

Pablo Aguiar De Maio

Um método de linearização local com  
passo adaptativo para solução numérica  
de equações diferenciais estocásticas com  
ruído aditivo

Rio de Janeiro  
2015

Pablo Aguiar De Maio

Um método de linearização local com  
passo adaptativo para solução numérica  
de equações diferenciais estocásticas com  
ruído aditivo

Dissertação apresentada à Escola de Matemática Aplicada da Fundação Getúlio Vargas, para a obtenção de Título de Mestre em Ciências, na Área de Modelagem Matemática da Informação.

Orientador: Hugo Alexander de la Cruz Cancino

**Rio de Janeiro  
2015**

De Maio, Pablo Aguiar

Um método de linearização local com passo adaptativo para solução numérica de equações diferenciais estocásticas com ruído aditivo / Pablo Aguiar De Maio. – 2015.

73 f.

Dissertação (mestrado) – Fundação Getulio Vargas, Escola de Matemática Aplicada.

Orientador: Hugo Alexander de la Cruz Cancino.

Inclui bibliografia.

1. Equações diferenciais estocásticas. 2. Simulação (Computadores). 3. Análise numérica. I. Cruz Cancino, Hugo Alexander de la. II. Fundação Getulio Vargas. Escola de Matemática Aplicada. III Título.

CDD – 519.2

**PABLO AGUIAR DE MAIO**

**UM MÉTODO DE LINEARIZAÇÃO LOCAL COM PASSO ADAPTATIVO PARA  
SOLUÇÃO NUMÉRICA DE EQUAÇÕES DIFERENCIAIS ESTOCÁSTICOS COM  
RUIDO ADITIVO.**

Dissertação apresentada ao Curso de Mestrado em Modelagem Matemática da Informação da Escola de Matemática Aplicada da Fundação Getúlio Vargas para obtenção do grau de Mestre em Modelagem Matemática da Informação.

Data da defesa: 31/07/2015.

**ASSINATURA DOS MEMBROS DA BANCA EXAMINADORA**



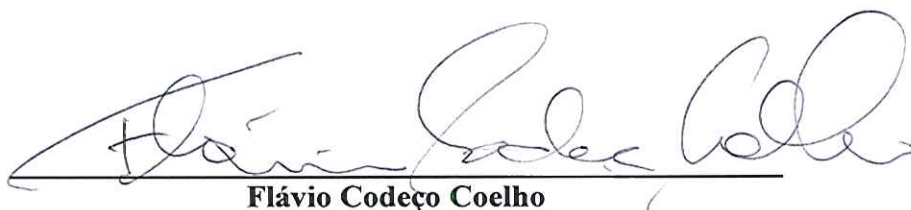
**Hugo Alexander de la Cruz Cancino**  
Orientador (a)



**Max Souza**



**Moacyr Alvim Horta Barbosa da Silva**



**Flávio Codeço Coelho**

*Dedicatória...*

"O maior bem do Homem é uma mente inquieta."

Isaac Asimov

## Agradecimentos

Agradeço ao meu orientador, Hugo, que não só me apresentou o tema como também me guiou por todo o caminho, aos meus amigos Ronald Simões e Fernando Menucci por terem tido paciência de ler este texto, e à minha noiva, Ana Paula, que foi mais do que compreensiva com os dias que deixamos de estar juntos.

### *Resumo*

Neste trabalho apresentamos um novo método numérico com passo adaptativo baseado na abordagem de linearização local, para a integração de equações diferenciais estocásticas com ruído aditivo. Propomos, também, um esquema computacional que permite a implementação eficiente deste método, adaptando adequadamente o algoritmo de Padé com a estratégia “scaling-squaring” para o cálculo das exponenciais de matrizes envolvidas. Antes de introduzirmos a construção deste método, apresentaremos de forma breve o que são equações diferenciais estocásticas, a matemática que as fundamenta, a sua relevância para a modelagem dos mais diversos fenômenos, e a importância da utilização de métodos numéricos para avaliar tais equações. Também é feito um breve estudo sobre estabilidade numérica. Com isto, pretendemos introduzir as bases necessárias para a construção do novo método/esquema. Ao final, vários experimentos numéricos são realizados para mostrar, de forma prática, a eficácia do método proposto, e compará-lo com outros métodos usualmente utilizados.

**Palavras-chave:** Equações diferenciais estocásticas, Esquemas exponenciais de Linearização local, Passo adaptativo, A-estabilidade, Análise numérica, Simulação computacional.

### *Abstract*

In this work we present a new numerical method with adaptive stepsize based on the local linearization approach, to integrate stochastic differential equations with additive noise. We also propose a computational scheme that allows efficient implementation of this method, properly adapting the algorithm of Padé with scaling-squaring strategy to compute the exponential of matrices involved. To introduce the construction of this method, we briefly explain what stochastic differential equations are, the mathematics that is behind them, their relevance to the modeling of various phenomena, and the importance of using numerical methods to evaluate this kind of equations. A succinct study of numerical stability is also presented on the following pages. With this dissertation, we intend to introduce the necessary basis for the construction of the new method/scheme. At the end, several numerical experiments are performed to demonstrate, in a practical way, the effectiveness of the proposed method, comparing it with other methods commonly used.

**Keywords:** Stochastic Differential Equations, Local Linearization Exponential schemes, Adaptive stepsize, A-stability, Numerical analysis, Computer simulation.

# Lista de Figuras

1.1	Exemplo de trajetória Browniana no intervalo $[0, 100]$ , com 300 amostras. . . . .	5
3.1	Realização da solução dos exemplos 3.0.1 e 3.0.2 sobre uma trajetória browniana gerada aleatoriamente. . . . .	31
3.2	Simulação do exemplo 3.0.1 utilizando o método de Milstein com 100 passos. . .	31
3.3	Simulação do exemplo 3.0.2 utilizando o método de Milstein com 100 passos. . .	31
3.4	Simulação do exemplo 3.0.2 utilizando o método de Milstein com 4995 passos. .	32
3.5	Simulação do exemplo 3.0.2 utilizando o método de Milstein com 5000 passos. .	32
3.6	Simulação do exemplo 3.0.2 utilizando o método LL de ordem 1 com 100 passos.	35
4.1	Exemplo motivacional . . . . .	38
5.1	Gráfico da solução “exata” sobre a mesma trajetória Browniana das simulações apresentadas nas figuras 5.2 e 5.3. . . . .	56
5.2	Simulação do exemplo 5.0.1, utilizando o LL 115 (adaptativo) com 27 passos. . .	56
5.3	Simulação do exemplo 5.0.1, utilizando o LL 1.5 (passo fixo) com 27 passos. . . .	56
5.4	Simulação do exemplo 5.0.1, utilizando o método de Milstein sobre a trajetória adaptativa da execução da figura 5.2. . . . .	57
5.5	Simulação do exemplo 5.0.1, utilizando o método de Milstein com tamanho de passo $h_{min} = 0,0382$ . . . . .	57
5.6	Simulação do exemplo 5.0.1, utilizando o método de Milstein com $h = 0,04$ . . .	58

5.7	Simulação do exemplo 5.0.1, utilizando o método de Milstein com $h = 0,05$ .	58
5.8	Simulação do exemplo 5.0.2, utilizando o método adaptativo com $tol = (0,001; 0,001)$ .	59
5.9	Tamanho de passo $h$ executado em cada tempo $t$ .	59
5.10	Simulação do exemplo 5.0.3, utilizando o método adaptativo com $tol = (0,5; 0,5; 0,5; 0,5)$ .	60
5.11	Tamanho de passo $h$ executado em cada tempo $t$ .	60
5.12	Simulação do exemplo 5.0.3, utilizando o método adaptativo com $tol = (0,1; 0,1; 0,1; 0,1)$ .	61
5.13	Tamanho de passo $h$ executado em cada tempo $t$ .	61
5.14	Simulação do exemplo 5.0.3, utilizando o método adaptativo com $tol = (0,01; 0,01; 0,01; 0,01)$ .	61
5.15	Tamanho de passo $h$ executado em cada tempo $t$ .	61
5.16	Simulação do exemplo 5.0.3, utilizando o método adaptativo com $tol = (0,001; 0,001; 0,001; 0,001)$ .	61
5.17	Tamanho de passo $h$ executado em cada tempo $t$ .	61

# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Objetivos . . . . .	2
1.2	O que são equações diferenciais estocásticas? . . . . .	4
1.3	O Processo de Wiener . . . . .	5
1.3.1	Processo de Wiener m-dimensional . . . . .	6
1.4	A integral de Itô . . . . .	6
1.5	A regra da cadeia de Itô . . . . .	9
1.6	Existência e unicidade de soluções . . . . .	10
1.7	Equações diferenciais estocásticas com ruído aditivo . . . . .	11
1.8	Método de variação de parâmetros . . . . .	12
<b>2</b>	<b>Métodos numéricos</b>	<b>13</b>
2.1	Convergência forte e fraca . . . . .	14
2.2	Taylor estocástico . . . . .	15
2.3	Método de Euler Maruyama . . . . .	17
2.4	Método de Milstein . . . . .	18
2.5	Métodos de Linearização Local . . . . .	21
2.5.1	Esquemas exponenciais para LL 1 e LL 1.5 . . . . .	24
<b>3</b>	<b>Estabilidade</b>	<b>29</b>

<b>4</b>	<b>Algoritmo adaptativo</b>	<b>37</b>
4.1	Motivação intuitiva . . . . .	37
4.2	Introdução . . . . .	38
4.3	Estimando o erro local e um próximo passo ideal . . . . .	39
4.4	Mantendo a trajetória Browniana coerente . . . . .	40
4.5	Algoritmo para um passo de LL 115 . . . . .	45
4.6	Uma implementação ótima para os métodos LL 1 e 1.5 . . . . .	46
4.6.1	O algoritmo de Padé para calcular a exponencial de uma matriz . . . . .	49
4.6.2	O algoritmo de Padé adaptado para computar $e^{Mh}$ . . . . .	50
<b>5</b>	<b>Simulações</b>	<b>55</b>
<b>6</b>	<b>Considerações Finais</b>	<b>65</b>
6.1	Trabalhos futuros . . . . .	66
	<b>Referências Bibliográficas</b>	<b>68</b>



# Capítulo 1

## Introdução

Equações diferenciais estocásticas, ou simplesmente EDEs, surgem da modelagem de vários problemas oriundos das mais diversas áreas da ciência, e nos permitem estudar fenômenos como ruídos na transmissão de sinais de telecomunicações, a trajetória de um satélite, um circuito elétrico que sofra efeitos de ruídos térmicos, a dinâmica de uma reação química, a dispersão de poluentes no ar ou na água, a dinâmica de populações; enfim, utilizamos EDEs para modelar sistemas que sofrem a interferência de ruídos aleatórios.

Podemos ainda dizer que equações diferenciais ordinárias (EDOs) são utilizadas para modelar situações ideais, onde o que afeta o sistema de forma significativa pode ser mensurado e explicado, mas quando queremos um modelo mais próximo da realidade, considerando suas imperfeições, as EDOs não são suficientes, e disto surge a necessidade da adição de um termo estocástico, um “*ruído*”, resultando no nosso objeto de estudo, que são as equações diferenciais estocásticas. A adição deste termo estocástico nos permite modelos mais próximos da realidade, porém, isto nos traz diversos outros problemas, pois ao sair da seara das equações determinísticas e adentrarmos os processos estocásticos, o cálculo diferencial tradicional que conhecemos não se aplica mais, e precisamos de uma

nova base matemática para podermos estudar suas soluções<sup>1</sup>, e mesmo assim, muitas destas equações não possuem solução com forma fechada, o que nos leva diretamente ao estudo de métodos numéricos para determinar soluções aproximadas de EDEs. Assim, iremos neste trabalho estudar métodos numéricos para EDEs, e diversos fatores relevantes ao desenvolvimento e à construção de tais métodos.

## 1.1 Objetivos

Este trabalho visa contribuir com o estudo de equações diferenciais estocásticas, em específico com o estudo de métodos numéricos para a aproximação forte de soluções de EDEs com ruído aditivo. Diversos métodos numéricos para EDEs têm sido utilizados nas últimas décadas, como os métodos strong Taylor (onde os mais conhecidos são o método Euler Maruyama e o método de Milstein), bem como os métodos de Runge-Kutta, Predictor-Corretor, dentre outras famílias de métodos que podem ser encontradas em praticamente qualquer livro sobre o tema, e.g. [7], [8] e [9]. E mais recentemente foi desenvolvida uma classe importante de métodos numéricos que utilizam a abordagem de linearização local (LL) para equações com ruído aditivo, onde para cada passo do método, a EDE é localmente aproximada através de uma EDE linear. Tais métodos LL vêm sendo estudados e há diversas publicações sobre o tema, como [10], [11], [12], [13], [14], [15] e [16] dentre outras, onde podemos ver que estes métodos numéricos, além de A-estáveis, preservam propriedades dinâmicas das soluções, e ainda necessitam de um esforço computacional muito menor que outros métodos com propriedades semelhantes.

Entretanto, nestes trabalhos só foram estudados métodos numéricos com tamanho de passo fixo, ou seja, todos os passos executados pelo método possuem o mesmo tamanho e não há um controle do erro de aproximação. De fato, nenhum método LL para EDEs com ruído aditivo até então publicado utiliza tamanho de passo adaptativo, isto é, quando

---

<sup>1</sup>Estas bases foram dadas pelo matemático Kiyoshi Itô em [6].

o tamanho de cada passo é definido de acordo com as características da equação que está sendo integrada, assim estes métodos são capazes de identificar os intervalos onde a solução é mais propensa a erros de aproximação numérica, e assim reduzir o tamanho de passo nestes intervalos afim de controlar este erro, e por sua vez efetuar passos maiores nos intervalos menos críticos da solução. Então, o que propomos com o presente trabalho é preencher esta lacuna, aplicando a metodologia de passo adaptativo a métodos exponenciais de linearização local, e construir esquemas computacionais eficientes para estes métodos.

*Observação.* Embora tenha sido desenvolvido recentemente um método LL com passo adaptativo para EDOs, vide [17], não existem referências na literatura de métodos exponenciais com passo adaptativo para EDEs. E disto vem a originalidade do método proposto neste trabalho.

Admitindo que o leitor possua conhecimentos sobre equações diferenciais ordinárias<sup>2</sup> e teoria da probabilidade<sup>3</sup>, vamos ainda neste capítulo, introduzir os conceitos necessários para a compreensão das equações diferenciais estocásticas e o ferramental necessário para o desenvolvimento dos capítulos seguintes. No capítulo 2, estabelecemos as bases necessárias à construção de métodos numéricos para solução forte de EDEs, apresentamos os métodos mais comuns e os métodos LL que iremos utilizar para a construção do nosso método adaptativo. Estudaremos também a estabilidade numérica dos métodos LL no capítulo 3. No capítulo 4, se dá a construção do algoritmo adaptativo, e introduzimos uma forma diferente de calcular as exponenciais de matrizes envolvidas no esquema numérico proposto, de forma que construímos uma implementação aninhada dos métodos LL envolvidos, reduzindo consideravelmente o custo computacional do método, e finalmente, no capítulo 5, apresentamos simulações de diferentes tipos de EDEs utilizando o método/esquema com passo adaptativo proposto nesta dissertação, e os métodos mais

---

<sup>2</sup>Uma boa referência para teoria de EDOs é [18], e para métodos para aproximação numérica de EDOs é [3].

<sup>3</sup>Duas boas referências para teoria da probabilidade são [4] e [5].

comuns.

## 1.2 O que são equações diferenciais estocásticas?

Inicialmente, vamos pensar em uma equação diferencial ordinária da forma

$$dX(t) = f(t, X(t))dt, \quad (1.1)$$

e vamos adicionar um “*ruído*”, assim a versão estocástica de (1.1) pode ser dada por

$$dX(t) = f(t, X(t))dt + \text{“ruído”}, \quad (1.2)$$

onde este “*ruído*” se apresenta na forma de um valor aleatório, cuja distribuição de probabilidade é conhecida. Desta forma, uma EDE possui um termo determinístico (como em uma equação diferencial ordinária) e um termo estocástico que representa o “*ruído*”, resultando assim que as suas soluções sejam processos estocásticos.

No âmbito deste trabalho, iremos tratar de EDEs nas quais as perturbações aleatórias podem ser descritas por um *ruído branco* contínuo no tempo, cuja integral é um processo de Wiener. Assim uma equação diferencial estocástica se apresenta da forma

$$dX(t) = f(t, X(t))dt + g(t, X(t))dW(t), \quad (1.3)$$

$$X(t_0) = X_0.$$

Onde a função  $f$  é chamada de coeficiente de deslocamento (ou “drift” em inglês, como é mais conhecida), a função  $g$  é chamada de coeficiente de difusão,  $\{W(t), t \geq 0\}$  é um processo de Wiener, o qual iremos definir na seção abaixo, e  $X_0$  é o valor inicial do problema.

## 1.3 O Processo de Wiener

O processo de Wiener, também conhecido como Movimento Browniano, tem origem no estudo do movimento irregular de grãos de pólen suspensos em um líquido. De acordo com [2], para descrever matematicamente o movimento destas partículas é usado o conceito de um processo estocástico, que dá a possível posição de uma partícula em um certo tempo  $t$ .

**Definição 1.3.1** (*Processo de Wiener*). Dado um espaço de probabilidade  $(\Omega, \mathcal{F}, \mathcal{P})$ , um processo estocástico  $\{W_t\}_{t \in [0, +\infty)}$  definido nesse espaço diz-se um *processo de Wiener* (ou movimento Browniano) se satisfizer as três propriedades a seguir:

- (i)  $W(0) = 0$ ,
- (ii)  $W(t) - W(s)$  tem distribuição normal com média 0 e variância  $|t - s|$ ,
- (iii) para todo  $t$ ,  $0 < t_1 < t_2 < \dots < t_n$ , as variáveis aleatórias

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

são independentes, ou seja, os incrementos são independentes.

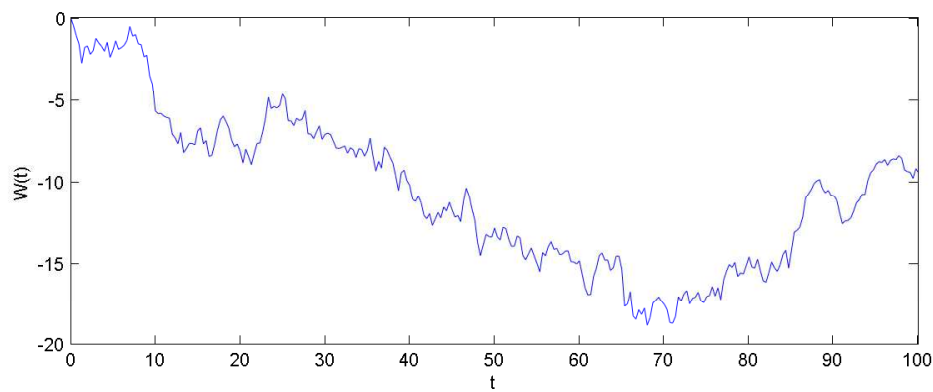


Figura 1.1: Exemplo de trajetória Browniana no intervalo  $[0, 100]$ , com 300 amostras.

