;
    mouse_yold=y;
  }
}

```

```

}

void mouse_goto(x,y)
int x,y;
{
    mouse_setposition(x,y);
    mouse_moved=TRUE;
}

void mouse_open(mouse_type)
int mouse_type;
{
    mouse_init("/dev/mouse",mouse_type,MOUSE_DEFAULTSAMPLERATE);
    mouse_setpointers();
    mouse_icon=StandardShapeCurs;
    mouse_setscale(16);
    mouse_free();
    mouse_xold=vga_getxdim()>>1;
    mouse_yold=vga_getydim()>>1;
    mouse_push(mouse_xold,mouse_yold);
    mouse_setposition(mouse_xold,mouse_yold);
    mouse_show();
    if(mouse_status) mouse_point(mouse_xold,mouse_yold);
}

void mouse_move()
{
    mouse_update();
    mouse_x=mouse_getx();
    mouse_y=mouse_gety();
    if((mouse_x!=mouse_xold) || (mouse_y!=mouse_yold)){
        mouse_point(mouse_x,mouse_y);
        mouse_moved=TRUE;
    }
    else mouse_moved=FALSE;
}

void mouse_refresh()
{
    mouse_update();
    mouse_x=mouse_getx();
    mouse_y=mouse_gety();
    mouse_point(mouse_x,mouse_y);
    if((mouse_x!=mouse_xold) || (mouse_y!=mouse_yold)) mouse_moved=TRUE;
    else mouse_moved=FALSE;
}

void go(mouse_type)
int mouse_type;
{
    char ch, done=FALSE;
    int i,j,k;
    for(k=0;k<=1000;k++){
        vga_setcolor(1+(rand() % 255));
        vga_drawline(rand() % vga_getxdim(),rand() % vga_getydim(),
            rand() % vga_getxdim(),rand() % vga_getydim());
    }

    mouse_open(mouse_type);
    mouse_show();

    while((!done) && (!mouse_right()))
    {
        mouse_move();
        done=(vga_getkey()==ESC);
    }
    mouse_close();
}

void mouse_test(mouse_type, svga_mode)
int mouse_type, svga_mode;
{
    printf("\nVGAlib & vgamouse test by L(cm) 1999\n\n > Press a key to start,
    left_button to draw, right_button || ESC=EXIT\n");
    getchar();

    if(vga_init()==0){
        vga_setmode(svga_mode);
        go(mouse_type);
        vga_setmode(TEXT);
    }
    else printf("SVGAlib Error!");
    printf("\n Done!\n\n");
}

```


E - Lista de Figuras

3.1	<i>Diagramas de bifurcação para o intervalo $a \in [0.2, 1.8]$, com o valor inicial X_0 aleatório no intervalo $[0, 1]$, para o mapa logístico (a); e para o mapa quártico com $b = 1.35$ (b).</i>	13
3.2	<i>Diagrama de bifurcação do mapa quártico, construído como o da Fig. 3.1b, para $b = 1.35$. A estrutura do seu multi-atrator é revelada. Para $a = [1.10, 1.55]$, o atrator depende da condição inicial usada, e para $a = [0.7, 0.9]$, por exemplo, os diferentes atratores coincidem.</i>	14
3.3	<i>A dependência de $X_{t=1000}$ em função de X_0 para o mapa quártico, com parâmetros ($a_0 = 0.35, b_0 = 1.35$), onde um ciclo-2 é encontrado. A janela em zoom mostra detalhes da auto-similaridade encontrada para esse gráfico no limite $X_0 \rightarrow X_{0\infty} = 1.41918530$.</i>	15
