



Universidade de Brasília

Instituto de Física

Pós-Graduação

**Caos em Sistemas Hamiltonianos:
Métodos Numéricos para o Cálculo dos
Expoentes de Lyapunov**

Luciano Hugo Miranda Filho

Tese de doutorado

Brasília

11 de Outubro de 2016

Universidade de Brasília

Instituto de Física

Luciano Hugo Miranda Filho

**Caos em Sistemas Hamiltonianos: Métodos Numéricos para
o Cálculo dos Expoentes de Lyapunov**

*Trabalho apresentado ao Programa de Pós-Graduação do
Instituto de Física da Universidade de Brasília como requi-
sito parcial para obtenção do grau de Doutor.*

Orientador: *Professor Marco A. Amato*

Co-orientador: *Professor Yves Elskens*

Brasília

11 de Outubro de 2016

*Dedico este trabalho à minha velha Mãe e meus quatro
sobrinhos: Manuela, Marina, Guilherme e Gabriel.*

Agradecimentos

Agradeço aos professores do programa de pós-graduação do instituto de Física da Universidade de Brasília pelo empenho. Em particular um muito obrigado ao professor Dr. Marco A. Amato, pela boa convivência e qualidade da orientação, aos professores Dr. Tarcísio Marciano, Dr. Annibal Figueiredo, pela disponibilidade.

Agradeço aos funcionários do Instituto de Física pelo comprometimento.

Aux collègues et professeurs du groupe Physique des Interactions ioniques et Moléculaires de l'Université d'Aix-Marseille, je vous remercie de m'avoir accueilli et de m'avoir fait sentir comme chez moi. Je suis tout spécialement reconnaissant au docteur Yves Elskens, de sa patience et de la qualité de ses conseils, et au docteur Lénaïc Couëdel.

A minha companheira Marine Bradshaw, pelo carinho e apoio durante esse período tão importante da minha vida.

Um salve ao Rugby sempre presente na minha vida! Abraço a todos os camaradas do UnB Rugby Club e para a equipe Dinovale, da universidade de Aix-Marseille.

Agradeço a CAPES pelo apoio financeiro no desenvolvimento do projeto.

Em memória, agradeço meu saudoso pai, por tudo que me ensinou.

Amar é querer estar perto, se longe; e mais perto, se perto.

—VINÍCIOS DE MORAES

Resumo

Os expoentes de Lyapunov medem o nível de sensibilidade das trajetórias à pequenas variações nas condições iniciais. Essa propriedade é essencial na determinação de caoticidade nos sistemas dinâmicos. No contexto dos sistemas com interação de longo alcance, comparamos dois métodos usualmente utilizados na obtenção dos expoentes de Lyapunov: O método dos clones e o método tangente. O método dos clones calcula o espectro de Lyapunov gerando cópias de uma dinâmica referência, perturbadas em diferentes direções do espaço de fases. Já o método tangente depende fortemente da versão linearizada do sistema estudado. Os resultados apresentados, através da aplicação de ambos os métodos, dizem respeito ao modelo Hamiltoniano de campo médio com potencial cosseno e ao sistema de folhas autogravitantes.

Palavras-chave: Caos, Estimativa do espectro de Lyapunov, interação de longo alcance, Hamiltoniano de campo médio, Sistema de folhas gravitacional

Abstract

The Lyapunov exponents measure the sensitivity level of the trajectories to small variations in the initial conditions. Such property is essential to determine the chaoticity of dynamical systems. In the context of the long range interacting systems, we compare two methods commonly used to obtain the Lyapunov exponents: the method of clones (CM) and the tangent map (TM). The method of clones calculates the Lyapunov spectrum (LS) generating small perturbations of the original dynamics for different directions in the phase space. The tangent map approach relies strongly on the construction of a linearized version of the system in study. Numerical results of both methods are applied to cosine Hamiltonian Mean Field Model (HMF) and the 1-dimensional Gravitational sheet model (1DGS).

Keywords: Chaos, Lyapunov spectrum estimation, Long range interaction, Hamiltonian mean field model, gravitating sheet system

Sumário

1	Introdução	1
2	Sistemas com interação de longo alcance	7
2.1	Hamiltoniano de campo médio	8
2.2	Modelo de folhas autogravitantes	14
3	Métodos para o cálculo dos expoentes de Lyapunov	19
3.1	Os Expoentes de Lyapunov	19
3.2	Descrição dos Métodos	21
3.2.1	Mapa Tangente	22
3.2.2	Dinâmica de Clones	23
3.3	Obtenção dos expoentes de Lyapunov para o HMF	24
3.4	Obtenção dos expoentes de Lyapunov para o modelo de folhas autogravitantes	26
3.5	Análise de convergência	28
3.6	Integradores numéricos	29
3.6.1	Integradores simpléticos	31
3.6.2	Primeira ordem	32
3.6.3	Segunda ordem (<i>Leapfrog</i>)	33
3.7	Ortonormalização de Gram-Schmidt	34
4	Resultados e discussões	37
4.1	Hamiltoniano de campo médio	37
4.2	Modelo de folhas autogravitantes	48
5	Considerações finais	51

A	Aplicação do integrador simplético ao modelo onda-partícula	53
B	Algoritmos	59
C	Rotinas computacionais relativas à dinâmica do HMF	65
C.1	Evolução e cálculo dos expoentes de Lyapunov utilizando o método tangente	65
C.2	Evolução e cálculo dos expoentes de Lyapunov utilizando o método dos clones	71
D	Rotinas computacionais relativas à dinâmica do modelo de folhas autogravitantes	77
D.1	Evolução e cálculo dos expoentes de Lyapunov	77

Lista de Figuras

1.1	Equação de Lorenz, $b = 8/3$, $r = 28$, $\sigma = 10$.	3
2.1	HMF, $N = 500$, $e = 0.5$. Na esquerda o espaço real. A figura da direita exhibe o espaço de fases. tempo total $t = 1000$, sendo o tempo definido como o número de iterações vezes o tamanho do passo do integrador.	9
2.2	Caracterização das partículas utilizando a energia como critério.	11
2.3	Fluxograma resumindo a evolução dos sistemas de longo alcance	12
2.4	HMF, $N = 20000$. Comportamento do módulo do vetor magnetização para diferentes valores da energia por partícula.	12
2.5	HMF, $N = 1000$, $e = 0.56$, $M_0 = 0.46$. Representação da distribuição <i>waterbag</i> .	13
2.6	HMF, $N = 40, 80, 450$. Parâmetro de ordem M versus energia por partícula e .	14
2.7	HMF, $N = 100$, $e = 0.6$. A esquerda temos o valor da curtose. O gráfico da direita mostra a distribuição dos momentos.	15
2.8	Folhas autogravitantes, $N = 1000$, $e = 1$. Representação da distribuição <i>waterbag</i> , sendo utilizada como condição inicial, seguida da configuração do sistema instantes depois.	17
3.1	Esquema da construção do integrador simplético <i>leapfrog</i> .	26
3.2	Método de Euler. O valor da derivada, utilizado como ponto de partida, é extrapolado em cada intervalo com o intuito de encontrar o próximo valor da função.	30
4.1	HMF, $N = 10000$, $e = 0.55$. Erro relativo da energia versus o tempo.	38
4.2	HMF, $N = 10000$, $e = 0.55$. Valor do momento total em cada instante de tempo.	38

- 4.3 HMF, $N = 200$, $e = 0.55$. Convergência do maior expoente de Lyapunov. As figuras exibem a performance dos dois métodos para duas condições iniciais distintas, *waterbag* e Gaussiana nas posições e momentos. 39
- 4.4 HMF, $N = 30$, $e = 0.55$. Espectro completo de Lyapunov. 40
- 4.5 HMF. Módulo do vetor magnetização e o valor λ_1 em função da energia para $N = 50$, $N = 100$ e $N = 450$ partículas. 40
- 4.6 HMF, $e = 1.2$. Valor do maior expoente de Lyapunov em função do número de partículas. 41
- 4.7 HMF. Valor do maior expoente de Lyapunov em função da energia para $N = 50$, $N = 100$, $N = 450$, e 1500 partículas. O gráfico da direita teve o eixo das ordenadas multiplicados pelo fator de escala $\alpha N^{-\beta}$, cujos valores $\alpha \approx 0.52$ e $\beta \approx 1/4$ estão explicitados na figura (4.6). 41
- 4.8 HMF, $N = 10000$, $e = 0.55$. Representação da distribuição δI_i no espaço de fases para o caso subcrítico. 42
- 4.9 HMF. Classificação, segundo o critério de energia, dos estados das partículas utilizando diferentes cores. 44
- 4.10 HMF, $N = 50$, $e = 0.1$. Convergência do maior expoente de Lyapunov em função do tempo. 44
- 4.11 HMF, $N = 50$, $e = 0.1$. Na figura foram registrados os valores dos expoentes de Lyapunov com diferentes tamanhos de passo do integrador. 47
- 4.12 Folhas gravitacionais, $N = 75$, $e = 1.0$. Erro relativo da energia versus o tempo. 48
- 4.13 Folhas gravitacionais, $N = 75$, $e = 1.0$. Erro relativo do momento versus o tempo. 48
- 4.14 Folhas gravitacionais, $N = 75$, $e = 1$. Convergência da medida do maior expoente de Lyapunov em função do tempo para várias realizações, incluindo o valor médio e o desvio. Medidas obtidas através do método tangente. 49
- 4.15 Folhas gravitacionais, $N = 30$, $e = 1$. Espectro completo de Lyapunov. 49
- 4.16 Folhas gravitacionais, $e = 1$. Valor do maior expoente de Lyapunov e da entropia de Kolmogorov-Sinai, definida como a soma de todos os expoentes positivos, contra o número de partículas. 50

- A.1 Onda- partícula, $N = M = 1$, $\omega = 1$, $\beta = 0$, $\kappa = 1$, $\varepsilon = 1$. As condições iniciais utilizadas são $x(0) = 0$, $p(0) = 1$, $X(0) = 1$ e $Y(0) = 0$, com passo de integração $h = 0.1$. 56
- A.2 Onda- partícula, $N = M = 1$, $\omega = 0$, $\beta = 1$, $\kappa = 1$, $\varepsilon = 1$. Comparação dos resultados do integrador simplético apresentado (tracejado, em preto) com resultados obtidos via Runge-Kutta (cheio, em vermelho). 57
- A.3 Erro relativo na energia e no momento total associado a resolução do sistema onda-partícula utilizando o integrador simplético proposto. 57

Lista de Tabelas

- 4.1 Análise de convergência, mapa tangente. A tabela relaciona o tamanho do passo utilizado no processo de integração com o valor medido do expoente. Os espaços completados com — correspondem aos casos onde os resultados envolvendo fração e/ou o logaritmo não estão definidos. 45
- 4.2 Análise de convergência, método dos clones. A tabela relaciona o tamanho do passo utilizado no processo de integração com o valor medido do expoente. Os espaços completados com — correspondem aos casos onde os resultados envolvendo a fração e/ou o logaritmo não estão definidos. 46

CAPÍTULO 1

Introdução

Os sistemas dinâmicos podem ser definidos como objetos agrupados segundo regras de causa e efeito, cujas grandezas que os caracterizam possam variar no tempo. São exemplos a dinâmica de fluídos, nosso sistema nervoso, a população de uma dada região entre outros [31]. Embora os sistemas dinâmicos atualmente sejam tema de caráter interdisciplinar, ele possui como origem a matemática e a física. O surgimento do tópico coincidiu com o desenvolvimento da Física clássica e com a formulação do cálculo diferencial, atribuídos a Isaac Newton durante o século XVII [8]. Suas leis explicavam muitas coisas, *e.g.* a dinâmica das marés, o movimento e o formato dos planetas, problemas balísticos e tantos outros, e foram necessários anos até que todas as implicações fossem assimiladas.

Basicamente, dada uma condição inicial ao sistema dinâmico juntamente com as regras que o governam, todos os estados futuros podem ser exatamente calculados. Foi assim, por exemplo, que Newton resolveu o problema de dois corpos, analisando o movimento da terra ao redor do sol e chegando à Lei gravitacional do inverso do quadrado. Posteriormente, gerações de matemáticos e físicos se dedicaram a estender os métodos analíticos de Newton considerando três corpos, que curiosamente revelou-se uma tarefa muito mais complicada. Décadas depois, entendeu-se que seria impossível através de fórmulas explícitas a resolução do problema de três corpos, ou seja a tal questão não possui solução analítica.

No fim do século XIX, prosseguiram-se as investigações em mecânica celeste, agora sob um ponto de vista mais qualitativo, dado por Henri Poincaré que desenvolveu uma abordagem geométrica na análise das questões. Ao invés de tentar determinar a evolução das posições no espaço real, ele optou por uma descrição no espaço de fases, onde as dimensões passam a ser dadas pelas variáveis dinâmicas, e cuja evolução pode ser feita por um conjunto de equações diferenciais de primeira ordem. Ele investigava seções do espaço de fases, o que simplificava o problema e esclarecia vários pontos relativos à periodicidade dos sistemas. Poincaré foi

também a primeira pessoa a notar a possibilidade de caos, ele observou que sistemas determinísticos que exibiam comportamento aperiódico podiam apresentar uma dependência sensível às condições iniciais, o que dificultava bastante a realização de previsões para tempos maiores [41].

No entanto, um estudo mais completo do assunto só foi ocorrer na segunda metade do século XX, com a publicação de Edward Lorenz intitulada *Deterministic nonperiodic flow* [24]. Ele estudou um modelo simplificado baseado no problema de rolos de convecção na atmosfera, cujas equações devido à simplicidade são abordadas até em cursos introdutórios de dinâmica não linear [41] e cálculo [9], com a expressão dada por

$$\begin{aligned}\dot{x} &= \sigma(y - z), \\ \dot{y} &= rx - y - xz, \\ \dot{z} &= xy - bz.\end{aligned}\tag{1.1}$$

Lorenz observou que assim como o problema dos três corpos abordado por Poincaré o conjunto de equações (1.1) não admitia solução analítica, as trajetórias não atingiam o equilíbrio ou assumiam movimento periódico, ao invés disso prosseguiam oscilando de forma irregular em uma órbita aperiódica. Após uma análise computacional detalhada, Lorenz mostrou que o modelo por ele estudado apresentava uma super sensibilidade à variações nas condições iniciais, definindo a principal característica dos sistemas caóticos (figura 1.1). Do ponto de vista prático, significa que pequenas incertezas nas condições iniciais implicavam em divergências exponencialmente maiores nas trajetórias finais, dando aos sistemas caóticos uma impressão de aleatoriedade.

Vale ressaltar a crucial importância dos computadores no desenvolvimento da teoria do caos e dinâmica não linear. A utilização dos transistores [7] e a substituição da linguagem de máquina ou *assembly* por linguagens de alto nível, ocorridas durante a década de 1950, proporcionaram aos dispositivos uma performance nunca experimentada antes. Os custos dos grandes centros computacionais com os programadores, pelo número de profissionais que a atividade demandava e pelo alto nível de especialização, motivaram o desenvolvimento de linguagens como FORTRAN [5], na qual os interpretadores, cuja a execução dispensava o uso de um computador específico, convertiam expressões algébricas em instruções de máquina. As

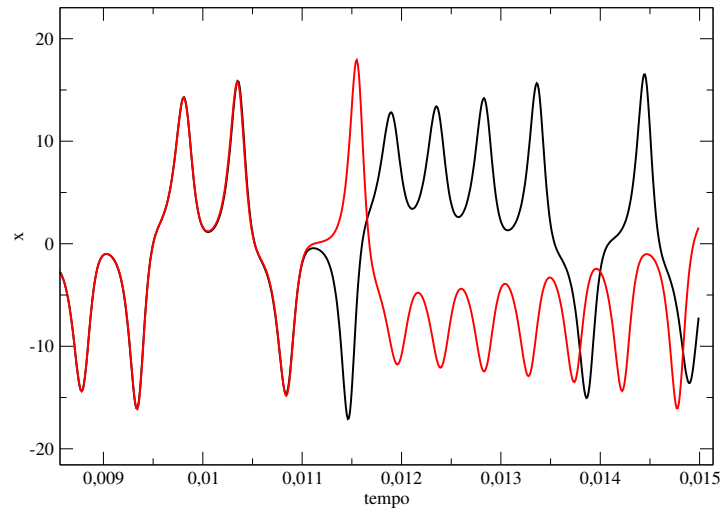


Figura 1.1 Equação de Lorenz, $b = 8/3$, $r = 28$, $\sigma = 10$. Soluções cujas condições iniciais em x diferem uma da outra em 10^{-2} . Resultados obtidos via Runge-Kutta de 4^o ordem, com passo de tempo $h = 10^{-5}$ e o tempo definido como $t = i \times h$, onde i é o número de iterações realizadas pelo integrador.

inovações culminaram na libertação dos computadores da exclusividade dos programadores, dando oportunidade a físicos, matemáticos e engenheiros de reinventar maneiras de analisar, modelar, simular e calcular utilizando os novos apetrechos.

No caso dos sistemas dinâmicos, a computação gráfica propiciou formas de visualizar como se manifesta a não linearidade das equações. Foi possível, então, intuir a respeito dessas soluções e entender melhor o papel dos parâmetros do sistema. Já os integradores numéricos, processados nos computadores, deram o refinamento necessário para perceber a super sensibilidade dos sistemas caóticos.

Alinhada ao contexto exposto, essa tese analisa técnicas computacionais utilizadas no estudo do caos aplicadas a modelos Hamiltonianos de longo alcance.

Os sistemas de longo alcance são conhecidos por possuírem propriedades de relaxação peculiares, identificadas originalmente em astrofísica, relevantes também em física de plasmas [35], dinâmica de vórtices em hidrodinâmica [44] e laser de elétrons livres[3]. Comparados aos sistemas de curto alcance, pode-se dizer que os de longo alcance evoluem segundo uma escala de tempo diferente. Primeiramente, há uma relaxação violenta e em seguida o sistema fica preso em estados fora do equilíbrio, usualmente conhecidos como quase estacionários, no

qual permanece por um longo período de tempo. Isso foi sugerido por Lynden-Bell [25], que introduziu uma abordagem não colisional ao explicar a rápida evolução das galáxias aos estados quase estacionários e finalmente seguir lentamente para o equilíbrio.

No entanto, grande parte dos avanços no entendimento de propriedades dos sistemas governados por forças de longo alcance foram obtidos por meio de simulações numéricas. Nos últimos anos, os *toy models* ganham cada vez mais visibilidade dentro da comunidade de Física estatística. Tratam-se de modelos Hamiltonianos reduzidos que, apesar das simplificações, manifestam de forma fiel propriedades de não equilíbrio atribuídas aos sistemas de longo alcance. Dentre eles vale citar o *ring model*, modelo onda-partícula, o Hamiltoniano de campo médio (*Hamiltonian Mean Field*, ou simplesmente HMF) [13] e o modelo de folhas autogravitantes [46].

No decorrer deste projeto, foram usados dois métodos numéricos diferentes, o mapa tangente e o método dos clones [39, 22] a fim de investigar o comportamento de protótipos em 1-D, i.e o potencial cosseno padrão do Hamiltoniano de campo médio e o modelo de folhas autogravitantes. As investigações tem base no cálculo dos expoentes de Lyapunov, que é um indicador importante de caos, e em como eles escalam com o tamanho do sistema.

O capítulo 2 trata essencialmente dos sistemas de longo alcance, contexto no qual escolhemos estudar numericamente os expoentes de Lyapunov. Serão apresentadas definições e propriedades que os distinguem dos sistemas de curto alcance. Em particular, nos aprofundaremos no HMF e no modelo das folhas autogravitantes, interpretando o Hamiltoniano e as regras de movimento, estabelecendo condições iniciais, discutindo as transições de fases, resultados analíticos e numéricos.

No capítulo 3, de fundamental importância, descreveremos o conjunto de procedimentos que compõe o método estudado e as ferramentas utilizadas, tais como integradores, processo de ortonormalização, análises de convergência entre outras.

O capítulo 4 foi reservado para apresentação dos resultados. Primeiramente, discute-se sob que condições foram obtidas as medidas e calcula-se o erro associado. Em seguida extraímos valores que confirmam propriedades fundamentais dos sistemas Hamiltonianos. Seguro do quanto os algoritmos desenvolvidos são robustos, o leitor pode avançar para os resultados

principais, que envolvem os sistemas de longo alcance. Concluindo, dois diferentes métodos são confrontados, por meio da avaliação de suas performances no cálculo dos expoentes de Lyapunov.

No capítulo 5 são dadas as considerações finais. Um breve apanhado de tudo que foi desenvolvido juntamente com as expectativas de trabalhos futuros.

Nos apêndices foram incluídos pseudocódigos dos algoritmos desenvolvidos e um exemplo, utilizando o sistema onda-partícula, da aplicação do integrador simplético utilizado.

Sistemas com interação de longo alcance

Uma interação de longo alcance é aquela cujo o potencial, para grandes distâncias r , decai com $r^{-\alpha}$, sendo d a dimensão espacial e $\alpha \leq d$ [12]. Tais sistemas são conhecidos por possuírem certas características incomuns se comparados aos de curto alcance. Dizemos que as interações de longo alcance evoluem segundo uma escala de tempo diferente, já que a relaxação para o equilíbrio de Boltzmann-Gibbs [38] ocorre muito lentamente.

Portanto, é necessário uma abordagem fora do equilíbrio para entender os detalhes da permanência dos sistemas de longo alcance nos estados quase-estacionários, conhecidos assim por apresentarem características do equilíbrio, como oscilações de variáveis dinâmicas em torno de um dado valor, com distribuições não Gaussianas das velocidades. Sabe-se hoje que os estados quase-estacionários dependem das condições iniciais [14] e seu tempo de duração diverge com o tamanho do sistema [40].

Se $\alpha \leq d$, critério que define os sistemas de longo alcance, implica no crescimento superlinear da energia potencial por partícula com o volume, que no limite termodinâmico resulta em divergência, permitindo a não aditividade da energia, calor específico negativo e inequivalência dos *ensembles* [10].

São exemplos de longo alcance os sistemas autogravitantes, os plasmas carregados e os Hamiltonianos de campo médio. Em comum, eles deixam subitamente as condições iniciais fora do equilíbrio através da relaxação violenta, em seguida assumem por um longo período o regime quase-estacionário, até finalmente atingirem de fato o equilíbrio termodinâmico.

Neste capítulo trataremos de conceitos importantes para o entendimento das interações de longo alcance, introduzindo de forma detalhada o Hamiltoniano de campo médio e o modelo de folhas autogravitantes, sistemas escolhidos para o estudo dos expoentes de Lyapunov.

2.1 Hamiltoniano de campo médio

O modelo Hamiltoniano de campo médio (com sigla HMF advinda do inglês, *Hamiltonian mean field*) é um protótipo para sistemas Hamiltonianos com potencial de longo alcance [13]. Tratam-se de N pontos de massa idênticos se movendo em um círculo de raio unitário, interagindo segundo um potencial de alcance infinito, com duas quantidades a serem conservadas, energia e momento total, cuja a função Hamiltoniana é dada por:

$$H = \sum_{i=1}^N \frac{p_i^2}{2} + \frac{1}{2N} \sum_{i,j=1}^N [1 - \cos(\theta_i - \theta_j)]. \quad (2.1)$$

O primeiro termo diz respeito à energia cinética, a expressão seguinte corresponde a um potencial de alcance infinito. O sinal positivo do potencial implica em uma dinâmica atrativa, a divisão por 2 acontece porque as interações entre as partículas são contadas duas vezes (não há restrições nos somatórios do potencial) e o N no denominador garante a extensividade na energia do sistema. A Hamiltoniana (2.1) corresponde ao modelo XY de Heisenberg¹ com alcance infinito, e por isso é conhecido também como modelo XY de campo médio².

