

UNIVERSIDADE DE SÃO PAULO
INSTITUTO DE MATEMÁTICA, ESTATÍSTICA E CIÊNCIA DA COMPUTAÇÃO
BACHARELADO EM MATEMÁTICA APLICADA E COMPUTACIONAL COM
HABILITAÇÃO EM MÉTODOS MATEMÁTICOS

**Uma abordagem estocástica para modelos
rápidos-lentos**

Lucas Amaral Taylor

MONOGRAFIA FINAL
MAP 2429 — TRABALHO DE
FORMATURA EM MATEMÁTICA APLICADA

Supervisor: Prof. Dr. Breno Raphaldini Ferreira da Silva

São Paulo
2025

*O conteúdo deste trabalho é publicado sob a licença CC BY 4.0
(Creative Commons Attribution 4.0 International License)*

Folha de Aprovação

Trabalho de Conclusão de Curso apresentado ao Instituto de Matemática, Estatística e Ciência da Computação da Universidade de São Paulo (IME-USP) para obtenção do título de Bacharel em Matemática Aplicada e Computacional.

Aprovado em: 12 de dezembro de 2025.

Banca Examinadora

Prof. Dr. Breno Raphaldini Ferreira da Silva

Orientador – IME-USP

Prof. Dr. Clodoaldo Grotta Ragazzo

IME-USP

Prof. Dr. Lucas Franceschini

IME-USP

*Na verdade, na verdade vos digo que, se o
grão de trigo, caindo na terra, não morrer,
fica ele só; mas se morrer, dá muito fruto.*

João 12:24

Agradecimentos

Primeiramente, gostaria de agradecer ao professor Breno Raphaldini Ferreira da Silva pela ótima orientação, desafios cativantes e pela confiança em todo processo. Além disso, gostaria de agradecer ao apoio do professor Honghu Liu por sua contribuição no desenvolvimento deste trabalho.

Estendo meu agradecimento a todos os professores da graduação que me proporcionaram uma nova forma de entender o mundo, da compreensão de fenômenos físicos à processos abstratos. Foi um privilégio estar presente nas aulas cativantes e ter contato com ideias tão belas e absolutamente maravilhosas.

Quero também agradecer à minha mãe e meu pai por todo apoio incondicional durante este período tão importante, às amizades que o BMAC ofereceu pelo caminho, em especial: Ana, Gustavo, Julio, Manuella e Matheus, e aos demais amigos sempre presentes.

Por fim, é necessário voltar ao começo, professor Mauricio Osca Souza Pereira, ou “Phalkon”, obrigado pelas histórias do IME foram elas que me trouxeram até aqui.

Resumo

Lucas Amaral Taylor. **Uma abordagem estocástica para modelos rápidos-lentos.**

Monografia (Bacharelado). Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2025.

O presente trabalho tem como objetivo estudar a aproximação de sistemas dinâmicos rápidos-lentos por meio de equações diferenciais estocásticas, com ênfase no modelo de Lorenz 80. Como principal referência metodológica, este trabalho segue a abordagem proposta no artigo “*Stochastic rectification of fast oscillations on slow manifold closures*” ([CHEKROUN, LIU e MCWILLIAMS, 2021](#)).

Palavras-chave: Equações diferenciais estocásticas. Modelo de Lorenz 80. Sistemas dinâmicos rápidos-lentos.

Abstract

Lucas Amaral Taylor. **A stochastic approach to fast-slow models.** Capstone Project Report (Bachelor). Institute of Mathematics and Statistics, University of São Paulo, São Paulo, 2025.

This undergraduate thesis aims to study the approximation of fast-slow dynamical systems using stochastic differential equations, with an emphasis on the Lorenz 80 model. As its main methodological reference, this work follows the approach proposed in the article “*Stochastic Rectification of Fast Oscillations on Slow Manifold Closures*” ([CHEKROUN, LIU, and MCWILLIAMS, 2021](#)).

Keywords: Stochastic differential equations. Lorenz 80 model. Fast-slow dynamical systems.

Listas de figuras

1.1	Diagrama do modelo de água-rasa adaptado	6
1.2	Evolução temporal das variáveis x_1, y_1, z_1	12
1.3	Projeções do modelo PE para $f_1 = 0.3027$	13
1.4	Projeções do modelo QG para $f_1 = 0.3027$	14
1.5	Projeções do modelo BE para $f_1 = 0.3027$	15
1.6	Comparação das séries temporais de y para os modelos PE, BE, QG . . .	16
2.1	Simulação do preço de uma ação a partir da EDE (2.3)	23
3.1	Comparação entre o modelo determinístico e o estocástico para diferentes valores de ϵ histogramas (esq.) e séries temporais (dir.).	27
3.2	Comparação dos modelos estocásticos gerados com $\epsilon = 0.1$ e $\epsilon = 0.2$. . .	28
3.3	Evolução temporal da parte real e imaginária	31
3.4	Projeção da parte real e imaginária	31
3.5	Comparação das projeções dos modelos PE e BE-SLO no plano (y_3, y_2) .	34
3.6	Comparação das projeções dos modelos PE e BE-SLO no plano (y_1, y_2) .	34
3.7	Comparação das projeções dos modelos PE e BE-SLO no plano (y_1, y_3) .	35
3.8	Comparação das séries temporais de y para os modelos PE e BE-SLO . .	35
C.1	Fluxograma do Processo de Simulação	62

Listas de tabelas

1.1	Comparação das amplitudes para cada modelo.	16
3.1	Valores de ε e dos respectivos coeficientes de difusão σ	26

Sumário

Introdução	1
1 O modelo de Lorenz 80 determinístico	3
1.1 Introdução	3
1.2 Breves considerações sobre dinâmica de fluidos geofísicos	3
1.3 O modelo de água-rasa	4
1.4 Construção dos modelos	5
1.4.1 Modelo PE	6
1.4.2 O modelo QG	9
1.4.3 O modelo BE	9
1.5 Simulações	12
1.5.1 Simulações do modelo PE	12
1.5.2 Simulações do modelo QG	14
1.5.3 Simulações do modelo BE	14
2 Introdução às equações diferenciais estocásticas	17
2.1 Motivação	17
2.2 Considerações estatísticas	18
2.3 Equações diferenciais estocásticas	19
2.3.1 Processo de Wiener	20
2.3.2 Integral estocástica	21
2.4 Simulação de uma equação diferencial estocásticas	23
3 Aproximação de sistemas determinísticos usando EDEs	25
3.1 Introdução	25
3.2 <i>Fast Chaotic Noise</i>	25
3.3 Lorenz 80 BE-SLO	28
3.3.1 Variedades e escolha do modelo BE	29
3.3.2 Oscilações estocásticas de Stuart-Landau	30

3.3.3	Ressonâncias de Ruelle-Pollicot	32
3.3.4	Construção do modelo BE-SLO	32
3.3.5	Simulações e comparação com o modelo PE	34
4	Conclusões	37
A	Formalismo Mori-Zwanzig	39
A.1	Introdução	39
A.2	Motivação	39
A.3	Preliminares	40
A.3.1	Convertendo sistemas de EDO não lineares em sistemas de EDPs lineares	40
A.3.2	Notação de semigrupo	40
A.3.3	Polinômios Hermitianos	41
A.4	Mori-Zwanzig	42
A.4.1	Construção	42
A.4.2	Análise termo a termo	44
B	Programas	47
B.1	Lorenz 80 determinístico	47
B.2	Simulação de preço de uma ação de mercado	50
B.3	<i>Fast Chaotic Noise</i>	50
B.4	Oscilação de Stuart Landau estocástica	53
B.5	Lorenz 80 BE-SLO	53
C	Simulações computacionais	57
C.1	Introdução	57
C.2	<i>Solvers</i>	57
C.2.1	RK45 em <i>Python</i>	57
C.2.2	Euler-Maruyama em <i>Julia</i>	58
C.3	Decisões de Implementação e Observações	60
C.3.1	Linguagem <i>Julia</i>	60
C.3.2	Estrutura geral dos programas de simulação	61
C.3.3	Sistema integrado para diferentes versões do modelo L80	61
C.3.4	Código de geração do ruído do Lorenz 80 BE-SLO	62
Referências		63

Introdução

Em 1980, Edward Norton Lorenz publicou o artigo intitulado *Attractor Sets and Quasi-Geostrophic Equilibrium* (LORENZ, 1980), no qual propôs um modelo composto por variáveis rápidas e lentas, com o objetivo de estudar a dinâmica de sistemas atmosféricos forçados e dissipativos. Em 2021, foi publicado o artigo *Stochastic Rectification of Fast Oscillations on Slow Manifold Closures*, que introduz uma abordagem estatística por meio de equações diferenciais estocásticas e do formalismo de Mori-Zwanzig (CHEKROUN, LIU e MCWILLIAMS, 2021). Este trabalho tem como objetivo estudar modelos rápidos-lentos por meio de uma abordagem estocástica, tomando como principal referência CHEKROUN, LIU e MCWILLIAMS (2021).

Para isso, organizamos o trabalho da seguinte forma: no Capítulo 1, apresentamos o modelo Lorenz 80 em sua versão determinística. Discutimos os aspectos teóricos e a construção do modelo com suas três principais abordagens: modelo PE, modelo QG e modelo BE. Realizamos simulações computacionais dos três modelos e compararmos seus comportamentos por meio de representações gráficas.

No Capítulo 2, introduzimos as equações diferenciais estocásticas (EDEs), explorando a motivação para seu estudo, aspectos teóricos fundamentais e um exemplo simples com simulação computacional.

Em seguida, no Capítulo 3, abordamos a aplicação das EDEs na aproximação de sistemas determinísticos. Para isso, utilizamos um exemplo simplificado, retirado de PAVLIOTIS e STUART (2008), complementado por simulações computacionais. Revisitamos também o modelo Lorenz 80 em sua formulação estocástica, discutindo os fundamentos teóricos que motivam sua construção, além de realizar simulações e compará-lo ao modelo PE apresentado inicialmente. Por fim, no último capítulo, sintetizamos as conclusões do trabalho e discutimos o conteúdo aprendido ao longo do estudo.

O trabalho conta ainda com três apêndices. No Apêndice A, introduzimos o leitor ao formalismo de Mori-Zwanzig, apresentando os fundamentos teóricos desse método físico-estatístico. Nos Apêndices B e C, detalhamos os programas utilizados nas simulações e aspectos do desenvolvimento computacional, respectivamente.

Por fim, seguindo boas práticas no uso de inteligência artificial, para este trabalho foram utilizadas as ferramentas *Clarice.ai* para o aperfeiçoamento do texto e revisão gramatical, e o *GitHub Copilot* para o desenvolvimento dos códigos.

Capítulo 1

O modelo de Lorenz 80 determinístico

1.1 Introdução

Este capítulo tem como objetivo apresentar a formulação determinística do modelo Lorenz 80. Iniciamos, na seção 1.2, com uma introdução a conceitos fundamentais da geofísica, que servirão de base para a modelagem proposta.

Na seção 1.3, é descrito o modelo de água rasa, do qual derivamos a formulação original do modelo de Lorenz 80, conhecida como modelo PE. A construção detalhada desse modelo é apresentada na seção 1.4, seguida das simplificações que levam aos modelos QG e BE. Por fim, na seção 1.5, realizamos simulações computacionais para cada um dos modelos, acompanhadas de representações gráficas e análise comparativa dos resultados.

1.2 Breves considerações sobre dinâmica de fluidos geofísicos

Nesta seção, reunimos um breve glossário com os principais conceitos de geofísica que servem de base para a compreensão do modelo de Lorenz 80. Todas as definições expostas abaixo estão detalhadas em [VALLIS \(2017\)](#).

- **Convecção.** Convecção é um processo de transferência de calor que ocorre em fluidos. Esse fenômeno envolve o movimento do próprio fluido e a transferência energia térmica de uma região para outra.
- **Parâmetro de Coriolis.** A *força de Coriolis* é uma quasi-força (ou pseudo-força) que surge devido à rotação da Terra. Quando analisamos o movimento de um corpo em um referencial rotativo, esse corpo parece sofrer a ação de uma força que desvia sua trajetória. Esse desvio é quantificado pelo parâmetro de Coriolis, definido pela expressão:

$$f = 2\Omega \sin(\theta)$$

onde Ω representa a velocidade angular de rotação da Terra e θ é a latitude, ou seja, o ângulo entre a posição do ponto e o equador terrestre.

- **Número de Rossby.** O número de Rossby é a razão entre a magnitude da aceleração relativa e a aceleração de Coriolis. É aproximado por:

$$Ro \equiv \frac{U}{fL},$$

onde U é a magnitude aproximada da velocidade horizontal e L é uma escala de comprimento e f é o parâmetro de Coriolis.

- **Equilíbrio hidrostático.** Matematicamente, a equação do equilíbrio hidrostático é dada por:

$$\frac{\partial p}{\partial z} = -\rho_0 g, \quad (1.1)$$

onde: p é a pressão do fluido, z é a coordenada vertical, ρ_0 é a densidade constante do fluido e g é a aceleração da gravidade.

- **Conservação de massa.** Em um escoamento de fluido, a densidade pode variar de acordo com o tempo ou a posição. No entanto, a quantidade total de massa do fluido permanece constante. Esse princípio estabelece que a massa não pode ser criada nem perdida durante o movimento.
- **Equações quasi-geostróficas.** As equações quasi-geostróficas são equações amplamente usadas em estudos teóricos da atmosfera e oceano. Elas atendem as seguintes características:
 1. O número de Rossby é pequeno;
 2. A escala do movimento não é significativamente maior do que a escala de deformação;
 3. As variações no parâmetro de Coriolis são pequenas;
 4. As escalas de tempo são advectivas, ou seja, $T = L/U$.
- **Condições de Hardley.** A circulação de Hadley é um padrão atmosférico típico da região entre o Trópico de Câncer e o Trópico de Capricórnio. Nessa faixa tropical, o ar quente sobe próximo ao equador, se desloca em altitude para latitudes maiores e desce, formando um ciclo convectivo. As condições que regem este sistema, será utilizado em nossa simulação como as condições ideias da atmosfera, referida como *condições de Hardley*.

1.3 O modelo de água-rasa

O modelo de água rasa descreve um fluido de densidade constante, em equilíbrio hidrostático, que pode ou não estar em rotação. Nele, a escala horizontal é significativamente maior que a profundidade. Esse fluido possui superfície livre e é limitado pelas bordas (VALLIS, 2017). No caso considerado, adotamos a versão de uma única camada.

Para a construção do modelo de água-rasa, consideramos a equação do equilíbrio hidrostático, expressa em (1.1). A partir das manipulações envolvendo os conceitos de momento e conservação de massa, detalhado em [VALLIS \(2017\)](#), obtemos as equações que descrevem o modelo:

$$\begin{aligned} \frac{\partial V}{\partial t} + (V \cdot \nabla) V + f \mathbf{k} \times V &= -g \nabla \eta \\ \frac{\partial \eta}{\partial t} + \nabla \cdot (\eta V) &= 0 \end{aligned} \quad (1.2)$$

Onde:

- t : tempo;
- \mathbf{r} : vetor de posição inicial;
- $V(t)$: Velocidade horizontal;
- $\eta(t)$: altura da superfície;
- \mathbf{k} : vetor da vertical.

1.4 Construção dos modelos

Edward Norton Lorenz (1917-2008) foi um importante matemático e meteorologista responsável pela publicação de vários artigos com desenvolvimento de modelos na área de meteorologia e geofísica.

O mais famoso deles foi o artigo “*Deterministic Nonperiodic Flow*”, publicado em 1963 [LORENZ \(1963\)](#). Nele, Lorenz desenvolveu um modelo matemático simplificado para a convecção atmosférica, composto por três equações diferenciais ordinárias, expressas abaixo:

$$\begin{cases} \frac{dx}{dt} = \sigma(y - x) \\ \frac{dy}{dt} = x(\rho - z) - y, \\ \frac{dz}{dt} = xy - \beta z \end{cases} \quad (1.3)$$

onde σ é o *número de Prandtl*, que regula a sensibilidade entre x e y ; ρ é o *número de Rayleigh*, associado à magnitude da convecção; e β está ligado à geometria da célula de convecção, influenciando a relação entre as taxas de x e z .

O modelo acima, conhecido como Lorenz 63, é um sistema determinístico desenvolvido para representar sistemas hidrodinâmicos ideais e dissipativos de força. O Lorenz 63 tornou-se amplamente conhecido por sua alta sensibilidade às condições iniciais — pequenas alterações nas variáveis x_0 , y_0 e z_0 podem levar a trajetórias completamente distintas no espaço de fases. Essa sensibilidade extrema é uma característica caótica do modelo.

Em 1980, Lorenz publica o artigo intitulado “*Attractor Sets and Quasi-Geostrophic Equilibrium*” ([LORENZ, 1980](#)). Neste artigo, Lorenz apresenta a construção e a simulação de dois modelos distintos: o primeiro, é formado a partir das equações primitivas (PE) com

nove EDOs (equações diferenciais ordinárias), derivado das equações de águas rasas com topografia e forçamento, enquanto o segundo é um modelo quasi-geostrófico (QG) com 3 EDOs, obtido ao descartar as variáveis associadas ao escoamento divergente x e seus termos correspondentes. O modelo PE contém tanto ondas gravitacionais rápidas quanto oscilações quasi-geostróficas lentas, enquanto o modelo QG mantém apenas estas últimas, em um quadro simplificado para atmosfera de latitudes médias.

Por fim, o modelo BE (*balanced equations*) foi introduzido em 1982 no artigo “*Intermediate Model Solutions to the Lorenz Equations: Strange Attractors and Other Phenomena*” (GENT e McWILLIAMS, 1982) e é uma simplificação do modelo PE realizada através dos truncamentos das equações de vorticidade e divergência.

A seguir, apresentaremos a construção de cada um dos modelos apresentados acima.

1.4.1 Modelo PE

Consideremos um fluido homogêneo e incompressível, ou seja, com densidade constante em todo o volume e volume invariável mesmo sob variações de pressão. O escoamento é predominantemente horizontal, descrito por uma velocidade $V(t, \mathbf{r})$ independente da altura, onde \mathbf{r} representa o vetor de posição inicial.

A componente vertical da velocidade é determinada pela continuidade de massa. A superfície livre do fluido está localizada na altura $H + z(t, \mathbf{r})$, onde H representa a profundidade média e a base se apoia sobre uma topografia variável $h(\mathbf{r})$. Temos também que $h(\mathbf{r})$ e $z(t, \mathbf{r})$ possuem média zero.

O sistema está sujeito à rotação planetária, com um parâmetro de Coriolis constante f . Tanto o campo de velocidades V quanto a elevação da superfície z sofrem dissipação difusiva, associada a movimentos de pequena escala: o termo v representa o coeficiente de difusão viscosa (dissipação de momento) e κ representa o coeficiente de difusão térmica. O modelo também inclui um termo de forçamento externo $F(\mathbf{r})$ e, por fim, adota-se a hipótese de equilíbrio hidrostático.

