



Universidade do Estado do Rio de Janeiro

Centro de Tecnologia e Ciências

Instituto de Matemática e Estatística

Sara Cristina Quintanilha Valente

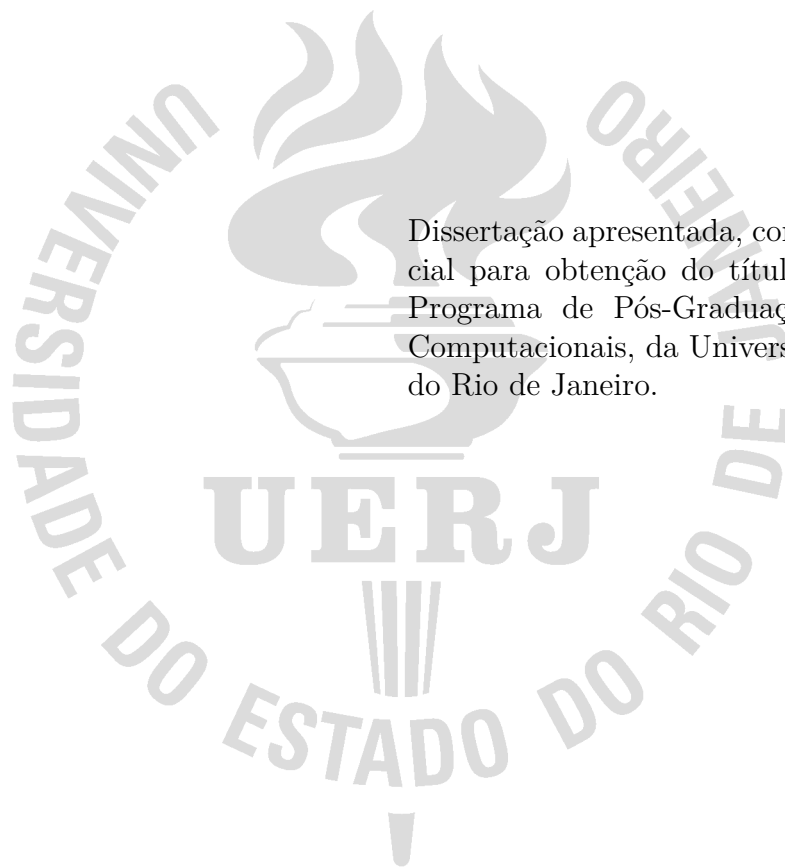
**O Movimento Browniano: aspectos teóricos e abordagem
computacional**

Rio de Janeiro

2019

Sara Cristina Quintanilha Valente

O Movimento Browniano: aspectos teóricos e abordagem computacional



Dissertação apresentada, como requisito parcial para obtenção do título de Mestre, ao Programa de Pós-Graduação em Ciências Computacionais, da Universidade do Estado do Rio de Janeiro.

Orientadora: Prof.^a Dra. Zochil González Arenas

Rio de Janeiro

2019

CATALOGAÇÃO NA FONTE
UERJ / REDE SIRIUS / BIBLIOTECA CTC-A

V154 Valente, Sara Cristina Quintanilha.
O movimento Browniano: aspectos teóricos e abordagem
computacional/ Sara Cristina Quintanilha Valente. – 2019.
102 f. : il.

Orientadora: Zochil González Arenas
Dissertação (Mestrado em Ciências Computacionais) - Universidade
do Estado do Rio de Janeiro, Instituto de Matemática e Estatística.

1. Equações diferenciais estocásticas - Teses. 2. Métodos numéricos
- Teses. 3. - Teses. I. González Arenas, Zochil. II. Universidade do
Estado do Rio de Janeiro. Instituto de Matemática e Estatística. III.
Título.

CDU 519.216

Patricia Bello Meijinhos CRB7/5217 -Bibliotecária responsável pela elaboração da ficha catalográfica

Autorizo, apenas para fins acadêmicos e científicos, a reprodução total ou parcial desta
dissertação, desde que citada a fonte

Assinatura

Data

Sara Cristina Quintanilha Valente

O Movimento Browniano: aspectos teóricos e abordagem computacional

Dissertação apresentada, como requisito parcial para obtenção do título de Mestre, ao Programa de Pós-Graduação em Ciências Computacionais, da Universidade do Estado do Rio de Janeiro.

Aprovada em 14 de Junho de 2019.

Banca Examinadora:

Prof.^a Dra. Zochil González Arenas (Orientadora)
Instituto de Matemática e Estatística – UERJ

Prof. Dr. Americo Barbosa da Cunha Junior
Instituto de Matemática e Estatística – UERJ

Prof. Dr. Daniel Gustavo Barci
Instituto de Física – UERJ

Prof. Dr. Hugo Alexander de la Cruz Cancino
Fundação Getúlio Vargas – FGV

Rio de Janeiro

2019

AGRADECIMENTOS

Quero agradecer primeiramente a Deus, por ter me concedido saúde e forças para eu poder chegar até aqui.

Aos professores do CCOMP, aos quais tive a honra de ser aluna. À minha orientadora, Dra. Zochil Arenas, pelo apoio, paciência, dedicação e incentivo.

Agradeço o auxílio da fundação CAPES, pois o presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

Aos meus familiares, em especial, meus pais Jorge e Cristina que sempre me apoiaram e incentivaram em minha caminhada.

E a todos que diretamente ou indiretamente participaram de minha formação, meus mais sinceros agradecimentos!

RESUMO

VALENTE, Sara Cristina Quintanilha *O Movimento Browniano: aspectos teóricos e abordagem computacional*. 2019. 103 f. Dissertação (Mestrado em Ciências Computacionais) – Instituto de Matemática e Estatística , Universidade do Estado do Rio de Janeiro, Rio de Janeiro, 2019.

O principal objetivo deste trabalho é a resolução de uma Equação Diferencial Estocástica (EDE) representada por um sistema físico superamortecido, sob efeito de um ruído multiplicativo no sistema. Como se sabe, é imensamente difícil solucionar uma EDE analiticamente, pois a equação agora, contém um termo relacionado diretamente ao ruído. Desta forma, para desenvolver sua solução, são empregadas propriedades presentes no cálculo estocástico e não mais as presentes no cálculo usual até então vistas. Em consequência disto, foram investigados métodos iterativos que fossem capazes de integrar sistemas, cuja modelagem utilizasse tais equações. Foram então programados diferentes métodos numéricos como Euler-Maruyama, Euler-Heun, Milstein, Milstein sem derivada e Runge-Kutta Estocástico, na linguagem de programação Matlab, a fim de resolver computacionalmente uma EDE no caso particular de um oscilador harmônico superamortecido. Para os resultados dos diferentes métodos numéricos foi feita uma análise estatística para identificar qual dos métodos possui a tendência de ir mais rapidamente à região de equilíbrio.

Palavras-chave: Equação Diferencial Estocástica. Sistema Físico Superamortecido. Cálculo Estocástico. Métodos Numéricos.

ABSTRACT

VALENTE, Sara Cristina Quintanilha *The Brownian Movement: theoretical aspects and computational approach*. 2019. 103 f. Dissertação (Mestrado em Ciências Computacionais) – Instituto de Matemática e Estatística, Universidade do Estado do Rio de Janeiro, Rio de Janeiro, 2019.

The main objective of this work is the resolution of a Stochastic Differential Equation (EDE) represented by a super-damped physical system, under the effect of a multiplicative noise in the system. As is well known, it is immensely difficult to solve an EDE analytically because the equation now contains a term directly related to noise. Thus, in order to develop its solution, properties present in the stochastic calculus are employed, and no longer those present in the usual calculus previously seen. As a result, iterative methods that were able to integrate systems whose modeling used such equations were investigated. Different numerical methods such as Euler-Maruyama, Euler-Heun, Milstein, Milstein without derivative and Stochastic Runge-Kutta were then programmed in the Matlab programming language in order to computationally solve an EDE in the particular case of an over-damped harmonic oscillator. For the results of the different numerical methods, a statistical analysis was performed to identify which method has the tendency to go faster to the equilibrium region.

Keywords: Stochastic Differential Equation. Super-Amortized Physical System. Stochastic Calculation. Numerical Methods.

LISTA DE FIGURAS

Figura 1	- Movimento Browniano	10
Figura 2	- Função Densidade de Probabilidade	22
Figura 3	- Particionamento do intervalo de tempo	27
Figura 4	- Oscilador Harmônico	40
Figura 5	- Oscilador Harmônico Super-amortecido	41
Figura 6	- Comparação das curvas de diferentes tamanhos de passo (h), utilizando o método de Euler-Maruyama com ruído aditivo e a mesma frequência do Oscilador harmônico.	44
Figura 7	- Método de Euler-Maruyama com ruído aditivo.	45
Figura 8	- Método de Euler-Heun com ruído aditivo.	46
Figura 9	- Método de Milstein conforme interpretação de Itô com ruído aditivo.	47
Figura 10	- Método de Milstein conforme interpretação de Stratonovich com ruído aditivo.	48
Figura 11	- Método de Milstein sem derivada conforme interpretação de Itô com ruído aditivo.	49
Figura 12	- Método de Milstein sem derivada conforme interpretação de Stratonovich com ruído aditivo.	50
Figura 13	- Método de Euler-Maruyama com ruído multiplicativo.	52
Figura 14	- Método de Euler-Heun com ruído multiplicativo.	53
Figura 15	- Método de Milstein conforme interpretação de Itô com ruído multiplicativo.	54
Figura 16	- Método de Milstein conforme interpretação de Stratonovich com ruído multiplicativo.	55
Figura 17	- Método de Milstein sem derivada conforme interpretação de Itô com ruído multiplicativo.	56
Figura 18	- Método de Milstein sem derivada com interp. de Stratonovich com ruído multiplicativo.	57
Figura 19	- Método de Runge-Kutta com ruído multiplicativo.	58
Figura 20	- Matriz com 100 diferentes realizações do ruído feita para cada método numérico.	59

LISTA DE ALGORITMOS

Algoritmo 1 - PROGRAMA IMPLEMENTADO PARA RESOLVER UMA EDE DE UM OSCILADOR HARMÔNICO SUPERAMORTECIDO DADO PELO MODELO PROPOSTO	66
Algoritmo 2 - FUNÇÃO QUE COMPUTA O TAMANHO DO IELE-TH ELEMENTO 1D .	76
Algoritmo 3 - FUNÇÃO QUE GERA UMA MALHA PARA O DOMÍNIO ANALISADO . .	77
Algoritmo 4 - FUNÇÃO QUE GERA RUÍDO BRANCO PARA O PROBLEMA EM QUESTÃO	78
Algoritmo 5 - FUNÇÃO QUE GERA UMA SOLUÇÃO NUMÉRICA USANDO O MÉTODO DE EULER-MARUYAMA	80
Algoritmo 6 - FUNÇÃO QUE GERA UMA SOLUÇÃO NUMÉRICA USANDO O MÉTODO DE MILSTEIN	83
Algoritmo 7 - FUNÇÃO QUE GERA UMA SOLUÇÃO NUMÉRICA USANDO O MÉTODO DE HEUN	89
Algoritmo 8 - FUNÇÃO QUE GERA UMA SOLUÇÃO NUMÉRICA USANDO O MÉTODO DE MILSTEIN SEM DERIVADA	92
Algoritmo 9 - FUNÇÃO QUE GERA UMA SOLUÇÃO NUMÉRICA USANDO O MÉTODO DE RUNGE-KUTTA	99

SUMÁRIO

	INTRODUÇÃO	9
1	EQUAÇÃO DE LANGEVIN	13
2	CONCEITOS BÁSICOS DE PROBABILIDADE E DE CÁLCULO ESTOCÁSTICO	20
2.1	Variável Aleatória Real	20
2.1.1	<u>Função de Distribuição de Probabilidade de uma Variável Aleatória Real</u>	20
2.1.2	<u>Função de Densidade de Probabilidade de uma Variável Aleatória Real</u>	21
2.1.3	<u>Função de Densidade de Probabilidade Usual: Normal ou Gaussiana</u>	21
2.2	Média e Variância de Variáveis Aleatórias	22
2.3	Processos Estocásticos	22
2.3.1	<u>Processo de Markov</u>	24
2.3.2	<u>Dinâmica Estocástica</u>	24
2.3.2.1	<u>Processo de Wiener</u>	25
2.4	Lema de Itô	29
3	MÉTODOS NUMÉRICOS	30
3.1	Expansão da Série Estocástica de Taylor	30
3.1.1	<u>Expansão de Itô-Taylor</u>	31
3.2	Ordem de convergência Forte	34
3.3	Ordem de convergência Fraca	34
3.4	Esquemas de Integração Numérica	34
3.4.1	<u>Método de Euler-Maruyama</u>	35
3.4.2	<u>Método de Euler-Heun</u>	36
3.4.3	<u>Método de Milstein</u>	36
3.4.4	<u>Método de Milstein Sem Derivada</u>	37
3.4.5	<u>Método de Runge-Kutta Estocástico</u>	37
4	RESULTADOS	39
4.1	Modelo Físico Proposto	42
4.1.1	<u>Análise Estatística</u>	59
	CONCLUSÕES	62
	REFERÊNCIAS	64
	APÊNDICE – Modelagem Computacional	66

INTRODUÇÃO

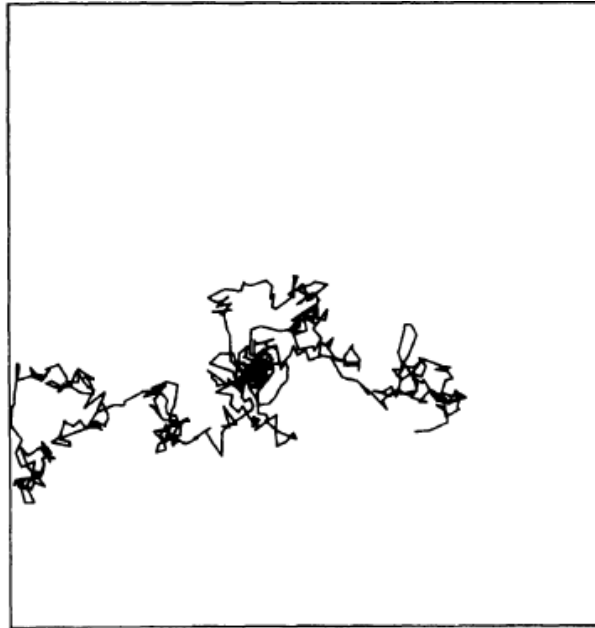
A observação microscópica de como os grãos de pólen suspensos em água desenvolviam um movimento irregular e imprevisível foi estudado de forma sistemática por Robert Brown, no ano de 1828. Devido ao seu trabalho pioneiro fundamental, esse fenômeno tomou o nome de Movimento Browniano. Brown era botânico e a sua primeira intenção foi verificar se o movimento observado era algum indício de vida ali existente. Foi quando verificou que este movimento também estava presente em qualquer suspensão de partículas finas e inanimadas de vidros e minerais. Assim foi descartada qualquer hipótese de que a origem do movimento fosse orgânica. Em décadas posteriores, diversos estudos foram realizados tentando entender a natureza de tal movimento. Experimentos mostravam que ao alterar algumas grandezas físicas, alterava-se a intensidade do movimento. Por exemplo, quando diminuía-se a viscosidade ou o tamanho da partícula ou até mesmo quando aumentava-se a temperatura da solução, o movimento apresentava-se mais intenso. Portanto, o movimento browniano pode ser definido como o movimento imprevisível de partículas suspensas num determinado fluido, podendo este ser líquido ou gasoso. Tal imprevisibilidade do movimento decorre da colisão entre as partículas e os átomos do fluido. Na Figura 1 mostra-se um exemplo da trajetória de uma partícula realizando um movimento Browniano [1, 2].

Os primeiros resultados teóricos que tiveram bastante êxito na explicação do Movimento Browniano foram os de Einstein [3], em 1905. Seus resultados comprovaram a teoria molecular da matéria. Sua teoria se baseava na similaridade entre o comportamento de soluções diluídas e suspensões diluídas, na associação entre difusão e viscosidade e numa inferência de forma probabilística da equação de difusão. Pouco tempo depois, em 1908, o físico francês Paul Langevin obteve também resultados importantes [4]. Inserindo o conceito físico de forças aleatórias, escreveu uma equação diferencial estocástica para uma partícula em suspensão. Para isso, na origem da equação de difusão considerou-se o fenômeno modelado por Stokes [5] na dinâmica dos fluidos e uma força complementar que fosse capaz de manter a agitação da partícula que descreveu como uma força aleatória. Caso essa força complementar não existisse, a força de atrito viscoso acabaria levando a partícula ao repouso. Essa força aleatória é referida ao choque contínuo das partículas em suspensão com as moléculas individuais do fluido [6, 7].

Motivação

A equação de Langevin é uma equação diferencial estocástica que apresenta um termo correspondente à variável aleatória. Sua solução analítica é bastante difícil por não

Figura 1 - Movimento Browniano



Legenda: Partícula realizando um movimento Browniano.

Fonte: Gardiner (2004)

ser possível utilizar ferramentas e técnicas matemáticas empregadas frequentemente no cálculo usual. O desenvolvimento de sua solução é complexa e requer a utilização de um ferramental matemático presente no cálculo estocástico. Sendo assim, a grande motivação deste trabalho é obter a solução numérica da equação de Langevin que descreve o movimento de um oscilador harmônico super-amortecido sob efeito de um ruído multiplicativo no sistema por diferentes métodos numéricos presentes na literatura.

A dinâmica de Langevin possui diversas aplicações em variados problemas. Na Química, por exemplo, é usada na Teoria de Kramers para meios viscosos, onde é comparado o formato da curva potencial ao valor da taxa de reações químicas [8], no estudo do transporte de corrente elétrica em materiais semicondutores [9] e entre outras aplicações. Na Biologia estuda-se a influência das flutuações do meio ambiente no organismo biológico [10] e também aplica-se em simulações de séries climáticas diárias de umidade relativa do ar, que é um fator climático indispensável na análise que seus efeitos de variação podem ocasionar no cenário ambiental [11]. Outros exemplos podem se encontrar na Economia, no estudo das flutuações do mercado de ações [12] e também na modelagem estocástica do crescimento populacional com limitação de recursos [13]. Na Engenharia, pode-se encontrar aplicações no método baseado na teoria do paramagnetismo em magnetização sem histerese [14] e de igual forma na investigação da agregação de nanopartículas de fuligem em escoamentos turbulentos [15]. A origem mesmo da equação deu-se na área da Física, usando-se no estudo de diversos fenômenos físicos que incluem algum comportamento aleatório [1, 8, 12].

Objetivos Gerais e Específicos

O objetivo geral deste trabalho é estudar a equação de Langevin, assim como os conceitos relacionados ao cálculo estocástico, que servem como base para entendimento e pesquisar a viabilidade de algoritmos existentes para programar e solucionar numericamente esta equação, que em particular descreve o movimento de um oscilador harmônico super-amortecido sob efeito de um ruído multiplicativo no sistema. Para o desenvolvimento deste trabalho, foi necessário efetuar uma revisão bibliográfica, para se ter uma maior compreensão de conceitos até então não vistos na Graduação, como o estudo de Cálculo Estocástico, Sistemas Dinâmicos Estocásticos e Métodos Numéricos para Equações Diferenciais Estocásticas. Os conceitos necessários para elaboração desta dissertação estão registrados neste documento, com o objetivo também de servir como material de estudo para futuros estudantes deste tema. Os objetivos específicos numéricos compreendem a análise dos resultados apresentados computacionalmente através dos diferentes métodos numéricos implementados. A verificação da influência dos parâmetros relacionados à frequência característica do oscilador e do fator que controla a intensidade do ruído na equação diferencial estocástica proposta. E a apuração de qual dos métodos possui a tendência de ir mais rapidamente à região de equilíbrio.

Estrutura do Trabalho

No capítulo 1 encontra-se a introdução do trabalho, onde é feita uma breve apresentação do tema de estudo e descreve-se a motivação, os objetivos gerais e específicos, assim como a estrutura do trabalho. Posteriormente no capítulo 2 é deduzida a equação de Langevin e são discutidos alguns dos seus aspectos essenciais. No capítulo 3, se faz uma breve revisão dos conceitos básicos de probabilidade e cálculo estocástico. A continuação, no capítulo 4, serão apresentados os métodos numéricos estocásticos, com o intuito de simular numericamente a equação diferencial estocástica proposta. São eles, o método de Euler-Maruyama, Euler-Heun, Milstein, Milstein sem derivada e Runge-Kutta Estocástico. A análise de Convergência de cada método, comparando-se suas vantagens e desvantagens também será feita. No capítulo 5 será formulado o sistema físico proposto, apresentando todos os conceitos físicos necessários para o entendimento do problema. A análise dos resultados numéricos fará parte deste capítulo. Aqui também serão vistos e discutidos os resultados gráficos para o modelo e será feito um estudo probabilístico dos resultados apresentados, pois como trata-se de uma Equação Diferencial Estocástica associada a um ruído aleatório, a análise numérica da equação não se pode contentar apenas com uma solução e sim em repetir o processo diversas vezes e então executar os cálculos e a análise probabilística. Por fim, no capítulo 6 apresentam-se as conclusões a respeito

de todo o trabalho desenvolvido e serão pontuadas sugestões de trabalhos que possam ser continuados futuramente. Como complemento do trabalho, no Apêndice A, encontra-se a modelagem computacional para o problema proposto, que foi escrito na linguagem de programação Matlab.

1 EQUAÇÃO DE LANGEVIN

Foi apresentado por Langevin, em 1908, uma definição do movimento browniano, que levou em consideração a inclusão das modificações das velocidades ou momentos originados das colisões entre as partículas brownianas e as do fluido no qual estão inseridas, de forma a se aproximar das equações de Newton. Então, devido a estas colisões era fácil visualizar que tais partículas brownianas realizavam um movimento bastante rápido e sem uma direção previamente conhecida. Analisando num intervalo de tempo maior, foi observado que a partícula experimentava um "arraste" originado pela viscosidade do fluido. Sendo assim, Langevin escreveu uma equação de Newton para a partícula, fundamentada na definição da segunda lei de Newton da seguinte maneira [16]

$$m \frac{dv}{dt} = -\frac{v}{B} + F(t). \quad (1)$$

Nessa equação, o primeiro termo da direita refere-se a uma força relacionada à viscosidade do meio, B é a *mobilidade* e $F(t)$ é uma força aleatória devido as inúmeras colisões entre as partículas do fluido e a partícula browniana. Todas as grandezas envolvidas nesse trabalho têm origem física e suas unidades não serão consideradas. Agora serão levadas em conta duas suposições e é importante ressaltar que essas suposições envolvem alguns conceitos de probabilidade que poderão ser vistos no próximo capítulo. Tais suposições são:

- A força média é igual a zero (devido as colisões)

$$\langle F(t) \rangle = \overline{F(t)} = \sum_{j=1}^M F(t)_j P(F(t)_j) = 0, \quad (2)$$

onde $F(t)$ é uma variável aleatória que pode assumir M valores discretos [17]. Vamos denotar o valor esperado usando-se $\langle . \rangle$

- A colisão subsequente não depende da colisão anterior, tais colisões estão descorrelacionadas, a função Delta de Dirac representa esta situação. De forma geral, a função Delta de Dirac pode ser expressa por

$$\delta(x - x_0) = \begin{cases} \infty, & x = x_0 \\ 0, & x \neq x_0 \end{cases} \quad e \quad \int_{-\infty}^{\infty} \delta(x - x_0) dx = 1 \quad (3)$$

Logo, tais colisões sucessivas descorrelacionadas descrevem-se matematicamente da

seguinte maneira

$$\langle F(t)F(t') \rangle = C\delta(t - t'), \quad (4)$$

em que C é uma constante e indica a intensidade do ruído estocástico. Do primeiro item, verifica-se que a força média aplicada sobre partícula refere-se a força viscosa. Agora, o segundo item, relaciona-se ao fato de que a partícula não detém memória das sucessivas colisões.

Dividindo a equação (1) pela massa m , a equação de Langevin escreve-se

$$\frac{dv}{dt} = -\gamma v + \eta(t), \quad (5)$$

onde $\gamma = \frac{1}{mB}$ é o coeficiente de atrito e $\eta(t) = \frac{F(t)}{m}$ é a força externa representada por um processo estocástico, que vamos nos referir assim a partir de agora, chamado de ruído branco. Esse processo estocástico apresenta uma função densidade de probabilidade do tipo normal (gaussiana).

Reescrevendo as propriedades (2) e (4), tem-se que o ruído branco caracteriza-se pelas seguintes propriedades

$$\langle \eta(t) \rangle = 0 \quad , \quad \langle \eta(t)\eta(t') \rangle = \Gamma\delta(t - t'), \quad (6)$$

onde $\Gamma = \frac{C}{m^2}$ é uma constante, considerada uma medida da intensidade do ruído estocástico.

De forma a obter a solução da equação de Langevin, vista na equação (5), a função velocidade será escrita na forma

$$v(t) = u(t)e^{-\gamma t}, \quad (7)$$

em que $u(t)$ é uma função a ser determinada. A análise que será feita a continuação corresponde à teoria de Einstein do Movimento Browniano [3, 7].