O par de variáveis conjugadas é representado por

- θ_i , coordenada que indica a posição angular da i -ésima partícula no círculo e obedece condições de contorno periódicas, $-\pi > \theta_i \geq \pi$;
- p_i , momento conjugado.

A mecânica estatística do equilíbrio prevê para o HMF uma transição de fases de segunda ordem, cujo parâmetro de ordem é o vetor magnetização, definido como

$$\mathbf{M} = \frac{1}{N} \sum_{i=1}^N (\cos \theta_i, \sin \theta_i) = (M_x, M_y). \quad (2.2)$$

¹Deve-se entretanto ressaltar que o modelo XY de Heisenberg é quântico enquanto o modelo Hamiltoniano de campo médio, abordado neste trabalho, é clássico.

²No contexto do sistema XY, a função (2.1) corresponde ao modelo ferromagnético. A dinâmica antiferromagnético pode ser obtida atribuindo sinal negativo ao potencial, que se torna repulsivo. As implicações do caso antiferromagnético, como por exemplo estado fundamental e as soluções no equilíbrio, diferem do caso abordado aqui. Para mais informações consultar referência [2].

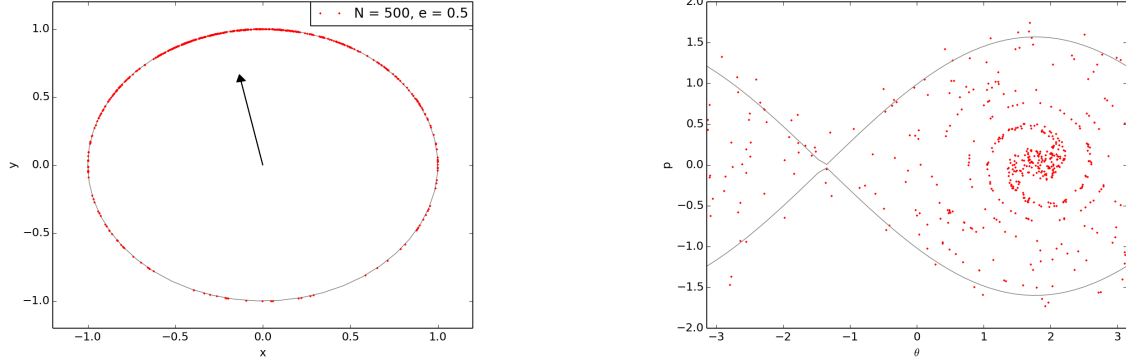


Figura 2.1 HMF, $N = 500$, $e = 0.5$. Na esquerda temos o espaço real, com pontos de massa interagindo e se movimentando no círculo. O tamanho do vetor nos dá ideia do módulo de \mathbf{M} enquanto a direção está relacionada a fase ϕ . A figura da direita exibe o espaço de fases, com o contorno claro representando a separatriz instantânea. tempo total $t = 1000$, sendo o tempo definido como o número de iterações vezes o tamanho do passo do integrador.

O módulo, $M = \sqrt{M_x^2 + M_y^2}$, mede o nível de aglomeração das partículas e a partir dele se pode definir um ângulo de fase dado por $\phi = \arctan(M_y/M_x)$.

No limite $N \rightarrow \infty$, a dinâmica do modelo HMF pode ser descrita por uma função distribuição $f_N(\theta_1, p_1, \dots, \theta_N, p_N)$, onde o estado mais provável no equilíbrio corresponde ao máximo da entropia de Gibbs [36],

$$S = \int f_N \ln f_N d\theta_1 \dots \theta_N dp_1 \dots dp_N, \quad (2.3)$$

satisfazendo os vínculos de normalização e energia média:

$$\int f_N d\theta_1 \dots \theta_N dp_1 \dots dp_N = 1, \quad (2.4)$$

$$\int H_N(\theta_1, p_1, \dots, \theta_N, p_N) f_N d\theta_1 \dots \theta_N dp_1 \dots dp_N = E, \quad (2.5)$$

obteremos uma expressão autoconsistente da magnetização no equilíbrio dada por

$$M = \frac{I_1(\beta M)}{I_0(\beta M)}. \quad (2.6)$$

Onde I_n é a função de Bessel modificada de ordem n e β o inverso da temperatura [34]. A distribuição representada por (2.6) possui uma energia crítica na qual para $e < e_c$ teremos so-

luções com $M \neq 0$ enquanto para $e \geq e_c$ as soluções serão necessariamente homogêneas, que implicam em $M = 0$ (ver figura 2.6).

As equações de movimento, obtidas através das equações de Hamilton, podem ser apresentadas na forma

$$\begin{aligned}\dot{\theta}_i &= \frac{\partial H}{\partial p_i} = p_i \\ \dot{p}_i &= -\frac{\partial H}{\partial \theta_i} = -M \sin(\theta_i - \phi).\end{aligned}\tag{2.7}$$

A dinâmica é regida pelos valores instantâneos das variáveis de campo médio M e ϕ . Quer dizer que a força sofrida por cada partícula depende de suas posições e de como todas as outras encontram-se arranjadas no círculo.

De acordo com as soluções de (2.7), o movimento pode ser interpretado como o de N pêndulos totalmente acoplados, com amplitude instantânea M e cuja separatriz, que correspondente à órbita limite onde a energia total do sistema coincide com o valor máximo do potencial (ver figura (2.1)), varia no tempo obedecendo a expressão

$$p = \pm \sqrt{2M[1 + \cos(\theta - \phi)]}.\tag{2.8}$$

Comparando a energia das partícula,

$$e_i = \frac{p_i^2}{2} + 1 - M \cos(\theta_i - \phi),\tag{2.9}$$

com a energia da separatriz, $E_s = 1 + M$, é possível distinguir regiões no espaço de fases, figura (2.2), onde:

- partículas estão confinadas na separatriz;
- partículas se encontram fora da separatriz com $p > 0$;
- partículas se encontram fora da separatriz com $p < 0$.

Dependendo dos parâmetros fornecidos se observa, após tempo suficiente, a formação de um aglomerado constituído por partículas menos energéticas.

Soluções numéricas mostram que uma configuração inicial fora do equilíbrio evolui rapidamente através de uma relaxação violenta para um estado quase-estacionário, cujo tempo de

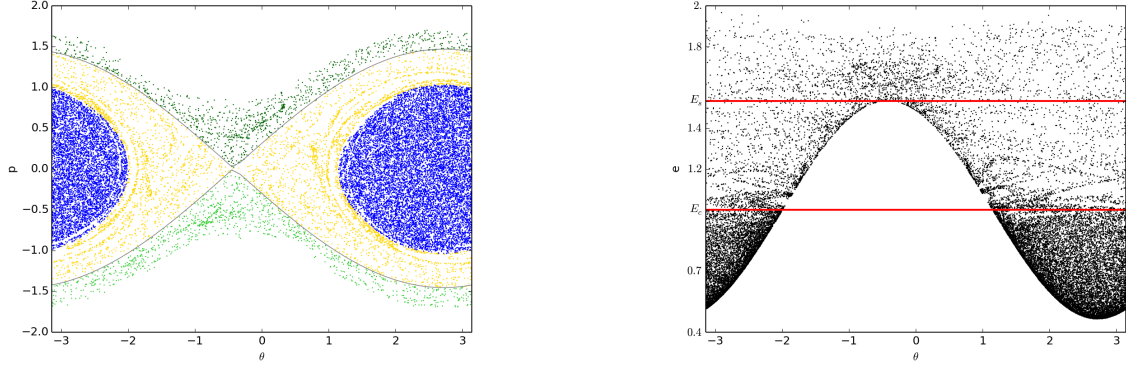


Figura 2.2 Utilizando a energia como critério de caracterização das partículas, distinguem-se três diferentes regiões no espaço de fases. O aglomerado em azul é formado pelas partículas de menor energia, o grupo de partículas marcadas de amarelo possui energia $E_c < e < E_s$ e finalmente o verde representa as partículas com energia superior à da separatriz, o verde escuro engloba as partículas com $p > 0$ enquanto o tom claro diz respeito as partículas com $p < 0$.

relaxação para o equilíbrio diverge com o número de partículas [4], ver figuras (2.3) e (2.4). A relaxação violenta é marcada pelo movimento desordenado das partículas o que ocasiona variações abruptas na magnetização [34]. O estado quase-estacionário ocorre logo após a relaxação violenta. Nesse regime, o sistema assume propriedade do equilíbrio, no entanto o que o caracteriza é a distribuição não Gaussiana das velocidades [6].

A distribuição *waterbag* [37] foi utilizada como condição inicial, figura (2.5), e é definida como

$$f(\theta, p, t = 0) = \begin{cases} 1/(\Delta\theta\Delta p), & \text{se } 0 < \theta_i \leq \Delta\theta, \quad |p_i| \leq \Delta p/2 \\ 0, & \text{qualquer outro caso.} \end{cases} \quad (2.10)$$

Portanto, a magnetização inicial e a energia por partícula, $e = E/N$, podem ser escritas como

$$M_0 = [(1 - \cos \Delta\theta)^2 + (\sin \Delta\theta)^2]^{1/2}, \quad (2.11)$$

$$e = \frac{\Delta p^2}{12} + \frac{(1 - M_0^2)}{2}, \quad (2.12)$$

obtidas trocando a definição discreta da magnetização, (2.2), pela contínua³, em $t = 0$.

³ $M_x = \int d\theta dp f(\theta, p, t) \cos(\theta), M_y = \int d\theta dp f(\theta, p, t) \sin(\theta)$

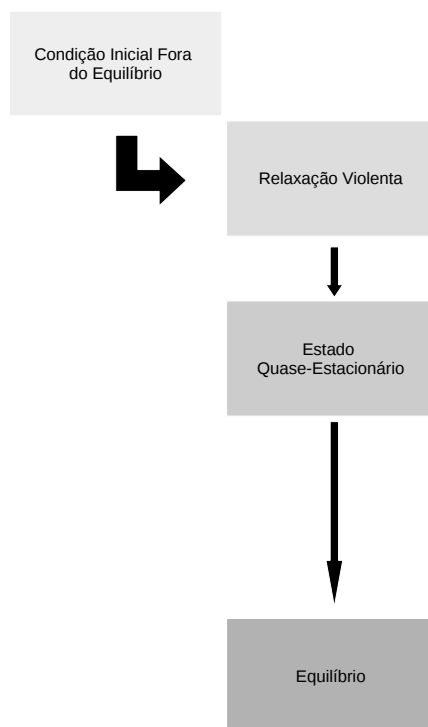


Figura 2.3 Fluxograma resumindo a evolução dos sistemas de longo alcance

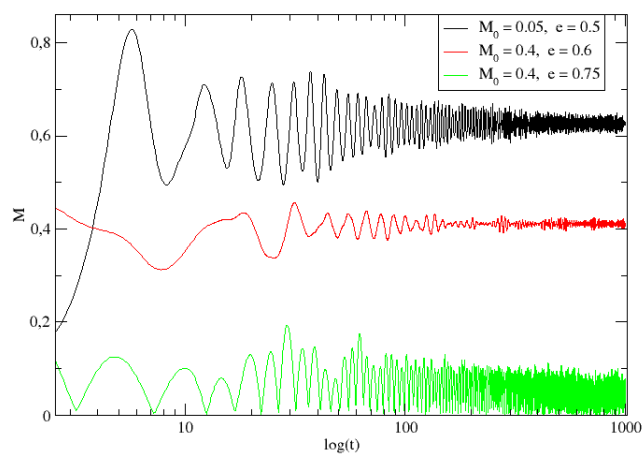


Figura 2.4 HMF, $N = 20000$. Comportamento do módulo do vetor magnetização para diferentes valores da energia por partícula.

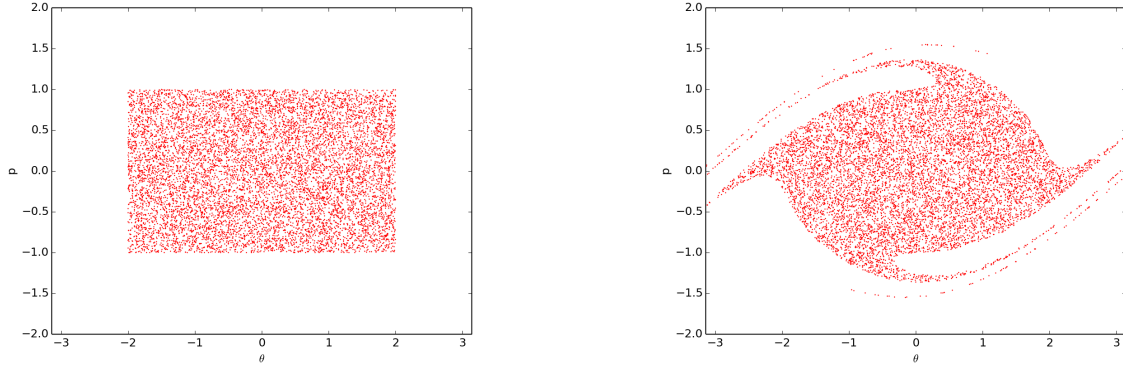


Figura 2.5 HMF, $N = 1000$, $e = 0.56$, $M_0 = 0.46$. Representação da distribuição *waterbag*, sendo utilizada como condição inicial, seguida da configuração do sistema instantes depois. Para fins didáticos os limites de $\Delta\theta$ são diferentes daqueles estabelecidos na equação (2.10), o que não compromete os resultados. Na direita temos o espaço de fases instantes depois de iniciada a dinâmica.

Conforme já mencionado, soluções exatas no equilíbrio mostram a existência de uma energia crítica que separa regimes distintos no HMF. Soluções no *ensemble* canônico, e microcanônico [2], revelam que a transição ocorre em $e_c = 3/4$. No caso subcrítico, $e < e_c$, observa-se a formação de um aglomerado, com a maioria das partículas ocupando uma região bem definida do círculo e outras poucas com energia suficiente para cruzá-lo ($M \neq 0$). No supercrítico, $e > e_c$, o sistema evolui para uma fase homogênea ($M = 0$), a largura da separatriz vai a zero e a maioria das partículas passam a ter acesso a todo o anel.

A figura (2.6) é uma representação numérica da transição de fases no HMF. Os resultados gerados por um integrador simplético⁴ exibem valores da média da magnetização para diferentes energias extraídos após um período suficientemente grande de integração (para que o sistema assumo o equilíbrio) e estão em harmonia com a teoria. Apesar da dificuldade em estabelecer rigorosamente o instante que o sistema deixa o quase-equilíbrio, devido à grandes oscilações na distribuição dos momentos.

A curtose expressada como $\kappa = m_4(\mu)/\sigma^4$, onde $m_4(\mu)$ e σ^4 são respectivamente o quarto momento e o quadrado da variância, mede o nível de achatamento do pico das distribuições,

⁴Os detalhes do procedimento de integração se encontram nas seções que descrevem o modelo e nos apêndices desse trabalho.

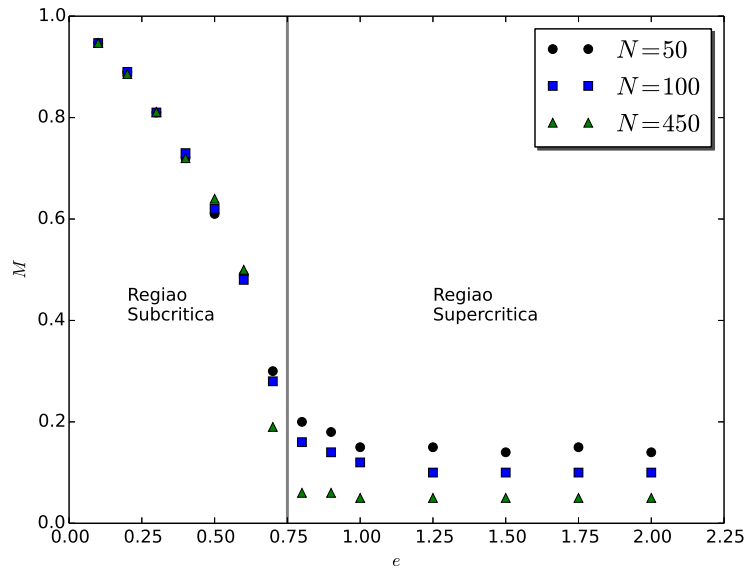


Figura 2.6 HMF, $N = 40, 80, 450$. Parâmetro de ordem M versus energia por partícula e , gráfico construído a partir da média do módulo da magnetização no equilíbrio. A reta vertical destacada em cinza distingue o regime de aglomeração, situado na região subcrítica, do homogêneo, na região supercrítica, separados pela transição de fases que ocorre em $e = 3/4$.

sendo o valor $\kappa = 3$ associado às distribuições normais. Ela é utilizada como critério que estabelece se um dado sistema se encontra no equilíbrio, através da análise das velocidades. As figuras (2.7) mostram a distribuição da curtose durante o regime no qual foram extraídos os valores para construção do gráfico (2.6) juntamente com a distribuição dos momentos, comparando-a com uma distribuição Gaussiana de mesma média e variância.

2.2 Modelo de folhas autogravitantes

O modelo unidimensional de folhas autogravitantes é baseado na dinâmica de N superfícies infinitas de massa situadas no plano $Y-Z$ se movendo no eixo X [29]. As densidades são todas uniformes e idênticas, considerando que as folhas gravitacionais sejam capazes de atravessar umas as outras. A função Hamiltoniana que expressa tal situação é

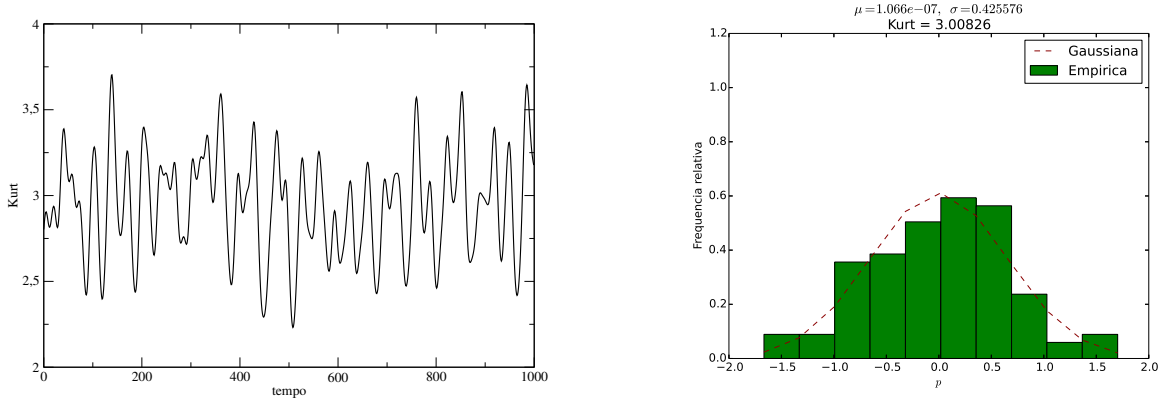


Figura 2.7 HMF, $N = 100$, $e = 0.6$. A esquerda temos o valor da curtose, dada pela razão entre o quarto momento pelo quadrado da variância, versus o tempo. O gráfico da direita mostra a distribuição dos momentos, em um dado instante supostamente no equilíbrio, comparando-a com uma distribuição Gaussiana com a mesma média e variância

$$H' = \sum_{i=1}^N \frac{P_i^2}{2m_i} + 2\pi G \sum_{i=1}^N \sum_{i<j}^N m_i m_j |X_j - X_i|, \quad (2.13)$$

onde G é a constante gravitacional, m_i a massa, X_i indica a posição e P_i o momento da i -ésima folha. Por questões de simplificação, a partir de agora, vamos nos referir as folhas como partículas e os cruzamentos como colisões. Vale lembrar que o caso abordado trata $m_i = m$ para qualquer i .

Para garantir a extensividade do sistema faremos a seguinte transformação canônica:

$$X = \frac{x}{a}, \quad P = ap, \quad a = (2\pi GN)^{1/3} m. \quad (2.14)$$

O resultado é $H' = \frac{a^2}{m} \left[\sum_{i=1}^N p_i^2 + \frac{1}{N} \sum_{i=1}^N \sum_{i<j}^N |x_j - x_i| \right]$. Uma mudança na escala de tempo nos permite escrever finalmente

$$H = \sum_{i=1}^N \frac{p_i^2}{2} + \frac{1}{2N} \sum_{i,j=1}^N |x_j - x_i|, \quad (2.15)$$

cujas mudanças nos índices do somatório duplo são compensadas pelo termo $1/2$ que aparece multiplicando o potencial.

As equações de Hamilton para a função (2.15) nos fornecem as regras de movimento do sistema,

$$\begin{aligned}\dot{x}_i &= p_i \\ \dot{p}_i &= -\frac{1}{N} \sum_{j=1}^N \frac{x_i - x_j}{|x_i - x_j|}.\end{aligned}\quad (2.16)$$

Observa-se a presença de uma força de atração mútua entre as partículas que independe da distância de separação entre elas. Temos um movimento uniformemente variado até que haja uma colisão, em seguida as partículas envolvidas experimentam uma variação descontínua da força que volta a ser constante imediatamente em seguida. Com base na discussão podemos reescrever as equações de movimento (2.16) como

$$\begin{aligned}\dot{x}_i &= p_i \\ \dot{p}_i &= -\frac{1}{N} \sum_{j \neq i}^N [2\Theta(x_i - x_j) - 1],\end{aligned}\quad (2.17)$$

sendo Θ a função Heaviside.

A expressão da derivada do momento pode ser escrita ainda de uma terceira maneira, $\dot{p}_i = \frac{1}{N}(N_+^i - N_-^i)$, mostrando que a força sofrida por um dado elemento depende apenas da diferença entre o número de partícula à direita, N_+^i , e à esquerda deste, N_-^i .

A condição inicial escolhida mais uma vez é a *waterbag*, que dessa vez obedece a função distribuição [46]

$$f(\theta, p, t = 0) = \begin{cases} 1/(4\Delta x \Delta p), & \text{se } x_i < \Delta x, \quad p_i < \Delta p \\ 0, & \text{qualquer outro caso.} \end{cases}\quad (2.18)$$

A energia média do sistema pode ser relacionada com a condição inicial através da expressão $e = \frac{\Delta p^2}{6} + \frac{\Delta x}{3}$.

O sistema de folhas autogravitantes originalmente é um modelo astrofísico com relevância também em física de plasma. Apesar da simplicidade, trata-se de um sistema unidimensional sob o governo de forças constantes⁵, as propriedades do quase-equilíbrio e a relaxação problemática para o equilíbrio ainda é objeto de muita discussão [20].

O foco deste trabalho é a abordagem numérica e o estudo características caóticas do sistema através do cálculo dos expoentes de Lyapunov.

⁵Que não tem dependência com as distâncias entre os pares.