A partir da descrição acima, podemos construir o seguinte diagrama:

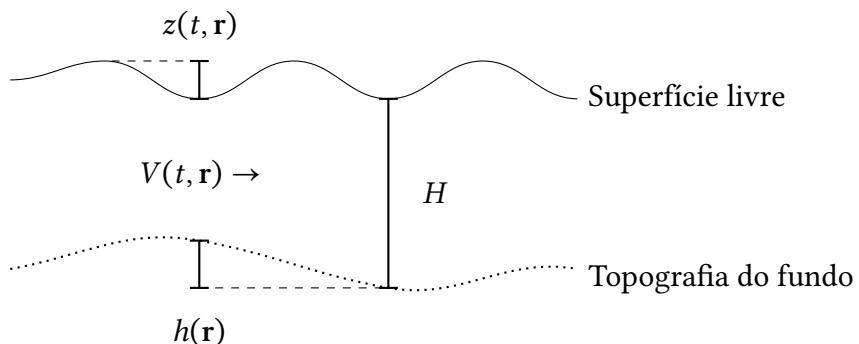


Figura 1.1: Diagrama do modelo de água-rasa adaptado

Além disso, o modelo de água-rasa adaptado é expresso por:

$$\frac{\partial V}{\partial t} = -(V \cdot \nabla) V - f \mathbf{k} \times V - g \nabla z + \nu \nabla^2 V \quad (1.4)$$

$$\frac{\partial z}{\partial t} = -(V \cdot \nabla)(z - h) - (H + z - h) \nabla \cdot V + \kappa \nabla^2 z + F \quad (1.5)$$

Onde:

- H : profundidade média do fluido;
- $h(\mathbf{r})$: variação da superfície topológica;
- $V(t, \mathbf{r})$: Velocidade horizontal;
- $z(t, \mathbf{r})$: altura da superfície;
- F : forças externas;
- κ : coeficiente de difusão térmica;
- ν : coeficiente de difusão viscosa;

Em seguida, aplicando o *Teorema de Helmholtz*¹ à equação (1.4), escrevendo

$$V = \nabla \chi + \mathbf{k} \times \nabla \psi,$$

onde χ é o potencial de velocidade associado à parte divergente e ψ a função corrente associada à parte rotacional. Dessa forma, $\nabla^2 \chi$ representa a divergência e $\nabla^2 \psi$ a vorticidade. Substituindo essa decomposição obtemos:

$$\begin{aligned} \frac{\partial \nabla^2 \chi}{\partial t} &= -\frac{1}{2} \nabla^2 (\nabla \chi \cdot \nabla \chi) - \nabla \chi \cdot \nabla (\nabla^2 \psi) \times \mathbf{k} + \nabla^2 (\nabla \chi \cdot \nabla \psi \times \mathbf{k}) \\ &\quad + \nabla \cdot (\nabla^2 \psi \nabla \psi) - \frac{1}{2} \nabla^2 (\nabla \psi \cdot \nabla \psi) + \nu \nabla^4 \chi + f \nabla^2 \psi - g \nabla^2 z, \end{aligned} \quad (1.6)$$

$$\frac{\partial \nabla^2 \psi}{\partial t} = -\nabla \cdot (\nabla^2 \psi \nabla \chi) - \nabla \psi \cdot \nabla (\nabla^2 \psi) \times \mathbf{k} - f \nabla^2 \chi + \nu \nabla^4 \psi. \quad (1.7)$$

Analogamente, aplicando (1.5), temos:

$$\frac{\partial z}{\partial t} = -\nabla \cdot [(z - h) \nabla \chi] - \nabla \psi \cdot \nabla (z - h) \times \mathbf{k} - H \nabla^2 \chi + \kappa \nabla^2 z + F. \quad (1.8)$$

Nosso objetivo é reduzir as equações (1.6)–(1.8) a um modelo de baixa ordem. Para

¹ O Teorema afirma que: “Um vetor é unicamente determinado ao se conhecer sua divergência e seu rotacional em uma região simplesmente conexa (sem buracos) e o seu componente normal sobre a fronteira” ([ARFKEN e WEBER, 2005](#)). A partir do Teorema, temos que campo vetorial suficientemente suave pode ser decomposto como

$$\mathbf{V} = \nabla \phi + \nabla \times \mathbf{A},$$

onde ϕ é o potencial escalar associado à divergência e \mathbf{A} é o potencial vetorial associado ao rotacional. Também referido como *decomposição de Helmholtz*.

isso, introduzimos três vetores adimensionais $\alpha_1, \alpha_2, \alpha_3$ que satisfazem

$$\alpha_1 + \alpha_2 + \alpha_3 = 0,$$

e adotamos as permutações cíclicas

$$(i, j, k) = (1, 2, 3), (2, 3, 1), (3, 1, 2). \quad (1.9)$$

Definimos então:

$$a_i = \alpha_i \cdot \alpha_i, \quad b_i = \alpha_j \cdot \alpha_k, \quad c = (b_1 b_2 + b_2 b_3 + b_3 b_1)^{1/2}.$$

Lorenz também apresenta uma forma alternativa, equivalente, mais conveniente para a implementação computacional:

$$b_i = \frac{1}{2}(a_i - a_j - a_k), \quad c_i = c.$$

Escolhido um comprimento característico L , construímos três funções ortogonais:

$$\phi_i(\mathbf{r}) = \cos\left(\alpha_i \cdot \frac{\mathbf{r}}{L}\right),$$

para as quais valem, por exemplo:

$$\begin{aligned} L^2 \nabla^2 \phi_i &= -a_i \phi_i, \\ L^2 \nabla \phi_i \cdot \nabla \phi_k &= -\frac{1}{2} b_{ik} \phi_i + \dots, \\ L^2 \nabla \cdot (\phi_j \nabla \phi_k) &= \frac{1}{2} b_{jk} \phi_i + \dots, \\ L^2 \phi_j \cdot \nabla \phi_k \times \mathbf{k} &= -\frac{1}{2} c_{jk} \phi_i + \dots, \end{aligned}$$

onde os termos omitidos são múltiplos de cossenos. Com essas funções, expandimos as variáveis em série e introduzimos escalas adimensionais:

$$\begin{aligned} t &= f^{-1} \tau, \\ \chi &= 2L^2 f^2 \sum_i x_i \phi_i, \\ \psi &= 2L^2 f^2 \sum_i y_i \phi_i, \\ z &= 2L^2 f^2 g^{-1} \sum_i z_i \phi_i, \\ h &= 2L^2 f^2 g^{-1} \sum_i h_i \phi_i, \\ F &= 2L^2 f^2 g^{-1} \sum_i F_i \phi_i. \end{aligned} \quad (1.10)$$

Substituindo as equações de (1.10) em (1.6)–(1.8), e projetando sobre a base $\{\phi_i\}$, obtemos

finalmente o modelo PE de baixa ordem, composto de nove equações diferenciais ordinárias:

$$a_i \frac{dx_i}{d\tau} = a_i b_i x_i x_k - c(a_i - a_k) x_i y_k + c(a_i - a_j) y_i x_k - 2c^2 y_i y_k - v_0 a_i^2 x_i + a_i y_i - a_i z_i, \quad (1.11)$$

$$a_i \frac{dy_i}{d\tau} = -a_i b_k x_i y_k - a_i b_i y_i x_k + c(a_k - a_i) y_i y_k - a_i x_i - v_0 a_i^2 y_i, \quad (1.12)$$

$$a_i \frac{dz_i}{d\tau} = -b_k x_i (z_k - h_k) - b_i (z_i - h_i) x_k + c y_i (z_k - h_k) - c(z_i - h_i) y_k + g_0 a_i x_i - \kappa_0 a_i z_i + F_i. \quad (1.13)$$

Temos que índices (i, j, k) devem ser interpretados como uma permutação cíclica de $(1, 2, 3)$, isto é, para cada equação fixa-se um índice $i \in \{1, 2, 3\}$, e os índices (j, k) são determinados ciclicamente como $(i, j, k) = (1, 2, 3), (2, 3, 1)$ ou $(3, 1, 2)$.

As variáveis x_i representam os modos divergentes do escoamento, sendo responsáveis pelas oscilações rápidas do tipo onda de gravidade, as variáveis y_i , por sua vez, correspondem aos modos rotacionais associados à vorticidade, caracterizando o componente quasi-geoestrófico da dinâmica, por fim, z_i são derivadas do campo de altura e atuam como variáveis auxiliares acopladas ao sistema ([LORENZ, 1980](#)).

No contexto da separação de escalas, x_i e z_i compõem o conjunto de variáveis rápidas, enquanto y_i corresponde às variáveis lentas do modelo ([CHEKROUN, LIU e McWILLIAMS, 2021](#)).

1.4.2 O modelo QG

Na construção do modelo QG, começamos desprezando todos os termos não lineares, assim como aqueles que envolvem as variáveis x , incluindo a derivada temporal, na equação (1.11). Fazemos o mesmo com os termos não lineares ou topográficos que dependem de x nas equações (1.12) e (1.13). Por fim, eliminamos as variáveis x e z , obtendo ao modelo QG apresentado a seguir:

$$(a_i g_0 + 1) \frac{dy_i}{d\tau} = g_0 c(a_k - a_j) y_j y_k - a_i (a_i g_0 v_0 + \kappa_0) y_i - c h_k y_j + c h_j y_k + F_i, \quad (1.14)$$

onde ainda são válidas as permutações apresentadas em (1.4.1).

1.4.3 O modelo BE

Como afirmado anteriormente, o modelo BE baseia-se em truncamentos das equações de vorticidade e divergência. No caso, na equação de vorticidade, o termo omitido é um termo de advecção vertical de modo que a equação de vorticidade e a equação de altura são exatas. No nosso caso, o truncamento é aplicado em (1.12) e (1.13), resultando na expressão:

$$a_i y_i - 2c^2 y_j y_k = a_i z_i. \quad (1.15)$$

Derivando a expressão acima, obtemos:

$$a_i \left(\frac{dy_i}{d\tau} - \frac{dz_i}{d\tau} \right) - 2c^2 \left(\frac{dy_j}{d\tau} y_k + y_j \frac{dy_k}{d\tau} \right) = 0$$

Aplicando em (1.12) e (1.13), obtemos:

$$\begin{aligned} & x_i [a_i a_j a_k (1 + g_0 a_i) - 2c^2 (a_j^2 b_j y_j^2 + a_k^2 b_k y_k^2)] \\ & - x_j [a_j a_k (y_k (2c^2 - a_k b_k) + a_j b_k (z_k - h_k)) + 2c^2 a_i b_j \dot{y}_j y_j] \\ & - x_k [a_i a_j a_k (y_j (2c^2 - a_j b_j) + a_i b_j (z_j - h_j)) + 2c^2 a_i b_k \dot{y}_k y_k] \\ & = a_j a_k [c(a_k - a_j) y_j y_k + c a_i ((z_j - h_j) y_k - y_j (z_k - h_k))] \\ & + a_i (v_0 a_i (z_i - y_i) - F_i) \\ & - 2c^2 [c a_j (a_j - a_i) y_j y_j^2 + c a_k (a_i - a_k) y_k y_k^2] \\ & - v_0 a_j a_k (a_j + a_k) y_j y_k. \end{aligned} \quad (1.16)$$

Novamente, assim como no caso anterior, ainda são válidas as permutações apresentadas em (1.4.1). Podemos representar de forma mais clara e compacta com o uso da função $\Phi(\mathbf{y})$, proposta em [CHEKROUN, LIU e McWILLIAMS \(2017\)](#), expressa por:

$$\Phi(\mathbf{y}) = (\Phi_1(\mathbf{y}), \Phi_2(\mathbf{y}), \Phi_3(\mathbf{y})) = [M(\mathbf{y}, G(\mathbf{y}))]^{-1} \begin{pmatrix} d_{1,2,3}(\mathbf{y}, G(\mathbf{y})) \\ d_{2,3,1}(\mathbf{y}, G(\mathbf{y})) \\ d_{3,1,2}(\mathbf{y}, G(\mathbf{y})) \end{pmatrix}. \quad (1.17)$$

onde,

$$M(\mathbf{y}, \mathbf{z}) \mathbf{x} = \begin{pmatrix} \Delta_{1,2,3}(\mathbf{y}) & \Gamma_{1,2,3}(\mathbf{y}, \mathbf{z}) & \Sigma_{1,2,3}(\mathbf{y}, \mathbf{z}) \\ \Sigma_{2,3,1}(\mathbf{y}, \mathbf{z}) & \Delta_{2,3,1}(\mathbf{y}) & \Gamma_{2,3,1}(\mathbf{y}, \mathbf{z}) \\ \Gamma_{3,1,2}(\mathbf{y}, \mathbf{z}) & \Sigma_{3,1,2}(\mathbf{y}, \mathbf{z}) & \Delta_{3,1,2}(\mathbf{y}) \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} d_{1,2,3}(\mathbf{y}, \mathbf{z}) \\ d_{2,3,1}(\mathbf{y}, \mathbf{z}) \\ d_{3,1,2}(\mathbf{y}, \mathbf{z}) \end{pmatrix}.$$

com

$$\begin{aligned} \Delta_{i,j,k}(\mathbf{y}) &= a_i a_j a_k (1 + g_0 a_i) - 2c^2 (a_j^2 b_j y_j^2 + a_k^2 b_k y_k^2), \\ \Gamma_{i,j,k}(\mathbf{y}, \mathbf{z}) &= - [a_j a_k (y_k (2c^2 - a_k b_k) + a_j b_k (z_k - h_k)) + 2c^2 a_i b_j \dot{y}_j y_j], \\ \Sigma_{i,j,k}(\mathbf{y}, \mathbf{z}) &= - [a_i a_j (y_j (2c^2 - a_j b_j) + a_i b_j (z_j - h_j)) + 2c^2 a_i b_k \dot{y}_k y_k], \\ d_{i,j,k}(\mathbf{y}, \mathbf{z}) &= a_j a_k [c(a_k - a_j) y_j y_k + c a_i ((z_j - h_j) y_k - y_j (z_k - h_k))] \\ &+ a_i (v_0 a_i (z_i - y_i) - F_i) \\ &- 2c^2 [c a_j (a_j - a_i) y_j y_j^2 + c a_k (a_i - a_k) y_k y_k^2] \\ &- v_0 a_j a_k (a_j + a_k) y_j y_k. \end{aligned}$$

Assim, a equação (1.4.3), pode ser expressa por:

$$a_i \frac{dy_i}{d\tau} = -a_k b_k \Phi_j(\mathbf{y}) y_k - a_j b_j \Phi_k(\mathbf{y}) + c(a_k - a_j) y_j y_k - a_i \Phi_i(\mathbf{y}) - v_0 a_i^2 y_i.$$

1.5 Simulações

Nesta seção, expor os resultados gráficos das simulações do Lorenz 80 determinístico com o objetivo de apresentar ao leitor o comportamento do modelo visualmente, bem como exibir características e propriedades de cada modelo. Para as simulações utilizamos a linguagem *Python*, em particular, bibliotecas *scipy*, *numpy* e *pandas*. O código utilizado está no apêndice B.1. Os parâmetros utilizados estão de acordo com LORENZ (1980) e CHEKROUN, LIU e MCWILLIAMS (2021), são: $\kappa = \nu = \frac{1}{48}$, $g_0 = 8 \text{ m} \cdot \text{s}^{-2}$, $a_1 = a_2 = 1$, $a_3 = 3$, $h_1 = -1$ e $h_2 = h_3 = f_2 = f_3 = 0$. Em relação a f_1 , testamos com dois valores $f_1 = 0.1$ e $f_1 = 0.3027$ que estão devidamente enunciados quando utilizados.

1.5.1 Simulações do modelo PE

Tomando $f_1 = 0.1$ e $x = y = z = (0.1, 0.1, 0.1)$. Para tais valores, obtivemos os seguintes resultados:

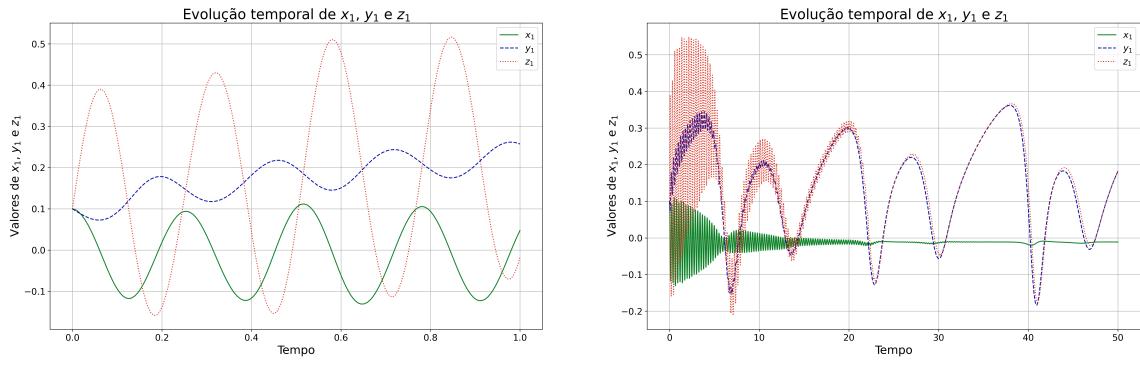


Figura 1.2: Evolução temporal das variáveis x_1, y_1, z_1

Inicialmente, no primeiro dia de simulação (figura 1.2a), observa-se que a variável x_1 apresenta amplitude superior à de y_1 , mas inferior à de z_1 . Além disso, seu período é ligeiramente menor que o das demais variáveis. Esse comportamento reflete a natureza de x_1 como variável rápida, associada às oscilações do tipo onda gravitacional.

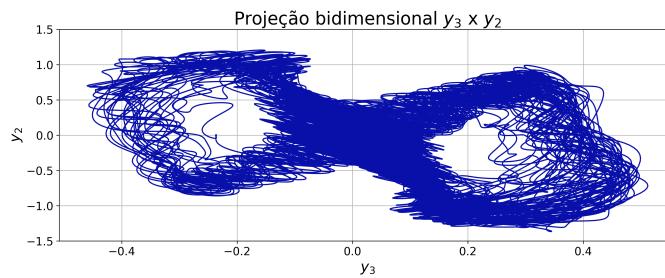
Na figura 1.2b, após 50 dias de simulação, nota-se uma redução significativa da amplitude de x_1 , acompanhada por oscilações mais rápidas e de menor intensidade. Isso é consistente com a dissipação progressiva das ondas gravitacionais, efeito esperado devido à presença de termos dissipativos no modelo.

Com relação às variáveis lentas y_1 e z_1 , percebe-se na figura 1.2a que z_1 possui a maior amplitude, enquanto y_1 apresenta a menor, ambas com períodos semelhantes. Já na figura 1.2b, observa-se que y_1 e z_1 passam a exibir comportamento oscilatório sincronizado, com amplitudes mais próximas. Esse acoplamento dinâmico evidencia a convergência das trajetórias do sistema para o atrator. Essa convergência e sincronização indicam que, ao longo do tempo, a solução se aproxima de um regime dominado por oscilações de baixa frequência e mais regulares, enquanto as componentes associadas a modos rápidos (como x_1) são dissipadas (LORENZ, 1980).

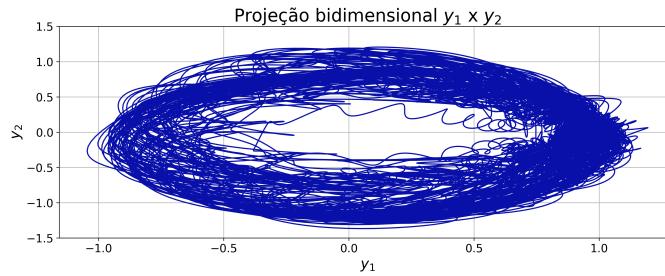
Para as projeções bidimensionais, utilizamos $f_1 = 0.3027$ e variáveis iniciais dadas pelas condições de Hadley expressas abaixo:

$$y_1 = \frac{f_1}{a_1 v_0 (1 + a_1 g_0)}, \quad x_1 = -v_0 a_1 y_1, \quad z_1 = y_1. \quad (1.18)$$

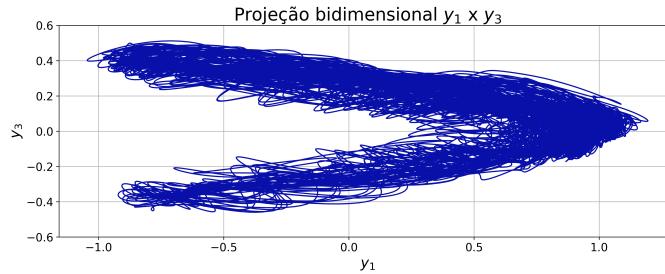
Além disso, adicionamos pequenas perturbações em y_2 e z_2 , dadas por $y_2 = -1 \times 10^{-5}$ e $z_2 = 1 \times 10^{-5}$, enquanto as demais variáveis foram fixadas em zero. As simulações foram realizadas ao longo de 400 dias, sendo descartados os 25% iniciais dos dados, correspondentes à fase transitória em que o sistema ainda está evoluindo até se ajustar ao formato do atrator. Assim, obtiveram-se os seguintes resultados:



(a) Projeção no plano (y_3, y_2).



(b) Projeção no plano (y_1, y_2).



(c) Projeção no plano (y_1, y_3).

Figura 1.3: Projeções do modelo PE para $f_1 = 0.3027$.

O principal objetivo destas simulações projetadas bidimensionalmente é evidenciar o comportamento de atrator caótico característico do sistema. Note que, através da imagem, pode-se evidenciar a aperiodicidade do sistema, junto com o fato de que não há um ponto de convergência quando $t \rightarrow \infty$. Por fim, ponto importante é que as trajetórias permanecem confinadas em uma região limitada do espaço de fases, não divergindo para o infinito, caracterizando o atrator.

1.5.2 Simulações do modelo QG

Para as simulações do modelo QG, utilizamos as mesmas condições utilizadas para as projeções do modelo PE, isto é, $f = 0.3027$ e as condições de Harley para a variável y_1 expressa em (1.18) e uma pequena perturbação em $y_2 = 1 \times 10^{-5}$. Simulação com t em 400 dias e com limpeza de dados de 25%.