Substituindo a expressão anterior na equação (5), tem-se que

$$u(t)(-\gamma e^{-\gamma t}) + \frac{du(t)}{dt}e^{-\gamma t} = -\gamma u(t)e^{-\gamma t} + \eta(t), \quad (8)$$

de onde obtém-se

$$\frac{du(t)}{dt}e^{-\gamma t} = \eta(t). \quad (9)$$

Tem-se então que $u(t)$ deve satisfazer a seguinte equação

$$\frac{du(t)}{dt} = e^{\gamma t}\eta(t). \quad (10)$$

Integrando ambos os lados da equação

$$\int_{u_0}^u du = \int_0^t e^{\gamma t'}\eta(t')dt', \quad (11)$$

resulta que

$$u(t) - u_0 = \int_0^t e^{\gamma t'}\eta(t')dt'. \quad (12)$$

A integral no lado direito da equação anterior não pode ser calculada do modo usual, por conter um processo estocástico. Vamos deixar nessa forma para trabalhar posteriormente com valores médios da expressão. Considerando agora como condição inicial $v(t=0) = u(t=0) = v_0 = u_0$, tem-se

$$u(t) = u_0 + \int_0^t e^{\gamma t'}\eta(t')dt', \quad (13)$$

e substituindo $u(t)$ na expressão (7)

$$v(t) = v_0e^{-\gamma t} + \int_0^t e^{-\gamma(t-t')}\eta(t')dt'. \quad (14)$$

Esta equação fornece $v(t)$ para uma realização particular do ruído. Como estamos lidando com processos estocásticos, vamos interpretar estes resultados no formato probabilístico. Empregando as propriedades do ruído branco acima consideradas, pode-se encontrar a velocidade média e a velocidade quadrática média da partícula browniana. Tomando o valor médio de $v(t)$, na expressão (14) anula-se o termo relacionado ao ruído e obtém-se

$$\langle v \rangle = v_0e^{-\gamma t}. \quad (15)$$

Pode-se perceber que, a velocidade média decai exponencialmente à zero, logo para um tempo $\tau \geq \frac{1}{\gamma}$, tem-se $\langle v \rangle \approx 0$. Este tempo é denominado como *tempo de relaxação*. Isso ocorre devido à viscosidade do fluido ou por conta da presença de forças dissipativas.

Considerando que para tempos longos a velocidade média se anula, vamos analisar o desvio quadrático médio da velocidade. Para o cálculo da velocidade quadrática média

serão utilizados os resultados obtidos nas equações (14) e (15)

$$v - \langle v \rangle = \int_0^t e^{-\gamma(t-t')} \eta(t') dt', \quad (16)$$

elevando ao quadrado

$$\begin{aligned} (v - \langle v \rangle)^2 &= \left(\int_0^t e^{-\gamma(t-t')} \eta(t') dt' \right) \left(\int_0^t e^{-\gamma(t-t'')} \eta(t'') dt'' \right) \\ &= \int_0^t \int_0^t e^{-\gamma(t-t')} e^{-\gamma(t-t'')} \eta(t') \eta(t'') dt' dt'' \\ &= \int_0^t \int_0^t e^{-\gamma t + \gamma t' - \gamma t + \gamma t''} \eta(t') \eta(t'') dt' dt'' \\ &= \int_0^t \int_0^t e^{-2\gamma t} e^{\gamma(t'+t'')} \eta(t') \eta(t'') dt' dt'', \end{aligned} \quad (17)$$

obtém-se então

$$(v - \langle v \rangle)^2 = e^{-2\gamma t} \int_0^t \int_0^t e^{\gamma(t'+t'')} \eta(t') \eta(t'') dt' dt''. \quad (18)$$

Utilizando as propriedades do ruído branco expostas na equação (6), toma-se o valor médio em (18) e obtém-se

$$\langle (v - \langle v \rangle)^2 \rangle = e^{-2\gamma t} \Gamma \int_0^t \int_0^t e^{\gamma(t'+t'')} \delta(t' - t'') dt' dt''$$

integrando em t'' e usando as propriedades do ruído

$$\langle (v - \langle v \rangle)^2 \rangle = e^{-2\gamma t} \Gamma \int_0^t e^{2\gamma t'} dt', \quad (19)$$

e calculando a integral, tem-se

$$e^{-2\gamma t} \Gamma \frac{e^{2\gamma t'}}{2\gamma} \Big|_0^t = e^{-2\gamma t} \Gamma \left[\frac{e^{2\gamma t}}{2\gamma} - \frac{e^0}{2\gamma} \right] = e^{-2\gamma t} \Gamma \left[\frac{e^{2\gamma t} - 1}{2\gamma} \right]. \quad (20)$$

A variância da velocidade é dada então por

$$\langle (v - \langle v \rangle)^2 \rangle = \langle v^2 \rangle - \langle v \rangle^2 = \frac{\Gamma}{2\gamma} (1 - e^{-2\gamma t}). \quad (21)$$

Note que a velocidade média se anula para tempos longos, porém a variância não é nula. Para tempos longos, a velocidade quadrática média se aproxima de

$$\langle v^2 \rangle = \frac{\Gamma}{2\gamma}. \quad (22)$$

Analisando um sistema clássico em equilíbrio termodinâmico, pode-se considerar o princípio de equipartição de energia. Este princípio é oriundo da Teoria Cinética dos

Gases, onde para cada grau de liberdade se associa a energia cinética média à energia térmica dada pela temperatura, da forma,

$$\frac{1}{2}m\langle v^2 \rangle = \frac{1}{2}k_B T, \quad (23)$$

onde k_B é a constante de Boltzmann e T é a temperatura absoluta.

Substituindo a variância da velocidade pelo resultado encontrado na equação (22)

$$\frac{1}{2}m\frac{\Gamma}{2\gamma} = \frac{1}{2}k_B T. \quad (24)$$

Logo, obtém-se a expressão onde a constante Γ está em função do coeficiente de atrito γ

$$\Gamma = \frac{2\gamma k_B T}{m}, \quad (25)$$

que é denominada *relação de Einstein* ou *relação de flutuação-dissipação*. Sendo assim, pode-se dizer que a intensidade das flutuações em Γ está diretamente relacionada à intensidade de dissipação em γ , quando o sistema está em equilíbrio à temperatura T .

Vamos realizar agora um desenvolvimento similar para estudar o deslocamento da partícula. Com base no mesmo raciocínio anterior, é possível encontrar o deslocamento médio da partícula que desenvolve um movimento Browniano. Considerando que a velocidade é derivada da posição, escreve-se

$$\frac{dx(t)}{dt} = v(t), \quad (26)$$

integrando ambos os lados da expressão anterior

$$\int_{x_0}^x dx = \int_0^t v(t') dt', \quad (27)$$

e considerando a condição inicial $x(t=0) = x_0$, obtém-se

$$x(t) = x_0 + \int_0^t v(t') dt'. \quad (28)$$

Substituindo agora $v(t)$ calculado anteriormente na equação (14), tem-se

$$x(t) = x_0 + \int_0^t v_0 e^{-\gamma t'} dt' + \int_0^t e^{-\gamma t'} \left[\int_0^{t'} e^{\gamma t''} \eta(t'') dt'' \right] dt'. \quad (29)$$

Por fim, integrando o segundo e o terceiro termo da equação anterior, obtém-se

$$x(t) = x_0 + v_0 \frac{1}{\gamma} (1 - e^{-\gamma t}) + \frac{1}{\gamma} \int_0^t \eta(t'') (1 - e^{\gamma(t''-t)}) dt''. \quad (30)$$

Agora é possível usar das propriedades do ruído apresentadas na equação (6), para obter, finalmente, o deslocamento médio da partícula

$$\langle x(t) \rangle = x_o + v_0 \frac{1}{\gamma} (1 - e^{-\gamma t}). \quad (31)$$

Pode-se ver que, diferentemente da velocidade, para tempos longos o deslocamento médio não se anula. Essa foi uma das grandes contribuições de Einstein nessa área, que percebeu que a grandeza a ser considerada seriam as distâncias ao invés das velocidades.

Agora, também de forma similar ao que foi efetuado no caso das velocidades, vamos calcular o deslocamento quadrático médio. Para isso, consideremos

$$x - \langle x \rangle = \frac{1}{\gamma} \int_0^t \eta(t'') (1 - e^{\gamma(t''-t)}) dt'', \quad (32)$$

e, elevando ao quadrado e tomando o valor médio, tem-se

$$\langle (x - \langle x \rangle)^2 \rangle = \frac{\Gamma}{\gamma^2} \int_0^t \int_0^t \delta(t' - t'') (1 - e^{\gamma(t''-t)})^2 dt'' dt'. \quad (33)$$

Calculando e aplicando as propriedades vistas do ruído, obtemos finalmente

$$\langle (x - \langle x \rangle)^2 \rangle = \frac{\Gamma}{\gamma^2} \left(t - \frac{2}{\gamma} + \frac{2e^{-\gamma t}}{\gamma} + \frac{1}{2\gamma} - \frac{e^{-2\gamma t}}{2\gamma} \right), \quad (34)$$

que pode ser reescrita da seguinte forma

$$\langle (x - \langle x \rangle)^2 \rangle = \langle x^2 \rangle - \langle x \rangle^2 = \frac{\Gamma}{\gamma^2} \left(t - \frac{2}{\gamma} (1 - e^{-\gamma t}) + \frac{1}{2\gamma} (1 - e^{-2\gamma t}) \right). \quad (35)$$

Analisando esse resultado para tempos muitos longos, pode-se ver que a equação (35) depende apenas do termo linear em t

$$\langle x^2 \rangle - \langle x \rangle^2 \approx Dt, \quad (36)$$

em que $D = \frac{\Gamma}{\gamma^2}$ é chamado de *constante de difusão*. Essa condição caracteriza um processo de difusão.

No geral, um processo do tipo $\langle x^2 \rangle - \langle x \rangle^2 \approx t^\alpha$ é chamado de difusão. O expoente $\alpha = 1$, corresponde ao movimento browniano e é chamado de difusão "normal", enquanto que o processo com $\alpha \neq 1$ são chamados de difusão anômala.

Utilizando agora a expressão para Γ encontrada antes, equação (25), pode-se escrever, a constante de difusão como

$$D = \frac{2k_B T}{\gamma m}, \quad (37)$$

que é uma outra forma da relação de flutuação-dissipação vista anteriormente. As relações de flutuação-dissipação são importantes na verificação do estado de equilíbrio em sistemas físicos.

2 CONCEITOS BÁSICOS DE PROBABILIDADE E DE CÁLCULO ESTOCÁSTICO

2.1 Variável Aleatória Real

O espaço amostral Ω de um experimento é definido como sendo o conjunto de todos os resultados possíveis deste experimento. Nem sempre os elementos do espaço Ω são números, mas convém que Ω possua uma representação numérica. Sendo assim, ao relacionar cada um dos elemento $\omega \in \Omega$ a um número real $x(\omega)$, será possível descrever em termos numéricos todos os resultados possíveis do experimento. Denomina-se variável aleatória a função x que relaciona os pontos de Ω em \mathbb{R} .

Um sistema de probabilidades é uma tripla (Ω, \mathcal{A}, P) , onde Ω é o espaço de amostras, \mathcal{A} é uma σ -álgebra de eventos e P a medida de probabilidade.

Uma variável aleatória x definida em Ω é uma função real que toma valores no conjunto \mathbb{R} , obedecendo aos seguintes critérios [18]

(i) O conjunto $A_X = \{\omega \in \Omega : x(\omega) \leq X\}$ é um evento, seja qual for $X \in \mathbb{R}$;

A probabilidade P desta função é nula, quando ela assume valores infinitos

(ii) $P\{\omega \in \Omega : x(\omega) = -\infty\} = P\{\omega \in \Omega : x(\omega) = \infty\} = 0$.

2.1.1 Função de Distribuição de Probabilidade de uma Variável Aleatória Real

Seja uma função F_x definida por

$$F_x : \mathbb{R} \mapsto \mathbb{R}$$

$$X \mapsto F_x(X) = P(A) \tag{38}$$

A função F_x é chamada de função de distribuição de probabilidade (FDP), também chamada de função de distribuição cumulativa, de uma variável aleatória x . A é o evento definido na seção anterior.

É comum usar a notação simplificada para representar a função de distribuição de probabilidade, pois para uma variável aleatória x , esta função associa a probabilidade da variável aleatória x apresentar um valor menor ou igual que X .

A notação Simplificada para FDP de uma variável aleatória x é dada da seguinte

maneira

$$F_x(X) = P(x \leq X). \quad (39)$$

2.1.2 Função de Densidade de Probabilidade de uma Variável Aleatória Real

A função de densidade de probabilidade de uma variável aleatória real é definida como a derivada da função de distribuição de probabilidade. Pode ser escrita na forma

$$p_x = \frac{d}{dx} F_x(X). \quad (40)$$

De acordo com a definição, percebe-se que sua aplicação dá-se naturalmente em funções de distribuição de probabilidade contínuas e que possam ser diferenciadas, ou até mesmo em funções contínuas que não sejam diferenciadas exclusivamente num número específico de pontos. Para variáveis aleatórias descontínuas ou mistas, graças as suas descontinuidades não é possível aplicá-la na equação de definição. Neste caso, é necessário recorrer a Função Impulso [19].

2.1.3 Função de Densidade de Probabilidade Usual: Normal ou Gaussiana

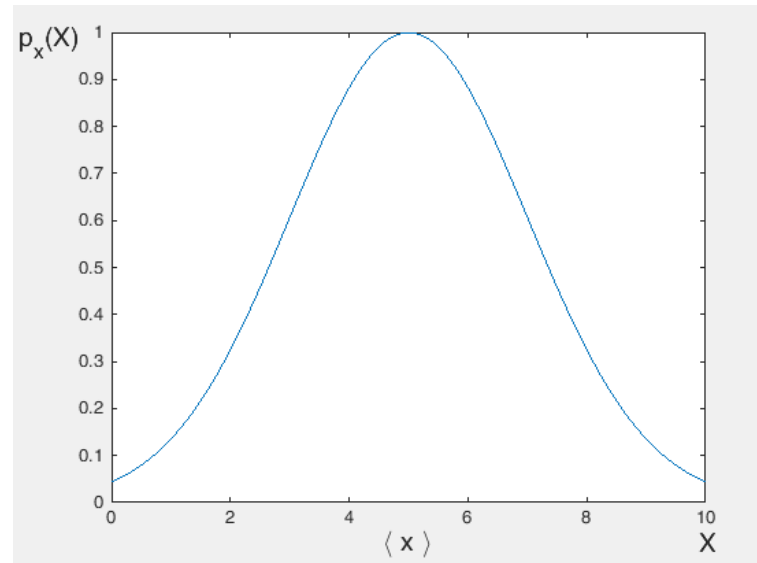
Afirma-se que a distribuição (gaussiana) de uma variável aleatória x é normalmente distribuída, se sua função densidade de probabilidade obedecer a seguinte condição

$$p_x(X) = \frac{1}{\sqrt{2\pi\sigma}} \exp^{-\frac{(X-\langle x \rangle)^2}{2\sigma^2}}, \quad (41)$$

onde $\langle x \rangle$ é a média e σ^2 é a variância da variável aleatória X .

O gráfico que representa a função de densidade de probabilidade Gaussiana pode ser visto na Figura 2. Sua curva tem a forma de um sino e é simétrica em relação à média, $m = \langle x \rangle$.

Figura 2 - Função Densidade de Probabilidade



Legenda: Gráfico que representa a Função de Densidade de Probabilidade Gaussiana.

2.2 Média e Variância de Variáveis Aleatórias

A média ou o valor esperado de uma variável aleatória x pode ser determinado através da seguinte formulação integral

$$\langle x \rangle = \int_{-\infty}^{\infty} X p_x(X) dX. \quad (42)$$

A variância σ_x^2 de uma variável aleatória x pode ser calculada da seguinte forma

$$\sigma_x^2 = \langle (x - \langle x \rangle)^2 \rangle = \langle x^2 \rangle - \langle x \rangle^2 = \int_{-\infty}^{\infty} (X - \langle x \rangle)^2 p_x(X) dX. \quad (43)$$

Ao realizar a raiz quadrada de σ_x^2 , da variância de uma variável aleatória x , obtém-se o **desvio-padrão** da variável aleatória x . Tanto a variância quanto o desvio-padrão são parâmetros de dispersão e estão relacionados à dispersão da variável aleatória ao redor de sua média.

2.3 Processos Estocásticos

Um processo estocástico $X(t)$ pode ser definido como a função de uma variável aleatória de um sistema que possua dependência com o tempo. Podem ser medidos x_1, x_2, x_3, \dots , valores de $X(t)$ no tempo t_1, t_2, t_3, \dots , onde assume-se que existam as den-

sidades de probabilidade conjunta [1]

$$p(x_1, t_1; x_2, t_2; x_3, t_3; \dots), \quad (44)$$

que representam o sistema de forma completa. Em vista das funções de densidade de probabilidade conjunta apresentadas, é possível determinar as densidades de probabilidade condicional

$$p(x_1, t_1; x_2, t_2; \dots | y_1, \tau_1; y_2, \tau_2; \dots) = \frac{p(x_1, t_1; x_2, t_2; \dots; y_1, \tau_1; y_2, \tau_2; \dots)}{p(y_1, \tau_1; y_2, \tau_2; \dots)}. \quad (45)$$

Tais definições possuem validade independente da ordem dos tempos. Contudo, de forma usual, consideram-se somente os tempos que aumentam da direita para a esquerda, ou seja

$$t_1 \geq t_2 \geq t_3 \geq \dots \geq \tau_1 \geq \tau_2 \geq \tau_3 \dots \quad (46)$$

A concepção de uma equação que possua uma evolução no tempo, conduz à consideração das probabilidades condicionais como sendo uma forma de prever os valores futuros de $X(t)$, ou seja, (x_1, x_2, x_3, \dots) , em tempos (t_1, t_2, t_3, \dots) , dado que o passado seja conhecido (y_1, y_2, y_3, \dots) , em tempos $(\tau_1, \tau_2, \tau_3, \dots)$.

Para definição do processo é necessário que sejam conhecidas todas as possíveis probabilidades conjuntas, vista na equação (44). Desta maneira, defini-se o processo como processo estocástico separável.

O modelo mais elementar de processo estocástico chama-se independência completa

$$p(x_1, t_1; x_2, t_2; x_3, t_3, \dots) = \prod_i p(x_i, t_i). \quad (47)$$

Sendo assim, pode-se verificar que o valor de X em t não depende do valor anterior ou posterior. Em especial quando uma única lei de probabilidade controla todo o processo, no momento em que $p(x_i, t_i)$ independem de t_i . Como exemplo, pode-se destacar os ensaios de Bernoulli que condiz a uma experiência aleatória onde apenas dois resultados são possíveis, verdadeiro ou falso e a probabilidade de sucesso não se altera qualquer que seja a experiência.

Posteriormente será visto, um processo mais simples, denominado processo de Markov, onde apenas o conhecimento do presente é que define o futuro.

2.3.1 Processo de Markov

A teoria de Markov é expressa em termos das probabilidades condicionais. Se for satisfeita a ordenação temporal vista na equação (46), determina-se a probabilidade de condição pelo conhecimento da condição mais recente. Tem-se que

$$p(x_1, t_1; x_2, t_2; \dots | y_1, \tau_1; y_2, \tau_2; \dots) = p(x_1, t_1; x_2, t_2; \dots | y_1, \tau_1). \quad (48)$$

Esta propriedade é de grande relevância, pois significa que seja qual for a probabilidade, a mesma poderá ser escrita em função de probabilidades condicionais simples $p(x_1, t_1 | y_1, \tau_1)$. Por exemplo, com a utilização da definição de probabilidade condicional $p(x_1, t_1; x_2, t_2 | y_1, \tau_1) = p(x_1, t_1 | x_2, t_2; y_1, \tau_1)$ e através da suposição de Markov, vista na equação (48), encontra-se

$$p(x_1, t_1; x_2, t_2; y_1, \tau_1) = p(x_1, t_1 | x_2, t_2)p(x_2, t_2 | y_1, \tau_1). \quad (49)$$

É simples verificar que uma probabilidade conjunta arbitrária pode ser representada por

$$p(x_1, t_1; x_2, t_2; x_3, t_3; \dots x_n, t_n) = p(x_1, t_1 | x_2, t_2)p(x_2, t_2 | x_3, t_3)p(x_3, t_3 | x_4, t_4) \dots \dots p(x_{n-1}, t_{n-1} | x_n, t_n)p(x_n, t_n), \quad (50)$$

quando

$$t_1 \geq t_2 \geq t_3 \geq \dots \geq t_{n-1} \geq t_n. \quad (51)$$

2.3.2 Dinâmica Estocástica

O termo *Dinâmica Estocástica* que nomeia esta subseção, está relacionado à evolução no tempo de variáveis aleatórias submetidas a um ruído. Aqui será visto, como se determina as soluções numéricas das equações chamadas *Equações Diferenciais Estocásticas* (EDE), equações estas que são chamadas assim, por conterem um termo de ruído. As EDEs também são chamadas de *equações de Langevin* e classificam-se em equações de ruído aditivo ou multiplicativo [20].

O cálculo infinitesimal que é utilizado para interpretar e obter a solução da EDE denomina-se *Cálculo Estocástico*. Este cálculo difere do cálculo infinitesimal usual.

2.3.2.1 Processo de Wiener

A progressão no tempo respeitada pela equação de Langevin com ruído gaussiano branco é um *Processo de Wiener* e está relacionado ao ruído branco $\eta(t)$ da seguinte forma

$$W(t) = \int_0^t \eta(t') dt'. \quad (52)$$

Algumas propriedades deste processo podem ser vistas abaixo. Estas seguem naturalmente sua definição integral na equação (52)

1. $\langle W(t) \rangle = 0$, para qualquer que seja t . Vem da propriedade do ruído branco, onde: $\langle \eta(t) \rangle = 0$;
2. $\langle W(t)W(t') \rangle = \Gamma \min(t, t')$, vem da definição de $W(t)$ e da função correlação de $\eta(t)$;
3. $\langle W(t)^2 \rangle = \sigma_W^2 = \Gamma t$, segue da propriedade 2;
4. $W(t_2) - W(t_1)$ ou $W(t_2) + W(t_1)$, para $0 \leq t_1 \leq t_2 \leq \infty$ são independentes;
5. $\Delta W(t) = W(t + \Delta t) - W(t)$, de acordo com a definição de $\Delta W(t)$, a variância pode ser dada por $\sigma_{\Delta W}^2 = \langle [\Delta W(t)]^2 \rangle = \Gamma \Delta t$;
6. $P_W(\omega, t) = \frac{1}{\sqrt{2\pi\Gamma t}} e^{-\frac{1}{2}(\frac{\omega^2}{\Gamma t})}$, é a sua distribuição de probabilidade. A equação geral pode ser vista na equação (41);
7. $T(\omega, t | \omega_0, t_0) = \frac{1}{\sqrt{2\pi\Gamma(t-t_0)}} e^{-\frac{1}{2}(\frac{\omega - \omega_0^2}{\Gamma(t-t_0)})}$, é a equação da probabilidade condicional ou de transição, vem da propriedade anterior.

Considerando $W(t_0) = \omega_0$, a equação (52) pode ser escrita como

$$W(t) = \omega_0 + \int_{t_0}^t \eta(t') dt'. \quad (53)$$

Da 5ª propriedade, verifica-se que a largura da distribuição para o *incremento de Wiener*, $\Delta W(t) = W(t + \Delta t) - W(t)$, é dada por

$$\sigma_{\Delta W} = \sqrt{\Delta t}. \quad (54)$$

Tais propriedades remetem ao processo de Wiener características matemáticas específicas. Exemplo: Ao tentar desenvolver analiticamente a derivada de $W(t)$ no cálculo usual, verifica-se que $W(t)$ não possui derivada temporal

$$\frac{dW}{dt} = \lim_{\Delta t \rightarrow 0} \frac{\Delta W}{\Delta t} \approx \lim_{\Delta t \rightarrow 0} \frac{\sqrt{\Delta t}}{\Delta t} \rightarrow \infty. \quad (55)$$

Logo, esta verificação faz com que o processo de Wiener definido anteriormente como a integral do ruído branco seja contraditória. Do cálculo usual, era de se esperar que a derivada de $W(t)$ fosse propriamente o ruído branco $\eta(t)$, mas isto se dá apenas em funções contínuas e diferenciáveis, não sendo este o caso do ruído branco.

Desta forma, a equação de Langevin escrita na forma mais usual [21]

$$\frac{dx}{dt} = f(x(t)) + g(x(t))\eta(t), \quad (56)$$

não existe, se for levado em conta um critério matemático mais rígido, mas a sua forma integral pode ser avaliada de maneira coerente

$$x(t) - x(0) = \int_0^t f(x(t'))dt' + \int_0^t g(x(t'))\eta(t')dt'. \quad (57)$$

Do último termo da equação anterior, tem-se que

$$\int_0^t g(x(t'))\eta(t')dt' = \int_0^t g(x(t'))dW(t'), \quad (58)$$

devido a definição do processo de Wiener $W(t)$ como $\eta(t) = dW(t)/dt$.

Considere $G(t)$ e $W(t)$ uma função "adequada" no tempo e o processo de Wiener, respectivamente. Define-se então a integral estocástica $\int_{t_0}^t G(t')dW(t')$ como um tipo de integral de Riemman-Stieltjes. O intervalo $[t_0, t]$ é subdividido em n subintervalos por intermédio dos pontos de particionamento de [1]

$$t_0 \leq t_1 \leq t_2 \leq t_3 \leq \dots \leq t_{n-2} \leq t_{n-1} \leq t, \quad (59)$$

conforme pode ser visto na Figura 3 e determinado os pontos intermediários τ_i como $t_{i-1} \leq \tau_i \leq t_i$.

A integral estocástica de Riemann-Stieltjes $\int_{t_0}^t G(t')dW(t')$ é definida como um limite (no sentido da média quadrática) das somas parciais

$$S_n = \sum_{i=1}^n G(\tau_i)[W(t_i) - W(t_{i-1})], \quad (60)$$

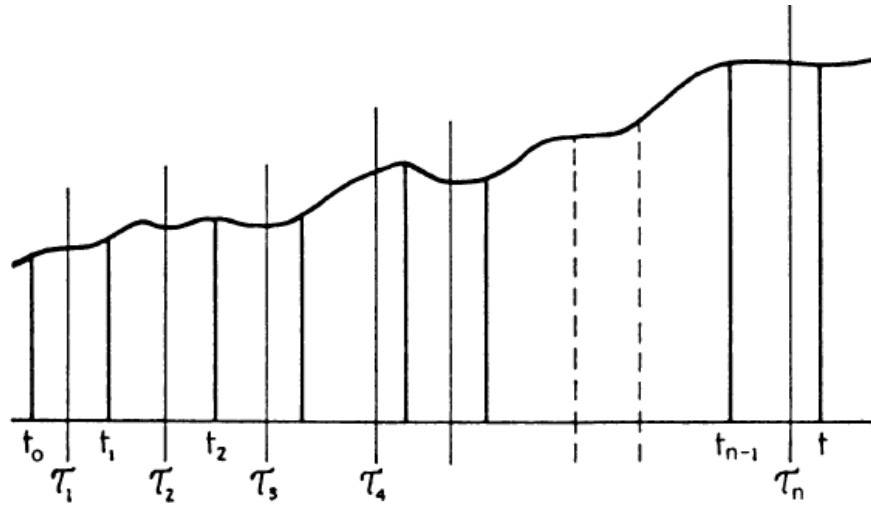
ou seja

$$\int_{t_0}^t G(t')dW(t') = \lim_{n \rightarrow \infty} \sum_{i=1}^n G(\tau_i)[W(t_i) - W(t_{i-1})]. \quad (61)$$

A integral definida na equação (61), depende da seleção particular do ponto intermediário τ_i .