*Observação.* Mais sobre espaços de probabilidade pode ser encontrado em [4].

### 1.3.1 Processo de Wiener m-dimensional

Um processo de Wiener m-dimensional é um vetor em que cada posição é um processo de Wiener como descrito na definição 1.3.1, assim definimos abaixo.

**Definição 1.3.2** (*Processo de Wiener m-dimensional*). É chamado de um *processo de Wiener m-dimensional* o processo  $\{W_t\}_{t \in [0, +\infty)} = \{W_t^{(1)}, \dots, W_t^{(m)},\}_{t \in [0, +\infty)}$  quando

- (i) para cada  $k = 1, \dots, m$ , a componente  $\{W_t^{(k)}\}_{t \in [0, +\infty)}$  é um processo de Wiener unidimensional, como na definição 1.3.1 e
- (ii) os processos  $\{W_t^{(k)}\}_{t \in [0, +\infty)}$ , para  $k = 1, \dots, m$  são independentes entre si.

Mais informações sobre o *processo de Wiener* podem ser encontradas em [1], [2], [7], [9], e [20].

## 1.4 A integral de Itô

O que nos permite estudar soluções de EDOs é o fato de termos a integral bem definida no caso determinístico, porém no caso das EDEs precisamos fazer uso de integrais estocásticas e apesar de existirem diversas definições possíveis para estas, a definição de Itô é a que possui as melhores propriedades probabilísticas, fazendo com que seja a mais utilizada. E portanto é a nossa escolha para este trabalho.

Considerando o problema de valor inicial 1.3, podemos escrever sua equação integral equivalente como

$$X(t) = X_0 + \underbrace{\int_0^t f(s, X(s)) ds}_{\text{Integral (1)}} + \underbrace{\int_0^t g(s, X(s)) dW(s)}_{\text{Integral (2)}} \quad (1.4)$$

Assim a *Integral (1)* é uma integral determinística, e a *Integral (2)* é uma integral estocástica, a qual é o tema desta secção.

Analogamente à aproximação de uma integral por uma soma de Riemann finita, vamos considerar uma soma de  $N$  subintervalos  $[t_{i-1}, t_i]$  da forma

$$S_N = \sum_{i=1}^N g(\tau_i, X(\tau_i))(W(t_i) - W(t_{i-1})) \quad (1.5)$$

onde  $\tau_i \in [t_{i-1}, t_i]$ . Esta soma converge na média quadrática para valores diferentes da integral, ou seja, para cada  $\tau_i$  existe um  $I_{\tau_i}$  tal que o valor esperado  $\mathbb{E}[(S_N - I_{\tau_i})^2] = 0$ , quando  $N \rightarrow \infty$ . Essa demonstração pode ser vista em [20], [9], [2], [1].

Se tomarmos  $\tau_i = t_{i-1}$ , ou seja, o extremo esquerdo de cada intervalo, temos a *integral de Itô*, definida como

$$I_{Itô} = \int_{t_0}^t g(s, X(s))dW(s) = \lim_{N \rightarrow \infty} \sum_{i=1}^N g(t_{i-1}, X(t_{i-1}))(W(t_i) - W(t_{i-1})) \quad (1.6)$$

onde  $g(t)$  é um processo estocástico definido no mesmo espaço de probabilidade  $(\Omega, \mathcal{F}, \mathcal{P})$  que o processo de Wiener  $W(t)$ , e  $g$  é *não antecipativo*, ou seja, a informação sobre  $g$  em um certo tempo  $t$  não depende de eventos posteriores ao tempo  $t$ . Desta forma, a integral de Itô preserva algumas propriedades do processo de Wiener, em específico, o fato de  $W(t)$  ser um martingale implica que a integral de Itô forme um martingale, o custo disto para o desenvolvimento da teoria é que nos leva ao estudo de um novo cálculo diferencial e integral, o chamado cálculo de Itô.

**Teorema 1.4.1** (Propriedades da Integral de Itô). *Sejam  $g_1, g_2$  duas funções Itô-integráveis, uma constante  $c$  e  $t_0$  e  $T$ , tais que  $0 \leq t_0 < T$ . Então*

- (i)  $\int_{t_0}^T (cg_1 + g_2)dW(t) = c \int_{t_0}^T g_1 dW(t) + \int_{t_0}^T g_2 dW(t)$
- (ii)  $\mathbb{E} \left[ \int_{t_0}^T g_1 dW(t) \right] = 0$
- (iii)  $\mathbb{E} \left[ \left( \int_{t_0}^T g_1 dW(t) \right)^2 \right] = \mathbb{E} \left[ \int_{t_0}^T g_1^2 ds \right]$  *Conhecido como Isometria de Itô.*

(iv)  $\int_{t_0}^T g_1 dW(t)$  é não antecipativa.

*Demonstração.*

(i)

$$\begin{aligned}
 \int_{t_0}^T (cg_1 + g_2) dW(t) &= \lim_{N \rightarrow \infty} \sum_{i=1}^N [cg_1(t_{i-1}, X(t_{i-1})) + \\
 &\quad + g_2(t_{i-1}, X(t_{i-1}))](W(t_i) - W(t_{i-1})) \\
 &= c \lim_{N \rightarrow \infty} \sum_{i=1}^N g_1(t_{i-1}, X(t_{i-1}))(W(t_i) - W(t_{i-1})) + \\
 &\quad + \lim_{N \rightarrow \infty} \sum_{i=1}^N g_2(t_{i-1}, X(t_{i-1}))(W(t_i) - W(t_{i-1})) \\
 &= c \int_{t_0}^T g_1 dW(t) + \int_{t_0}^T g_2 dW(t)
 \end{aligned}$$

(ii)

$$\begin{aligned}
 \mathbb{E} \left[ \int_{t_0}^T g_1 dW(t) \right] &= \mathbb{E} \left[ \lim_{N \rightarrow \infty} \sum_{i=1}^N g_1(t_{i-1}, X(t_{i-1}))(W(t_i) - W(t_{i-1})) \right] \\
 &\quad \text{(como } g_1 \text{ e } (W(t_i) - W(t_{i-1}))) \text{ são independentes entre si)} \\
 &= \lim_{N \rightarrow \infty} \sum_{i=1}^N \underbrace{\mathbb{E}[g_1(t_{i-1}, X(t_{i-1}))]}_{\text{é finita}} \underbrace{\mathbb{E}[(W(t_i) - W(t_{i-1}))]}_{=0} \\
 &= 0
 \end{aligned}$$

(iii)

$$\begin{aligned}
 \mathbb{E} \left[ \left( \int_0^T g_1 dW(t) \right)^2 \right] &= \lim_{N \rightarrow \infty} \sum_{i=1}^N \mathbb{E} \left[ g_{1_i}^2 (W(t_i) - W(t_{i-1}))^2 \right] \\
 &= \lim_{N \rightarrow \infty} \sum_{i=1}^N \mathbb{E} \left[ g_{1_i}^2 \right] \underbrace{\mathbb{E} \left[ (W(t_i) - W(t_{i-1}))^2 \right]}_{\substack{= \text{Var} \left[ (W(t_i) - W(t_{i-1}))^2 \right] \\ = t_i - t_{i-1}}} \\
 &= \lim_{N \rightarrow \infty} \sum_{i=1}^N \mathbb{E} \left[ g_{1_i}^2 (t_i - t_{i-1}) \right] = \mathbb{E} \left[ \int_0^T g_1^2 dt \right]
 \end{aligned}$$

(iv) Vem direto do fato de  $g$  ser não antecipativa.

□

*Observação.* Uma outra integral estocástica importante é a chamada *integral estocástica de Stratonovich*, que ao contrário da integral de Itô, atende às regras do cálculo real, mas não tem boas propriedades probabilísticas. O que a difere da integral de Itô é que a função integranda é avaliada no ponto médio do intervalo, e não no seu extremo esquerdo. Ou seja

$$I_{Stratonovich} = \lim_{N \rightarrow \infty} \sum_{i=1}^N g\left(\frac{t_i - t_{i-1}}{2}, X\left(\frac{t_i - t_{i-1}}{2}\right)\right) (W(t_i) - W(t_{i-1})). \quad (1.7)$$

Note que uma integral de Itô pode ser representada como uma integral de Stratonovich e vice-versa, como pode ser visto em [2]

## 1.5 A regra da cadeia de Itô

Um dos inconvenientes do uso da integral de Itô é que a regra da cadeia do cálculo tradicional não mais tem validade, e então há a necessidade de construirmos uma nova regra da cadeia, ou como iremos chamar, a *Fórmula de Itô*.

**Teorema 1.5.1** (Fórmula de Itô). *Seja  $X(t)$  tal que  $dX(t) = f(t, X(t))dt + g(t, X(t))dW(t)$ , e seja  $Y(t) = u(t, X(t))$ , onde  $u(t, x)$  é uma função com derivadas parciais contínuas  $\frac{\partial u}{\partial t}$ ,  $\frac{\partial u}{\partial x}$ , e  $\frac{\partial^2 u}{\partial x^2}$ . Assim a fórmula de Itô é dada por*

$$\begin{aligned} dY(t) &= \frac{\partial u(t, X(t))}{\partial t} dt + f(t, X(t)) \frac{\partial u(t, X(t))}{\partial x} dt + \\ &+ \frac{1}{2} g^2(t, X(t)) \frac{\partial^2 u(t, X(t))}{\partial x^2} dt + g(t, X(t)) \frac{\partial u(t, X(t))}{\partial x} dW(t) \end{aligned} \quad (1.8)$$

E na forma integral temos

$$\begin{aligned}
Y(t) = & Y(0) + \int_0^t \left( \frac{\partial u(s, X(s))}{\partial s} + f(s, X(s)) \frac{\partial u(s, X(s))}{\partial x} + \right. \\
& \left. + \frac{1}{2} g^2(s, X(s)) \frac{\partial^2 u(s, X(s))}{\partial x^2} \right) ds + \\
& + \int_0^t g(s, X(s)) \frac{\partial u(s, X(s))}{\partial x} dW(s)
\end{aligned} \tag{1.9}$$

A demonstração deste teorema pode ser encontrado em [8] e [2].

## 1.6 Existência e unicidade de soluções

**Teorema 1.6.1** (Existência e unicidade de soluções de equações diferenciais estocásticas).

Sejam  $f(\cdot, \cdot) : [0, T] \times \mathbb{R}^d \rightarrow \mathbb{R}^d$ ,  $g(\cdot, \cdot) : [0, T] \times \mathbb{R}^d \rightarrow \mathbb{R}^{d \times m}$  duas funções mensuráveis,  $T > 0$  e uma constante  $C$ , tais que para todo  $t \in [0, T]$  e  $x, y \in \mathbb{R}^d$  satisfazem

(i)

$$|f(t, x)|^2 + |g(t, x)|^2 \leq C^2(1 + |x|)^2 \tag{1.10}$$

onde que  $|g|^2 = \sum |g_{ij}|^2$ ;

(ii)

$$|f(t, x) - f(t, y)| + |g(t, x) - g(t, y)| \leq C|x - y| \tag{1.11}$$

Além disso  $X_0$  é independente de  $W(t)$  para  $t > 0$ , e  $\mathbb{E}[|X_0|^2] < \infty$ .

Então existe uma única solução  $X(t)$  da equação (1.3) no intervalo  $[0, T]$  que é contínua em  $t$ , ou seja, dadas duas soluções de (1.3) em  $[0, T]$ , a probabilidade que elas sejam iguais entre si é 1.

A demonstração deste teorema pode ser encontrada em diversos livros como [2], [19] e [20]. É uma construção muito interessante da solução de uma EDE unidimensional

pode ser vista em [1].

## 1.7 Equações diferenciais estocásticas com ruído aditivo

Equações diferenciais estocásticas com ruído aditivo são utilizadas em modelos matemáticos para descrever processos aleatórios em diversas áreas, como em modelos de energia celular, sistemas de coagulação de sangue, atividade elétrica de massas neurais dentre muitos outros.

Uma EDE é dita *com ruído aditivo* quando o coeficiente de difusão  $g$  não depende da variável  $X$ , ou seja  $g(t, X(t)) = g(t)$ , sendo assim é dada pela equação

$$dX(t) = f(t, X(t))dt + g(t)dW(t). \quad (1.12)$$

Agora suponha que estejamos trabalhando com um modelo que sofre mais de uma perturbação aleatória, onde estas perturbações sejam independentes, desta forma teríamos uma EDE com mais de uma função de difusão. Assim, dado um processo de Wiener  $m$ -dimensional  $\{W_t\}_{t \in [0, +\infty)} = \{W_t^{(1)}, \dots, W_t^{(m)}\}_{t \in [0, +\infty)}$  e  $m$  funções de difusão  $g_i(t)$ , temos que

$$dX(t) = f(t, X(t))dt + \sum_{i=1}^m g_i(t)dW^i(t) \quad (1.13)$$

Outra notação para a equação (1.13) é dada por

$$dX(t) = f(t, X(t))dt + G(t)dW(t) \quad (1.14)$$

onde  $G : \mathbb{R} \rightarrow \mathbb{R}^{d \times m}$ , é uma matriz onde cada coluna é um vetor  $g_i(t)$  de dimensão  $d$ .

Usaremos uma das definições (1.12), (1.13) ou (1.14) quando esta for mais conveniente.

Mais a frente apresentaremos um método/esquema para a solução deste tipo de EDEs, o qual é o tema principal deste trabalho.

## 1.8 Método de variação de parâmetros

O lema abaixo descreve a versão estocástica do método de variação de parâmetros para equações com ruído aditivo, que usaremos em alguns momentos quando houver a necessidade de encontrar soluções de alguns tipos de EDEs.

**Lema 1.8.1** (Método de variação de parâmetros para equações diferenciais estocásticas com ruído aditivo). *Seja a EDE (1.13), com condição inicial  $X(t_0) = X_0$ ,  $X_0 \in \mathbb{R}^d$ . Suponhamos que sejam satisfeitas as condições de existência e unicidade de uma solução forte  $X(t)$ . Então, consideremos que  $f(t, X) = A(t)X + a(t, X)$ , e  $g^j(t, X) = B^j(t)X + b^j(t, X)$ , com  $A, B^j \in C(\mathbb{R}_+, \mathbb{R}^{d \times d})$ . Seja  $\phi(t, t_0)$  a matriz fundamental que satisfaz*

$$\begin{aligned} d\phi(t, t_0) &= A(t)\phi(t, t_0)dt + \sum_{j=1}^m B^j(t)\phi(t, t_0)dW_t^j \\ \phi(t_0, t_0) &= I_{d \times d}, \end{aligned}$$

então

$$\begin{aligned} X(t) &= \phi(t, t_0) \left[ X_0 + \int_{t_0}^t \phi^{-1}(s, t_0) \left( a(s, X(s, t_0, X_0)) - \sum_{j=1}^m B^j(s)b^j(s, X(s, t_0, X_0)) \right) ds + \right. \\ &\quad \left. + \sum_{j=1}^m \int_{t_0}^t \phi^{-1}(s, t_0)b^j(s, X(s, t_0, X_0))dW_s^j \right] \end{aligned} \quad (1.15)$$

Uma demonstração para este lema pode ser encontrada em [22].

## Capítulo 2

# Métodos numéricos

Raramente, equações diferenciais estocásticas utilizadas para modelar problemas do mundo real possuem solução explícita, e esta é a principal motivação para o estudo de métodos numéricos para determinar soluções de EDEs. Há também a recorrente necessidade de analisar não só a solução de uma EDE em um certo instante do tempo, mas também a sua trajetória.

Neste trabalho, vamos tratar de aproximações numéricas de EDEs que se baseiam em uma discretização de um intervalo de tempo  $[t_0, T]$ , como

$$t_0 < t_1 < t_2 < \cdots < t_{N-1} < t_N = T$$

onde se quer avaliar o comportamento de uma EDE. Essa discretização também é conhecida como uma *partição* do intervalo  $[t_0, T]$ . No caso de métodos de passo temporal fixo, temos que a distância entre dois pontos consecutivos de uma partição é dado por  $h_i = t_i - t_{i-1} = \frac{T-t_0}{N}$  para todo  $i$  tal que  $1 \leq i \leq N \in \mathbb{N}$ . E como veremos mais a frente, em métodos com passo temporal variável (ou adaptativo), essa distância  $h_i$  entre dois pontos consecutivos da partição vai depender da estimativa do erro local obtido no passo anterior pelo método/esquema numérico.

## 2.1 Convergência forte e fraca

Sempre que trabalhamos com métodos numéricos é imprescindível ter formas de mensurar a sua eficiência, em específico, analisando a sua ordem de convergência. Assim, dependendo do tipo de aproximação utilizada, temos diversos critérios que podem ser utilizados para analisar a convergência de um método numérico.

Em específico, quando tratamos de modelos estocásticos, os dois principais tipos de convergência são a convergência forte, que tange aproximações das trajetórias de soluções de EDEs em um intervalo de tempo, e a convergência fraca, que diz respeito à aproximações de EDEs à distribuições correspondentes. Para definir convergência forte e fraca, vamos primeiro definir os padrões de erro utilizados.

**Definição 2.1.1** (Erros). Seja  $x_n$  a variável aleatória correspondente a aproximação numérica da solução de uma EDE no momento  $t_n$ , então

(i) o *erro forte* é dado por

$$erro_h^{forte} = \sup_n (\mathbb{E}[|x(t_n) - x_n|]), \quad (2.1)$$

(ii) e o *erro fraco* é dado pela diferença entre os valores esperados de  $x(t_n)$  e  $x_n$ , ou seja

$$erro_h^{fraco} = \sup_n (\mathbb{E}[p(x(t_n))] - \mathbb{E}[p(x_n)]), \quad (2.2)$$

onde  $p : \mathbb{R}^d \rightarrow \mathbb{R}$  é uma função de teste.

Podemos dizer que o *erro forte* é a média do erro, e que o *erro fraco* é o erro das médias.

**Definição 2.1.2.** (Convergência)

(i) Dizemos que um método tem uma ordem de *convergência forte*  $\gamma$ , se existe uma constante  $C$  tal que

$$erro_h^{forte} = \sup_n (\mathbb{E}[|x(t_n) - x_n|]) \leq C.h^\gamma = \mathcal{O}(h^\gamma),$$

(ii) e um método tem uma ordem de *convergência fraca*  $\beta$ , se existe uma constante  $C$  tal que

$$erro_h^{fraco} = \sup_n (\mathbb{E}[p(x(t_n))] - \mathbb{E}[p(x_n)]) \leq C.h^\beta = \mathcal{O}(h^\beta).$$

Neste trabalho, vamos nos ater ao estudo da convergência forte de EDEs. Mais sobre convergência de métodos numéricos para EDEs pode ser encontrado em [8].

## 2.2 Taylor estocástico

Aqui vamos apresentar a versão estocástica da famosa expansão de Taylor. Esta serve como base para a construção de diversos métodos para solução forte de EDEs, e é conhecida como *expansão de Itô-Taylor*.