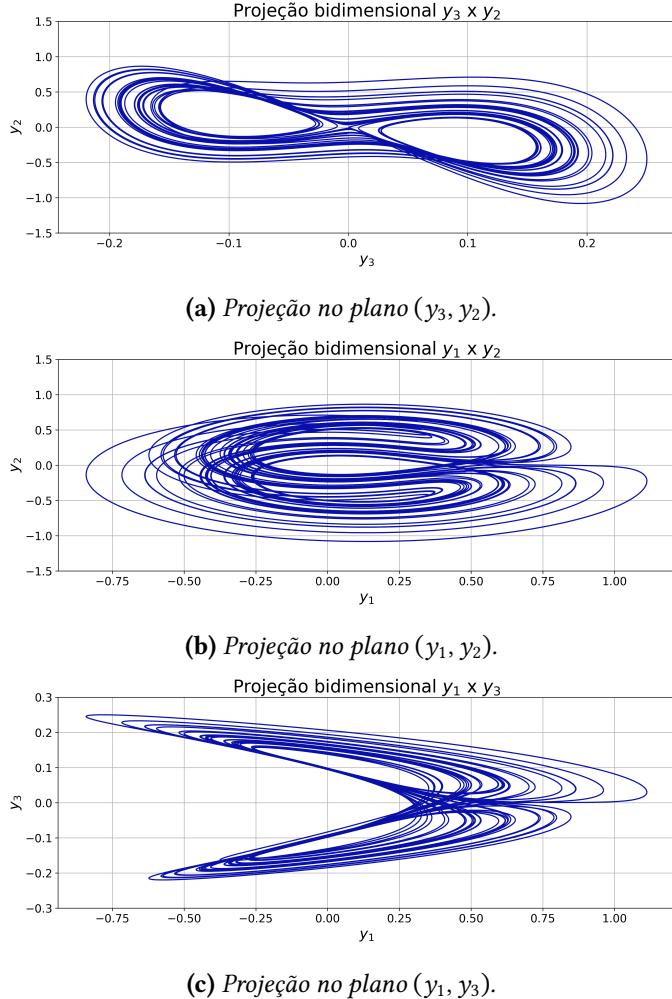


Figura 1.4: Projeções do modelo QG para $f_1 = 0.3027$.

Nas projeções mostradas nas Figuras 1.4a–1.4b, percebe-se a formação de trajetórias regulares e confinadas, indicando uma forte correlação entre as variáveis y_i . Além disso, o modelo QG apresenta um regime visivelmente mais organizado. Ainda assim, o sistema mantém comportamento caótico: não há convergência para um ponto fixo quando $t \rightarrow \infty$, e o regime estacionário exibe um atrator estranho.

1.5.3 Simulações do modelo BE

Para o modelo BE, utilizamos os mesmos parâmetros e condições iniciais utilizadas nas projeções dos modelos anteriores, ainda com 400 dias e 25% dos dados iniciais eliminados.

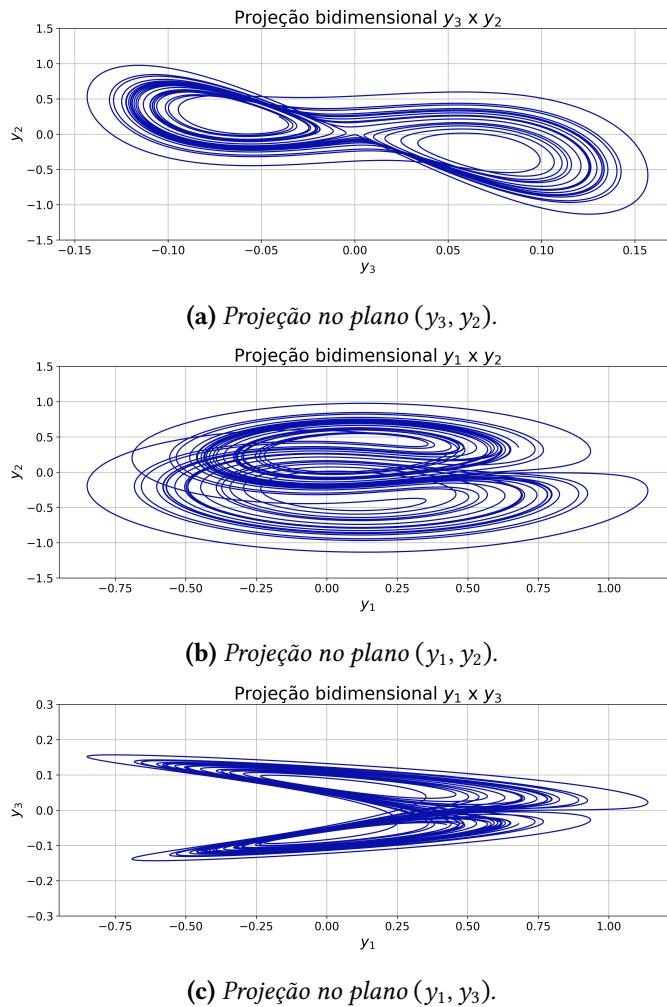


Figura 1.5: Projeções do modelo BE para $f_1 = 0.3027$.

Visualmente, temos que o modelo BE, assim como os dois modelos anteriores, não há convergência quando $t \rightarrow \infty$, e o regime estacionário exibe um atrator estranho. Ao comparar aos modelos anteriores, temos que o modelo BE é muito semelhante ao modelo QG e, consequentemente, apresenta um comportamento mais comportado ao comparar com o modelo PE.

Apesar das semelhanças entre os modelos BE e QG, observam-se diferenças nas escalas das variáveis y . Ao comparar as projeções (y_3, y_2) e (y_1, y_3) (Figs. 1.4a vs. 1.5a e 1.4c vs. 1.5c), verifica-se que as trajetórias do modelo QG ocupam uma região ligeiramente menor no plano de fases. Essa diferença reflete-se nas amplitudes das variáveis, conforme resumido na Tabela 1.1. As componentes y_1 e y_2 apresentam amplitudes ligeiramente menores no modelo QG, enquanto a variável y_3 exibe o comportamento oposto, com amplitude superior no modelo QG.

Modelo	Δy_1	Δy_2	Δy_3
PE	2.24	2.57	0.97
QG	1.95	1.95	0.47
BE	1.99	2.11	0.30

Tabela 1.1: Comparação das amplitudes para cada modelo.

Também podemos verificar tal diferença na série temporal das variáveis y_i em determinada faixa de tempo, como mostra a figura abaixo:

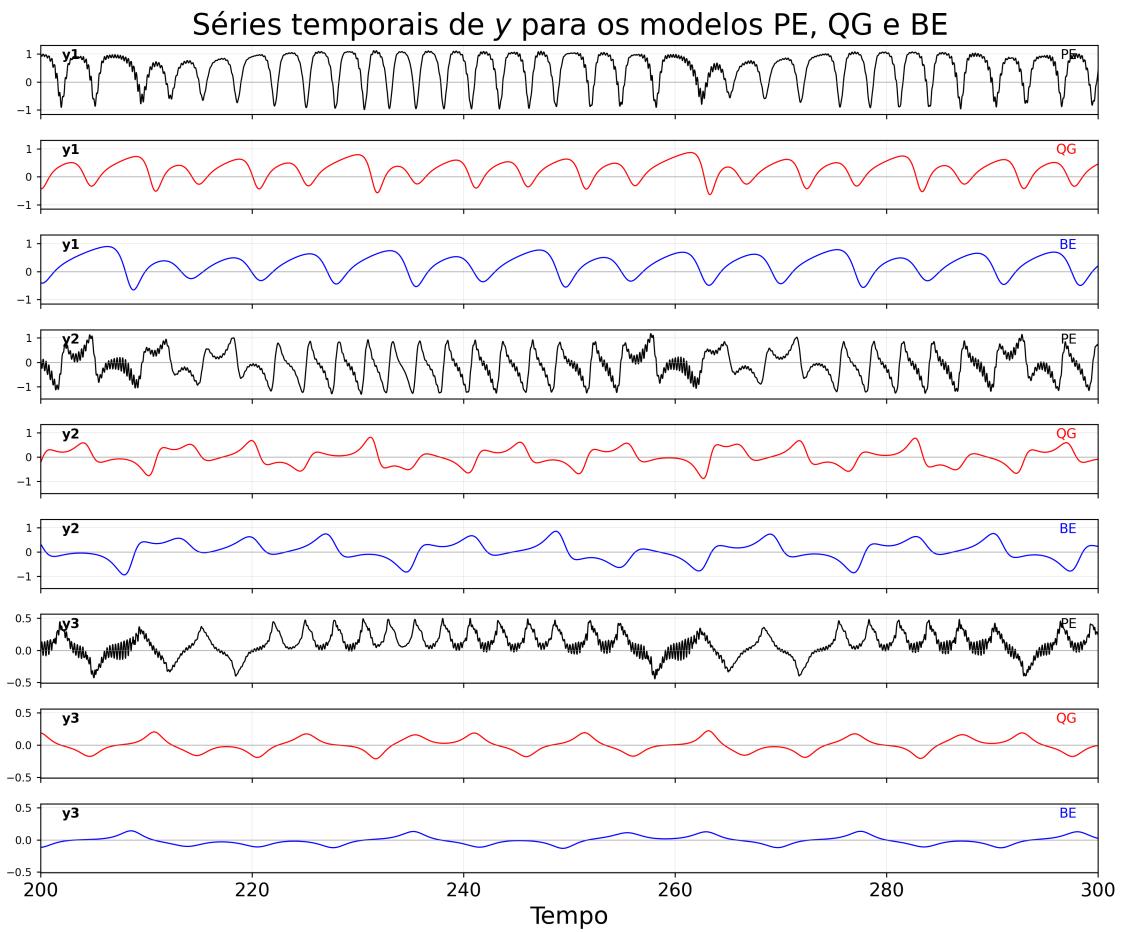


Figura 1.6: Comparação das séries temporais de y para os modelos PE, BE, QG

Capítulo 2

Introdução às equações diferenciais estocásticas

Neste capítulo, faremos uma introdução às equações diferenciais estocásticas (EDEs). Iniciaremos na Seção 2.1 com a apresentação de um problema motivador, que nos ajudará a entender a necessidade e a utilidade das EDEs. Na sequência, na Seção 2.2, reuniremos algumas definições estatísticas fundamentais que serão utilizadas ao longo do texto.

Por fim, com base em todas essas considerações, retornaremos ao problema motivador inicial e realizaremos uma simulação computacional utilizando a linguagem Julia, com foco na biblioteca SciML: Open Source Software for Scientific Machine Learning.

2.1 Motivação

Para iniciar a introdução às equações diferenciais estocásticas (EDEs), vamos apresentar a motivação para o seu estudo. Começamos com uma equação diferencial ordinária:

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{b}(\mathbf{x}(t)) \\ \mathbf{x}(0) = \mathbf{x}_0, \end{cases} \quad (2.1)$$

onde $t > 0$, $\mathbf{b} : \mathbb{R}^n \rightarrow \mathbb{R}^n$ é um vetor suave, e $\mathbf{x}(t) : [0, \infty) \rightarrow \mathbb{R}^n$ representa a trajetória do sistema ao longo do tempo. O sistema (2.1) é classificado como determinístico, pois, uma vez conhecida a condição inicial, a trajetória do sistema é totalmente determinada, conforme garante o Teorema de Existência e Unicidade.

Contudo, suponha que desejamos modelar a evolução de um fenômeno sujeito a incertezas. Em outras palavras, queremos aplicar o mesmo princípio das equações diferenciais ordinárias a um processo aleatório. Para isso, utilizamos as equações diferenciais estocásticas. Esse tipo de equação é definido como:

$$\begin{cases} \dot{X}(t) = b(t, X(t)) + \sigma(t, X(t))\xi(t) \\ X(0) = x_0, \end{cases} \quad (2.2)$$

onde $X(t) \in \mathbb{R}^d$, $b : [0, T] \times \mathbb{R}^d \mapsto \mathbb{R}^d$, e $\sigma : [0, T] \times \mathbb{R}^d \mapsto \mathbb{R}^{d \times m}$. Usamos a notação $\xi(t) = \frac{dW}{dt}$ para denotar (de forma formal) a derivada do movimento Browniano em \mathbb{R}^m , ou seja, um processo de ruído branco — um vetor Gaussiano generalizado, de média zero.

Como exemplo motivador para o uso de EDEs, consideremos a evolução do preço de uma ação na bolsa de valores ao longo do tempo. Esse preço é influenciado por diversos fatores externos, o que introduz um grau de incerteza quanto ao seu comportamento futuro. Assim, podemos modelá-lo por meio de uma EDE:

$$\begin{cases} \frac{ds}{s} = \mu dt + \sigma dW \\ S(0) = s_0, \end{cases} \quad (2.3)$$

Nessa equação, $\frac{ds}{s}$ representa a variação relativa do preço da ação ao longo do tempo; μ é uma constante que representa a taxa média de crescimento; e σ expressa a volatilidade, ou seja, a incerteza associada aos diversos fatores que influenciam no preço.

Antes de continuar com esse exemplo, apresentaremos a base teórica necessária para a formulação da equação (2.3), bem como suas propriedades.

2.2 Considerações estatísticas

Primeiramente, é importante destacar que parte das definições apresentadas nesta seção fazem uso da teoria da medida no contexto da teoria das probabilidades. Embora esses conceitos não sejam abordados neste trabalho, o leitor interessado em se aprofundar no tema pode recorrer a [EVANS \(2014\)](#), que oferece uma excelente (e rápida) introdução ao assunto.

Definição 2.1 (Processo Estocástico). *Uma coleção $\{X(t) \mid t \geq 0\}$ de variáveis aleatórias é chamada de processo estocástico.*

Definição 2.2 (Filtração). *Seja $W(\cdot)$ um processo estocástico. Denotamos por $\mathcal{W}(t) := \sigma(W(s) : 0 \leq s \leq t)$ a σ -álgebra gerada por W até o tempo t , isto é, o conjunto de todos os eventos observáveis do processo até esse instante.*

Dizemos que uma família $\mathcal{F}(\cdot)$ de σ -álgebras $\subseteq \mathcal{U}$ é uma filtração (com respeito a $W(\cdot)$) se satisfaz as seguintes propriedades:

1. $\mathcal{F}(t) \supseteq \mathcal{F}(s)$ para todo $t \geq s \geq 0$;
2. $\mathcal{F}(t) \supseteq \mathcal{W}(t)$ para todo $t \geq 0$;
3. $\mathcal{F}(t)$ é independente de $\mathcal{W}^+(t) := \sigma(W(u) - W(t) : u > t)$ para todo $t \geq 0$.

Definição 2.3 (\mathcal{F} -mensurabilidade). Seja (Ω, \mathcal{F}, P) um espaço de probabilidade. Dizemos que uma função

$$X : \Omega \rightarrow \mathbb{R}^n$$

é \mathcal{F} -mensurável se, para todo boreiano $B \subseteq \mathbb{R}^n$, a pré-imagem

$$X^{-1}(B) = \{\omega \in \Omega : X(\omega) \in B\}$$

pertence a \mathcal{F} .

Definição 2.4 (Martingal). Seja $\{\mathcal{F}_t\}_{t \in [0, T]}$ uma filtração definida no espaço de probabilidade $(\Omega, \mathcal{F}, \mu)$, e seja $\{\mathcal{M}_t\}_{t \in [0, T]}$ um processo adaptado a \mathcal{F}_t , com $\mathcal{M}_t \in L^1(0, T)$. Dizemos que \mathcal{M}_t é uma martingal em relação a \mathcal{F}_t se

$$\mathbb{E}[\mathcal{M}_t | \mathcal{F}_s] = \mathcal{M}_s \quad \forall t \geq s.$$

2.3 Equações diferenciais estocásticas

Dadas as considerações estatísticas, passemos para a construção da equação (2.1), reproduzida novamente a seguir:

$$\begin{cases} \dot{X}(t) = b(t, X(t)) + \sigma(t, X(t)) \xi(t), \\ X(0) = x_0, \end{cases}$$

A solução $X(\cdot)$ é então expressa como:

$$X(t) = x_0 + \int_0^t b(s, X(s)) ds + \int_0^t \sigma(s, X(s)) dW_s. \quad (2.4)$$

A equação (2.4) possui uma solução bem definida, garantida pelo Teorema de Existência e Unicidade para equações diferenciais estocásticas, cuja demonstração completa pode ser consultada em [EVANS \(2014\)](#). Esse resultado possui papel fundamental, assim como no caso das equações diferenciais ordinárias, pois assegura não apenas que uma solução existe, mas que ela é única dadas as condições iniciais especificadas. Tal garantia é essencial para que a modelagem estocástica tenha validade teórica e aplicação prática.

Para que o sistema (2.3) esteja completamente definido, ainda é necessário esclarecer dois elementos fundamentais: a natureza do ruído $\xi(t)$ e a interpretação da integral estocástica $\int_0^t f(s, X(s)), dW_s$. Nesta seção, abordaremos em detalhes esses dois conceitos, explorando suas definições formais e principais propriedades, a fim de justificar rigorosamente a formulação de uma EDE.

2.3.1 Processo de Wiener

Desenvolvimento histórico

Um dos elementos fundamentais para o estudo de equações diferenciais estocásticas é o *movimento browniano*. Trata-se de um fenômeno físico que envolve o estudo do movimento de grãos de pólen suspensos na água.

Historicamente, o problema foi inicialmente proposto e estudado por Robert Brown entre 1826 e 1827. Em seus estudos, Brown observou que a trajetória das partículas era irregular e que o movimento de duas partículas distintas parecia ser independente. Em 1905, Albert Einstein retomou o problema, relacionando-o à equação da difusão.¹

Finalmente, na década de 1920, o fenômeno foi formalizado matematicamente por Norbert Wiener. Essa formalização é crucial para a definição rigorosa do movimento browniano, tanto que outra forma de referir-se a ele é *processo de Wiener* – forma que utilizaremos a fim de diferenciar do problema físico original.

Definição e propriedades

Definição 2.5. Um processo estocástico real, denotado por $W(\cdot)$ é chamado de processo de Wiener padrão quando satisfaz as seguintes propriedades:

1. $W(0) = 0$, quase certamente;
2. Para todo $t \geq s \geq 0$, tem-se que $W(t) - W(s) \sim \text{Normal}(0, t - s)$;
3. $W(\cdot)$ possui incrementos independentes, isto é, para $0 < t_1 < t_2 < \dots < t_n$, as variáveis aleatórias

$$W(t_1), W(t_2) - W(t_1), \dots, W(t_n) - W(t_{n-1}) \text{ são independentes.}$$

(EVANS, 2014)

A função densidade de probabilidade do processo de Wiener padrão unidimensional, definido como $W(t) : \mathbb{R}^+ \rightarrow \mathbb{R}$, é dada por:

$$f(x; t, s) = (2\pi(t - s))^{-\frac{1}{2}} \exp\left(-\frac{x^2}{2(t - s)}\right).$$

No caso do processo de Wiener padrão n -dimensional, $W(t) : \mathbb{R}^+ \rightarrow \mathbb{R}^n$, onde cada componente $W_i(t)$, com $i = 1, \dots, n$, é um processo de Wiener unidimensional independente, a função densidade do vetor gaussiano aleatório $W(t) - W(s)$ é dada por:

$$g(\mathbf{x}; t, s) = (2\pi(t - s))^{-n/2} \exp\left(-\frac{\|\mathbf{x}\|^2}{2(t - s)}\right).$$

As principais propriedades do processo de Wiener padrão, de acordo com PAVLIOTIS

¹ Detalhes em EVANS (2014)

(2014), são:

1. **Reescalonamento.** Para cada $c > 0$, defina $X_t = \frac{1}{\sqrt{c}} W(ct)$. Então $(X_t, t \geq 0) = (W_t, t \geq 0)$.
2. **Translação.** Para cada $c > 0$, $W_{c+t} - W_c$, $t \geq 0$, é um processo de Wiener que é independente de W_u , $u \in [0, c]$.
3. **Reversão no tempo.** Defina $X_t = W_{1-t} - W_1$, $t \in [0, 1]$. Então $(X_t, t \in [0, 1]) = (W_t, t \in [0, 1])$.
4. **Inversão.** Seja X_t , $t \geq 0$, definido por $X_0 = 0$, $X_t = t W(\frac{1}{t})$. Então $(X_t, t \geq 0) = (W_t, t \geq 0)$.

2.3.2 Integral estocástica

Nesta seção, vamos definir uma integral estocástica, representada genericamente abaixo:

$$I(t) = \int_0^t f(s) dW_s. \quad (2.5)$$

Para definir o que significa uma integral estocástica, seguiremos a abordagem de EVANS (2014). A partir dela, definiremos a integral de Itô e suas propriedades, além de mencionar a integral de Stratonovich conforme apresentada por PAVLIOTIS (2014).