A escolha do valor intermediário τ_i , para todo i , pode ser parametrizada por exem-

Figura 3 - Particionamento do intervalo de tempo



Legenda: Particionamento do intervalo de tempo, de t_0 a t , em n subintervalos, através de n pontos intermediários τ .

Fonte: Gardiner(2004)

plo, por

$$\tau_i = \alpha t_i + (1 - \alpha)t_{i-1}, \quad (0 < \alpha < 1). \quad (62)$$

Para definir a Integral Estocástica de Itô, serão escolhidos pontos intermediários cujo $\alpha = 0$, isto é: $\tau_i = t_{i-1}$ e será definido o limite média quadrática que usaremos adiante. Este corresponde a um limite empregado na área de probabilidade, onde o limite esperado da diferença quadrática é igual a zero. Seja X_n uma série de variáveis aleatórias definidas no espaço de probabilidade Ω e suponha que $X_n(\omega)$ estejam em função de ω . Pode-se afirmar que X_n aproxima-se de X no limite média quadrática quando

$$\lim_{n \rightarrow \infty} \int p(\omega) [X_n(\omega) - X(\omega)]^2 d\omega \equiv \lim_{n \rightarrow \infty} \langle (X_n - X)^2 \rangle = 0, \quad (63)$$

esse tipo de limite é facilmente encontrado na teoria espacial de Hilbert. Escrevendo-se então

$$ms - \lim_{n \rightarrow \infty} X_n = X. \quad (64)$$

Portanto a *Integral Estocástica de Itô* da função $G(t)$ é definida da seguinte forma

$$\int_{t_0}^t G(t') dW(t') = ms - \lim_{n \rightarrow \infty} \left\{ \sum_{i=1}^n G(t_{i-1}) [W(t_i) - W(t_{i-1})] \right\}. \quad (65)$$

Stratonovich introduziu um conceito alternativo, onde é considerada somente a média em $G(t)$, ou seja, $\frac{1}{2} [G(t_i) + G(t_{i-1})]$. Para o cálculo de $\int_{t_0}^t G(t')dW(t')$ é fácil verificar que se dará da seguinte forma

$$\int_{t_0}^t G(t')dW(t') = ms - \lim_{n \rightarrow \infty} \left\{ \sum_{i=1}^n \frac{G(t_i) + G(t_{i-1})}{2} [W(t_i) - W(t_{i-1})] \right\}. \quad (66)$$

Para quaisquer funções arbitrárias $G(t)$, não há ligação entre as integrais definidas por Stratonovich e as definidas por Itô, pois não existe relação geral entre ambas as integrais definidas anteriormente. porém é importante salientar que pode-se determinar que a função $G(t)$ encontra-se associada a uma EDE.

Reescrevendo a equação (66), com o integrando em função de $x(t)$, tem-se que [21]

$$\int_{t_0}^t g(x(t'))dW(t') = ms - \lim_{n \rightarrow \infty} \left\{ \sum_{i=1}^n \frac{g(x(t_i)) + g(x(t_{i-1}))}{2} [W(t_i) - W(t_{i-1})] \right\}. \quad (67)$$

Vale ressaltar que as diversas especificações existentes para determinar a integral estocástica estão agrupadas na chamada "prescrição generalizada de Stratonovich", onde a equação (67), levando em consideração a parametrização vista na equação (62), nos remete a

$$\int_{t_0}^t g(x(t'))dW(t') = ms - \lim_{n \rightarrow \infty} \left\{ \sum_{i=1}^n \frac{\alpha g(x(t_i)) + (1 - \alpha)g(x(t_{i-1}))}{2} [W(t_i) - W(t_{i-1})] \right\}. \quad (68)$$

considerando $0 \leq \alpha \leq 1$.

Desta forma, existem diferentes valores para α , sendo as seguintes interpretações mais usadas

- Interpretação de Itô - Equação (65)
 $\alpha = 0$
- Interpretação de Stratonovich - Equação (66)
 $\alpha = \frac{1}{2}$
- Interpretação de Hanggi-Klimontovich ou Prescrição Cinética
 $\alpha = 1$

2.4 Lema de Itô

O lema de Itô é o que proporciona a regra essencial para obtenção do diferencial de funções compostas no cálculo estocástico, sendo sua utilização muito importante nos métodos numéricos. Em outras palavras, o lema de Itô pode ser definido como sendo o correspondente estocástico da regra da cadeia no cálculo tradicional, sendo utilizado para solucionar algumas equações diferenciais estocásticas simples. Considere uma função $F(x, t)$ que seja diferenciável. Do teorema de Taylor, tem-se que

$$F(t + dt, x + dx) = F(t, x) + F_x(t, x)dx + F_t(t, x)dt + \frac{1}{2} \left[F_{xx}(dx)^2 + 2F_{tx}(dt)(dx) + F_{tt}(dt)^2 \right] + \dots, \quad (69)$$

onde usa-se o índice para se referir à derivada de F com respeito à variável indicada, por exemplo, $F_x = \frac{\partial F}{\partial x}$.

Vamos considerar o diferencial dx correspondente a um processo de origem estocástica, definido pela equação diferencial estocástica abaixo

$$dx = a(t, x)dt + b(t, x)dW_t. \quad (70)$$

Substituindo a equação (70) na equação (69), tem-se

$$\begin{aligned} F(t + dt, x + dx) - F(t, x) &= \frac{\partial F}{\partial t} dt + \frac{\partial F}{\partial x} [a(t, x)dt + b(t, x)dW_t] \\ &+ \frac{1}{2} \frac{\partial^2 F}{\partial x^2} \left[a^2(t, x)(dt)^2 + 2a(t, x)b(t, x)dt dW_t + b^2(t, x)(dW_t)^2 \right] \\ &+ \frac{\partial^2 F}{\partial t \partial x} (dt) \left[a(t, x)dt + \frac{1}{2} b(t, x)dW_t \right] + \frac{\partial^2 F}{\partial t^2} (dt)^2 + o(dt^2). \end{aligned}$$

Considerando que dt é muito pequeno, vamos ficar apenas com os termos de ordem linear em dt . Observa-se que, pelas propriedades do processo de Wiener, $dW^2 \sim dt$, logo, os termos de ordem quadrática em dW deverão ser conservados, porém $dt dW \sim dt^{\frac{3}{2}}$. Em seguida, desprezando os termos de ordem superior em dt o resultado final desta operação é o lema de Itô em duas dimensões [1]

$$dF = \left[\frac{\partial F(t, x)}{\partial t} + \frac{\partial F(t, x)}{\partial x} a(t, x) + \frac{b^2(t, x)}{2} \frac{\partial^2 F(t, x)}{\partial x^2} \right] dt + \frac{\partial F}{\partial x} b(t, x) dW_t. \quad (71)$$

3 MÉTODOS NUMÉRICOS

Até pouco tempo, por causa da complexidade de resolução, devido à escassez de métodos numéricos apropriados e à indisponibilidade de computadores que fossem eficientes o bastante, diversos modelos criados a fim de representar fenômenos físicos evitavam os efeitos estocásticos. Porém, atualmente é imprescindível estudar as equações diferenciais estocásticas, pois estas efetuam uma função de extrema importância nos setores da indústria e ciência, devido à sua aplicabilidade nos modelos empregados, como por exemplo, em finanças, biologia, medicina e outros. Ao acrescentar um ou mais elementos aleatórios à equação diferencial determinista, não se tem mais uma equação diferencial ordinária (EDO) e sim uma equação diferencial estocástica (EDE) [22, 23].

Para uma análise mais aprofundada das equações diferenciáveis estocásticas é necessário recorrer a simulações numéricas, pois é de tamanha dificuldade tratar analiticamente as EDEs dadas as peculiaridades do Processo de Wiener. Existem diversos métodos iterativos para integrar sistemas que utilizem EDEs. As simulações mais comuns são feitas com a implementação de algoritmos do tipo Euler-Maruyama. Não só será implementado o método de Euler-Maruyama neste trabalho, também serão implementados os métodos de Euler-Heun, conhecido como preditor-corretor, Milstein, Milstein sem derivada e Runge-Kutta Estocástico. É importante pontuar que os diferentes métodos numéricos fundamentam-se em aproximações discretas da solução da equação diferencial estocástica a ser analisada, solução esta que é contínua. Sua definição, convergência e o esquema iterativo [24, 22, 25] para cada método serão apresentados neste capítulo.

3.1 Expansão da Série Estocástica de Taylor

A função que a fórmula de Taylor exerce na análise numérica é de significativa importância. Através dela é possível aproximar uma função suave ao redor de um ponto, seja qual for a ordem que se deseja obter de precisão com esta fórmula. Nesse sentido, ter uma fórmula de expansão estocástica que possua características similares à expansão de Taylor é muito vantajoso. Uma expansão estocástica que será vista a seguir é a chamada expansão de Itô-Taylor [24, 22].

3.1.1 Expansão de Itô-Taylor

No contexto estocástico, para se determinar uma expansão de Itô-Taylor é necessário supor a seguinte equação diferencial estocástica [24]

$$dX(t) = f(X(t))dt + g(X(t))dW(t), \quad (72)$$

onde as funções f e g são suaves e apresentam uma evolução linear dentro de um dado intervalo.

Considere agora que F é uma função independente do tempo, duas vezes diferenciável e contínua de $X(t)$. Através do lema de Itô, tem-se que

$$dF[X(t)] = \left[f[X(t)] \frac{\partial F[X(t)]}{\partial X} + \frac{1}{2} g^2[X(t)] \frac{\partial^2 F[X(t)]}{\partial X^2} \right] dt + g[X(t)] \frac{\partial F[X(t)]}{\partial X} dW(t). \quad (73)$$

Definindo L^0 e L^1 como operadores, especificados como

$$L^0 \equiv f[X(t)] \frac{\partial}{\partial X} + \frac{1}{2} g^2[X(t)] \frac{\partial^2}{\partial X^2} \quad e \quad L^1 \equiv g[X(t)] \frac{\partial}{\partial X}, \quad (74)$$

pode-se escrever a equação (73) como

$$dF[X(t)] = L^0 F[X(t)]dt + L^1 F[X(t)]dW(t). \quad (75)$$

A forma integral da equação (75) é

$$F[X(t)] = F[X(t_0)] + \int_{t_0}^t L^0 F[X(t')]dt' + \int_{t_0}^t L^1 F[X(t')]dW(t'). \quad (76)$$

Tomando agora $F(x) = x(t)$, a equação (76) escreve-se

$$X(t) = X(t_0) + \int_{t_0}^t f[X(t')]dt' + \int_{t_0}^t g[X(t')]dW(t'), \quad (77)$$

agora, considerando $F(x) = f(x)$, a equação (76) fica

$$f[X(t)] = f[X(t_0)] + \int_{t_0}^t L^0 f[X(t')]dt' + \int_{t_0}^t L^1 f[X(t')]dW(t'), \quad (78)$$

por fim, tornando $F(x) = g(x)$, a equação (76) se altera para

$$g[X(t)] = g[X(t_0)] + \int_{t_0}^t L^0 g[X(t')]dt' + \int_{t_0}^t L^1 g[X(t')]dW(t'). \quad (79)$$

Substituindo as equações (78) e (79) na equação (77) para desenvolver um processo

iterativo, origina-se

$$\begin{aligned}
X(t) = X(t_0) + \int_{t_0}^t \left(f[X(t_0)] + \int_{t_0}^{t'_1} L^0 f[X(t'_2)] dt'_2 + \int_{t_0}^{t'_1} L^1 f[X(t'_2)] dW(t'_2) \right) dt'_1 \\
+ \int_{t_0}^t \left(g[X(t_0)] + \int_{t_0}^{t'_1} L^0 g[X(t'_2)] dt'_2 + \int_{t_0}^{t'_1} L^1 g[X(t'_2)] dW(t'_2) \right) dW(t'_1).
\end{aligned} \tag{80}$$

Nota-se que

$$\begin{aligned}
L^0 f &= f[X(t)] \frac{\partial f}{\partial x} + \frac{1}{2} g^2[X(t)] \frac{\partial^2 f}{\partial x^2} = f f' + \frac{1}{2} g^2 f'' \\
L^0 g &= f g' + \frac{1}{2} g^2 g'' \\
L^1 f &= g f' \\
L^1 g &= g g'.
\end{aligned} \tag{81}$$

Separando na equação (80) os termos constantes dos restantes contendo integrais duplas, temos

$$X(t) = X(t_0) + f[X(t_0)] \int_{t_0}^t dt'_1 + g[X(t_0)] \int_{t_0}^t dW(t'_1) + R, \tag{82}$$

onde R representa os termos restantes que apresentam integrais duplas

$$\begin{aligned}
R \equiv \int_{t_0}^t \int_{t_0}^{t'_1} L^0 f[X(t'_2)] dt'_2 dt'_1 + \int_{t_0}^t \int_{t_0}^{t'_1} L^1 f[X(t'_2)] dW(t'_2) dt'_1 \\
+ \int_{t_0}^t \int_{t_0}^{t'_1} L^0 g[X(t'_2)] dt'_2 dW(t'_1) + \int_{t_0}^t \int_{t_0}^{t'_1} L^1 g[X(t'_2)] dW(t'_2) dW(t'_1).
\end{aligned} \tag{83}$$

Da equação (82) temos então

$$X(t) = X(t_0) + f[X(t_0)](t - t_0) + g[X(t_0)][W(t) - W(t_0)] + R, \tag{84}$$

de forma que temos separado termos (integrando constantes) a ordem linear ou menor em t .

A essência do método que estamos usando para chegar na Expansão de Itô-Taylor é iterar repetidamente, isto é, substituir reiteradamente as equações (78) e (79) para obter integrandos constantes em ordem cada vez maior de dt , como corresponde a expansão de Taylor. Assim sendo, vamos substituir agora $g[X(t)]$ dado pela equação (79) na última integral dupla em R da equação (83)

$$\begin{aligned}
\int_{t_0}^t \int_{t_0}^{t'_1} L^1 g[X(t'_2)] dW(t'_2) dW(t'_1) = \int_{t_0}^t \int_{t_0}^{t'_1} \left(L^1 g[X(t_0)] + \int_{t_0}^{t'_2} L^1 L^0 g[X(t'_3)] dt'_3 \right. \\
\left. + \int_{t_0}^{t'_2} L^1 L^1 g[X(t'_3)] dW(t'_3) \right) dW(t'_2) dW(t'_1),
\end{aligned} \tag{85}$$

substituindo em (82) e considerando (84) temos

$$X(t) = X(t_0) + f[X(t_0)](t - t_0) + g[X(t_0)] [W(t) - W(t_0)] \\ + \int_{t_0}^t \int_{t_0}^{t'_1} g[X(t_0)]g'[X(t_0)]dW(t'_2)dW(t'_1) + \tilde{R}, \quad (86)$$

onde \tilde{R} é dado por

$$\tilde{R} \equiv \int_{t_0}^t \int_{t_0}^{t'_1} L^0 f[X(t'_2)]dt'_2dt'_1 + \int_{t_0}^t \int_{t_0}^{t'_1} L^1 f[X(t'_2)]dW(t'_2)dt'_1 \\ + \int_{t_0}^t \int_{t_0}^{t'_1} L^0 g[X(t'_2)]dt'_2dW(t'_1) + \int_{t_0}^t \int_{t_0}^{t'_1} \int_{t_0}^{t'_2} L^1 L^0 g[X(t'_3)]d(t'_3)dW(t'_2)dW(t'_1) \\ + \int_{t_0}^t \int_{t_0}^{t'_1} \int_{t_0}^{t'_2} L^1 L^1 g[X(t'_3)]dW(t'_3)dW(t'_2)dW(t'_1). \quad (87)$$

Vamos trabalhar com o quarto termo da direita na equação (86)

$$\int_{t_0}^t \int_{t_0}^{t'_1} g[X(t_0)]g'[X(t_0)]dW(t'_2)dW(t'_1) = g[X(t_0)]g'[X(t_0)] \int_{t_0}^t [W(t'_1) - W(t_0)]dW(t'_1),$$

usando a definição de Integração Estocástica [1] e a fórmula de Itô, chega-se a

$$\int_{t_0}^t [W(t'_1) - W(t_0)]dW(t'_1) = \frac{1}{2}(W(t'_1))^2 \Big|_{t_0}^t - W(t_0)W(t'_1) \Big|_{t_0}^t - \frac{1}{2}(t'_1) \Big|_{t_0}^t \\ = \frac{1}{2} [W(t)^2 - W(t_0)^2] - [W(t_0)W(t) - W(t_0)^2] - \frac{1}{2}(t - t_0) \\ = \frac{1}{2} [W(t) - W(t_0)]^2 - \frac{1}{2}(t - t_0). \quad (88)$$

Assim, chega-se na Expansão de Itô-Taylor para equação (72) na sua forma integral

$$X(t) = X(t_0) + f[X(t_0)] \int_{t_0}^t dt'_1 + g[X(t_0)] \int_{t_0}^t dW(t'_1) + \\ g[X(t_0)]g'[X(t_0)] \left[\frac{1}{2}[W(t) - W(t_0)]^2 - \frac{1}{2}(t - t_0) \right] + \tilde{R}, \quad (89)$$

onde em \tilde{R} , encontram-se os termos de ordem superior em $(t - t_0)$.

A expressão equivalente, realizando as integrações possíveis, é dada por

$$X(t) = X(t_0) + f[X(t_0)](t - t_0) + g[X(t_0)][W(t) - W(t_0)] + \\ \frac{1}{2}gg'[W(t) - W(t_0)]^2 - (t - t_0) + \tilde{R}. \quad (90)$$

3.2 Ordem de convergência Forte

Uma aproximação discretizada no tempo Y_δ de um processo contínuo Y , onde δ é o máximo acréscimo de tempo do processo de discretização, é dito ser de ordem de convergência Forte γ para Y se para qual for o horizonte de tempo fixo T seja certo que [26]

$$\mathbb{E} | Y_\delta(T) - Y(T) | \leq C\delta^\gamma, \forall \delta < \delta_0, \quad (91)$$

em que C é uma constante que independe de δ e $\delta_0 > 0$. A metodologia é similar às aproximações das trajetórias em sistemas dinâmicos que não são estocásticos.

3.3 Ordem de convergência Fraca

A ordem fraca de convergência pode ser definida quando Y_δ se aproxima, digamos "fracamente" de ordem β para Y . Seja qual for o horizonte de tempo fixo T e qualquer função que seja contínua e diferenciável g de crescimento polinomial, é correto afirmar que

$$| \mathbb{E}g(Y(T)) - \mathbb{E}g(Y_\delta(T)) | \leq C\delta^\beta, \forall \delta < \delta_0, \quad (92)$$

em que C é uma constante que independe de δ e $\delta_0 > 0$.

Em geral, sistemas de aproximação que apresentam uma determinada ordem de convergência forte, apresentam uma ordem de convergência fraca maior. Por exemplo, sob determinadas situações que garantam a suavidade nos coeficientes da equação diferencial estocástica, a ordem de convergência teórica para o método de Euler pode ser vista abaixo

- Convergência Forte: $\gamma = \frac{1}{2}$
- Convergência Fraca: $\beta = 1$

3.4 Esquemas de Integração Numérica

Partindo da expansão de Itô-Taylor, podem ser definidos diferentes métodos de integração numérica da equação diferencial estocástica (72). Considerando o intervalo $[t_i, t_{i+1}]$ e escolhendo $t_0 = t_i$ e $t = t_{i+1}$, temos $\Delta t = t_{i+1} - t_i$ e $\Delta W_i = W(t_{i+1}) - W(t_i)$,

da equação (89) temos

$$X(t_{i+1}) = X(t_i) + f(X(t_i))\Delta t + g(X(t_i))\Delta W_i + \frac{1}{2}g(X(t_i))g'(X(t_i)) [(\Delta W_i)^2 - \Delta t] + \tilde{R}. \quad (93)$$

Podemos agora estender para o domínio $[0, T]$. Considerando uma discretização do tempo $0 = t_0 < t_1 < \dots < t_N = T$ e a condição inicial $X(t_0) = X_0$. As variáveis aleatórias ΔW_i são independentes e normalmente distribuídas $N(0, \Delta t)$

Vamos apresentar em continuação, vários esquemas diferentes de integração para equações diferenciais estocásticas.

3.4.1 Método de Euler-Maruyama

A aproximação numérica estocástica mais simples é a do método de Euler-Maruyama, sendo essencial que a descrição do problema esteja caracterizada com a utilização do Esquema de Itô para a definição da equação diferencial estocástica.

Dada uma solução do processo de Itô $X(t)$, em $0 \leq t \leq T$, da equação diferencial estocástica definida na equação (72), onde f é chamado de coeficiente de arrasto (*drift*), g é chamado de coeficiente de difusão e $W(t)$ é chamado de processo de Wiener [22, 25] e considerando o valor inicial $X_{t_0} = X_0$ e a discretização de $N + 1$ elementos, dados no intervalo de $[0, T]$, $0 = t_0 < t_1 < \dots < t_N = T$. Tem-se que a aproximação de Euler-Maruyama ou método de Euler-Maruyama de X é um processo estocástico de tempo contínuo que obedece o seguinte esquema iterativo

$$Y_{n+1} = Y_n + f(Y_n)h_n + g(Y_n)\Delta W_n, \quad (94)$$

para $n = 0, 1, 2, \dots, N - 1$, com condição inicial $Y_0 = X_0$. Na mesma equação, $\Delta W_n = W(t_{n+1}) - W(t_n)$ e $h_n = t_{n+1} - t_n$ é o tamanho do passo. Vamos denotar, $f(Y_n) = f_n$ e $g(Y_n) = g_n$.

Os resultados numéricos apresentam uma certa imprevisibilidade, a não ser que se utilize um tamanho de passo pequeno, pois a ordem de convergência [1] para o método de Euler-Maruyama, assim como do método de Euler-Heun que será visto mais adiante é baixa:

- Ordem Forte de Convergência: $\frac{1}{2}$
- Ordem Fraca de Convergência: 1

Com a adição de mais um termo oriundo da expansão estocástica de Taylor, alcança-se uma ordem forte de convergência de 1 do esquema de convergência chamado

de Esquema de Milstein.

3.4.2 Método de Euler-Heun

Este método também conhecido de preditor-corretor é usado quando o problema está definido com a convenção da integral de Stratonovich. Neste caso, o método de Euler-Maruyama não é mais válido, pois somente é usado para as EDEs que utilizam do esquema de Itô.

Como na definição de Stratonovich, a função é avaliada no valor médio de cada intervalo da partição considerada, é inserido um termo preditor, com a finalidade de corrigir a equação para a próxima iteração. A aproximação é dada pelo seguinte esquema iterativo

$$Y_{n+1} = Y_n + f_n h_n + \frac{1}{2} g_n \Delta W_n + \frac{1}{2} g(\hat{Y}_n) \Delta W_n, \quad (95)$$

onde \hat{Y}_n é o termo preditor, dado por:

$$\hat{Y}_n = Y_n + g_n \Delta W_n. \quad (96)$$

3.4.3 Método de Milstein

O esquema de Milstein emprega o lema de Itô com a adição do termo de segunda ordem, de forma a ampliar sua precisão. O método de Milstein apresenta ordem forte e fraca de convergência iguais a 1 para a solução da EDE. O esquema de Taylor de ordem forte 1 apresentado pelo esquema Milstein é [25].

$$Y_{n+1} = Y_n + f_n h + g_n \Delta W_n + \frac{1}{2} g_n g'_n [(\Delta W_n)^2 - h], \quad (97)$$

$$Y_{n+1} = Y_n + f_n h + g_n \Delta W_n + \frac{1}{2} g_n g'_n (\Delta W_n)^2, \quad (98)$$

$$\Delta W_n = [W_{t+h} - W_t] \sim \sqrt{h} \mathcal{N}(0, 1), \quad (99)$$

em que, g'_n é a primeira derivada da função g_n .

O método iterativo apresentado na equação (97) deverá ser utilizado em EDEs definidas segundo a integral de Itô, já a equação (98), para as que sejam definidas segundo Stratonovich. Repare que se o ruído for aditivo, quando g_n for uma constante ($g'_n = 0$) e não função de Y_n , ambas as interpretações de Itô e Stratonovich serão iguais.

3.4.4 Método de Milstein Sem Derivada

O método anterior é desvantajoso, na medida que necessita da especificação analítica da primeira deriva da função $g(Y_n)$. Expressão esta que rapidamente pode apresentar um altíssimo nível de complexidade. O seguinte esquema iterativo faz uma aproximação desta derivada, graças ao auxílio de uma abordagem Runge-Kutta

$$Y_{n+1} = Y_n + f_n h + g_n \Delta W_n + \frac{1}{2\sqrt{h}} [g(\tilde{Y}) - g_n] [(\Delta W_n)^2 - h], \quad (100)$$

$$Y_{n+1} = Y_n + f_n h + g_n \Delta W_n + \frac{1}{2\sqrt{h}} [g(\tilde{Y}) - g_n] (\Delta W_n)^2, \quad (101)$$

$$\tilde{Y}_n = Y_n + f_n h + g_n \sqrt{h}, \quad (102)$$

$$\Delta W_n = [W_{t+h} - W_t] \sim \sqrt{h} \mathcal{N}(0, 1). \quad (103)$$

Similarmente ao método de Milstein, o método apresentado na equação (100) deverá ser utilizado em EDEs definidas segundo o esquema de Itô. Agora, o da equação (101), nas definidas segundo esquema de Stratonovich. De igual modo, caso $g(t)$ não dependa do estado do sistema, ambas as interpretações Itô e Stratonovich serão equivalentes.