3.4	<i>Mapa de segundo retorno $X_{t+2} \times X_t$ para o mapa quártico com $a = 0.35$ (a). Vista parcial em “zoom” (b). As intersecções da curva com a reta unitária (linha pontilhada) são os pontos fixos do mapa.</i>	17
3.5	<i>Mapa de segundo retorno $X_{t+2} \times X_t$ para o mapa quártico com $a = 0.35$, para (a) $b = 1.30$ e (b) $b = 1.35$, com órbita do ponto inicial $X_0 = -0.645$ mostrada para ambos.</i>	18
3.6	<i>Mapa de segundo retorno $X_{t+2} \times X_t$ para o mapa quártico com $a = 0.35$ e $b = 1.40$. O ponto inicial $X_0 = -0.645$ é atraído pelo atrator caótico.</i>	19
3.7	<i>O expoente de Lyapunov do mapa quártico para $a = 0.35$.</i>	21
3.8	<i>Tempo de convergência da série X_t do mapa quártico, para ($a = 0.35, b = 1.35$), inicializado com diferentes valores X_0 em $[0, 1.4]$.</i>	21
3.9	<i>Espaço de parâmetros do mapa quártico em “zoom” centrado em ($a_0 = 0.35, b_0 = 1.35$), para a condição inicial $X_0 = 1.0$ (a), e $X_0 = 0.0$ (b). Nas regiões pretas, o mapa é periódico de ciclo finito, e nas brancas, a dinâmica do mapa é caótica.</i>	22
3.10	<i>A dependência da condição inicial X_0 em torno de (a_0, b_0), com $b_0 = 1.35$ mantido constante (a); e com $a = 0.35$, constante. Nas regiões cinza (laterais) o mapa diverge, nas pretas a dinâmica é periódica, e nas brancas, caótica.</i>	23
4.1	<i>A rede hexagonal onde cada vértice possui três conexões ($S = 3$) mostrada em (c), pode ser mapeada sobre uma rede quadrada usual (a), removendo-se desta as conexões alternadas, mostras em (b) como linhas pontilhadas.</i>	27
4.2	<i>Dependência da rugosidade e p_3, para a rede desacoplada ($\varepsilon = 0$) de tamanho 512×512, iterada 256 vezes ($t = 256$), em função da amplitude RAD usada para a distribuição inicial da rede. Note-se que p_3 é proporcional à G_0, nesse caso.</i>	35
4.3	<i>Interface gráfica do programa <code>qmcml-2.0.0.c</code>, usado para a exploração interativa da rede (a). A paleta <code>rainbow.pal</code> de 256 cores usada para a visualização da rede.</i>	38
4.4	<i>Visualização tridimensional da rede na iteração seguinte à mostrada na Figura 4.3.</i>	38
5.1	<i>O espectro $\mathcal{P}[X(t)]$ para ($\varepsilon = 0$), nos instantes $t = 9999$ (a), e $t = 10000$ (b). Os picos observados estão sobre os valores fixos X_3^* e X_5^* do mapa local.</i>	40
5.2	<i>Estados finais da rede de mapas quárticos desacoplada ($\varepsilon = 0$), em duas iterações sucessivas $t = 9999$ (a), e $t = 10000$ (b).</i>	41
5.3	<i>Os instantâneos da rede em $t = 9999$ (a), e $t = 10000$ (b), para $\varepsilon = 0.40$.</i>	42

5.4	<i>As distribuições dos valores de X_t, para $t = 9999$ (a), e $t = 10000$ (b), para $\varepsilon = 0.40$.</i>	43
5.5	<i>Gráfico da rugosidade $G \times \varepsilon$ para a rede quadrada, primeiros vizinhos e inicialização RANDOM.</i>	44