Inicialmente, é importante destacar que EVANS (2014) introduz a definição de $\int_0^t f(\cdot) dW$ em três etapas. Primeiro, quando $f(\cdot)$ é uma função determinística simples, constante por partes. Depois, quando $f(\cdot)$ é determinística geral em $L^2([0, T])$. Por fim, quando $f(\cdot)$ é um processo estocástico adaptado mais geral.

Embora todos os casos estejam bem definidos, focaremos apenas neste último, aquele em que $f(\cdot)$ é um processo estocástico adaptado, por ser o mais relevante para os nossos objetivos. Essa escolha também se justifica pelo fato de PAVLIOTIS (2014) adotar abordagem semelhante, ainda que de forma menos detalhada.

Assumiremos, portanto, que f pertence a $L^2([0, t] \times \Omega)$, isto é:

$$\mathbb{E} \left(\int_0^t f(s)^2 ds \right) < \infty. \quad (2.6)$$

Além disso, assumimos que o integrando $f(t)$ é \mathcal{F}_t -mensurável, onde \mathcal{F}_t é a filtragem gerada pelo movimento browniano $W(t)$. No nosso contexto, a partir da definição 2.2, isso significa que f depende apenas da informação disponível até o instante t , ou, de forma equivalente, não depende do futuro.

Dadas essas considerações em relação à função f , voltemos à equação (2.5). Primeiro, tomamos um intervalo $[0, T]$, em seguida, definimos uma partição P deste intervalo:

$$P := \{0 = t_0 < t_1 < \dots < t_m = T\}.$$

Para $0 \leq \lambda \leq 1$ fixado e P uma partição de $[0, T]$, definimos

$$\tau_k := (1 - \lambda)t_k + \lambda t_{k+1} \quad k = 0, \dots, m - 1. \quad (2.7)$$

Para a partição P e para $0 \leq \lambda \leq 1$, definimos

$$\sum_{k=0}^{m-1} f(\tau_k)(W(t_{k+1}) - W(t_k)).$$

A partir do conceito de aproximação de Riemann, tomamos $m \rightarrow \infty$ e podemos definir a integral $I(t)$ da seguinte forma:

$$I(t) = \lim_{m \rightarrow \infty} \sum_{k=0}^{m-1} f(\tau_k)(W(t_{k+1}) - W(t_k)). \quad (2.8)$$

Temos que a definição de (2.8) depende da escolha de $\lambda \in [0, 1]$ em (2.7). Se $\lambda = 0$, obtemos a integral estocástica de Itô:

$$I_I(t) := \lim_{K \rightarrow \infty} \sum_{k=0}^{K-1} f(t_k)(W(t_{k+1}) - W(t_k)) = \int_0^t f(s) dW_s.$$

Se tomarmos $\lambda = \frac{1}{2}$, definimos a integral estocástica de Stratonovich:

$$I_S(t) := \lim_{K \rightarrow \infty} \sum_{k=0}^{K-1} f\left(\frac{1}{2}(t_k + t_{k+1})\right)(W(t_{k+1}) - W(t_k)) = \int_0^t f(s) \circ dW_s.$$

Ambas são duas formas diferentes de abordar a mesma questão, porém, segundo [EVANS \(2014\)](#), cada uma contém suas próprias vantagens. Para a integral de Itô:

1. Valem as seguintes fórmulas:

$$\begin{aligned} \mathbb{E}\left(\int_0^t f(s) dW_s\right) &= 0 \\ \mathbb{E}\left(\left(\int_0^t f(s) dW_s\right)^2\right) &= \mathbb{E}\left(\int_0^t f^2(s) ds\right) \end{aligned}$$

2. I_s é um martingal (vide definição 2.4)

Para a forma de Stratonovich as vantagens são:

1. A regra da cadeia usual é válida.
2. As soluções de EDEs são estáveis em relação a mudanças nos termos aleatórios.

2.4 Simulação de uma equação diferencial estocásticas

Dadas as considerações teóricas, voltaremos ao problema motivador.

$$\begin{cases} \frac{dS}{S} = \mu dt + \sigma dW \\ S(0) = s_0, \end{cases}$$

Aplicando a regra da cadeia de Itô,² temos que a solução do problema é dada por:

$$S(t) = s_0 \cdot \exp \left(\sigma W(t) + \left(\mu - \frac{\sigma^2}{2} t \right) \right)$$

Dada toda teoria exposta, sabemos que o resultado está bem definido. Mais que isso, a partir dele podemos realizar uma simulação computacional utilizando a biblioteca *SciML: Open Source Software for Scientific Machine Learning* ([RACKAUCKAS e NIE, 2017](#)). O código-fonte correspondente pode ser consultado no Apêndice B.2.

Para a simulação, adotamos os seguintes parâmetros: $\mu = 0,05$, representando uma taxa de crescimento anual; $\sigma = 1,0$, quantificando a volatilidade do mercado no período considerado; e um passo de tempo de $\frac{1}{365 \cdot 24}$, o que corresponde à variação por hora dos preços. Estabelecemos ainda o preço inicial da ação em R\$ 12,00. Com base nesses parâmetros, obtivemos o gráfico a seguir:

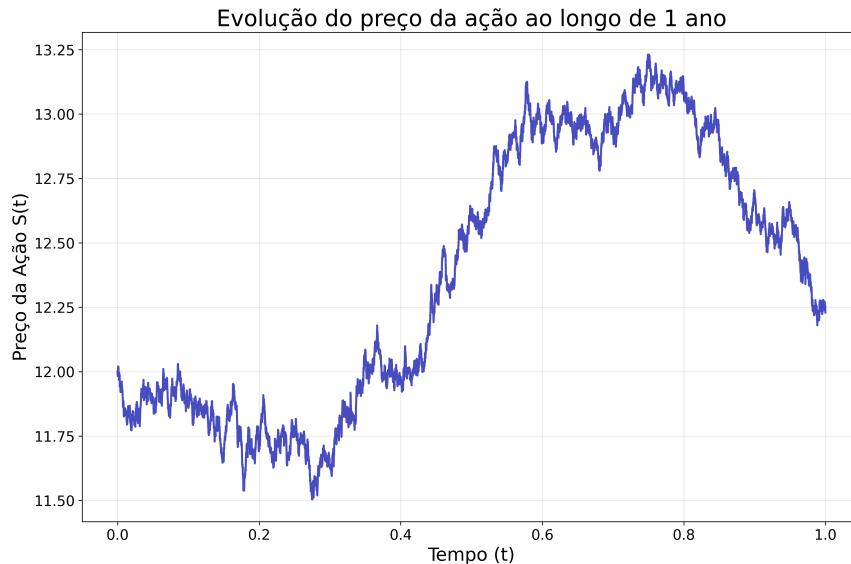


Figura 2.1: Simulação do preço de uma ação a partir da EDE (2.3)

Vale destacar que: por se tratar de um fenômeno aleatório, temos que diferentes simulações geram diferentes resultados. Contudo, nas simulações computacionais, é

² Definida em [EVANS \(2014\)](#)

possível definir o valor de uma *seed* garantindo reproduzibilidade do código.

Capítulo 3

Aproximação de sistemas determinísticos usando EDEs

3.1 Introdução

Neste capítulo, abordaremos a aproximação de sistemas determinísticos por meio de equações diferenciais estocásticas (EDEs). Para isso, analisaremos dois modelos distintos. O primeiro, apresentado na Seção 3.2, é um modelo mais simples que nos ajudará a ilustrar o processo de aproximação. Nessa etapa, descreveremos o modelo, discutiremos brevemente os aspectos teóricos que viabilizam a aproximação estocástica e, por fim, compararemos os comportamentos dos modelos determinístico e estocástico, por meio da evolução temporal da variável de interesse e da análise de histogramas dos resultados. Na Seção 3.3, abordaremos a versão estocástica do modelo Lorenz 80,BE-SLO, conforme desenvolvido em [CHEKROUN, LIU e McWILLIAMS \(2021\)](#). Serão apresentados alguns fundamentos teóricos do modelo, como variedades, ressonâncias e oscilações. Em seguida, descreveremos o processo de construção do modelo estocástico e, por fim, exibiremos os resultados das simulações, comparando-os com aqueles obtidos a partir do modelo determinístico original.

3.2 *Fast Chaotic Noise*

O exemplo a seguir, intitulado *Fast Chaotic Noise*, foi retirado de [PAVLIOTIS e STUART \(2008, p. 169\)](#) e é descrito pelo sistema

$$\begin{aligned} \frac{dx}{dt} &= x - x^3 + \frac{\lambda}{\varepsilon} y_2, \\ \frac{dy_1}{dt} &= \frac{10}{\varepsilon^2} (y_2 - y_1), \\ \frac{dy_2}{dt} &= \frac{1}{\varepsilon^2} (28y_1 - y_2 - y_1 y_3), \\ \frac{dy_3}{dt} &= \frac{1}{\varepsilon^2} (y_1 y_2 - \frac{8}{3} y_3). \end{aligned} \tag{3.1}$$

Temos que $y^T = (y_1, y_2, y_3)$ satisfaz o modelo de Lorenz 63 (ver equação (1.3)), e o parâmetro $\varepsilon > 0$ é responsável por separar as escalas de tempo. No sistema apresentado, os termos proporcionais a ε^{-2} nas equações de y_i fazem com que $y(t)$ evolua em uma escala muito mais rápida do que $x(t)$, de modo que o sistema assume uma estrutura de modelo rápido-lento.

O fluxo de Lorenz 63 é ergódico com respeito à sua medida SRB (de Sinai-Ruelle-Bowen). Segundo [YOUNG \(2002\)](#), podemos caracterizar a medida SRB como uma medida invariante que é a mais compatível com o volume quando o volume não é preservado. Além disso, ela fornece um mecanismo que permite obter estatísticas coerentes, como médias temporais, para as órbitas no atrator. Em [PAVLIOTIS e STUART \(2008\)](#) é destacado que o atrator do sistema Lorenz 63 tem volume zero e que a medida SRB desempenha o mesmo papel que uma medida invariante desempenha em sistemas ergódicos clássicos.

Isto posto, para funções integráveis, as médias temporais ao longo de uma trajetória coincidem com as médias na medida invariante. Em particular, pela condição de centralização adotada em [PAVLIOTIS e STUART \(2008\)](#), o componente y_2 possui média zero na medida invariante. Quando $\varepsilon \rightarrow 0$, o termo rápido induz um ruído efetivo na dinâmica de x , e o sistema pode ser aproximado pela EDE

$$dX_t = (X_t - X_t^3)dt + \sigma W_t, \quad (3.2)$$

onde,

$$\sigma^2 = 2\lambda^2 \int_0^\infty \text{Cov}(y_2(0), y_2(s)) ds, \quad (3.3)$$

ou seja, a variância efetiva é dada pela autocorrelação integrada de y_2 .

Dada as devidas considerações teóricas, para as simulações computacionais, utilizamos a linguagem *Python* para a simulação do sistema determinístico (3.1) e para o cálculo do σ em (3.3). Para o sistema estocástico de (3.2), utilizamos a linguagem *Julia*. Os códigos utilizados para a simulação estão em [B.3](#). Vale destacar que, conforme discutido na seção anterior, à medida que $\varepsilon \rightarrow 0$, o termo rápido tende a gerar um ruído efetivo, aproximando a dinâmica do sistema da solução da equação diferencial estocástica (EDE).

Para ilustrar esse comportamento, consideramos três valores de ε : 0.2, 0.1 e 0.01. Embora o caso $\varepsilon = 0.001$ seja teoricamente relevante, os erros absolutos associados a esse valor, da ordem de 10^{-9} e 10^{-12} , exigem precisão numérica e capacidade computacional significativamente superiores às disponíveis, inviabilizando sua simulação. Isto posto, para os valores de ε mencionados, obtivemos diferentes valores de σ expressos na tabela a seguir:

ε	σ
0.2	205.05745681770404
0.1	142.6677313864854
0.01	5.754046294922591

Tabela 3.1: Valores de ε e dos respectivos coeficientes de difusão σ

Note que embora σ seja definido teoricamente de forma independente de ε , os valores de σ obtidos numericamente dependem de ε . Isso ocorre porque, nas simulações, o cálculo de σ é realizado a partir do sistema determinístico original, no qual o termo rápido ainda está sujeito à escala temporal imposta por ε . A partir dos valores, obtivemos os seguintes resultados gráficos:

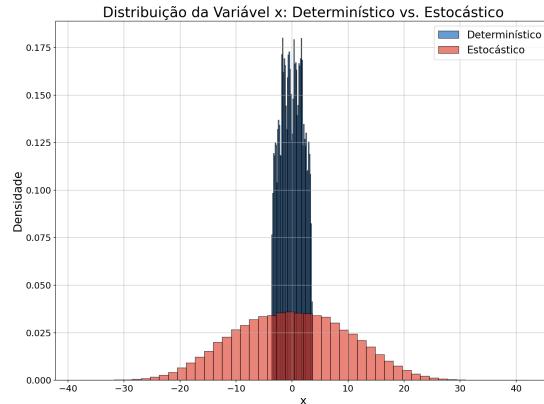
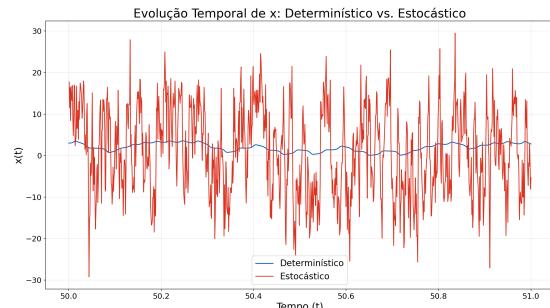
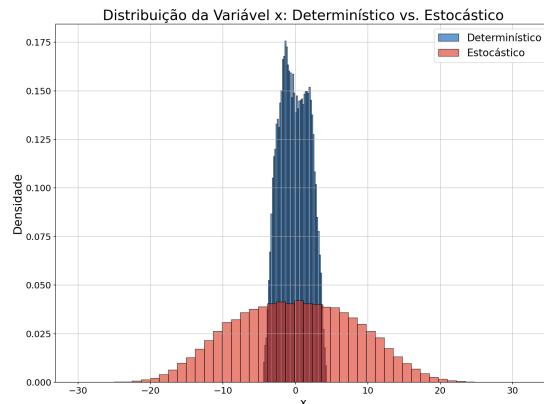
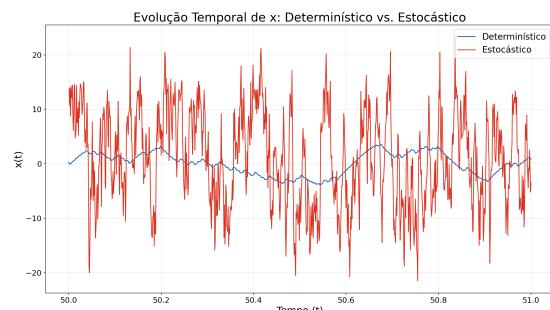
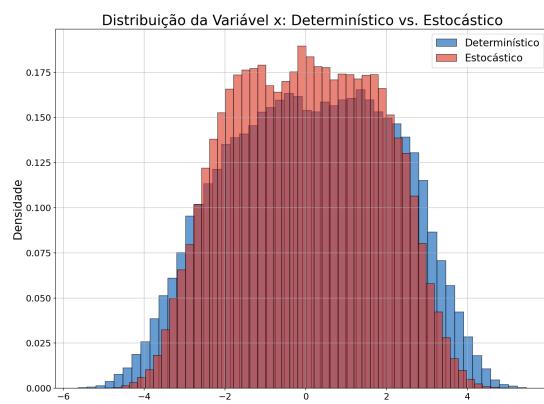
(a) $\varepsilon = 0.2$: Histograma(b) $\varepsilon = 0.2$: Série temporal(c) $\varepsilon = 0.1$: Histograma(d) $\varepsilon = 0.1$: Série temporal(e) $\varepsilon = 0.01$: Histograma(f) $\varepsilon = 0.01$: Série temporal

Figura 3.1: Comparação entre o modelo determinístico e o estocástico para diferentes valores de ε (esq.) e séries temporais (dir.).

Ao analisar as imagens obtidas, observa-se que, inicialmente, para $\varepsilon = 0.2$, o modelo estocástico apresenta uma amplitude significativamente maior em comparação ao modelo determinístico. Pelo histograma, é possível inferir que o modelo determinístico assume valores aproximadamente entre -4 e 4 , enquanto o estocástico varia entre -20 e 20 . Essa diferença também é evidente na série temporal correspondente, onde o comportamento do modelo estocástico se mostra consideravelmente mais ruidoso do que o determinístico.

Para $\varepsilon = 0.1$, observam-se os mesmos padrões identificados em $\varepsilon = 0.2$. No entanto, os valores estão mais concentrados em torno da média, indicando uma menor dispersão em relação ao caso anterior.

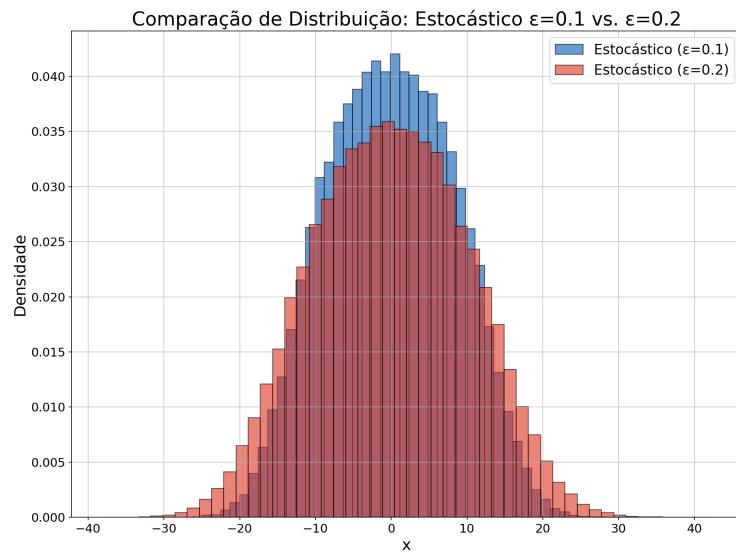


Figura 3.2: Comparação dos modelos estocásticos gerados com $\varepsilon = 0.1$ e $\varepsilon = 0.2$

Para $\varepsilon = 0.01$, os histogramas revelam uma boa aproximação entre os dois modelos, tanto em termos de densidade quanto de amplitude. Apesar das trajetórias apresentarem diferenças visuais, é importante destacar que, devido à aleatoriedade característica do sistema, cada realização do processo estocástico produz uma trajetória distinta.¹ No entanto, tais diferenças não afetam a distribuição estacionária, os picos e vales presentes numa realização são compensados, em média, pelos de qualquer outra realização. Assim, mesmo com trajetórias diferentes, a aproximação permanece válida.

3.3 Lorenz 80 BE-SLO

O modelo BE-SLO foi desenvolvido no artigo de “*Stochastic rectification of fast oscillations on slow manifold closures*” (CHEKROUN, LIU e McWILLIAMS, 2021). O modelo em questão é uma versão estocástica do modelo Lorenz 80 que baseia-se na versão BE. Nesta seção, vamos abordados os aspectos teóricos necessários para o desenvolvimento do modelo BE-SLO e além disso realizar a construção deste e compará-lo com o modelo PE.

¹ Para proporcionar reporodutibilidade do código, utilizamos *seeds* randômicas

3.3.1 Variedades e escolha do modelo BE

Na atmosfera, temos variáveis sob diferentes escalas temporais, como, por exemplo, as variáveis rápidas associadas às ondas de gravidade e aquelas relacionadas aos movimentos mais lentos, como os associados às ondas de *Rossby*.

A partir deste fato, surge a motivação para realizar um “filtro” capaz de reduzir discrepâncias e estabelecer um equilíbrio entre as variáveis, de forma que seja possível realizar análises coerentes com a escala de interesse. Essa filtragem é feita através de uma variedade² que representa o espaço geométrico onde o balanço entre as componentes rápidas e lentas é mantido.

A ideia original de utilizar uma variedade para representar esse tipo de equilíbrio foi introduzida por Leith no artigo “*Nonlinear Normal Mode Initialization and Quasi-Geostrophic Theory*” (LEITH, 1980). Nesse trabalho, Leith introduz o conceito de *slow manifold*, uma variedade que filtra as oscilações rápidas e aproxima o sistema para um regime quasi-geostrófico.