3.4.5 Método de Runge-Kutta Estocástico

Ordens de convergência maiores podem ser conseguidas, incrementando termos da expansão estocástica de Taylor no esquema de Milstein. A implementação que se segue, possibilita obter uma ordem forte de convergência de valor igual a $\frac{3}{2}$. Essa é a maior ordem forte de Runge-Kutta que pode ser alcançada, permanecendo sua estrutura a mais simples.

Para descrever a expansão da série Itô-Taylor numa configuração multi-dimensional e multi-processos de Wiener, será utilizada a seguinte terminologia para uma integral múltipla de Stratonovich [23]

$$J_{(j_1, j_2, \dots, j_l), t} = \int_0^t \int_0^{s_1} \dots \int_0^{s_{l-1}} \circ dW_{s_1}^{j_1} \dots \circ dW_{s_l}^{j_l} \quad (104)$$

onde $j_i \in \{0, 1, \dots, d\}$ para d processos de Wiener e $dW_{s_i}^0 \equiv ds_i$

Introduz-se então, duas variáveis aleatórias J_1 e J_{10}/h na formulação do método. Estas variáveis são definidas no intervalo $[t, t + h]$ e podem ser facilmente geradas à medida que são correlacionados. Amostrando dois números aleatórios independentes M_1 e M_2 de

$\mathcal{N}(0, 1)$, é possível calcular J_1 e J_{10}/h e obter o seguinte resultado

$$J_1 = \sqrt{h}M_1 \quad (105)$$

$$J_{10}/h = \frac{\sqrt{h}}{2} \left(M_1 + \frac{M_2}{\sqrt{3}} \right) \quad (106)$$

Isolando o termo J_{10} da equação (106)

$$J_{10} = \frac{1}{2} \left(M_1 + \frac{M_2}{\sqrt{3}} \right) h^{\frac{3}{2}} \quad (107)$$

Chamaremos as variáveis aleatórias J_1 e J_{10} de ΔW_n e ΔZ_n , respectivamente. Sendo ΔZ_n uma distribuição gaussiana com média igual a zero, variância igual a $\frac{1}{3}h^3$ e correlação $E(\Delta W_n \Delta Z_n) = \frac{1}{2}h^2$.

O método iterativo de Runge-Kutta estocástico de ordem forte de convergência $\frac{3}{2}$ é dado por

$$\begin{aligned} \Delta Y_{n+1} = & Y_n + f_n h + g_n \Delta W_n + \frac{1}{2} g_n g'_n [(\Delta W_n)^2 - h] \\ & + f'_n g_n \Delta Z_n + \frac{1}{2} \left[f_n f'_n + \frac{1}{2} g_n^2 f''_n \right] h^2 + \left[f_n g'_n + \frac{1}{2} g_n^2 g''_n \right] [\Delta W_n h - \Delta Z_n] \\ & + \frac{1}{2} g_n [g_n g''_n + (g'_n)^2] \left[\frac{1}{3} (\Delta W_n)^2 - h \right] \Delta W_n. \end{aligned} \quad (108)$$

4 RESULTADOS

Conforme dito anteriormente, a principal motivação deste trabalho é o estudo do movimento produzido através de uma equação de Langevin e sua simulação numérica. Para isso, trabalharemos com um sistema conhecido, um oscilador super-amortecido sob efeito de um ruído multiplicativo no sistema.

Vamos analisar o movimento de um objeto de massa m que está preso a uma mola com constante elástica k , inicialmente em sua posição de equilíbrio. Após o objeto sair de sua posição de equilíbrio, supõe-se que exista uma força de dissipação proporcional a velocidade e uma força restauradora provocada pela mola, oposta ao movimento. Tais forças podem ser visualizadas no esquema representado na Figura 4.

Portanto, a equação de movimento para este sistema, respeita a segunda lei de Newton (Princípio Fundamental da Dinâmica)

$$m \frac{d^2x}{dt^2} = -kx - b \frac{dx}{dt}. \quad (109)$$

Conforme pode ser visto na Figura 4, x é o deslocamento do objeto em relação à sua posição de equilíbrio, já $-kx$ é a força restauradora exercida sobre a mola e $-bv$ é a força retardadora, onde b é a constante de amortecimento. Organizando os termos da equação (109)

$$m \frac{d^2x}{dt^2} + b \frac{dx}{dt} + kx = 0 \quad (110)$$

A solução para esta equação diferencial será suposta na forma exponencial do tipo $x(t) = Ae^{rt}$, satisfazendo, para qualquer que seja o t , a seguinte equação

$$m \frac{d^2}{dt^2}(Ae^{rt}) + b \frac{d}{dt}(Ae^{rt}) + k(Ae^{rt}) = 0. \quad (111)$$

Derivando e dividindo todos os termos da equação (111) por Ae^{rt} , tem-se:

$$mr^2 + br + k = 0, \quad (112)$$

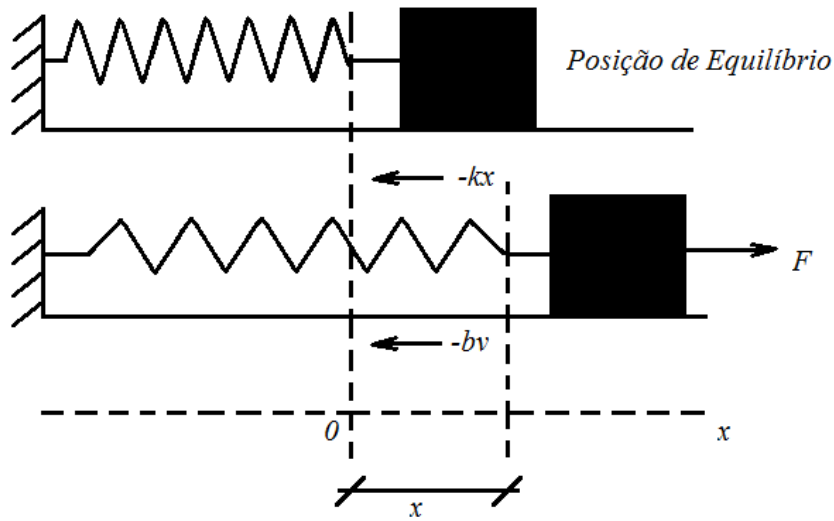
Resolvendo esta equação de segundo grau, encontram-se duas raízes, as quais são:

$$r_{1,2} = \frac{-b \pm \sqrt{b^2 - 4km}}{2m} \quad (113)$$

Sendo assim, existem três casos diferentes fisicamente:

- $b^2 > 4km$: Super-amortecido

Figura 4 - Oscilador Harmônico



Legenda: A imagem superior ilustra um Oscilador Harmônico em sua posição de equilíbrio. Na inferior é exibido como as forças atuam no movimento, se o oscilador for tirado de sua posição de equilíbrio por uma força F .

- $b^2 = 4km$: Criticamente Amortecido
- $b^2 < 4km$: Sub-amortecido

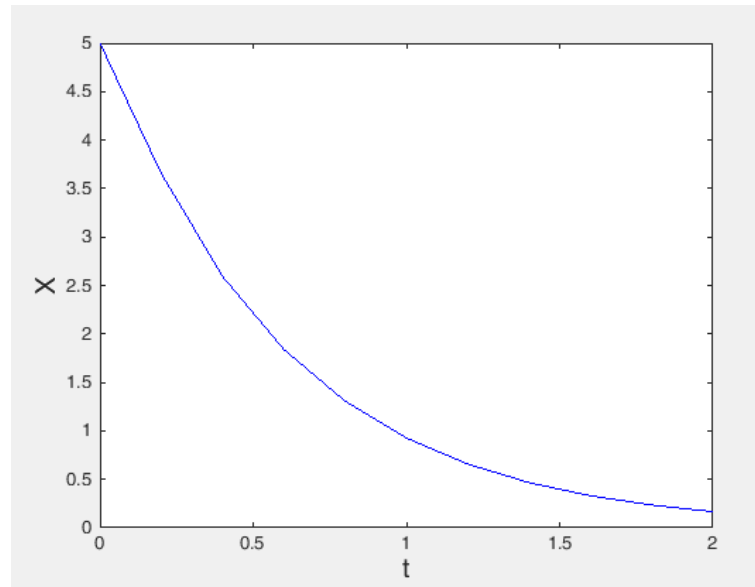
No presente trabalho, será analisado apenas o primeiro caso, que é o caso Super-amortecido. Assim, desprezando o termo de inércia ficaremos com uma Equação Diferencial de 1º Grau.

Ambas as raízes da equação (112) são negativas, pois partindo que $b^2 - 4km > 0$, que é o caso escolhido, a análise de r_1 e r_2 pode ser feita da seguinte forma

$$\begin{aligned}
 \frac{-b \pm \sqrt{b^2 - 4km}}{2m} &< 0 \\
 -b \pm \sqrt{b^2 - 4km} &< 0 \\
 \pm \sqrt{b^2 - 4km} &< b \\
 (\pm \sqrt{b^2 - 4km})^2 &< b^2 \\
 b^2 - 4km &< b^2 \\
 -4km &< 0,
 \end{aligned} \tag{114}$$

onde conclui-se que $r_{1,2} < 0$

Figura 5 - Oscilador Harmônico Super-amortecido



Legenda: Oscilador Harmônico Super-amortecido
caminhando para o equilíbrio, em torno de zero.

A solução geral, então poderá ser escrita da seguinte forma

$$x = A_1 e^{-|r_1|t} + A_2 e^{-|r_2|t}. \quad (115)$$

Como o movimento é dado por uma exponencial elevado a um número negativo, o movimento não é oscilatório e a medida que o tempo passa, o deslocamento de x decai exponencialmente até chegar a zero. Conforme pode ser verificado na Figura 5.

Sendo assim, no decorrer do tempo, o sistema super-amortecido vai para o equilíbrio assintoticamente. Para o real interesse proposto, introduzindo o ruído ao sistema, ficaremos com a seguinte equação

$$-b \frac{dx}{dt} = kx - D\eta(t), \quad (116)$$

onde D mede a intensidade do ruído gaussiano branco $\eta(t)$.

Isolando a velocidade, tem-se uma equação geral de 1ª ordem

$$\frac{dx}{dt} = -\frac{k}{b}x + \frac{D}{b}\eta(t). \quad (117)$$

É válido ressaltar que o ruído branco gaussiano além de possuir as propriedades descritas na equação (6), ele também apresenta uma distribuição de probabilidade gaussi-

ana. Outra consideração importante é que tanto a constante de amortecimento quanto a constante que mede a intensidade do ruído branco deverão ser diferentes de zero. O ruído gaussiano $\eta(t)$ é chamado de branco, pois seu espectro é formado pela presença de todas as frequências possíveis e com intensidades iguais, similarmente à luz branca [21].

É importante salientar que D poderá ser uma constante, independente do estado do sistema. Sendo assim, o ruído será chamado de aditivo que é o caso mais simples a ser estudado. Por outro lado, caso D não seja uma constante e sim uma função que dependa do estado do sistema, o ruído será chamado de multiplicativo. Tais sistemas apresentam dificuldades específicas no trato matemático, em especial na complexidade de definições e formalismos gerais para poder abordá-los.

Numa primeira abordagem, foi utilizado um modelo simples com ruído aditivo, do qual é conhecido o seu comportamento, para verificar o código e validar o modelo. Contudo, o nosso real interesse é trabalhar com ruído multiplicativo e controlar as variações do modelo.

4.1 Modelo Físico Proposto

O modelo proposto descreve um sistema físico cujo comportamento é característico de um oscilador harmônico super-amortecido. Sua função potencial $U(x)$ é dada por

$$U(x) = \frac{\omega}{2}x^2, \quad (118)$$

onde ω é a frequência característica do oscilador harmônico.

O coeficiente de *difusão* $g(x)$ será dado pela função

$$g(x) = 1 + \lambda^2x^2, \quad (119)$$

onde λ é a constante que controla a intensidade do ruído. Nota-se que o valor do λ permite analisar um regime perto do ruído aditivo ($\lambda = 0$).

O coeficiente de *drift* $f(x)$ é dado por

$$f(x) = -\frac{g^2(x)}{2} \frac{\partial U}{\partial x} = -\frac{\omega}{2}x(1 + \lambda^2x^2)^2, \quad (120)$$

garantindo assim que o sistema satisfaz um equivalente da relação de Einstein.

Uma equação diferencial estocástica só é bem definida se for adotada uma prescrição de discretização. Para a solução da equação de Langevin estudada neste trabalho, usaremos as prescrições de Itô e Stratonovich. Veremos em seguida, que a solução da

equação de Langevin depende da prescrição adotada [27].

Vimos anteriormente, que a Equação Diferencial Estocástica de 1ª Ordem escrita na prescrição de Itô é dada por:

$$\frac{dx}{dt} = f(x, t)dt + g(x, t)dW. \quad (121)$$

Por outro lado, para representar a prescrição de Stratonovich utiliza-se o símbolo \circ antes do diferencial dW e modifica-se a função de *drift*, como será visto adiante:

$$\frac{dx}{dt} = \tilde{f}(x, t)dt + g(x, t) \circ dW, \quad (122)$$

em que $\tilde{f}(x, t) = f(x, t) - 1/2g(x, t)\partial g/\partial x$, ou seja, $\tilde{f} = f - (1/2)gg'$.

Substituindo as funções de *drift* $f(x)$ e *difusão* $g(x)$ na equação (121), obtém-se a EDE para o modelo proposto escrita na prescrição de Itô

$$\frac{dx}{dt} = -\frac{\omega}{2}x(1 + \lambda^2x^2)^2 + (1 + \lambda^2x^2)\eta(t). \quad (123)$$

De igual forma, substituindo ambas as funções na equação (122), obtém-se a EDE para o modelo proposto, escrita na prescrição de Stratonovich

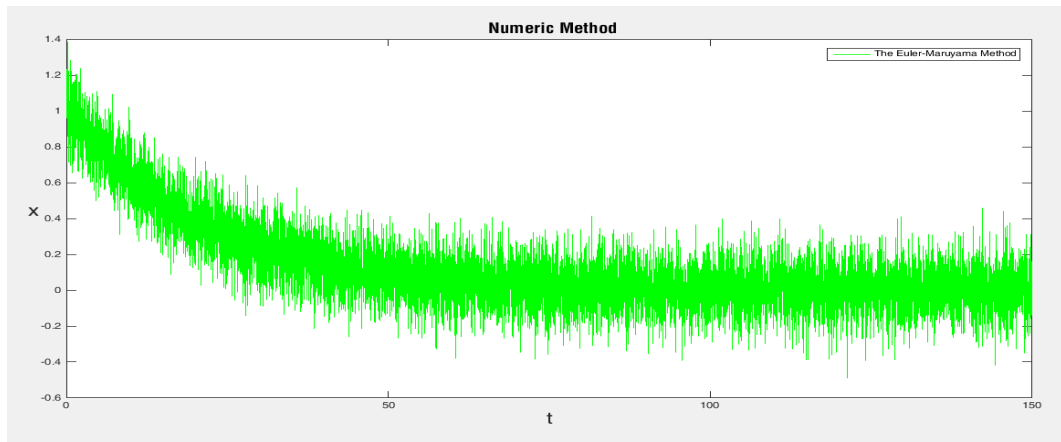
$$\frac{dx}{dt} = -\frac{\omega}{2}x(1 + \lambda^2x^2)^2 - [\lambda^2x(1 + \lambda^2x^2)] + (1 + \lambda^2x^2)\eta(t). \quad (124)$$

A resolução numérica da EDE do modelo proposto, foi simulada computacionalmente utilizando-se diferentes métodos numéricos. Lembrando que para os métodos de Euler-Maruyama, Milstein para Itô e Milstein sem derivada para Itô, foi utilizada a equação (123) cuja prescrição é a de Itô. Já para usar os métodos Euler-Heun, Milstein para Stratonovich, Milstein sem derivada para Stratonovich e Runge-Kutta, empregou-se a equação (124) escrita na prescrição de Stratonovich.

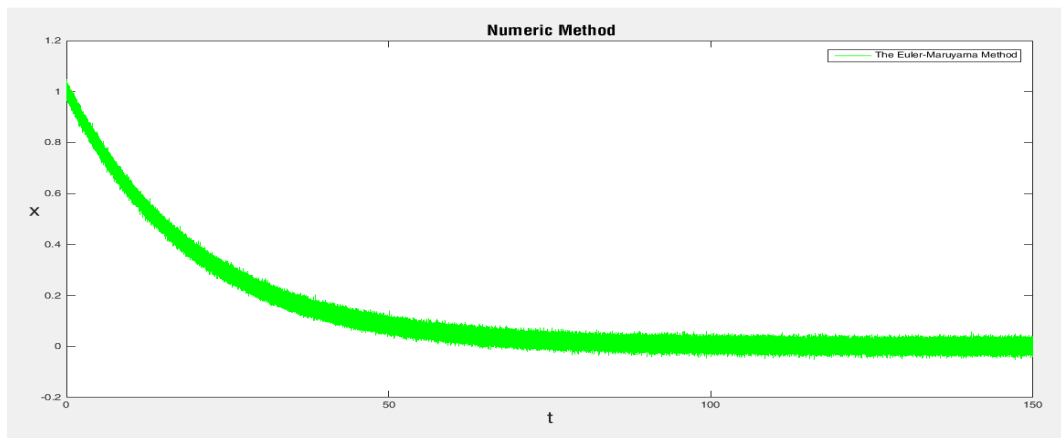
O primeiro modelo abordado utiliza ruído aditivo. Este é o caso mais simples a ser estudado, nessa condição o coeficiente de *difusão* $g(x)$ é uma constante e não depende do estado do sistema físico. O modelo proposto então é um processo estocástico aditivo. Na implementação computacional, considerou-se $g(x) = 1$, fazendo $\lambda = 0$.

Inicialmente, empregando o método numérico de Euler-Maruyama com ruído aditivo, verifica-se que para diferentes tamanhos de passo de tempo, ou seja, para diferentes valores de h há variação na sensibilidade à flutuação. Por exemplo, na Figura 6a, verifica-se que para um $h = 0,015$ existe uma maior sensibilidade ao ruído, exibindo assim uma curva com maiores variações, quanto que para um $h = 0,00015$ consideravelmente pequeno, gráfico visto na Figura 6b, há menores variações, portanto uma menor sensibilidade ao ruído resultando em uma curva mais fina, se comparado a anterior. Em contrapartida,

Figura 6 - Comparação das curvas de diferentes tamanhos de passo (h), utilizando o método de Euler-Maruyama com ruído aditivo e a mesma frequência do Oscilador harmônico.



(a) Método de Euler-Maruyama do modelo proposto com ruído aditivo, para um $h = 0,015$, com $\omega = 0.5$. Neste gráfico, percebe-se uma maior sensibilidade ao ruído. A curva é mais grossa.

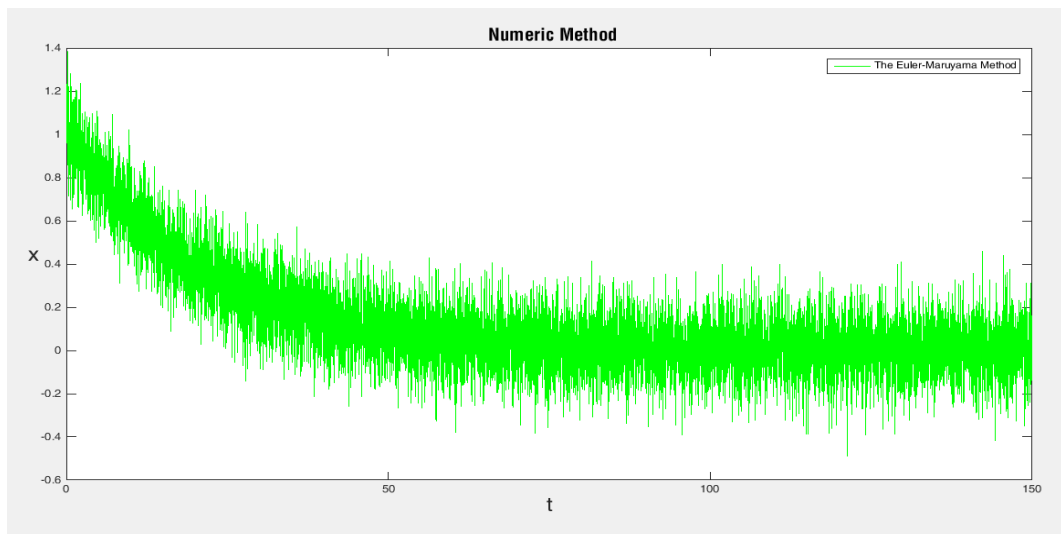


(b) Método de Euler-Maruyama do modelo proposto com ruído aditivo, para um $h = 0,00015$, com $\omega = 0.5$. Neste gráfico, percebe-se uma menor sensibilidade ao ruído. A curva é mais fina.

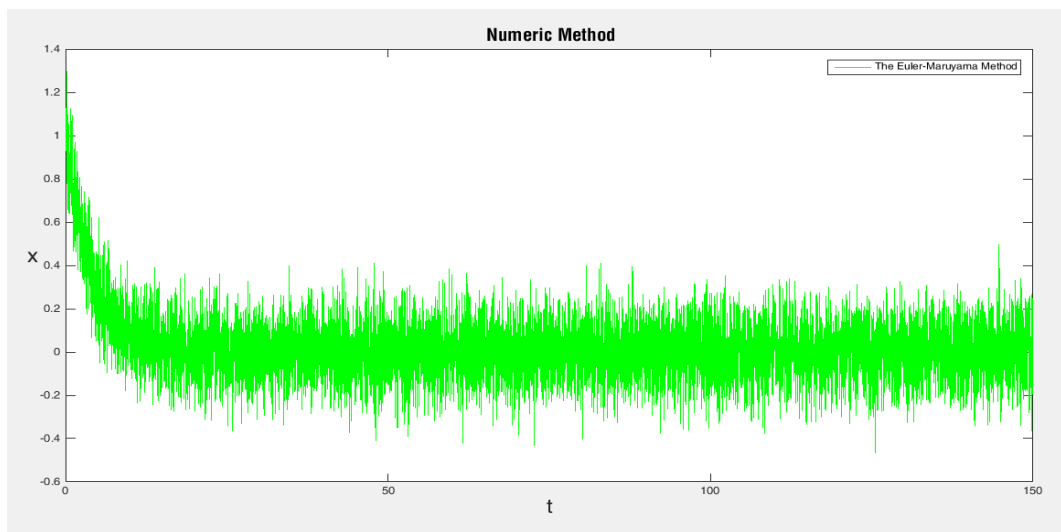
o custo computacional para um h muito pequeno é bastante alto. Quando reduzimos muito o tamanho, aumentamos o número de pontos (elementos) na malha e o tempo do processamento iterativo é diretamente proporcional ao número de elementos. Será visto mais adiante que para o tratamento estatístico dos resultados, utilizou-se muitos dados experimentais. O tratamento estatístico deu-se através da análise dos resultados de cada método numérico. Foram construídas diferentes matrizes, uma para cada método numérico para 100 realizações do ruído. Para $h = 0,015$ o tempo de execução do programa a fim de obter as matrizes foi em média 6,4s. Agora, com $h = 0,00015$ o tempo de execução demorava em média 16min. Essa análise foi feita em um notebook Acer, cujo sistema operacional é Windows 8.1, o processador é Intel Core i3 e memória RAM de 4GB.

Na análise do comportamento do sistema físico proposto para o primeiro modelo

Figura 7 - Método de Euler-Maruyama com ruído aditivo.



(a) Grandezas: $\lambda = 0$ e $\omega = 0.1$



(b) Grandezas: $\lambda = 0$ e $\omega = 0.5$

com ruído aditivo, utilizaram-se diferentes frequências ω do oscilador para o estudo nos diferentes métodos numéricos. Nas Figuras 7a, 8a, 9a, 10a, 11a e 12a estão os resultados gráficos para cada método numérico com ruído aditivo, utilizando a frequência do oscilador harmônico igual a 0.1. Já as Figuras 7b, 8b, 9b, 10b, 11b e 12b utilizou-se a frequência igual a 0.5. Os resultados dos métodos claramente ilustram que aumentando-se o valor da frequência do oscilador, mais rapidamente o sistema atinge o equilíbrio. É importante ressaltar também que neste caso, todos os métodos apresentaram resultados equivalentes e isso já era esperado, visto que o coeficiente de difusão $g(x) = 1$ para o caso aditivo é uma constante, o termo da derivada $g'(x)$ é igual a zero para os métodos numéricos utilizados, o que torna seus esquemas iterativos iguais.

Figura 8 - Método de Euler-Heun com ruído aditivo.