5.6	<i>Uma solução diferente para $\varepsilon = 0.54$, a rede apresenta uma região sincronizada e outra caótica, para $t = 10000$ (a). O espectro correspondente para X_t (b). Devido à condição periódica de contorno, a única região sincronizada apresenta-se fragmentada nos vértices da rede, mas são partes de uma mesma região contígua.</i>	45
5.7	<i>Gráfico do valor médio $\langle X(t) \rangle \times \varepsilon$ para a rede quadrada, primeiros vizinhos e inicialização RANDOM.</i>	46
5.8	<i>Paredes fixas são encontradas separando diferentes regiões caóticas na rede, para $\varepsilon = 0.65$. A figura mostra a rede nos instantes $t=9999$ (a) e $t=10000$ (b), e os espectros correspondentes de X_t em (c) e (d).</i>	47
5.9	<i>A superfície espectral para todos os valores de $\varepsilon \in [0, 1]$ usados, avaliados nos instantes $t=9999$ (a) e $t=10000$ (b).</i>	48
5.10	<i>O efeito de desdobramento da rede completa para $\varepsilon = 1$ (a), em duas sub-redes independentes (b) e (c).</i>	48
5.11	<i>O estado final da rede para o acoplamento democrático $\varepsilon_d = 0.80$ (a), e um detalhe (b) centrado em uma pequena região sincronizada (3), cercada por regiões de caos (1-2), evidenciando as paredes fixas diagonais. A região ampliada tem tamanho igual a 32×32 mapas, e está marcada em (a).</i>	49
5.12	<i>O estado da rede no instante $t = 10000$, para $\varepsilon = 0.65$ (a), e $\varepsilon = 1.00$ (b), mostrando o sistema de paredes fixas.</i>	50
5.13	<i>Gráfico da rugosidade G para a rede quadrada, segundos vizinhos e inicialização RANDOM (a). A evolução de $\langle X(t) \rangle$ para os diferentes valores de ε (b).</i>	51
5.14	<i>A superfície de espectros $\mathcal{P}[X(t)]$ para o acoplamento de segundos vizinhos, rede quadrada e inicialização RANDOM, para $t=9999$ (a) e $t=10000$ (b).</i>	51
5.15	<i>Gráfico da rugosidade $G \times \varepsilon$ para a rede quadrada, segundos vizinhos e inicialização RING50 (a). A evolução de $\langle X(t) \rangle$ para os diferentes valores de ε (b).</i>	52
5.16	<i>A evolução dos espectros $\mathcal{P}[X(t)]$ para a inicialização RING50%, nos instantes consecutivos $t = 9999$ (a) e $t = 10000$ (b).</i>	53
5.17	<i>Dois estados singulares da rede no instante $t = 9999$, para $\varepsilon = 0.70$ (a), e $\varepsilon = 0.83$ (b), mostrando uma fase caótica (circular) sobre a rede sincronizada.</i>	54
5.18	<i>Um estado característico da rede para $\varepsilon = 0.51$ nos instantes $t = 9999$ (a), e $t = 10000$ (b), mostrando as fases caóticas isoladas por paredes fixas.</i>	55
5.19	<i>Gráfico da rugosidade $G \times \varepsilon$ para a rede quadrada, segundos vizinhos e inicialização RING50% (a). E o gráfico de $\langle X(t) \rangle \times \varepsilon$ (b).</i>	56