Em LORENZ (1980), Lorenz emprega uma ideia conceitualmente semelhante para transicionar do modelo PE para o modelo QG sem perder as propriedades fundamentais do sistema original. Vale destacar que, no artigo de Lorenz, ele não utiliza o termo *slow manifold*, mas sim *invariant manifold*. No capítulo 1, mostramos essa transição de modelos. Contudo, a partir do momento em que aplicamos as hipóteses de simplificação apresentadas em 1.4.2, não estamos apenas simplificando o modelo, mas também alterando a variedade na qual ele está definido. Isso é essencial, pois este processo preserva as propriedades dinâmicas do modelo original.

O estudo dessas variedades foi aprofundado ao longo do tempo, com novos desenvolvimentos, como em “*The Slow Manifold—What Is It?*” (LORENZ, 1992). Em particular, destacamos o *Parameterizing Manifold* (PM Manifold), introduzido no artigo “*Finite-Horizon Parameterizing Manifolds, and Applications to Suboptimal Control of Nonlinear Parabolic PDEs*” e no livro “*Stochastic Parameterizing Manifolds and Non-Markovian Reduced Equations*” (CHEKROUN e LIU, 2014; CHEKROUN, LIU e WANG, 2015). No artigo CHEKROUN, LIU e McWILLIAMS (2017), os autores desenvolvem uma versão ótima do PM, *slow optimal parameterizing manifold*.

O OPM, como dito anteriormente, é uma variedade que reduz as discrepâncias entre as variáveis rápidas e lentas, contudo, nessa formulação, essa redução ocorre por meio de uma abordagem estatística baseada no movimento médio do sistema, relacionando as variáveis com a expectativa condicional nas teorias de previsão ótima.

Para a construção do OPM, considera-se uma medida de probabilidade invariante μ que satisfaz propriedades ergódicas. No espaço E onde o sistema está definido, assume-se a decomposição $E = E_c \oplus E_s$, onde E_s corresponde ao espaço das variáveis rápidas e E_c ao das

² De acordo com KUZNETSOV (2023), podemos definir uma variedade de forma simplificada como um conjunto de pontos em \mathbb{R}^n que satisfazem um sistema de m equações escalares:

$$F(x) = 0,$$

onde $F : \mathbb{R}^n \rightarrow \mathbb{R}^m$ para algum $m \leq n$.

lentas. Procura-se então uma função $\Psi : E_c \rightarrow E_s$ que satisfaça o problema de minimização

$$\min_{\Psi \in \mathcal{E}} \int_{E_c \times E_s} \|Y - \Psi(X)\|^2 d\mu,$$

onde \mathcal{E} é o espaço das funções de E_c para E_s . Esse problema admite solução única dada por

$$\Psi^*(X) = \int_{E_s} Y d\mu_X(Y), \quad X \in E_c,$$

onde μ_X é a medida de probabilidade condicionada ao valor X das variáveis lentas. Em particular, sob hipóteses adequadas, o OPM fornece uma aproximação ótima de $\mathbb{E}[F | X]$, que é essencial para a aplicação do formalismo de Mori-Zwanzig.

Por fim, vale destacar que, em [CHEKROUN, LIU e McWILLIAMS \(2017\)](#), por meio de simulações numéricas, observou-se que a variedade OPM apresenta comportamento muito próximo à variedade associada ao modelo BE. Assim, o modelo BE foi utilizado como base para a construção do modelo estocástico, servindo como aproximação adequada do movimento lento necessário para o BE-SLO.

3.3.2 Oscilações estocásticas de Stuart-Landau

As oscilações estocásticas de Stuart-Landau são utilizadas em [CHEKROUN, LIU e McWILLIAMS \(2021\)](#) para a geração do ruído. Para definir uma oscilação estocástica de Stuart-Landau (ESL), primeiro vamos tratar das equações de Stuart-Landau em sua forma determinística para então tratar da sua forma estocástica. Ao final da seção, apresentaremos uma simulação de uma ESL simples proposta em [CHEKROUN, LIU e McWILLIAMS \(2021\)](#).

Antes de definir uma equação de Stuart Landau, é necessário definir uma *bifurcação de Hopf*. Segundo [KUZNETSOV \(2023\)](#) bifurcação genérica é definida como:

Definição 3.1. *O surgimento de um retrato de fase topologicamente não equivalente sob variação de parâmetros é chamado de bifurcação.*

Ou seja, podemos interpretar uma bifurcação como um ponto crítico no espaço de parâmetros onde a solução muda de comportamento qualitativo. Em especial, temos a *bifurcação de Hopf*, é uma definida por [KUZNETSOV \(2023\)](#), como:

Definição 3.2. *A bifurcação correspondente à presença de autovalores $\lambda_{1,2} = \pm i\omega_0$, com $\omega_0 > 0$, é chamada de bifurcação de Hopf (ou de Andronov-Hopf).*

Por fim, a partir da bifurcação de Hopf, podemos definir uma equação de Stuart-Landau. Uma equação deste tipo é definida a partir de pequenas amplitudes válidas perto do ponto de bifurcação de Hopf. Por [KURAMOTO \(1984\)](#), a equação que descreve as equações de Stuart-Landau são dadas por:

$$\frac{dA}{dt} = \lambda A - \frac{g}{2} |A|^2 A,$$

onde:

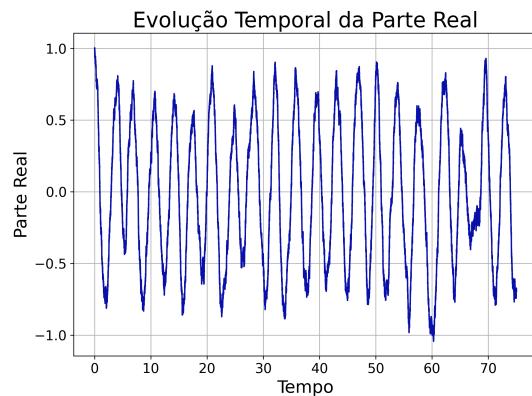
- $A = |A|e^{i\phi}$ é uma quantidade complexa que descreve a perturbação;
- $\lambda = \lambda_r + i\lambda_i$ o autovalor complexo
- g é um número complexo

Dada a definição do sistema determinístico, podemos considerar que a sua versão estocástica expressa por:

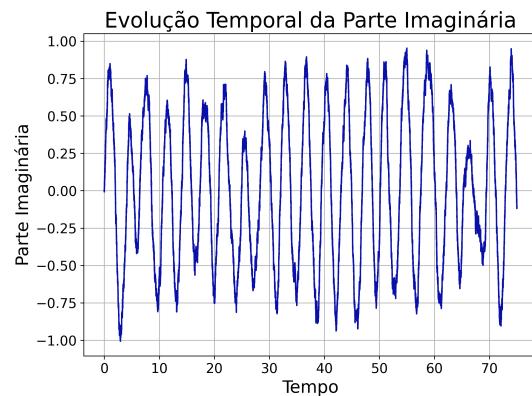
$$\frac{dz}{dt} = (\alpha + i\omega)z - \beta z|z|^2 + \sigma \frac{dW}{dt}, \quad (3.4)$$

onde $\frac{W}{dt}$ denota um ruído branco complexo, enquanto σ , α e ω são escalares reais, e β é um coeficiente complexo com parte real positiva.

Realizamos simulações em *Julia* para retratar o comportamento visual do 3.4, com os seguinte parâmetros: $\alpha = \frac{1}{2}$, $\omega = 2$, $\beta = 1 + \frac{1}{2}i$ e $\sigma = \frac{1}{4}$. Os resultados obtidos foram:



(a) Projeção no plano (y_1, y_2) .



(b) Projeção no plano (y_1, y_3) .

Figura 3.3: Evolução temporal da parte real e imaginária

Além disso, temos que a projeção da parte real e imaginária é dada por:

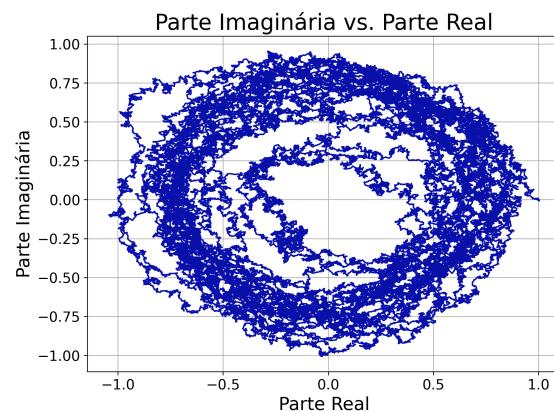


Figura 3.4: Projeção da parte real e imaginária

O código utilizado para a simulação computacional consta no Apêndice B.4

3.3.3 Ressonâncias de Ruelle-Pollicot

De acordo com TANTET *et al.* (2020), as ressonâncias de Ruelle–Pollicott³ (RP) correspondem aos valores próprios do gerador do semigrupo de operadores de transferência, responsável por descrever a evolução das densidades.

Elas têm como objetivo fornecer uma caracterização completa da evolução dos observáveis. As ressonâncias RP permitem decompor as funções de correlação em vários componentes que estão associados a uma taxa de decaimento e a uma frequência determinadas pelos autovalores do operador associado ao semigrupo que descreve a dinâmica. Em suma, o uso de ressonâncias RP fornece uma descrição das propriedades de dinâmicas não lineares sujeitas a perturbações de ruído.

3.3.4 Construção do modelo BE-SLO

Agora, construiremos o modelo BE-SLO com base no artigo de CHEKROUN, LIU e McWILLIAMS (2021). O primeiro passo é definir o OPM (*Optimal Prediction Manifold*), que representa uma variedade capaz de fornecer a melhor aproximação média das variáveis rápidas em função das variáveis lentas, reduzindo assim a discrepância entre elas.

Na sequência, aplicamos o formalismo de Mori-Zwanzig⁴ (MZ). Em termos gerais, esse método projeta o sistema em variáveis resolvíveis e não resolvíveis, reescrevendo a dinâmica das variáveis resolvíveis de modo que os efeitos das variáveis não resolvíveis se manifestem por meio de dois termos: um de memória e outro de ruído. A partir desse formalismo, obtemos – de forma formalmente exata – a seguinte decomposição para a dinâmica das variáveis resolvíveis:

$$\dot{X} = \underbrace{\mathbb{E}[F | X]}_{\text{termo determinístico ótimo}} + \underbrace{\int_0^t \Gamma(t-s; F) X(s) ds}_{\text{memória}} + \underbrace{\eta(t)}_{\text{ruído}}.$$

Utilizando o OPM aliado ao formalismo de Mori-Zwanzig, obtemos um modelo reduzido para o sistema de interesse, no caso o modelo Lorenz 80, capaz de reproduzir com boa fidelidade a dinâmica multiescala e as estatísticas das variáveis lentas.

Para determinar o termo de ruído $\eta(t)$, analisamos as propriedades das oscilações rápidas presentes nos resíduos do modelo. O ponto de partida é o modelo BE, cuja variedade aproxima, sob as condições discutidas em CHEKROUN, LIU e McWILLIAMS (2017), a variedade OPM.

A seguir, emprega-se uma análise espectral do tipo Ruelle–Pollicott aplicada às séries temporais dos resíduos determinísticos. Essa análise permite identificar, de forma numérica, as frequências e taxas efetivas de decaimento das oscilações rápidas. Com esses parâmetros,

³ A definição formal e suas propriedades estão detalhadas em CHEKROUN, TANTET *et al.* (2020) e TANTET *et al.* (2020).

⁴ Explicado em detalhes no apêndice A

constrói-se uma rede de osciladores estocásticos de Stuart–Landau, cuja solução define os termos $\eta(t)$ usados para representar o efeito das componentes rápidas do sistema.

$$\begin{aligned}\dot{\eta}_1 &= (\alpha_1 + i\omega_1)\eta_1 - \beta_1\eta_1|\eta_1|^2 + \sum_{j \neq 1} c_{1j}\eta_j + \sum_{j=1}^3 Q_{1j}\dot{W}_t^j, \\ \dot{\eta}_2 &= (\alpha_2 + i\omega_2)\eta_2 - \beta_2\eta_2|\eta_2|^2 + \sum_{j \neq 2} c_{2j}\eta_j + \sum_{j=1}^3 Q_{2j}\dot{W}_t^j, \\ \dot{\eta}_3 &= (\alpha_3 + i\omega_3)\eta_3 - \beta_3\eta_3|\eta_3|^2 + \sum_{j \neq 3} c_{3j}\eta_j + \sum_{j=1}^3 Q_{3j}\dot{W}_t^j.\end{aligned}$$

onde $\alpha_j, \omega_j \in \mathbb{R}$ e $\beta_j, c_{ij} \in \mathbb{C}$ são coeficientes ajustados por regressão, e a matriz $Q = (Q_{ij})$ é escolhida de modo que QQ^\top coincida com a matriz de covariância desejada para o ruído. Com os ruídos definidos, podemos expressar o sistema BE-SLO como:

$$\begin{aligned}\frac{dY_i}{dt} &= -a_i^{-1}a_k b_k [\Phi_j(Y) + \Re \eta_j(t)] Y_k \\ &\quad - a_i^{-1}a_j b_j Y_j [\Phi_k(Y) + \Re \eta_k(t)] \\ &\quad + ca_i^{-1}(a_k - a_j) Y_j Y_k - [\Phi_i(Y) + \Re \eta_i(t)] - \nu_0 a_i Y_i,\end{aligned}$$

3.3.5 Simulações e comparação com o modelo PE

A simulação do modelo BE-SLO foi realizada em duas etapas principais: a geração do ruído estocástico e a execução do modelo propriamente dito. O código utilizado para a geração do ruído foi fornecido pelo professor Honghu Liu.⁵ A implementação da simulação do modelo BE-SLO está descrita no Apêndice B.5.

Foram adotados os mesmos parâmetros utilizados nas projeções bidimensionais apresentadas no Capítulo 1. A principal diferença está no fato de que, nesta simulação, o tempo é controlado pelo número de passos e pelo tamanho de passo, em vez de ser definido por um número fixo de dias. Na simulação apresentada, foram utilizados 5×10^5 passos com tamanho de passo igual a $0.00416666666667 \approx 4.2 \times 10^{-3}$. Esse valor totaliza aproximadamente 260 dias simulados. Essa escolha foi necessária porque o ruído gerado depende da discretização temporal em passos.

Com o objetivo de permitir uma comparação adequada com o modelo PE,⁶ adaptamos o código do modelo PE para empregar a mesma configuração temporal adotada na simulação do modelo BE-SLO. A seguir, apresentamos os resultados obtidos com o modelo BE-SLO, acompanhados de sua comparação com o modelo PE.

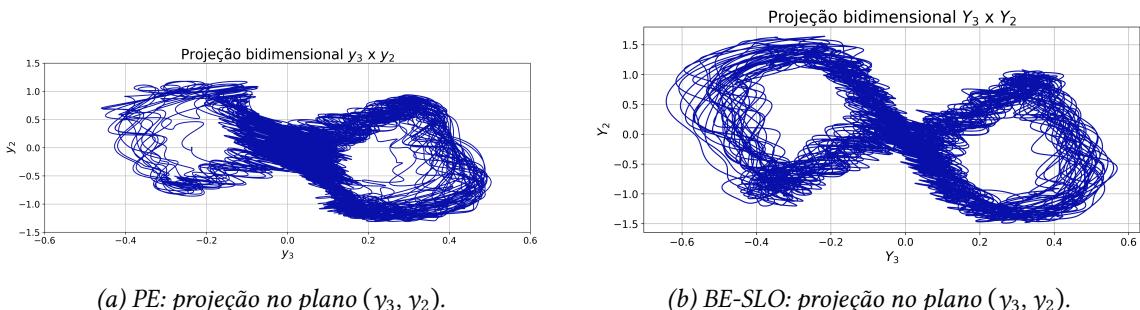


Figura 3.5: Comparação das projeções dos modelos PE e BE-SLO no plano (y_3, y_2)

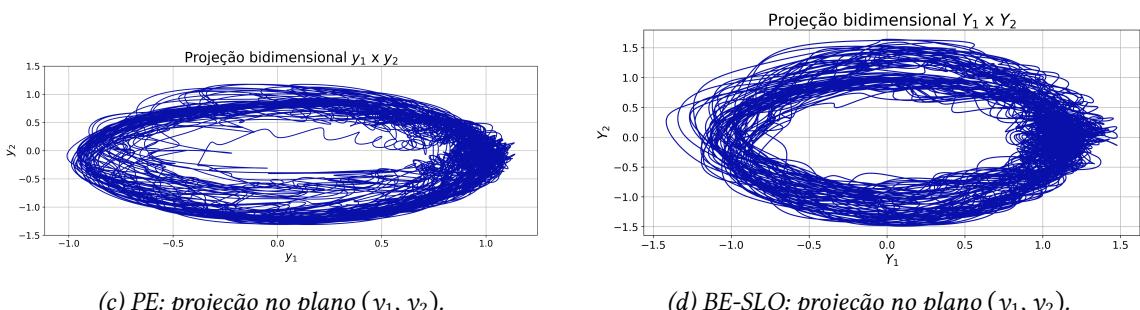


Figura 3.6: Comparação das projeções dos modelos PE e BE-SLO no plano (y_1, y_2)

⁵ Optou-se por omitir o código-fonte. A justificativa encontra-se na seção C.3.4

⁶ No artigo **CHEKROUN, LIU e McWILLIAMS (2021)**, os autores utilizam uma versão escalonada do modelo PE, conforme apresentada em **CHEKROUN, LIU e McWILLIAMS (2017)**, com a introdução de uma variável ε para representar a separação explícita das escalas de tempo. Neste trabalho, optou-se por seguir a formulação original proposta por **LORENZ (1980)**. Essa decisão pode gerar pequenas diferenças de escala, mas ainda deixa viável a comparação com o modelo estocástico

3.3 | LORENZ 80 BE-SLO

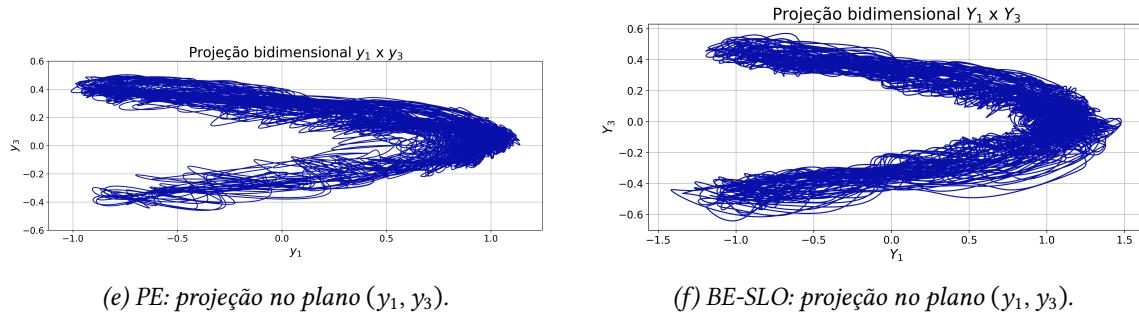


Figura 3.7: Comparação das projeções dos modelos PE e BE-SLO no plano (y_1, y_3)

Em relação às séries temporais das variáveis de y_i em $[100, 200]$, obtemos o seguinte resultado:

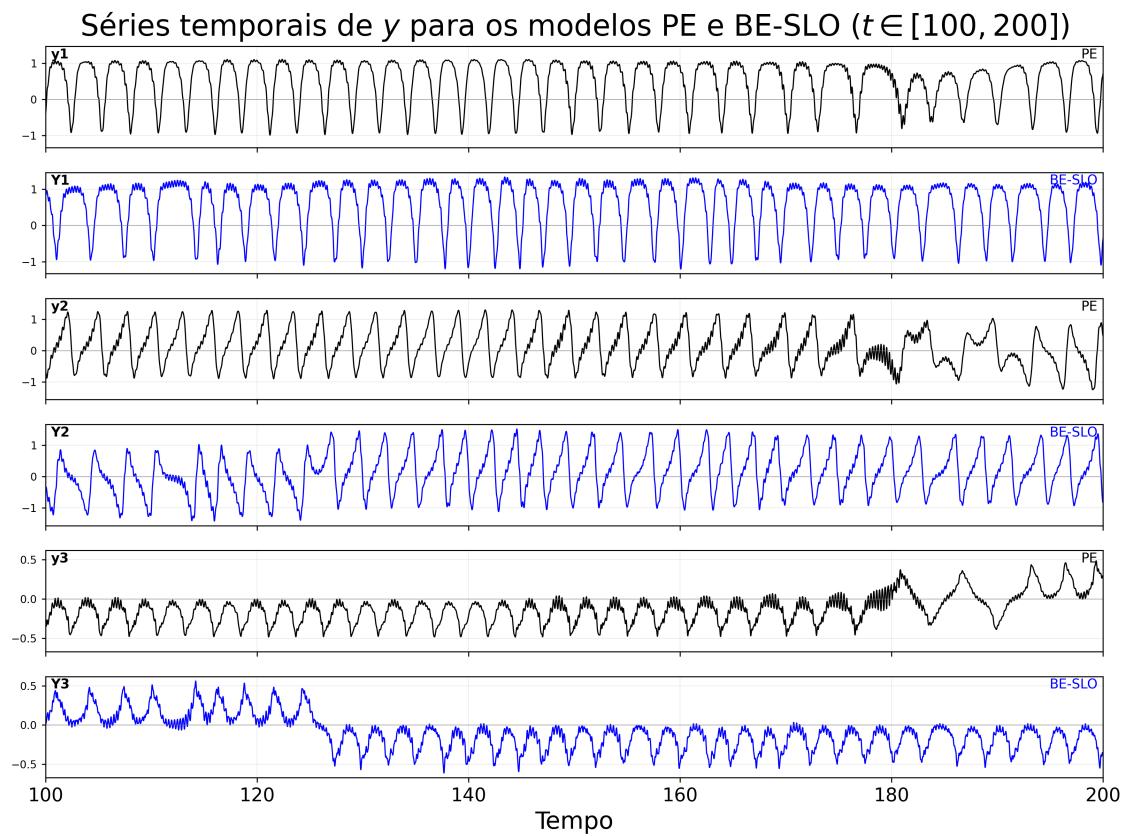


Figura 3.8: Comparação das séries temporais de y para os modelos PE e BE-SLO

Analizando as figuras comparativas das projeções bidimensionais dos modelos BE-SLO e PE nos planos (y_1, y_2) , (y_1, y_3) e (y_3, y_2) , observa-se uma notável semelhança entre os atratores gerados por ambos os modelos. A estrutura geral das trajetórias é preservada com padrões bem definidos indicando que o modelo estocástico BE-SLO consegue capturar as principais características dinâmicas do modelo determinístico PE.