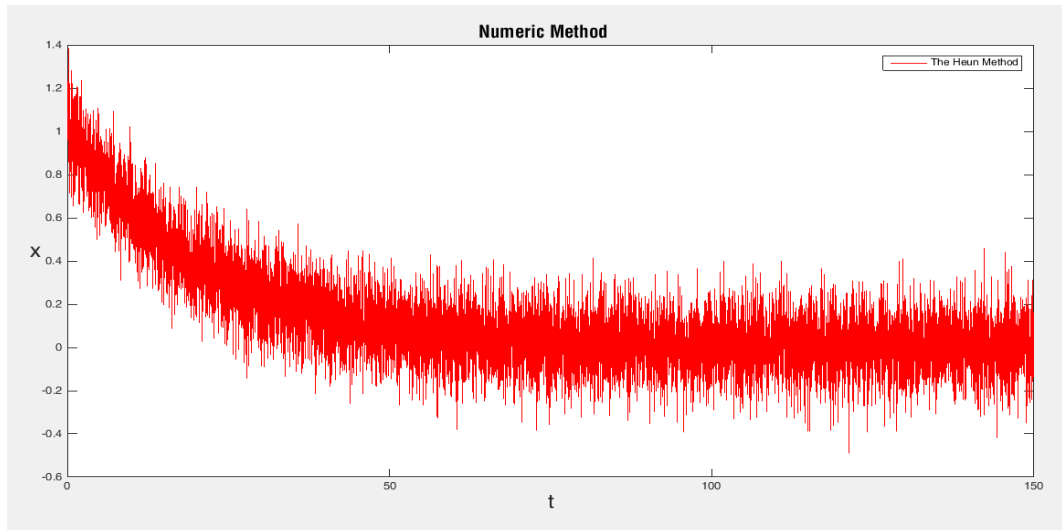
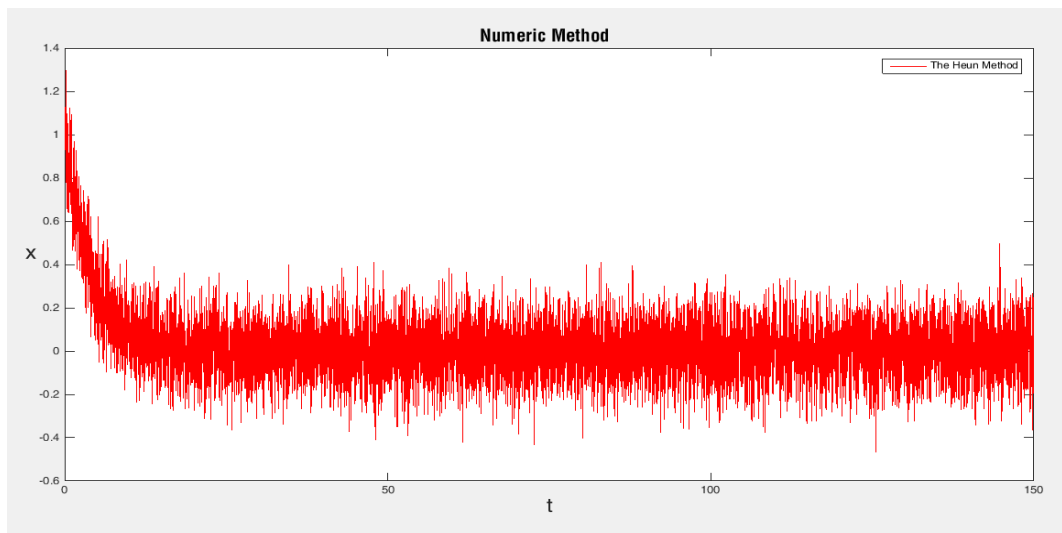
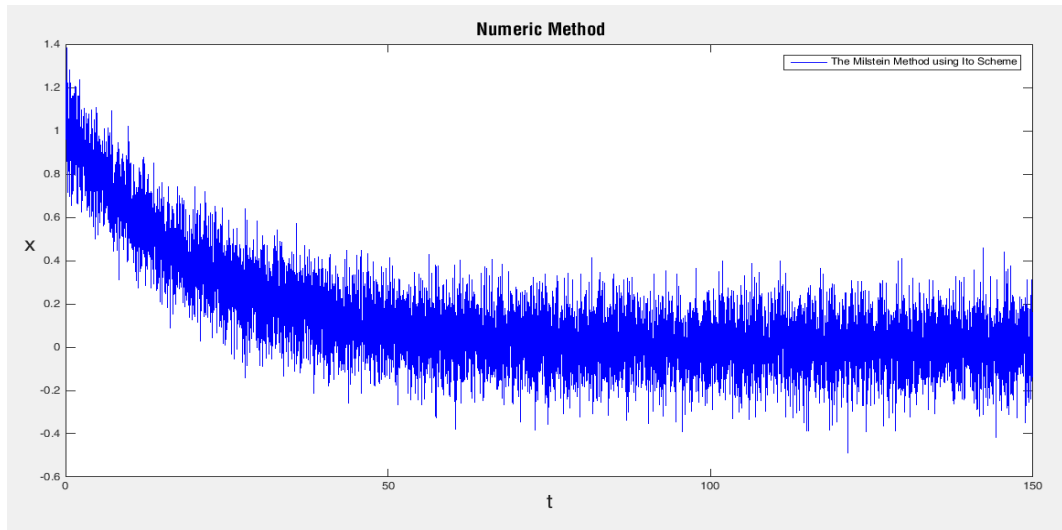
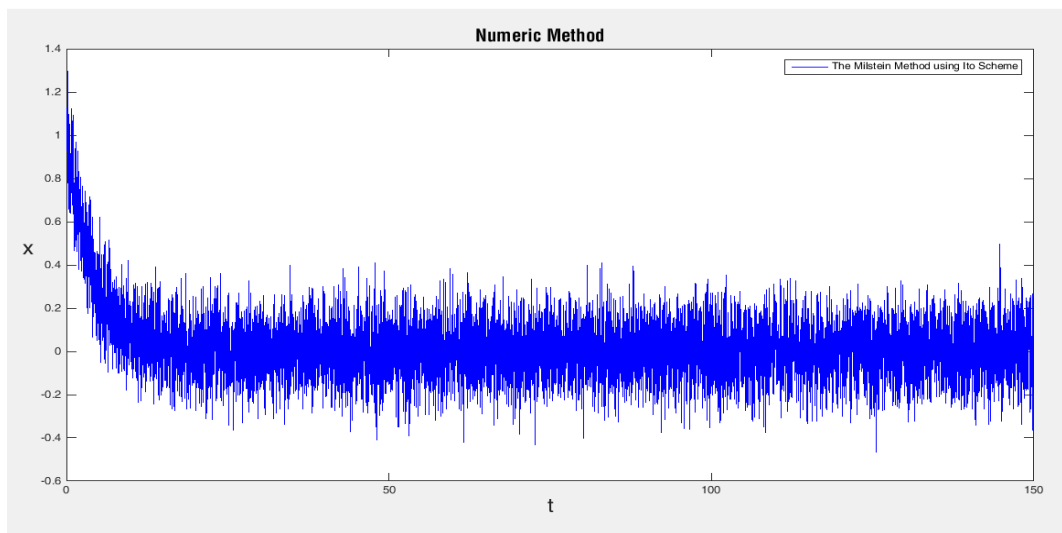
(a) Grandezas: $\lambda = 0$ e $\omega = 0.1$ (b) Grandezas: $\lambda = 0$ e $\omega = 0.5$

Figura 9 - Método de Milstein conforme interpretação de Itô com ruído aditivo.

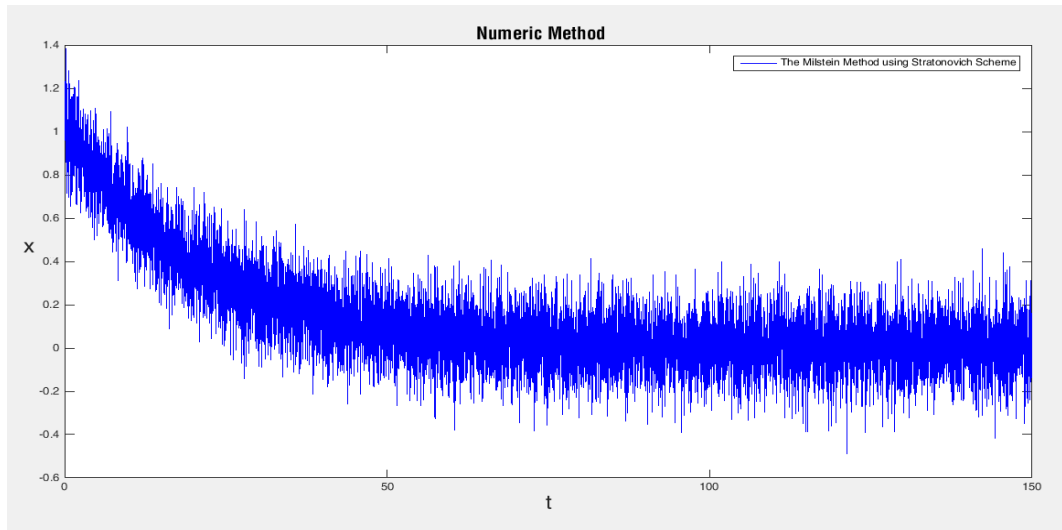


(a) Grandezas: $\lambda = 0$ e $\omega = 0.1$

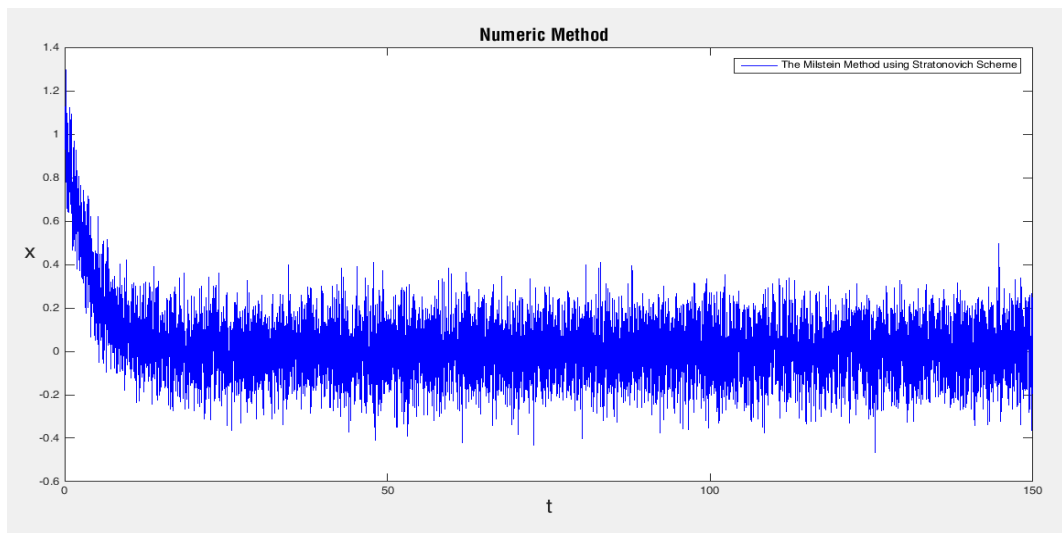


(b) Grandezas: $\lambda = 0$ e $\omega = 0.5$

Figura 10 - Método de Milstein conforme interpretação de Stratonovich com ruído aditivo.



(a) Grandezas: $\lambda = 0$ e $\omega = 0.1$



(b) Grandezas: $\lambda = 0$ e $\omega = 0.5$

Figura 11 - Método de Milstein sem derivada conforme interpretação de Itô com ruído aditivo.

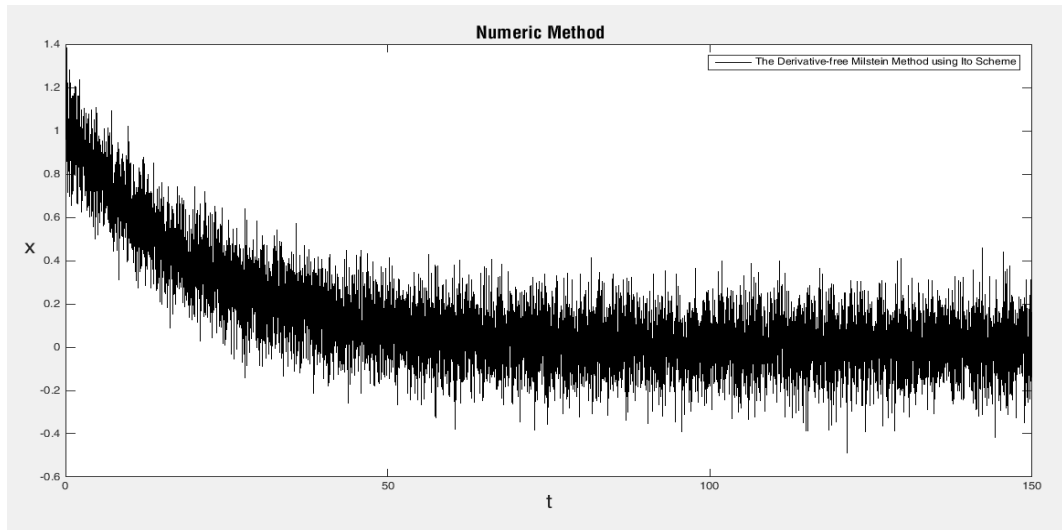
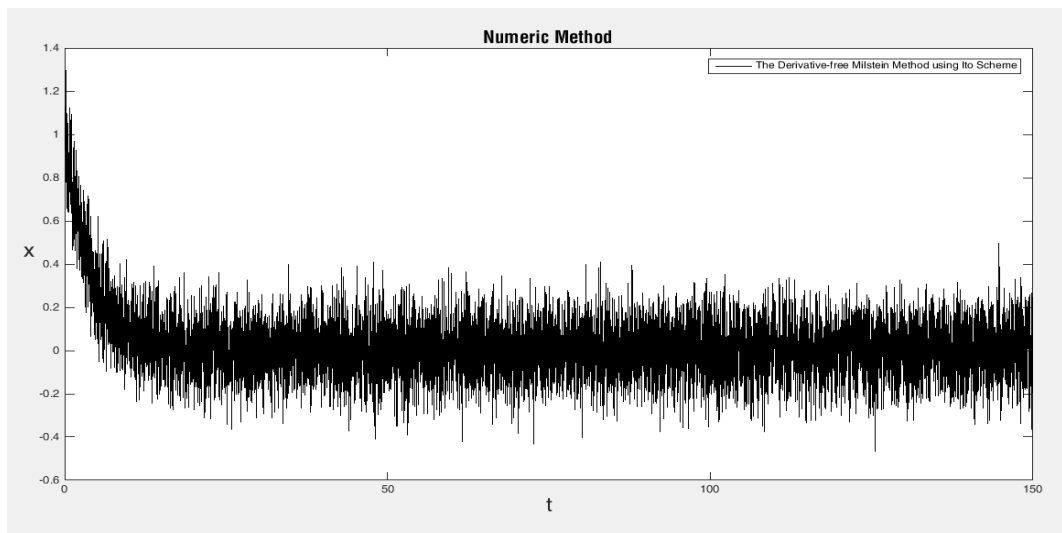
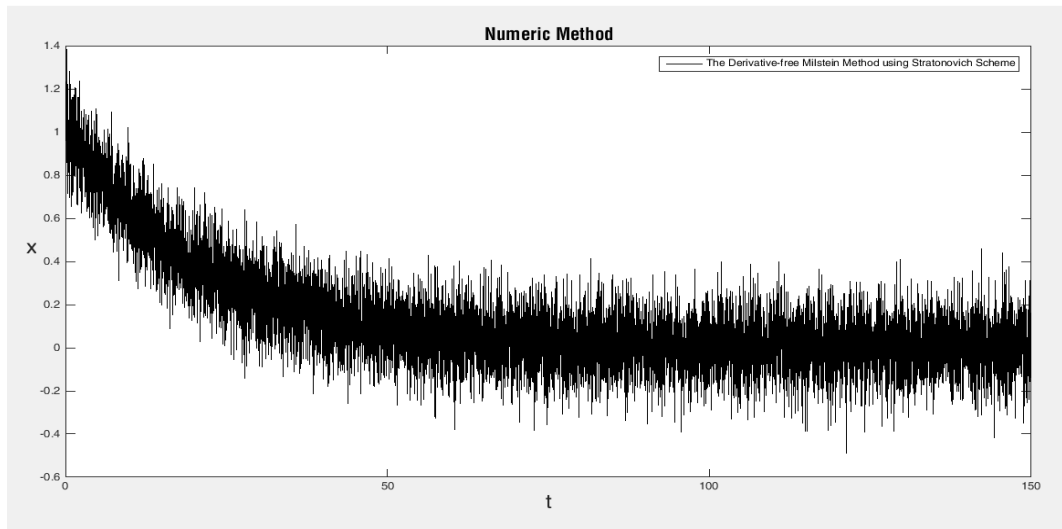
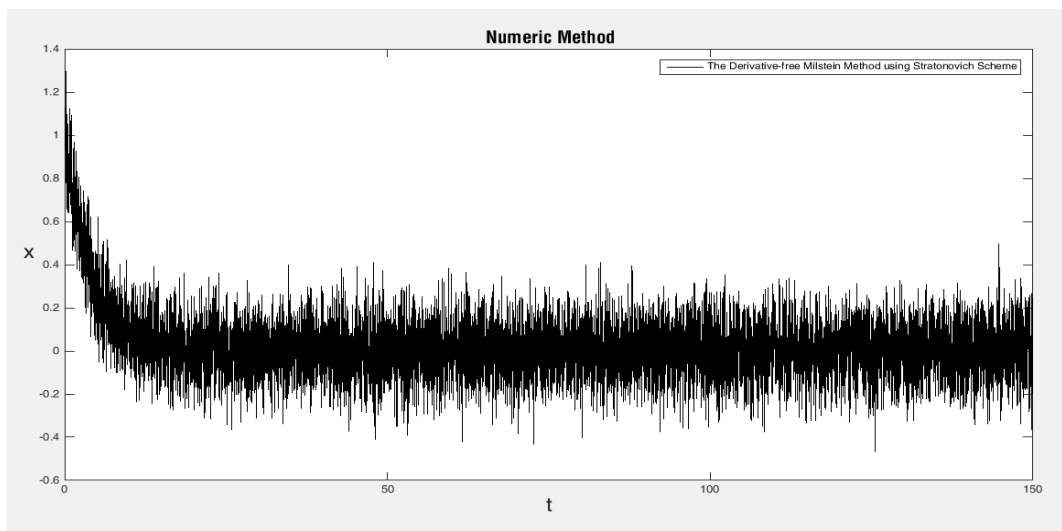
(a) Grandezas: $\lambda = 0$ e $\omega = 0.1$ (b) Grandezas: $\lambda = 0$ e $\omega = 0.5$

Figura 12 - Método de Milstein sem derivada conforme interpretação de Stratonovich com ruído aditivo.



(a) Grandezas: $\lambda = 0$ e $\omega = 0.1$



(b) Grandezas: $\lambda = 0$ e $\omega = 0.5$

Por outro lado, para o segundo modelo estudado, com ruído multiplicativo, o coeficiente de *difusão* $g(x)$ não será mais uma constante e agora dependerá do estado do sistema físico proposto. As equações estocásticas com ruído multiplicativo requerem uma maior atenção, pois necessitam ser tratadas por um cálculo especial, o cálculo estocástico, podendo ser o cálculo de Itô ou Stratonovich. As oscilações para este caso são consideradas como sendo o produto entre a função da variável de estado e a força aleatória. O modelo proposto é um processo estocástico multiplicativo [21].

Na implementação computacional, $g(x)$ não é mais uma constante. Para esse segundo modelo $\lambda \neq 0$ e é a grandeza que representa a intensidade do ruído. Foram escolhidos três valores de λ para analisar o comportamento do oscilador, tais quais: $\lambda = 0.01$, $\lambda = 0.1$ $\lambda = 1.0$. Os resultados gráficos de cada método implementado podem ser visualizados nas Figuras 13a, 14a, 15a, 16a, 17a, 18a e 19a com ruído multiplicativo e frequência do oscilador igual 0.1. Já nas Figuras 13b, 14b, 15b, 16b, 17b, 18b e 19b a frequência do oscilador foi de 0.5. Assim como nos gráficos dos métodos numéricos nas simulações do modelo anterior com ruído aditivo, quando a frequência do oscilador é aumentada, o sistema tende a ir pra sua região de equilíbrio mais rapidamente. Vemos também que é possível controlar a intensidade do ruído, nota-se que com $\lambda = 0.01$ (muito pequeno) chega-se próximo ao modelo com ruído aditivo. A medida que a intensidade do ruído é aumentada, o sistema caminha mais rapidamente para a sua região de equilíbrio.

Figura 13 - Método de Euler-Maruyama com ruído multiplicativo.

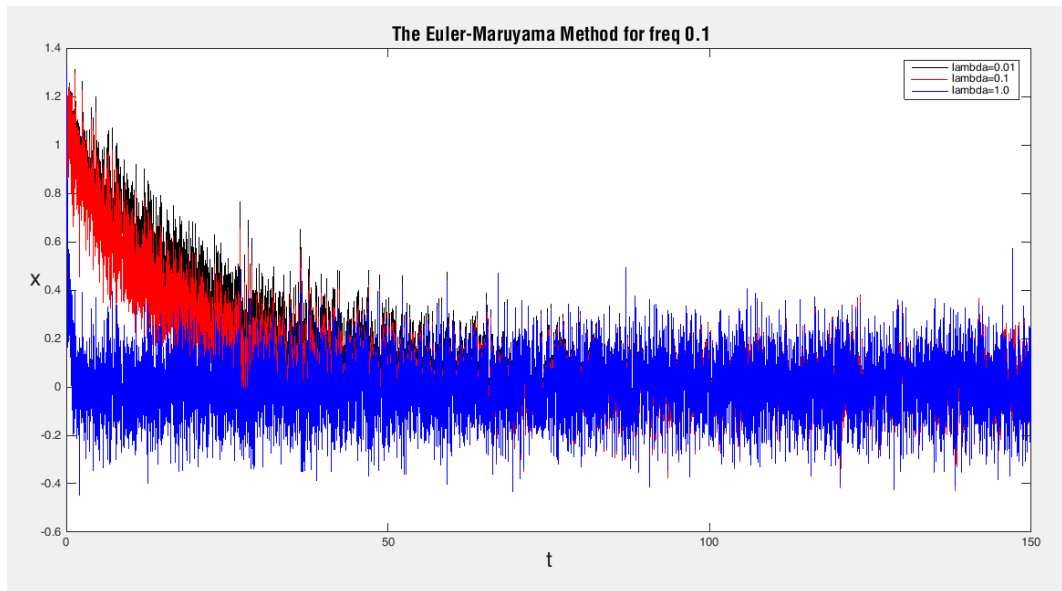
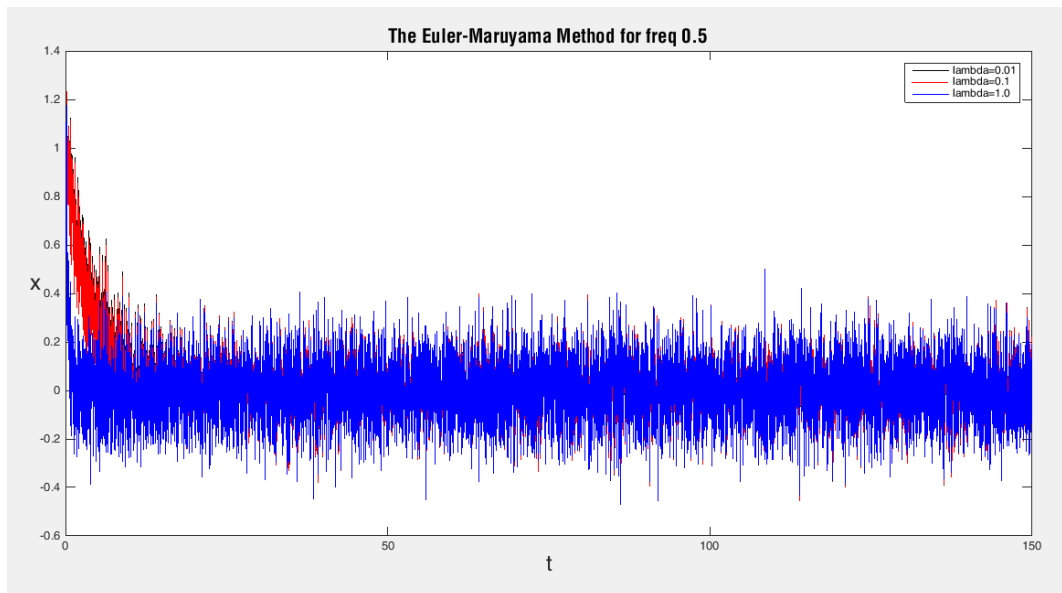
(a) Grandezas: $\lambda = 0.01$, $\lambda = 0.1$, $\lambda = 1.0$ e $\omega = 0.1$ (b) Grandezas: $\lambda = 0.01$, $\lambda = 0.1$, $\lambda = 1.0$ e $\omega = 0.5$

Figura 14 - Método de Euler-Heun com ruído multiplicativo.

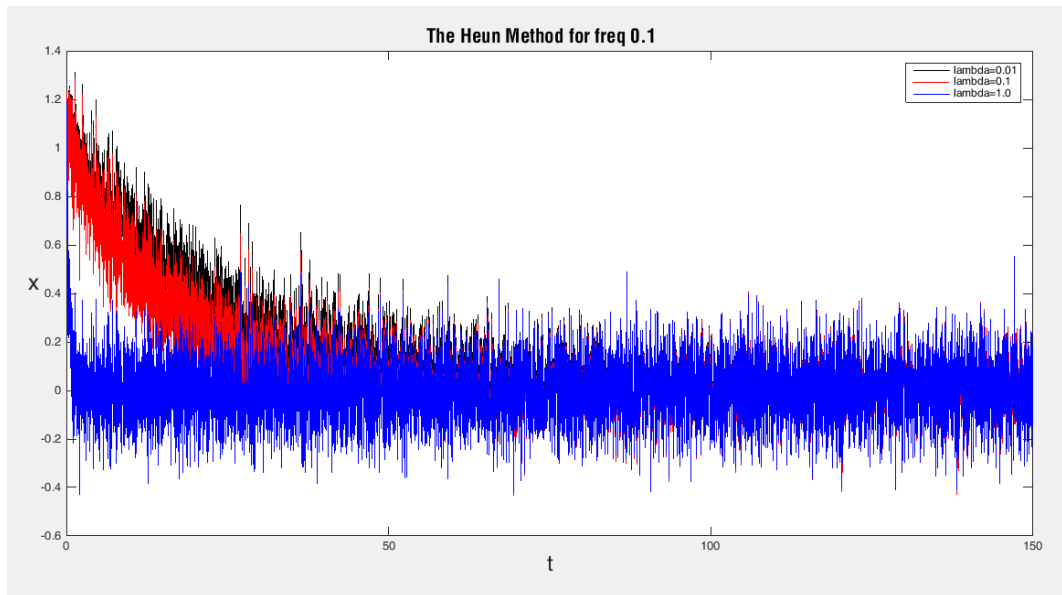
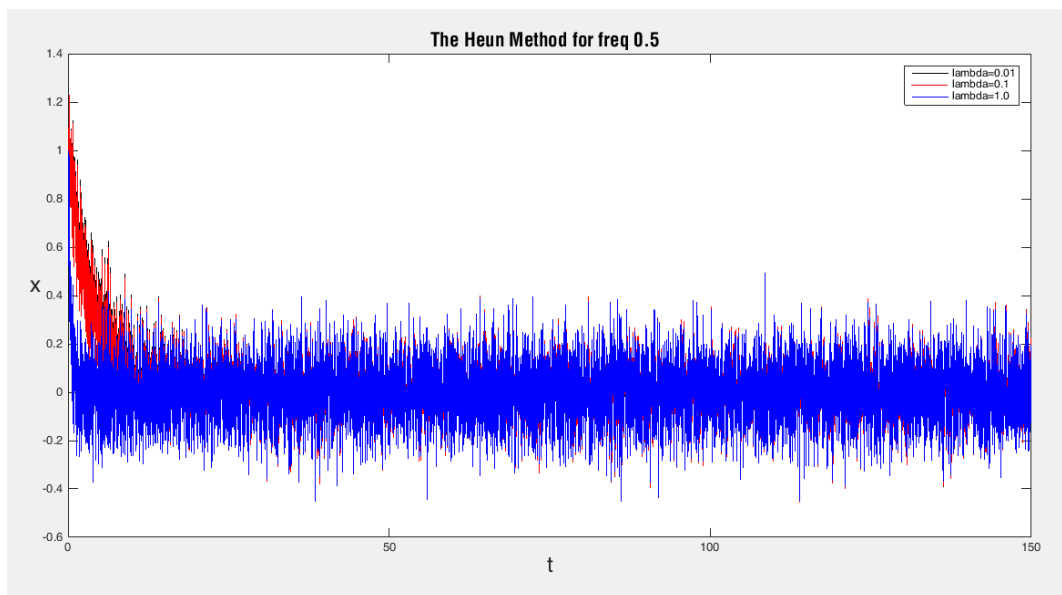
(a) Grandezas: $\lambda = 0.01$, $\lambda = 0.1$, $\lambda = 1.0$ e $\omega = 0.1$ (b) Grandezas: $\lambda = 0.01$, $\lambda = 0.1$, $\lambda = 1.0$ e $\omega = 0.5$

Figura 15 - Método de Milstein conforme interpretação de Itô com ruído multiplicativo.