Assim dada a equação integral

$$X(t) = X_0 + \int_0^t f(s, X(s))ds + \int_0^t g(s, X(s))dW(s)$$

e considerando os operadores

$$L^0 = \frac{\partial}{\partial t} + \sum_{k=1}^d f_k \frac{\partial}{\partial x_k} + \frac{1}{2} \sum_{k,l=1}^d \sum_{j=1}^m g_{k,j} g_{l,j} \frac{\partial^2}{\partial x_k \partial x_l} \quad (2.3)$$

$$L^j = \sum_{k=1}^d g_{k,j} \frac{\partial}{\partial x_k} \quad (2.4)$$

que no caso escalar ficam da forma

$$\begin{aligned} L^0 &= \frac{\partial}{\partial t} + f \frac{\partial}{\partial x} + \frac{1}{2} g^2 \frac{\partial^2}{\partial x^2} \\ L^1 &= g \frac{\partial}{\partial x} \end{aligned}$$

e aplicando a *fórmula de Itô* apresentada no teorema 1.5, temos que

$$\begin{aligned} X(t) = X(t_0) &+ \int_{t_0}^t \left[ f(t_0, X(t_0)) + \int_{t_0}^s L^0 f(u, X(u)) du + \right. \\ &\quad \left. + \int_{t_0}^s L^1 f(u, X(u)) dW(u) \right] ds + \\ &+ \int_{t_0}^t \left[ g(t_0, X(t_0)) + \int_{t_0}^s L^0 g(u, X(u)) du + \right. \\ &\quad \left. + \int_{t_0}^s L^1 g(u, X(u)) dW(u) \right] dW(s) \end{aligned} \quad (2.5)$$

reorganizando a equação (2.5) ficamos com

$$X(t) = X(t_0) + f(t_0, X(t_0)) \int_{t_0}^t ds + g(t_0, X(t_0)) \int_{t_0}^t dW(s) + r_1 \quad (2.6)$$

onde o resto  $r_1$  é dado por

$$\begin{aligned} r_1 &= \int_{t_0}^t \int_{t_0}^s L^0 f(u, X(u)) du ds + \\ &+ \int_{t_0}^t \int_{t_0}^s L^1 f(u, X(u)) dW(u) ds + \\ &+ \int_{t_0}^t \int_{t_0}^s L^0 g(u, X(u)) du dW(s) + \\ &+ \int_{t_0}^t \int_{t_0}^s L^1 g(u, X(u)) dW(u) dW(s) \end{aligned} \quad (2.7)$$

Expansões de ordem superior podem ser encontradas aplicando sucessivamente a

fórmula de Itô às funções integrandas do resto.

Uma construção detalhada da *expansão de Itô-Taylor* pode ser encontrada em [8].

## 2.3 Método de Euler Maruyama

Este método é a versão estocástica do método de Euler para EDOs, e se trata da aplicação direta da *expansão de Itô-Taylor* apresentada acima.

Então, dada a EDE (1.3), temos que o processo estocástico  $X = \{X_t, t_0 \leq t \leq T\}$  é uma solução de (1.3), assim tomamos uma partição do intervalo de tempo  $[t_0, T]$ , e aplicamos a expansão de Taylor descartando o resto em cada “subintervalo” desta partição, temos que a aproximação de Euler Maruyama de  $X(t_{n+1})$  é dada por

$$\begin{aligned} X_{n+1} &= X_n + f(t_n, X_n) \int_{t_n}^{t_{n+1}} ds + g(t_n, X_n) \int_{t_n}^{t_{n+1}} dW(s) \\ &= X_n + f(t_n, X_n)(t_{n+1} - t_n) + g(t_n, X_n)(W(t_{n+1}) - W(t_n)) \end{aligned} \quad (2.8)$$

tomando  $h_{n+1} = t_{n+1} - t_n$ , e a propriedade (ii) da definição 1.3.1 e obtemos a expressão recursiva do método de Euler Maruyama

$$X_{n+1} = X_n + f(t_n, X_n)h_{n+1} + g(t_n, X_n)\sqrt{h_{n+1}}N(0,1) \quad (2.9)$$

onde  $N(0,1)$  é uma variável aleatória com distribuição normal, com média 0 e variância 1, que é gerada para cada passo do método.

**Teorema 2.3.1.** (*Ordem de convergência do método de Euler Maruyama*)

Seja  $X_n$  a aproximação da solução  $X(t)$  de uma EDE pelo método de Euler Maruyama, suponhamos que

$$(i) \quad \mathbb{E}(|X(t_0)|^2) < \infty,$$

$$(ii) \quad \mathbb{E}(|X(t_0) - X_0|^2)^{\frac{1}{2}} \leq K_1 h^{\frac{1}{2}},$$

$$(iii) \quad |f(t, x) - f(t, y)| + |g(t, x) - g(t, y)| \leq K_2 |x - y|,$$

$$(iv) \quad |f(t, x)| + |g(t, x)| \leq K_3(1 + |x|),$$

$$(v) \quad |f(s, x) - f(t, x)| + |g(s, x) - g(t, x)| \leq K_4(1 + |x|)|s - t|^{\frac{1}{2}},$$

para todo  $s, t \in [t_0, T]$ , com  $t_0 \leq 0$  e  $x, y \in \mathbb{R}^d$ , com  $K_1, K_2, K_3, K_4$  e  $C$  são constantes que não dependem de  $h$  então podemos afirmar que o erro forte do método de Euler Maruyama é dado por

$$erro_h^{forte} = \sup_n (\mathbb{E}[|x(t_n) - x_n|]) \leq C.h^{\frac{1}{2}}$$

e assim a ordem de convergência forte é dada por  $\gamma_{euler} = \frac{1}{2}$ .

A demonstração deste teorema pode ser encontrada em [8].

Para EDEs específicas, o método de Euler Maruyama pode ter ordem de convergência superior a 0,5, em específico, para as equações com ruído aditivo (conforme descritas na seção 1.7 do capítulo 1) sob certas condições acerca da suavidade da função de deslocamento  $f(t, X(t))$  e da função de difusão  $g(t)$ , este método tem ordem de convergência forte  $\gamma = 1$ .

## 2.4 Método de Milstein

O método de Milstein é construído de forma semelhante ao método de Euler Maruyama, pois também se baseia em aplicar a expansão de Itô-Taylor a cada subintervalo de uma partição do intervalo  $[t_0, T]$ , mas tomamos mais um termo da expansão de Taylor. Assim,

aplicando a fórmula de Itô ao integrando da última integral dupla do resto  $r_1$  temos

$$\begin{aligned} \int_{t_0}^t \int_{t_0}^s L^1 g(u, X(u)) dW(u) dW(s) &= L^1 g(t_n, X(t_n)) \int_{t_0}^t \int_{t_0}^s dW(u) dW(s) + \\ &+ \int_{t_0}^t \int_{t_0}^s \int_{t_n}^u L^0 L^1 g(v, X(v)) dv dW(u) dW(s) + \\ &+ \int_{t_0}^t \int_{t_0}^s \int_{t_0}^u L^1 L^1 g(v, X(v)) dW(v) dW(u) dW(s) \end{aligned} \quad (2.10)$$

Substituindo (2.10) em (2.7) e depois em (2.6) obtemos

$$\begin{aligned} X(t) &= X(t_0) + f(t_0, X(t_0)) \int_{t_0}^t ds + g(t_0, X(t_0)) \int_{t_0}^t dW(s) + \\ &+ L^1 g(t_0, X(t_0)) \int_{t_0}^t \int_{t_0}^s dW(u) dW(s) + r_2 \end{aligned} \quad (2.11)$$

onde  $r_2$  é dado por

$$\begin{aligned} r_2 &= \int_{t_0}^t \int_{t_0}^s L^0 f(u, X(u)) du ds + \\ &+ \int_{t_0}^t \int_{t_0}^s L^1 f(u, X(u)) dW(u) ds + \\ &+ \int_{t_0}^t \int_{t_0}^s L^0 g(u, X(u)) du dW(s) + \\ &+ \int_{t_0}^t \int_{t_0}^s \int_{t_0}^u L^0 L^1 g(v, X(v)) dv dW(u) dW(s) + \\ &+ \int_{t_0}^t \int_{t_0}^s \int_{t_0}^u L^1 L^1 g(v, X(v)) dW(v) dW(u) dW(s) \end{aligned} \quad (2.12)$$

Analogamente à construção do método de Euler, descartamos o resto  $r_2$  e aplicamos a equação (2.11) à cada subintervalo de uma partição do intervalo de tempo  $[t_0, T]$ , e assim temos

$$\begin{aligned}
X(t_{n+1}) &= X(t_n) + f(t_n, X(t_n)) \int_{t_n}^{t_{n+1}} ds + g(t_n, X(t_n)) \int_{t_n}^{t_{n+1}} dW(s) + \\
&\quad + L^1 g(t_n, X(t_n)) \int_{t_n}^{t_{n+1}} \int_{t_n}^s dW(u) dW(s) \\
&= X(t_n) + f(t_n, X(t_n))(t_{n+1} - t_n) + g(t_n, X(t_n))(W(t_{n+1}) - W(t_n)) + \\
&\quad + g(t_n, X(t_n))g_x(t_n, X(t_n)) \int_{t_n}^{t_{n+1}} (W(t_{n+1}) - W(t_n)) dW(s) \\
&= X(t_n) + f(t_n, X(t_n))(t_{n+1} - t_n) + g(t_n, X(t_n))(W(s) - W(t_n)) + \\
&\quad + g(t_n, X(t_n))g_x(t_n, X(t_n)) \left[ \int_{t_n}^{t_{n+1}} W(s) dW(s) - W(t_n)W(t_{n+1}) + \right. \\
&\quad \left. + W^2(t_n) \right] \\
&= X(t_n) + f(t_n, X(t_n))(t_{n+1} - t_n) + g(t_n, X(t_n))(W(s) - W(t_n)) + \\
&\quad + g(t_n, X(t_n))g_x(t_n, X(t_n)) \left[ \frac{1}{2}(W^2(t_{n+1}) - W^2(t_n) - (t_{n+1} - t_n)) + \right. \\
&\quad \left. - W(t_n)W(t_{n+1}) + W^2(t_n) \right] \\
&= X(t_n) + f(t_n, X(t_n))(t_{n+1} - t_n) + g(t_n, X(t_n))(W(t_{n+1}) - W(t_n)) + \\
&\quad + \frac{1}{2}g(t_n, X(t_n))g_x(t_n, X(t_n)) \left[ (W(t_{n+1}) - W(t_n))^2 - (t_{n+1} - t_n) \right]
\end{aligned} \tag{2.13}$$

e finalmente tomamos  $h_{n+1} = t_{n+1} - t_n$ , e utilizamos a propriedade (ii) da definição 1.3.1, e obtemos a expressão recursiva do método de Milstein

$$\begin{aligned}
X_{n+1} &= X_n + f(t_n, X_n)(t_{n+1} - t_n) + g(t_n, X_n)\sqrt{h_{n+1}}N(0,1) + \\
&\quad + \frac{1}{2}g(t_n, X_n)g_x(t_n, X_n)h_{n+1} \left[ (N(0,1))^2 - 1 \right]
\end{aligned} \tag{2.14}$$

A ordem de convergência do método de Milstein pode ser verificada de forma análoga ao que fizemos para Euler no teorema 2.3.1, mas vamos mostrar de uma forma mais intuitiva, assim, voltando à primeira igualdade da equação (2.13), note que o método

de Milstein utiliza uma integral estocástica múltipla em  $dW.dW$ , como a ordem de convergência surge do quão maior for a expansão Itô-Taylor, temos que

$$erro_h^{forte} \leq dW.dW \approx \Delta W.\Delta W = \mathcal{O}(h^{1/2}.h^{1/2}) = \mathcal{O}(h^1)$$

logo a ordem de convergência do método de Milstein é dada por  $\gamma = 1$ . Uma prova rigorosa pode ser encontrada em [8].

Uma abordagem numérica muito interessante para a verificação da ordem de convergência dos métodos de Euler Maruyama e Milstein é dada em [25].

Outros métodos de ordem superior podem ser construídos seguindo a mesma ideia da construção do método de Euler Maruyama e do método de Milstein, ou seja, usando expansões Itô-Taylor de ordens ainda maiores.

*Observação.* No caso de EDEs com ruído aditivo, temos que  $g$  não depende de  $X$ , então  $g_x = 0$ , logo o método de Milstein fica idêntico ao método de Euler Maruyama.

Em [26] podem ser vistas algumas aplicações dos métodos de Euler Maruyama e de Milstein em sistemas famosos, como o modelo de Black-Scholes<sup>1</sup>.

Até aqui, estivemos focados em apresentar as noções básicas de EDEs, e os métodos numéricos mais comuns e gerais quanto a sua aplicação. Da próxima seção em diante iremos finalmente tratar do assunto que motivou este trabalho, que são os métodos de linearização local para EDEs com ruído aditivo, métodos estes que serão utilizados no capítulo 4, na construção do novo esquema adaptativo proposto nesta dissertação.

## 2.5 Métodos de Linearização Local

Os métodos de linearização local (LL) consistem em aproximar a função de deslocamento (drift) com uma função linear, pois esta costuma oferecer uma aproximação melhor do que o uso de uma constante, como é o caso dos métodos que apresentamos anteriormente.

---

<sup>1</sup>Modelo matemático para precificação de opções financeiras (Europeias).

Geralmente, cada método de linearização local atende a algum tipo específico de EDE, por exemplo, o método de Ozaki<sup>2</sup> é destinado a EDEs com função de difusão constante. E como já estabelecido, nesta dissertação vamos trabalhar com EDEs com ruído aditivo, sendo assim iremos nos focar em métodos LL para equações do tipo (1.14). Então dada a EDE

$$dX(t) = f(t, X(t))dt + G(t)dW(t).$$

Queremos aproximá-la por outra EDE que seja mais simples de resolver, particularmente uma EDE linear que seja próxima da nossa equação original. Para tal, podemos utilizar a expansão de Taylor determinística para aproximar a função de deslocamento  $f(t, X(t))$  como uma função de  $(t, X)$ , assim, como queremos aproximar  $f$  no instante  $(t, X(t))$  partindo de  $(t_n, X_n)$ , obtemos

$$f(t, X(t)) \approx f(t_n, X_n) + f_x(t_n, X_n)(X(t) - X_n) + f_t(t_n, X_n)(t - t_n) \quad (2.15)$$

onde  $f_x$  é a matriz jacobiana de  $f$ , e  $f_t = \frac{\partial f}{\partial t}$ . Então usando (2.15), temos a seguinte “aproximação” de (1.14)

$$dX(t) = [f_x(X(t) - X_n) + (f_t(t_n, X_n)(t - t_n) + f(t_n, X_n))] dt + G(t)dW_n(t) \quad (2.16)$$

Assim, podemos definir a fórmula recursiva do método de Linearização Local 1 (LL 1) como sendo a solução da equação (2.16) em  $t_{n+1}$ , partindo de  $X_n$ , aplicado a cada subintervalo de uma partição de  $[t_0, T]$ .

Note que esta aproximação da função de deslocamento pode ser mais precisa se utilizarmos que  $X(t)$  satisfaz a equação (1.14), sendo assim, vamos tomar uma função

---

<sup>2</sup>Mais sobre o método de Ozaki pode ser encontrado em [27].

auxiliar  $U : [t_0, T] \times \mathbb{R}^d \rightarrow \mathbb{R}^k$ , tal que

$$U(t, X(t)) = U(t_n, X_n) + \int_{t_n}^t L^0 U(s, X(s)) ds + \sum_{j=1}^m \int_{t_n}^t L^j U(s, X(s)) dW_s^j$$

onde os operadores  $L^0$  e  $L^j$  são dados por (2.3) e (2.4).

Agora, para  $U \equiv f$ , obtemos

$$f(t, X(t)) = f(t_n, X_n) + \int_{t_n}^t L^0 f(s, X(s)) ds + \sum_{j=1}^m \int_{t_n}^t L^j f(s, X(s)) dW_s^j$$

e para  $U \equiv L^0 f$ , a equação acima fica

$$f(t, X(t)) = f(t_n, X_n) + L^0 f(t_n, X_n) \int_{t_n}^t ds + \sum_{j=1}^m L^j f(t_n, X_n) \int_{t_n}^t dW_s^j + r_1$$

onde o resto  $r_1$  é formado por integrais estocásticas múltiplas.

Mas note que

$$\begin{aligned} L^0 f(t_n, X_n) &= \left[ f_t + \underbrace{\langle f, f_x \rangle}_{f_x f} + \frac{1}{2} \sum_{j=1}^m (I \otimes g_j^\top) f_{xx} g_j \right]_{(t_n, X_n)} \\ L^j f(t_n, X_n) &= \underbrace{[\langle g_j, f_x \rangle]}_{f_x g_j}_{(t_n, X_n)} \end{aligned}$$

onde  $\otimes$  representa o produto de Kronecker e  $f_{xx}$  é a matriz Hessiana de  $f$ . Assim temos

$$\begin{aligned} f(t, X(t)) = f(t_n, X_n) + & \left[ f_t + f_x f + \frac{1}{2} \sum_{j=1}^m (I_{d \times d} \otimes g_j^\top) f_{xx} g_j \right] (t - t_n) + \\ & + \sum_{j=1}^m (f_x g_j) (W^j(t) - W^j(t_n)) + r_1 \end{aligned} \quad (2.17)$$

com  $f$ ,  $f_x$ ,  $f_{xx}$  e  $f_t$  avaliados em  $(t_n, X_n)$ , e  $g$  em  $t_n$ .

Agora para  $U \equiv X(t)$ ,

$$X(t) = X_n + f(t_n, X_n) \int_{t_n}^t ds + \sum_{j=1}^m g_j(t_n) \int_{t_n}^t dW^j(s) + r_2$$

$\Rightarrow$

$$X(t) = X_n + f(t_n, X_n)(t - t_n) + \sum_{j=1}^m g_j(t_n)(W^j(t) - W^j(t_n)) + r_2 \quad (2.18)$$

Multiplicando (2.18) por  $f_x(t_n, X_n)$  obtemos

$$\begin{aligned} f_x(t_n, X_n)(X(t) - X_n) &= f_x(t_n, X_n)f(t_n, X_n)(t - t_n) + \\ &+ \sum_{j=1}^m f_x(t_n, X_n)g_j(t_n)(W^j(t) - W^j(t_n)) + \bar{R}_2 \end{aligned} \quad (2.19)$$

Assim, de (2.17) e (2.19), e desconsiderando os restos, temos

$$\begin{aligned} dX(t) &= \left[ f_x(X(t) - X_n) + \left( f_t + \frac{1}{2} \sum_{j=1}^m (I_{d \times d} \otimes g_j^\top) f_{xx} g_j \right) (t - t_n) + f \right] dt + \\ &+ G(t) dW(t) \end{aligned} \quad (2.20)$$

com  $f$ ,  $f_x$ ,  $f_{xx}$  e  $f_t$  avaliados em  $(t_n, X_n)$ , e  $g$  em  $t_n$ . E assim definimos o método LL 1.5 como a solução da equação (2.20), aplicada a cada subintervalo de uma partição de  $[t_0, T]$ .