Apesar disso, nota-se que o BE-SLO apresenta algumas diferenças sutis, como pequenas variações de amplitude e uma leve defasagem temporal em trechos alguns trechos.

Ainda assim, o comportamento qualitativo das variáveis y_i é bastante semelhante entre os dois modelos. Esses resultados reforçam que o modelo BE-SLO é uma boa aproximação estocástica para a dinâmica lenta do modelo PE.

Capítulo 4

Conclusões

Neste trabalho, estudamos o artigo de **CHEKROUN, LIU e McWILLIAMS (2021)**. No entanto, para que fosse possível compreender seu conteúdo com profundidade, foi necessário construir uma base teórica ao longo do trabalho.

O primeiro passo foi a análise do artigo de **LORENZ (1980)**, no qual investigamos a construção do modelo determinístico PE e suas variações: o modelo quasi-geostrófico (QG) e o modelo baseado em equações de balanço (BE), este último apresentado por **GENT e McWILLIAMS (1982)**. Essa etapa permitiu desenvolver uma compreensão sobre a motivação por trás dos modelos atmosféricos, bem como sobre os aspectos computacionais que tornam possível a simulação desses sistemas.

Em seguida, no segundo capítulo, realizamos um estudo introdutório de equações diferenciais estocásticas. Buscamos entender sua motivação, aplicações e propriedades fundamentais, utilizando como referência **PAVLIOTIS (2014)** e **EVANS (2014)**. Esse estudo serviu como preparação teórica para os modelos estocásticos analisados no capítulo seguinte.

No capítulo 3, abordamos o conteúdo central do artigo estudado, que envolve ferramentas modernas em sistemas dinâmicos e processos estocásticos. Para sua compreensão, foi necessário aprender conceitos avançados, como o formalismo de Mori-Zwanzig, com base em **CHORIN e HALD (2013)** e apresentado no apêndice A, além de noções de osciladores, ressonância e variedades, essenciais para a interpretação do texto.

Um elemento constante ao longo de todo o trabalho foi a aplicação de simulações computacionais. Um dos principais aprendizados foi compreender a estrutura dos códigos envolvidos, atuando na organização, adaptação e validação das implementações — com o apoio do *Github Copilot*. Esse processo foi essencial para garantir que os modelos fossem simulados corretamente, com equilíbrio entre precisão numérica e eficiência computacional.

Vale ressaltar que o sistema Lorenz 80 foi revisitado, e os códigos para simulação dos modelos PE, QG e BE foram reestruturados e disponibilizados publicamente em uma versão moderna no *GitHub*.

Por fim, este trabalho percorreu e interligou diversos temas centrais do curso de Bacharelado em Matemática Aplicada e Computacional com habilitação em Métodos

Matemáticos, especialmente nas áreas de estatística, computação, equações diferenciais e análise. Em suma, este trabalho consolidou grande parte do conhecimento adquirido, ampliou minha noção sobre pesquisa em matemática aplicada e possibilitou o contato com novos tópicos e o aprofundamento de outros já estudados.

Apêndice A

Formalismo Mori-Zwanzig

A.1 Introdução

Neste capítulo, apresentaremos o formalismo de Mori-Zwanzig (MZ). Para isso iniciamos com uma breve discussão sobre o formalismo MZ e sua importância no contexto deste trabalho (seção A.2). Em seguida, revisamos conceitos matemáticos fundamentais para sua formulação, como o operador de Liouville, a notação de semigrupos e os polinômios hermitianos (seção A.3). Por fim, na seção A.4, desenvolvemos o formalismo passo a passo, analisando cada termo de forma detalhada. Como referência utilizamos o trabalho do matemático Alexandre J. Chorin: [CHORIN, HALD e KUPFERMAN \(2000\)](#) e [CHORIN e HALD \(2013\)](#).

A.2 Motivação

O método de Mori-Zwanzig, desenvolvido por Robert Walter Zwanzig e Hajime Mori na segunda metade do século XX, é utilizado em sistemas hamiltonianos. Esse método consiste em classificar as variáveis do sistema em duas categorias: “resolvidas” e “não resolvidas”. As variáveis resolvidas são aquelas cujos comportamentos e valores são bem conhecidos, enquanto as não resolvidas são aquelas para as quais não se possui informações diretas. Para substituir essas variáveis não resolvidas, o método introduz termos estocásticos, denominados ruídos (*noise*), além de um termo de amortecimento (*damping*), também conhecido como termo de memória (*memory term*). Essa abordagem permite que o comportamento do sistema de interesse seja preservado de maneira adequada, mesmo sem conhecer completamente as variáveis não resolvidas.

O principal objetivo do artigo de [CHEKROUN, LIU e McWILLIAMS \(2021\)](#) é tratar o modelo de Lorenz 80 com uma abordagem estocástica, preservando seu comportamento. Para isso, utilizaremos o método de Mori-Zwanzig, que é uma abordagem física-estatística aplicável em sistemas como o Lorenz 80. Dada a relevância deste método para o trabalho de [CHEKROUN, LIU e McWILLIAMS \(2021\)](#), optamos por incluir uma introdução ao formalismo de Mori-Zwanzig, a fim de proporcionar uma melhor compreensão de sua aplicação no contexto do modelo de Lorenz 80 e permitir uma base teórica sólida para eventuais

explorações.

A.3 Preliminares

A.3.1 Convertendo sistemas de EDO não lineares em sistemas de EDPs lineares

Considere o sistema de equações diferenciais ordinárias (EDO) dado por:

$$\frac{d}{dt}\varphi(x, t) = R(\varphi(x, t)), \quad \varphi(x, 0) = x, \quad (\text{A.1})$$

onde R é uma função não linear, φ é uma função dependente do tempo, e R , φ e x podem assumir dimensões infinitas, sendo formados pelos vetores R_i , φ_i e x_i , respectivamente.

A partir disso, podemos definir o *Operador de Liouville* associado à equação (A.1) como:

$$L = \sum_i R_i(x) \frac{\partial}{\partial x_i} \quad (\text{A.2})$$

Utilizando o *Operador de Liouville*, podemos transformar o sistema de EDOs não lineares em um sistema de equações diferenciais parciais (EDPs) lineares da forma:

$$u_t = Lu, \quad u(x, 0) = g(x) \quad (\text{A.3})$$

A solução desse sistema existe, é única, e é dada por:

$$u(x, t) = g(\varphi(x, t)) \quad (\text{A.4})$$

Portanto, temos que a equação (A.3) é bem definida.¹

A.3.2 Notação de semigrupo

Definição A.1 (Semigrupo). *Um semigrupo é um conjunto não vazio X , munido de uma operação binária*

$$*: X \times X \rightarrow X,$$

que satisfaz a propriedade de associatividade:

$$(a * b) * c = a * (b * c), \quad \forall a, b, c \in X.$$

Isto posto, considere o operador Δ definido por:

$$\Delta\psi = \psi_{xx}, \quad \text{onde } \psi \text{ é uma função suave.}$$

¹ Detalhes da demonstração podem ser encontrados em CHORIN e HALD (2013, p. 181-182)

Agora, considere a equação diferencial

$$\frac{dv}{dt} - kv = 0, \quad v(0) = v_0,$$

cuja solução é $v(t) = v_0 e^{kt}$. De forma análoga, considere a equação do calor:

$$v_t - \frac{1}{2}\Delta v = 0, \quad v(x, 0) = \varphi(x),$$

onde v_t é a derivada de v em relação ao tempo e $\varphi(x)$ é a condição inicial. Em vez de resolver diretamente, expressamos a solução utilizando a notação de semigrupo:

$$v(\cdot, t) = e^{\frac{1}{2}t\Delta} \varphi.$$

Aqui, $e^{\frac{1}{2}t\Delta}$ é o semigrupo gerado pelo operador de difusão Δ . Ele é aplicado à condição inicial $\varphi(x)$, e a solução $v(\cdot, t)$ descreve a evolução temporal ao longo do tempo t . Essa notação permite representar soluções de forma compacta, explorando a estrutura associativa do semigrupo:

$$e^{\frac{1}{2}(t+s)\Delta} = e^{\frac{1}{2}t\Delta} e^{\frac{1}{2}s\Delta}.$$

Aplicando essa notação à equação (A.4), obtemos:

$$e^{tL} g(x) = g(\varphi(x, t)), \quad (\text{A.5})$$

Note que e^{tL} é um operador que age sobre funções; a expressão acima indica que a composição com g acompanha o fluxo $\varphi(x, t)$ gerado por L . Além disso, a função g é independente do tempo, e sua variação ocorre apenas pela evolução das variáveis do sistema. Assim, também vale a identidade

$$Le^{tL} = e^{tL}L. \quad (\text{A.6})$$

De forma análoga, para matrizes A e B obtém-se a fórmula de Duhamel:

$$\exp(t(A + B)) = \exp(tA) + \int_0^t \exp((t-s)(A+B))B \exp(sA) ds. \quad (\text{A.7})$$

Esta equação é bem definida e tem papel análogo ao da identidade anterior para operadores.

A.3.3 Polinômios Hermitianos

Primeiramente, definimos o produto interno como:

$$\langle u, v \rangle = \int_{-\infty}^{+\infty} \frac{e^{-x^2/2}}{\sqrt{2\pi}} u(x)v(x) dx \quad (\text{A.8})$$

Os polinômios $p_n(x)$ e $p_m(x)$ são ortonormais em relação a esse produto interno (A.8) quando satisfazem a seguinte condição:

$$\langle p_n, p_m \rangle = \int_{-\infty}^{\infty} e^{-x^2/2} p_n(x) p_m(x) dx = \delta_{nm} = \begin{cases} 1, & \text{se } n = m \\ 0, & \text{se } n \neq m \end{cases}$$

em que δ_{nm} é o delta de Kronecker.

No caso n -dimensional, o produto interno se generaliza para:

$$\langle u, v \rangle = \int_{-\infty}^{+\infty} \cdots \int_{-\infty}^{+\infty} (2\pi)^{-n/2} \exp\left(-\sum_{i=1}^n \frac{x_i^2}{2}\right) u(x)v(x) dx_1 \dots dx_n$$

De forma mais geral, se $H(q, p)$ é um Hamiltoniano, é possível definir uma família de polinômios nas variáveis q e p que sejam ortonormais com respeito à densidade canônica $Z^{-1} e^{-H/T}$. Os polinômios que satisfazem essa condição ainda são chamados de *polinômios hermitianos*.

Por fim, para o formalismo de Mori-Zwanzig, consideraremos um espaço n -dimensional Γ com uma densidade de probabilidade dada. Dividiremos as coordenadas em dois tipos: \hat{x} e \tilde{x} . Seja g uma função de x ; então $\mathbb{P}g = \mathbb{E}[g | \hat{x}]$ é uma projeção ortogonal sobre o subespaço das funções de \hat{x} . Temos que essa projeção gera um subespaço de polinômios hermitianos que são funções de \hat{x} e projetando sobre esses polinômios.

A.4 Mori-Zwanzig

A.4.1 Construção

Tomemos novamente o sistema (A.1), reproduzido abaixo:

$$\frac{d}{dt} \varphi(x, t) = R(\varphi(x, t)), \quad \varphi(x, 0) = x,$$

Lembremos que a equação é composta por componentes de dimensão n . Dentre essas n componentes, definimos as primeiras m componentes de φ , com $m < n$, como as variáveis de interesse. Em seguida, classificamos $\hat{\varphi}$ como as variáveis “resolvidas” e $\tilde{\varphi}$ como as variáveis “não resolvidas”:

$$\varphi = (\hat{\varphi}, \tilde{\varphi}), \quad \hat{\varphi} = (\varphi_1, \dots, \varphi_m), \quad \tilde{\varphi} = (\varphi_{m+1}, \dots, \varphi_n)$$

O mesmo vale para x e R : $x = (\hat{x}, \tilde{x})$ e $R = (\hat{R}, \tilde{R})$. A partir das variáveis resolvidas, buscamos criar previsões para o modelo de interesse, utilizando as soluções de uma parte da equação.

Utilizando o *Operador de Liouville* e a *notação de semigrupo*, podemos reescrever as

componentes de $\hat{\varphi}$ como²:

$$\hat{\varphi}_j(x, t) = e^{tL}x_j, \quad 1 \leq j \leq m$$

Ainda na notação de semigrupo, a equação dessas componentes é dada por:

$$\frac{\partial}{\partial t} e^{tL}x_j = L e^{tL}x_j = e^{tL}Lx \quad (\text{A.9})$$

A partir dos polinômios hermitianos, podemos definir *projeção ortogonal*, definimos \mathbb{P} como a projeção dada por: $\mathbb{P}g(x) = \mathbb{E}[g|\hat{x}]$. Assumimos que, no instante $t = 0$, conhecemos a densidade conjunta de todas as variáveis x , mas apenas os dados iniciais \hat{x} são conhecidos. A densidade das variáveis em \hat{x} é, então, a densidade conjunta de todas as variáveis x com \hat{x} fixado. Assim, \mathbb{P} é uma projeção sobre um espaço de funções com variáveis fixas e, portanto, independente do tempo.

As projeções $\mathbb{P}\hat{\varphi}(t) = \mathbb{E}[\hat{\varphi}(t)|\hat{x}]$ são de nosso maior interesse, pois estimam o comportamento do sistema a partir de um conjunto reduzido de variáveis.

Definindo $\mathbb{Q} = I - \mathbb{P}$ e considerando que as seguintes propriedades são válidas para quaisquer projeções ortogonais:

1. $\mathbb{P}^2 = \mathbb{P}$;
2. $\mathbb{Q}^2 = \mathbb{Q}$;
3. $\mathbb{P}\mathbb{Q} = 0$.

Podemos reescrever a equação (A.9) como:

$$\frac{\partial}{\partial t} e^{tL}x_j = e^{tL}\mathbb{P}Lx_j + e^{tL}\mathbb{Q}Lx_j \quad (\text{A.10})$$

Utilizando agora a *fórmula de Duhamel*, com $A = \mathbb{Q}L$ e $B = \mathbb{Q}L$, obtemos:

$$e^{tL} = e^{t\mathbb{Q}L} + \int_0^t e^{(t-s)L}\mathbb{P}Le^{s\mathbb{Q}L} ds \quad (\text{A.11})$$

Pela linearidade da equação de Liouville e a partir das equações (A.10) e (A.11), obtemos:

$$\frac{\partial}{\partial t} e^{tL}x_j = e^{tL}\mathbb{P}Lx_j + e^{t\mathbb{Q}L}\mathbb{Q}Lx_j + \int_0^t e^{(t-s)L}\mathbb{P}Le^{s\mathbb{Q}L}\mathbb{Q}Lx_j ds \quad (\text{A.12})$$

A equação acima é *equação de Mori-Zwanzig*.

² Note que cada componente depende de **todos** os valores de x . Portanto, se \hat{x} for aleatório, então $\hat{\varphi}$ também será.

A.4.2 Análise termo a termo

O primeiro termo é dado por:

$$e^{tL} \mathbb{P}Lx_j \quad (\text{A.13})$$

Observe que:

$$Lx_j = \sum_i R_i \left(\frac{\partial}{\partial x_i} \right) x_j = R_j(x)$$

Portanto,

$$\mathbb{P}Lx_j = \mathbb{E}[R_j(x) | \hat{x}]$$

Note que esta é uma função exclusivamente de \hat{x} . Enfim, podemos concluir que:

$$e^{tL} \mathbb{P}Lx_j = \bar{R}_j(\hat{\phi}(x, t))$$

Mais do que isso, o primeiro termo representa a dinâmica própria do sistema nas variáveis resolvidas. Além disso, trata-se de um termo markoviano, pois depende apenas do estado atual do sistema no tempo t .

Para o segundo termo, definimos:

$$w_j = e^{tQL} QLx_j$$

Por definição, temos:

$$\frac{\partial}{\partial t} w_j(x, t) = QLw_j(x, t), \quad (\text{A.14})$$

$$w_j(x, 0) = QLx_j = (I - \mathbb{P})R_j(x) = R_j(x) - \mathbb{E}[R_j | \hat{x}]. \quad (\text{A.15})$$

Note que $w_j(x, 0) = QLx_j = R_j(x) - \mathbb{E}[R_j(x) | \hat{x}]$ representa a *parte flutuante* da variável $R_j(x)$, ou seja, o componente imprevisível dado \hat{x} . Essa parte evolui de acordo com as *dinâmicas ortogonais*, de modo que $\mathbb{P}w_j(x, t) = 0$ para todo t , mantendo o termo como um ruído puramente não resolvido ao longo do tempo.

Mais especificamente, o subespaço do ruído (*noise subspace*) é formado pelas componentes das funções que são ortogonais às funções de \hat{x} , geralmente, isso corresponde a termos que dependem de \tilde{x} .

O terceiro termo, dado por:

$$\int_0^t e^{(t-s)L} \mathbb{P}L e^{sQL} QLx_j$$

é classificado como o termo de memória (*memory term*), já que este envolve a integração de quantidades que dependem de estados anteriores ao atual.

Tomemos que \mathbb{P} seja projetar na extensão dos polinômios hermitianos $H - 1, H_2, \dots$ com argumentos em \hat{x} . Assim, para dada função ψ , temos que: $\mathbb{P}\psi = \sum(\psi, H_k)H_k$, assim, temos:

$$\begin{aligned} \mathbb{P}Le^{sQL}QLx_j &= \mathbb{P}L(\mathbb{P} + Q)e^{sQL}QLx_j \\ &= \mathbb{P}LQe^{sQL}QLx_j \\ &= \sum_k \langle LQe^{sQL}QLx_j, H_k(\hat{x}) \rangle H_k(\hat{x}). \end{aligned}$$

O produto interno é definido como um valor esperado com respeito à densidade de probabilidade inicial. Vamos assumir que L é antissimétrico, ou seja, $(u, Lv) = -(Lu, v)$, então:

$$\begin{aligned} (LQe^{sQL}QLx_j, H_k(\hat{x})) &= -(Qe^{sQL}QLx_j, LH_k) \\ &= -(e^{sQL}QLx_j, QLH_k). \end{aligned}$$

Tanto QLx_j quanto QLH_k estão no subespaço de ruído, e $\mathbb{E}^{sQL}QLx_j$ é uma solução no tempo s da equação de dinâmica ortogonal com dados no subespaço de ruído; $PLe^{sQL}QLx_j$ é então uma soma de covariâncias temporais de ruídos.