5.20	<i>Alguns estados finais da rede em $t = 10000$ para a inicialização PATCH3, para $\varepsilon = 0.00$ (a), $\varepsilon = 0.47$ (b) e $\varepsilon = 0.49$ (c), em cima. E para $\varepsilon = 0.67$ (instável para $t \rightarrow \infty$) (d), $\varepsilon = 0.70$ (e) e $\varepsilon = 0.85$ (f), em baixo. A pequena área de 16×16 mapas marcada em (c) tem seus valores numéricos apresentados na Tabela 5.1.</i>	59
5.21	<i>Gráfico da rugosidade $G \times \varepsilon$ para a rede quadrada, primeiros vizinhos e inicialização PATCH3 (a). E o gráfico de $\langle X(t) \rangle \times \varepsilon$ (b).</i>	60
5.22	<i>A evolução dos espectros $\mathcal{P}[X(t)]$ para a inicialização PATCH3, para $t = 9999$ (a), e $t = 10000$ (b).</i>	61
5.23	<i>A rede quadrada com inicialização PATCH3, segundos vizinhos, para $\varepsilon = 0.49$ (a), e $\varepsilon = 0.56$ (b).</i>	61
5.24	<i>A rugosidade da rede quadrada, primeiros vizinhos e inicialização PATCH3 (a), e o valor médio de $X_t(\varepsilon)$ (b).</i>	62
5.25	<i>Para $\varepsilon = 0.11$ a rede apresenta a região central sincronizada e outra bloqueada, para $t = 10000$ (a). O espectro correspondente (b).</i>	63
5.26	<i>Gráfico da rugosidade $G \times \varepsilon$ para a rede quadrada, primeiros vizinhos e inicialização PATCH3 (a). E o gráfico de $\langle X(t) \rangle \times \varepsilon$ (b).</i>	64
5.27	<i>Para $\varepsilon = 0.45$, paredes hexagonais separam regiões de CCNT sobre a rede sincronizada (a); o espectro de X_t correspondente (b).</i>	64
5.28	<i>A evolução dos espectros $\mathcal{P}[X(t)]$ para a inicialização PATCH3, para $t = 10000$.</i>	65

1	<i>Rede QUADRADA, inicialização RANDOM, PRIMEIROS vizinhos, t = 9999 e t = 10000.</i>	72
2	<i>Rede QUADRADA, inicialização RANDOM, SEGUNDOS vizinhos, t = 9999 e t = 10000.</i>	73
3	<i>Rede HEXAGONAL, inicialização RANDOM, PRIMEIROS vizinhos, t = 9999 e t = 10000.</i>	74
4	<i>Rede QUADRADA, inicialização RING%50, PRIMEIROS vizinhos, t = 9999 e t = 10000.</i>	75
5	<i>Rede QUADRADA, inicialização RING%50, SEGUNDOS vizinhos, t = 9999 e t = 10000.</i>	76
6	<i>Rede HEXAGONAL, inicialização RING50%, PRIMEIROS vizinhos, t = 9999 e t = 10000.</i>	77
7	<i>Rede QUADRADA, inicialização PATCH3, PRIMEIROS vizinhos, t = 9999 e t = 10000.</i>	78
8	<i>Rede QUADRADA, inicialização PATCH3, SEGUNDOS vizinhos, t = 9999 e t = 10000.</i>	79
9	<i>Rede HEXAGONAL, inicialização PATCH3, PRIMEIROS vizinhos, t = 9999 e t = 10000.</i>	80
10	<i>Rede QUADRADA, inicialização PATCH5, PRIMEIROS vizinhos, t = 9999 e t = 10000.</i>	81
11	<i>Rede QUADRADA, inicialização PATCH5, SEGUNDOS vizinhos, t = 9999 e t = 10000.</i>	82
12	<i>Rede HEXAGONAL, inicialização PATCH5, PRIMEIROS vizinhos, t = 9999 e t = 10000.</i>	83
13	<i>Fluxograma simplificado para o programa qmcml-2.0.0.c</i>	84

F - Lista de Tabelas

2.1	<i>Os sistemas espacialmente extensos mais estudados, classificados quanto ao tipo de variáveis.</i>	8