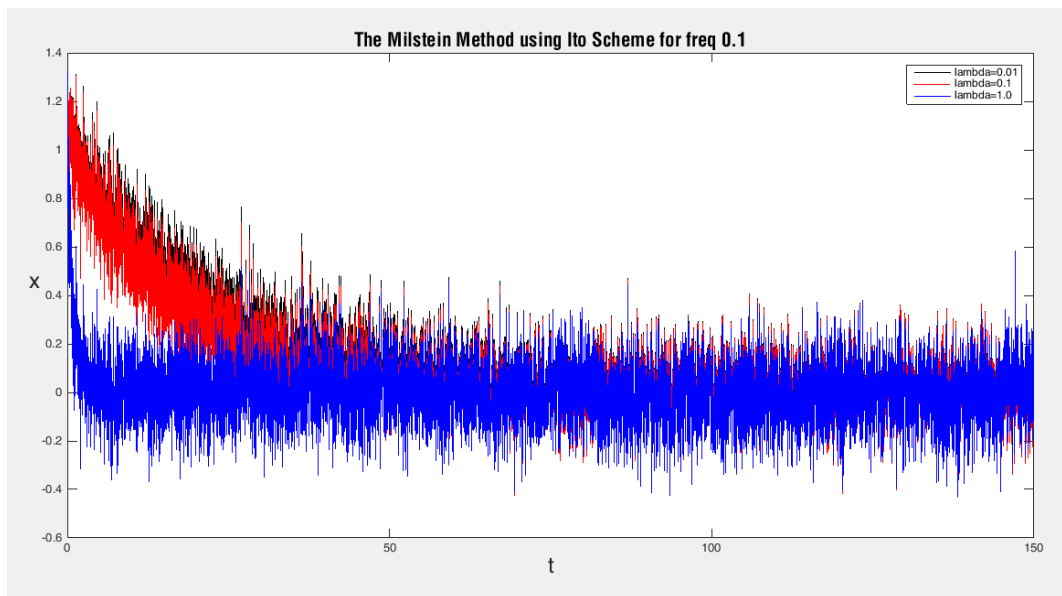
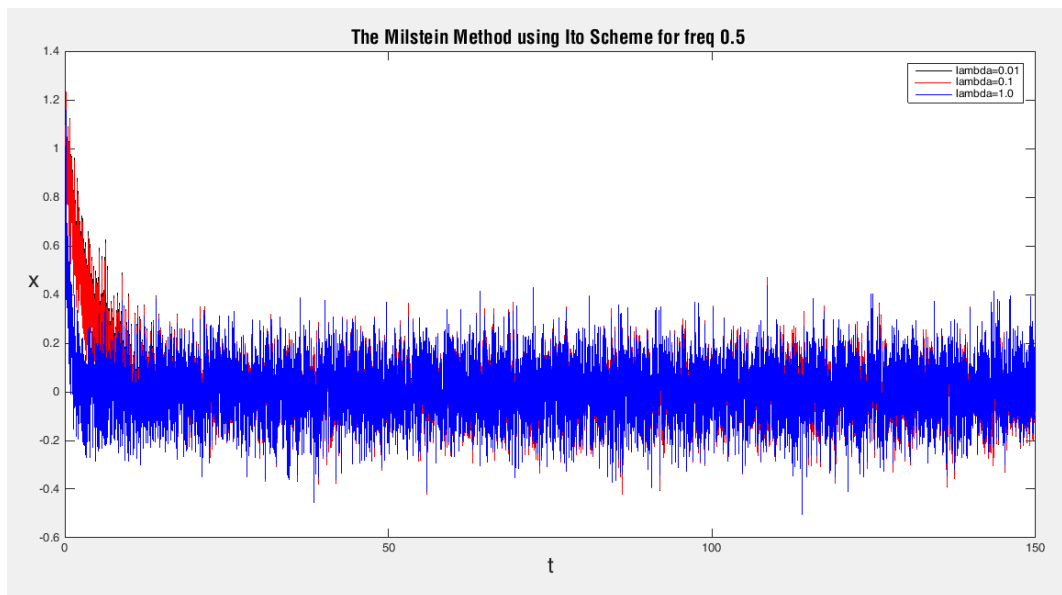
(a) Grandezas: $\lambda = 0.01$, $\lambda = 0.1$, $\lambda = 1.0$ e $\omega = 0.1$ (b) Grandezas: $\lambda = 0.01$, $\lambda = 0.1$, $\lambda = 1.0$ e $\omega = 0.5$

Figura 16 - Método de Milstein conforme interpretação de Stratonovich com ruído multiplicativo.

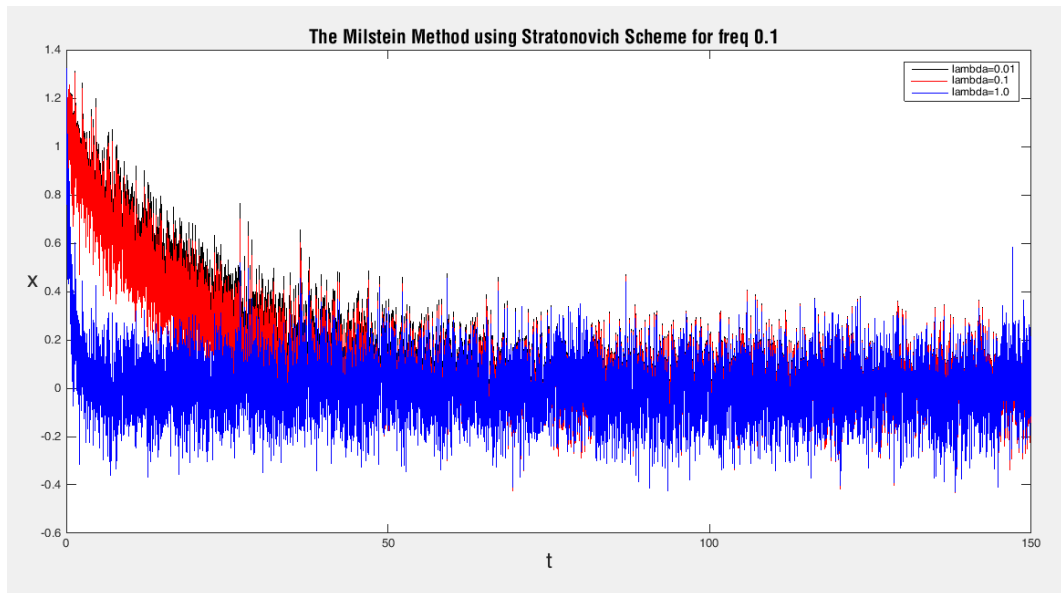
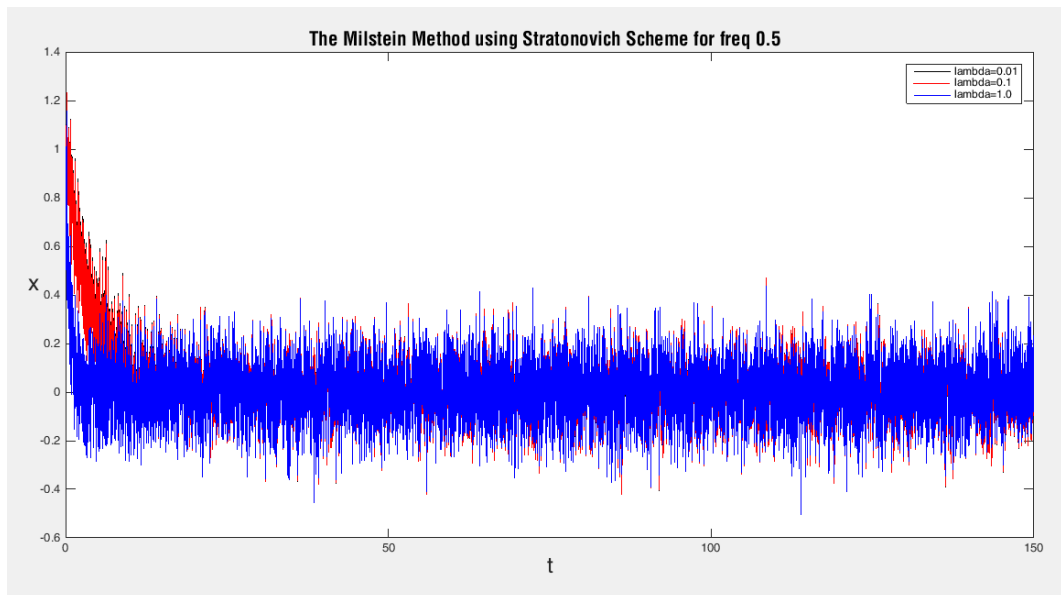
(a) Grandezas: $\lambda = 0.01$, $\lambda = 0.1$, $\lambda = 1.0$ e $\omega = 0.1$ (b) Grandezas: $\lambda = 0.01$, $\lambda = 0.1$, $\lambda = 1.0$ e $\omega = 0.5$

Figura 17 - Método de Milstein sem derivada conforme interpretação de Itô com ruído multiplicativo.

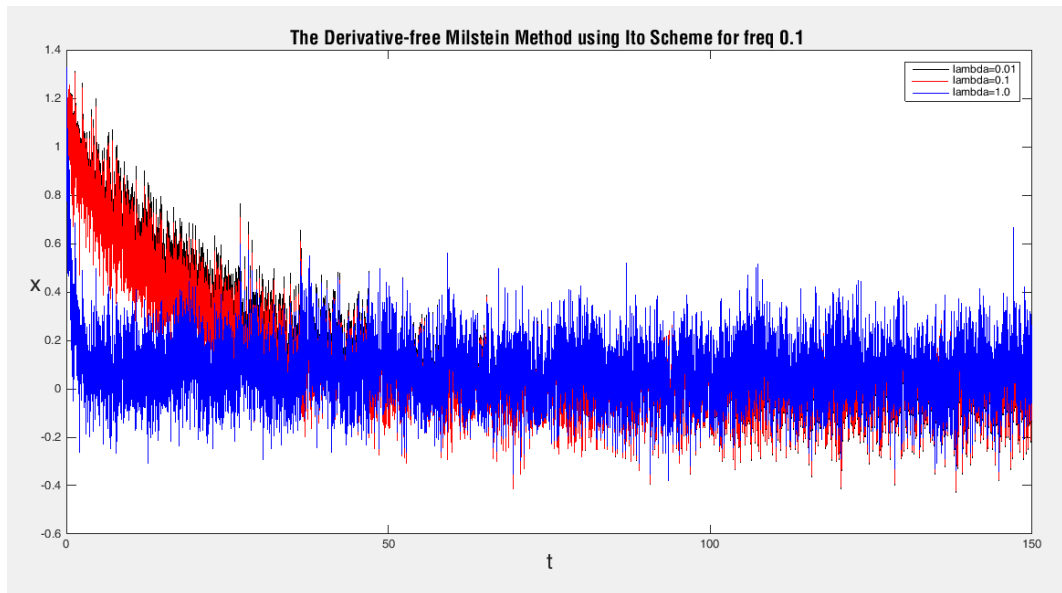
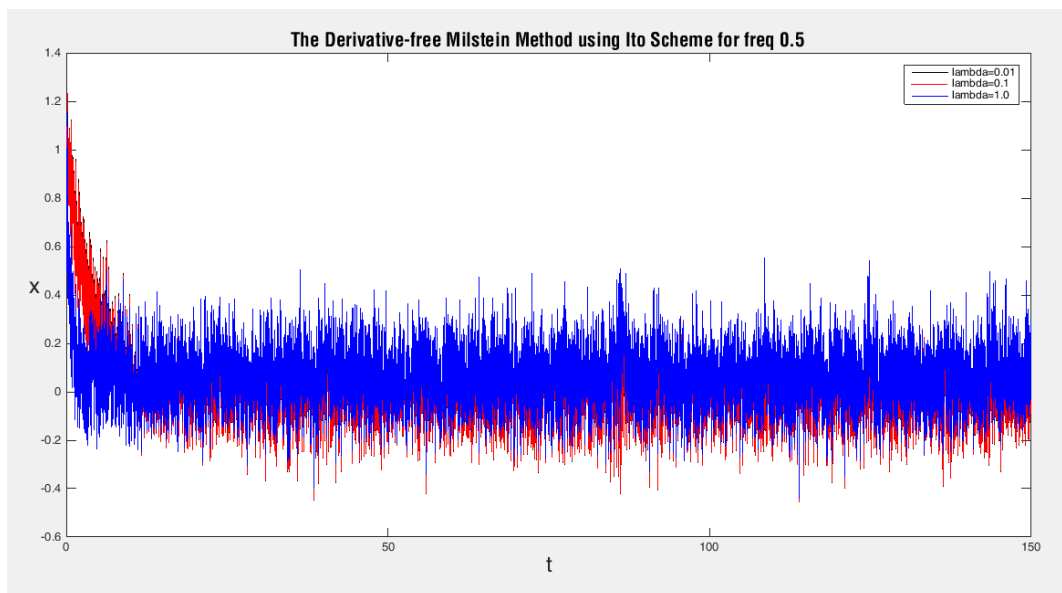
(a) Grandezas: $\lambda = 0.01$, $\lambda = 0.1$, $\lambda = 1.0$ e $\omega = 0.1$ (b) Grandezas: $\lambda = 0.01$, $\lambda = 0.1$, $\lambda = 1.0$ e $\omega = 0.5$

Figura 18 - Método de Milstein sem derivada com interp. de Stratonovich com ruído multiplicativo.

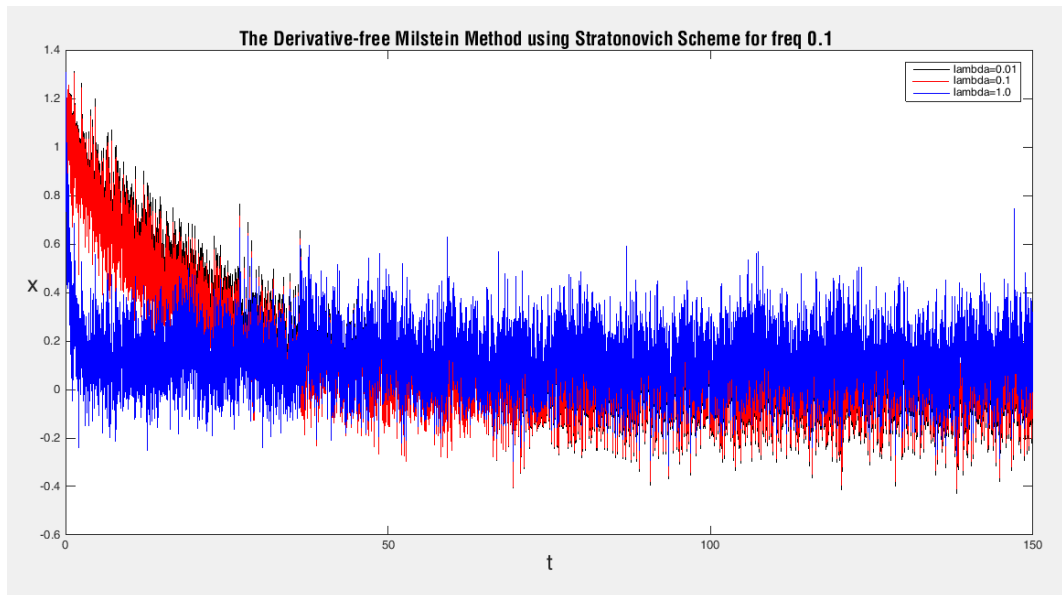
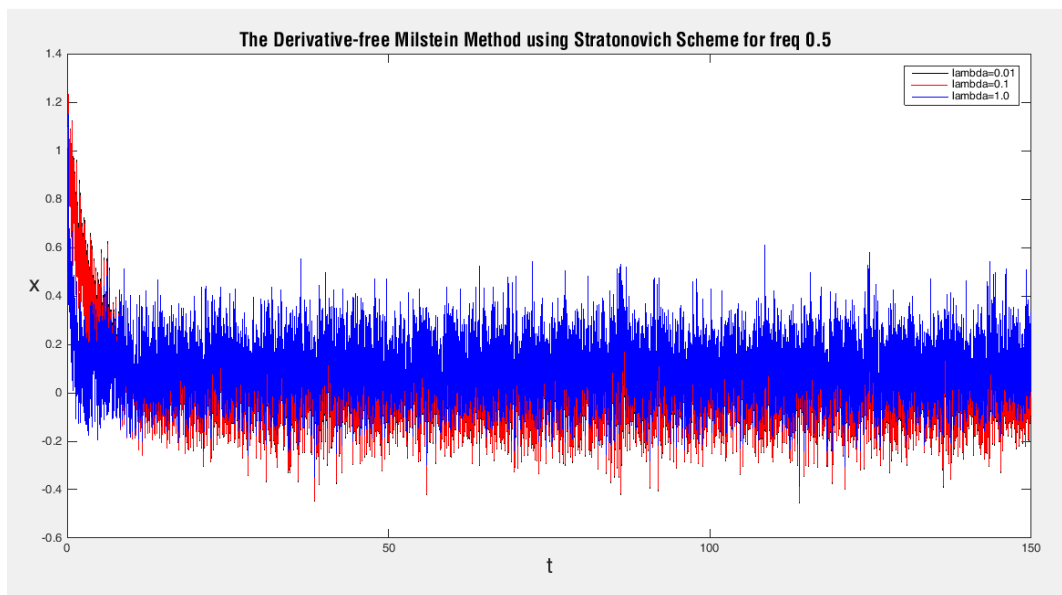
(a) Grandezas: $\lambda = 0.01$, $\lambda = 0.1$, $\lambda = 1.0$ e $\omega = 0.1$ (b) Grandezas: $\lambda = 0.01$, $\lambda = 0.1$, $\lambda = 1.0$ e $\omega = 0.5$

Figura 19 - Método de Runge-Kutta com ruído multiplicativo.

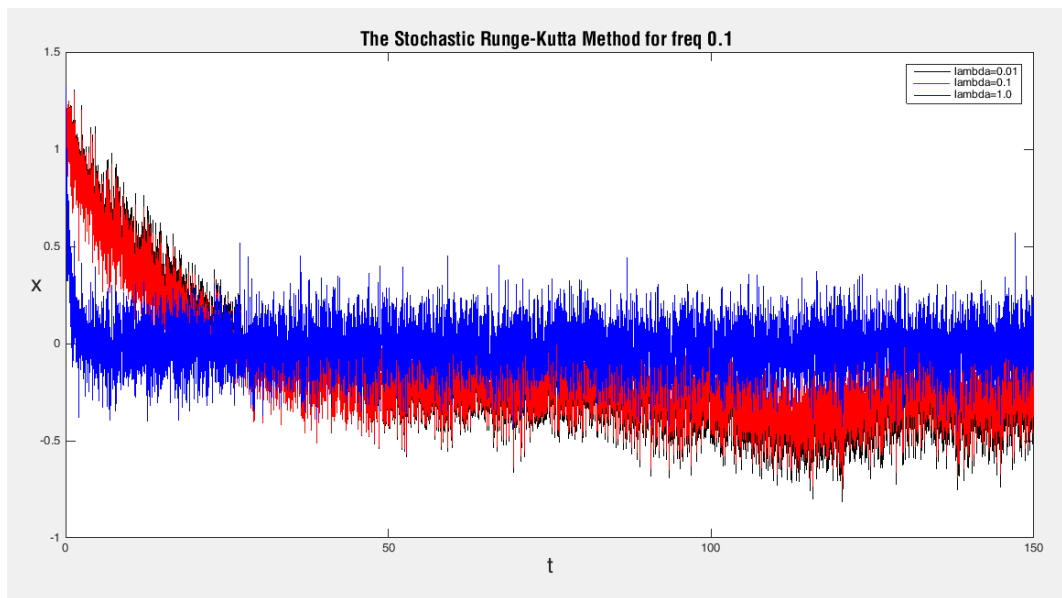
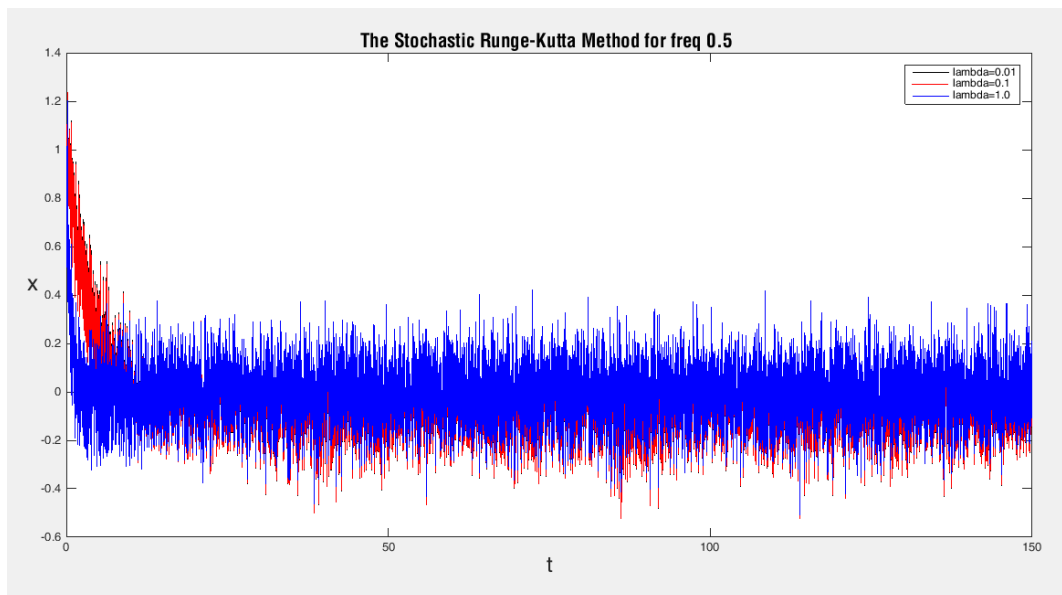
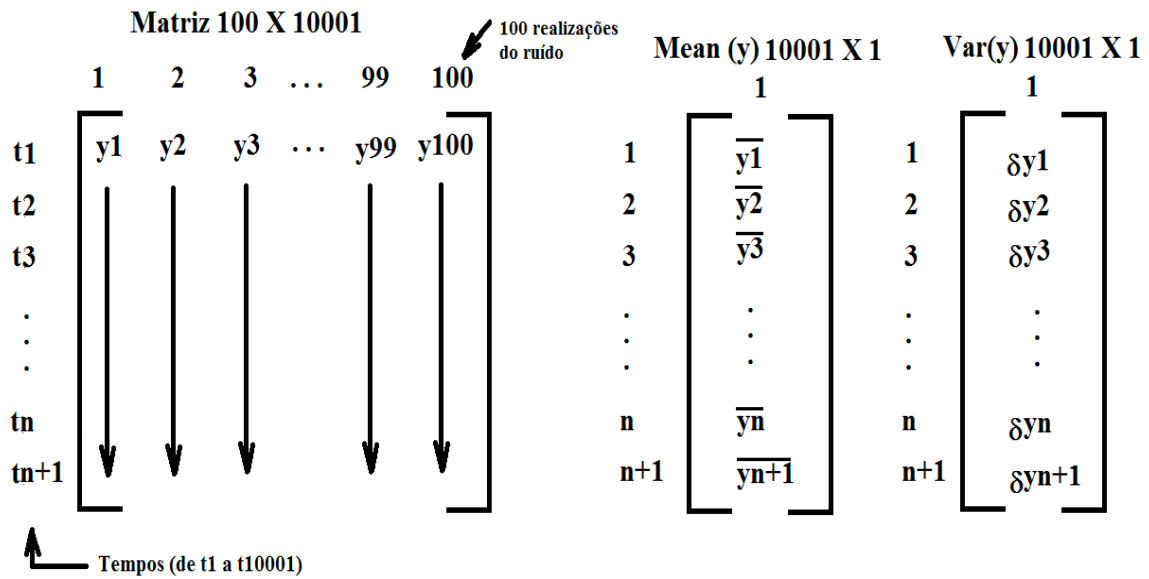
(a) Grandezas: $\lambda = 0.01$, $\lambda = 0.1$, $\lambda = 1.0$ e $\omega = 0.1$ (b) Grandezas: $\lambda = 0.01$, $\lambda = 0.1$, $\lambda = 1.0$ e $\omega = 0.5$

Figura 20 - Matriz com 100 diferentes realizações do ruído feita para cada método numérico.



Legenda: A figura representa a construção da matriz que contém valores de y , resultados da simulação numérica para 100 realizações do ruído. Estão exibidos também os vetores relacionados à média e variância analisadas em cada linha da matriz de 100 X 10001.