Apêndice B

Programas

B.1 Lorenz 80 determinístico

```

1 import numpy as np
2 from scipy.integrate import solve_ivp
3 import pandas as pd
4 from phi import Phi
5 from parameters import a, b, c, h, f, nu_0, kappa_0, g_0
6
7 TIMESCALE_FACTOR = 8.0
8
9 def cyc(i):
10     return i, (i + 1) % 3, (i + 2) % 3
11
12 def pe_model(t, u):
13     x = u[0:3]
14     y = u[3:6]
15     z = u[6:9]
16     dx = np.zeros(3)
17     dy = np.zeros(3)
18     dz = np.zeros(3)
19
20     for i in range(3):
21         i_, j, k = cyc(i)
22         dx[i] = (
23             a[i] * b[i] * x[j] * x[k]
24             - c * (a[i] - a[k]) * x[j] * y[k]
25             + c * (a[i] - a[j]) * y[j] * x[k]
26             - 2 * c**2 * y[j] * y[k]
27             - nu_0 * (a[i]**2) * x[i]
28             + a[i] * y[i]
29             - a[i] * z[i]
30         ) / a[i]
31         dy[i] = (

```

```

32             -a[k] * b[k] * x[j] * y[k]
33             - a[j] * b[j] * y[j] * x[k]
34             + c * (a[k] - a[j]) * y[j] * y[k]
35             - a[i] * x[i]
36             - nu_0 * (a[i] ** 2) * y[i]
37         ) / a[i]
38         dz[i] = (
39             -b[k] * x[j] * (z[k] - h[k])
40             - b[j] * (z[j] - h[j]) * x[k]
41             + c * y[j] * (z[k] - h[k])
42             - c * (z[j] - h[j]) * y[k]
43             + g_0 * a[i] * x[i]
44             - kappa_0 * a[i] * z[i]
45             + f[i]
46         )
47     return np.concatenate([dx, dy, dz])
48
49 def pe_simulate(x0, y0, z0, days):
50     initial_u = np.concatenate([x0, y0, z0])
51     t_final = days * TIMESCALE_FACTOR
52     t_span = (0, t_final)
53
54     print("Iniciando integracao numerica para o modelo PE...")
55     sol = solve_ivp(
56         pe_model, t_span, initial_u, method="RK45", atol=1e-8, rtol=1e
57         -6
58     )
59     print("Aguarde...")
60
61     t = sol.t / TIMESCALE_FACTOR
62     x, y, z = sol.y[0:3].T, sol.y[3:6].T, sol.y[6:9].T
63
64     df = pd.DataFrame({
65         "time": t,
66         "x1": x[:, 0], "x2": x[:, 1], "x3": x[:, 2],
67         "y1": y[:, 0], "y2": y[:, 1], "y3": y[:, 2],
68         "z1": z[:, 0], "z2": z[:, 1], "z3": z[:, 2],
69     })
70     return df
71
72 def be_model(tau, y, phi_vec):
73     dydt = np.zeros(3, dtype=float)
74     for i in range(3):
75         i_, j, k = cyc(i)
76         dydt[i] = (
77             - (a[k] * b[k] / a[i]) * phi_vec[j] * y[k]
78             - (a[j] * b[j] / a[i]) * y[j] * phi_vec[k]
79             + (c * (a[k] - a[j]) / a[i]) * y[j] * y[k]
80             - phi_vec[i]

```

B.1 | LORENZ 80 DETERMINÍSTICO

```

80             - nu_0 * a[i] * y[i]
81         )
82     return dydt
83
84 def qg_model(t, y):
85     dy = np.zeros(3, dtype=float)
86     for i in range(3):
87         _, j, k = cyc(i)
88         dy[i] = (
89             g_0 * c * (a[k] - a[j]) * y[j] * y[k]
90             - a[i] * (a[i] * g_0 * nu_0 + kappa_0) * y[i]
91             - c * h[k] * y[j]
92             + c * h[j] * y[k]
93             + f[i]
94         ) / (a[i] * g_0 + 1.0)
95     return dy
96
97 def simulate_model(model_func, y0, days, model_name, dt=0.001,
98     model_args=()):
99     t_final = days * TIMESCALE_FACTOR
100    t_span = (0, t_final)
101    t_eval = np.arange(0, t_final + dt, dt)
102
103    print(f"Iniciando integracao numerica para o modelo {model_name}
104        ....")
105    sol = solve_ivp(
106        model_func, t_span, y0, t_eval=t_eval, args=model_args,
107        method="RK45", atol=1e-8, rtol=1e-6
108    )
109    print("Aguarde...")
110
111    t = sol.t / TIMESCALE_FACTOR
112    y = sol.y
113
114    df = pd.DataFrame({
115        "time": t,
116        "y1": y[0], "y2": y[1], "y3": y[2],
117    })
118    return df
119
120 def be_simulate(y0, days):
121     phi_vec = Phi(y0)
122     return simulate_model(be_model, y0, days, "BE", model_args=(
123         phi_vec,))
124
125 def qg_simulate(y0, days):
126     return simulate_model(qg_model, y0, days, "QG")

```

Programa B.1: Simulação do modelo L80 PE, QG e BE

B.2 Simulação de preço de uma ação de mercado

```

1  using DifferentialEquations, DataFrames, CSV
2  import StochasticDiffEq as SDE
3
4  f(u,p,t) = mu*u
5  g(u,p,t)= sigma
6  dt = 1/(365*24)
7  mu = 0.05
8  sigma = 1.0
9  tspan = (0.0, 1.0)
10 u0= 12
11 prob = SDEProblem(f,g,u0,tspan, seed=13865062)
12 sol = SDE.solve(prob, EM(), dt = dt);
13
14 df = DataFrame(t = sol.t, x = sol.u)
15 outfile = joinpath(@__DIR__, "stockmarket.csv")
16 CSV.write(outfile, df)
17 println("Dados salvos em: \$outfile")

```

Programa B.2: Simulação do preço de uma ação no mercado

B.3 Fast Chaotic Noise

```

1  import numpy as np
2  import pandas as pd
3  from scipy.integrate import solve_ivp
4  from pathlib import Path
5  import time
6
7  eps = 0.1
8  lam = 1.0
9
10 def rhs(t, u):
11     x, y1, y2, y3 = u
12     dx = x - x**3 + (lam/eps) * y2
13     dy1 = (10 / eps**2) * (y2 - y1)
14     dy2 = (1 / eps**2) * (28*y1 - y2 - y1*y3)
15     dy3 = (1 / eps**2) * (y1*y2 - (8/3)*y3)
16     return [dx, dy1, dy2, dy3]
17
18
19 p = 2.0
20 rtol = 1e-3 * eps**p
21 atol = 1e-6 * eps**p
22
23 print(f"Usando rtol={rtol:.2e}, atol={atol:.2e}")
24

```

```

25 t0, tf = 0.0, 100.0
26 x0 = 0.1
27 y0 = [x0, 1e-2, 1e-2, 1e-2]
28 t_eval = np.arange(t0, tf, 1e-3)
29
30 print("Iniciando integracao numerica...")
31
32 start_time = time.time()
33 sol = solve_ivp(rhs, (t0, tf), y0, method="RK45", t_eval=t_eval, rtol=
    rtol, atol=atol)
34 end_time = time.time()
35
36 execution_time = end_time - start_time
37
38 x_vals = sol.y[0]
39 y2_vals = sol.y[2]
40
41 BASE = Path(__file__).resolve().parent if "__file__" in globals() else
    Path(".")
42 DATADIR = BASE / "data"
43 DATADIR.mkdir(parents=True, exist_ok=True)
44 eps_str = str(eps).replace(".", "")
45 out_file = DATADIR / f"deterministico_{eps_str}.csv"
46
47 print("\nSalvando dados...")
48 df = pd.DataFrame({"t": sol.t, "y2": y2_vals, "x": x_vals})
49 with open(out_file, 'w') as f:
50     f.write(f'# eps={eps}, rtol={rtol:.2e}, atol={atol:.2e}, tempo={
51         execution_time:.2f}s\n')
52     df.to_csv(f, index=False)
53 print(f"Dados salvos em: {out_file}")

```

Programa B.3: Simulação do modelo determinístico

```

1 import numpy as np
2 import pandas as pd
3 import os
4
5 def sigma_from_y2(y2, dt, lam=1.0):
6     y = y2 - np.mean(y2)
7     n = len(y)
8     nfft = n
9     fy = np.fft.rfft(y, n=nfft)
10    ac = np.fft.irfft(fy * np.conj(fy), n=nfft)[:n] / np.arange(n, 0,
-1)
11    sigma2 = abs(2 * lam**2 * dt * np.sum(ac))
12    return np.sqrt(max(sigma2, 0.0))
13
14

```

```

15 csv_path = os.path.join(os.path.dirname(__file__), "data", "deterministico_001.csv")
16 df = pd.read_csv(csv_path, comment="#")
17 dt = df["t"][1] - df["t"][0]
18 sigma = sigma_from_y2(df["y2"].to_numpy(), dt)
19 print("sigma", sigma)

```

Programa B.4: Cálculo do sigma

```

1 using DifferentialEquations, DataFrames, CSV, Random
2 eps = 0.1
3 # Valores calculados previamente
4 if eps == 0.5
5     sigma = 242.69367736299662
6 elseif eps == 0.2
7     sigma = 205.05745681770404
8 elseif eps == 0.1
9     sigma = 142.6677313864854
10 elseif eps == 0.01
11     sigma = 5.754046294922591
12 else
13     error("Valor de epsilon invalido. Use 0.5, 0.2 ou 0.1.")
14     exit()
15 end
16 x0 = 0.1
17 tspan = (0.0, 100.0)
18
19 f1(X, p, t) = X - X^3
20 f2(X, p, t) = p
21
22 prob = SDEProblem(f1, f2, x0, tspan, sigma)
23 solucao_estocastico = solve(prob, EM(), dt = 1e-3, seed = 13865062)
24 df = DataFrame(t = solucao_estocastico.t, x = solucao_estocastico.u)
25 cd(@__DIR__)
26
27 if eps == 0.5
28     eps_suffix = "_05"
29 elseif eps == 0.2
30     eps_suffix = "_02"
31 elseif eps == 0.1
32     eps_suffix = "_01"
33 elseif eps == 0.01
34     eps_suffix = "_001"
35 end
36
37 filename = "data/estocastico$(eps_suffix).csv"
38 CSV.write(filename, df)
39 println("Dados estocasticos salvos em: $filename")

```

Programa B.5: Simulação do preço de uma ação no mercado

B.4 Oscilação de Stuart Landau estocástica

```

1  using DifferentialEquations
2  using CSV
3  using DataFrames
4
5  alpha = 0.5
6  omega = 2.0
7  beta = 1.0 + 0.5im
8  sigma = 0.25
9
10 f(z, p, t) = (alpha + im*omega) * z - beta * z * abs2(z)
11 g(z, p, t) = sigma
12
13 z0      = 1.0 + 0.0im
14 tspan = (0.0, 75.0)
15
16 prob = SDEProblem(f, g, z0, tspan)
17 sol  = solve(prob, EM(), dt=0.001)
18
19 df = DataFrame(t=sol.t, real=real.(sol.u), imag=imag.(sol.u))
20
21 outfile = joinpath(@_\_DIR\_\_, "slo.csv")
22 CSV.write(outfile, df)
23 println("Dados salvos em: \$outfile")

```

Programa B.6: Simulação de oscilação de Stuart Landau estocástica

B.5 Lorenz 80 BE-SLO

```

1  import numpy as np
2  import pandas as pd
3  from pathlib import Path
4  from scipy.integrate import solve_ivp
5  from parameters import a, b, c, h, f, nu_0, g_0, y
6  from phi import Phi
7
8  try:
9      BASE = Path(__file__).resolve().parent
10 except NameError:
11     BASE = Path.cwd()
12
13 OUTDIR = BASE / "data"
14 OUTDIR.mkdir(parents=True, exist_ok=True)
15 SAVE_PATH = OUTDIR / "ruido_slo.csv"
16
17 def cyc(i: int):
18     return i, (i + 1) % 3, (i + 2) % 3

```

```

19
20 def carregar_ruido(filepath=SAVE_PATH):
21     df = pd.read_csv(filepath)
22     t = df["time"].to_numpy()
23     dt = np.mean(np.diff(t))
24
25
26     eta_cols = ["eta1_real", "eta2_real", "eta3_real"]
27     eta = df[eta_cols].to_numpy(dtype=float)
28     return eta, dt, t
29
30 def be_slo_model(Y: np.ndarray, n: int, eta: np.ndarray) -> np.ndarray:
31     PhiY = Phi(Y)
32     eta_n = eta[n]
33
34     dY = np.zeros(3, dtype=float)
35     for i in range(3):
36         _, j, k = cyc(i)
37
38         dY[i] = (
39             -(a[k] * b[k] * (PhiY[j] + eta_n[j]) * Y[k]) / a[i]
40             - (a[j] * b[j] * Y[j] * (PhiY[k] + eta_n[k])) / a[i]
41             + (c * (a[k] - a[j]) * Y[j] * Y[k]) / a[i]
42             - (PhiY[i] + eta_n[i])
43             - nu_0 * a[i] * Y[i]
44         )
45     return dY
46
47
48
49 def simulate_be_slo(Y0: np.ndarray, eta: np.ndarray, dt: float):
50     N = len(eta)
51     t0 = 0.0
52     t_span = [t0, (N - 1) * dt]
53     t_eval = t0 + np.arange(N) * dt
54
55     def rhs_wrapper(t, y):
56         n = int(round(t / dt))
57         # Ensure n is within bounds
58         n = min(n, N - 1)
59         return be_slo_model(y, n, eta)
60
61     sol = solve_ivp(
62         rhs_wrapper,
63         t_span,
64         Y0.astype(float),
65         t_eval=t_eval,
66         method="RK45",

```

```
67 )
68
69 t = sol.t/8
70 Y = sol.y.T
71
72 out = pd.DataFrame({"time": t, "Y1": Y[:, 0], "Y2": Y[:, 1], "Y3":
73 Y[:, 2]})  
74 save_path = OUTDIR / "be_slo.csv"  
75 out.to_csv(save_path, index=False)  
76  
return t, Y, save_path
```

Programa B.7: *Modelo BE-SLO*

Apêndice C

Simulações computacionais

C.1 Introdução

Este apêndice apresenta os principais aspectos relacionados às simulações computacionais desenvolvidas no contexto deste trabalho, com ênfase na escolha dos *solvers*, no seu funcionamento e nas decisões adotadas durante a implementação dos códigos.

Na Seção C.2, discutimos os fundamentos teóricos dos *solvers* utilizados e suas propriedades. Em seguida, na Seção C.3, descrevemos as decisões de implementação, dificuldades enfrentadas, soluções adotadas e observações relevantes sobre o processo de desenvolvimento computacional.

C.2 *Solvers*

Para as simulações computacionais realizadas neste trabalho, foram utilizados *solvers* prontos disponíveis nas bibliotecas das linguagens de programação *Python* e *Julia*. De forma geral, a linguagem *Python* foi empregada para a simulação de sistemas determinísticos modelados por EDOs, majoritariamente, enquanto a linguagem *Julia* foi utilizada para a modelagem de EDEs.

Nas simulações em *Python*, foi adotado o método explícito de Runge-Kutta de ordem 5(4) (RK45), por meio da biblioteca *SciPy*, versão 1.6.2. Já nas simulações em *Julia*, utilizou-se o método de Euler-Maruyama não-rígido (EM), disponível na biblioteca *SciML*, versão 7.17.0. A seguir, será apresentada uma descrição do funcionamento dos *solvers* utilizados, com base nas referências disponíveis na documentação oficial das respectivas bibliotecas.

C.2.1 RK45 em *Python*

O método RK45 é um método amplamente utilizado para a simulação de sistemas de EDOs. De acordo com ROMA *et al.* (2023), um método Runge-Kutta é definido da seguinte forma:

Um método é de *Runge–Kutta explícito* de ordem q se, e só se, satisfaz três propriedades:

1. é um método de passo único explícito;
2. tem as derivadas de $y(t)$ (e, portanto, de $D^j f(t, y)$) aproximadas em pontos “convenientemente escolhidos”; e, por fim,
3. “concorda” com o Método da Série de Taylor até o termo de q -ésima ordem, para algum $q > 0$.

Definimos um método RK explícito com R estágios de forma genérica como

$$\begin{cases} y_0 = y(t_0), \\ y_{k+1} = y_k + h \Phi(t_k, y_k, h), \end{cases}$$

onde $\Phi(t, y, h) = \sum_{r=1}^R c_r \kappa_r(t, y, h)$

Ainda segundo [ROMA et al. \(2023\)](#), em particular o método RK45 é definido da seguinte maneira:

$$\Phi(t, y, h) = \frac{25}{216} \kappa_1 + \frac{1408}{2565} \kappa_3 + \frac{2197}{4104} \kappa_4 - \frac{1}{5} \kappa_5,$$

com

$$\begin{cases} \kappa_1 = f(t, y), \\ \kappa_2 = f\left(t + \frac{h}{4}, y + \frac{h}{4} \kappa_1\right), \quad \kappa_3 = f\left(t + \frac{3}{8}h, y + \frac{3}{32}h\kappa_1 + \frac{9}{32}h\kappa_2\right), \\ \kappa_4 = f\left(t + \frac{12}{13}h, y + \frac{1932}{2197}h\kappa_1 - \frac{7200}{2197}h\kappa_2 + \frac{7296}{2197}h\kappa_3\right), \\ \kappa_5 = f\left(t + h, y + \frac{439}{216}h\kappa_1 - 8h\kappa_2 + \frac{3680}{513}h\kappa_3 - \frac{845}{4104}h\kappa_4\right). \end{cases}$$

O método RK45 utilizado no *SciPy* possui algumas modificações que permitem uma melhor performance computacional.

Primeiro, temos que implementa o par embutido de Dormand–Prince, proposto em [DORMAND e PRINCE \(1980\)](#). Nesse esquema, duas fórmulas de Runge–Kutta compartilham os mesmos estágios: uma de ordem 5, utilizada para avançar a solução, e outra de ordem 4, utilizada para estimar o erro local e adaptar o passo.

Além disso, utiliza um polinômio interpolador de ordem 4 para produzir saídas densas entre pontos consecutivos da malha, conforme descrito em [SHAMPINE \(1986\)](#). Esse polinômio denso é ajustado a partir dos valores e derivadas obtidos nos extremos de cada passo, fornecendo uma aproximação contínua e suavemente diferenciável (C^1) da solução. Sua principal vantagem é permitir a avaliação de $y(t)$ em qualquer ponto intermediário $t \in [t_n, t_{n+1}]$ sem a necessidade de recomputar o integrador, aumentando a eficiência e a precisão da integração numérica.

C.2.2 Euler-Maruyama em *Julia*

O método Euler-Maruyama é um método de discretização para EDEs. Nas simulações feitas para este trabalho, ele foi selecionado pelo fato de todas as EDEs simuladas serem consideradas simples, na forma de Itô e com escala de tempo pequena.

De acordo com **KLOEDEN e PLATEN, 1992**, para uma EDE do tipo:

$$dX_t = \mu(t, X_t) dt + \sigma(t, X_t) dW_t$$

Para $t_0 \leq t \leq T$, onde t_0 representa o tempo inicial e T o tempo final, com valor inicial definido por: $X_{t_0} = X_0$, discretizamos o tempo da seguinte maneira:

$$t_0 = \tau_0 < \tau_1 < \tau_2 < \dots < \tau_{N-1} < \tau_N = T$$

Definimos $Y_0 = X_0$ e recursivamente, definimos $Y_n + 1$ como:

$$Y_{n+1} = Y_n + \mu(\tau_n, Y_n)(\tau_{n+1} - \tau_n) + \sigma(\tau_n, Y_n)(W_{\tau_{n+1}} - W_{\tau_n}),$$

As propriedades do método apresentadas em **RACKAUCKAS e NIE (2017)**, são:

- Intervalo de tempo: apenas intervalo de tempo fixo
- Tipos de ruído: todas as formas (ruído diagonal, não diagonal, escalar, aditivo e colorido)
- Interpretação EDE: Itô

Em relação à convergência do método, temos duas abordagens possíveis: a convergência forte e a convergência fraca (**KLOEDEN e PLATEN, 1992**). A convergência forte baseia-se no erro absoluto dado por:

$$\varepsilon = \mathbb{E}(|X_T - Y(T)|)$$

onde X_T é a variável aleatória dada pelo processo de Itô e $Y(T)$ é a aproximação numérica. Dizemos que um processo converge fortemente quando

$$\lim_{\delta \rightarrow 0^+} \mathbb{E}(|X_T - Y^\delta(T)|) = 0$$

onde δ representa o tamanho do passo do método. Além disso, definimos a ordem $\gamma > 0$: no instante T , existir uma constante $C > 0$, que não depende de δ , e um $\delta_0 > 0$ tais que

$$\varepsilon(\delta) = \mathbb{E}(|X_T - Y^\delta(T)|) \leq C \delta^\gamma$$

para cada $\delta \in (0, \delta_0)$.