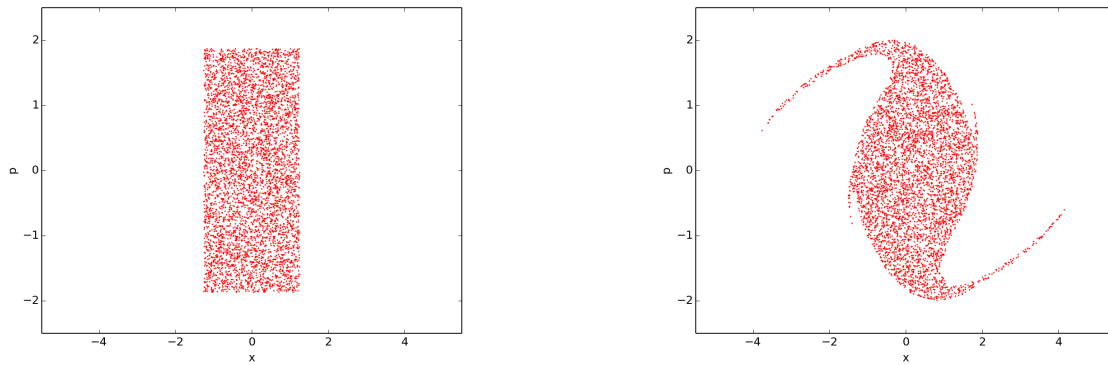


Figura 2.8 Folhas autogravitantes, $N = 1000$, $e = 1$. Representação da distribuição *waterbag*, sendo utilizada como condição inicial, seguida da configuração do sistema instantes depois.

Métodos para o cálculo dos expoentes de Lyapunov

3.1 Os Expoentes de Lyapunov

Sistemas caóticos apresentam super sensibilidade a variações nas condições iniciais, isso significa que duas trajetórias partindo de regiões muito próximas do espaço de fases irão rapidamente divergir. Em termos práticos isso nos torna incapazes de realizar previsões nesse tipo de dinâmica uma vez que o erro associado a uma medida nos impossibilita definir exatamente as condições iniciais e isso é suficiente para gerar soluções com trajetórias completamente distintas.

A divergência exponencial entre trajetórias infinitesimalmente próximas pode ser representada pela expressão $|\zeta(t)| \sim |\zeta_0|e^{\lambda t}$, onde ζ é o vetor da diferença entre as trajetórias vizinhas. Sendo o expoente de Lyapunov uma estimativa da taxa de divergência ou convergência dessas soluções podemos defini-lo como

$$\lambda = \lim_{\substack{t \rightarrow \infty, \\ |\zeta_0| \rightarrow 0}} \frac{1}{t} \ln \frac{|\zeta(t)|}{|\zeta_0|}. \quad (3.1)$$

Em um problema n -dimensional teríamos n expoentes, cada um referente a uma dada auto-direção do sistema. O conjunto de todos eles é denominado espectro de Lyapunov e pode ser organizado da seguinte maneira

$$\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n. \quad (3.2)$$

Basta que o maior entre todos, λ_1 por definição, seja positivo e podemos falar em regimes caóticos. Como é de se esperar, valores positivos estão associados a expansões no fluxo das soluções enquanto expoentes negativos são relativos as contrações.

Vamos considerar que a dinâmica em questão seja regida pelo sistema de equações diferen-

ciais a seguir

$$\begin{aligned}\dot{x}_1 &= f_1(x_1, \dots, x_n) \\ &\vdots \\ \dot{x}_n &= f_n(x_1, \dots, x_n),\end{aligned}\tag{3.3}$$

ou sob forma mais compacta

$$\dot{\mathbf{x}}(t) = \mathbf{F}(\mathbf{x}(t)).\tag{3.4}$$

O problema de valor inicial acima é descrito como um sistema de equações diferenciais, onde $\mathbf{x} = (x_1, x_2, \dots, x_n)$, de primeira ordem autônomo, já que a função vetorial $\mathbf{F}(\mathbf{x}(t))$ não depende explicitamente do tempo. A solução $\mathbf{x}(t)$ deve obedecer as equações (3.4), as trajetórias para cada condição inicial e o teorema da existência e unicidade garantem um fluxo de soluções no espaço de fases. Nos interessa a trajetória partindo de um dado $\mathbf{x}(0)$ juntamente com aquelas cujas condições iniciais diferem de um valor infinitesimal $\delta\mathbf{x}(0)$. A evolução da diferença $\delta\mathbf{x}(t)$ em torno de \mathbf{x}^* pode ser escrita como

$$\delta\dot{\mathbf{x}}(t) = \mathbf{F}(\mathbf{x}(t) + \delta\mathbf{x}(t)) - \mathbf{F}(\mathbf{x}(t)) \approx \left. \frac{\partial \mathbf{F}}{\partial \mathbf{x}} \right|_{\mathbf{x}=\mathbf{x}^*} \delta\mathbf{x}(t),\tag{3.5}$$

que por sua vez é equivalente a

$$\delta\dot{\mathbf{x}} = J(\mathbf{x}, t)\delta\mathbf{x}.\tag{3.6}$$

J é a matriz contendo todas as derivadas parciais de primeira ordem da função vetorial $\mathbf{F}(\mathbf{x}(t))$. Tal matriz, conhecida como Jacobiana, nos fornece os coeficientes do sistema linear de equações (3.6), que fisicamente, significa a evolução da diferença entre órbitas de condição iniciais próximas, na vizinhança de \mathbf{x}^* .

Por meio da solução do sistema linear de equações (3.6), dada por

$$\delta\mathbf{x}(t) = H(\mathbf{x}_0, t)\delta\mathbf{x}(0),\tag{3.7}$$

onde $H = \exp(\int_0^t dt' J(\mathbf{x}, t'))$, juntamente com a definição (3.1) é possível obter estimativas, tanto analíticas¹ quanto numéricas, para o valor dos expoentes de Lyapunov. Esse trabalho

¹propriedades geométricas do espaço de fases são utilizadas como argumentação teórica na avaliação analítica dos expoentes de Lyapunov, para mais detalhes ver referência [17]

tem foco nos métodos numéricos e no atual capítulo serão apresentados dois procedimentos computacionais, um que utiliza equações linearizadas e um outro que é baseado na evolução de cópias perturbadas do sistema. O intuito é confrontar o desempenho dos métodos no cálculo dos expoentes de Lyapunov para sistemas de longo alcance, em particular o Hamiltoniano de campo médio, modelo de folhas autogravitantes e futuramente para o sistema de interação onda-partícula [16].

3.2 Descrição dos Métodos

O procedimento computacional para o cálculo dos expoentes de Lyapunov é baseado na evolução das variações, conforme mostra definição (3.1). Isso pode ser feito por meio da resolução das equações linearizadas (3.6) ou considerando o vetor diferença no espaço de fases entre uma trajetória saindo de \mathbf{x}^* e uma outra $\mathbf{x}^* + \varepsilon\boldsymbol{\delta}$, com uma perturbação ε na direção $\boldsymbol{\delta}$.

A definição (3.1) considera $t \rightarrow \infty$, que do ponto de vista computacional apresenta as seguintes complicações:

- A dinâmica para $\delta\mathbf{x}(t)$ é não autônoma, conforme mostra a equação (3.6). Portanto, os expoentes de Lyapunov poderão ser extraídos de médias temporais, se assumirmos que $\mathbf{x}(t)$ seja um processo ergótico estatisticamente estacionário [32];
- O sistema de equações lineares (3.6) admite como solução uma função exponencial, significa que em tempos não tão grandes teremos $|\delta\mathbf{x}(t)|$ grande o suficiente para nos gerar problemas de extrapolação de memória;
- trajetórias infinitesimalmente próximas podem se distanciar tão rapidamente que continuar considerando-as vizinhas passa a ser inadequado, sendo inapropriado também o uso da definição (3.1) para estimar os expoentes.

O problema é contornado através de intervenções periódicas no sistema, realizadas durante o processo de integração das equações diferenciais [28]. O procedimento se baseia na aplicação sucessiva de ortonormalizações nas soluções $\delta\mathbf{x}(t)$. A normalização, no método tangente, evita

a divergência do vetor $\delta \mathbf{x}(t)$ (já no método dos clones, garante que as trajetórias continuem infinitesimalmente próximas) e a ortogonalização assegura o cálculo da taxa de divergência de direções linearmente independentes, resultando na obtenção de todo o espectro de Lyapunov.

As particularidades a respeito da implementação de cada método, mapa tangente e modelo dos clones, serão reportadas nas próximas seções.

3.2.1 Mapa Tangente

É possível obter uma estimativa numérica dos expoentes de Lyapunov por meio da construção do espaço tangente que por sua vez está associado a evolução temporal das equações linearizadas de movimento. De maneira sucinta quer dizer que nos basearemos em:

$$\begin{aligned} \mathbf{x} &\equiv (x_1, x_2, \dots, x_n), & \dot{\mathbf{x}}(t) &= \mathbf{F}(\mathbf{x}(t)) \\ \mathbf{w}_i &\equiv (\delta x_1, \delta x_2, \dots, \delta x_n), & \dot{\mathbf{w}}_i &= \mathbf{J}|_{\mathbf{x}=\mathbf{x}^*} \mathbf{w}_i, & 1 \leq i \leq n \end{aligned} \quad (3.8)$$

$$\lambda_i = \lim_{t \rightarrow \infty} \frac{1}{t} \ln |\mathbf{w}_i(t)|,$$

onde \mathbf{x} indica um ponto no espaço de fases n -dimensional enquanto \mathbf{w} está relacionado às equações linearizadas sendo o módulo utilizado na medida do espectro de Lyapunov.

Teremos um vetor \mathbf{x}_0 como condição inicial da dinâmica e uma base ortonormalizada $\{\boldsymbol{\delta}_{01}, \boldsymbol{\delta}_{02}, \dots, \boldsymbol{\delta}_{0n}\}$ referente às equações das variações. O procedimento consiste na integração, simultânea, de uma equação não linear e n equações linearizadas. Cada novo valor de \mathbf{x} implica em uma matriz Jacobiana diferente, isso significa que os parâmetro das equações linearizadas são reescritos a cada passo de tempo.

Sabe-se que $|\mathbf{w}(t)|$ diverge exponencialmente e por isso é necessário em intervalo regulares de tempo T aplicar o processo de ortonormalização de Gram–Schmidt [19] na base $\{\boldsymbol{\delta}_1, \boldsymbol{\delta}_2, \dots, \boldsymbol{\delta}_n\}$. O procedimento é repetido por K vezes e finalmente todo o espectro de Lyapunov é obtido na forma de média.

O algoritmo se resume a

- definir uma condição inicial \mathbf{x}_0 para o sistema não linear e n condições iniciais, por meio da base ortonormalizada $\{\boldsymbol{\delta}_{01}, \boldsymbol{\delta}_{02}, \dots, \boldsymbol{\delta}_{0n}\}$, para as equações linearizadas;

- integrar simultaneamente todos os sistemas de equações diferenciais, resultando em $\mathbf{x}_0 \rightarrow \mathbf{x}$ e $\{\boldsymbol{\delta}_{01}, \boldsymbol{\delta}_{02}, \dots, \boldsymbol{\delta}_{0n}\} \rightarrow \{\boldsymbol{\delta}_1, \boldsymbol{\delta}_2, \dots, \boldsymbol{\delta}_n\}$;
- a cada tempo T intervir nas soluções das equações linearizadas aplicando ortogonalização, que leva $\boldsymbol{\delta} \rightarrow \mathbf{v}$, seguida de normalização, $\mathbf{v} \rightarrow \mathbf{u}$;
- utilizando como condição inicial do sistema linear a base $\{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_n\}$, repetir o procedimento K vezes;
- finalmente obter o valor dos expoentes de Lyapunov através da média $\lambda_n = \frac{1}{KT} \sum_{k=1}^K \ln |\mathbf{v}_n^{(k)}(t)|$, sendo K grande o suficiente para que se observe a convergência do valor de λ .

3.2.2 Dinâmica de Clones

O método dos clones oferece uma abordagem alternativa, na qual a evolução das variações é definida como a diferença entre uma trajetória referência e uma outra cuja condição inicial é uma perturbação da primeira. Desta vez será levado em conta:

$$\begin{aligned}
 \mathbf{x} &\equiv (x_1, x_2, \dots, x_n), & \dot{\mathbf{x}}(t) &= \mathbf{F}(\mathbf{x}(t)) \\
 \mathbf{x}_{ci} &\equiv \mathbf{x} + \varepsilon \boldsymbol{\delta}_i, & \dot{\mathbf{x}}_{ci}(t) &= \mathbf{F}(\mathbf{x}_{ci}(t)), & 1 \leq i \leq n \\
 \mathbf{w}_i &= \mathbf{x} - \mathbf{x}_{ci} & & & (3.9)
 \end{aligned}$$

$$\lambda_i = \lim_{t \rightarrow \infty} \frac{1}{t} \ln \frac{|\mathbf{w}_i(t)|}{\varepsilon},$$

onde \mathbf{x} é, mais uma vez, um ponto no espaço de fases e assume o papel de sistema referência, já \mathbf{x}_{ci} nos fornece a evolução da dinâmica copiada cuja diferença \mathbf{w} está relacionada ao grau de divergência, ou convergência, entre trajetórias próximas e portanto usado na obtenção dos expoentes de Lyapunov.

Se denotarmos com \mathbf{x}_0 a condição inicial do sistema referência as órbitas clones devem ser iniciadas com $\mathbf{x}_0 + \boldsymbol{\delta}_i$, sendo $\{\boldsymbol{\delta}_1, \boldsymbol{\delta}_2, \dots, \boldsymbol{\delta}_n\} = \varepsilon \cdot \{\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_n\}$ com os \mathbf{e}_i formando uma base ortonormal e o escalar ε representando pequenas perturbações nas condições iniciais.

As intervenções, através do processo de ortonormalização, aplicadas durante a integração seguem no método dos clones. Além do problema de divergência aqui as trajetórias podem

se distanciar ao ponto que o termo "órbitas infinitesimalmente próximas" deixe de ser verdade e neste caso a grandeza computada não pode ser associada aos expoentes de Lyapunov. A correção é realizada de forma semelhante ao método do mapa tangente e na prática equivale a trazer as trajetórias das equações clone de volta as proximidades da órbita referência.

O algoritmo segue os seguintes passos

- Determinação das condições iniciais, \mathbf{x}_0 para a equação referência e $\mathbf{x}_0 + \boldsymbol{\delta}_n$ para a dinâmica clonada, sendo $\{\boldsymbol{\delta}_1, \boldsymbol{\delta}_2, \dots, \boldsymbol{\delta}_n\} = \varepsilon \cdot \{\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_n\}$ com os \mathbf{e}_n formando uma base ortonormal e o escalar ε definido como muito pequeno;
- resolução simultânea de todos os sistemas de equações diferenciais, que totaliza em uma dinâmica referência e n clonadas, resultando em $\mathbf{x}_0 \rightarrow \mathbf{x}$ e $\{\boldsymbol{\delta}_{01}, \boldsymbol{\delta}_{02}, \dots, \boldsymbol{\delta}_{0n}\} \rightarrow \{\boldsymbol{\delta}_1, \boldsymbol{\delta}_2, \dots, \boldsymbol{\delta}_n\}$;
- a cada tempo T intervir nas soluções das equações linearizadas aplicando ortogonalização, que leva $\boldsymbol{\delta} \rightarrow \mathbf{v}$, seguida de normalização, $\mathbf{v} \rightarrow \mathbf{u}$;
- redefinir condições iniciais das equações clones, como $\mathbf{x}_0 + \varepsilon \mathbf{u}_n$;
- repetir o processo K vezes;
- finalmente obter o valor dos expoentes de Lyapunov através da média $\lambda_n = \frac{1}{KT} \sum_{k=1}^K \ln \frac{|\mathbf{v}_n^{(k)}(t)|}{\varepsilon}$, sendo K grande o suficiente para que se observe a convergência do valor de λ .

3.3 Obtenção dos expoentes de Lyapunov para o HMF

Discutiremos a aplicabilidade do método tangente e dos clones no cálculo dos expoentes de Lyapunov para os sistemas de longo alcance, em particular o HMF e o modelo de folhas gravitacionais.

A implementação do método tangente exige a linearização das equações de movimento,

para o HMF temos

$$\begin{aligned}\delta\dot{\theta}_i &= \delta p_i \\ \delta\dot{p}_i &= -M \cos(\varphi - \theta_i) \delta\theta_i + \frac{1}{N} \sum_j \cos(\theta_j - \theta_i) \delta\theta_j.\end{aligned}\quad (3.10)$$

Denotando

$$\mathbf{w} \equiv (\delta\theta_1, \dots, \delta\theta_N, \delta p_1, \dots, \delta p_N) \quad (3.11)$$

a equação (3.10) adquire a forma

$$\dot{\mathbf{w}} = J(\mathbf{x}(t))\mathbf{w}. \quad (3.12)$$

Considerando N partículas, a matriz Jacobiana para o HMF será do tipo $2N \times 2N$ e obedecerá

$$J = \begin{pmatrix} 0 & I \\ \tilde{J} & 0 \end{pmatrix}. \quad (3.13)$$

Nesta representação cada elemento é uma matriz $N \times N$. A diagonal principal, na qual todos os elementos são nulos, juntamente com a matriz identidade, representada por I , são implicações diretas da linearização de um sistema Hamiltoniano do tipo $H = K(p) + V(\theta)$. A matriz \tilde{J} que segue a regra $\tilde{J}_{ij} = -\frac{\partial^2 V}{\partial\theta_i\partial\theta_j}$ dependerá da forma do potencial. No HMF

$$\begin{cases} \tilde{J}_{ii} = -\cos(\theta_i)M_x - \sin(\theta_i)M_y + \frac{1}{N} \\ \tilde{J}_{ij} = \frac{1}{N} \cos(\theta_i - \theta_j), & \text{se } i \neq j. \end{cases} \quad (3.14)$$

A representação (3.12) é útil no processo de integração.

Para obtenção dos expoentes devemos resolver o sistema não-linear juntamente com o linear. Para o HMF é recomendado o uso do integrador simplético *leapfrog*, pela simplicidade na implementação e eficiência no procedimento de integração, as variáveis de campo médio \mathbf{M} e ϕ facilitam bastante o cálculo da força sofrida por cada partícula.

O *leapfrog* consiste na atualização completa da velocidade entre duas meias atualizações da posição, ver figura (3.1), utilizando de forma direta as equações de Hamilton ².

²O *leapfrog* pode ser derivado de transformações canônicas infinitesimais no tempo, cuja a ordem do integrador é dada pela ordem do novo Hamiltoniano e as regras de evolução pelas equações que relacionam as variáveis antigas com as novas, para mais detalhes consulte a seção 3.6.

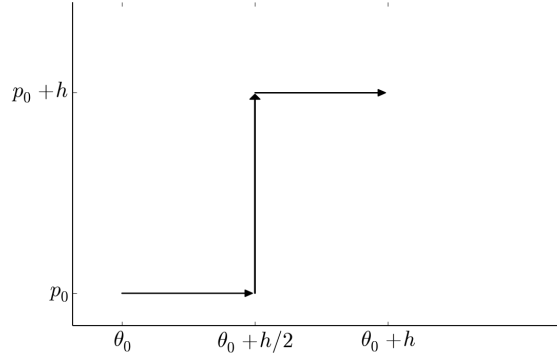


Figura 3.1 Esquema da construção do integrador simplético *leapfrog*. O procedimento é realizado através de dois passos, sendo o primeiro uma meia atualização de θ com p constante e o segundo passo dado por uma atualização completa de p e uma outra meia atualização em θ , respectivamente.

O cálculo de $|\mathbf{w}| = \sqrt{\sum_{i=1}^N (\delta\theta_i^2 + \delta p_i^2)}$ em intervalos regulares de tempo, a utilização da definição (3.1) e a ortonormalização do vetor \mathbf{w} resulta no método tangente para avaliação do espectro de Lyapunov para o HMF.

Na abordagem via método dos clones, conforme mostra a seção 3.2.2, como $\mathbf{w} = (\theta_i, p_i) - (\theta_{ic}, p_{ic})$ o *leapfrog* será aplicado diretamente as equações não lineares sendo desnecessário a construção da matriz Jacobiana (3.13).

O algoritmo do método de *leapfrog* desenvolvido em linguagem C encontra-se disponíveis no apêndice desse trabalho e as implicações sobre o uso do método tangente ou dos clone para determinação do espectro de Lyapunov no HMF serão discutidas na seção de resultados desse trabalho.

3.4 Obtenção dos expoentes de Lyapunov para o modelo de folhas autogravitantes

No problema das folhas autogravitantes, um estado no espaço de fases é representado pelo vetor $2N$ -dimensional $\mathbf{x} \equiv (x_1, p_1, \dots, x_N, p_N)$ que evolui segundo as equações de movimento

(2.17) que por sua vez dependerá das condições iniciais do problema.

A forma linearizada dessas equações, dada por

$$\begin{aligned}\Delta\dot{x}_i &= \Delta p_i \\ \Delta\dot{p}_i &= -\frac{1}{N} \sum_{j \neq i}^N (\Delta x_i - \Delta x_j) \delta(x_i - x_j),\end{aligned}\tag{3.15}$$

nos concede a evolução das variações aqui escritas como $\mathbf{w} \equiv (\Delta x_1, \Delta p_1, \dots, \Delta x_N, \Delta p_N)^3$ e usadas diretamente no procedimento numérico de avaliação dos expoentes de Lyapunov.

Sabe-se, segundo análises realizadas sobre esse modelo na seção 2.2, que o movimento das folhas é uniformemente variado até que ocorra uma colisão, nesse caso a força experimentada pelo par de partículas envolvidas será redefinida de maneira descontínua⁴. O sistema de equações (3.15) resume bem isso, a função delta garante variações na força apenas durante o processo de colisão enquanto a evolução das variações espaciais são uniformes.

Assim como o HMF, o problema das folhas gravitacionais pode ser resolvido numericamente utilizando o *leapfrog*. A singularidade na força, que ocorre quando $x_i = x_j$, pode ser ignorada já que é muito improvável do ponto de vista computacional que tais variáveis durante o procedimento sejam exatamente as mesmas, ou seja que apresentem todos os dígitos fornecidos por uma variável de precisão dupla iguais.

Contudo, as equações linearizadas (3.15) nos atentam que as colisões são realmente relevantes dentro da dinâmica das variações. Significa que para obter uma estimativa do grau de divergência entre soluções muito próximas necessitaremos de um integrador que tome nota das variáveis do sistema durante todos os processos de colisão, o que torna a utilização do *leapfrog* sob esse ponto de vista pouco eficiente.

Uma alternativa é a abordagem via *event-driven* [11], na qual associa a manifestação de um dado evento a mudanças no estado do sistema, que devem ocorrer instantaneamente e em pontos específicos do tempo.

No problema das folhas gravitacionais vamos utilizar as colisões⁵ como indicador de vari-

³A notação utilizada na representação das variações foi modificada, trocando o δ por Δ , para evitar confusão com a função $\delta(x)$, utilizada nas equações de movimento (3.15).

⁴A descontinuidade na força pode ser observada através da equação (2.16).

⁵Conforme foi esclarecido na seção 2.2, o termo partícula aqui diz respeito às folhas gravitacionais enquanto

ação na força sofrida pelas partículas. O procedimento consiste em [1]:

- (a) Identificar quando ocorrerá a próxima colisão;
- (b) mover todas as partículas, com força constante e diferente de zero, até tal instante;
- (c) redefinir o valor das forças sofridas pelo par de partículas envolvidas na colisão, integrar as equações linearizadas e retornar ao passo (a).