### 2.5.1 Esquemas exponenciais para LL 1 e LL 1.5

Vamos apresentar agora como construir os esquemas de cálculo para os métodos LL 1 e LL 1.5 conforme pode ser visto em [12]. Primeiro, note que a equação (2.16) referente ao método LL 1 e a equação (2.20) do método LL 1.5, representam aproximações da

EDE com ruído aditivo (1.14), e por si só não podem ser calculadas, precisamos então, determinar uma solução para essas EDEs para obtermos esquemas calculáveis para aproximar numericamente esse tipo de equação.

Desta forma, dado  $h_{n+1} = t_{n+1} - t_n$ , suponhamos que

$$(LL\ 1) \quad X_{n+1} = X_n + \phi_1(t_n, X_n; h_{n+1}) + \xi_1(t_n, X_n; h_{n+1}) \quad (2.21)$$

$$(LL\ 1.5) \quad X_{n+1} = X_n + \phi_{1.5}(t_n, X_n; h_{n+1}) + \xi_{1.5}(t_n, X_n; h_{n+1}) \quad (2.22)$$

sejam soluções de (2.16) e (2.20) respectivamente. Onde  $\phi$  representa a parte determinística e  $\xi$  a parte estocástica da solução.

Assim, de [12] temos que  $\phi_1$  de (2.21) pode ser expressado por

$$\phi_1(t_n, X_n; h_{n+1}) = \begin{pmatrix} I_{d \times d} & \mathbf{0}_{d \times 2} \end{pmatrix} e^{\Lambda_1(t_n, X_n)h_{n+1}} \cdot \begin{pmatrix} \mathbf{0}_{d \times 1} \\ 0 \\ 1 \end{pmatrix} \quad (2.23)$$

onde a matriz  $\Lambda_1$  é dada por

$$\Lambda_1(t_n, X_n) = \begin{pmatrix} f_x(t_n, X_n) & f_t(t_n, X_n) & f(t_n, X_n) \\ \mathbf{0}_{1 \times d} & 0 & 1 \\ \mathbf{0}_{1 \times d} & 0 & 0 \end{pmatrix} \quad (2.24)$$

com  $I_{d \times d}$  é a matriz identidade de dimensão  $d$ , e  $\mathbf{0}_{a \times b}$  é a matriz nula com  $a$  linhas, e  $b$  colunas. e  $\xi_1$  de (2.21) é

$$\xi_1(t_n, X_n; h_{n+1}) = G(t_n) \Delta W_{[t_n, t_{n+1}]} \quad (2.25)$$

onde  $\Delta W_{[t_n, t_{n+1}]} = W_{t_{n+1}} - W_{t_n} \sim \sqrt{h_{n+1}} \cdot N(0,1)$ . E assim fica definido o *esquema exponencial de LL 1*.

Analogamente ao LL 1, temos que o *esquema exponencial para LL 1.5* é dado por

$$\phi_{1.5}(t_n, X_n; h_{n+1}) = \begin{pmatrix} I_{d \times d} & \mathbf{0}_{d \times 2} \end{pmatrix} e^{\Lambda_{1.5}(t_n, X_n) h_{n+1}} \cdot \begin{pmatrix} \mathbf{0}_{d \times 1} \\ 0 \\ 1 \end{pmatrix} \quad (2.26)$$

com  $\Lambda_{1.5}$  dado por

$$\Lambda_{1.5}(t_n, X_n) = \begin{pmatrix} f_x(t_n, X_n) & f_t(t_n, X_n) + \frac{1}{2} \sum_{j=1}^m (I_{d \times d} \otimes g_j(t_n)^\top) f_{xx}(t_n, X_n) g_j(t_n) & f(t_n, X_n) \\ \mathbf{0}_{1 \times d} & 0 & 1 \\ \mathbf{0}_{1 \times d} & 0 & 0 \end{pmatrix} \quad (2.27)$$

Agora, para  $\xi_{1.5}$  de (2.22) vamos tomar a expansão Itô-Taylor de ordem 1.5 apenas para a parte estocástica de (2.20), assim, descartando o resto  $r_3$  temos

$$\begin{aligned} \xi_{1.5}(t_n, X_n; h_{n+1}) &= G(t_n) \int_{t_n}^{t_{n+1}} dW(s) + f_x(t_n, X_n) G(t_n) \int_{t_n}^{t_{n+1}} \int_{t_n}^s dW(u) ds + \\ &+ G_t(t_n) h_{n+1} \int_{t_n}^{t_{n+1}} dW(s) - G_t(t_n) \int_{t_n}^{t_{n+1}} \int_{t_n}^s dW(u) ds \end{aligned}$$

Tomando

$$\begin{aligned} \Delta W_{[t_n, t_{n+1}]} &= \int_{t_n}^{t_{n+1}} dW(s) \\ \Delta Z_{[t_n, t_{n+1}]} &= \int_{t_n}^{t_{n+1}} \int_{t_n}^s dW(u) ds \end{aligned}$$

onde  $\Delta W_{[t_n, t_{n+1}]} = W_{t_{n+1}} - W_{t_n}$  tem distribuição normal com média 0 e desvio padrão  $\sqrt{h_{n+1}}$ , e  $\Delta Z_{[t_n, t_{n+1}]}$  tem distribuição normal com média 0, e desvio padrão  $\frac{h_{n+1}^{3/2}}{\sqrt{3}}$ , e a

covariância  $E(\Delta W, \Delta Z) = \frac{1}{2}h_{n+1}^2$ . Assim, dadas duas variáveis aleatórias com distribuição normal  $N_1$  e  $N_2$ , independentes entre si, com média 0 e desvio padrão 1, ficamos com

$$\Delta W_{[t_n, t_{n+1}]} = \sqrt{h_{n+1}} N_1 \quad (2.28)$$

$$\Delta Z_{[t_n, t_{n+1}]} = \frac{h_{n+1}\sqrt{h_{n+1}}}{2} \left( N_1 + \frac{N_2}{\sqrt{3}} \right) \quad (2.29)$$

e assim temos

$$\begin{aligned} \xi_{1.5}(t_n, X_n; h_{n+1}) &= G(t_n)\Delta W_{[t_n, t_{n+1}]} + f_x(t_n, X_n)G(t_n)\Delta Z_{[t_n, t_{n+1}]} + \\ &\quad + G_t(t_n)(\Delta W_{[t_n, t_{n+1}]}h_{n+1} - \Delta Z_{[t_n, t_{n+1}]}) \end{aligned} \quad (2.30)$$

e assim, de (2.22), (2.26) e (2.30) fica definido o *esquema exponencial de LL 1.5*.

De [16] temos que LL 1 e LL 1.5 possuem ordens de convergência 1 e 1,5 respectivamente, como os próprios nomes já sugerem. Esquemas LL de ordem superior para EDEs com ruído aditivo podem ser construídos, e podem ser encontrados em [14] e [16].

Estes esquemas nos fornecem uma forma para implementar os métodos LL de ordem 1 e 1,5.



## Capítulo 3

# Estabilidade

O estudo da estabilidade numérica surgiu quando começou-se a utilizar computadores para calcular aproximações de integrais, e por consequência, determinar soluções numéricas de equações diferenciais. Métodos comprovadamente convergentes geravam resultados com erros muito maiores do que se esperava, erros os quais não eram explicados apenas pela discretização do problema, enfim, este fenômeno era causado pelo acúmulo de sucessivos erros de truncamento. Duas boas referências para estabilidade numérica são [29] e [30].

De maneira informal, podemos dizer que estabilidade é a insensibilidade do estado do sistema em relação a pequenas mudanças nos seus parâmetros, ou no seu estado inicial. Assim, em um sistema estável, as trajetórias que são próximas em determinado instante, permanecerão próximas em cada um dos instantes seguintes. No caso de métodos para solução numérica de EDEs não é diferente.

Para ilustrar o problema no caso estocástico, construímos dois exemplos de EDEs diferentes, mas que possuem mesma solução.

**Exemplo 3.0.1.**

$$d \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} = \left[ A_1 \cdot \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} + \begin{pmatrix} 2\text{sen}(t) \\ 2(\cos(t) - \text{sen}(t)) \end{pmatrix} \right] dt + e^{A_1 t} \begin{pmatrix} 1 \\ 1 \end{pmatrix} dW(t) \quad (3.1)$$

onde

$$A_1 = \begin{pmatrix} -2 & 1 \\ 1 & -2 \end{pmatrix}$$

com condição inicial

$$\begin{pmatrix} x_1(0) \\ x_2(0) \end{pmatrix} = \begin{pmatrix} 2 \\ 3 \end{pmatrix}$$

**Exemplo 3.0.2.**

$$d \begin{pmatrix} y_1(t) \\ y_2(t) \end{pmatrix} = \left[ A_2 \cdot \begin{pmatrix} y_1(t) \\ y_2(t) \end{pmatrix} + \begin{pmatrix} 2\text{sen}(t) \\ 998(\cos(t) - \text{sen}(t)) \end{pmatrix} \right] dt + e^{A_2 t} \begin{pmatrix} 1 \\ 1 \end{pmatrix} dW(t) \quad (3.2)$$

onde

$$A_2 = \begin{pmatrix} -2 & 1 \\ 998 & -999 \end{pmatrix}$$

com condição inicial

$$\begin{pmatrix} y_1(0) \\ y_2(0) \end{pmatrix} = \begin{pmatrix} 2 \\ 3 \end{pmatrix}$$

Aplicando o método de variação de parâmetros (apresentado na seção 1.8.1) aos exemplos 3.0.1 e 3.0.2, obtemos que ambas as EDEs possuem a mesma solução

$$\begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} = \begin{pmatrix} y_1(t) \\ y_2(t) \end{pmatrix} = 2e^{-t} \cdot \begin{pmatrix} 1 \\ 1 \end{pmatrix} + \begin{pmatrix} \text{sen}(t) \\ \cos(t) \end{pmatrix} + e^{-t} \cdot \begin{pmatrix} 1 \\ 1 \end{pmatrix} \cdot W(t). \quad (3.3)$$

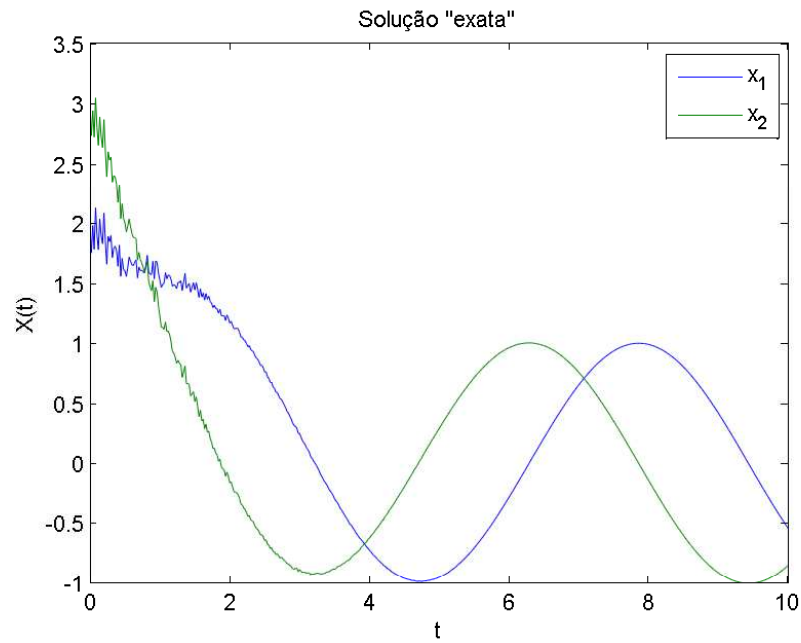


Figura 3.1: Realização da solução dos exemplos 3.0.1 e 3.0.2 sobre uma trajetória browniana gerada aleatoriamente.

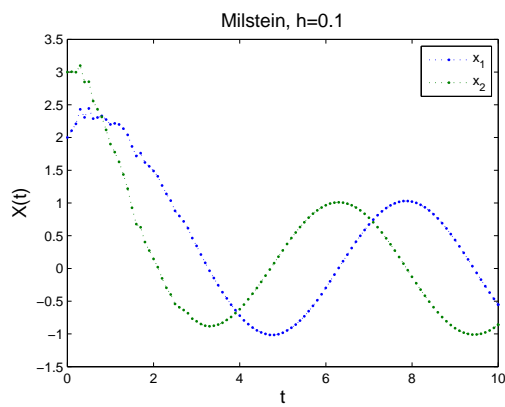


Figura 3.2: Simulação do exemplo 3.0.1 utilizando o método de Milstein com 100 passos.

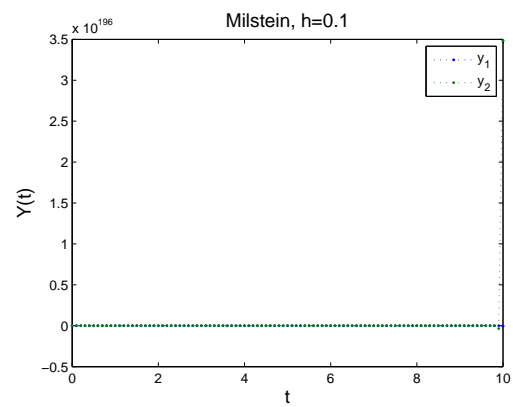


Figura 3.3: Simulação do exemplo 3.0.2 utilizando o método de Milstein com 100 passos.

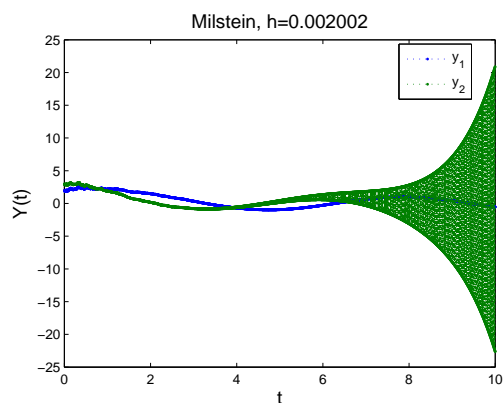


Figura 3.4: Simulação do exemplo 3.0.2 utilizando o método de Milstein com 4995 passos.

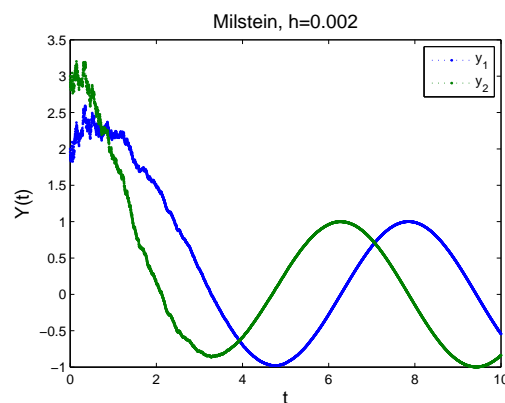


Figura 3.5: Simulação do exemplo 3.0.2 utilizando o método de Milstein com 5000 passos.

Na figura 3.1 podemos ver como a solução dos exemplos 3.0.1 e 3.0.2 se comporta, e vamos comparar visualmente com os resultados obtidos simulando as equações por meio do método de Milstein (figuras 3.2, 3.3, 3.4 e 3.5). Na figura 3.2 vemos a simulação do primeiro exemplo com tamanho de passo  $h = 0,1$  (100 passos), e podemos ver que a dinâmica da solução se mantém. Já na figura 3.3, temos a simulação do segundo exemplo com  $h = 0,1$ , e podemos ver que a dinâmica do problema se perde completamente, e nas figuras que se seguem temos simulações do segundo exemplo com 4995 passos ( $h = 0,002002$ ), e com 5000 passos ( $h = 0,002$ ) onde a dinâmica do problema se mantém, assim, podemos ver que embora os exemplos 3.0.1 e 3.0.2 tenham exatamente a mesma solução, utilizando o método de Milstein, a solução numérica do exemplo 3.0.1 consegue se manter próxima da solução exata com passos  $h = 0,1$ , já no caso do exemplo 3.0.2 precisamos reduzir significativamente o tamanho de passo para  $h = 0,002$  para obter um resultado aceitável.

A conclusão que queremos extrair dos exemplos acima é que existem equações que geram instabilidade em alguns métodos numéricos, métodos esses ditos instáveis. Como pudemos ver, este fenômeno não está ligado à forma da solução, e sim ao sistema diferencial em si.

A equação do exemplo 3.0.2 é uma EDE *rígida*<sup>1</sup>, e esta rigidez se apresenta por que o vetor  $(y_1, y_2)^\top$  tem uma componente “lenta” e outra “rápida”, a componente rápida,  $y_2$  está associada ao autovalor  $-1000$ , que mesmo quando está presente na solução, esta decai rapidamente e se torna desprezível para o resultado final, mas para a solução numérica esta componente exige uma redução considerável no tamanho de passo  $h$  para que esse comportamento seja reproduzido pelo método sem gerar instabilidade na solução. É importante frisar que não há uma definição precisa do conceito de rigidez, mas há boas discussões sobre o tema (no caso determinístico) em [31], [29] e [3].

Uma boa definição para métodos numéricos estáveis é dada em [33], como pode ser visto abaixo.

**Definição 3.0.1.** Um método estável é aquele cujos resultados dependem continuamente dos dados iniciais.

Para estudar a estabilidade linear de métodos numéricos para EDEs vamos considerar a EDE de teste

$$dX(t) = \lambda X(t)dt + dW(t), \quad (3.4)$$

onde  $W$  é um processo de Wiener padrão, e a constante  $\lambda$  é um número complexo. Seja  $t_{n+1} = t_n + h$  uma partição do intervalo de tempo  $[t_0, T]$ , e  $X_n$  uma discretização da equação (3.4) através de um método numérico dado. Suponhamos então que para qualquer tamanho de passo  $h$ ,  $X_n$  pode ser representado por

$$X_{n+1} = R(\lambda h)X_n + U_n, \quad (3.5)$$

onde  $R : \mathbb{C} \rightarrow \mathbb{C}$ , e  $U_n$  é uma variável aleatória que não depende de nenhum  $X_n$ . Assim, podemos definir o conceito de região de estabilidade.

---

<sup>1</sup>Rígidas do inglês *stiff*.

**Definição 3.0.2.** Dada a EDE (3.4), com uma discretização (3.5), uma região de estabilidade  $S$  é dada por

$$S := \{\lambda h \in \mathbb{C} : h > 0, \operatorname{Re}\{\lambda\} < 0, \quad |R(\lambda h)| < 1\}. \quad (3.6)$$

Agora podemos definir o conceito de A-estabilidade.

**Definição 3.0.3.** Um método numérico é dito A-estável quando a sua região de estabilidade contém todo o semiplano negativo do plano complexo.

Então, se 3.5 é A-estável, temos que se  $(X_n)$  e  $(X'_n)$ , com condições iniciais  $X_0$  e  $X'_0$ , são aproximações da solução de (3.4), então

$$X_n - X'_n = (R(\lambda h))^n (X_0 - X'_0) \rightarrow 0 \text{ quando } n \rightarrow \infty,$$

para todo tamanho de passo  $h$ .

O resultado a seguir garante que os métodos LL que apresentamos aqui são A-estáveis.