A convergência fraca é inicialmente motivada pelo erro no primeiro momento:

$$\mu = \mathbb{E}(Y(T)) - \mathbb{E}(X_T)$$

Para obter uma definição geral, introduzimos uma função-teste suave $g : \mathbb{R}^d \rightarrow \mathbb{R}$. Nesse contexto, o erro fraco pode ser escrito como

$$\mu_g = \mathbb{E}(g(Y(T))) - \mathbb{E}(g(X_T))$$

Dizemos que uma aproximação Y^δ converge fracamente para X em T , quando $\delta \rightarrow 0^+$,

se satisfaz a condição abaixo:

$$\lim_{\delta \rightarrow 0^+} \left| \mathbb{E}(g(X_T)) - \mathbb{E}(g(Y^\delta(T))) \right| = 0$$

Definimos a ordem $\beta > 0$ da convergência fraca para X no instante T quando $\delta \downarrow 0$ se, para cada $g \in C_P^{2(\beta+1)}(\mathbb{R}^d, \mathbb{R})$, existir $C > 0$, que não depende de δ , e um $\delta_0 > 0$ tais que

$$\left| \mathbb{E}(g(X_T)) - \mathbb{E}(g(Y^\delta(T))) \right| \leq C \delta^\beta$$

para cada $\delta \in (0, \delta_0)$.

Segundo [RACKAUCKAS e NIE \(2017\)](#), o método de Euler–Maruyama possui ordem de convergência forte igual a 0.5 (no sentido de Itô) e ordem de convergência fraca igual a 1.0.

C.3 Decisões de Implementação e Observações

Durante o processo de desenvolvimento das simulações computacionais, algumas escolhas se mostraram essenciais para a evolução do projeto. A seguir, apresentamos algumas considerações e observações feitas até a versão final.

C.3.1 Linguagem *Julia*

A ideia inicial era utilizar exclusivamente a linguagem *Julia* para as simulações computacionais, contudo alguns desafios surgiram ao longo do caminho. Inicialmente, os códigos foram organizados em *Jupyter Notebooks*, com duas funções principais: realizar as simulações e gerar os gráficos. Contudo, essa abordagem foi descartada. O motivo principal foi que, para o *fast chaotic noise* descrito na seção 3.2, o código ultrapassava o limite de memória RAM de 16GB quando $\varepsilon = 0.1$.

Para contornar o problema, decidimos abandonar o uso dos arquivos *Jupyter Notebook* e migramos para arquivos *Julia* (.jl). Além disso, ao invés de realizar a simulação e a plotagem no mesmo arquivo, optamos por dividir o processo: em um arquivo, a simulação geraria um arquivo .csv, enquanto em outro arquivo seria feita a geração dos gráficos. Essa mudança foi crucial para o progresso do trabalho, como será detalhado nas seções seguintes.

Com essas alterações, o exemplo simplificado começou a funcionar, mas ainda com limitações. Quando $\varepsilon = 0.1$, o modelo falhava, e para $\varepsilon = 0.01$, a simulação não funcionava. Como o correto funcionamento para $\varepsilon = 0.01$ era essencial, decidimos testar o código em *Python*, que se comportou conforme o esperado. Os outros códigos relacionados ao modelo Lorenz 80 haviam sido escritos originalmente em *Python*, em um trabalho anterior que também utilizava a abordagem de *Jupyter Notebook*, para o trabalho, apenas convertemos para arquivos *Python* (.py). Assim, buscando maior consistência no trabalho, optamos por adotar o *Python* como a linguagem para os sistemas determinísticos.

Para os sistemas estocásticos, no entanto, escolhemos continuar com *Julia*, com excessão do modelo Lorenz 80 BE-SLO. Isso se deve ao fato de que esses sistemas são mais simples e não exigem grande poder computacional. Além disso, a biblioteca *Python sdeint*, que

possui *solvers* para equações diferenciais estocásticas (EDE), não recebe tanta manutenção e suporte quanto a *SciML* de *Julia*. A *SciML* não só é bem documentada, mas também conta com tutoriais em vídeo, que foram de grande ajuda para introdução à linguagem.

C.3.2 Estrutura geral dos programas de simulação

Todos os arquivos de simulação seguem uma estrutura padronizada. Um programa principal é responsável por executar as simulações e gerar os dados no formato `.csv`, que são armazenados na pasta `data`. Paralelamente, um segundo programa é dedicado à criação das imagens no formato `.png`, salvas na pasta `img`. Essa organização mostrou-se particularmente vantajosa pelos seguintes motivos:

1. **Organização.** A separação das funcionalidades em arquivos distintos facilita a manutenção, a leitura e a modificação do código.
2. **Desempenho.** A divisão entre simulação e visualização contribuiu para mitigar problemas de performance, conforme discutido na seção anterior.
3. **Gerenciamento de dados.** A utilização de diretórios específicos para dados e imagens facilita o compartilhamento e está alinhada com boas práticas de gestão de arquivos em projetos computacionais.

C.3.3 Sistema integrado para diferentes versões do modelo L80

Com o objetivo de facilitar a comparação entre as diferentes formulações do modelo de Lorenz 80 PE, QG e BE, foi desenvolvido um sistema integrado que unifica a execução desses modelos em um único programa. A estrutura do código é dividida em seis módulos principais, o que favorece a organização, a manutenção e a extensibilidade do sistema:

1. `initial_conditions.py` – Define as condições iniciais das variáveis x , y e z .
2. `parameters.py` – Armazena os parâmetros utilizados nas equações e na definição das condições iniciais, centralizando valores que podem ser facilmente ajustados.
3. `phi.py` – Implementa a função $\Phi(y)$, conforme definida na equação (1.4.3), utilizada no modelo BE.
4. `plot.py` – Responsável pela geração dos gráficos das soluções obtidas, permitindo a visualização dos resultados de forma automatizada.
5. `simulations.py` – Centraliza as rotinas de simulação dos modelos PE, QG e BE, integrando o sistema de equações com os métodos numéricos definidos.
6. `main.py` – Gerencia o fluxo de execução e a interação com o usuário, permitindo a escolha do modelo.

Por fim, podemos esquematizar o funcionamento do programa da seguinte maneira:

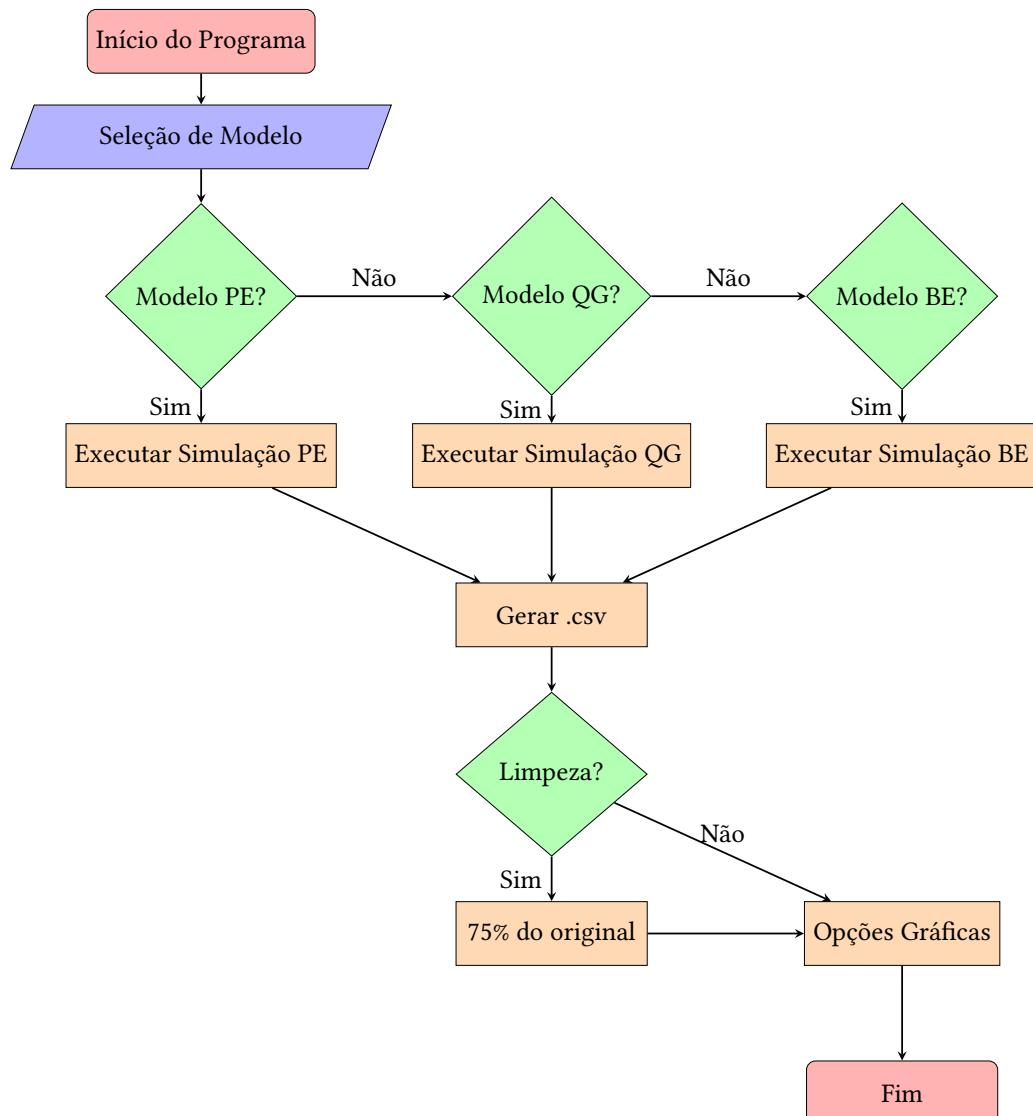


Figura C.1: Fluxograma do Processo de Simulação

C.3.4 Código de geração do ruído do Lorenz 80 BE-SLO

Optou-se por omitir o código-fonte da geração de ruído do modelo Lorenz 80 BE-SLO. Tal decisão fundamenta-se no fato de que os arquivos responsáveis pela geração do ruído do modelo foram gentilmente cedidos pelo professor Honghu Liu, coautor de [CHEKROUN, LIU e McWILLIAMS \(2021\)](#). Como não houve especificação explícita quanto aos direitos de uso e compartilhamento desse material, decidiu-se pela não inclusão do referido código neste relatório.

Referências

- [ARFKEN e WEBER 2005] George B ARFKEN e Hans J WEBER. *Mathematical methods for physicists international student edition*. en. 6^a ed. San Diego, CA: Academic Press, jun. de 2005 (citado na pg. 7).
- [CHEKROUN e LIU 2014] Mickaël D. CHEKROUN e Honghu LIU. “Finite-horizon parameterizing manifolds, and applications to suboptimal control of nonlinear parabolic pdes”. *Acta Applicandae Mathematicae* 135.1 (set. de 2014), pp. 81–144. ISSN: 1572-9036. DOI: [10.1007/s10440-014-9949-1](https://doi.org/10.1007/s10440-014-9949-1) (citado na pg. 29).
- [CHEKROUN, LIU e McWILLIAMS 2017] Mickaël D. CHEKROUN, Honghu LIU e James C. McWILLIAMS. “The emergence of fast oscillations in a reduced primitive equation model and its implications for closure theories”. *Computers & Fluids* 151 (jun. de 2017), pp. 3–22. ISSN: 0045-7930. DOI: [10.1016/j.compfluid.2016.07.005](https://doi.org/10.1016/j.compfluid.2016.07.005) (citado nas pgs. 10, 29, 30, 32, 34).
- [CHEKROUN, LIU e McWILLIAMS 2021] Mickaël D. CHEKROUN, Honghu LIU e James C. McWILLIAMS. “Stochastic rectification of fast oscillations on slow manifold closures”. *Proceedings of the National Academy of Sciences* 118.48 (nov. de 2021). ISSN: 1091-6490. DOI: [10.1073/pnas.2113650118](https://doi.org/10.1073/pnas.2113650118) (citado nas pgs. iii, v, 1, 9, 12, 25, 28, 30, 32, 34, 37, 39, 62).
- [CHEKROUN, LIU e WANG 2015] Mickaël D. CHEKROUN, Honghu LIU e Shouhong WANG. *Stochastic Parameterizing Manifolds and Non-Markovian Reduced Equations: Stochastic Manifolds for Nonlinear SPDEs II*. Springer International Publishing, 2015. ISBN: 9783319125206. DOI: [10.1007/978-3-319-12520-6](https://doi.org/10.1007/978-3-319-12520-6) (citado na pg. 29).
- [CHEKROUN, TANTET *et al.* 2020] Mickaël D. CHEKROUN, Alexis TANTET, Henk A. DIJKSTRA e J. David NEELIN. “Ruelle–pollicott resonances of stochastic systems in reduced state space. part i: theory”. *Journal of Statistical Physics* 179.5–6 (abr. de 2020), pp. 1366–1402. ISSN: 1572-9613. DOI: [10.1007/s10955-020-02535-x](https://doi.org/10.1007/s10955-020-02535-x) (citado na pg. 32).
- [CHORIN e HALD 2013] Alexandre J. CHORIN e Ole H. HALD. *Stochastic Tools in Mathematics and Science*. Springer New York, 2013. ISBN: 9781461469803. DOI: [10.1007/978-1-4614-6980-3](https://doi.org/10.1007/978-1-4614-6980-3) (citado nas pgs. 37, 39, 40).

- [CHORIN, HALD e KUPFERMAN 2000] Alexandre J. CHORIN, Ole H. HALD e Raz KUPFERMAN. *Optimal prediction and the Mori–Zwanzig representation of irreversible processes*. 2000. DOI: [10.1073/pnas.97.7.2968](https://doi.org/10.1073/pnas.97.7.2968) (citado na pg. 39).
- [DORMAND e PRINCE 1980] J.R. DORMAND e P.J. PRINCE. “A family of embedded runge-kutta formulae”. *Journal of Computational and Applied Mathematics* 6.1 (1980), pp. 19–26. ISSN: 0377-0427. DOI: [10.1016/0771-050X\(80\)90013-3](https://doi.org/10.1016/0771-050X(80)90013-3) (citado na pg. 58).
- [EVANS 2014] Lawrence C. EVANS. *An Introduction to Stochastic Differential Equations*. Providence, RI: American Mathematical Society, jan. de 2014 (citado nas pgs. 18–23, 37).
- [GENT e McWILLIAMS 1982] Peter R. GENT e James C. McWILLIAMS. “Intermediate model solutions to the lorenz equations: strange attractors and other phenomena”. *Journal of the Atmospheric Sciences* 39.1 (jan. de 1982), pp. 3–13. ISSN: 1520-0469. DOI: [10.1175/1520-0469\(1982\)039<0003:imsttl>2.0.co;2](https://doi.org/10.1175/1520-0469(1982)039<0003:imsttl>2.0.co;2) (citado nas pgs. 6, 37).
- [KLOEDEN e PLATEN 1992] Peter E. KLOEDEN e Eckhard PLATEN. *Numerical Solution of Stochastic Differential Equations*. Springer Berlin Heidelberg, 1992. ISBN: 9783662126165. DOI: [10.1007/978-3-662-12616-5](https://doi.org/10.1007/978-3-662-12616-5) (citado na pg. 59).
- [KURAMOTO 1984] Yoshiki KURAMOTO. *Chemical Oscillations, Waves, and Turbulence*. Springer Berlin Heidelberg, 1984. ISBN: 9783642696893. DOI: [10.1007/978-3-642-69689-3](https://doi.org/10.1007/978-3-642-69689-3). URL: <http://dx.doi.org/10.1007/978-3-642-69689-3> (citado na pg. 30).
- [KUZNETSOV 2023] Yuri A. KUZNETSOV. *Elements of Applied Bifurcation Theory*. Springer International Publishing, 2023. ISBN: 9783031220074. DOI: [10.1007/978-3-031-22007-4](https://doi.org/10.1007/978-3-031-22007-4). URL: <http://dx.doi.org/10.1007/978-3-031-22007-4> (citado nas pgs. 29, 30).
- [LEITH 1980] C. E. LEITH. “Nonlinear normal mode initialization and quasi-geostrophic theory”. *Journal of the Atmospheric Sciences* 37.5 (mai. de 1980), pp. 958–968. ISSN: 1520-0469. DOI: [10.1175/1520-0469\(1980\)037<0958:nnmiaq>2.0.co;2](https://doi.org/10.1175/1520-0469(1980)037<0958:nnmiaq>2.0.co;2) (citado na pg. 29).
- [LORENZ 1963] Edward N. LORENZ. “Deterministic nonperiodic flow”. *Journal of the Atmospheric Sciences* 20.2 (1963), pp. 130–141. DOI: [10.1175/1520-0469\(1963\)020<0130:DNF>2.0.CO;2](https://doi.org/10.1175/1520-0469(1963)020<0130:DNF>2.0.CO;2) (citado na pg. 5).
- [LORENZ 1980] Edward N. LORENZ. “Attractor sets and quasi-geostrophic equilibrium”. *Journal of the Atmospheric Sciences* 37.8 (ago. de 1980), pp. 1685–1699. ISSN: 1520-0469. DOI: [10.1175/1520-0469\(1980\)037<1685:asaqge>2.0.co;2](https://doi.org/10.1175/1520-0469(1980)037<1685:asaqge>2.0.co;2) (citado nas pgs. 1, 5, 9, 12, 29, 34, 37).
- [LORENZ 1992] Edward N. LORENZ. “The slow manifold—what is it?” *Journal of the Atmospheric Sciences* 49.24 (dez. de 1992), pp. 2449–2451. ISSN: 1520-0469. DOI: [10.1175/1520-0469\(1992\)049<2449:tsmii>2.0.co;2](https://doi.org/10.1175/1520-0469(1992)049<2449:tsmii>2.0.co;2) (citado na pg. 29).

REFERÊNCIAS

- [PAVLIOTIS 2014] Grigoris A. PAVLIOTIS. *Stochastic Processes and Applications: Diffusion Processes, the Fokker-Planck and Langevin Equations*. Springer New York, 2014. ISBN: 9781493913237. DOI: [10.1007/978-1-4939-1323-7](https://doi.org/10.1007/978-1-4939-1323-7) (citado nas pgs. 20, 21, 37).
- [PAVLIOTIS e STUART 2008] Grigoris A. PAVLIOTIS e Andrew STUART. *Multiscale Methods: Averaging and Homogenization*. Springer New York, 2008. ISBN: 9780387738291. DOI: [10 . 1007 / 978 - 0 - 387 - 73829 - 1](https://doi.org/10.1007/978-0-387-73829-1) (citado nas pgs. 1, 25, 26).
- [RACKAUCKAS e NIE 2017] Christopher RACKAUCKAS e Qing NIE. “DifferentialEquations.jl – a performant and feature-rich ecosystem for solving differential equations in Julia”. *Journal of Open Research Software* 5.1 (2017) (citado nas pgs. 23, 59, 60).
- [ROMA *et al.* 2023] A. M. ROMA, J. S. BEVILACQUA e R. L. Nós. “Métodos para a solução numérica de equações diferenciais ordinárias a valores iniciais”. Notas de aula, curso de Métodos Numéricos, USP. 2023 (citado nas pgs. 57, 58).
- [SHAMPINE 1986] L. W. SHAMPINE. “Some practical runge–kutta formulas”. *Mathematics of Computation* 46.173 (1986), pp. 135–150. DOI: <https://doi.org/10.1090/S0025-5718-1986-0815836-3> (citado na pg. 58).
- [TANTET *et al.* 2020] Alexis TANTET, Mickaël D. CHEKROUN, Henk A. DIJKSTRA e J. David NEELIN. *Ruelle-Pollicott Resonances of Stochastic Systems in Reduced State Space. Part II: Stochastic Hopf Bifurcation*. 2020. DOI: <https://doi.org/10.48550/arXiv.1705.07573>. arXiv: [1705.07573](https://arxiv.org/abs/1705.07573) (citado na pg. 32).
- [VALLIS 2017] Geoffrey K. VALLIS. *Atmospheric and Oceanic Fluid Dynamics: Fundamentals and Large-Scale Circulation*. Cambridge University Press, jun. de 2017. ISBN: 9781107588417. DOI: [10.1017/9781107588417](https://doi.org/10.1017/9781107588417) (citado nas pgs. 3–5).
- [YOUNG 2002] Lai-Sang YOUNG. “What are srb measures, and which dynamical systems have them?” *Journal of Statistical Physics* 108.5–6 (set. de 2002), pp. 733–754. ISSN: 1572-9613. DOI: [10.1023/a:1019762724717](https://doi.org/10.1023/a:1019762724717) (citado na pg. 26).