A integração no espaço tangente [30] se dá desde um tempo imediatamente antes (−) até um outro imediatamente depois (+) da colisão, que consideraremos acontecer instantaneamente. Colidindo uma partícula k com uma outra l teremos como solução das equações (3.15)

$$\begin{aligned}\Delta x_k^+ &= \Delta x_k^- \\ \Delta p_k^+ &= \Delta p_k^- - \frac{1}{N} \int_{t_c^-}^{t_c^+} (\Delta x_k - \Delta x_l) \delta(x_k - x_l) dt \\ &= \Delta p_k^- - \frac{1}{N} \frac{(\Delta x_k - \Delta x_l)_{t=t_c}}{|(\dot{x}_k - \dot{x}_l)_{t=t_c}|},\end{aligned}\tag{3.16}$$

onde o mesmo vale para l .

No período entre colisões teremos

$$\begin{aligned}\Delta x_i(t) &= t \Delta p_i(0) + \Delta x_i(0) \\ \Delta p_i(t) &= \Delta p_i(0).\end{aligned}\tag{3.17}$$

O cálculo de $|\mathbf{w}| = \sqrt{\sum_{i=1}^N (\Delta x_i^2 + \Delta p_i^2)}$ em intervalos regulares, a utilização da definição (3.1) e a ortonormalização do vetor \mathbf{w} resulta no método tangente para avaliação do espectro de Lyapunov para o modelo das folhas autogravitantes.

3.5 Análise de convergência

Uma conduta recomendada aqueles que fazem uso de técnicas numéricas é a verificação sistemática da implementação do método, monitorando o erro, por exemplo, que é dado por

$$u - \tilde{u}_h = Ch^p + O(h^{p+1}),\tag{3.18}$$

colisão faz referência ao cruzamento delas.

onde \tilde{u}_h é a aproximação utilizada na avaliação da função u . Sendo h suficientemente pequeno e C , independente de h , diferente de zero, podemos dizer que o método em questão é de ordem p . Tal consideração é importante e mostra a credibilidade, quanto maior p mais confiável é o procedimento.

Podemos obter a ordem de um integrador explicitamente, expandindo a função u e comparando com \tilde{u}_h conforme mostra (3.18), desde que u seja conhecido e o conjunto de procedimentos no qual \tilde{u}_h é construído permita a comparação. Nem sempre é possível, e uma segunda forma de conferir a precisão do processo é pela convergência de (3.18), dada uma sequência \tilde{u}_{h_1} , \tilde{u}_{h_2} e assim por diante, gerada a partir de diferentes tamanhos de h . Da equação (3.18) temos

$$\log |u - \tilde{u}_h| = \log |C| + p \log h + O(h). \quad (3.19)$$

Considerando que conhecemos u , então basta escrevermos $|u - \tilde{u}_h|$ como função de h e conforme o segundo membro da equação (3.19) sugere, o resultado deve ser uma reta cujo coeficiente angular é numericamente igual a ordem do método estudado.

É possível realizar estimativas sem o uso da função u . Nesse caso, compara-se a razão das soluções utilizando valores de h que obedecem uma progressão geométrica, por exemplo $\frac{\tilde{u}_h - \tilde{u}_{h/2}}{\tilde{u}_{h/2} - \tilde{u}_{h/4}}$, $\frac{\tilde{u}_{h/2} - \tilde{u}_{h/4}}{\tilde{u}_{h/4} - \tilde{u}_{h/8}}$ e assim por diante. Generalizando e com base na equação (3.19), teremos

$$\frac{\tilde{u}_h - \tilde{u}_{\alpha h}}{\tilde{u}_{\alpha h} - \tilde{u}_{\alpha^2 h}} = \frac{Ch^p - C(\alpha h)^p + O(h^{p+1})}{C(\alpha h)^p - C(\alpha^2 h)^p + O(h^{p+1})} = \alpha^{-p} + O(h), \quad 0 < \alpha < 1, \quad (3.20)$$

onde α representa a fração de h escolhida para estudar a convergência do método. Finalmente podemos escrever

$$p = - \lim_{h \rightarrow 0} \log \frac{\tilde{u}_h - \tilde{u}_{\alpha h}}{\tilde{u}_{\alpha h} - \tilde{u}_{\alpha^2 h}}. \quad (3.21)$$

3.6 Integradores numéricos

Ao deparar-se com um problema, o desejo inicial do cientista é a obtenção de uma expressão utilizando técnicas analíticas, como integração ou expansão, com o propósito de adquirir uma fórmula exata que descreva o fenômeno investigado. Infelizmente, na grande maioria das

questões, em engenharia e ciências, especialmente nos casos não-lineares, esses métodos ou não se aplicam ou tornam-se complicados demais para serem utilizados.

Uma abordagem alternativa são os métodos numéricos, que consistem no conjunto de ferramentas ou técnicas utilizadas na resolução aproximada de problemas matemáticos.

No contexto dos problemas de valor inicial explicitados por um conjunto de equações diferenciais autônomas de primeira ordem, trataremos dos integradores numéricos [9].

Em síntese, gostaríamos de descobrir, para um dado intervalo de tempo, o comportamento de $\mathbf{x}(t)$ sabendo que $\dot{\mathbf{x}} = \mathbf{F}(\mathbf{x})$ e $\mathbf{x}_0 = \mathbf{x}(0)$. Sendo \mathbf{F} uma função bem comportada o suficiente (sem grandes variações e/ou descontinuidades nas derivadas), garantindo a unicidade das soluções, nossa questão pode ser resolvida por meio de um integrador numérico. O mais antigo e mais simples deles é o método de Euler [33], ou método da reta tangente, cuja fórmula é dada por

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \mathbf{F}_n h, \quad n = 1, 2, \dots, \quad (3.22)$$

onde h , sendo $h \rightarrow 0$, é o tamanho do passo de tempo.

A ideia é calcular, recursivamente a partir das condições iniciais \mathbf{x}_0 e utilizando informações de \mathbf{F} , o passo sucessor baseado no valor do anterior. Assim, obtém-se uma sequência de valores $\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \dots$ que deverá se aproximar de $\mathbf{x}(t)$ em t_0, t_1, t_2, \dots

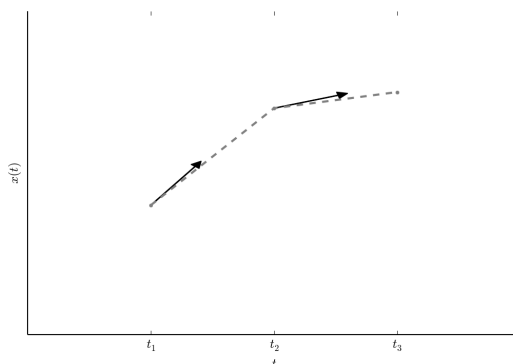


Figura 3.2 Método de Euler. O valor da derivada, utilizado como ponto de partida, é extrapolado em cada intervalo com o intuito de encontrar o próximo valor da função.

Entretanto, para uso prático existem razões que tornam o método de Euler não recomen-

dados. Para haver convergência a função integrada deve apresentar bom comportamento, sem grandes variações ou descontinuidades nas derivadas. Além disso, ele não é muito preciso comparado a outros métodos com o mesmo tamanho de passo de tempo. Mesmo assim, foi a partir de aprimoramentos do método de Euler que temos toda a família de métodos de Runge-Kutta [33].

A classe de equações diferenciais de interesse aqui é derivada de uma Hamiltoniana utilizando as equações de Hamilton, não existem garantias de que um dado integrador numérico necessariamente preserve propriedades canônicas desse sistema. E é nesse contexto que surge a proposta dos integradores simpléticos.

3.6.1 Integradores simpléticos

Tomemos a função Hamiltoniana

$$H = \frac{\mathbf{p}^2}{2} + V(\mathbf{x}, t), \quad (3.23)$$

cujas equações de movimento são dadas por

$$\mathbf{x}(\mathbf{x}_0, \mathbf{p}_0, t) \quad \mathbf{p}(\mathbf{x}_0, \mathbf{p}_0, t), \quad (3.24)$$

onde $(\mathbf{x}_0, \mathbf{p}_0)$ indica o estado do sistema no tempo $t = 0$.

Pensando na evolução temporal (3.24) como uma transformação canônica [23] teremos a expressão

$$(x, p) = M(t)(x_0, p_0), \quad (3.25)$$

sendo $M(t)$ o objeto responsável pela transformação de variáveis, no nosso caso uma evolução temporal. A expressão (3.25) é também conhecida como mapa simplético.

As transformações canônicas ocorrem nas variáveis do espaço de fases e preservam a forma canônica das equações de movimento. Ou seja, dadas as variáveis canônicas (q, p) , estaremos interessados na transformação

$$Q_i = Q_i(q, p, t), \quad P_i = P_i(q, p, t). \quad (3.26)$$

uma vez que seja possível determinar a função $K(Q, P, t)$ tal que

$$\dot{Q}_i = \frac{\partial K}{\partial P_i}, \quad \dot{P}_i = -\frac{\partial K}{\partial Q_i}. \quad (3.27)$$

Os integradores simpléticos, utilizados na abordagem numérica desse tipo de problema, devem conservar a relação (3.25). A elaboração do método consiste em

- tomar o parâmetro t pequeno;
- considerar cada aplicação do mapa (3.25) como uma iteração no tempo de modo que o erro do método será controlado pelo tamanho de t .

Consequentemente, podemos dizer que se um mapa simplético tem ordem n , ou seja

$$(x, p) = M_n(t)(x_0, p_0), \quad (3.28)$$

significa que

$$|M(t) - M_n(t)| = O(t^{n+1}). \quad (3.29)$$

o integrador simplético proposto se baseia em uma transformação canônica infinitesimal que corresponde a uma evolução temporal. O método se resume na determinação de $M_n(t)$.

3.6.2 Primeira ordem

O caso de primeira ordem não apresenta relevância prática, o objetivo da subseção é ilustrar o método e utilizar a técnica na demonstração da segunda ordem, que conforme veremos equivale ao integrador *leapfrog*.

Efetuada uma transformação canônica em (3.24), cujo o novo Hamiltoniano H' é expressado em termos das condições iniciais, as equações de movimento serão

$$\dot{\mathbf{x}}_0 = 0 \quad \dot{\mathbf{p}}_0 = 0. \quad (3.30)$$

A outra possibilidade é que o Hamiltoniano é identicamente zero (ou que independe de \mathbf{x}_0 e \mathbf{p}_0). A ideia é realizar as sucessivas transformações canônicas antes de chegarmos às condições

iniciais, ou pelo menos para outro conjunto de coordenadas nas proximidades de $(\mathbf{x}_0, \mathbf{p}_0)$ em alguma ordem de t .

Sejam $(\mathbf{x}_1, \mathbf{p}_1)$ as novas variáveis. A maneira conveniente de fazer as transformações é por meio da função geradora [23] do tipo

$$F_3(x_1, p, t) = -x_1 p + G(x_1, p, t); \quad (3.31)$$

$$x = -\frac{\partial F_3}{\partial p} = x_1 - \frac{\partial G}{\partial p}, \quad p_1 = -\frac{\partial F_3}{\partial x_1} = p - \frac{\partial G}{\partial x_1}; \quad (3.32)$$

$$H_1 = H + \frac{\partial F_3}{\partial t}. \quad (3.33)$$

que implica em

$$G = -\left[\frac{p^2}{2} + V(x_1, 0)\right]t \quad (3.34)$$

$$x = x_1 + pt \quad p_1 = p - f(x_1, 0)t, \quad (3.35)$$

sendo a força, f , definida como $f(x, 0) = -\frac{\partial}{\partial x}V(x, 0)$, onde $V(x, 0)$ é o potencial.

Considerando (3.33), tomando t pequeno e expandindo até a primeira ordem, teremos

$$H_1 = t \frac{\partial}{\partial t}V(x_1, 0) - t p_1 f(x_1, 0) + O(t^2), \quad (3.36)$$

cujas as equações de Hamilton nos fornece

$$x_1 = const + O(t^2) \quad p_1 = const + O(t^2) \quad (3.37)$$

A conclusão é que se usarmos x_1 e p_1 como condição inicial o erro introduzido será $O(t^2)$ e portanto teremos um mapa simplético de primeira ordem, $M_1(t)$. Uma outra característica importante dessa abordagem é que do ponto de vista da condição inicial, segundo as equações (3.35), o momento p deve ser calculado primeiro e então usado para avaliar x . Vale recordar que as iterações do mapa simplético nos levam a instantes antes de (x, p) , o que torna necessário a inversão da equação do momento.

3.6.3 Segunda ordem (*Leapfrog*)

A determinação do mapa simplético de segunda ordem será baseada nos resultados da subseção anterior, no entanto com reformulações no procedimento que serão úteis na demonstração. A modificação consiste em executar a transformação canônica $(x, p) \rightarrow (x_2, p_2)$, onde

(x_2, p_2) são as novas variáveis e conseqüentemente as condições iniciais, considerando uma transformação intermediária $(x, p) \rightarrow (x_1, p_1)$, conforme

$$\begin{aligned} (x, p) \rightarrow (x_1, p_1) : \quad x &= x_1 + apt & p_1 &= p - tf(x_1, bt) \\ F &= -x_1 p - \frac{a}{2} p^2 t - tV(x_1, bt) \end{aligned} \quad (3.38)$$

$$\begin{aligned} (x_1, p_1) \rightarrow (x_2, p_2) : \quad x_1 &= x_2 + (1-a)p_1 t & p_2 &= p_1, \\ F &= -x_2 p_1 - \frac{a-1}{2} p_1^2 t \end{aligned} \quad (3.39)$$

onde a e b representam a fração do tempo utilizada na transformação intermediária para a posição e o momento, respectivamente.

Considerando H_2 como o novo Hamiltoniano relativo a transformação $(x_1, p_1) \rightarrow (x_2, p_2)$, expandindo-o em t pequeno, teremos

$$H_2 = t(1-2a)p_2 f(x_2, 0) + t(1-2b) \frac{\partial}{\partial t} V(x_2, 0) + O(t^2). \quad (3.40)$$

Se (x_2, p_2) são usados como condição inicial e escolhendo $a = b = 1/2$ teremos no mínimo termos de segunda ordem em H_2 , o que significa dizer que o mapa total é de ordem 2.

Finalizando o processo, com a notação $(x_2 \rightarrow x_0, p_2 \rightarrow p_0, t_0)$, tornando a escrever as transformações na ordem inversa e realizando as generalizações para muitas dimensões, teremos um integrador simplético de segunda ordem também conhecido por *leapfrog*

$$\begin{aligned} \text{Mapa } (\mathbf{x}, \mathbf{p}) &= M_2(h)(\mathbf{x}_0, \mathbf{p}_0) \\ (1) \mathbf{p}_1 &= \mathbf{p}_0, & \mathbf{x}_1 &= \mathbf{x}_0 + \mathbf{p}_1 h/2 \\ (2) \mathbf{p} &= \mathbf{p}_1 + hf(\mathbf{x}_1, t_0 + h/2), & \mathbf{x} &= \mathbf{x}_1 + \mathbf{p} h/2 \end{aligned} \quad (3.41)$$

3.7 Ortonormalização de Gram-Schmidt

O processo de Gram-Schmidt é uma técnica relativamente simples, baseada em álgebra linear, com aplicações importantes em física matemática, mecânica quântica e análise numérica [18, 19, 32].

É sempre possível converter uma base qualquer v em uma outra ortonormalizada, ou seja todos os vetores que a compõem devem apresentar módulo unitário e o produto interno entre

qualquer par de elementos diferentes é sempre nulo. O processo no qual a transformação é realizada, é conhecido como ortonormalização de Gram-Schmidt.

Considerando uma base $B = \{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_N\}$, queremos tomar uma combinação linear de \mathbf{a}_i de maneira tal que resulte em N vetores normalizados e ortogonais entre si. Primeiro vamos escolher um vetor como referência tal que $\mathbf{e}_1 = \mathbf{a}_1 / \sqrt{\mathbf{a}_1 \cdot \mathbf{a}_1}$, e note que $\mathbf{e}_1 \cdot \mathbf{e}_1 = 1$. Se subtrairmos de \mathbf{a}_2 sua projeção ao longo de \mathbf{e}_1 , ou seja $\mathbf{e}'_2 = \mathbf{a}_2 - (\mathbf{e}_1 \cdot \mathbf{a}_2)\mathbf{e}_1$, teremos um vetor que é ortogonal a \mathbf{e}_1 . Dividindo-o pelo próprio módulo, $\mathbf{e}_2 = \mathbf{e}'_2 / \sqrt{\mathbf{e}'_2 \cdot \mathbf{e}'_2}$, teremos um segundo vetor normalizado e perpendicular a \mathbf{e}_1 .

Para o \mathbf{a}_3 , devemos subtrair sua projeções ao longo de \mathbf{e}_1 e \mathbf{e}_2 e em seguida normalizá-lo. O processo deve ser aplicado aos demais vetores, e encontra-se resumido nas equações a seguir

$$\begin{aligned} \mathbf{e}'_{m+1} &= \mathbf{a}_{m+1} - \sum_{i=1}^m (\mathbf{e}_i \cdot \mathbf{a}_{m+1})\mathbf{e}_i, \\ \mathbf{e}_{m+1} &= \frac{\mathbf{e}'_{m+1}}{\sqrt{\mathbf{e}'_{m+1} \cdot \mathbf{e}'_{m+1}}}. \end{aligned} \quad (3.42)$$

O processo de ortonormalização de Gram-Schmidt é parte importante deste trabalho, nossa versão do método foi desenvolvida em linguagem C e encontra-se à disposição nos apêndices B e C.

Do ponto de vista numérico para cálculo dos expoentes de Lyapunov, Gram-Schmidt é relevante porque

- as sucessivas normalizações evitam a divergência do vetor \mathbf{w} , equações (3.8) e (3.9), no mapa tangente e garantem que as órbitas perturbadas permaneçam infinitesimalmente próximas da trajetória referência no método dos clones;
- as ortogonalizações permitem obter todo o espectro de Lyapunov.

Resultados e discussões

Este capítulo é dedicado à exposição dos resultados obtidos durante o andamento do doutorado. É um trabalho de essência numérica cujos objetos de estudo são os sistemas Hamiltonianos de longo alcance e as técnicas de obtenção dos expoentes de Lyapunov. Os resultados foram obtidos em simulações computacionais desenvolvidas em C e tratados em Python.

Os resultados incluem a ratificação de aspectos relacionados aos sistemas Hamiltonianos, o desempenho dos integradores utilizados, dados sobre a eficiência no método tangente e dos clones e propriedades do espaço de fases ligadas a caoticidade dos sistemas.

4.1 Hamiltoniano de campo médio

As medidas feitas no HMF estão relacionadas com a evolução do sistema e com a dinâmica das diferenças sob o regime linearizado. Para integrar as equações de movimento foi utilizado o *leapfrog*, com tamanho do passo de tempo $h = 0.05$ sendo o tempo definido como $i \times h$, onde i é o número de iterações executadas pelo integrador. Como condição inicial foi utilizada a distribuição *waterbag*, introduzida na seção 2.1. As flutuações na energia e no momento total são da ordem de 10^{-5} e 10^{-11} , respectivamente, conforme mostram as figuras (4.1) e (4.2).

A obtenção de todo o espectro de Lyapunov para o HMF é feita numericamente pelo método tangente e/ou pelo método dos clones, discutidos nas seções anteriores e cujos algoritmos se encontram nos apêndices desse trabalho. Os expoentes de Lyapunov foram aferidos da média de diferentes realizações correspondentes às mesmas condições iniciais, ou seja sem alterações na magnetização inicial, energia e momento total.

O conjunto de resultados da figura (4.3), são referentes ao caso subcrítico com $N = 200$ partículas, e exibem o funcionamento dos métodos estudados, tangente e clones, partindo de uma

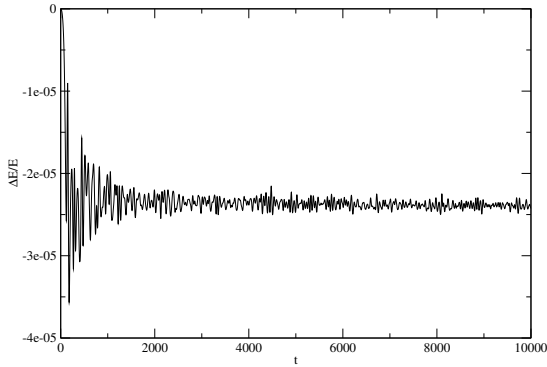


Figura 4.1 HMF, $N = 10000$, $e = 0.55$. Erro relativo da energia versus o tempo.

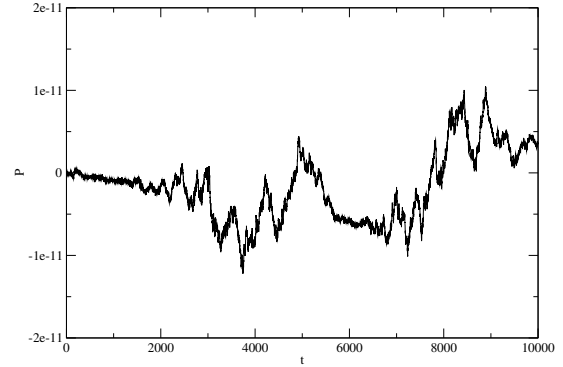


Figura 4.2 HMF, $N = 10000$, $e = 0.55$. Valor do momento total em cada instante de tempo.

distribuição *waterbag* e uma outra distribuição, Gaussiana nas posições e nas velocidades. O valor das medidas e seus respectivos erros revelam concordância, pode-se dizer que os métodos apresentam congruência e os resultados são robustos em relação as condições iniciais.

Uma outra verificação tem vínculo com propriedades dos sistemas Hamiltonianos por meio da representação de todo o espectro de Lyapunov, que implica em $\sum_i^{2N} \lambda_i = 0$. O valor do expoente positivo, ligado a divergências do fluxo de soluções infinitesimalmente próximas no espaço de fases, é acompanhado necessariamente de um outro negativo, com fundamento na conservação do volume no espaço de fases. Em conformidade com os resultados obtidos no modelo de folhas, figura (4.4).

Por conseguinte na condição dos sistemas Hamiltonianos, é suficiente a análise apenas dos N maiores expoentes.

Uma outra quantidade a ser explorada é a entropia de Kolmogorov-Sinai, utilizada também na caracterização de regimes caóticos. Para sistemas fechados e isolados coincide com a soma de todos os expoentes positivos, $S_{KS} = \sum_{\lambda_i > 0}^N \lambda_i$ [15]. Por ser obtida da taxa de separação de todas as direções nas quais as trajetórias divergem no espaço de fases, a entropia Kolmogorov-Sinai, S_{KS} , porta informações mais gerais do sistema, comparando-a com o maior expoente de Lyapunov λ_1 .