3.1	<i>Os conjunto dos valores iniciais críticos $\{X_{0k}\}$, para as inversões $X_3^* \rightarrow X_5^*$ para o ciclo-2 do mapa quártico com $a = 0.35$ e $b = 1.35$.</i>	16
3.2	<i>Os pontos fixos para o diagrama de segundo retorno $X_{t+2} = X_t$ para o ciclo-2 do mapa quártico com $a = 0.35$ e $b = 1.35$. Destes pontos fixos, apenas dois X_3^* e X_5^* são estáveis (E), os outros são instáveis (I).</i>	16
4.1	<i>Valores numéricos para a probabilidade p_3 e a rugosidade g_0, para redes de vários tamanhos, com distribuição uniforme no intervalo $[-1.4, 1.4]$.</i>	34
5.1	<i>Os valores numéricos de $X_{t=10000}^{i,j}$ para $i, j = 45, 46, \dots, 60$ para a região marcada na Fig. 5.20c, na fronteira da parte central sincronizada (cinza). O brilho e o contraste da imagem da rede foram aumentados para facilitar a visualização dos números.</i>	57
5.2	<i>A evolução temporal dos valores numéricos para a linha 55 da rede mostrada na Tabela 5.1, isto é, $X_t^{i=55,j}$ para $j = 45, \dots, 60$, e $t = 9985, \dots, 10000$. As colunas 52 – 54 (em negrito) correspondem, respectivamente, às diagonais azul, amarela e vermelha daquela tabela.</i>	58
5.3	<i>Classificação geral das diferentes dinâmicas assintóticas para as redes de mapas quárticos acoplados, com parâmetros ($a_0 = 0.35, b_0 = 1.35$), para as inicializações e geometrias testadas. Nos tipos marcados com ‘*’ a rede apresenta paredes rígidas bem definidas, com ‘!’ apresenta pequenas regiões sincronizadas, com ‘!!’ apresenta regiões de caos sobre fundo sincronizado e com ‘?’ apresenta alternância dos regimes marcados, para diversos valores de ε no intervalo.</i>	66

H - Lista de Citações

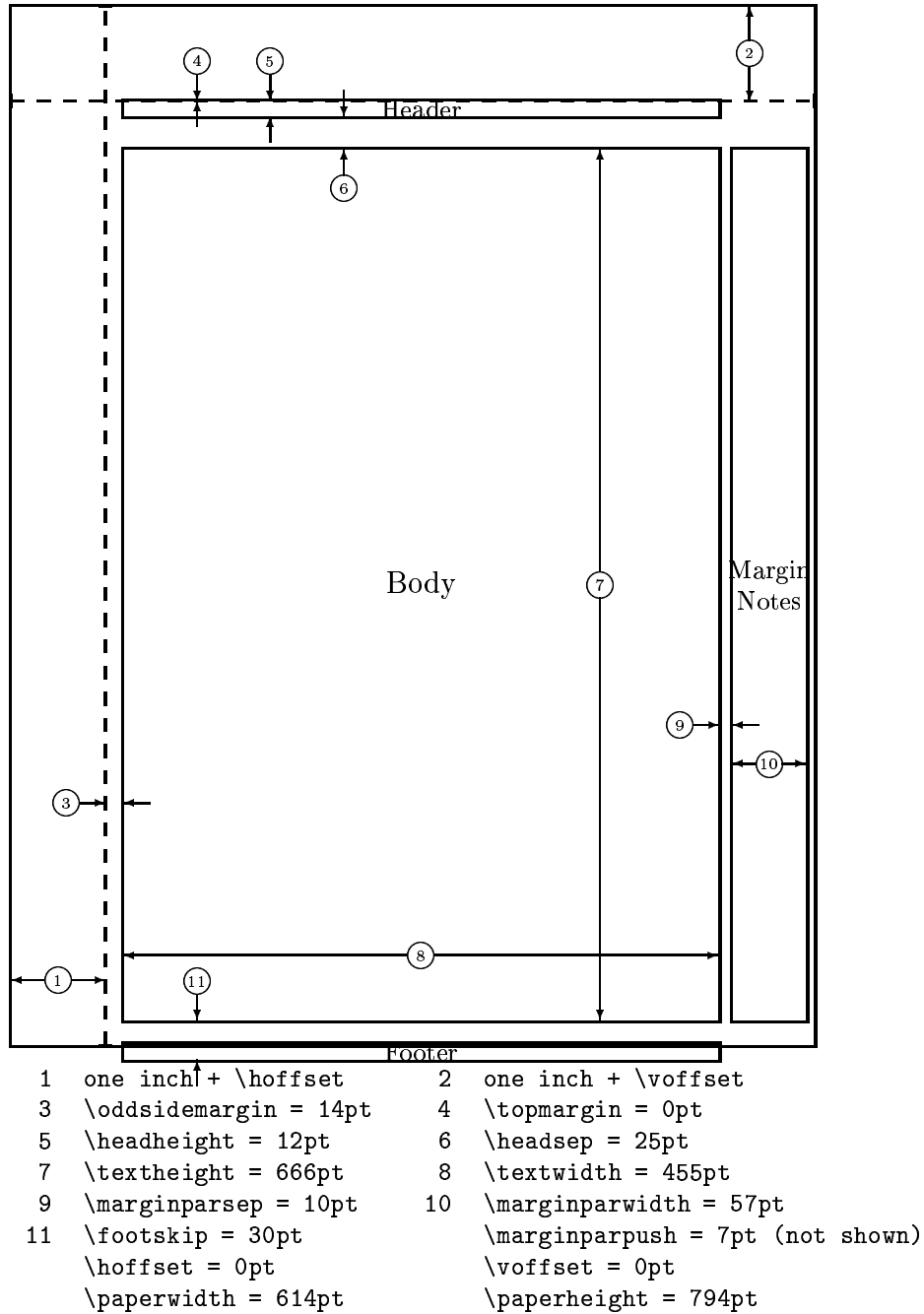
Formato:

[AUTOR, ANO] capítulo, página(s)

[ABRAMSON <i>et al.</i> , 1998]	1, 5	[KANEKO, 1990]	2, 11
[ANDRADE, 1999]	1, 3	[KANEKO, 1993]	2, 8
[BERNARDES <i>et al.</i> , 1997]	1, 2	[KURAMOTO, 1984]	1, 3
[BRUNET <i>et al.</i> , 1994]	2, 8	[LEMAÎTRE <i>et al.</i> , 1998]	6, 69
[BRUNET <i>et al.</i> , 1997]	1, 2	[LEMAÎTRE <i>et al.</i> , 1999b]	6, 69
[BRUNET <i>et al.</i> , 1998a]	2, 10	[LEMAÎTRE <i>et al.</i> , 1999]	6, 69
[BRUNET <i>et al.</i> , 1998b]	1, 1, 4	[LOPES <i>et al.</i> , 1998]	2, 9
[BRUNET <i>et al.</i> , 1998b]	3, 15	[LOSSON <i>et al.</i> , 1994]	2, 10
[BRUNET <i>et al.</i> , 1998b]	4, 24	[MARCQ <i>et al.</i> , 1996]	6, 70
[CHATÉ <i>et al.</i> , 1992]	4, 30	[MARCQ <i>et al.</i> , 1997]	1, 3
[CHATÉ <i>et al.</i> , 1996]	4, 25	[MARCQ <i>et al.</i> , 1997]	4, 27
[CHATÉ <i>et al.</i> , 1997]	6, 69	[MARCQ <i>et al.</i> , 1998]	6, 70
[CHATÉ, 1998]	4, 24	[MEDVEDEV <i>et al.</i> , 1998]	2, 9
[COSENZA, 1995]	4, 25	[MORELLI <i>et al.</i> , 1998]	2, 9
[COSENZA <i>et al.</i> , 1996]	2, 11	[OSIPOV <i>et al.</i> , 1997]	2, 10
[GALLAS, 1993]	1, 3	[PIKOVSKY <i>et al.</i> , 1997]	1, 3