4.1.1 Análise Estatística

Conforme mencionado anteriormente, para o tratamento estatístico é necessário trabalhar com um número elevado de dados. Sendo assim, para cada método numérico foi construída uma matriz com 100 realizações do ruído. Cada coluna da matriz representa valores de y (solução numérica) para cada realização do ruído. A média foi calculada para cada linha da matriz produzindo uma matriz coluna. A variância, calculada também em cada linha da matriz em relação à média, gerando uma matriz coluna. A Figura 20 expõe a formação da matriz feita para cada um dos métodos, assim como as matrizes colunas referentes à média dos 100 resultados linha a linha e à variância em relação às médias.

Na análise estatística vista na Tabela 1, foram escolhidos 4 pontos para se verificar após 100 realizações do ruído, calculado o tempo médio e a variância para cada um destes pontos. Critério para seleção dos pontos: O primeiro ponto tem a intenção de analisar como os métodos numéricos se comportaram imediatamente após inicializar a simulação. O segundo encontra-se na região onde os métodos ainda estão caminhando na direção ao equilíbrio. Já no terceiro os métodos estão bem próximo ao equilíbrio. Por fim, o quarto compreende ao tempo onde todos os métodos estão em equilíbrio. A intenção de estudá-lo é analisar qual dos métodos oscila mais em torno de zero. Os 4 tempos escolhidos foram analisados em função de t_{mesh} . Os tempos 2,01s, 24,99s, 50,01s e 100,17s correspondem

Análise Estatística			
Método Numérico	t	$\langle x \rangle_t$	σ
Euler-Maruyama	2.01	0,900259442	0,012134063
	24.99	0,158471243	0,013655264
	50.01	0,035195531	0,014636240
	100.17	0,002919561	0,014791372
Euler-Heun	2.01	0,900072674	0,012014043
	24.99	0,158696681	0,013616240
	50.01	0,035185350	0,014639134
	100.17	0,002904428	0,014791330
Milstein Itô	2.01	0,918413649	0,012103483
	24.99	0,209358704	0,013589254
	50.01	0,055126207	0,014643839
	100.17	0,004497960	0,014771201
Milstein Stratonovich	2.01	0,918413649	0,012103483
	24.99	0,209358704	0,013589254
	50.01	0,055126207	0,014643839
	100.17	0,004497960	0,014771201
Milstein s/ derivada Itô	2.01	0,919465620	0,012102623
	24.99	0,217065964	0,013591193
	50.01	0,064663779	0,014677097
	100.17	0,014620124	0,014771065
Milstein s/ derivada Stratonovich	2.01	0,920466508	0,012103147
	24.99	0,224635703	0,013592409
	50.01	0,074222577	0,014677245
	100.17	0,024758521	0,014771282
Runge-Kutta	2.01	0,876086521	0,013608797
	24.99	-0,021824493	0,014835103
	50.01	-0,170513095	0,014528748
	100.17	-0,210922167	0,015602624

Tabela 1 - Análise Estatística de cada método numérico, utilizando o modelo com ruído multiplicativo, sob as mesmas condições de ruído $\lambda = 0.1$ e frequência = 0.1.

respectivamente aos pontos que equivalem a 135s, 1667s, 3335s e 6679s na malha.

Verifica-se na Tabela 1 que para o mesmo ponto onde $t(s) = 100, 17s$, o método que apresentou menor variância foi o Milstein sem derivada com interpretação de Itô. Analisando os demais pontos, nota-se que o método de Runge-Kutta, se comparado aos outros métodos, foi o que mais rapidamente caminhou em direção a região de equilíbrio, porém após atingir o equilíbrio, foi método o que apresentou maior valor médio de variância.

CONCLUSÕES

Esta dissertação teve como objetivo estudar de forma analítica a Equação de Langevin e analisar o seu comportamento através da simulação numérica em diferentes métodos numéricos. Como na composição do sistema prescrito pela Equação de Langevin existe uma força aleatória devido às colisões existentes entre as partículas do fluido e a partícula browniana, foi necessário compreender novos conceitos, conceitos estes não vistos na Graduação e que são relacionados ao Cálculo Estocástico. Após estudos teóricos, pôde-se compreender que existe uma complexidade na solução de uma Equação Diferencial Estocástica, pois seu tratamento matemático difere do utilizado no cálculo usual. Isto quer dizer que a integração do termo composto pelo ruído tem que ter um tratamento especial, o qual é diferente ao que estávamos acostumados usar até agora. Para efetuar a integração do ruído foram vistas duas definições diferentes quanto ao tratamento destas equações. Desta forma, estudou-se duas prescrições denominadas prescrição de Itô e prescrição de Stratonovich.

A Equação Diferencial Estocástica de estudo nesta dissertação, tem como modelo um sistema físico, cujo comportamento é de um oscilador harmônico super-amortecido. Este sistema foi escolhido por já conhecermos seu comportamento e dessa maneira poderíamos controlar os resultados numéricos. Contudo, como existem diferentes prescrições para solucionar a integral do termo que contém o ruído, deve-se haver um cuidado especial no tratamento do coeficiente de arrasto (*drift*), dado por $f(x)$ na EDE, o qual difere para ambas as prescrições. Ressalta-se que foram tomadas estas precauções para correta implementação computacional.

Um aspecto importante do presente trabalho foi a programação de vários métodos numéricos em MATLAB e a verificação de estarem calculando corretamente.

Estudou-se diferentes métodos numéricos com a finalidade de implementar computacionalmente seus esquemas iterativos para o modelo proposto. A intenção foi estudar o comportamento do sistema físico, primeiramente com o modelo que utilizasse ruído aditivo e posteriormente um segundo modelo que utilizasse ruído multiplicativo. A ideia foi analisar a evolução do sistema aditivo, no caso em que $\lambda = 0$, observando seu comportamento com frequências do oscilador iguais a 0.1 e 0.5, depois observar como se comportava em três diferentes intensidades do ruído multiplicativo com $\lambda = 0.01$ (muito próximo do ruído aditivo), $\lambda = 0.1$ e $\lambda = 1.0$ e frequências do oscilador também iguais a 0.1 e 0.5. A conclusão que se chegou foi que aumentando-se a frequência do oscilador observou-se em todas as simulações dos métodos numéricos, que o sistema vai para o equilíbrio mais rapidamente. Quanto à variação da intensidade do ruído, no segundo modelo, com ruído multiplicativo, notou-se que a influência do ruído no sistema pode-se considerar "benéfica" uma vez que o ruído acelerou a convergência do sistema para a região do equilíbrio. Por

outro lado, a influência do ruído pode ser prejudicial ao sistema, como por exemplo acontece em sistemas eletrônicos, onde temos que filtrar o ruído para limpeza do sinal e obter uma melhor monitoração. Os resultados obtidos graficamente ilustram estas ponderações.

Uma análise estatística foi feita, uma vez que foi obtido um grande volume de dados através das matrizes construídas para cada método numérico, com 100 resultados diferentes (100 realizações do ruído). O propósito de construir estas matrizes foi de analisar, em média, qual dos sistemas tende mais rapidamente para sua região de equilíbrio, concluindo que foi o método de Runge-Kutta, e de avaliar em termos da variância, qual dos métodos após atingir o equilíbrio, atentando-se para um mesmo ponto, oscilou mais em torno da região de equilíbrio. Neste último caso, chegamos a conclusão de que foi o método de Milstein sem derivada com interpretação de Itô.

Trabalhos Futuros

Como continuação do trabalho desenvolvido nesta dissertação, futuramente podem ser estudados diferentes sistemas físicos e analisados seus comportamentos em cada método numérico, utilizando a implementação computacional aqui desenvolvida.

Posteriormente também, podem ser efetuadas melhorias na otimização da estrutura do código computacional produzido. Beneficiando assim o leitor, com uma maior clareza nos passos da implementação.

Foi muito interessante analisar que a influência do ruído multiplicativo no sistema é benéfica, conforme mencionado anteriormente, uma vez que o ruído contribui para que o sistema atinja mais rapidamente a região de equilíbrio. Recomenda-se que seja feita uma análise mais aprofundada desta ocorrência, pesquisando-se sobre *Ressonância Estocástica* [28].

REFERÊNCIAS

- 1 GARDINER, C. W. *Handbook of Stochastic Methods: for physics, chemistry and the natural sciences*. 3. ed. Rio de Janeiro: Springer, 2004. 440 p.
- 2 SILVA, J.; LIMA, J. *Quatro abordagens para o movimento browniano*. 1. ed. Rio Grande do Norte, São Paulo: Revista Brasileira de Ensino de Física, 2007. v. 1. 11 p.
- 3 EINSTEIN, A. *Investigations on the Theory of the Brownian Movement. Texto editado com notas, por R. Fürth*. New York: Dover Publications, 1956.
- 4 LANGEVIN, P. *Comptes Rendues*. New York: Acad. Sci, 1908. 146,530-533 p.
- 5 VASQUEZ E.J., M.; SOUZA, M. de. Explorando a conexão entre a mecânica dos fluidos e a teoria cinética. *Revista Brasileira de Ensino de Física*, v. 38, p. 9, 2016.
- 6 ROCHA, J. F. M. *Origens e Evolução das Ideias da Física*. Salvador: Editora da Universidade Federal da Bahia, 2011. 374 p.
- 7 SALINAS, S. R. *Einstein e a teoria do movimento browniano*. 2. ed. São Paulo: Revista Brasileira de Ensino de Física, 2005. v. 27. 7 p.
- 8 ZORNIO, B. F. *Estudo teórico e experimental da teoria de Kramers utilizando pinças ópticas e dinâmica de Langevin*. 2014. 86 p. Tese (Mestrado) — Universidade Estadual de Campinas, 2014.
- 9 MIRANDA, B. M.; RODRIGUES, C. G. Estudo do transporte eletrônico no semicondutor arseneto de gálio usando a equação de newton-langevin. *Revista Arithmós-Revista da Escola de Ciências Exatas e da Computação, Goiânia*, v. 1, p. 47–53, 2019.
- 10 OLIVEIRA, P. M. *O papel das flutuações na Biologia*. 3. ed. Rio de Janeiro: Revista Brasileira de Ensino de Física, 2007. v. 29. 8 p.
- 11 FILHO, J. S. e. a. D. V. Modelo computacional estocástico para simulações de séries climáticas diárias de umidade relativa do ar, baseado na parametrização dinâmica das distribuições de probabilidade decorrente da retroalimentação de dados. *Revista Brasileira de Climatologia, Ponta Grossa - Paraná*, v. 5, p. 19, 2017.
- 12 BESSADA, D. F. *Generalizações do Movimento Browniano e suas Aplicações à Física e a Finanças*. 2005. 186 p. Tese (Mestrado) — Instituto de Física Teórica, 2005.
- 13 NEVES, L. R.; MAIA, L. P. Um modelo estocástico de crescimento populacional com limitação de recursos.: Livro de resumos. Universidade de São Paulo, São Paulo, v. 5, 2018.
- 14 KOLTERMANN, P. I. *Cálculo de Campo Magnéticos considerando Histerese*. 2001. 134 p. Tese (Doutorado) — Universidade Federal de Santa Catarina, 2001.
- 15 INCI, G. e. a. Langevin dynamics simulation of transport and aggregation of soot nano-particles in turbulent flows. *flow, turbulence and combustion*. Springer Netherlands, v. 98, p. 1065–1085, 2017.

- 16 STARIOLO, D. A. *Introdução aos Processos Estocásticos e Aplicações*. Rio Grande do Sul: Instituto de Física da Universidade Federal do Rio Grande do Sul, 2006. 102 p.
- 17 SALINAS, S. R. A. *Introdução à Física Estatística*. 2. ed. São Paulo: Editora da Universidade de São Paulo, 2008. v. 2.
- 18 ALBUQUERQUE, J. P. d. A.; FORTES, J. M. P.; FINAMORE, W. A. *Probabilidade, Variáveis Aleatórias e Processos Estocásticos*. Rio de Janeiro: Pontifícia Universidade Católica-RJ, 2008. 334 p.
- 19 TECHNOLOGY, M. I. of. *The Dirac Delta Function and Convolution*. Massachusetts: <https://pdfs.semanticscholar.org/d3fb/8a11ce8fd94cd085ae6d02cd13dfd28b282a.pdf>. Acesso em: 16 ago. 2019.
- 20 SCHERER, C. *Métodos Computacionais da Física: Versão scilab*. 2. ed. São Paulo: Livraria da Física, 2010. 299 p.
- 21 ARENAS, Z. G. *Formulação Supersimétrica de Processos Estocásticos com Ruído Multiplicativo*. 2012. 94 p. Tese (Doutorado) — Universidade do Estado do Rio de Janeiro, 2012.
- 22 BAYRAM MUSTAFA, P. T. B. G. O. Numerical methods for simulation of stochastic differential equations. *Advance in Difference Equations: SpringerOpen Journal*, v. 1, p. 10, 2018.
- 23 BURRAGE, K.; BURRAGE, P. High strong order explicit runge-kutta methods for stochastic ordinary differential equations. Elsevier, Australia, p. 21, 1996.
- 24 KLOEDEN, P.; PLATEN, E. *Numerical Solution of Stochastic Differential Equations. Applications of Mathematics (Stochastic Modelling and Applied Probability)*. Berlin, Heidelberg: Springer, 1992. v. 23. 146,530-533 p.
- 25 SCHAFFTER, T. *Numerical Integrations of SDEs. A short Tutorial*. Lausanne: Swiss Federal Institute of Technology in Lausanne (École polytechnique fédérale de Lausanne, 2010. 8 p.
- 26 IACUS, S. M. *Simulation and Inference for Stochastic Differential Equations: With r examples*. Nova York: Springer, 2008. 284 p.
- 27 BRUNELLI, J. C. *Renormalização na quantização estocástica de teorias de campos*. 1991. 109 p. Tese (Doutorado) — Universidade de São Paulo, 1991.
- 28 GAMMAITONI, L. e. a. Stochastic resonance. *Reviews of modern physics*, v. 70, p. 65, 1998.

APÊNDICE – Modelagem Computacional

Algoritmo 1 - PROGRAMA IMPLEMENTADO PARA RESOLVER UMA EDE DE UM OSCILADOR
HARMÔNICO SUPERAMORTECIDO DADO PELO MODELO PROPOSTO

```

1  %


---


2  %  main_sde_multiplicativeoscillator.m
3  %
4  %  This is the main file for a program that solves
5  %  SDE super-damped harmonic oscillator problem given
6  %  by the model below:
7  %
8  %   $dx = -\omega/2 * x(1 + \lambda^2 * x^2)^2 dt +$ 
9  %   $+ (1 + \lambda^2 * x^2)^2 dW,$     for t in (0.3,30)
10 %


---


11 %  programmer: Sara Cristina Quintanilha Valente
12 %  saravalenteq@gmail.com
13 %
14 %  last update: May 21, 2019
15 %


---


16
17  clc
18  clear all
19
20  close all
21
22
23 %  program header
24 %


---



```

```

25 disp(' ')
26 disp(' SDE super-damped harmonic oscillator problem ')
27 disp(' ')
28 disp(' dx = -omega/2 * x(1 + lambda^2 * x^2)^2 dt + ')
29 disp('      +(1 + lambda^2 * x^2)^2 dW,      for t in (1,30) ')
30 disp(' ')
31 %

```

```

32
33
34 % Arrays that hold 100 results for each method used, using
      different
35 % generations of noise
36 %

```

```

37 Eu = zeros(10001,100); % matrix that stores the result of 100
38 %repetitions of Euler-Maruyama method
39
40 He = zeros(10001,100); % matrix that stores the result of 100
41 %repetitions of Euler-Heun method
42
43 Mii = zeros(10001,100); %matrix that holds the result of the 100
44 %repetitions of Milstein with interpretation of Ito method
45
46 Mis = zeros(10001,100); %matrix that holds the result of the 100
47 %repetitions of Milstein with interpretation of Stratonovich
      method
48
49 Midi = zeros(10001,100); %matrix that holds the result of the
      100
50 %repetitions of Derivative-free Milstein with interpretation of
      Ito method
51
52 Mids = zeros(10001,100);%matrix that holds the result of the 100
53 %repetitions of Derivative-free Milstein with interpretation
54 %of Stratonovich method
55

```

```

56 Ru = zeros(10001,100); % matrix that stores the result of 100
57 %repetitions of Stochastic Runge–Kutta
58
59
60 q=0; % contador
61 while q<100
62     q = q+1;
63
64
65 % define physical and numerical parameters
66 %

```

```

67 % left extreme of the domain (tmin)
68 tmin=0;
69
70 % right extreme of the domain (tmax)
71 %tmax=30;
72 tmax=150;
73
74 % constant that controls the intensity of the noise (lambda)
75 % when the function g is a constant (additive noise)
76 lambda = 0.0;
77
78 % when the function g depends on the state of the system
79 % (multiplicative noise)
80 %lambdazero = 0.01; %%%%%%%%%%
81 %lambdaum = 0.1; %%%%%%%%%%
82 %lambdadois = 1.0; %%%%%%%%%%
83 %lambda = 0.5;
84 %lambda = 1.5;
85
86 % lambda squared (l)
87 %l=lambda^2;
88 l = lambda^2;
89 %lzero = lambdazero^2; %%%%%%%%%%
90 %lum = lambdaum^2; %%%%%%%%%%
91 %ldois = lambdadois^2; %%%%%%%%%%
92

```

```
93 % oscillator frequency (freq)
94 %freq=0.1;
95 freq=0.5;
96 %freq=1.0;
97 %freq=1.5;
98 %freq=5.0;
99
100 % number of elements (n)
101 %n=1000000;
102 n=10000;
103
104 % number of nodes
105 nm = n+1;
106
107 %


---


108
109
110 % step size .
111 %


---


112 disp(' step size ')
113 disp(' ')
114
115 h = sde_elem_size(tmax,tmin,n);
116 %


---


117
118
119 % generate mesh
120 %


---


121 disp(' generate mesh ')
122 disp(' ')
123
```

```

124 tmesh = sde_generate_mesh(tmin, tmax, h);
125 %


---


126
127
128 % generate noise white
129 %


---


130 disp(' generate white noise ')
131 disp(' ')
132
133 dW = sde_generate_white_noise(nm, h, n);
134 %


---


135
136
137 % numeric solution euler-mauyama, milstein, heun respectively
138 %


---


139
140 % addtive noise
141 %


---


142
143 y = sde_numericmethod_eulermaruyama(nm, h, dW, n, l, freq);
144 [xito, xstra] = sde_numericmethod_milstein(nm, h, dW, n, l, freq);
145 z = sde_numericmethod_heun(nm, h, dW, n, l, freq);
146 [kito, kstra] = sde_numderivativefree_milstein(nm, h, dW, n, l, freq);
147 r = sde_numericmethod_rungekutta(nm, h, dW, n, l, freq);
148
149 % multiplicative noise
150 %


---



```



```

151
152 %[ yzero ,yum, ydois ] = ...
153 %    sde_numericmethod_eulermaruyama(nm,h,dW,n,lzero ,lum ,ldois ,
      freq);
154 %[ xitozero ,xitoum ,xitodois ,xstrazero ,xstraum ,xstradois ] = ...
155 %    sde_numericmethod_milstein(nm,h,dW,n,lzero ,lum ,ldois ,freq);
156 %[ zzero ,zum, zdois ] = ...
157 %    sde_numericmethod_heun(nm,h,dW,n,lzero ,lum ,ldois ,freq);
158 %[ kitozero ,kitoum ,kitodois ,kstrazero ,kstraum ,kstradois ] = ...
159 %    sde_numderivativefree_milstein(nm,h,dW,n,lzero ,lum ,ldois ,
      freq);
160 %[ rzero ,rum, rdois ] = ...
161 %    sde_numericmethod_rungekutta(nm,h,dW,n,lzero ,lum ,ldois ,freq
      );
162
163
164 % plot the aproximation solution
165 %

```

```

166
167 % additive noise
168 %

```

```

169
170 %figure
171 %plot(tmesh,y,'g')
172 %title('Numeric Method'); legend('The Euler-Maruyama Method')
173 %xlabel('t','FontSize',18); ylabel('y','FontSize',18,'Rotation
      ',0)
174
175 %figure
176 %plot(tmesh,xito,'b')
177 %title('Numeric Method'); legend('The Milstein Method using Ito
      Scheme')
178 %xlabel('t','FontSize',18); ylabel('xito','FontSize',18,'
      Rotation',0)
179

```

```

180 %figure
181 %plot(tmesh, xstra, 'b')
182 %title('Numeric Method');
183 %legend('The Milstein Method using Stratonovich Scheme')
184 %xlabel('t', 'FontSize', 18); ylabel('xstra', 'FontSize', 18, '
    Rotation', 0)
185
186 %figure
187 %plot(tmesh, kito, 'k')
188 %title('Numeric Method');
189 %legend('The Derivative-free Milstein Method using Ito Scheme')
190 %xlabel('t', 'FontSize', 18); ylabel('kito', 'FontSize', 18, '
    Rotation', 0)
191
192 %figure
193 %plot(tmesh, kstra, 'k')
194 %title('Numeric Method');
195 %legend('The Derivative-free Milstein Method using Stratonovich
    Scheme')
196 %xlabel('t', 'FontSize', 18); ylabel('kstra', 'FontSize', 18, '
    Rotation', 0)
197
198 %figure
199 %plot(tmesh, z, 'r')
200 %title('Numeric Method'); legend('The Heun Method')
201 %xlabel('t', 'FontSize', 18); ylabel('z', 'FontSize', 18, 'Rotation
    ', 0)
202
203 %figure
204 %plot(tmesh, r, 'c')
205 %title('Numeric Method'); legend('The Stochastic Runge-Kutta
    Method')
206 %xlabel('t', 'FontSize', 18); ylabel('r', 'FontSize', 18, 'Rotation
    ', 0)
207
208 % multiplicative noise
209 %

```

```

210
211 %figure
212 %plot(tmesh,yzero,'k',tmesh,yum,'r',tmesh,ydois,'b')
213 %title('The Euler–Maruyama Method for freq 0.5');
214 %legend('lambda=0.01','lambda=0.1','lambda=1.0')
215 %xlabel('t','FontSize',18); ylabel('y','FontSize',18,'Rotation
      ',0)

216
217 %figure
218 %plot(tmesh,xitozero,'k',tmesh,xitoum,'r',tmesh,xitodois,'b')
219 %title('The Milstein Method using Ito Scheme for freq 0.5');
220 %legend('lambda=0.01','lambda=0.1','lambda=1.0')
221 %xlabel('t','FontSize',18); ylabel('y','FontSize',18,'Rotation
      ',0)

222
223 %figure
224 %plot(tmesh,xstrazero,'k',tmesh,xstraum,'r',tmesh,xstradois,'b')
225 %title('The Milstein Method using Stratonovich Scheme for freq
      0.5');
226 %legend('lambda=0.01','lambda=0.1','lambda=1.0')
227 %xlabel('t','FontSize',18); ylabel('y','FontSize',18,'Rotation
      ',0)

228
229 %figure
230 %plot(tmesh,zzero,'k',tmesh,zum,'r',tmesh,zdois,'b')
231 %title('The Heun Method for freq 0.5');
232 %legend('lambda=0.01','lambda=0.1','lambda=1.0')
233 %xlabel('t','FontSize',18); ylabel('y','FontSize',18,'Rotation
      ',0)

234
235 %figure
236 %plot(tmesh,kitozero,'k',tmesh,kitoum,'r',tmesh,kitodois,'b')
237 %title('The Derivativefree Milstein Method using Ito Scheme for
      freq 0.5');
238 %legend('lambda=0.01','lambda=0.1','lambda=1.0')
239 %xlabel('t','FontSize',18); ylabel('y','FontSize',18,'Rotation
      ',0)

240
241 %figure

```

```

242 %plot(tmesh,kstrazero,'k',tmesh,kstraum,'r',tmesh,kstradois,'b')
243 %title('TheDerivativefree Milstein Method using Stratonovich for
        freq0.5');
244 %legend('lambda=0.01','lambda=0.1','lambda=1.0')
245 %xlabel('t','FontSize',18); ylabel('y','FontSize',18,'Rotation
        ',0)
246
247 %figure
248 %plot(tmesh,rzero,'k',tmesh,rum,'r',tmesh,rdois,'b')
249 %title('The Stochastic Runge–Kutta Method for freq 0.5');
250 %legend('lambda=0.01','lambda=0.1','lambda=1.0')
251 %xlabel('t','FontSize',18); ylabel('y','FontSize',18,'Rotation
        ',0)
252
253
254 %

```

```

255
256 % Matrix
257 %

```

```

258 % Euler–Maruyama method
259     Eu(:,q) = y';
260
261 % Euler–Heun method
262     He(:,q) = z';
263
264 % Milstein with interpretation Ito method
265     Mii(:,q) = xito';
266
267 % Milstein with interpretatimon Stratonovich method
268     Mis(:,q) = xstra';
269
270 % Derivative–free Milstein with interpretation Ito method
271     Midi(:,q) = kito';
272
273 % Derivative–free Milstein with interpretation Stratonovich

```

```
method
274     Mids(:,q) = kstra';
275
276 % Stochastic Runge–Kutta
277     Ru(:,q) = r';
278 end
279
280 % Mean for each matrix hold 100 results for each method used,
281 % using different generations of noise
282 meanEu = mean(Eu,2);
283 meanHe = mean(He,2);
284 meanMii = mean(Mii,2);
285 meanMis = mean(Mis,2);
286 meanMidi = mean(Midi,2);
287 meanMids = mean(Mids,2);
288 meanRu = mean(Ru,2);
289
290 %Variance of each matrix
291 varEu = var(Eu,0,2);
292 varHe = var(He,0,2);
293 varMii = var(Mii,0,2);
294 varMis = var(Mis,0,2);
295 varMidi = var(Midi,0,2);
296 varMids = var(Mids,0,2);
297 varRu = var(Ru,0,2);
```

Algoritmo 2 - FUNÇÃO QUE COMPUTA O TAMANHO DO IELE-TH ELEMENTO 1D

```

1
2 %


---


3 % mfem_1D_elem_size.m
4 %
5 % This function computes the iele-th 1D element size.
6 %
7 % Input:
8 % n      - number of elements
9 % tmin   - minimum t
10 % tmax  - maximum t
11 %
12 % Output:
13 % h      - element size
14 %


---


15 % programmer: Sara Cristina Quintanilha Valente
16 %             saravalenteq@gmail.com
17 %
18 % last update: Jan 21, 2019
19 %


---


20
21 %


---


22 function h = sde_elem_size(tmax,tmin,n)
23     h = (tmax-tmin)/n;
24 return
25 %


---