A dependência do maior expoente com a energia do sistema é um tópico importante. Acredita-se que instabilidades atribuídas ao caos estejam de certa forma relacionadas à transição de

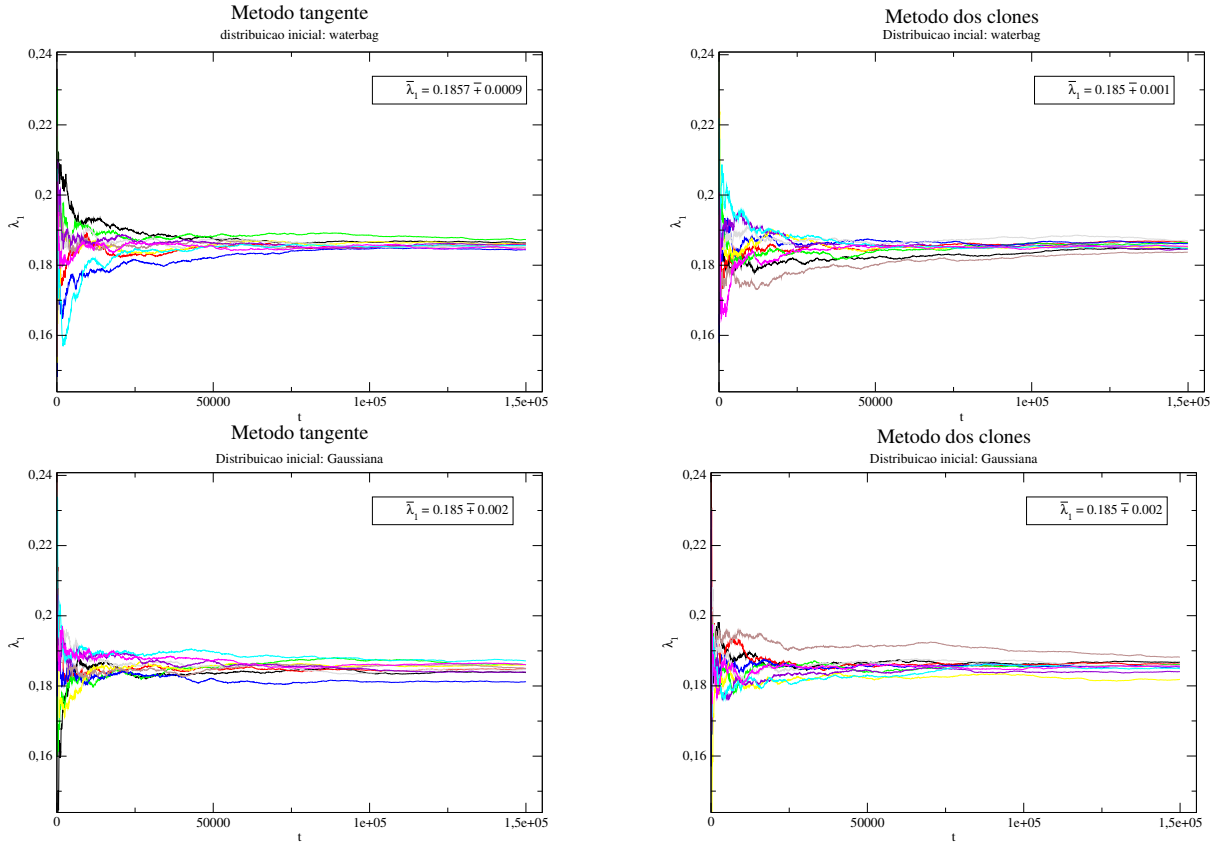


Figura 4.3 HMF, $N = 200$, $e = 0.55$. Convergência do maior expoente de Lyapunov, estabelecida quando as medidas experimentam pequenas oscilações em torno de um dado valor. A média extraída entre dez realizações nos fornece o valor da medida enquanto o erro é obtido através da variância. As figuras exibem a performance dos dois métodos para duas condições iniciais distintas, *waterbag* e Gaussiana nas posições e momentos. O tempo de computação para o mapa tangente foi em torno de 55 minutos enquanto o método dos clones utilizou 50 minutos para desempenhar a mesma tarefa.

fase no HMF [45], já que λ_1 apresenta valor máximo em torno da energia crítica de transição $e_c = E/N = 3/4$, conforme mostra o gráfico da figura (4.5).

Segundo previsões teóricas, espera-se que o maior expoente seja estritamente positivo no caso $0 < e < e_c$ e escale com $N^{-1/3}$ para $e > e_c$ e $N \rightarrow \infty$ [17]. A respeito de como o valor do maior expoente de Lyapunov se comporta no HMF com diferentes números de partículas, utilizando as rotinas aqui desenvolvidas, foram extraídos resultados tanto para o caso subcrítico quanto para o supercrítico.

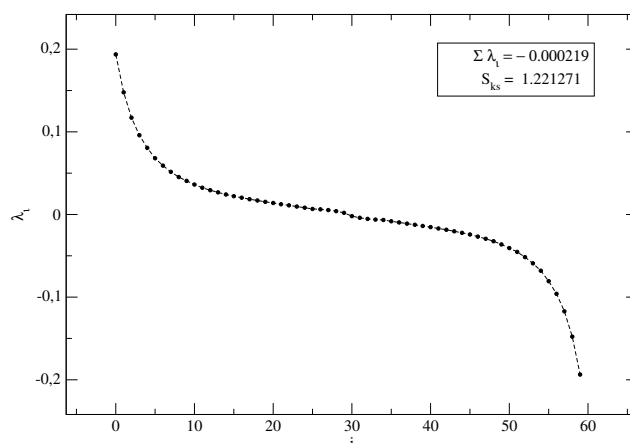


Figura 4.4 HMF, $N = 30$, $e = 0.55$. Espectro completo de Lyapunov. Cada ponto representa a taxa de divergência entre soluções infinitesimalmente próximas em uma dada direção do espaço de fases, portanto $N = 30$ partículas teremos 60 expoente, sendo o índice $i = 1$ atribuído à direção de maior expansão. Os resultados mostram que a soma de todos os expoente é numericamente zero, uma vez que seu valor é da ordem do erro associado a uma medida, conforme mostram os gráficos da figura (4.3). O valor de λ_1 juntamente com S_{k_s} , cujo a medida concorda com os resultados da referência [26], diz respeito como o valor dos N expoentes positivos estão distribuídos.

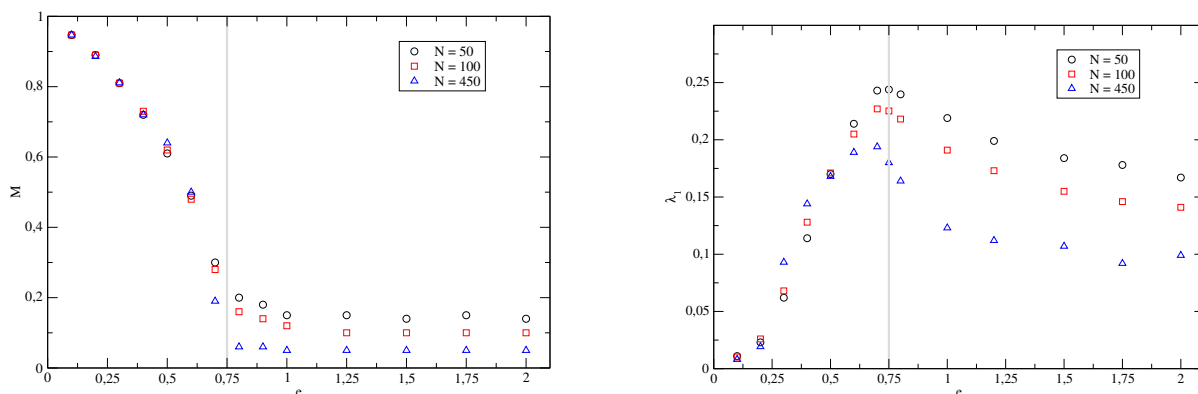


Figura 4.5 HMF. Módulo do vetor magnetização e o valor λ_1 em função da energia para $N = 50$, $N = 100$ e $N = 450$ partículas. As figuras indicam que o intervalo de energia associado a transição de fases coincide com a região onde o expoente λ_1 assume valor máximo.

Esses resultados envolvem um número modesto de partículas, o que provavelmente justifica o fato do fator de escala encontrado não se tratar rigorosamente do teórico. A figura (4.6) mostra o valor do maior expoente de Lyapunov com o número de partículas, para energia $e = 1.2$. O

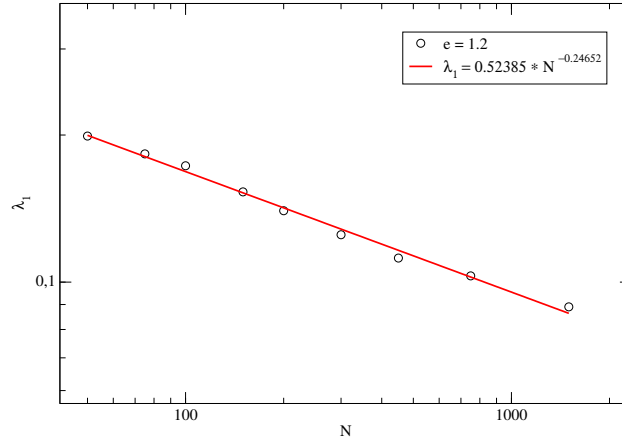


Figura 4.6 HMF, $e = 1.2$. Valor do maior expoente de Lyapunov em função do número de partículas em escala logarítmica. A reta de vermelho segue uma expressão do tipo $\alpha N^{-\beta}$, onde nossos resultados apontam $\beta \approx 1/4$ como sendo o fator de escala apropriado.

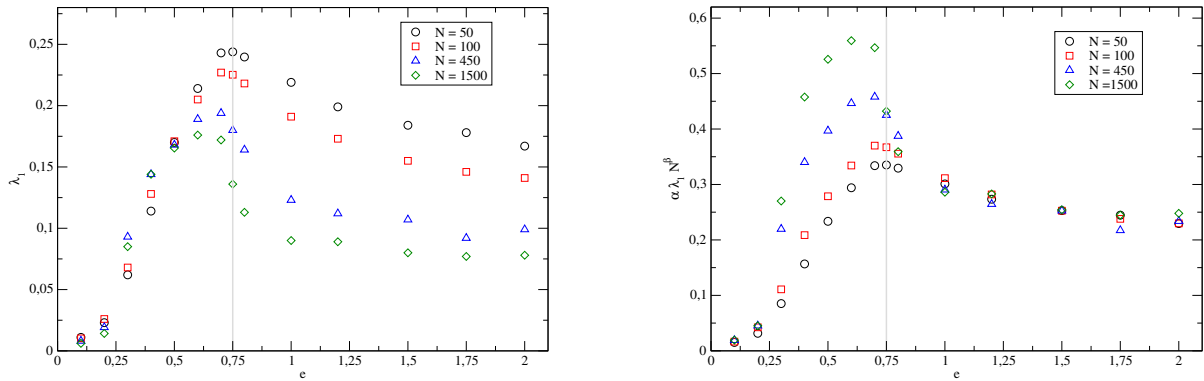


Figura 4.7 HMF. Valor do maior expoente de Lyapunov em função da energia para $N = 50$, $N = 100$, $N = 450$, e 1500 partículas. O gráfico da direita teve o eixo das ordenadas multiplicados pelo fator de escala $\alpha N^{-\beta}$, cujos valores $\alpha \approx 0.52$ e $\beta \approx 1/4$ estão explicitados na figura (4.6).

gráfico, em escala logarítmica, compara nossos resultados numéricos com uma reta do tipo $\lambda_1 = \alpha N^{-\beta}$, sendo os valores encontrados $\alpha \approx 0.52$ e $\beta \approx 1/4$. Os gráficos seguintes, (4.7), justificam o uso da lei de escala obtida para o caso supercrítico, multiplicando a ordenada do gráfico λ_1 versus e por $\alpha N^{1/4}$.

Manos e Ruffo, na referência [26], reportam a dificuldade em obter a escala teórica $N^{-1/3}$ e sugere um fator "intermediário" $N^{-1/4}$, utilizado no caso numérico, para um número moderadamente grande de partículas, na escala de energias mais altas e está em concordância com os

resultados obtidos utilizando os algoritmos desenvolvidos durante o projeto de doutoramento.

Também propusemos uma maneira de como identificar as regiões mais sensíveis do espaço de fases. Atribuindo a cada partícula a quantidade

$$\delta I_i = \frac{(\delta\theta_i)^2 + (\delta p_i)^2}{|\mathbf{w}(\delta\theta, \delta p)|^2}, \quad (4.1)$$

teremos a fração da contribuição de cada estado na taxa de divergência entre soluções infinitesimalmente próximas. Sendo $\delta\theta_i$ e δp_i componentes do vetor do espaço tangente, a distribuição deve obedecer a relação $\sum_i^N \delta I_i = 1$.

A distribuição dos valores δI_i foi calculada para o HMF durante o estado quase-estacionário após a convergência do valor do maior expoente de Lyapunov. Os pontos da figura (4.8) são

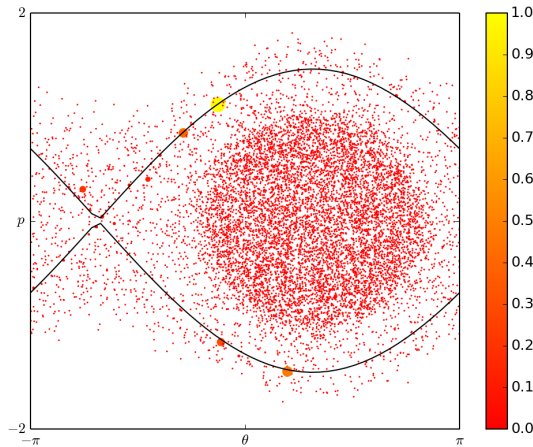


Figura 4.8 HMF, $N = 10000$, $e = 0.55$. Representação da distribuição δI_i no espaço de fases para o caso subcrítico.

os pares conjugados (θ_i, p_i) , enquanto a linha cheia representa a separatriz obtida pelos valores instantâneos de M e ϕ . A escala de cor e tamanho tem relação com valores δI_i . A escala de cores é uma medida relativa, onde o amarelo representa o maior valor δI_i e o vermelho o menor, o tamanho dos pontos é proporcional ao valor de δI_i .

Nossos resultados revelam que a grande maioria das partículas apresentam δI_i praticamente nulo (da ordem 10^{-7}), com pouco valores apreciáveis situados claramente na região da separatriz.

Faz sentido pensar que a separatriz seja de fato a região mais sensível do espaço de fases. Ela divide as partículas mais energéticas, capazes de cruzar o círculo, daquelas que estão confinadas devido ao potencial exercido pelo aglomerado. Perturbações infinitesimais são irrelevantes tanto para as partículas livres quanto para as fortemente aprisionadas. Percebe-se assim, a região da separatriz como uma espécie de "limbo", onde o estado de uma partícula poderia evoluir tanto para o regime livre quanto para o confinado, dependendo apenas de uma pequena alteração.

Na figura (4.9), em um dado instante durante o estado quase estacionário, distingue-se os estados das partículas através de cores, de verde estão as partículas mais energéticas (verde escuro com velocidade positiva e claro negativa), azul representando as partículas do *core* e em amarelo aquelas situadas no halo. Evoluindo o sistema por um tempo razoável, as pulsações ocorridas na separatriz permitem a inversão das partículas verdes claras com as escuras, e incluem parte das partículas amarelas nessa mistura. Como as pulsações ocorrem longe do aglomerado, as partículas azuis permanecem fora desse processo. Isso mostra que, para esse conjunto de parâmetros no HMF, a região do espaço de fases com os maiores valores δI_i coincide com o lugar onde ocorrem mais trocas de estados.

Como resultado final da seção, confrontamos os dois métodos desenvolvidos durante o projeto de doutorado, o mapa tangente e o método dos clones. Foram realizadas análises de convergência e comparações dos tempos de processamento até a obtenção do maior expoente de Lyapunov.

A estimativa da taxa de convergência consiste na análise sistemática dos resultados obtidos via método numérico para diferentes tamanhos de passo de iteração. Tal avaliação ratifica se de fato há convergência e o quão rápido ela ocorre.

O estudo foi realizado para o HMF com $N = 50$ e energia por partícula $e = 0.1$. Para este conjunto de parâmetro o valor do maior expoente de Lyapunov é da ordem de 10^{-2} , conforme revelam os resultados (4.10), em concordância com a referência [26].

As tabelas a seguir mostram a dependência do maior expoente com o tamanho do passo do integrador *leapfrog*, obtidos pelo método tangente e dinâmica de clones. As soluções de ambas as técnicas de fato convergem, já que os valores de λ_1 , extraídos com h pequeno o suficiente,

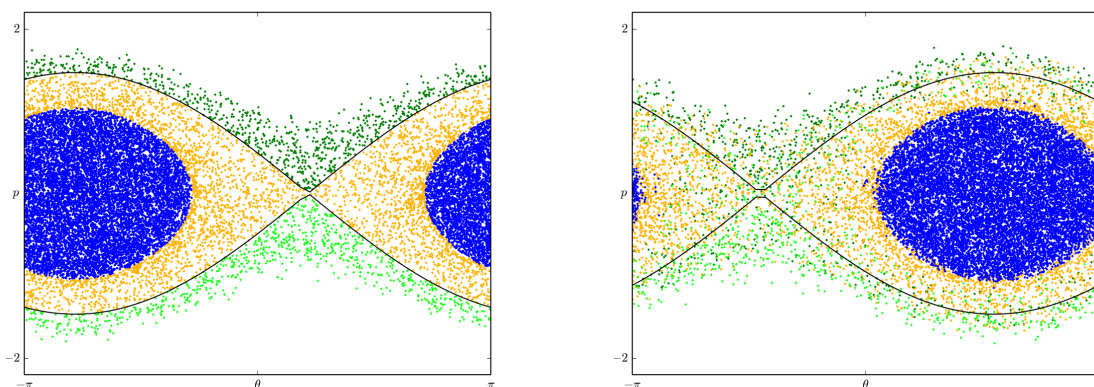


Figura 4.9 HMF. Classificação, segundo o critério de energia, dos estados das partículas utilizando diferentes cores. Permitindo que o sistema evolua por um tempo razoável, constata-se que as trocas de estados ocorrem principalmente em torno da separatriz, região na qual as quantidades δI_i apresentaram maiores valores.

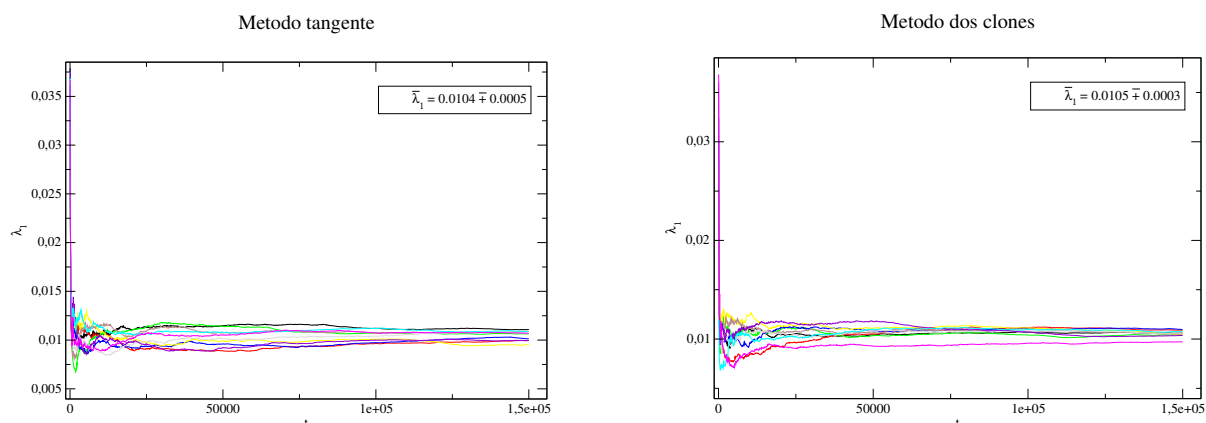


Figura 4.10 HMF, $N = 50$, $e = 0.1$. Convergência do maior expoente de Lyapunov em função do tempo. A média extraída entre dez realizações nos fornece o valor da medida enquanto o erro é obtido através da variância. As figuras mostram o funcionamento dos métodos segundo os parâmetros já mencionados.

oscilam dentro da margem de erro tolerada, segundo as medidas (4.10).

As medidas $(\lambda_h - \lambda_{h/2})$, $\frac{\lambda_h - \lambda_{h/2}}{\lambda_{h/2} - \lambda_{h/4}}$ e $\log_2 \frac{\lambda_h - \lambda_{h/2}}{\lambda_{h/2} - \lambda_{h/4}}$, estão relacionadas a que taxa ocorrem as convergências (ver seção 3.5).

Mapa tangente

h	λ_h	$\lambda_h - \lambda_{h/2}$	$\frac{\lambda_h - \lambda_{h/2}}{\lambda_{h/2} - \lambda_{h/4}}$	$\log_2 \frac{\lambda_h - \lambda_{h/2}}{\lambda_{h/2} - \lambda_{h/4}}$	tempo
3.2	0.327	0.029	0.136	-2.878	12.85"
1.6	0.298	0.214	5.0235	2.3287	24.252"
0.8	0.0842	0.0426	2.0481	1.0343	23.763"
0.4	0.0416	0.02008	2.0594	1.0422	44.765"
0.2	0.02080	0.0101	—	—	1'29.531"
0.1	0.0107	0.0000	0.0000	—	2'53.352"
0.05	0.0107	0.0003			5'52.983"
0.025	0.0104				11'26.497"

Tabela 4.1 Análise de convergência, mapa tangente. A tabela relaciona o tamanho do passo utilizado no processo de integração com o valor medido do expoente. Os espaços completados com — correspondem aos casos onde os resultados envolvendo fração e/ou o logaritmo não estão definidos.

Método dos clones

h	λ_h	$\lambda_h - \lambda_{h/2}$	$\frac{\lambda_h - \lambda_{h/2}}{\lambda_{h/2} - \lambda_{h/4}}$	$\log_2 \frac{\lambda_h - \lambda_{h/2}}{\lambda_{h/2} - \lambda_{h/4}}$	tempo
3.2	0.2295	0.0458	0.2655	-1.9132	11.143"
1.6	0.1837	0.1725	191.6667	7.5825	21.003"
0.8	0.0112	0.0009	-9.0000	—	21.504"
0.4	0.0103	-0.0001	0.5000	-1.0000	41.789"
0.2	0.0104	-0.0002	-2.0000	—	1'19.555"
0.1	0.0106	0.0001	—	—	2'36.490"
0.05	0.0105	0.0000			5'12.015"
0.025	0.0105				10'10.474"

Tabela 4.2 Análise de convergência, método dos clones. A tabela relaciona o tamanho do passo utilizado no processo de integração com o valor medido do expoente. Os espaços completados com — correspondem aos casos onde os resultados envolvendo a fração e/ou o logaritmo não estão definidos.

Os resultados mostram que a convergência no método dos clones ocorre para um valor mais alto de h , algo em torno de 0.8 e 0.4, enquanto no mapa tangente acontece em $h = 0.1$. A última coluna reporta o tempo de processamento até a obtenção da medida de λ_1 , extraída da média entre dez diferentes realizações.

A possibilidade de reproduzir corretamente as medidas utilizando um h mais alto implica em menos iterações, concedendo ao método dos clones mais eficiência, se comparado com o mapa tangente (ver figura 4.11).