[GLASS <i>et al.</i> , 1997]	1, 2	[PINTO, 1999]	2, 10
[GOOSSENS <i>et al.</i> , 1994] Apêndice I , 118		[SINHA <i>et al.</i> , 1998]	1, 2
[HEMMINGSSON <i>et al.</i> , 1993]	2, 9	[SOLÉ <i>et al.</i> , 1992]	1, 2
[HOHL <i>et al.</i> , 1997]	2, 10	[STROGATZ <i>et al.</i> , 1993]	1, 2
[JÁNOSI <i>et al.</i> , 1999]	6, 69	[STROGATZ, 1994]	1, 2
[KANEKO, 1983]	1, 1	[THAM <i>et al.</i> , 1992]	2, 8
[KANEKO, 1983]	2, 10	[VIEIRA <i>et al.</i> , 1994]	1, 3
[KANEKO, 1989b]	6, 69	[WIESENFELD <i>et al.</i> , 1989]	1, 2
[KANEKO, 1989c]	2, 10	[WIESENFELD <i>et al.</i> , 1989]	2, 10
[KANEKO, 1989c]	4, 24	[WIESENFELD <i>et al.</i> , 1996]	1, 2
[KANEKO, 1989a]	1, 1	[WIESENFELD <i>et al.</i> , 1996]	2, 10
[KANEKO, 1989a]	2, 11		

I - L^AT_EX Layout

Este documento foi preparado com o L^AT_EX₂ ϵ , no formato `report.cls` escalonado em 12pt, com a formatação original mantida, exceto a altura (`\textheight`) e a largura do texto (`\textwidth`) [GOOSSENS *et al.*, 1994].



Referências Bibliográficas

ABRAMSON, G.; ZANETTE, D. H. Globally coupled maps with asynchronous updating. *Physical Review E*, Woodbury, v. 58, n. 4, p. 4454-4460, Oct. 1998.

ANDRADE, R. F. Thermodynamical behavior of aperiodic Ising models on hierarchical lattices. *Physical Review E*, Woodbury, v. 59, n. 1, p. 150-157, Jan. 1999.

BERNARDES, A. T.; SANTOS, R. M. Z. Immune network at the edge of chaos. *Journal of Theoretical Biology*, London, v. 186, n. 2, p. 173-187, May 1997.

BRUNET, L. G.; CHATÉ, H. Phase coherence in chaotic oscillatory media. *Physica A*, Amsterdam, v. 257, n. 1/4, p. 347-356, Aug. 1998.

BRUNET, L. G.; CHATÉ, H.; MANNEVILLE, P. Long-range order with local chaos in lattice of diffusively coupled ODEs. *Physica D*, Amsterdam, v. 78, n. 3/4, p. 141-154, Nov. 1994.

BRUNET, L. G.; GALLAS, J. A. C. Exploring collective behaviors with a multi-attractor quartic map. *Physica A*, Amsterdam, v. 257, n. 1/4, p. 329-333, Aug. 1998.

BRUNET, L. G.; GONÇALVES, S. Celular automaton block model of traffic in a city. *Physica A*, Amsterdam, v. 237, n. 1/2, p. 59-66, Mar. 1997.

CHATÉ, H. Emergence of collective behavior in large chaotic dynamical systems. *International Journal of Modern Physics B*, Singapore, v. 12, n. 3, p. 299-308, Jan. 1998.

CHATÉ, H.; LEMAITRE, A.; MARCQ, Ph.; MANNEVILLE, P. Non-trivial collective behavior in extensively-chaotic dynamical systems: an update. *Physica A*, Amsterdam, v. 224, n. 1/2, p. 447-457, Feb. 1996.

CHATÉ, H.; LOSSON J. Non-trivial collective behavior in coupled map lattices: a transfer operator perspective. *Physica D*, Amsterdam, v. 103, n. 1/4, p. 51-72, Apr. 1997.

CHATÉ, H.; MANNEVILLE, P. Collective behaviors in spatially extended systems with local interactions and synchronous updating. *Progress in Theoretical Physics*, Kyoto, v. 87, n. 1, p. 1-60, Jan. 1992.