**Teorema 3.0.1.** *Os métodos LL 1 e LL 1.5 são A-estáveis.*

*Demonstração.* Para a equação de teste (3.4), a EDO de sua parte determinística é dada por

$$dX(t) = \lambda X(t)dt, \quad (3.7)$$

assim, tanto a equação (2.16) quanto a equação (2.20) ficam da seguinte forma

$$dX(t) = \lambda X(t)dt + G(t)dW(t), \quad (3.8)$$

Então, toda discretização  $X_n$  de (3.4) é dada na forma

$$X_{n+1} = X_n + \phi(t_n, X_n) + \xi(t_n)$$

onde  $X_n + \phi(t_n, X_n)$  é um passo de uma discretização LL para a EDO (3.7), e  $\xi(t_n)$  é um passo de alguma discretização para o problema estocástico (3.8). De [34] sabemos que o esquema LL é A-estável para EDOs e então,  $\xi$  não depende de  $X_n$  para todo  $n$ , o que implica que LL 1 e LL 1.5 são A-estáveis.  $\square$

Voltando ao exemplo 3.0.2, vamos mostrar como o método LL 1 que como acabamos de ver, é A-estável, se comporta para simular uma EDE rígida.

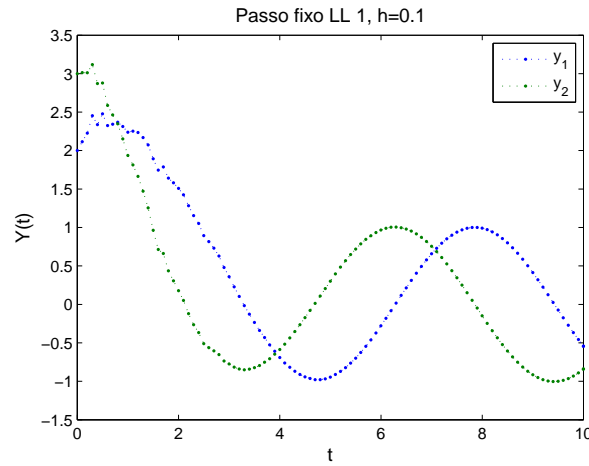


Figura 3.6: Simulação do exemplo 3.0.2 utilizando o método LL de ordem 1 com 100 passos.

E como era de se esperar, o método LL 1 não teve dificuldades para determinar a solução numérica de uma EDE rígida, inclusive, com este método podemos aumentar ainda mais o tamanho de passo  $h$  e ainda obter “boas” soluções.

Assim, podemos concluir que a importância do estudo de métodos estáveis está no fato de que tais métodos nos permitem obter soluções numéricas satisfatórias de equações rígidas (ou rígidas em alguns intervalos) sem ter que reduzir em demasia o tamanho de passo, que é pertinente, pois EDEs rígidas surgem com certa frequência na modelagem de diversos problemas. Entretanto, ainda é possível aperfeiçoar a utilização destes métodos com a utilização de passos de tamanho variável como veremos no próximo capítulo.



## Capítulo 4

# Algoritmo adaptativo

### 4.1 Motivação intuitiva

Vamos começar com uma analogia ao problema de integrar numericamente uma função que tem grande variação em apenas uma parte de seu domínio. Suponha que queiramos calcular numericamente a integral entre 0 e  $c$  de uma função  $f(t)$  como esta do gráfico abaixo.

Note que a função é praticamente constante nos intervalos  $[0, a]$  e  $[b, c]$ , mas no intervalo  $[a, b]$  ela é bem complicada. Se fossemos utilizar um método de passo fixo  $h$ , teríamos que usar um  $h$  bem pequeno para que possamos obter uma boa estimativa para esta integral, mas apenas o intervalo  $[a, b]$  é problemático. Um método com passo temporal adaptativo permite que nas partes mais “simples” da função o método avance com passos maiores, e nas partes mais “complicadas” avance com passos menores, controlando o *erro* numérico do método. Agora vamos estender essa ideia para a integração numérica de EDEs.

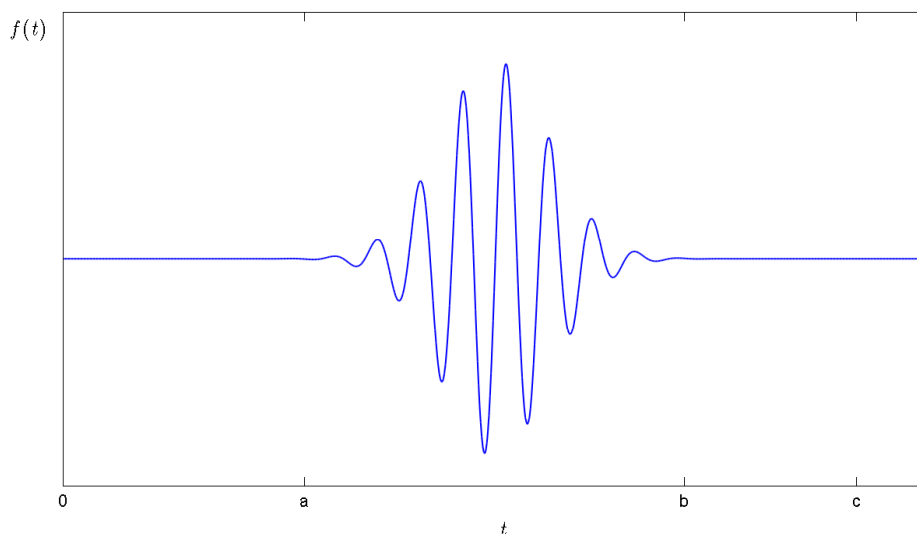


Figura 4.1: Exemplo motivacional

## 4.2 Introdução

Pensando de forma análoga ao que foi mostrado acima, ao incorporar um passo temporal variável (ou adaptativo) em um método para solução numérica de EDEs, pretendemos dar mais atenção aos intervalos onde a EDE sofra variações mais bruscas, efetuando assim passos menores, e podendo então, dar passos maiores no restante do cálculo da solução.

Desta forma, iremos neste capítulo apresentar um algoritmo que determina qual o tamanho do próximo passo que o método irá executar, bem como analisa se este foi ou não um bom passo, e enquanto isso constrói (ou reconstrói) a trajetória Browniana de forma concisa.

O algoritmo adaptativo que vamos apresentar foi adaptado do algoritmo publicado em [35], e este foi selecionado depois do estudo de diversos outros métodos/esquemas, como [36], [37], [38], [39], [40] e outros menos relevantes. Embora este esquema se baseie fortemente no que foi apresentado em [41] e [36], ele foi escolhido pelo fato de trabalhar

com métodos para convergência forte, poder ser aplicado diretamente a EDEs com ruído aditivo, e por se adaptar bem aos métodos LL com esquema exponencial que foram apresentados, além de fornecer uma forma simples de lidar com o problema da reconstrução da trajetória Browniana, garantindo assim que esta seja seguida de forma correta, mantendo a coerência do método. Desta forma, ele se encaixa perfeitamente para os objetivos deste trabalho.

Foi provado em [41] que quando se utiliza passo temporal variável, um método numérico estocástico deve ter pelo menos ordem 1 para garantir a convergência para a solução correta. Como o método que iremos utilizar é a linearização local de ordem 1,5, esta questão está satisfeita.

### 4.3 Estimando o erro local e um próximo passo ideal

Seja  $X_1(t)$  a aproximação de  $X$  no tempo  $t$  pelo método LL de ordem 1, e  $X_{1.5}(t)$  a aproximação de  $X$  no tempo  $t$  pelo método LL de ordem 1.5, assim iremos utilizar  $X_{1.5}(t)$  para construir a nossa trajetória-solução, e  $X_1(t)$  e  $X_{1.5}(t)$  para estimar o *erro*. Então para uma EDE d-dimensional, seja  $tol_i$  com  $i = 1, \dots, d$  a tolerância aceita para cada posição  $i$  do vetor solução  $X$ , e dado  $q$  a ordem do método que estamos usando para avançar com a solução, temos que *erro* local é dado por

$$erro = \sqrt{\frac{1}{d} \sum_{i=1}^d \left( \frac{X_{1.5;i} - X_{1;i}}{tol_i} \right)^2} \quad (4.1)$$

Como é desejável que  $X_{1.5;i} - X_{1;i} \approx tol_i$ , o passo é aceito se  $erro \leq 1$ , e descartamos o passo se  $erro > 1$ .

Agora, vamos determinar um próximo passo ótimo. É desejável que o *erro* seja o mais próximo possível de 1, sendo assim

$$\begin{aligned}
erro &\approx Ch^{q+\frac{1}{2}} \\
1 &\approx Ch_{\acute{o}timo}^{q+\frac{1}{2}} \\
\Rightarrow \\
h_{\acute{o}timo} &= h \left( \frac{1}{erro} \right)^{\frac{1}{q+\frac{1}{2}}}
\end{aligned} \tag{4.2}$$

onde  $h$  é o tamanho do último passo efetuado. E como estamos usando o método LL 1.5, temos  $q = 1,5$  então

$$h_{\acute{o}timo} = h \sqrt{\frac{1}{erro}} \tag{4.3}$$

*Observação.* Note que por estarmos utilizado métodos A-estáveis (conforme visto no capítulo 3) para a construção do método adaptativo que estamos propondo, temos que o algoritmo não precisa reduzir o tamanho de passo para controlar a estabilidade do método, mas apenas para controlar a convergência da aproximação.

## 4.4 Mantendo a trajetória Browniana coerente

Suponha que estejamos resolvendo numericamente uma EDE utilizando passo temporal adaptativo, conforme visto na seção 4.3, e tenhamos dado um passo  $h$  e obtivemos  $erro > 1$ , quando este passo foi dado foram gerados números aleatórios e encontramos os valores de  $\Delta W_{[t,t+h]}$  e  $\Delta Z_{[t,t+h]}$ . Mas como o  $erro$  não foi satisfatório, vamos determinar pela equação (4.3) um novo tamanho de passo, que vamos chamar de  $h_a$ , note que como  $erro > 1$  implica que  $h_a < h$ , assim ao executar o novo passo, estaremos dentro de uma região onde o Browniano já foi definido, e não podemos simplesmente desconsiderá-lo e gerar novos valores aleatórios, pois poderíamos mascarar um método ruim com a natureza aleatória das EDEs, sendo assim, vamos apresentar a seguir uma forma de reconstruir

a trajetória Browniana sem perder as informações obtidas anteriormente. Desta forma, vamos tomar  $h_b$  tal que  $h_a + h_b = h$ , assim

$$\begin{array}{c}
 \overbrace{\hspace{10em}}^h \\
 \overbrace{\hspace{4em}}^{h_a} \quad \overbrace{\hspace{6em}}^{h_b} \\
 | \text{---} \text{---} \text{---} | \quad | \text{---} \text{---} \text{---} \text{---} \text{---} \text{---} | \\
 t \qquad \qquad t + h_a \qquad \qquad \qquad t + h \\
 \\
 \underbrace{\hspace{10em}} \\
 \begin{array}{cc}
 \Delta W_{[t, t+h_a]} & \Delta W_{[t+h_a, t+h_a+h_b]} \\
 \Delta Z_{[t, t+h_a]} & \Delta Z_{[t+h_a, t+h_a+h_b]}
 \end{array} \\
 \underbrace{\hspace{10em}} \\
 \begin{array}{c}
 \Delta W_{[t, t+h]} \\
 \Delta Z_{[t, t+h]}
 \end{array}
 \end{array} \tag{4.4}$$

De (2.28) e (2.29) onde  $N_1, N_2$  tem distribuição normal com média 0, e desvio padrão 1, temos que

$$\Delta W_{[t, t+h]} = \sqrt{h} N_1 \tag{4.5}$$

$$\Delta Z_{[t, t+h]} = \frac{h\sqrt{h}}{2} \left( N_1 + \frac{N_2}{\sqrt{3}} \right) \tag{4.6}$$

De [35], sabemos que

$$\Delta W_{[t, t+h]} = \Delta W_{[t, t+h_a]} + \Delta W_{[t+h_a, t+h]} \tag{4.7}$$

$$\Delta Z_{[t, t+h]} = \Delta Z_{[t, t+h_a]} + \Delta Z_{[t+h_a, t+h]} + h_b \Delta W_{[t, t+h_a]} \tag{4.8}$$

Agora, vamos particionar o intervalo  $[t, t+h]$  em  $[t, t+h_a] \cup [t+h_a, t+h]$  de forma que (4.7) e (4.8) sejam satisfeitas. Assim, conhecendo  $\Delta W_{[t, t+h]}$  e  $\Delta Z_{[t, t+h]}$ , e tomando duas novas variáveis aleatórias  $N_1, N_2 \sim N(0,1)$ , temos que

$$\begin{pmatrix} \Delta W_{[t,t+h_a]} \\ \Delta Z_{[t,t+h_a]} \\ \Delta W_{[t+h_a,t+h_a+h_b]} \\ \Delta Z_{[t+h_a,t+h_a+h_b]} \end{pmatrix} = \mathcal{T} \begin{pmatrix} N_1 \\ N_2 \\ \frac{\Delta W_{[t,t+h]}}{\sqrt{h}} \\ \sqrt{3} \left( \frac{2\Delta Z_{[t,t+h]}}{h\sqrt{h}} - \frac{\Delta W_{[t,t+h]}}{\sqrt{h}} \right) \end{pmatrix} \quad (4.9)$$

onde  $\mathcal{T}$  é dado por

$$\mathcal{T} = \begin{pmatrix} 0 & -\sqrt{\frac{h_a h_b}{h^3}}(h_a^2 - h_a h_b + h_b^2) & \frac{h_a}{\sqrt{h}} & \frac{\sqrt{3}h_a h_b}{h\sqrt{h}} \\ \frac{h_a \sqrt{h_a h_b} \sqrt{h_b}}{2\sqrt{3}\sqrt{h}\sqrt{h_a^2 - h_a h_b + h_b^2}} & \frac{h_a h_b \sqrt{h_a h_b} (h_a - h_b)}{2h\sqrt{h}\sqrt{h_a^2 - h_a h_b + h_b^2}} & \frac{h_a^2}{2\sqrt{h}} & \frac{h_a^2 (h_a + 3h_b)}{2h\sqrt{3}h} \\ 0 & \sqrt{\frac{h_a h_b}{(h^3)}}(h_a^2 - h_a h_b + h_b^2) & \frac{h_b}{\sqrt{h}} & -\frac{\sqrt{3}h_a h_b}{h\sqrt{h}} \\ \frac{-h_a \sqrt{h_a h_b} \sqrt{h_b}}{2\sqrt{3}\sqrt{h}\sqrt{h_a^2 - h_a h_b + h_b^2}} & \frac{h_b \sqrt{h_a h_b} \sqrt{h_b} (h_a^2 - h_a h_b + 2h_b^2)}{2h\sqrt{h}\sqrt{h_a^2 - h_a h_b + h_b^2}} & \frac{h_b^2}{2\sqrt{h}} & \frac{h_b^2 h_b - 3h_a}{2\sqrt{3}h\sqrt{h}} \end{pmatrix} \quad (4.10)$$

No algoritmo original, quando partindo de um instante  $t$  várias tentativas de passo são rejeitados, apenas a geração de  $\Delta W$  e  $\Delta Z$  da primeira tentativa falha é atualizada na trajetória Browniana, o nosso algoritmo atualiza a trajetória com todas as tentativas, sejam elas rejeitadas ou aceitas.

Outro problema que surge ao utilizarmos passo adaptativo é quando temos uma partição do intervalo  $[t_0, T]$  dada por  $[t_0, t_1] \cup [t_1, t_2] \cup \dots \cup [t_{n-1}, t_n] \cup [t_n, t_{n+1}] \dots \cup [t_{n+k}, T]$  e conhecemos  $\Delta W$  e  $\Delta Z$  para cada subintervalo da partição, e vamos dar um passo de tamanho  $h$  que engloba vários subintervalos da partição que já temos, então suponhamos sem perda de generalidade que estejamos no tempo  $t_0$ , e que vamos dar um passo  $h$  que vai até o tempo  $t_n$ , sendo assim queremos encontrar  $\Delta W_{[t_0, t_n]}$  e  $\Delta Z_{[t_0, t_n]}$ . Assim, outra contribuição que fizemos ao algoritmo adaptativo é a proposição a seguir (mais precisamente o segundo item da proposição).

**Proposição 4.4.1.** *As equações (4.7) e (4.8) implicam em*

$$(i) \quad \Delta W_{[t_0, t_n]} = \sum_{i=1}^n \Delta W_{[t_{i-1}, t_i]}$$

$$(ii) \quad \Delta Z_{[t_0, t_n]} = \sum_{i=1}^n \Delta Z_{[t_{i-1}, t_i]} + \sum_{i=1}^n h_i \Delta W_{[t_0, t_{i-1}]}$$

*Demonstração.*

(i) vem diretamente de (4.7).

(ii) De (4.8) podemos deduzir que

$$\Delta Z_{[t_0, t_n]} = \Delta Z_{[t_0, t_{n-1}]} + \Delta Z_{[t_{n-1}, t_n]} + h_n \Delta W_{[t_0, t_{n-1}]} \quad (4.11)$$

Assim, temos que para  $n = 1$

$$\begin{aligned} \Delta Z_{[t_0, t_1]} &= \sum_{i=1}^1 \Delta Z_{[t_{i-1}, t_i]} + \sum_{i=1}^1 h_i \Delta W_{[t_0, t_{i-1}]} \\ &= \Delta Z_{[t_0, t_1]} + h_a \underbrace{\Delta W_{[t_0, t_0]}}_{=0} \\ &= \Delta Z_{[t_0, t_1]} \end{aligned}$$

Agora para  $n = 2$  temos

$$\begin{aligned} \Delta Z_{[t_0, t_2]} &= \sum_{i=1}^2 \Delta Z_{[t_{i-1}, t_i]} + \sum_{i=1}^2 h_i \Delta W_{[t_0, t_{i-1}]} \\ &= \Delta Z_{[t_0, t_1]} + \Delta Z_{[t_1, t_2]} + h_a \underbrace{\Delta W_{[t_0, t_0]}}_{=0} + h_b \Delta W_{[t_0, t_1]} \\ &= \Delta Z_{[t_0, t_1]} + \Delta Z_{[t_1, t_2]} + h_b \Delta W_{[t_0, t_1]} \end{aligned}$$

que satisfaz (4.11).

E por indução, temos que

$$\begin{aligned}
 \Delta Z_{[t_0, t_{n+1}]} &= \sum_{i=1}^{n+1} \Delta Z_{[t_{i-1}, t_i]} + \sum_{i=1}^{n+1} h_i \Delta W_{[t_0, t_{i-1}]} \\
 &= \sum_{i=1}^n \Delta Z_{[t_{i-1}, t_i]} + \Delta Z_{[t_n, t_{n+1}]} + \sum_{i=1}^n h_i \Delta W_{[t_0, t_{i-1}]} + h_{n+1} \Delta W_{[t_0, t_n]} \\
 &= \underbrace{\sum_{i=1}^n \Delta Z_{[t_{i-1}, t_i]} + \sum_{i=1}^n h_i \Delta W_{[t_0, t_{i-1}]}}_{=\Delta Z_{[t_0, t_n]}} + \Delta Z_{[t_n, t_{n+1}]} + h_{n+1} \Delta W_{[t_0, t_n]} \\
 &= \Delta Z_{[t_0, t_n]} + \Delta Z_{[t_n, t_{n+1}]} + h_{n+1} \Delta W_{[t_0, t_n]}
 \end{aligned}$$

□

## 4.5 Algoritmo para um passo de LL 115

Abaixo apresentamos o algoritmo que efetua um passo do método que construímos neste trabalho, utilizando o passo temporal adaptativo que foi descrito neste capítulo.