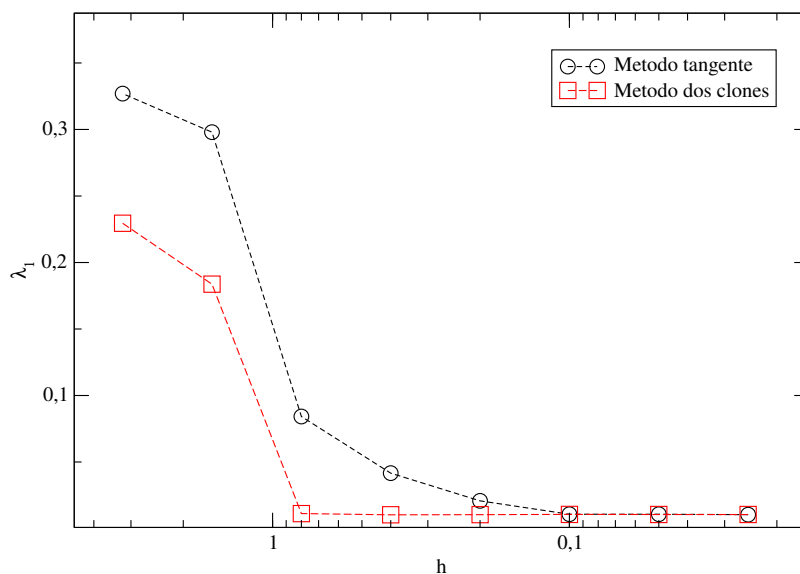


Figura 4.11 HMF, $N = 50$, $e = 0.1$. Na figura foram registrados os valores dos expoentes de Lyapunov obtidos, rigorosamente para o mesmo conjunto de parâmetros, com diferentes tamanhos de passo do integrador. É importante notar que o eixo horizontal segue escala logarítmica e encontra-se invertido. Esse resultado mostra que a convergência do método dos clones ocorre para valores consideravelmente maiores de h , do ponto de vista computacional implica em menos iterações que significa menor tempo de computação.

Até para o mesmo tamanho de passo h a dinâmica de clones se mostrou mais eficaz, cujo o tempo de computação foi inferior em absolutamente todas as comparações, conforme mostram nossas análises ao longo da seção. Isso acontece provavelmente porque a resolução das equações linearizadas, no método tangente, exige a construção de uma matriz Jacobiana $2N \times 2N$ a cada passo de tempo, que é dispensável no método dos clones.

4.2 Modelo de folhas autogravitantes

Introduzida em seções anteriores, a dinâmica das folhas autogravitantes ocorre segundo um potencial que independe da distância entre os pares. Conseqüente, todo processo é baseado em movimentos uniformemente variados, onde as forças são reformuladas após cada cruzamento. O fato dos eventos entre colisões serem analíticos, conferem às variações na força relevância na emergência de caos no sistema.

Na prática, o cálculo dos expoentes de Lyapunov exige um integrador que ressalte as colisões. Tal tarefa, esclarecida na seção 3.2.2, foi superada pelo *event driven* proposto por este projeto, que estabelece a evolução do sistema com base nas colisões.

Os gráficos das figuras (4.12) e (4.13) exibem o erro relativo da energia e do momento total. Como condição inicial foi utilizada uma distribuição *waterbag*, que consiste em definir valores Δx e Δp , baseados na energia do sistema e a partir dos quais os estados são distribuídos aleatoriamente.

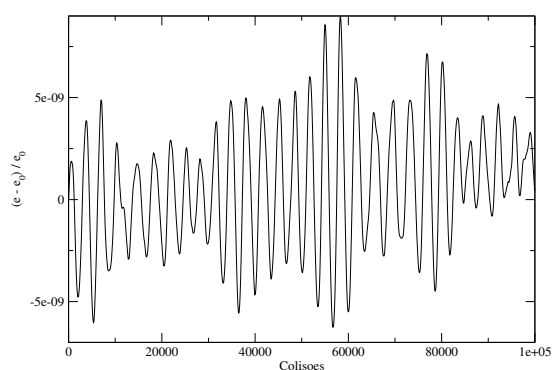


Figura 4.12 Folhas gravitacionais, $N = 75$, $e = 1.0$. Erro relativo da energia versus o tempo.

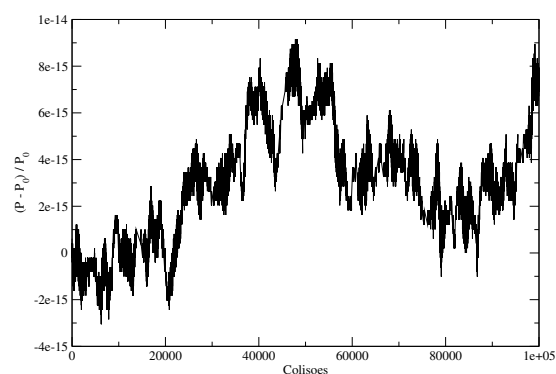


Figura 4.13 Folhas gravitacionais, $N = 75$, $e = 1.0$. Erro relativo do momento versus o tempo.

As medidas foram extraídas da média entre dez realizações que correspondem a mesma energia, ver figura (4.14).

Propriedades de conservação do volume no espaço de fases, que se manifestam através da relação $\sum_i^{2N} \lambda_i = 0$, são recuperadas também no modelo de folhas gravitacionais, conforme

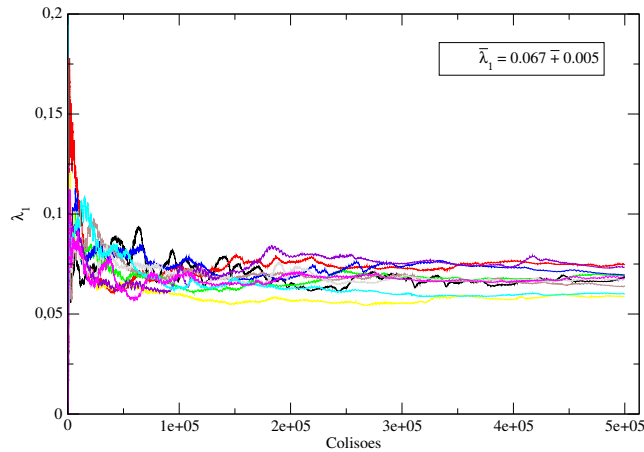


Figura 4.14 Folhas gravitacionais, $N = 75$, $e = 1$. Convergência da medida do maior expoente de Lyapunov em função do tempo para várias realizações, incluindo o valor médio e o desvio. Medidas obtidas através do método tangente.

mostram os resultados da figura (4.15).

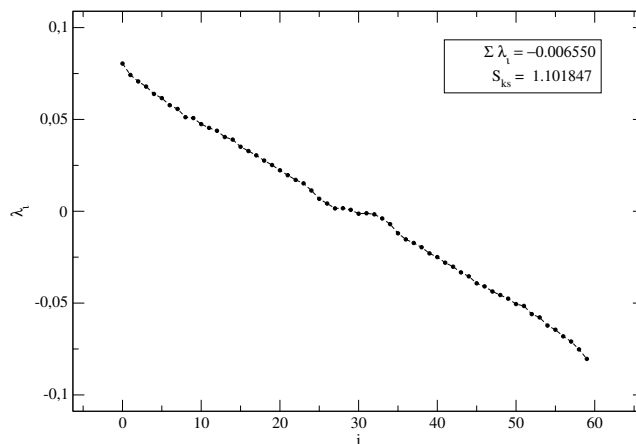


Figura 4.15 Folhas gravitacionais, $N = 30$, $e = 1$. Espectro completo de Lyapunov. Cada ponto representa a taxa de divergência entre soluções infinitesimalmente próximas em uma dada direção do espaço de fases, portanto $N = 30$ partículas teremos 60 expoentes, sendo o índice $i = 1$ atribuído a direção de maior expansão. Os resultados mostram que a soma de todos os expoente é numericamente zero, uma vez que seu valor é da ordem do erro associado a uma medida, conforme mostram os gráficos da figura (4.14).

O comportamento do maior expoente com o tamanho do sistema é um tema recorrente

nas investigações de modelos de longo alcance. Em relação as folhas autogravitantes, reporta-se que com o aumento na quantidade de partículas há um enfraquecimento nas propriedades ligadas a caos [43]. Nossos resultados, exibidos no gráfico da figura (4.16), corroboram com o que existe na literatura.

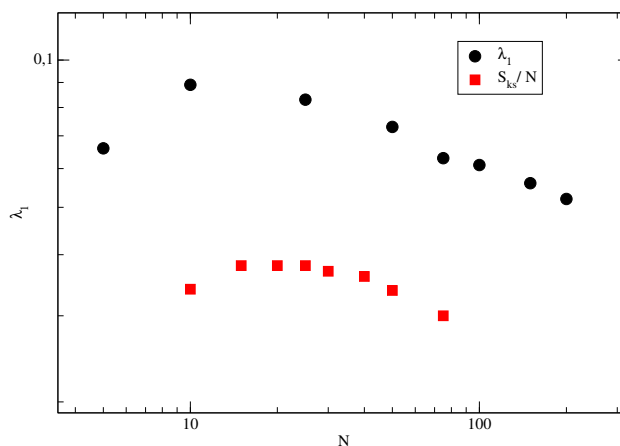


Figura 4.16 Folhas gravitacionais, $e = 1$. Valor do maior expoente de Lyapunov e da entropia de Kolmogorov-Sinai, definida como a soma de todos os expoentes positivos, contra o número de partículas.

A princípio foi aplicado ao modelo de folhas a mesma metodologia utilizada no HMF, incluindo o tipo de integração. Sabe-se da relevância das colisões na taxa de divergência entre soluções muito próximas. *Leapfrog* aplicado ao modelo de folhas, entre um passo de tempo e outro, suprime a ocorrência de várias colisões, implicando na não convergência do valor do expoente de Lyapunov. O que justifica o uso do *event driven*, procedimento cuja a integração é baseada nas colisões ocorridas no sistema.

Optar pelo *event driven* como integrador, apesar de importante na resolução das equações lineares e conseqüentemente na obtenção do espectro de Lyapunov, significou abrir mão das informações do sistema durante o período entre duas colisões. Além disso, simulações baseadas no mesmo número de cruzamentos podem dispendir tempos consideravelmente diferentes, que é de fato um problema na implementação do método dos clones, o qual reservamos os estudos do HMF e cujo os principais resultados se encontram compilados na seção anterior.

Considerações finais

Os resultados obtidos juntamente com as técnicas implementadas e divulgadas neste trabalho visam incrementar o entendimento sobre Caos e os Sistemas Hamiltoniano de longo alcance, através da implementação, e análises, dos métodos tangente, que utiliza equações linearizadas, e dos clones, fundamentado em cópias perturbadas, responsáveis por estimar os expoentes de Lyapunov. Deve-se ressaltar que todos os mecanismos utilizados na construção dos métodos foram inteiramente desenvolvidos durante o período do doutorado utilizando linguagem C. Significa dizer que foi desnecessária a utilização de bibliotecas específicas ou pacotes computacionais prontos, e ao invés disso nos dedicamos a investigar os integradores numéricos, as técnicas de análise de convergência e caracterização de distribuições, os processos de ortonormalização, análises de erro, entre outros, descritos com detalhes no capítulo 3 de métodos.

A competência adquirida no assunto permitiu-nos a construção de métodos "sob medida", onde o conjunto de procedimentos adotado para estimar os expoentes de Lyapunov sofreria variações dependendo do sistema abordado. Por exemplo, as folhas autogravitantes, onde o cálculo dos expoentes de Lyapunov exigiu a implementação de um integrador baseado nas colisões ocorridas no sistema, o *event driven*. Nesse caso, duas dinâmicas regidas pelo mesmo número de colisões podem despendar tempos completamente diferentes, o que inviabiliza a utilização do método dos clones. Diferente do HMF que teve sua evolução (tanto do caso não linear quanto do linear) estabelecida pelo método usual denominado *leap-frog*, onde o tempo é definido como o tamanho do passo do integrador vezes o número de iterações.

Com respeito às medidas, nos empenhamos em ser o mais criterioso possível. Todas elas foram extraídas da média de várias realizações, com condições iniciais equivalentes, e cujos algoritmos significativos foram rigorosamente definidos, baseados no erro derivado da raiz da variância. Nossos resultados além de satisfazer propriedades importantes dos sistemas Hamil-

tonianos, figuras (4.4) e (4.15), estão em concordância com o que há na literatura [27, 42].

Também propusemos uma forma de quantificar regiões sensíveis do espaço de fases através da distribuição δI_i , cujos resultados apontam, no caso subcrítico do HMF, para a separatriz. Os estudos nesse viés poderiam ser ampliados, investigando outras maneiras de correlacionar a separatriz com os valores δI_i no caso subcrítico. Abordar também o regime supercrítico do HMF e o modelo de folhas.

Um outro aspecto importante deste trabalho foram as análises numéricas acerca dos métodos para o cálculo dos expoentes de Lyapunov no HMF. O mapa tangente, que na literatura se apresenta como a forma padrão de obter os expoentes de Lyapunov, foi confrontado com o método dos clones, no qual o espectro é determinado por cópias perturbadas do sistema. Como resultado imediato, o tempo de computação do método dos clones se apresentou inferior em absolutamente todos os casos reportados, tabelas (4.1) e (4.2), possivelmente pela construção recursiva da matriz Jacobiana $2N \times 2N$, exigida pelo método tangente e desnecessária no método dos clones. Além disso, análises nos mostram que é possível chegar à convergência do expoente de Lyapunov com passos de integração maiores, utilizando o método dos clones, o que implica em menos iterações e tempos de computação ainda menores. O fato do método dos clones dispensar o uso da versão linearizada do sistema, possibilita-o estimar os expoentes de dinâmicas onde não se sabe explicitamente as equações que o regem (ou as regras de movimento não são dadas por equações diferenciais e mapas).

Para finalizar, vale ressaltar que os resultados atestam o bom funcionamento de nossas rotinas, no entanto como é de interesse, também, o comportamento dos expoentes de Lyapunov próximo ao limite termodinâmico, seria razoável aperfeiçoar nossos algoritmos, com o intuito de que os novos programas suportem o elevado número de partículas considerando um tempo aceitável de computação, que poderia ser realizado otimizando o *event driven*, por exemplo, e/ou traduzindo nossos códigos para linguagem CUDA (*Compute Unified Device Architecture*) [21], extensão para a linguagem C que permite o uso de programação paralela, por meio da utilização da unidade de processamento gráfica.

Aplicação do integrador simplético ao modelo onda-partícula

O modelo onda-partícula se refere à dinâmica de N partículas eletrostaticamente acopladas e cujo comportamento coletivo admite movimentos vibracionais que podem ser descritos como ondas. A abordagem, utilizando a função distribuição de velocidades, consiste em classificar as partículas, em ressonantes e não ressonantes. As partículas ressonantes têm velocidades próximas às velocidades de fase e possuem uma dinâmica autoconsistente com as ondas, que se comportam como osciladores harmônicos. Já as partículas não ressonantes, com velocidades consideravelmente diferentes da velocidade de fase, atuam apenas como meio de suporte para a propagação das ondas.

A Hamiltoniana que representa a interação de N partículas com M ondas, conforme a descrição acima [16] é

$$H = \sum_{r=1}^N \frac{p_r^2}{2} + \sum_{j=1}^M \omega_j \frac{X_j^2 + Y_j^2}{2} + \varepsilon \sum_{r=1}^N \sum_{j=1}^M \frac{\beta_j}{\kappa_j} [Y_j \sin(\kappa_j x_r) + X_j \cos(\kappa_j x_r)], \quad (\text{A.1})$$

onde x_r e p_r são respectivamente a posição e o momento das partículas e X_j e Y_j são coordenadas cartesianas da amplitude da onda. ε e β_j são parâmetros de acoplamentos, ω_j a frequência da onda e κ_j o vetor de onda.

Relevante em física de plasma, a interação onda-partícula originalmente foi descrita segundo teoria cinética. O modelo introduzido pela Hamiltoniana (A.1) é uma abordagem utilizando mecânica clássica.

Comparado ao campo médio do HMF, existe certa equivalência às ondas autoconsistente do modelo WP. Com a ressalva de que as componentes da magnetização compõem um parâmetro de ordem enquanto as coordenadas da onda, no sistema WP, são variáveis dinâmicas do problema.

As regras de movimento, com origem nas equações de Hamilton, são dadas por

$$\begin{aligned}
 \dot{x}_r &= \frac{\partial H}{\partial p} = p_r \\
 \dot{p}_r &= -\frac{\partial H}{\partial x} = -\varepsilon \sum_{s=1}^M \beta_s [Y_s \cos(\kappa_s x_r) + X_s \sin(\kappa_s x_r)] \\
 \dot{X}_j &= \frac{\partial H}{\partial Y} = \omega_j Y_j + \frac{\varepsilon \beta_j}{\kappa_j} \sum_{s=1}^N \sin(\kappa_j x_s) \\
 \dot{Y}_j &= -\frac{\partial H}{\partial X} = -\omega_j X_j + \frac{\varepsilon \beta_j}{\kappa_j} \sum_{s=1}^N \cos(\kappa_j x_s).
 \end{aligned} \tag{A.2}$$

Mais uma vez o problema é resolvido numericamente, utilizado o integrador simplético *leap frog*. No entanto, o fato da função Hamiltoniana possuir mais de um par de variáveis canônicas conjugadas exige um pouco mais de atenção durante a implementação.

Os códigos, desenvolvidos em C, encontram-se disponível logo a baixo. A respeito da notação utilizada nos comentários do algoritmo, vale saber que

- H_qX diz respeito a parte dependente de X da função $-\partial H/\partial q_r$;
- H_qY diz respeito a parte dependente de Y da função $-\partial H/\partial q_r$;
- H_X diz respeito a função $-\partial H/\partial X_j$;
- H_Y diz respeito a função $\partial H/\partial Y_j$;
- H_p diz respeito a função $\partial H/\partial p_r$.

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <math.h>
4
5  void integrator_step(int n, int m, double h, double epsilon, double beta[], double k[], double w0[], double q[], double p[],
6  double X[], double Y[])
7  {
8  int i, j;
9  double bac;
10 // (meio passo H_qX, H_X) *****
11
12 for( i = 0; i < n; i++ )
13 {
14     bac = 0.;
15     for( j = 0; j < m; j++ )

```

```

16         bac += beta[j]*X[j]*sin(k[j]*q[i]);
17
18         p[i] = p[i] + (-epsilon*bac)*h/2.;
19     }
20
21     for( i = 0; i < m; i++ )
22     {
23         bac = 0.;
24         for( j = 0; j < n; j++ )
25             bac += (beta[i]/k[i])*cos(k[i]*q[j]);
26
27         Y[i] = Y[i] + (-w0[i]*X[i] + epsilon*bac)*h/2.;
28     }
29
30     //(meio passo H_p)*****
31
32     for(i=0; i<n; i++)
33         q[i]= q[i] + p[i]*h/2.;
34
35     //(passo H_qY, H_Y)*****
36
37     for(i=0; i<n; i++)
38     {
39         bac=0.;
40         for(j=0; j<m; j++)
41             bac+=beta[j]*Y[j]*cos(k[j]*q[i]);
42
43         p[i] = p[i] + (-epsilon*bac)*h;
44     }
45
46     for(i=0; i<m; i++)
47     {
48         bac=0.;
49         for(j=0; j<n; j++)
50             bac += (beta[i]/k[i])*sin(k[i]*q[j]);
51
52         X[i]= X[i] + (w0[i]*Y[i] + epsilon*bac)*h;
53     }
54
55     //(meio passo H_p)*****
56
57     for(i=0; i<n; i++)
58         q[i]= q[i] + p[i]*h/2.;
59
60     //(meio passo H_qX, H_X)*****
61
62     for( i = 0; i < n; i++ )
63     {
64         bac = 0.;
65         for( j = 0; j < m; j++ )
66             bac += beta[j]*X[j]*sin(k[j]*q[i]);
67
68         p[i] = p[i] + (-epsilon*bac)*h/2.;
69     }
70
71     for( i = 0; i < m; i++ )
72     {
73         bac=0.;
74         for( j = 0; j < n; j++ )
75             bac += (beta[i]/k[i])*cos(k[i]*q[j]);

```

```

76
77     Y[i] = Y[i] + (-w0[i]*X[i] + epsilon*bac)*h/2.;
78 }
79
80 //fim*****
81
82     return;
83 }

```

As investigações do modelo onda-partícula se encontram em construção. No momento, os resultados gerados envolvem os casos mais simples.

Considerando uma partícula e uma onda ($N = 1$, $M = 1$), sob o conjunto de parâmetros $\omega = 1$, $\beta = 0$, $\kappa = 1$ e $\varepsilon = 1$, as equações de movimento serão dadas por

$$\begin{aligned}
 \dot{x} &= p \\
 \dot{p} &= 0 \\
 \dot{X} &= Y \\
 \dot{Y} &= -X.
 \end{aligned}
 \tag{A.3}$$

Trata-se de uma partícula livre na presença de uma onda cujas componentes evoluem segundo as equações de um oscilador harmônico simples. Não há desacordo com os resultados gerados pelo nosso integrador, como podemos ver nas figuras (A.1).

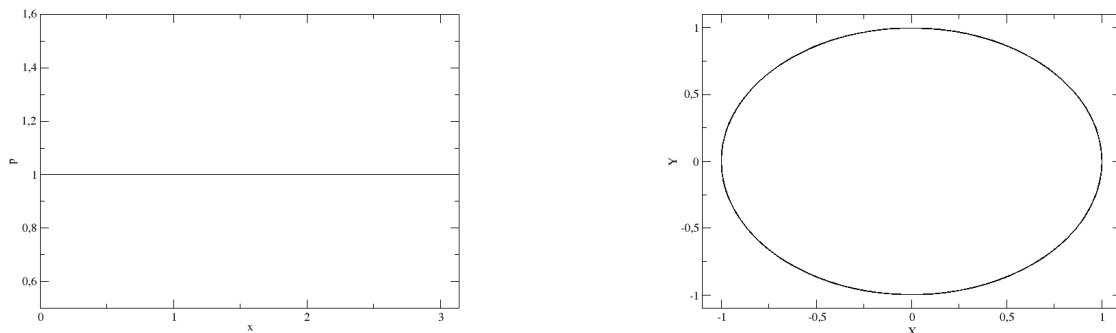


Figura A.1 Onda- partícula, $N = M = 1$, $\omega = 1$, $\beta = 0$, $\kappa = 1$, $\varepsilon = 1$. As condições iniciais utilizadas são $x(0) = 0$, $p(0) = 1$, $X(0) = 1$ e $Y(0) = 0$, com passo de integração $h = 0.1$.

Ainda para uma partícula e uma onda, mas considerando os parâmetros $\omega = 0$, $\beta = 1$, $\kappa = 1$ e $\varepsilon = 1$, obtivemos os gráficos das figuras (A.2), que comparam os resultados do nosso

integrador simplético com soluções obtidas utilizando um Runge-kutta , e mais uma vez há concordância.

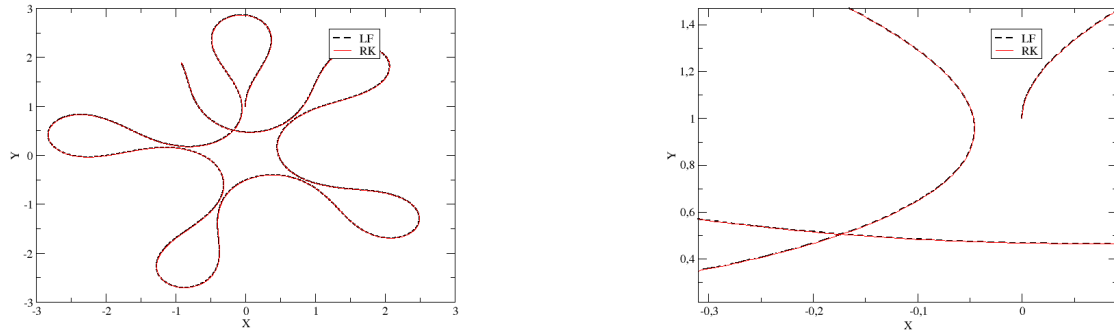


Figura A.2 Onda- partícula, $N = M = 1$, $\omega = 0$, $\beta = 1$, $\kappa = 1$, $\varepsilon = 1$. Comparação dos resultados do integrador simplético apresentado (tracejado, em preto) com resultados obtidos via Runge-Kutta (cheio, em vermelho).

O erro relativo na energia e no momento são da ordem de 10^{-3} , conforme mostram os gráficos da figura (A.3).

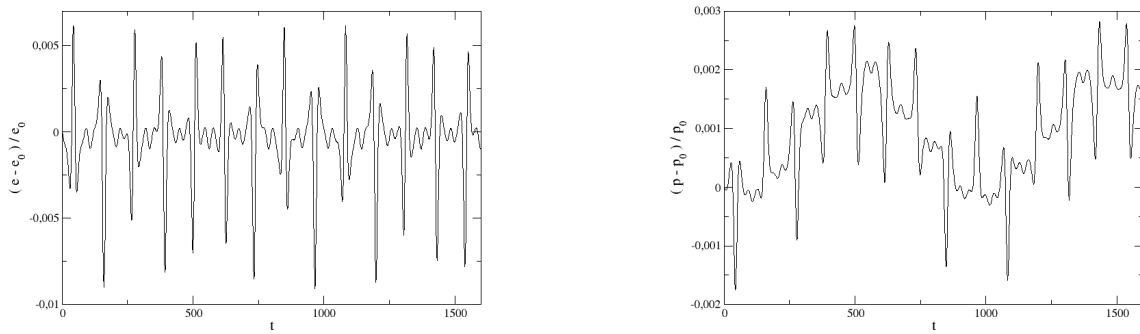


Figura A.3 Erro relativo na energia e no momento total associado a resolução do sistema onda-partícula utilizando o integrador simplético proposto.

No contexto da tese, o próximo passo seria a resolução das equações linearizadas do pro-

blema,

$$\begin{aligned}
\delta \dot{x}_r &= \delta p_r \\
\delta \dot{p}_r &= -\varepsilon \sum_{s=1}^M \beta_s [\delta Y_s \cos(\kappa_s x_r) - Y_s \sin(\kappa_s x_r) \kappa_s \delta x_r + \delta X_s \sin(\kappa_s x_r) + X_s \cos(\kappa_s x_r) \kappa_s \delta x_r] \\
\delta \dot{X}_j &= \omega_j \delta Y_j + \varepsilon \beta_j \sum_{s=1}^N [\cos(\kappa_j x_s) \delta x_s] \\
\delta \dot{Y}_j &= -\omega_j \delta X_j - \varepsilon \beta_j \sum_{s=1}^N [\sin(\kappa_j x_s) \delta x_s],
\end{aligned} \tag{A.4}$$

juntamente com a implementação dos métodos para o cálculo dos expoentes de Lyapunov (mapa tangente e método dos clones). No entanto, constatou-se, durante estudos envolvendo mais partículas e mais ondas, que certas escolhas de parâmetros geram resultados questionáveis, com erros assumindo valores consideráveis. Entendida tal sensibilidade do método, seria possível prosseguir os estudos sobre caos, por exemplo analisando como o expoente de Lyapunov escala com o tamanho do sistema e investigar as regiões mais sensíveis do espaço de fases.

APÊNDICE B

Algoritmos

Algorithm *Cálculo dos expoentes de Lyapunov, mapa tangente*

Input N : Número de partículas

t_0 : Tempo de evolução a partir do qual se iniciam os cálculos para estimar o expoentes de Lyapunov

K : Número de vezes que procedimento para o cálculo do expoente é repetido

T : Tempo entre os processos de ortonormalização

h : Tamanho do passo do integrador

$lyap_num$: Número de expoentes a ser calculado

$samples$: Número de amostras

Output λ_0 : Maior expoente de Lyapunov

λ_1 : Segundo maior expoente de Lyapunov

⋮

λ_{2N-1} : Menor expoente de Lyapunov

1 { \mathbf{x} vetor relacionado às que regem a dinâmica}

2 { \mathbf{w}_i conjunto de vetores relacionado às equações linearizadas}

3 Condições_iniciais(\mathbf{x})

4 Condições_iniciais(\mathbf{w}_i) {base ortonormal, ($i = 1, 2, \dots, lyap_num$)}

5 **for** t **from** 0 **to** t_0 **do**

6 integrador(\mathbf{x})

7 **end for**