CHATTERJEE, N.; GUPTA, N. Synchronization in coupled sine circle maps. *Physical Review E*, Woodbury, v. 53, n. 5, p. 4457-4466, May 1996.

COSENZA, M. G. Collective behavior of coupled chaotic maps. *Physics Letters A*, Amsterdam, v. 204, n. 2, p. 128-132, Aug. 1995.

COSENZA, M. G.; PARRAVANO, A. Turbulence in globally coupled maps. *Physical Review E*, Woodbury, v. 53, n. 6, p. 6032-6037, June 1996.

GALLAS, J. A. C. Simulating memory effects with discrete dynamic systems. *Physica A*, Amsterdam, v. 195, n. 3/4, p. 417-430, May 1993.

GLASS, L.; MACKEY, M. C. *Dos relógios ao caos: os ritmos da vida*. São Paulo: Editora da Universidade de São Paulo, 1997.

GOOSSENS, M.; MITTELBACH, F.; SAMARIN A. *The LaTeX Companion*. Reading: Addison-Wesley, 1994.

GRASSBERGER, P.; SCHREIBER, T. Phase transitions in coupled map lattices. *Physica D*, Amsterdam, v. 50, n. 2, p. 177-188, June 1991.

HEMMINGSSON, J.; SORENSEN, A.; FLYVBJERG, H.; HERRMANN, H. J. What synchronization? *Europhysics Letters*, Les Ulis, v. 23, n. 9, p. 629-634, Sept. 1993.

HOHL, A.; GAVRIELIDES, A.; ERNEUX, T.; KOVANIS, V. Localized synchronization in two coupled nonidentical semiconductor lasers. *Physical Review Letters*, Woodbury, v. 78, n. 25, p. 4745-4748, June 1997.

JÁNOSI, I. M.; GALLAS, J. A. C. Globally coupled multiattractor maps: mean field dynamics controlled by the number of elements. *Physical Review E*, Wood-

bury, v. 59, n. 1, p. R28-R31, Jan. 1999.

KANEKO, K. *Collapse of tori and genesis of chaos in dissipative systems*. Singapore: World Scientific, 1986. Thesis.

KANEKO, K. Pattern dynamics in spatiotemporal chaos. *Physica D*, Amsterdam, v. 34, n. 1/2, p. 1-41, Jan./Feb. 1989.

KANEKO, K. Self-consistent Perron-Frobenius operator for spatiotemporal chaos. *Physics Letters A*, Amsterdam, v. 139, n. 1/2, p. 47-52, July 1989.

KANEKO, K. Spatiotemporal chaos in one and two-dimensional coupled map lattices. *Physica D*, Amsterdam, v. 37, n. 1/3, p. 60-82, July 1989.

KANEKO, K. Clustering, coding, switching, hierarchical ordering, and control in a network of chaotic elements, *Physica D*, Amsterdam, v. 41, n. 2, p. 137-172, Mar. 1990.

KANEKO, K. *Theory and applications of coupled map lattices*. New York: John Wiley, 1993.

KELLER, G.; KÜNZLE, M.; NOWICKI, T. Some phase transitions in coupled map lattices. *Physica D*, Amsterdam, v. 59, n. 1/3, p. 39-51, Oct. 1992.

KURAMOTO, Y. *Chemical oscillations, waves and turbulence*. Berlin: Springer-Verlag, 1984.

LEMAÎTRE, A.; CHATÉ, H. Nonperturbative renormalization group for chaotic coupled map lattices. *Physical Review Letters*, Woodbury, v. 80, n. 25, p. 5528-5531, June 1998.

LEMAÎTRE, A.; CHATÉ, H. Macroscopic model of collective behavior of chaotic coupled map lattices. *Europhysics Letters*, Les Ulis, v. 46, n. 5, p. 565-570, June 1999.

LEMAÎTRE, A.; CHATÉ, H. Phase ordering and onset of collective behavior in chaotic coupled map lattices. *Physical Review Letters*, Woodbury, v. 82, n. 6, p. 1140-1143, Feb. 1999.

LOPES, S.R.; RIZZATO, F. B. Chaos and energy redistribution in the nonlinear interaction of two spatio-temporal wave triplets. *Physica D*, Amsterdam, v. 117, n. 1/4, p. 13-25, June 1998.