**Entrada:** tempo anterior ao passo  $t$ , o valor de  $X$  no tempo  $t$ , e o tamanho do passo  $h$ ;

**Saída:** valor de  $X$  no tempo  $t + h$ , e o valor de  $h$  para o próximo passo;

1. Reconstruir a trajetória Browniana de acordo com o passo  $h$ ;
2. Calcular a solução de  $X$  no tempo  $t + h$  através dos métodos LL 1 e LL 1.5;
3. **Se** o *erro* entre os resultados de LL 1 e LL 1.5 for satisfatório:
  - Utiliza-se do *erro* para determinar valor de  $h$  para o próximo passo;
  - Retorna  $h$ ;
  - Retorna a trajetória Browniana atualizada;
  - Retorna o valor de  $X$  calculado por LL 1.5 em  $t + h$ ;
4. **Senão**
  - Utiliza-se do *erro* para determinar o valor do novo  $h$ ;
  - Volta ao item [1](#);

## 4.6 Uma implementação ótima para os métodos LL 1 e 1.5

Embora os esquemas exponenciais construídos na seção 2.5.1 sejam satisfatórios, no nosso algoritmo adaptativo temos que calcular LL 1 e LL 1.5 para cada passo, e isto é custoso computacionalmente, principalmente por termos que calcular duas exponenciais de matrizes. O que veremos nesta seção é uma forma de reduzir o cálculo das duas exponenciais de matrizes a apenas uma exponencial, fazendo assim com que o trabalho computacional seja significativamente reduzido. O que é apresentado nesta seção é que faz com que o nosso algoritmo adaptativo seja viável, ou seja, é o que torna este trabalho relevante, pois não basta construirmos um método funcional, este método tem que ser computacionalmente eficiente e capaz de determinar uma aproximação da solução de EDEs com ruído aditivo com uma complexidade computacional próxima dos métodos com passo fixo. Desta forma, a implementação aninhada (ou “embedded”) que desenvolvemos e é apresentada nesta seção, é fundamental para este trabalho, e permite o desenvolvimento de outros métodos adaptativos eficientes para EDEs (e mesmo para EDOs) que utilizem esquemas exponenciais.

De (2.16) e (2.20) temos que os métodos LL 1 e LL 1.5 são obtidos (depois de fazer a aproximação de Taylor correspondente), resolvendo em cada intervalo  $[t_n, t_{n+1}]$ , uma EDE linear da forma

$$\begin{aligned} R'(t) &= (A_n(R(t) - R_n) + b_n(t - t_n) + d_n) dt + G(t) dW_t, \\ R(t_n) &= R_n. \end{aligned}$$

Onde  $A_n = f_x(t_n, X_n)$ ,  $d_n = f(t_n, X_n)$  coincidem para os dois LL, mas  $b_n$  é diferente para cada LL.

- Para LL 1  $b_n = f_t(t_n, X_n)$ , e
- para LL 1.5  $b_n = f_t(t_n, X_n) + \frac{1}{2} \sum_{j=1}^m (I_{d \times d} \otimes g_j^\top(t_n, X_n)) f_{xx}(t_n, X_n) g_j(t_n, X_n)$

Vamos representar a solução desta equação utilizando o método de variação de parâmetros (Lema 1.8.1), mas vamos fazer isto de modo que o procedimento possa ser usado tanto para LL 1 quanto para LL 1.5, se tornando menos custoso para a implementação do algoritmo adaptativo, sendo assim, seja

$$Z(t) = (R(t) - R_n, t - t_n, d_n, 1)^\top \in \mathbb{R}^{2d+2}$$

então

$$\begin{aligned} Z'(t) &= M \cdot Z(t)dt + \begin{pmatrix} G(t)^\top & 0 & 0 \end{pmatrix}^\top dW_t \\ Z(t_n) &= [\mathbf{0}_{1 \times d} \ 0 \ d_n \ 1]^\top, \end{aligned}$$

onde

$$M = \begin{pmatrix} A_n & b_n & I_{d \times d} & \mathbf{0}_{d \times 1} \\ \mathbf{0}_{1 \times d} & 0 & \mathbf{0}_{1 \times d} & 1 \\ \mathbf{0}_{d \times d} & \mathbf{0}_{d \times 1} & \mathbf{0}_{d \times d} & \mathbf{0}_{d \times 1} \\ \mathbf{0}_{1 \times d} & 0 & \mathbf{0}_{1 \times d} & 0 \end{pmatrix}.$$

A solução desta equação (pelo método de variação de parâmetros) é

$$Z(t) = e^{M(t-t_n)} Z(t_n) + \int_{t_n}^t e^{M(t-s)} \begin{pmatrix} G(s) \\ 0 \\ 0 \end{pmatrix} dW_s$$

*Observação.* Note que o vetor  $Z$  sem as suas últimas duas posições é uma solução  $X(t)$ . Ou seja, estamos interessados no primeiro componente de  $Z(t)$  que corresponde a  $R(t)$ .

Portanto, observando o primeiro componente de  $Z(t)$ , segue que  $R(t)$  pode ser calcu-

lado por

$$\begin{aligned}
 R(t) &= R_n + \begin{bmatrix} I_{d \times d} & \mathbf{0}_{d \times (d+2)} \end{bmatrix} e^{M(t-t_n)} [\mathbf{0}_{1 \times d} \ 0 \ d_n \ 1]^\top + \\
 &\quad + \begin{bmatrix} I_{d \times d} & \mathbf{0}_{d \times (d+2)} \end{bmatrix} \int_{t_n}^t e^{M(t-s)} \begin{pmatrix} G(s) \\ 0 \\ 0 \end{pmatrix} dW_s \\
 &= R_n + \begin{bmatrix} I_{d \times d} & \mathbf{0}_{d \times (d+2)} \end{bmatrix} e^{M(t-t_n)} [\mathbf{0}_{1 \times d} \ 0 \ d_n \ 1]^\top + \int_{t_n}^t e^{A_n(t-s)} G(s) dW_s
 \end{aligned}$$

Assim, em particular temos que

$$R(t_{n+1}) = R_n + \begin{bmatrix} I_{d \times d} & \mathbf{0}_{d \times (d+2)} \end{bmatrix} e^{Mh} [\mathbf{0}_{1 \times d} \ 0 \ d_n \ 1]^\top + \int_{t_n}^t e^{A_n(t-s)} G(s) dW_s \quad (4.12)$$

Evidentemente a matriz  $M$  é diferente para cada LL, (isto porque  $b_n$  difere de LL 1 para LL 1.5). De fato  $e^{Mh}$  é o único termo diferente no momento de avançar com cada um dos métodos.

Como estamos utilizando o Matlab para implementar os métodos propostos neste trabalho, é pertinente que expliquemos algumas coisas antes de continuar. A função do Matlab que calcula a exponencial de uma matriz é `expm.m`, e esta função utiliza para tal o algoritmo de Padé que está descrito em detalhes em [42]. Dito isto, o que vamos fazer é calcular  $e^{Mh}$  usando o algoritmo Padé, porém, ao invés de utilizar a função `expm.m`, iremos alterar o seu algoritmo de forma que irá nos permitir calcular “quase simultaneamente”  $e^{Mh}$  para ambos os métodos LL, calculando apenas uma exponencial em vez de duas.

A ideia principal é explorar a estrutura especial da matriz  $M$  e adaptar de forma conveniente a estratégia utilizada por Padé de modo que obtenhamos um algoritmo que seja computacionalmente mais eficiente de maneira realmente significativa.

Primeiramente, vamos explicar de forma sucinta o algoritmo de Padé e a sua estratégia de cálculo com a qual iremos trabalhar.

### 4.6.1 O algoritmo de Padé para calcular a exponencial de uma matriz

Dado  $e^C$ , a sua aproximação racional de Padé  $(p, q)$  é definida por

$$P_{p,q}(C) = [d_{p,q}(C)]^{-1} N_{p,q}(C),$$

onde

$$N_{p,q}(C) = \sum_{j=0}^p \frac{(p+q-j)!p!}{(p+q)!j!(p-j)!} C^j$$

e

$$d_{p,q}(C) = \sum_{j=0}^q \frac{(p+q-j)!q!}{(p+q)!j!(q-j)!} (-C)^j.$$

Aproximações diagonais (quando  $p = q$  são preferíveis, desde que  $P_{p,q}$  com  $p > q$  ( $p < q$ ) é menos preciso que  $P_{p,p}$  ( $P_{q,q}$ ), e  $P_{p,p}$  ( $P_{q,q}$ ) possuem o mesmo custo computacional. A partir de agora, para facilitar a escrita vamos denotar  $d_{q,q}(C)$ ,  $N_{q,q}(C)$ ,  $P_{q,q}(C)$  por  $d_q(C)$ ,  $N_q(C)$ ,  $P_q(C)$  respectivamente.

Nos voltando para a exponencial, temos que  $e^C$  só é bem aproximado por Padé nas proximidades da origem, ou seja, quando  $\|C\|$  é pequeno. Por isso  $e^C$  é aproximado por  $\left(P_q\left(\frac{C}{m}\right)\right)^m$  onde  $m$  é o menor inteiro que  $\left\|\frac{C}{m}\right\| < \frac{1}{2}$ . Para reduzir o número de multiplicações matriciais, vamos escolher  $m$  tal que este seja uma potência de dois. Então  $\left(P_q\left(\frac{C}{m}\right)\right)^m$  pode ser eficientemente calculado por repetidas potenciações (potências de 2).

O algoritmo de Padé <sup>1</sup> para computar  $e^C$  pode ser descrito da seguinte forma

1. Determine o menor inteiro  $k$  tal que  $\left\|\frac{C}{2^k}\right\| < \frac{1}{2}$
2. Calcule  $N_q\left(\frac{C}{2^k}\right)$  e  $P_q\left(\frac{C}{2^k}\right)$
3. Calcule  $P_q\left(\frac{C}{2^k}\right) = [d_q\left(\frac{C}{2^k}\right)]^{-1} N_q\left(\frac{C}{2^k}\right)$ , resolvendo o sistema  $d_q\left(\frac{C}{2^k}\right) P_q\left(\frac{C}{2^k}\right) = N_q\left(\frac{C}{2^k}\right)$  (usando eliminação Gaussiana ou outro método que seja conveniente)

---

<sup>1</sup>O algoritmo de Padé utiliza uma estratégia conhecida em inglês como “scaling-squaring”.

4. Calcule  $[P_q(\frac{C}{2^k})]^{2^k}$  através do quadrado de  $P_q(\frac{C}{2^k})$ ,  $k$  vezes.

#### 4.6.2 O algoritmo de Padé adaptado para computar $e^{Mh}$

Agora, aplicamos o método de Padé  $(q, q)$  para computar a exponencial da matriz que nos interessa,  $C = Mh$ .

Seja  $k$  o menor inteiro tal que  $\left\| \frac{C}{2^k} \right\| < \frac{1}{2}$  e os coeficientes  $\alpha_j = \frac{(2q-j)!q!}{(2q)!j!(q-j)!}$  ( $j = 0, \dots, q$ ), então

$$\left( \frac{C}{2^k} \right) = \begin{pmatrix} \frac{A_n}{2^k} & \frac{b_n}{2^k} & \frac{I_{d \times d}}{2^k} & \mathbf{0}_{d \times 1} \\ \mathbf{0}_{1 \times d} & 0 & \mathbf{0}_{1 \times d} & \frac{1}{2^k} \\ \mathbf{0}_{d \times d} & \mathbf{0}_{d \times 1} & \mathbf{0}_{d \times d} & \mathbf{0}_{d \times 1} \\ \mathbf{0}_{1 \times d} & 0 & \mathbf{0}_{1 \times d} & 0 \end{pmatrix} h,$$

$$\left( \frac{C}{2^k} \right)^2 = \begin{pmatrix} \frac{A_n^2}{(2^k)^2} & \frac{A_n b_n}{(2^k)^2} & \frac{A_n}{(2^k)^2} & \frac{b_n}{(2^k)^2} \\ \mathbf{0}_{1 \times d} & 0 & \mathbf{0}_{1 \times d} & 0 \\ \mathbf{0}_{d \times d} & \mathbf{0}_{d \times 1} & \mathbf{0}_{d \times d} & \mathbf{0}_{d \times 1} \\ \mathbf{0}_{1 \times d} & 0 & \mathbf{0}_{1 \times d} & 0 \end{pmatrix} h^2, \dots,$$

e por indução, temos que para todo  $j \geq 2$ ,  $j \in \mathbb{N}$

$$\left( \frac{C}{2^k} \right)^j = \begin{pmatrix} \frac{A_n^j}{(2^k)^j} & \frac{A_n^{j-1} b_n}{(2^k)^j} & \frac{A_n^{j-1}}{(2^k)^j} & \frac{A_n^{j-2} b_n}{(2^k)^j} \\ \mathbf{0}_{1 \times d} & 0 & \mathbf{0}_{1 \times d} & 0 \\ \mathbf{0}_{d \times d} & \mathbf{0}_{d \times 1} & \mathbf{0}_{d \times d} & \mathbf{0}_{d \times 1} \\ \mathbf{0}_{1 \times d} & 0 & \mathbf{0}_{1 \times d} & 0 \end{pmatrix} h^j.$$

Para facilitar a escrita, vamos denotar  $A = A_n h$  e  $b = b_n h$ . E consequentemente,

$$N_q \left( \frac{C}{2^k} \right) = \sum_{j=0}^q \alpha_j \left( \frac{C}{2^k} \right)^j$$

$$= \begin{pmatrix} I + A \left( \alpha_1 \left( \frac{I}{2^k} \right) + AS \right) & \left( \frac{\alpha_1}{2^k} I + AS \right) b & \left( \frac{\alpha_1}{2^k} I + AS \right) & Sb \\ \mathbf{0}_{1 \times d} & 1 & \mathbf{0}_{1 \times d} & \frac{\alpha_1}{2^k} h \\ \mathbf{0}_{d \times d} & \mathbf{0}_{d \times 1} & I_{d \times d} & \mathbf{0}_{d \times 1} \\ \mathbf{0}_{1 \times d} & 0 & \mathbf{0}_{1 \times d} & 1 \end{pmatrix},$$

onde,

$$S = \left( \alpha_2 \frac{I}{(2^k)^2} + \alpha_3 \frac{A}{(2^k)^3} + \dots + \alpha_q \frac{(A)^{q-2}}{(2^k)^q} \right).$$

Analogamente temos que,

$$d_q \left( \frac{C}{2^k} \right) = \sum_{j=0}^q \alpha_j \left( -\frac{C}{2^k} \right)^j$$

$$= \begin{pmatrix} I + A \left( -\frac{\alpha_1}{2^k} I + A\bar{S} \right) & \left( -\frac{\alpha_1}{2^k} I + A\bar{S} \right) b & \left( -\frac{\alpha_1}{2^k} I + A\bar{S} \right) & \bar{S}b \\ \mathbf{0}_{1 \times d} & 1 & \mathbf{0}_{1 \times d} & -\frac{\alpha_1}{2^k} \\ \mathbf{0}_{d \times d} & \mathbf{0}_{d \times 1} & I_{d \times d} & \mathbf{0}_{d \times 1} \\ \mathbf{0}_{1 \times d} & 0 & \mathbf{0}_{1 \times d} & 1 \end{pmatrix},$$

onde,

$$\bar{S} = \left( \alpha_2 \frac{I}{(2^k)^2} - \alpha_3 \frac{A}{(2^k)^3} + \dots + (-1)^{q-2} \alpha_q \frac{(A)^{q-2}}{(2^k)^q} \right).$$

Agora vamos calcular  $P_q \left( \frac{C}{2^k} \right) = \left[ d_q \left( \frac{C}{2^k} \right) \right]^{-1} N_q \left( \frac{C}{2^k} \right)$  resolvendo o sistema linear  $d_q \left( \frac{C}{2^k} \right) P_q \left( \frac{C}{2^k} \right) = N_q \left( \frac{C}{2^k} \right)$ .

Assim não é difícil de ver que  $P_q\left(\frac{C}{2^k}\right)$  fica da forma

$$P_q\left(\frac{C}{2^k}\right) = \begin{pmatrix} U1 & U2 & U3 & U4 \\ \mathbf{0}_{1 \times d} & 1 & \mathbf{0}_{1 \times d} & \frac{\alpha_1}{2^{k-1}} \\ \mathbf{0}_{d \times d} & \mathbf{0}_{d \times 1} & I_{d \times d} & \mathbf{0}_{d \times 1} \\ \mathbf{0}_{1 \times d} & 0 & \mathbf{0}_{1 \times d} & 1 \end{pmatrix}$$

onde as matrizes  $U1$ ,  $U3$  e os vetores  $U2$ ,  $U4$  satisfazem os sistemas de equações lineares

$$\begin{aligned} \left[ I + A \left( -\frac{\alpha_1}{2^k} I + A\bar{S} \right) \right] U1 &= \left[ I + A \left( \frac{\alpha_1}{2^k} I + AS \right) \right] \\ \left[ I + A \left( -\frac{\alpha_1}{2^k} I + A\bar{S} \right) \right] U2 &= \left[ \left( \frac{\alpha_1}{2^k} I + AS \right) b \right] - \left[ \left( -\frac{\alpha_1}{2^k} I + A\bar{S} \right) b \right] \\ \left[ I + A \left( -\frac{\alpha_1}{2^k} I + A\bar{S} \right) \right] U3 &= \left( \frac{\alpha_1}{2^k} I + AS \right) - \left( -\frac{\alpha_1}{2^k} I + A\bar{S} \right) \\ \left[ I + A \left( -\frac{\alpha_1}{2^k} I + A\bar{S} \right) \right] U4 &= \left( S - \left[ \left( -\frac{\alpha_1}{2^k} I + A\bar{S} \right) \frac{\alpha_1}{2^{k-1}} \right] - \bar{S} \right) b \end{aligned}$$

Desta forma, desde que a matriz fundamental de cada sistema acima seja a mesma para ambos os métodos LL, podemos explorar isso, e fazer a decomposição  $LU$  para  $\left[ I + A \left( -\frac{\alpha_1}{2^k} I + A\bar{S} \right) \right]$  (essa matriz não depende do método LL que estamos usando) e então usar o procedimento padrão para obtenção da solução definitiva do sistema linear de equações (veja [44]).