```

8 for  $k$  from 0 to  $K$  do
9   for  $t$  from 0 to  $T$  do
10     integrador( $\mathbf{w}_i$ )
11     integrador( $\mathbf{x}$ )
12   end for
13   for  $i$  from 0 to  $lyap\_num$  do
14      $\lambda_{i+} = \ln|\mathbf{w}_i|$ 
15   end for
16   ortonormalização( $\mathbf{w}_i$ )
17 end for
18 for  $i$  from 0 to  $lyap\_num$  do
19    $\lambda_i = \frac{1}{KT} \cdot \lambda_i$ 
20 end for
end

```

Algorithm *Cálculo dos expoentes de Lyapunov, método dos clones*

Input N : Número de partículas

t_0 : Tempo de evolução a partir do qual se iniciam os cálculos para estimar o expoentes de Lyapunov

K : Número de vezes que procedimento para o cálculo do expoente é repetido

T : Tempo entre os processos de ortonormalização

h : Tamanho do passo do integrador

$lyap_num$: Número de expoentes a ser calculado

ε : tamanho da perturbação

$samples$: Número de amostras

Output λ_0 : Maior expoente de Lyapunov

λ_1 : Segundo maior expoente de Lyapunov

⋮

λ_{2N-1} : Menor expoente de Lyapunov

```

1 {x vetor relacionado às que regem a dinâmica}
2 {xci conjunto de vetores relacionado às cópias perturbadas}
3 {wi conjunto de vetores relacionado às direções das perturbações}

4 Condições_iniciais(x)
5 Condições_iniciais(wi) {base ortonormal, (i = 1, 2, ..., lyap_num)}
6 for t from 0 to t0 do
7     integrador(x)
8 end for
9 for i from 0 to lyap_num do
10     xci = x + ε · wi
11 end for
12 for k from 0 to K do
13     for t from 0 to T do
14         integrador(xci)
15         integrador(x)
16         wi = x - xci
17     end for
18     for i from 0 to lyap_num do
19         λi+ = ln  $\frac{|\mathbf{w}_i|}{\varepsilon}$ 
20     end for
21     ortonormalização(wi)
22     for i from 0 to lyap_num do
23         xci = x + ε · wi
24     end for

```

```

25 end for
26 for i from 0 to lyap_num do
27      $\lambda_i = \frac{1}{KT} \cdot \lambda_i$ 
28 end for
end

```

Algorithm *Condições_iniciais*(\mathbf{w}_i)

```

1 {RAN  $\equiv$  número aleatório entre 0 e 1}
2 for i from 0 to lyap_num do
3     for j from 0 to 2N do
4          $w_i[j] = 2 * \text{RAN} - 1$ 
5     end for
6 end for
7 for i from 0 to lyap_num do
8     for j from 0 to 2N do
9          $w_i[j] = \frac{1}{|\mathbf{w}_i|} \cdot w_i[j]$ 
10    end for
11 end for
end

```

Algorithm *ortonormalização*(\mathbf{w}_i)

```

1 { $\mathbf{v}_i$ , vetores backup}
2 { $\mathbf{u}_i$ , base de vetores ortonormalizados}
3 for i from 0 to lyap_num do
4      $\mathbf{v}_i = \mathbf{w}_i$ 
5 end for
6 for i from 0 to 2N do

```

```
7    $u_0[i] = \frac{1}{|v_0|} \cdot v_0[i]$ 
8   end for
9   for  $i$  from 1 to  $lyap\_num$  do
10  for  $j$  from 0 to  $j < i$  do
11  for  $l$  from 0 to  $2N$  do
       $v_i[l] = \langle \mathbf{w}_i, \mathbf{u}_j \rangle \cdot u_j[l]$ 
12  end for
13  end for
14  for  $j$  from 0 to  $2N$  do
15   $u_i[j] = \frac{1}{|v_i|} \cdot v_i[j]$ 
16  end for
17 end for
18 for  $i$  from 1 to  $lyap\_num$  do
19   $\mathbf{w}_i = \mathbf{u}_i$ 
20 end for
end
```


APÊNDICE C

Rotinas computacionais relativas à dinâmica do HMF

C.1 Evolução e cálculo dos expoentes de Lyapunov utilizando o método tangente

Corpo principal do programa

```
1 /*Programa desenvolvido pelo grupo de Fisica estatistica do instituto de Fisica da Universidade de Brasilia, Calculo dos
   expoentes de Lyapunov para o HMF, metodo tangente.*/
2 #include <stdio.h>
3 #include <stdlib.h>
4 #include <math.h>
5 //#include "initial_conditions_eq.c"
6 #include "initial_conditions_wb.c"
7 #include "initial_conditions_tm.c"
8 #include "energy.c"
9 #include "momentum.c"
10 #include "integrator_step.c"
11 #include "integrator_step_tm.c"
12 #include "lyap_calculo.c"
13 #include "plot.c"
14 #include "delta_I_distribution.c"
15 #include "check_phi_w.c"
16 #include "bissecacao.c"
17
18 int main()
19 {
20     FILE *fout;
21     FILE *fout2;
22     FILE *fout3;
23     FILE *fout4;
24     FILE *fout5;
25     FILE *fin;
26
27     int i, j, l, L, N, t0, tmax, frame, lyap_num, k, samples;
28     double h, E0, E, P0, P, M0, M, M2, mx, my, time, phi0, phi, w, bac;
29     char filename[20];
30
31     fout = fopen("./output/energy.dat", "w");
32     fout2 = fopen("./output/momentum.dat", "w");
33     fout3 = fopen("./output/lyapunov_exponent.dat", "w");
34     fout4 = fopen("./output/lyapunov_exponent_samples.dat", "a");
35     fout5 = fopen("./output/mag.dat", "w");
```

```

36     fin = fopen("parameters.dat", "r");
37
38     fscanf(fin,"%d \n", &N);
39     fscanf(fin,"%d \n", &t0);
40     fscanf(fin,"%d \n", &tmax);
41     fscanf(fin,"%d \n", &frame);
42     fscanf(fin,"%d \n", &k);
43     fscanf(fin,"%lf \n", &h);
44     fscanf(fin,"%d \n", &lyap_num);
45     fscanf(fin,"%lf \n", &M0);
46     fscanf(fin,"%lf \n", &E0);
47     fscanf(fin,"\n%d \n", &samples);
48
49     printf("\n\nN = %d\n", N);
50     printf("t0 = %d \n", t0);
51     printf("tmax = %d \n", tmax);
52     printf("frame = %d \n", frame);
53     printf("k = %d \n", k);
54     printf("h = %g \n", h);
55     printf("lyap_num = %d \n", lyap_num);
56     printf("M0 = %g \n",M0);
57     printf("e_entrada = %g \n",E0);
58     printf("samples = %d\n", samples);
59
60     double *q = (double*)malloc(N*sizeof(double));
61     double *p = (double*)malloc(N*sizeof(double));
62     double *F = (double*)malloc(N*sizeof(double));
63     double *delta_I = (double*)malloc(N*sizeof(double));
64     double *norma_v=(double*)malloc(lyap_num*sizeof(double));
65     double *lambda=(double*)malloc(lyap_num*sizeof(double));
66     double *norma_delta2=(double*)malloc(lyap_num*sizeof(double));
67     double **delta=(double**)malloc((2*N)*sizeof(double));
68     double **v=(double**)malloc((2*N)*sizeof(double));
69     double **u=(double**)malloc((2*N)*sizeof(double));
70
71     for(i=0; i<2*N; i++)
72     {
73         delta[i] = (double*) malloc(lyap_num * sizeof(double));
74         v[i] = (double*) malloc(lyap_num * sizeof(double));
75         u[i] = (double*) malloc(lyap_num * sizeof(double));
76     }
77
78     for(L = 0; L < samples; L++)
79     {
80         printf("\nsample %d\n", L + 1);
81
82         for(l = 0; l < lyap_num; l++)
83             lambda[l] = 0.;
84
85         initial_conditions_wb ( N, q, p, M0, E0 );
86         //initial_conditions_eq ( N, q, p, M0, E0 );
87
88         initial_conditions_tm ( N, lyap_num, delta );
89
90         E0 = energy( N, q, p );
91         P0 = momentum( N, q, p );
92
93
94         mx = my =0.;
95         for(i = 0; i < N; i++)

```



```

96     {
97         mx += cos(q[i]);
98         my += sin(q[i]);
99     }
100
101     mx = mx/( (double)N );
102     my = my/( (double)N );
103     M2 = mx*mx + my*my;
104     M = sqrt(M2);
105
106     printf("\n\ntempo = %g \n", 0.);
107     printf("M = %g \n", M);
108     printf("E0 = %g \n",E0);
109     printf("P0 = %g \n\n",P0);
110
111     for( i = 0; i < t0; i++ )
112         integrator_step (N, h, q, p, F, &mx, &my );
113
114     E0 = energy( N, q, p );
115     P0 = momentum( N, q, p );
116     M2 = mx*mx + my*my;
117     M = sqrt(M2);
118     phi0 = atan2(my,mx);
119
120     printf("tempo = %g \n", h*t0);
121     printf("M = %g \n", M);
122     printf("E0 = %g \n",E0);
123     printf("P0 = %g \n\n",P0);
124     printf("phi0 = %g \n\n",phi0);
125
126     for( i = 1; i <= tmax; i++ )
127     {
128         time = (double)i*(double)k*h;
129
130         for(j = 0; j < k; j++) //integrator
131         {
132             integrator_step (N, h, q, p, F, &mx, &my );
133
134             for(l = 0; l < lyap_num; l++)
135                 integrator_step_tm ( N, h, q, p, F, mx, my, l, delta );
136         }
137
138         if ( L == samples -1 && i%100 == 0)
139         {
140             E = energy( N, q, p );
141             P = momentum( N, q, p );
142
143             fprintf(fout, "%g\n", (E-E0)/E0);
144             //fprintf(fout2, "%g\n", (P-P0)/P0);
145             fprintf(fout2, "%g\n", P);
146
147             M2 = mx*mx + my*my;
148             M = sqrt(M2);
149
150             fprintf(fout5, "%g\n", M);
151         }
152
153         //check_phi_w( mx, my, &M, &M2, &phi0, &phi, &w, h );
154
155         /*

```

```

156         if ( L == samples - 1 && i == tmax )
157         {
158             delta_I_distribution ( N, delta, delta_I );
159             plot ( figname, frame, i, N, q, p, h, M, phi, w );
160         }
161     */
162
163     lyap_calculo ( N, delta , v, u, lambda, lyap_num, norma_v );
164
165     if ( i%1000 == 0 )
166         fprintf ( fout3, "%g %lf\n", time, lambda[0]/time );
167     }
168
169     fprintf ( fout3, "\n");
170     printf("\nlambda = %g\n\n", lambda[0]/time);
171
172     fprintf(fout4, "%g %g\n", E0, lambda[0]/time);
173 }
174
175 fclose(fout);
176 fclose(fout2);
177 fclose(fout3);
178 fclose(fout4);
179 fclose(fout5);
180 fclose(fin);
181
182 return 0;
183 }

```

Integrador Numérico *Leap-Frog*

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <math.h>
4
5  #include "force.c"
6
7  void integrator_step(int N, double h, double q[], double p[], double f[], double *mx, double *my )
8  {
9      int i;
10
11     for( i = 0; i < N; i++)
12         q[i] = q[i] + p[i]*h/2.;
13
14     force( N, q, f, &*mx, &*my );
15
16     for( i=0; i < N; i++ )
17     {
18         p[i]=p[i] + f[i]*h;
19         q[i]=q[i] + p[i]*h/2.;
20     }
21
22     return;
23 }

```

Cálculo da força caso não linear

```

1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <math.h>
4
5 void force(int N, double q[], double f[], double *mx, double *my)
6 {
7     int i, j;
8
9     *mx = *my = 0.;
10    for( i = 0; i < N; i++ )
11    {
12        *mx += cos(q[i]);
13        *my += sin(q[i]);
14    }
15
16    *mx = *mx/(double)N;
17    *my = *my/(double)N;
18    for( i = 0; i < N; i++ )
19        f[i] = cos(q[i])*(*my) - sin(q[i])*(*mx);
20
21    return;
22 }

```

Cálculo da força para as equações linearizadas

```

1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <math.h>
4
5 void force_tm(int N, double q[], double f[], double mx, double my, int l, double **delta)
6 {
7     int i;
8     double delta_mx, delta_my;
9
10    delta_mx = delta_my = 0.;
11    for( i = 0; i < N; i++ )
12    {
13        delta_mx += sin(q[i])*delta[i][l];
14        delta_my += cos(q[i])*delta[i][l];
15    }
16
17    delta_mx /= -(double)N;
18    delta_my /= (double)N;
19    for( i = 0; i < N; i++ )
20        f[i] = (-mx*cos(q[i]) -my*sin(q[i]))*delta[i][l] +cos(q[i])*delta_my -sin(q[i])*delta_mx;
21
22    return;
23 }

```

Método de ortonormalização

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <math.h>
4
5  double proj(int N, double **delta, double **u, int i, int j)
6  {
7      int l;
8      double bac = 0.;
9
10     for( l = 0; l < 2*N; l++ )
11         bac += delta[l][i]*u[l][j];
12
13     return bac;
14 }
15
16 void lyap_calculo(int N, double **delta , double **v, double **u, double lambda[], int lyap_num, double norma_v[])
17 {
18
19     int i,j,l;
20     double bac;
21
22     for( i = 0; i < lyap_num; i++ )
23         norma_v[i] = 0.;
24
25     for( i = 0; i < lyap_num; i++ )
26         for( j = 0; j < 2*N; j++ )
27             v[j][i] = delta[j][i];
28
29     for( i = 0; i < 2*N; i++ )
30         norma_v[0] += v[i][0]*v[i][0];
31
32     norma_v[0] = sqrt(norma_v[0]);
33
34     for( i = 0; i < 2*N; i++ )
35         u[i][0] = v[i][0]/norma_v[0];
36
37     for( i=1; i < lyap_num; i++ )
38     {
39         for( j = 0; j < i; j++ )
40         {
41             bac = proj( N, delta, u, i, j );
42
43             for( l = 0; l < 2*N; l++ )
44                 v[l][i] -= bac*u[l][j];
45         }
46
47         for( j = 0; j < 2*N; j++ )
48             norma_v[i] += v[j][i]*v[j][i];
49
50         norma_v[i] = sqrt(norma_v[i]);
51
52         for( j = 0; j < 2*N; j++ )
53             u[j][i] = v[j][i]/norma_v[i];
54     }
55
56     for( i=0; i < lyap_num; i++ )
57         lambda[i] += log(norma_v[i]);
58

```

```

59     for( i = 0; i < lyap_num; i++ )
60         for( j=0; j<2*N; j++ )
61             delta[j][i] = u[j][i];
62
63     return;
64 }

```

C.2 Evolução e cálculo dos expoentes de Lyapunov utilizando o método dos clones

Corpo principal do programa

```

1  /*Programa desenvolvido pelo grupo de Fisica estatistica do instituto de Fisica da Universidade de Brasilia, Calculo dos
   expontes de Lyapunov para o HMF, metodo dos clones.*/
2  #include <stdio.h>
3  #include <stdlib.h>
4  #include <math.h>
5  //#include "initial_conditions_eq.c"
6  #include "initial_conditions_wb.c"
7  #include "initial_conditions_cl.c"
8  #include "integrator_step.c"
9  #include "integrator_step_cl.c"
10 #include "energy.c"
11 #include "momentum.c"
12 #include "lyap_calculo.c"
13 #include "plot.c"
14 #include "deltaI_distribution.c"
15 #include "corrector_initial_conditions.c"
16 #include "check_phi_w.c"
17 #include "bissecacao.c"
18
19 int main()
20 {
21     FILE *fout;
22     FILE *fout2;
23     FILE *fout3;
24     FILE *fout4;
25     FILE *fout5;
26     FILE *fin;;
27
28     int i, j, l, L, N, t0, tmax, frame, lyap_num, k, samples;
29     double h, E0, E, P0, P, M0, M, M2, mx, my, time, phi0, phi, w, bac, epsilon;
30
31     fout = fopen("./output/energy.dat", "w");
32     fout2 = fopen("./output/momentum.dat", "w");
33     fout3 = fopen("./output/lyapunov_exponent.dat", "w");
34     fout4 = fopen("./output/lyapunov_exponent_samples.dat", "a");
35     fout5 = fopen("./output/mag.dat", "w");
36     fin = fopen("parameters.dat", "r");
37
38     fscanf(fin,"%d \n", &N);
39     fscanf(fin,"%d \n", &t0);

```

```

40     fscanf(fin,"%d \n", &tmax);
41     fscanf(fin,"%d \n", &frame);
42     fscanf(fin,"%lf \n", &h);
43     fscanf(fin,"%d \n", &k);
44     fscanf(fin,"%d \n", &lyap_num);
45     fscanf(fin,"%lf \n", &M0);
46     fscanf(fin,"%lf \n", &E0);
47     fscanf(fin,"%lf \n", &epsilon);
48     fscanf(fin,"\n%d \n", &samples);
49
50     printf("\nN = %d \n",N);
51     printf("t0 = %d \n",t0);
52     printf("tmax = %d \n",tmax);
53     printf("frame = %d \n",frame);
54     printf("h = %g \n",h);
55     printf("k = %d \n",k);
56     printf("lyap_num = %d \n",lyap_num);
57     printf("M0 = %g \n",M0);
58     printf("e_entrada = %g \n",E0);
59     printf("epsilon = %g \n", epsilon );
60     printf("samples = %d\n", samples);
61
62     double *q=(double*)malloc(N*sizeof(double));
63     double *p=(double*)malloc(N*sizeof(double));
64     double *f=(double*)malloc(N*sizeof(double));
65     double *norma_v=(double*)malloc(lyap_num*sizeof(double));
66     double *lambda=(double*)malloc(lyap_num*sizeof(double));
67     double *norma_delta2=(double*)malloc(lyap_num*sizeof(double));
68     double **delta=(double**)malloc((2*N)*sizeof(double));
69     double **v=(double**)malloc((2*N)*sizeof(double));
70     double **u=(double**)malloc((2*N)*sizeof(double));
71
72     for(i=0; i<2*N; i++)
73     {
74         delta[i] = (double*) malloc(lyap_num * sizeof(double));
75         v[i] = (double*) malloc(lyap_num * sizeof(double));
76         u[i] = (double*) malloc(lyap_num * sizeof(double));
77     }
78
79     double **delta_I = (double**)malloc(N*sizeof(double));
80
81     for(i = 0; i < N; i++)
82         delta_I[i] = (double*) malloc(lyap_num * sizeof(double));
83
84     double **qc = (double**)malloc((N)*sizeof(double));
85     double **pc = (double**)malloc((N)*sizeof(double));
86
87     for(i=0; i<N; i++)
88     {
89         qc[i] = (double*) malloc(lyap_num * sizeof(double));
90         pc[i] = (double*) malloc(lyap_num * sizeof(double));
91     }
92
93     for(L = 0; L < samples; L++)
94     {
95         printf("\nsample %d\n", L + 1);
96
97         for(l = 0; l < lyap_num; l++)
98             lambda[l] = 0.;
99

```