LOSSON, J.; MACKEY, M. C. Statistical cycling in coupled map lattices. *Physical Review E*, Woodbury, v. 50, n. 2, p. 843-856, Aug. 1994.

MARCQ, P.; CHATÉ, H.; MANNEVILLE, P. Universal critical behavior in two-dimensional coupled map lattices. *Physical Review Letters*, Woodbury, v. 77, n. 19, p. 4003-4006, Nov. 1996.

MARCQ, P.; CHATÉ, H.; MANNEVILLE, P. Universality in Ising-like phase transitions of lattices of coupled chaotic maps. *Physical Review E*, Woodbury, v. 55, n. 3, p. 2606-2627, Mar. 1997.

MARCQ, P.; CHATÉ, H.; MANNEVILLE, P. Universality in Ising-like phase transitions of lattices of coupled chaotic maps. *Physical Review E*, Woodbury, v. 57, n. 2, p. 1591-1603, Feb. 1998.

MEDVEDEV, M. V.; DIAMOND, P. H. Self-organized states in cellular automata: exact solution. *Physical Review E*, Woodbury, v. 58, n. 5, p. 6824-6827, Nov. 1998.

MORELLI, L. G.; ZANETTE, D. Synchronization of stochastically coupled cellular automata. *Physical Review E*, Woodbury, v. 58, n. 1, p. R8-R11, July 1998.

OSIPOV, G. V.; PIKOVSKY, A.; ROSEMBLUM, M.; KURTHS, J. Phase synchronization effects in a lattice of nonidentical Rössler oscillator. *Physical Review E*, Woodbury, v. 55, n. 3, p. 2353-2361, Mar. 1997.

PIKOVSKY, A.; OSIPOV, G. V.; ROSEMBLUM, M.; ZAKS, M.; KURTHS, J. Attractor-repeller collision and eyelet intermittency at the transition to phase synchronization. *Physical Review Letters*, Woodbury, v. 79, n. 1, p. 47-50, July 1997.

PINTO, S. E. de S. *Platôs de sincronização em redes de sistemas dinâmicos com tempo discreto*. [Curitiba]: Curso de Pós-Graduação em Física - UFPR, [1999]. Diss. maestr. Física.

SOLÉ, R. V.; VALLS, J. On structural stability and chaos in biological systems. *Journal of Theoretical Biology*, London, v. 155, n. 1, p. 87-102, Mar. 1992.

SOMPOLINSKI, H.; GOLOMB, D.; KLEINFED, D. Cooperative dynamics in visual processing. *Physical Review A*, Woodbury, v. 43, n. 12, p. 6990-7011, June 1991.

SINHA, S.; DITTO, W. L. Dynamics based computation. *Physical Review Letters*, Woodbury, v. 81, n. 10, p. 2156-2159, Sept. 1998.

STROGATZ, S. H.; STEWART, I. Coupled oscillators and biological synchronization. *Scientific American*, New York, v. 269, n. 6, p. 102-109, Dec. 1993.

STROGATZ, S. H. *Nonlinear dynamics and chaos*. New York: Perseus Books, 1994.

TANAKA, H.; LICHTENBERG, A. J.; OISHI, S. First order phase transition resulting from finite inertia in coupled oscillator systems. *Physical Review Letters*, Woodbury, v. 78, n. 11, p. 2104-2107, Mar. 1997.

THAM, P.; SEN, A. K. Remote multimode feedback stabilization of plasma instabilities. *Physical Review A*, Woodbury, v. 46, n. 8, p. R4520-R4523, Oct. 1992.

VIANA, R. L.; BATISTA, A. M. Synchronization of coupled kicked limit cycle systems. *Chaos, Solitons and Fractals*, Oxford, v. 9, n. 12, p. 1931-1944, Dec. 1998.

VIEIRA, M. S.; LICHTENBERG, A. J.; LIEBERMAN, M. A. Self-synchronization of many coupled oscillators. *International Journal of Bifurcation and Chaos in Applied Sciences and Engineering*, Singapore, v. 4, n. 6, p. 1563-1577, Dec. 1994.

WIESENFELD, K.; HADLEY, P. Attractor crowding in oscillator arrays. *Physical Review Letters*, Woodbury, v. 62, n. 12, p. 1335-1338, Mar. 1989.

WIESENFELD, K.; COLET, P.; STROGATZ, S. H. Synchronization transitions in a disordered Josephson series array. *Physical Review Letters*, Woodbury, v. 76, n. 3, p. 404-407, Jan. 1996.