Vamos listar abaixo alguns fatos muito importantes:

- $U1$ ,  $U3$  são os mesmos para os dois LL (1 e 1.5)
- $U2 = (U3)b$
- $U4 = (U5)b$ , onde  $U5$  é a solução do sistema

$$\left[ I + A \left( -\frac{\alpha_1}{2^k} I + A\bar{S} \right) \right] U5 = \left( S - \left[ \left( -\frac{\alpha_1}{2^k} I + A\bar{S} \right) \frac{\alpha_1}{2^{k-1}} \right] - \bar{S} \right)$$

(e portanto  $U5$  também não depende do método LL utilizado)

Uma vez que tenhamos obtido  $U1, U2, U3, U4$ , nos resta calcular  $\left(P_q \left(\frac{C}{2^k}\right)\right)^m$ , onde  $m = 2^k$ , para concluir o método de Padé. Não é difícil provar por indução que

$$\begin{aligned} \left(P_q \left(\frac{C}{2^k}\right)\right)^m &= \begin{pmatrix} U1 & U2 & U3 & U4 \\ \mathbf{0}_{1 \times d} & 1 & \mathbf{0}_{1 \times d} & \frac{\alpha_1}{2^{k-1}} \\ \mathbf{0}_{d \times d} & \mathbf{0}_{d \times 1} & I_{d \times d} & \mathbf{0}_{d \times 1} \\ \mathbf{0}_{1 \times d} & 0 & \mathbf{0}_{1 \times d} & 1 \end{pmatrix}^m \\ &= \begin{pmatrix} (U1)^m & \left(\sum_{i=0}^{m-1} (U1)^i\right) U2 & \left(\sum_{i=0}^{m-1} (U1)^i\right) U3 & (*) \\ 0_{1 \times d} & 1 & 0_{1 \times d} & \frac{\alpha_1}{2^{k-1}} \\ 0_{d \times d} & 0_{d \times 1} & I_{d \times d} & 0_{d \times 1} \\ 0_{1 \times d} & 0 & 0_{1 \times d} & 1 \end{pmatrix}. \end{aligned}$$

onde

$$(*) = \left(\sum_{i=0}^{m-1} (U1)^i\right) U4 + \frac{\alpha_1}{2^{k-1}} \left(\sum_{i=0}^{m-2} (m-1-i) (U1)^i\right) U2$$

Então, substituindo a aproximação de Padé para  $e^{Mh}$  em (4.12), temos que

$$\begin{aligned} R_{n+1} &= R_n + \begin{bmatrix} I_{d \times d} & \mathbf{0}_{d \times (d+2)} \end{bmatrix} \left(P_q \left(\frac{C}{2^k}\right)\right)^m \begin{bmatrix} \mathbf{0}_{1 \times d} & 0 & d_n & 1 \end{bmatrix}^\top + \zeta_n \\ &= L R_n + Qb + \zeta_n, \end{aligned} \quad (4.13)$$

onde  $\zeta_n$  é a aproximação da integral estocástica (isto é feito independentemente para cada método LL), onde

$$L = \left(\sum_{i=0}^{m-1} (U1)^i\right) (U3),$$

e

$$Q = \left[ \left(\sum_{i=0}^{m-1} (U1)^i\right) (U5) + \frac{\alpha_1}{2^{k-1}} \left(\sum_{i=0}^{m-2} (m-1-i) (U1)^i\right) (U3) \right].$$

Desta forma, não precisamos calcular uma exponencial matricial para cada método LL, basta calcular  $L$  e  $Q$  (que são iguais para LL 1 e LL 1.5) e então cada método LL avança de  $R_n$  para  $R_{n+1}$  usando (4.13). Assim, a diferença computacional entre os dois métodos é simplesmente a multiplicação de uma matriz  $Q$  por um vetor  $b$ . E por fim, se um passo é rejeitado na execução do algoritmo adaptativo, e vamos tentar efetuar outro passo partindo do mesmo instante  $t$  da tentativa anterior, temos que a matriz  $A_n$  não muda, logo podemos reutilizar quase tudo que foi calculado, reduzindo consideravelmente o custo computacional final.

## Capítulo 5

# Simulações

Neste capítulo, iremos através de exemplos aplicar os métodos estudados neste trabalho, com ênfase no método adaptativo LL 115, de forma a poder visualizar o seu funcionamento. Em cada exemplo vamos apontar diferentes aspectos sobre as características dos métodos e das simulações, mas sem a pretensão de fazer um estudo aprofundado sobre os modelos.

**Exemplo 5.0.1.** Seja a equação diferencial estocástica com ruído aditivo

$$dX(t) = -t^2 X(t)dt + \frac{3}{2} \left( \frac{e^{-(t^3-t_0^3)/3}}{t+1} \right) dW(t), \quad X(0) = 1$$

no intervalo  $t \in [0, 10]$ .

De [10] temos que esta EDE possui solução exata

$$X(t) = e^{\frac{t^3-t_0^3}{3}} \left( X(t_0) + \frac{3}{2} \int_{t_0}^t \frac{1}{s+1} dW(s) \right).$$

Para este exemplo vamos fazer dois experimentos conforme descritos a seguir:

*Experimento I:* Comparar o resultado da solução numérica através do método adaptativo que apresentamos (LL 115), com a solução obtida através do método LL 1.5 com passo fixo efetuando o mesmo número de passos utilizados pelo anterior.

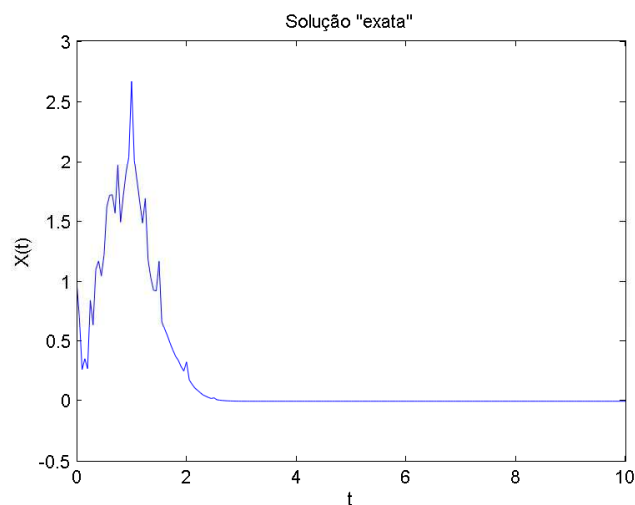


Figura 5.1: Gráfico da solução “exata” sobre a mesma trajetória Browniana das simulações apresentadas nas figuras 5.2 e 5.3.

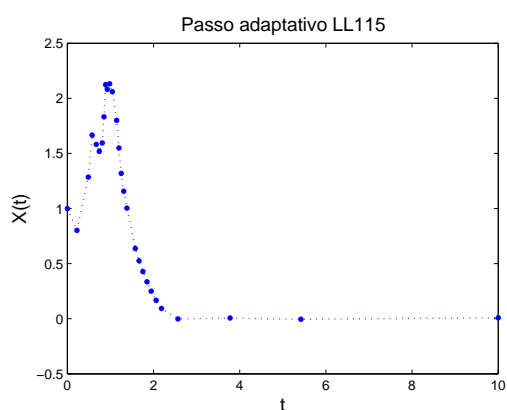


Figura 5.2: Simulação do exemplo 5.0.1, utilizando o LL 115 (adaptativo) com 27 passos.

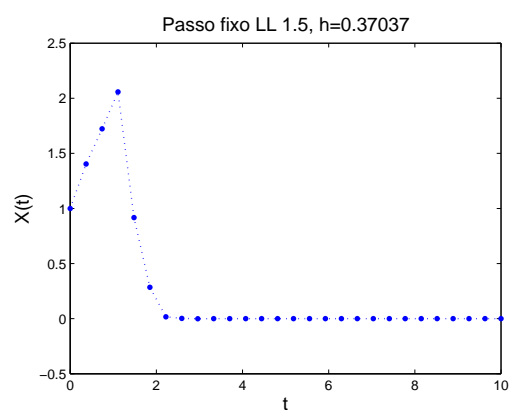


Figura 5.3: Simulação do exemplo 5.0.1, utilizando o LL 1.5 (passo fixo) com 27 passos.

A figura 5.2 mostra a simulação utilizando passo adaptativo com tolerância  $tol = 0,001$ , e tentativa de passo inicial de  $h = \frac{2}{9}$ , nesta execução foram efetuados 27 passos para atingir a solução final, e 25 passos descartados. O tamanho de passo mínimo e máximo efetuados pelo algoritmo adaptativo foram  $h_{min} = 0,0382$  e  $h_{max} = 4,5778$ . E na figura 5.3, com esquema exponencial LL de ordem 1.5 com passo fixo, também com 27 passos, sobre uma trajetória Browniana mantida coerente com a que o método anterior

percorreu (conforme mostramos na seção 4.4). Podemos observar que na região de “pico” o método adaptativo nos fornece um resultado muito mais detalhado.

Comparando os gráficos 5.2 e 5.3, podemos notar que no primeiro resultado, o método dá mais atenção ao intervalo da solução onde há mais variações (quando  $t \in [0, 2]$ ) e avança mais rapidamente, efetuando passos maiores depois do instante  $t = 2$ . Ou seja, o algoritmo adaptativo se comportou como era esperado. Comparando os resultados das duas simulações com a solução exata, temos que o LL 115 (adaptativo) nos retorna um erro local máximo de 0,1123, e o LL 1.5 (passo fixo) retorna um erro local máximo de 0,3025, e a ocorrência deste erro se dá no tempo  $t = 1,4815$ , ou seja, dentro do intervalo onde a equação tem um comportamento mais oscilatório (quando  $t \in [0, 2]$ , onde o método adaptativo avança mais lentamente e fornece uma solução mais detalhada da dinâmica do problema.

*Experimento II:* Analisar o comportamento do método de Milstein quando este efetua passos com mesmo tamanho que o método adaptativo, conforme mostrado na figura 5.2. E quando utiliza passo fixo igual a  $h_{min}$  do método adaptativo, bem como identificar qual tamanho de passo é suficiente para que o método de Milstein não tenha comportamento explosivo.

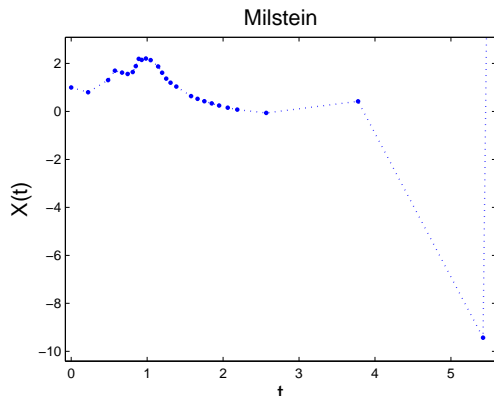


Figura 5.4: Simulação do exemplo 5.0.1, utilizando o método de Milstein sobre a trajetória adaptativa da execução da figura 5.2.

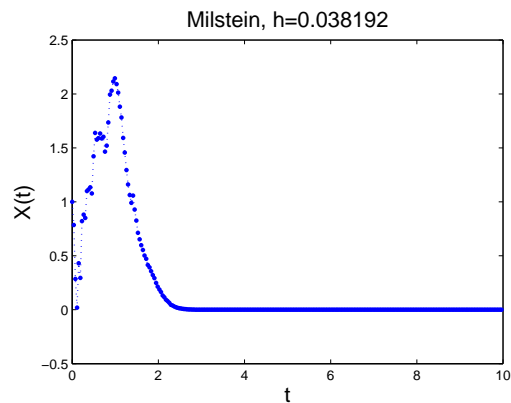


Figura 5.5: Simulação do exemplo 5.0.1, utilizando o método de Milstein com tamanho de passo  $h_{min} = 0,0382$ .

Na figura 5.4 vemos que a solução pelo método de Milstein explode quando passos maiores são efetuados. Mas com tamanho de passo reduzido conforme a figura 5.5, o método consegue reproduzir os picos. Agora, aumentando gradualmente o tamanho de passo utilizado obtemos

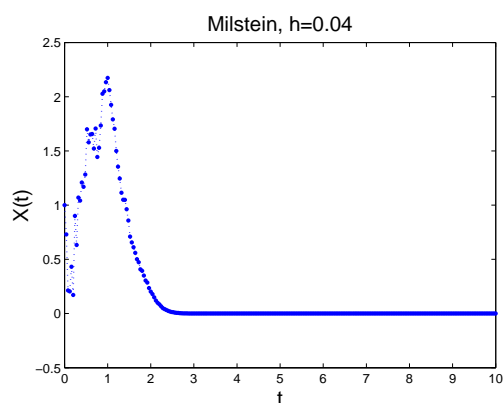


Figura 5.6: Simulação do exemplo 5.0.1, utilizando o método de Milstein com  $h = 0,04$ .

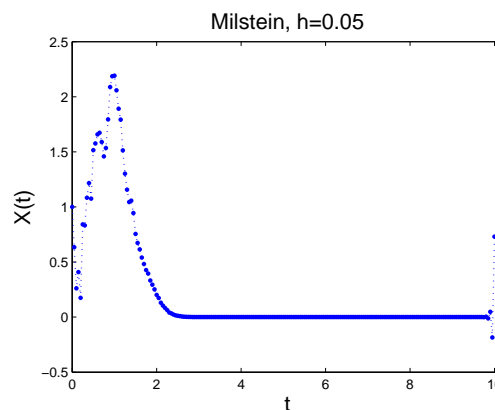


Figura 5.7: Simulação do exemplo 5.0.1, utilizando o método de Milstein com  $h = 0,05$ .

Como podemos ver, o método de Milstein precisa de passos muito pequenos para nos fornecer soluções numéricas satisfatórias. Pois, com passo  $h = 0,05$  (figura 5.7) a solução já explode no final da trajetória, e falha em retornar uma solução para  $X(10)$ .

**Exemplo 5.0.2** (Pêndulo não amortecido). Este exemplo corresponde a um modelo de um pendulo não amortecido com perturbações causadas por forças aleatórias descrito em [45]. Seja  $X(t) = (x_1(t), x_2(t))^T$ , tal que

$$dX(t) = \begin{pmatrix} x_2(t) \\ -\sin(x_1(t)) \end{pmatrix} dt + \begin{pmatrix} 0 \\ \sigma \end{pmatrix} dW(t), \quad X(0) = \begin{pmatrix} 1,4 \\ 0 \end{pmatrix}$$

no intervalo  $t \in [0, 50]$ , com  $\sigma = 0,1$ .

O objetivo com este exemplo é apenas mostrar o comportamento do método adapta-

tivo com um oscilador, pois passos exageradamente grandes geram distorções que levam à perda das características oscilatórias do problema.

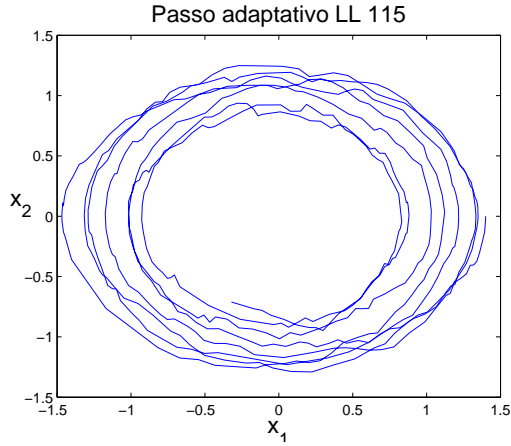


Figura 5.8: Simulação do exemplo 5.0.2, utilizando o método adaptativo com  $tol = (0,001; 0,001)$ .

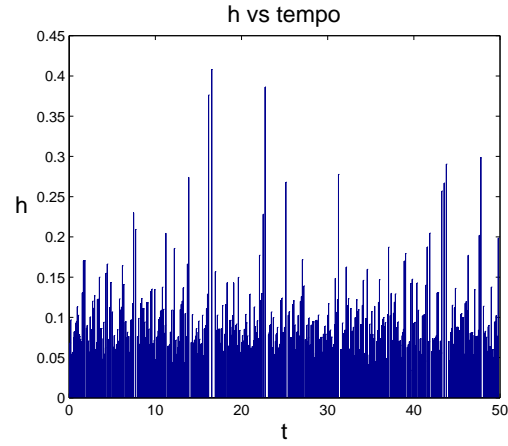


Figura 5.9: Tamanho de passo  $h$  executado em cada tempo  $t$ .

Na figura 5.8 temos o resultado da simulação utilizando o método adaptativo LL 115, e como estamos traçando o plano de fase  $(x_1, x_2)$ , e não fica visualmente bem definido a utilização de passos de tamanhos variados, apresentamos na figura 5.9 um gráfico que mostra o tamanho do passo que foi executado em cada instante de tempo, e podemos ver que há grande variação no tamanho dos passos executados. Nesta simulação utilizamos  $tol = (0,001; 0,001)$ , e o método precisou de 559 passos para chegar até o fim da simulação, e foram rejeitados outros 3647 passos, número elevado, porém, o método adaptativo foi bem sucedido em replicar a dinâmica do sistema original. Este exemplo também foi estudado em [46], onde foram feitas simulações utilizando outros métodos/esquemas.

**Exemplo 5.0.3** (Osciladores de Van der Pol acoplados). Este exemplo representa o modelo de dois osciladores de Van der Pol acoplados,

$$dX(t) = d \begin{pmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \end{pmatrix} = \begin{pmatrix} x_3(t) \\ x_4(t) \\ \frac{1}{2}x_3(t)(1 - x_1^2(t)) - x_1 + \frac{1}{10}x_2 \\ 5x_4(t)(1 - x_2^2(t)) - x_2(t) + \frac{1}{10}x_1 \end{pmatrix} dt + \begin{pmatrix} 0 \\ 0 \\ \sigma_1 \\ \sigma_2 \end{pmatrix} dW(t),$$

$$X(0) = \begin{pmatrix} 1 & 1 & 0 & 0 \end{pmatrix}^\top$$

no intervalo  $t \in [0, 20]$ , e com  $\sigma_1 = 1,5$  e  $\sigma_2 = 2,5$ .

Com este exemplo, pretendemos apresentar as diferenças causadas pela variação da tolerância utilizada pelo algoritmo adaptativo. Então, foram feitas simulações com tolerâncias  $tol_1 = 0,5 \times (1; 1; 1; 1)$ ,  $tol_2 = 0,1 \times (1; 1; 1; 1)$ ,  $tol_3 = 0,01 \times (1; 1; 1; 1)$  e  $tol_4 = 0,001 \times (1; 1; 1; 1)$ . E os resultados podem ser vistos a seguir.

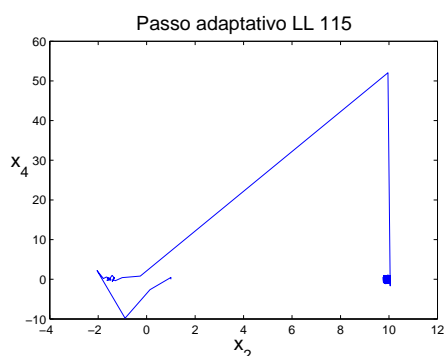


Figura 5.10: Simulação do exemplo 5.0.3, utilizando o método adaptativo com  $tol = (0,5; 0,5; 0,5; 0,5)$ .

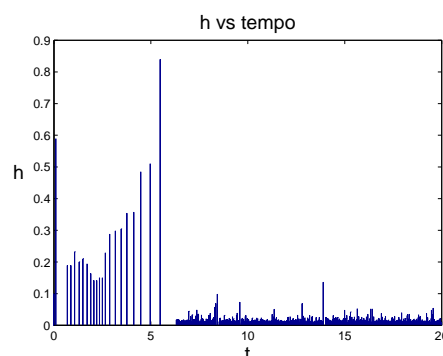


Figura 5.11: Tamanho de passo  $h$  executado em cada tempo  $t$ .