```

100  initial_conditions_wb( N, q, p, M0, E0 );
101  //corrector_initial_conditions( N, q, p );
102
103  E0 = energy( N, q, p );
104  P0 = momentum( N, q, p );
105
106  mx = my =0.;
107  for(i = 0; i < N; i++)
108  {
109      mx += cos(q[i]);
110      my += sin(q[i]);
111  }
112
113  mx = mx/( (double)N );
114  my = my/( (double)N );
115
116  M2 = mx*mx + my*my;
117  M = sqrt(M2);
118
119  printf("\n\ntempo = %g \n", 0.);
120  printf("M = %g \n", M);
121  printf("E0 = %g \n",E0);
122  printf("P0 = %g \n\n",P0);
123
124  for(i=0; i< t0; i++)
125      integrator_step( N, h, q, p, f, &mx, &my );
126
127  E0 = energy( N, q, p );
128  P0 = momentum( N, q, p );
129
130  mx = my =0.;
131  for(i = 0; i < N; i++)
132  {
133      mx += cos(q[i]);
134      my += sin(q[i]);
135  }
136
137  mx = mx/( (double)N );
138  my = my/( (double)N );
139
140  M2 = mx*mx + my*my;
141  M = sqrt(M2);
142
143  printf("\n\ntempo = %g \n", t0*h);
144  printf("M = %g \n", M);
145  printf("E0 = %g \n",E0);
146  printf("P0 = %g \n\n",P0);
147
148  initial_conditions_cl( N, q, p, qc, pc, lyap_num, delta, epsilon );
149
150  for( i=0; i < lyap_num; i++ )
151      lambda[i] = 0.;
152
153  for(i = 1; i < tmax; i++)
154  {
155      time = (double)i*(double)k*h;
156
157      for(j = 0; j < k; j++) //integrator
158      {
159          integrator_step( N, h, q, p, f, &mx, &my );

```

```

160
161         for(l = 0; l < lyap_num; l++)
162             integrator_step_cl( N, h, q, p, qc, pc, f, l, delta );
163     }
164
165     if ( L == samples -1 && i%100 == 0)
166     {
167         E = energy( N, q, p );
168         P = momentum( N, q, p );
169
170         fprintf(fout, "%g\n", (E-E0)/E0);
171         //fprintf(fout2, "%g\n", (P-P0)/P0);
172         fprintf(fout2, "%g\n", P);
173
174         M2 = mx*mx + my*my;
175         M = sqrt(M2);
176
177         fprintf(fout5, "%g\n", M);
178     }
179
180     lyap_calculo( N, delta , v, u, lambda, lyap_num, norma_v, q, p, qc, pc, epsilon );
181
182     if ( i%1000 == 0)
183         fprintf ( fout3, "%g %lf\n", time, lambda[0]/time );
184 }
185
186 printf("\nlambda = %g\n\n", lambda[0]/time);
187
188 fprintf ( fout3, "\n");
189 //fprintf(fout4, "%g %g\n", log10(N), log10(lambda[0]/time));
190 //fprintf(fout4, "%g\n", lambda[0]/time);
191 fprintf(fout4, "%g\n", lambda[0]/time);
192
193 //for(j = 0; j < lyap_num; j++) //integrator
194 //fprintf(fout4, "%g\n", lambda[j]/time);
195 }
196
197 fclose(fout);
198 fclose(fout2);
199 fclose(fout3);
200 fclose(fout4);
201 fclose(fout5);
202 fclose(fin);
203
204 return 0;
205 }

```

Integração das equações clone e determinação das variações δq e δp

```

1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <math.h>
4
5 #include "force_cl.c"
6
7 void integrator_step_cl( int N, double h, double q[], double p[], double **qc, double **pc, double f[], int l, double **delta
    )

```



```

8 {
9     int i;
10
11     for( i=0; i < N; i++ )
12         qc[i][1] += pc[i][1]*h/2.;
13
14     force_cl( N, qc, f, l, delta );
15
16     for( i = 0; i < N; i++ )
17     {
18         pc[i][1] += f[i]*h;
19         qc[i][1] += pc[i][1]*h/2.;
20     }
21
22     for( i = 0; i < N; i++ )
23     {
24         delta[i][1] = q[i] - qc[i][1];
25         delta[i + N][1] = p[i] - pc[i][1];
26     }
27
28     return;
29 }

```

Método de ortonormalização

```

1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <math.h>
4
5 #include "proj.c"
6
7 void lyap_calculo(int N, double **delta , double **v, double **u, double lambda[], int lyap_num, double norma_v[], double q
8     [], double p[],
9     double **qc, double **pc, double epsilon)
10 {
11     int i,j,l;
12     double bac;
13
14     for( i = 0; i < lyap_num; i++ )
15         norma_v[i] = 0.;
16
17     for( i = 0; i < lyap_num; i++ )
18         for( j = 0; j < 2*N; j++ )
19             v[j][i] = delta[j][i];
20
21     for( i = 0; i < 2*N; i++ )
22         norma_v[0] += v[i][0]*v[i][0];
23
24     norma_v[0] = sqrt(norma_v[0]);
25
26
27     for( i = 0; i < 2*N; i++ )
28         u[i][0] = v[i][0]/norma_v[0];
29
30     for( i=1; i < lyap_num; i++ )
31     {

```

```

32     for( j = 0; j < i; j++ )
33     {
34         bac = proj( N, delta, u, i, j );
35
36         for( l = 0; l < 2*N; l++ )
37             v[l][i] -= bac*u[l][j];
38     }
39
40     for( j = 0; j < 2*N; j++ )
41         norma_v[i] += v[j][i]*v[j][i];
42
43     norma_v[i] = sqrt(norma_v[i]);
44
45     for( j = 0; j < 2*N; j++ )
46         u[j][i] = v[j][i]/norma_v[i];
47     }
48
49     for( i=0; i < lyap_num; i++ )
50         lambda[i] += log(norma_v[i]/epsilon);
51
52     for( i = 0; i < lyap_num; i++ )
53         for( j=0; j<2*N; j++ )
54             delta[j][i] = u[j][i];
55
56     for(i = 0; i < lyap_num; i++)
57         for( j = 0; j < N; j++)
58             qc[j][i] = q[j] + epsilon*delta[j][i];
59
60     for(i = 0; i < lyap_num; i++)
61         for( j = 0; j < N; j++)
62             pc[j][i] = p[j] + epsilon*delta[j+N][i];
63
64     return;
65 }

```

APÊNDICE D

Rotinas computacionais relativas à dinâmica do modelo de folhas autogravitantes

D.1 Evolução e cálculo dos expoentes de Lyapunov

Corpo principal do programa

```
1  /*Programa desenvolvido pelo grupo de Fisica estatistica do instituto de Fisica da Universidade de Brasilia, Calculo dos
   expoentes de Lyapunov para o modelo de folhas autogravitantes, metodo tangente.*/
2  #include <stdio.h>
3  #include <stdlib.h>
4  #include <math.h>
5  #include "initial_conditions.c"
6  #include "energy.c"
7  #include "momentum.c"
8  #include "temperature.c"
9  #include "plot.c"
10 #include "colision_time.c"
11 #include "progress_bar.c"
12 #include "initial_conditions_tm.c"
13 #include "integrator_step_tm.c"
14 #include "lyap_calculo.c"
15
16 int main()
17 {
18     FILE *fout;
19     FILE *fout2;
20     FILE *fout3;
21     FILE *fout4;
22     FILE *fout5;
23     FILE *fout6;
24     FILE *fout7;
25     FILE *fin;
26
27     int i, j, l, N, t0, tmax, frame, lyap_num, baci, bacj, samples;
28     double h, E0, E, P0, P, time, tij, energy_input, KS;
29     char figname[20];
30
31     fout = fopen("./output/energy.dat", "w");
32     fout2 = fopen("./output/momentum.dat", "w");
33     fout3 = fopen("./output/temperature.dat", "w");
34     fout4 = fopen("./output/time.dat", "w");
35     fout5 = fopen("./output/bac.dat", "a");
36     fout6 = fopen("./output/LE.dat", "w");
37     fout7 = fopen("./output/LS.dat", "a");
38     fin = fopen("parameters.dat", "r");
```

```

39
40     fscanf(fin,"%d \n", &N);
41     fscanf(fin,"%d \n", &t0);
42     fscanf(fin,"%d \n", &tmax);
43     fscanf(fin,"%d \n", &frame);
44     fscanf(fin,"%lf \n", &energy_input);
45     fscanf(fin,"%d \n", &lyap_num);
46     fscanf(fin,"%d \n", &samples);
47
48     printf("\n\nN = %d\n", N);
49     printf("t0 = %d \n", t0);
50     printf("tmax = %d \n", tmax);
51     printf("frame = %d \n", frame);
52     printf("energy_input = %g \n", energy_input);
53     printf("lyap_num = %d \n", lyap_num);
54     printf("samples = %d \n", samples);
55
56     double *q = (double*)malloc(N*sizeof(double));
57     double *p = (double*)malloc(N*sizeof(double));
58     double *F = (double*)malloc(N*sizeof(double));
59     //double *delta_I = (double*)malloc(N*sizeof(double));
60     double *norma_v=(double*)malloc(lyap_num*sizeof(double));
61     double *lambda=(double*)malloc(lyap_num*sizeof(double));
62     double *norma_delta2=(double*)malloc(lyap_num*sizeof(double));
63     double **delta=(double**)malloc((2*N)*sizeof(double));
64     double **v=(double**)malloc((2*N)*sizeof(double));
65     double **u=(double**)malloc((2*N)*sizeof(double));
66
67     for(i=0; i<2*N; i++)
68     {
69         delta[i] = (double*) malloc(lyap_num * sizeof(double));
70         v[i] = (double*) malloc(lyap_num * sizeof(double));
71         u[i] = (double*) malloc(lyap_num * sizeof(double));
72     }
73
74     for(l = 0; l < samples; l++)
75     {
76         for(i = 0; i < lyap_num; i++)
77             lambda[i] = 0.;
78
79         initial_conditions( N, q, p, energy_input );
80         initial_conditions_tm ( N, lyap_num, delta );
81
82         E0 = energy( N, q, p );
83         P0 = momentum( N, q, p );
84
85         printf("\n\ntempo = %g \n", 0.);
86         printf("E0 = %g \n", E0);
87         printf("P0 = %g \n\n", P0);
88
89         time = 0.;
90         for( i = 0; i < t0; i++ )
91         {
92             colision_time( N, q, p, F, &tij, &baci, &bacj );
93             time += tij;
94
95             if (i%1 == 0 && l == 0)
96                 fprintf(fout3,"%g\n", temperature( N, p ) );
97         }
98

```

```

99     E0 = energy( N, q, p );
100     P0 = momentum( N, q, p );
101
102     printf("\n\ntempo = %g \n", time);
103     printf("E0 = %g \n", E0);
104     printf("P0 = %g \n\n", P0);
105
106     time = 0.;
107     for( i = 1; i <= tmax; i++ )
108     {
109         colision_time( N, q, p, E, &tij, &baci, &bacj );
110         time += tij;
111
112         for(j = 0; j < lyap_num; j++)
113             integrator_step_tm( N, p, j, delta, tij, baci, bacj );
114
115         lyap_calculo ( N, delta , v, u, lambda, lyap_num, norma_v );
116
117         if (i%500 == 0 && l == 0)
118         {
119             fprintf ( fout6,"%g %lf\n", time, lambda[0]/time );
120
121             //plot( figname, frame, i, N, q, p, h);
122
123             E = energy( N, q, p );
124             P = momentum( N, q, p );
125
126             fprintf(fout, "%g\n", (E-E0)/E0);
127             fprintf(fout2, "%g\n", (P-P0)/P0);
128             //fprintf(fout2, "%g\n", P);
129
130             fprintf(fout4, "%g %g\n", tij, time);
131         }
132     }
133
134
135     fprintf(fout5, "%g\n", lambda[0]/time);
136     printf( "\n\ncolisions = %d,      time = %g,      lambda = %g\n\n", tmax, time, lambda[0]/time );
137
138     KS = 0.;
139     for(i=0; i < lyap_num; i++)
140         KS += lambda[i]/time;
141
142     //fprintf( fout7, "%d %g\n", N, KS/( (double)N ) );
143     fprintf( fout7, "%g\n", KS/( (double)N ) );
144 }
145
146 //for(i=0; i < lyap_num; i++)
147 //fprintf(fout7, "%g\n", lambda[i]/time);
148
149 fclose(fout);
150 fclose(fout2);
151 fclose(fout3);
152 fclose(fout4);
153 fclose(fout5);
154 fclose(fout6);
155 fclose(fout7);
156 fclose(fin);
157
158 return 0;

```

159 }

Determinação dos tempos de colisão utilizados na implementação do *event-driven*

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <math.h>
4
5  #include "force.c"
6
7  void colision_time(int N, double q[], double p[], double F[], double *tij, int *baci, int *bacj)
8  {
9      int i, j, r, s;
10     double qij, pij, Fij, delta[2], t[4], bac, bac_time, time_min, sigma;
11
12     force ( N, q, F );
13
14     sigma = 1.e-10;
15     time_min = 1.e10;
16
17     for( i = 0; i < N-1; i++ )
18         for( j = i+1; j < N; j++ )
19             {
20                 qij = q[i] - q[j];
21                 pij = p[i] - p[j];
22
23                 Fij = F[i] - F[j];
24
25                 delta[0] = pij*pij - 2.*Fij*qij;
26                 delta[1] = pij*pij - 2.*Fij*qij;
27
28                 if( delta[0] >= 0. )
29                     {
30                         t[0] = ( -pij + sqrt(delta[0]) )/Fij;
31                         t[1] = ( -pij - sqrt(delta[0]) )/Fij;
32                     }
33
34                 else
35                     t[0] = t[1] = -1.;
36
37
38                 if( delta[1] >= 0. )
39                     {
40                         t[2] = ( -pij + sqrt(delta[1]) )/Fij;
41                         t[3] = ( -pij - sqrt(delta[1]) )/Fij;
42                     }
43
44                 else
45                     t[2] = t[3] = -1.;
46
47                 for( r = 0; r < 4; r++ )
48                     for( s = r+1; s < 4; s++ )
49                         {
50                             if( t[s] < t[r] )
51                                 {
52                                     bac = t[r];
53                                     t[r] = t[s];

```



```

25         NO += 1;
26     }
27 }
28
29     F[i] = ( (double)N1 - (double)N0 ) / ( (double)N );
30 }
31
32 for( i = 0; i < N; i++ )
33 {
34     p[i] = p[i] + F[i]*h;
35     q[i] = q[i] + p[i]*h/2.;
36 }
37
38 return;
39 }

```

Integração da dinâmica caso linearizado

```

1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <math.h>
4
5 void integrator_step_tm(int N, double p[], int l, double **delta, double tij, int baci, int bacj )
6 {
7     int i;
8
9     for( i = 0; i < N; i++ )
10         delta[i][l] += delta[i+N][l]*tij;
11
12     delta[baci+N][l] += 2.*( (delta[bacj][l] - delta[baci][l]) / ( (double)N*fabs(p[bacj] - p[baci]) ) );
13     delta[bacj+N][l] += 2.*( (delta[baci][l] - delta[bacj][l]) / ( (double)N*fabs(p[baci] - p[bacj]) ) );
14
15     return;
16 }

```


Referências Bibliográficas

- [1] ALLEN, M. P., AND TILDESLEY, D. J. *Computer Simulation of Liquids*. Oxford University Press, 1990.
- [2] ANTONI, M., AND RUFFO, S. Clustering and relaxation in hamiltonian long-range dynamics. *Phys. Rev. E* 52 (Sep 1995), 2361–2374.
- [3] ANTONIAZZI, A., ELSKENS, Y., FANELLI, D., AND RUFFO, S. Statistical mechanics and vlasov equation allow for a simplified hamiltonian description of single-pass free electron lasersaturated dynamics. *The European Physical Journal B - Condensed Matter and Complex Systems* 50 (2006), 603–611.
- [4] ANTONIAZZI, A., FANELLI, D., BARRÉ, J., CHAVANIS, P.-H., DAUXOIS, T., AND RUFFO, S. Maximum entropy principle explains quasistationary states in systems with long-range interactions: The example of the hamiltonian mean-field model. *Phys. Rev. E* 75 (2007), 011112.
- [5] BACKUS, J. In *History of Programming Languages I*. ACM, 1981, ch. The History of Fortran I, II, and III.
- [6] BALDOVIN, F., BRIGATTI, E., AND TSALLIS, C. Quasi-stationary states in low-dimensional hamiltonian systems. *Physics Letters A* 320, 4 (2004), 254 – 260.
- [7] BASHE, C., JOHNSON, L., PALMER, J., AND PUGH, E. *IBM's Early Computers*. MIT Press, 1985.
- [8] BILL BRYSON, I. K. *Breve História de Quase Tudo*. Companhia das Letras, 2005.
- [9] BOYCE, W. E., AND DIPRIMA, R. C. *Equações Diferenciais e Problemas de Valores de Contorno*. ed. Rio de Janeiro: LTC, 2012.

- [10] CAMPA, A., DAUXOIS, T., AND RUFFO, S. Statistical mechanics and dynamics of solvable models with long-range interactions. *Physics Reports* 480 (2009), 57 – 159.
- [11] CASSANDRAS, C. G., AND LAFORTURE, S. *Discrete Event Systems*. Springer, 2008.
- [12] DA SILVA JÚNIOR, M. F. P. *Comparação de métodos computacionais para o estudo termodinâmico de sistemas com interações de longo alcance*. PhD thesis, Universidade de Brasília, 2016.
- [13] DAUXOIS, T., LATORA, V., RAPISARDA, A., RUFFO, S., AND TORCINI, A. The hamiltonian mean field model: From dynamics to statistical mechanics and back. *Springer* 602 (2002), 458–487.
- [14] DE SOUZA MACIEL, J. M. B. *Estudo de Modelos Simplificados com Interações de Longo Alcance no Ensemble Microcanônico*. PhD thesis, Universidade de Brasília, 2015.
- [15] DE WIJN, A. S. Kolmogorov-sinai entropy for dilute systems of hard particles in equilibrium. *Phys. Rev. E* 71 (2005), 046211.
- [16] ELSKENS, Y., AND ESCANDE, D. *Microscopic Dynamics of Plasmas and Chaos*. Taylor & Francis, 2002.
- [17] FIRPO, M.-C. Analytic estimation of the lyapunov exponent in a mean-field model undergoing a phase transition. *Phys. Rev. E* 57 (1998), 6599–6603.
- [18] GRIFFITHS, D. *Introduction to Quantum Mechanics*. Pearson international edition. Pearson Prentice Hall, 2005.
- [19] HASSANI, S. *Mathematical Physics: A Modern Introduction to Its Foundations*. Springer, 1999.
- [20] JOYCE, M., AND WORRAKITPOONPON, T. Quasistationary states in the self-gravitating sheet model. *Phys. Rev. E* 84 (2011), 011139.

- [21] KARUNADASA, N. P., AND RANASINGHE, D. N. Accelerating high performance applications with cuda and mpi. In *2009 International Conference on Industrial and Information Systems (ICIIS)* (2009), pp. 331–336.
- [22] LAFFARGUE, T., LAM, K.-D. N. T., KURCHAN, J., AND TAILLEUR, J. Large deviations of lyapunov exponents. *Journal of Physics A: Mathematical and Theoretical* 46, 25 (2013), 254002.
- [23] LEMOS, N. *Mecânica Analítica*. Livraria da Física, 2007.
- [24] LORENZ, E. N. Deterministic nonperiodic flow. *Journal of the Atmospheric Sciences* 20 (1963), 130–141.
- [25] LYNDEN-BELL, D. Statistical mechanics of violent relaxation in stellar systems. *Monthly Notices of the Royal Astronomical Society* 136 (1967), 101.
- [26] MANOS, T., AND RUFFO, S. Scaling with system size of the lyapunov exponents for the hamiltonian mean field model. *Transport Theory and Statistical Physics* (2011), 360–381.
- [27] MANOS, T., AND RUFFO, S. Scaling with system size of the lyapunov exponents for the hamiltonian mean field model. *Transport Theory and Statistical Physics* 40 (2011), 360–381.
- [28] MARÃO, J., LIU, X., AND FIGUEIREDO, A. Using impulses to control the convergence toward invariant surfaces of continuous dynamical systems. *Chaos, Solitons & Fractals* 45 (2012), 1067–1079.
- [29] MILANOVIĆ, L., POSCH, H. A., AND THIRRING, W. Statistical mechanics and computer simulation of systems with attractive positive power-law potentials. *Phys. Rev. E* 57 (1998), 2763–2775.
- [30] MILANOVIĆ, L., AND POSCH, H. A. Statistical mechanics and computer simulation of systems with attractive positive power-law potentials. *Phys. Rev. E* 57 (1998), 2763–2775.
- [31] MONTEIRO, L. *Sistemas dinâmicos*. Livraria da Física, 2006.

- [32] PIKOVSKY, A., AND POLITI, A. *Lyapunov Exponents*. Cambridge University Press, 2016.
- [33] PRESS, W. H., TEUKOLSKY, S. A., VETTERLING, W. T., AND FLANNERY, B. P. *Numerical Recipes in C*. Cambridge University Press, 1988.
- [34] RIBEIRO, B. V. *Abordagens Dinâmica e Estocástica na Descrição de Processos Físicos na Física de Plasmas*. PhD thesis, Universidade de Brasília, 2014.
- [35] RIZZATO, F. B., PAKTER, R., AND LEVIN, Y. Driven one-component plasmas. *Phys. Rev. E* 80 (2009), 021109.
- [36] ROCHA FILHO, T. M., AMATO, M. A., AND FIGUEIREDO, A. A novel approach to the determination of equilibrium properties of classical hamiltonian systems with long-range interactions. *Journal of Physics A: Mathematical and Theoretical* 42, 16 (2009), 165001.
- [37] ROCHA FILHO, T. M., AMATO, M. A., AND FIGUEIREDO, A. Nonequilibrium phase transitions and violent relaxation in the hamiltonian mean-field model. *Phys. Rev. E* 85 (2012), 062103.
- [38] SALINAS, S. *Introdução a Física Estatística*. EDUSP, 1997.
- [39] SORIANO, D. C., FAZANARO, F. I., SUYAMA, R., DE OLIVEIRA, J. R., ATTUX, R., AND MADRID, M. K. A method for lyapunov spectrum estimation using cloned dynamics and its application to the discontinuously-excited fitzhugh–nagumo model. *Nonlinear Dynamics* 67 (2012), 413–424.
- [40] STANISCIA, F., CHAVANIS, P. H., DE NINNO, G., AND FANELLI, D. Out-of-equilibrium phase re-entrance(s) in long-range interacting systems. *Phys. Rev. E* 80 (2009), 021138.
- [41] STROGATZ, S. H. *Nonlinear dynamics and chaos : with applications to physics, biology, chemistry, and engineering*. Westview Press, 1994.

- [42] TSUCHIYA, T., AND GOUDA, N. Relaxation and lyapunov time scales in a one-dimensional gravitating sheet system. *Phys. Rev. E* 61 (2000), 948–951.
- [43] TSUCHIYA, T., KONISHI, T., AND GOUDA, N. Quasiequilibria in one-dimensional self-gravitating many-body systems. *Phys. Rev. E* 50 (1994), 2607–2615.
- [44] VENAILLE, A., DAUXOIS, T., AND RUFFO, S. Violent relaxation in two-dimensional flows with varying interaction range. *Phys. Rev. E* 92 (2015), 011001.
- [45] YAMAGUCHI, Y. Y. Slow relaxation at critical point of second order phase transition in a highly chaotic hamiltonian system. *Progress of Theoretical Physics* 95, 4 (1996), 717–731.
- [46] YAMAGUCHI, Y. Y. One-dimensional self-gravitating sheet model and lynden-bell statistics. *Phys. Rev. E* 78 (2008), 041114.