Como foi utilizada a mesma tolerância para cada componente, por simplicidade vamos chamar  $tol_1 = 0,5 \times (1; 1; 1; 1)$  de  $tol_1 = 0,5$ , então para  $tol_1 = 0,5$ , como pode ser visto nas figuras 5.10 e 5.11 a dinâmica do problema se perde por completo, e por consequência dos grandes passos gerados no início da trajetória, grandes erros começam a ser gerados, o que obriga a uma redução drástica nos tamanhos de passo.

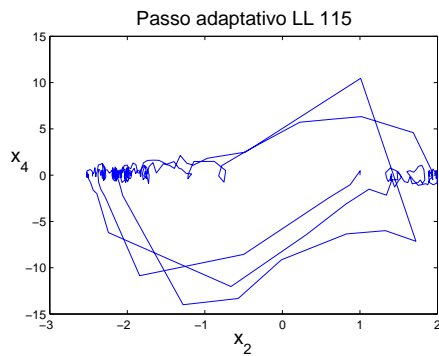


Figura 5.12: Simulação do exemplo 5.0.3, utilizando o método adaptativo com  $tol = (0,1; 0,1; 0,1; 0,1)$ .

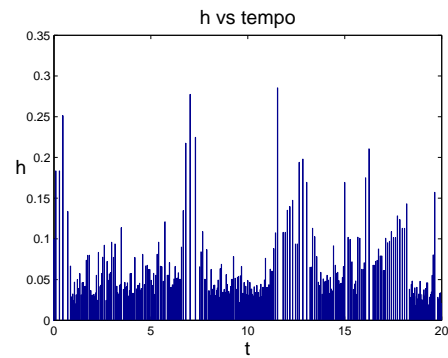


Figura 5.13: Tamanho de passo  $h$  executado em cada tempo  $t$ .

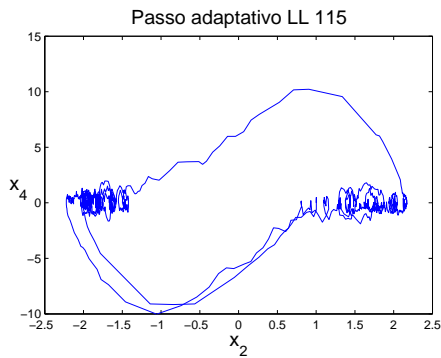


Figura 5.14: Simulação do exemplo 5.0.3, utilizando o método adaptativo com  $tol = (0,01; 0,01; 0,01; 0,01)$ .

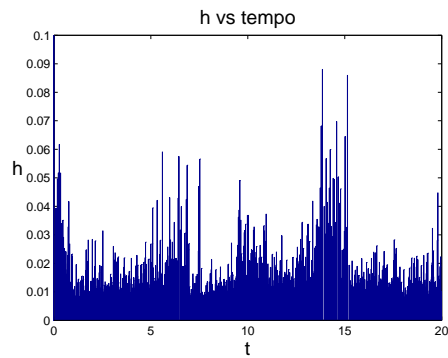


Figura 5.15: Tamanho de passo  $h$  executado em cada tempo  $t$ .

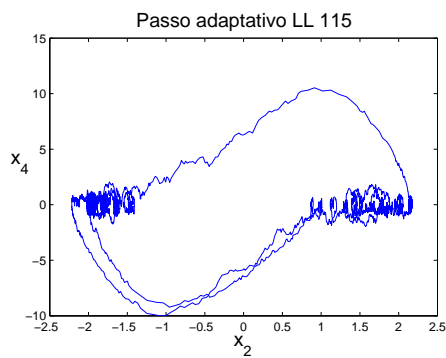


Figura 5.16: Simulação do exemplo 5.0.3, utilizando o método adaptativo com  $tol = (0,001; 0,001; 0,001; 0,001)$ .

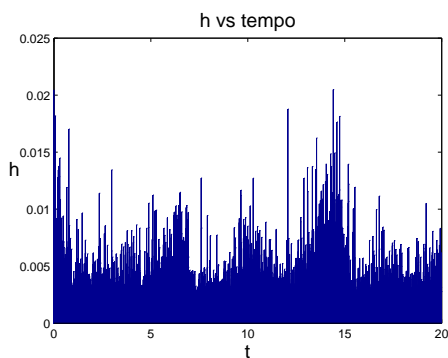


Figura 5.17: Tamanho de passo  $h$  executado em cada tempo  $t$ .

$\text{tol} \times (1, 1, 1, 1)$	# passos aceitos	# passos rejeitados	maior passo efetuado	menor passo efetuado
0,5	973	7299	0,8395	0,0055
0,1	348	2995	0,2856	0,0175
0,01	1419	10847	0,1000	0,0040
0,001	6675	53170	0,0205	0,0007

Tabela 5.1: Resultados

Podemos ver na figura 5.12 que para  $\text{tol}_2 = 0.1$  a forma do oscilador começa a aparecer, porém ainda não resulta na dinâmica esperada. O curioso é a redução no número de passos efetuados e rejeitados em relação ao obtido com  $\text{tol}_1 = 0,5$ , quando utilizamos  $\text{tol}_1$ , o método avança com passos grandes demais no início da trajetória, o que gera um erro muito grande, e faz com que mais a frente a diferença entre LL 1 e LL 1.5 seja tão grande, que mesmo trabalhando com uma tolerância relativamente grande, o algoritmo impõe uma redução drástica no tamanho dos passos para manter a solução dentro da tolerância.

Agora na figura 5.14 temos a dinâmica esperada do problema, e podemos supor que a tolerância  $\text{tol}_3 = 0,01$  seja suficiente para estudarmos este exemplo.

E por fim, com tolerância  $\text{tol}_4 = 0,001$ , a figura 5.16, no que tange à dinâmica do problema, não mostra um ganho significativo em relação ao resultado obtido com  $\text{tol}_3 = 0,01$ , porém é um resultado que mostra as oscilações de forma mais detalhada.

Outras simulações para este exemplo podem ser encontradas em [46].

**Exemplo 5.0.4.** Seja  $X(t) = (x_1(t), x_2(t))^T$  a solução da EDE bidimensional

$$dX(t) = d \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} = \begin{pmatrix} 0 \\ \frac{x_1^2(t)}{2} \end{pmatrix} dt + \begin{pmatrix} 1 \\ 0 \end{pmatrix} dW(t), \quad X(0) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

no intervalo  $t \in [0, 1]$ .

De [13] sabemos que dada uma função  $p(X(t)) = e^{-x_2(t)}$  o valor esperado de  $p(X(1))$

é dado por

$$\mathbb{E}(p(X(1))) = \sqrt{\frac{2e}{1+e^2}} = 0,805018...$$

Neste exemplo, iremos mostrar o uso do método adaptativo que propomos para determinar a aproximação de um funcional da solução e não aproximações de trajetórias como fizemos até aqui. Então vamos aproveitar que conhecemos o valor esperado de  $p(X(1))$ , e efetuar simulações de Monte Carlo para verificar a acurácia do nosso método. Para tal foram executadas 10.000 simulações para o intervalo  $t \in [0, 1]$  com o nosso método adaptativo LL 115 com uma tolerância  $tol = (0,0001; 0,0001)$ , cada uma sobre uma nova trajetória Browniana gerada aleatoriamente, sendo assim, se  $p_i$  é o resultado da  $i$ -ésima simulação, estimamos o seu valor esperado como

$$\bar{\mathbb{E}}(p(X(1))) = \frac{1}{10^4} \sum_{i=1}^{10^4} p_i$$

E encontramos  $\bar{\mathbb{E}}(p(X(1))) = 0,805642...$ , com um desvio padrão amostral  $sd_a = 0,174831...$ , e comparando com o valor esperado real, obtemos uma distância menor que 0,000625 entre o estimado e o real.



## Capítulo 6

# Considerações Finais

Não encontramos na literatura sobre equações diferenciais estocásticas (incluindo livros, artigos, dissertações e teses) a utilização de métodos LL com passo variável ou mesmo de métodos adaptativos exponenciais em geral. Isto faz com que o método para aproximação numérica de soluções de EDEs com ruído aditivo aqui proposto traga, por si só, um aspecto de originalidade a este trabalho. Ademais, pudemos conferir no capítulo 4 uma implementação que incorpora o cálculo do método LL 1.5 ao cálculo de LL 1, reaproveitando resultados, e gerando uma implementação “embedding” para o método adaptativo, reduzindo assim, não só o esforço computacional, como também o uso de memória para a execução do método adaptativo LL 115. Ressaltamos que também não encontramos referências desta implementação na literatura. Além disso, no capítulo 3 fizemos um estudo de estabilidade de EDEs com um exemplo que traz uma cognição visual do resultado obtido ao utilizar um método instável para simular uma EDE rígida, e a importância de trabalharmos com métodos A-estáveis; e no capítulo 5 pudemos ver que:

- com o exemplo 5.0.1 que o método adaptativo LL 115 fornece soluções com erros locais menores que LL 1.5 com passo fixo, e ainda tem sucesso onde o método de Milstein falha;

- com o exemplo 5.0.2 que LL 115 consegue preservar a dinâmica do problema;
- com o exemplo 5.0.3 os efeitos que a alteração da tolerância utilizada por LL 115 causa nas soluções obtidas;
- e com o exemplo 5.0.4 que o método consegue obter ótimas estimativas quando usado em conjunto com o método de Monte Carlo.

Durante os testes e as simulações citadas acima, surgiram algumas questões às quais ainda podemos aperfeiçoar no algoritmo adaptativo, que listamos a seguir junto com outras possibilidades de continuação desta pesquisa.

## 6.1 Trabalhos futuros

Ainda há questões a serem aperfeiçoadas no algoritmo adaptativo, como

- estudar a influência do tamanho do primeiro passo efetuado na execução do algoritmo adaptativo;
- determinar um tamanho de passo inicial  $h_1$  ótimo;
- encontrar uma forma melhor para determinação de um novo passo, talvez tentando controlar a variação do tamanho de um passo para o próximo.

Além das questões que envolvem diretamente o algoritmo adaptativo, temos que o método/esquema que apresentamos neste trabalho, mesmo possuindo boas propriedades de estabilidade, não possui alta ordem de convergência. Então, temos como objetivo futuro desenvolver um algoritmo adaptativo com uso de métodos LL de ordem superior, métodos estes que foram propostos em [28] e [14].

Também há grande interesse em estudar como um algoritmo adaptativo se comporta para o cálculo de soluções fracas de EDEs, tema que devemos tratar em uma pesquisa futura.

E por fim, muito nos interessa estudar esquemas exponenciais para EDEs com ruído multiplicativo, ou seja, quando a função de difusão  $g$  depende de  $t$  e de  $X(t)$ , já que estas equações tem grande relevância para a modelagem de diversos sistemas. Dito isto, temos o objetivo de construir um método adaptativo com esquema exponencial para este tipo de EDE.

# Referências Bibliográficas

- [1] Lawrence C. Evans. *An introduction to stochastic differential equations*, volume 82. American Mathematical Soc., 2013.
- [2] Bernt Øksendal. *Stochastic differential equations*. Springer, 2003.
- [3] David F. Griffiths e Desmond J. Higham. *Numerical methods for ordinary differential equations*. Springer, 2010.
- [4] Jeffrey S. Rosenthal. *A First Look at Rigorous Probability Theory*. World Scientific Singapore, 2nd edition, 2006.
- [5] Boris V. Gnedenko. *Theory of Probability*. MIR Publishers, 1978.
- [6] Kiyoshi Itô. *On Stochastic Differential Equations*. American Mathematical Soc., 1951.
- [7] Peter Eris Kloeden, Eckhard Platen, e Henri Schurz. *Numerical solution of SDE through computer experiments*, volume 1. Springer Science & Business Media, 1994.
- [8] Peter Eris Kloeden e Eckhard Platen. *Numerical solution of stochastic differential equations*, volume 23. Springer Science & Business Media, 1995.
- [9] Edward Allen. *Modeling with Itô stochastic differential equations*, volume 22. Springer Science & Business Media, 2007.

- [10] R. Biscay, J. C. Jimenez, J.J. Riera, e P. A. Valdes. Local linearization method for the numerical solution of stochastic differential equations. *Annals of the Institute of Statistical Mathematics*, 48(4):631–644, 1996.
- [11] J. C. Jimenez, I. Shoji, e T. Ozaki. Simulation of stochastic differential equations through the local linearization method. a comparative study. *Journal of Statistical Physics*, 94(3-4):587–602, 1999.
- [12] Juan Carlos Jimenez. A simple algebraic expression to evaluate the local linearization schemes for stochastic differential equations. *Applied Mathematics Letters*, 15(6):775–780, 2002.
- [13] F. Carbonell, J. C. Jimenez, e R. J. Biscay. Weak local linear discretizations for stochastic differential equations: convergence and numerical schemes. *Journal of computational and applied mathematics*, 197(2):578–596, 2006.
- [14] Hugo De la Cruz Cancino, Rolando J Biscay, Juan C Jimenez, F Carbonell, and T Ozaki. High order local linearization methods: An approach for constructing a-stable explicit schemes for stochastic differential equations with additive noise. *BIT Numerical Mathematics*, 50(3):509–539, 2010.
- [15] Juan Carlos Jimenez. Strong local linearization methods for the numerical integration of stochastic differential equations with additive noise: An overview. Technical report, Abdus Salam International Centre for Theoretical Physics, Trieste (Italy), 2010.
- [16] Juan Carlos Jimenez e Hugo de la Cruz Cancino. Convergence rate of strong local linearization schemes for stochastic differential equations with additive noise. *BIT Numerical Mathematics*, 52(2):357–382, 2012.
- [17] Juan Carlos Jimenez, Alina Sotolongo, e Jose Miguel Sanchez-Bornot. Locally

- linearized runge kutta method of dormand and prince. *Applied Mathematics and Computation*, 247:589–606, 2014.
- [18] Djairo G. Figueiredo e Aloisio F. Neves. Equações diferenciais aplicadas; coleção matemática universitária, impa. *Rio de Janeiro*, 2005.
- [19] Carlos A. Braumann. *Uma Introdução às Equações Diferenciais Estocásticas e Aplicações*. Sociedade Portuguesa de Estatística, 2005.
- [20] Thomas C. Gard. *Introduction to stochastic differential equations*. M. Dekker, 1988.
- [21] Pamela Marion Burrage. *Runge-Kutta methods for stochastic differential equations*. PhD thesis, The University of Queensland Brisbane, 1999.
- [22] Philip Protter. *Stochastic Integration and Differential Equations, A New Approach (Applications of Mathematics 21)*. Springer-Verlag, 1990.
- [23] Peter Eris Kloeden. The systematic derivation of higher order numerical schemes for stochastic differential equations. *Milan Journal of Mathematics*, 70(1):187–207, 2002.
- [24] Eckhard Platen. An introduction to numerical methods for stochastic differential equations. *Acta numerica*, 8:197–246, 1999.
- [25] Desmond J. Higham. An algorithmic introduction to numerical simulation of stochastic differential equations. *SIAM review*, 43(3):525–546, 2001.
- [26] Fabrice Douglas Rouah. Euler and milstein discretization, *Documento de trabalho, Sapient Global Markets, EUA*. <http://www.frouah.com/finance%20notes/Euler%20and%20Milstein%20Discretization.pdf>, 2011. Acessado em 11-06-2014.
- [27] Stefano M. Iacus. *Simulation and inference for stochastic differential equations: with R examples*, volume 1. Springer, 2009.

- [28] Hugo de la Cruz Cancino. *Métodos de Linealización Local de orden superior para la integración de ecuaciones diferenciales ordinarias y estocásticas*. PhD thesis, Universidad de La Habana, 2006.
- [29] Arieh Iserles. *A first course in the numerical analysis of differential equations*. Cambridge University Press, 2009.
- [30] SD Conte e Carl de Boor. *Elementary Numerical Analysis?An Algorithmic Approach*. MacGraw-Hill, New York, 1980.
- [31] John Denholm Lambert. *Numerical methods for ordinary differential systems: the initial value problem*. John Wiley & Sons, Inc., 1991.
- [32] André Nachbin e Esteban Tabak. *21<sup>o</sup> Colóquio Brasileiro de Matemática: equações diferenciais em modelagem matemática computacional*. Conselho Nacional de Desenvolvimento Científico e Tecnológico, Instituto de Matemática Pura e Aplicada, 1997.
- [33] Richard L. Burden e J. Douglas Faires. *Análise numérica*. Cengage Learning, 2008.
- [34] Hugo de la Cruz, Rolando J Biscay, Juan C Jiménez, e Felix Carbonell. Local linearization runge–kutta methods: A class of a-stable explicit integrators for dynamical systems. *Mathematical and Computer Modelling*, 57(3):720–740, 2013.
- [35] Pamela Marion Burrage e Kevin Burrage. A variable stepsize implementation for stochastic differential equations. *SIAM Journal on Scientific Computing*, 24(3):848–864, 2002.
- [36] Susanne Mauthner. Step size control in the numerical solution of stochastic differential equations. *Journal of computational and applied mathematics*, 100(1):93–109, 1998.

- [37] Vassilios Sotiropoulos e Yiannis N. Kaznessis. An adaptive time step scheme for a system of stochastic differential equations with multiple multiplicative noise: chemical langevin equation, a proof of concept. *The Journal of chemical physics*, 128(1):014103, 2008.
- [38] Anders Szepessy, Raúl Tempone, e Georgios E. Zouraris. Adaptive weak approximation of stochastic differential equations. *Communications on Pure and Applied Mathematics*, 54(10):1169–1214, 2001.
- [39] Silvana Ilie, Kenneth R. Jackson, e Wayne H. Enright. Adaptive time-stepping for the strong numerical solution of stochastic differential equations. *Numerical Algorithms*, pages 1–22, 2014.
- [40] A. Foroush Bastani e S. Mohammad Hosseini. A new adaptive runge–kutta method for stochastic differential equations. *Journal of Computational and Applied Mathematics*, 206(2):631–644, 2007.
- [41] Jessica G. Gaines e Terry J. Lyons. Variable step size control in the numerical solution of stochastic differential equations. *SIAM Journal on Applied Mathematics*, 57(5):1455–1484, 1997.
- [42] Cleve Moler e Charles Van Loan. Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later. *SIAM review*, 45(1):3–49, 2003.
- [43] Hugo la Cruz Cancino, C. Olivera, e Jorge Passamani Zubelli. Computer simulation of the stochastic transport equation. *Preprint*, 2014.
- [44] Gene H. Golub e Charles F. Van Loan. *Matrix computations*, 3rd, 1996.
- [45] Moshe Gitterman. *The noisy oscillator: the first hundred years, from Einstein until now*. World Scientific, 2005.

- [46] Hugo la Cruz Cancino, Juan Carlos Jimenez, e Jorge Passamani Zubelli. Numerical simulation of stochastic oscillators driven by random forces. *Submitted to Journal of Statistical Physics*, 2015.
- [47] Pedro Jesus Fernandez. *Medida e integração*. IMPA, 2007.