```

Algoritmo 3 - FUNÇÃO QUE GERA UMA MALHA PARA O DOMÍNIO ANALISADO

```
1 %  


---

  
2 % sde_generate_mesh.m  
3 %  
4 % This function generates a mesh for [tmin,tmax] domain.  
5 %  
6 % Input:  
7 % tmin    - minimum t  
8 % tmax    - maximum t  
9 % h       - element size  
10 %  
11 % Output:  
12 % tmesh   - mesh points vector  
13 %  


---

  
14 % programmer: Sara Cristina Quintanilha Valente  
15 %             saravalenteq@gmail.com  
16 %  
17 % last update: Jan 21, 2019  
18 %  


---

  
19  
20 %  


---

  
21 function tmesh = sde_generate_mesh(tmin,tmax,h)  
22     tmesh=tmin:h:tmax;  
23 return  
24 %  


---


```

Algoritmo 4 - FUNÇÃO QUE GERA RÚIDO BRANCO PARA O PROBLEMA EM QUESTÃO

```

1  %


---


2  % sde_generate_white_noise.m
3  %
4  % This function generates white noise for the problem in
   % question.
5  %
6  % Input:
7  %
8  % nm – mesh number of nodes
9  % h – step size
10 % n – number of elements
11 %
12 % Output:
13 % dW – vector with white noise (independent random variable
   % with mean
14 % zero and variance 1)
15 %


---


16 % programmer: Sara Cristina Quintanilha Valente
17 % saravalenteq@gmail.com
18 %
19 % last update: Jan 21, 2019
20 %


---


21
22 %


---


23 function dW = sde_generate_white_noise(nm,h,n)
24     randn('state',sum(100*clock)); %Resets to a different state
   % every time
25
26     W=zeros(1,n+2);

```



```
27     dW=zeros(1,nm);
28
29     for j=1:n+2
30         W(j)=sqrt(h)*randn;
31     end
32
33     for j=1:nm
34         dW(j)=W(j+1)-W(j);
35     end
36     return
37     %
```

Algoritmo 5 - FUNÇÃO QUE GERA UMA SOLUÇÃO NUMÉRICA USANDO O MÉTODO DE
EULER-MARUYAMA

```

1  %


---


2  % sde_numericmethod_eulermaruyama.m
3  %
4  % This function generates solution numeric using
5  % the numeric method Euler-Maruyama.
6  %
7  % Input:
8  %
9  % nm - mesh number of nodes
10 % f - drift coefficient
11 % g - diffusion coefficient
12 % h - step size
13 % dW - white noise
14 % n - number of elements
15 % l - lambda squared
16 % freq - oscillator frequency
17 %
18 % Output:
19 % y - vector with numeric solution of numeric method Euler-
20 % Maruyama


---


21 % programmer: Sara Cristina Quintanilha Valente
22 % saravalenteq@gmail.com
23 %
24 % last update: May 21, 2019
25 %


---


26
27 %


---



```

```

28
29 % aditive noise
30
31 function y = sde_numericmethod_eulermaruyama(nm,h,dW,n,l , freq)
32
33     y = zeros(1,nm); % vector pre-allocation for results
34
35     y(1) = 1.0; % initial condition for Euler-Maruyama
36
37     for i = 1:n
38
39         g = (1+(l*(y(i)^2))); % difusion coefficient
40
41         f = ((-0.5*freq*y(i))*(g^2)); % drift coefficient
42
43         y(i+1) = y(i) + (f*h) + (g*dW(i)); % iterative method
44     end
45     return
46
47 % multiplicative noise
48
49 %function [yzero ,yum, ydois] = ...
50 %     sde_numericmethod_eulermaruyama(nm,h,dW,n,lzero ,lum ,ldois ,
51     freq)
52     %yzero=zeros(1,nm); % vector pre-allocation for results
53     %yum=zeros(1,nm); % vector pre-allocation for results
54     %ydois=zeros(1,nm); % vector pre-allocation for results
55
56     %yzero(1)=1.0; % initial condition for Euler-Maruyama
57     %yum(1)=1.0; % initial condition for Euler-Maruyama
58     %ydois(1)=1.0; % initial condition for Euler-Maruyama
59
60     %for i=1:n
61         %gzero = (1+(lzero*(yzero(i)^2))); % difusion
62         coefficient
63         %gum = (1+(lum*(yum(i)^2))); % difusion coefficient
64         %gdois = (1+(ldois*(ydois(i)^2))); % difusion
65         coefficient
66
67     end
68     return
69
70 %function [yzero ,yum, ydois] = ...
71 %     sde_numericmethod_eulermaruyama(nm,h,dW,n,lzero ,lum ,ldois ,
72     freq)
73     %yzero=zeros(1,nm); % vector pre-allocation for results
74     %yum=zeros(1,nm); % vector pre-allocation for results
75     %ydois=zeros(1,nm); % vector pre-allocation for results
76
77     %yzero(1)=1.0; % initial condition for Euler-Maruyama
78     %yum(1)=1.0; % initial condition for Euler-Maruyama
79     %ydois(1)=1.0; % initial condition for Euler-Maruyama
80
81     %for i=1:n
82         %gzero = (1+(lzero*(yzero(i)^2))); % difusion
83         coefficient
84         %gum = (1+(lum*(yum(i)^2))); % difusion coefficient
85         %gdois = (1+(ldois*(ydois(i)^2))); % difusion
86         coefficient
87
88     end
89     return

```

```
64     %fzero = ((-0.5*freq*yzero(i))*(gzero^2)); % drift
        coefficient
65     %fum = ((-0.5*freq*yum(i))*(gum^2)); % drift coefficient
66     %fdois = ((-0.5*freq*ydois(i))*(gdois^2)); % drift
        coefficient
67
68     %yzero(i+1) = yzero(i) + (fzero*h) + (gzero*dW(i));
69     %yum(i+1) = yum(i) + (fum*h) + (gum*dW(i));
70     %ydois(i+1) = ydois(i) + (fdois*h) + (gdois*dW(i));
71     %end
72 %return
73 %
```

Algoritmo 6 - FUNÇÃO QUE GERA UMA SOLUÇÃO NUMÉRICA USANDO O MÉTODO DE
MILSTEIN

```

1 %


---


2 % sde_numericmethod_milstein.m
3 %
4 % This function generates solution numeric using
5 % the numeric method Milstein.
6 %
7 % Input:
8 %
9 % nm – mesh number of nodes
10 % f – drift coefficient
11 % g – diffusion coefficient
12 % h – step size
13 % dW – white noise
14 % n – number of elements
15 % l – lambda squared
16 % freq – oscillator frequency
17 %
18 % Output:
19 % xito – vector with numeric solution of numeric method
20 % Milstein (Ito
21 % interpretation)
22 % xstra – vector with numeric solution of numeric method
23 % Milstein
24 % (Stratonovich interpretation)
25 %


---


24 % programmer: Sara Cristina Quintanilha Valente
25 % saravalenteq@gmail.com
26 %
27 % last update: May 21, 2019
28 %


---



```

```

29
30 %

```

```

31
32 % aditive noise
33
34 function [xito ,xstra] = sde_numericmethod_milstein(nm,h,dW,n,l ,
    freq)
35     %x=zeros(1,nm);
36     xito = zeros(1,nm); % vector pre-allocation for results
37     %(Ito interpretation)
38
39     xstra = zeros(1,nm); % vector pre-allocation for results
40     %(Stratonovich interpretation)
41
42     %x(1)=1.0; % initial condition for Euler-Maruyama
43     xito(1) = 1.0; % initial condition for Euler-Maruyama (Ito)
44
45     xstra(1) = 1.0; % initial condition for Euler-Maruyama (
46         Stratonovich)
47
48     %%%%% Important: For additive noise, both results will be
49     the same
50
51     for i = 1:n
52         %

```

```

53         % Ito Interpretation
54         %

```

```

55         %g = (1+(1*(x(i)^2))); % difusion coefficient
56         %f = ((-0.5*freq*x(i))*(g^2)); % drift coefficient
57         %g_linha = 2*x(i)*1; % derived from g
58         gito = (1+(1*(xito(i)^2))); % difusion coefficient

```

```

59         fito = ((-0.5*freq*xito(i))*(gito^2)); % drift

```

```

        coefficient
59
60     gito_linha = 2*xito(i)*l; % derived from g
61
62
63     xito(i+1) = xito(i) + (fito*h) + (gito*dW(i)) + ...
64         +((0.5*gito*gito_linha)*((dW(i)^2)-h));
65
66
67     %


---


68     % Stratonovich Interpretation
69     %


---


70     gstra = (1+(l*(xstra(i)^2))); % difusion coefficient
71
72     fstra = (((-0.5*freq*xstra(i))*(gstra^2))-(l*xstra(i)*
73         gstra));
74     % drift coefficient
75
76     gstra_linha = 2*xstra(i)*l; % derived from g
77
78     %x(i+1) =x(i) + (f*h) + (g*dW(i)) + ((0.5*g*g_linha)*((dW
79         (i)^2)-h));
80     xstra(i+1) = xstra(i) + (fstra*h) + (gstra*dW(i)) + ...
81         +((0.5*gstra*gstra_linha)*(dW(i)^2));
82
83     end
84
85     return
86
87     % multiplicative noise
88
89 %function [xitozero ,xitoum ,xitodois ,xstrazero ,xstraum ,xstradois ]
    = ...
90     % sde_numericmethod_milstein(nm,h,dW,n,lzero ,lum ,ldois ,freq)
91     % xitozero=zeros(1,nm); % vector pre-allocation for results
92     % xitoum=zeros(1,nm); % vector pre-allocation for results
93     % xitodois=zeros(1,nm); % vector pre-allocation for results

```

```

90     %(Ito interpretation)
91
92 %     xstrazero=zeros(1,mm); % vector pre-allocation for results
93 %     xstraum=zeros(1,mm); % vector pre-allocation for results
94 %     xstradois=zeros(1,mm); % vector pre-allocation for results
95     %(Stratonovich interpretation)
96
97 %     xitozero(1)=1.0; % initial condition for Euler-Maruyama (
    Ito)
98 %     xitoum(1)=1.0; % initial condition for Euler-Maruyama (Ito)
99 %     xitodois(1)=1.0; % initial condition for Euler-Maruyama (
    Ito)
100
101 %     xstrazero(1)=1.0; % initial condition for Euler-Maruyama(
    Stratonovich)
102 %     xstraum(1)=1.0; % initial condition for Euler-Maruyama(
    Stratonovich)
103 %     xstradois(1)=1.0; % initial condition for Euler-Maruyama(
    Stratonovich)
104
105     %%%%%%%%% Important: For additive noise, both results will be
        the same
106 %     for i=1:n
107         %


---


108         % Ito Interpretation
109         %


---


110 %         gitozero = (1+(lzero*(xitozero(i)^2))); % difusion
        coefficient
111 %         gitoum = (1+(lum*(xitoum(i)^2))); % difusion
        coefficient
112 %         gitodois = (1+(ldois*(xitodois(i)^2))); % difusion
        coefficient
113
114 %         fitozero = ((-0.5*freq*xitozero(i))*(gitozero^2)); %
        drift coeffic.

```



```

115 %          fitoum = ((-0.5*freq*xitoum(i))*(gitoum^2)); % drift
      coefficient
116 %          fitodois = ((-0.5*freq*xitodois(i))*(gitodois^2)); %
      drift coeffic.
117
118 %          gitozero_linha = 2*xitozero(i)*lzero; % derived from g
119 %          gitoum_linha = 2*xitoum(i)*lum; % derived from g
120 %          gitodois_linha = 2*xitodois(i)*ldois; % derived from g
121
122 %          xitozero(i+1) = xitozero(i) + (fitozero*h) + (gitozero*
      dW(i)) +...
123 %          +((0.5*gitozero*gitozero_linha)*((dW(i)^2)-h));
124
125 %          xitoum(i+1) = xitoum(i) + (fitoum*h) + (gitoum*dW(i)) +
      ...
126 %          +((0.5*gitoum*gitoum_linha)*((dW(i)^2)-h));
127
128 %          xitodois(i+1) = xitodois(i) + (fitodois*h) + (gitodois*
      dW(i)) +...
129 %          +((0.5*gitodois*gitodois_linha)*((dW(i)^2)-h));
130
131
132 %
      _____

133 % Stratonovich Interpretation
134 %
      _____

135 %          gstrazero = (1+(lzero*(xstrazero(i)^2))); % difusion
      coefficient
136 %          gstraum = (1+(lum*(xstraum(i)^2))); % difusion
      coefficient
137 %          gstradois = (1+(ldois*(xstradois(i)^2))); % difusion
      coefficient
138
139 %          fstrazero = (((-0.5*freq*xstrazero(i))*(gstrazero^2)) -...
140 %          -(lzero*xstrazero(i)*gstrazero)); % drift
      coefficient

```

```

141 %      fstraum = (((-0.5*freq*xstraum(i))*(gstraum^2)) -...
142 %          -(lum*xstraum(i)*gstraum)); % drift coefficient
143 %      fstradois = (((-0.5*freq*xstradois(i))*(gstradois^2))
-...
144 %          -(ldois*xstradois(i)*gstradois)); % drift
coefficient
145
146 %      gstrazero_linha = 2*xstrazero(i)*lzero; % derived from
g
147 %      gstraum_linha = 2*xstraum(i)*lum; % derived from g
148 %      gstradois_linha = 2*xstradois(i)*ldois; % derived from
g
149
150
151 % x(i+1) =x(i) + (f*h) + (g*dW(i)) +((0.5*g*g_linha)*((dW
(i)^2)-h));
152 %      xstrazero(i+1) = xstrazero(i) + ...
153 %          + (fstrazero*h) + (gstrazero*dW(i)) + ...
154 %          +((0.5*gstrazero*gstrazero_linha)*(dW(i)^2));
155
156 %      xstraum(i+1) = xstraum(i) + ...
157 %          +(fstraum*h) + (gstraum*dW(i)) + ...
158 %          +((0.5*gstraum*gstraum_linha)*(dW(i)^2));
159
160 %      xstradois(i+1) = xstradois(i) + ...
161 %          +(fstradois*h) + (gstradois*dW(i)) + ...
162 %          +((0.5*gstradois*gstradois_linha)*(dW(i)^2));
163 %      end
164 %return
165 %

```

Algoritmo 7 - FUNÇÃO QUE GERA UMA SOLUÇÃO NUMÉRICA USANDO O MÉTODO DE HEUN

```

1 %


---


2 % sde_numericmethod_heun.m
3 %
4 % This function generates solution numeric using
5 % the numeric method heun.
6 %
7 % Input:
8 %
9 % nm – mesh number of nodes
10 % f – drift coefficient
11 % g – diffusion coefficient
12 % h – step size
13 % dW – white noise
14 % c – predictor
15 % n – number of elements
16 % l – lambda squared
17 % freq – oscillator frequency
18 %
19 % Output:
20 % z – vector with numeric solution of numeric method heun
21
22 %


---


23 % programmer: Sara Cristina Quintanilha Valente
24 % saravalenteq@gmail.com
25 %
26 % last update: May 21, 2019
27 %


---


28
29 %


---



```

```

30
31 % aditive noise
32
33 function z = sde_numericmethod_heun(nm,h,dW,n,l,freq)
34
35     z=zeros(1,nm); % vector pre-allocation for results
36
37     v=zeros(1,n);
38
39     z(1)=1.0; % initial condition for Heun
40
41     for i=1:n
42         g = (1+(1*(z(i)^2))); % difusion coefficient
43
44         f = (((-0.5*freq*z(i))*(g^2))-(1*z(i)*g)); % drift
            coefficient
45
46         c=(1+(1*v(i)^2)); % predictor
47
48         v(i)=z(i)+(g*dW(i));
49
50         z(i+1) = z(i) + (f*h) + (0.5*(g+c)*dW(i));
51     end
52 return
53
54 % multiplicative noise
55
56 %function [zzero ,zum, zdois] = ...
57 %     sde_numericmethod_heun(nm,h,dW,n,lzero ,lum, ldois ,freq)
58 %     zzero=zeros(1,nm); % vector pre-allocation for results
59 %     zum=zeros(1,nm); % vector pre-allocation for results
60 %     zdois=zeros(1,nm); % vector pre-allocation for results
61
62 %     vzero=zeros(1,n);
63 %     vum=zeros(1,n);
64 %     vdois=zeros(1,n);
65
66 %     zzero(1)=1.0; % initial condition for Heun
67 %     zum(1)=1.0; % initial condition for Heun

```

```

68 %     zdois(1)=1.0; % initial condition for Heun
69
70 %     for i=1:n
71 %         gzero = (1+(lzero*(zzero(i)^2))); % difusion
        coefficient
72 %         gum = (1+(lum*(zum(i)^2))); % difusion coefficient
73 %         gdois = (1+(ldois*(zdois(i)^2))); % difusion
        coefficient
74
75 %         fzero = (((-0.5*freq*zzero(i))*(gzero^2))-(lzero*zzero(i)*gzero));
        % drift coefficient
76 %         fum = (((-0.5*freq*zum(i))*(gum^2))-(lum*zum(i)*gum));
        % drift coefficient
77 %         fdois = (((-0.5*freq*zdois(i))*(gdois^2))-(ldois*zdois(i)*gdois));
        % drift coefficient
78
79 %         czero=(1+(lzero*vzero(i)^2)); % predictor
80 %         cum=(1+(lum*vum(i)^2)); % predictor
81 %         cdois=(1+(ldois*vdois(i)^2)); % predictor
82
83 %         vzero(i)=zzero(i)+(gzero*dW(i));
84 %         zzero(i+1) = zzero(i) + (fzero*h) + (0.5*(gzero+czero)*
        dW(i));
85
86 %         vum(i)=zum(i)+(gum*dW(i));
87 %         zum(i+1) = zum(i) + (fum*h) + (0.5*(gum+cum)*dW(i));
88
89 %         vdois(i)=zdois(i)+(gdois*dW(i));
90 %         zdois(i+1) = zdois(i) + (fdois*h) + (0.5*(gdois+cdois)*
        dW(i));
91
92 %     end
93 %return
94 %
95
96

```

Algoritmo 8 - FUNÇÃO QUE GERA UMA SOLUÇÃO NUMÉRICA USANDO O MÉTODO DE
MILSTEIN SEM DERIVADA

```

1 %


---


2 % sde_nunderivativefree_milstein.m
3 %
4 % This function generates solution numeric using
5 % the Derivative-free Milstein method.
6 %
7 % Input:
8 %
9 % nm - mesh number of nodes
10 % f - drift coefficient
11 % g - difusion coefficient
12 % h - step size
13 % dW - white noise
14 % n - number of elements
15 % l - lambda squared
16 % freq - oscillator frequency
17 %
18 % Output:
19 % kito - vector with numeric solution of numeric method
    Milstein
20 % (Ito interpretation)
21 % kstra - vector with numeric solution of numeric method
    Milstein
22 % (Stratonovich interpretation)
23 %


---


24 % programmer: Sara Cristina Quintanilha Valente
25 % saravalenteq@gmail.com
26 %
27 % last update: May 21, 2019
28 %


---



```

```

29
30 %

```

```

31
32 % aditive noise
33
34 function [kito ,kstra] = sde_numderivativefree_milstein(nm,h,dW,n
    ,l,freq)
35     kito=zeros(1,nm); % vector pre-allocation for results
36     %(Ito interpretation)
37     kstra=zeros(1,nm); % vector pre-allocation for results(
        Stratonovich
38     %interpretation)
39     kito(1)=1.0; % initial condition for Euler-Maruyama (Ito)
40     kstra(1)=1.0; % initial condition for Euler-Maruyama (
        Stratonovich)
41
42     %%% Important: For additive noise, both results will be
        the same
43     for i=1:n
44         %

```

```

45         % Ito Interpretation
46         %

```

```

47         gito = (1+(l*(kito(i)^2))); % difusion coefficient
48         fito = ((-0.5*freq*kito(i))*(gito^2)); % drift
            coefficient
49
50         jito = kito(i) + (fito*h) + (gito*sqrt(h));
51         gitoj = (1+(l*((jito)^2))); % difusion coefficient
52
53         kito(i+1) = kito(i) + (fito*h) + (gito*dW(i)) + ...
54             +((0.5*(1/sqrt(h))*(gitoj-gito))*((dW(i)^2)-h));
55
56         %

```

```

57     % Stratonovich Interpretation
58     %

```

```

59     gstra = (1+(1*(kstra(i)^2))); % difusion coefficient
60     fstra = (((-0.5*freq*kstra(i))*(gstra^2))-(1*kstra(i)*
61         gstra)); ...
62         % drift coefficient
63
64     jstra = kstra(i) + (fstra*h) + (gstra*sqrt(h));
65     gstraj = (1+(1*((jstra)^2))); % difusion coefficient
66
67     kstra(i+1) = kstra(i) + (fstra*h) + (gstra*dW(i)) + ...
68         +((0.5*(1/sqrt(h))*(gstraj-gstra))*((dW(i)^2)));
69
70     end
71     return
72
73     % multiplicative noise
74
75     %function [kitozero ,kitoum ,kitodois ,kstrazero ,kstraum ,kstradois]
76     = ...
77     sde_numderivativefree_milstein(nm,h,dW,n,lzero ,lum ,ldois ,
78     freq)
79     kitozero=zeros(1,nm); % vector pre-allocation for results
80     kitoum=zeros(1,nm); % vector pre-allocation for results
81     kitodois=zeros(1,nm); % vector pre-allocation for results
82     % (Ito interpretation)
83
84     kstrazero=zeros(1,nm); %vector pre-allocation for results(
85     Stratonovich
86
87     kstraum=zeros(1,nm); %vector pre-allocation for results(
88     Stratonovich
89
90     kstradois=zeros(1,nm); %vector pre-allocation for results(
91     Stratonovich
92     %interpretation)

```



```

86
87 %    kitzero(1)=1.0; % initial condition for Euler-Maruyama (
      Ito)
88 %    kitoum(1)=1.0; % initial condition for Euler-Maruyama (Ito)
89 %    kitodois(1)=1.0; % initial condition for Euler-Maruyama (
      Ito)
90
91 %    kstrazero(1)=1.0; %initial condition for Euler-Maruyama (
      Stratonovich)
92 %    kstraum(1)=1.0; %initial condition for Euler-Maruyama (
      Stratonovich)
93 %    kstradois(1)=1.0;% initial condition for Euler-Maruyama (
      Stratonovich)
94
95 %%% Important: For additive noise, both results will be
      the same
96 %    for i=1:n
97 %
      _____
98 %    Ito Interpretation
99 %
      _____

100 %    gitozero = (1+(lzero*(kitozero(i)^2))); %difusion
      coefficient
101 %    fitozero = ((-0.5*freq*kitozero(i))*(gitozero^2));%
      drift coeficent
102
103 %    gitoum = (1+(lum*(kitoum(i)^2))); % difusion
      coefficient
104 %    fitoum = ((-0.5*freq*kitoum(i))*(gitoum^2)); % drift
      coefficient
105
106 %    gitodois = (1+(ldois*(kitodois(i)^2))); % difusion
      coefficient
107 %    fitodois = ((-0.5*freq*kitodois(i))*(gitodois^2));%
      drift coeficent
108

```

```

109
110
111 %      jitozero = kitozero(i) + (fitozero*h) + (gitozero*sqrt(
      h));
112 %      gitojzero = (1+(lzero*((jitozero)^2))); % difusion
      coefficient
113
114 %      jitoum = kitoum(i) + (fitoum*h) + (gitoum*sqrt(h));
115 %      gitojum = (1+(lum*((jitoum)^2))); % difusion
      coefficient
116
117 %      jitodois = kitodois(i) + (fitodois*h) + (gitodois*sqrt(
      h));
118 %      gitojdois = (1+(ldois*((jitodois)^2))); % difusion
      coefficient
119
120
121
122 %      kitozero(i+1) = kitozero(i) + (fitozero*h) + (gitozero*
      dW(i)) + ...
123 %      +((0.5*(1/sqrt(h))*(gitojzero-gitozero))*((dW(i)^2)
      -h));
124
125 %      kitoum(i+1) = kitoum(i) + (fitoum*h) + (gitoum*dW(i)) +
      ...
126 %      +((0.5*(1/sqrt(h))*(gitojum-gitoum))*((dW(i)^2)-h))
      ;
127
128 %      kitodois(i+1) = kitodois(i) + (fitodois*h) + (gitodois*
      dW(i)) + ...
129 %      +((0.5*(1/sqrt(h))*(gitojdois-gitodois))*((dW(i)^2)
      -h));
130
131 %
      _____

132 % Stratonovich Interpretation
133 %
      _____

```

```

134 %          gstrazero = (1+(lzero*(kstrazero(i)^2))); % difusion
          coefficient
135 %          fstrazero = (((-0.5*freq*kstrazero(i))*(gstrazero^2))-
          ...
136 %          -(lzero*kstrazero(i)*gstrazero)); % drift
          coefficient
137
138 %          gstraum = (1+(lum*(kstraum(i)^2))); % difusion
          coefficient
139 %          fstraum = (((-0.5*freq*kstraum(i))*(gstraum^2))- ...
140 %          -(lum*kstraum(i)*gstraum)); % drift coefficient
141
142 %          gstradois = (1+(ldois*(kstradois(i)^2))); % difusion
          coefficient
143 %          fstradois = (((-0.5*freq*kstradois(i))*(gstradois^2))-
          ...
144 %          -(ldois*kstradois(i)*gstradois)); % drift
          coefficient
145
146
147
148 %          jstrazero = kstrazero(i) + (fstrazero*h) + (gstrazero*
          sqrt(h));
149 %          gstrajzero = (1+(lzero*((jstrazero)^2))); % difusion
          coefficient
150
151 %          jstraum = kstraum(i) + (fstraum*h) + (gstraum*sqrt(h));
152 %          gstrajum = (1+(lum*((jstraum)^2))); % difusion
          coefficient
153
154 %          jstradois = kstradois(i) + (fstradois*h) + (gstradois*
          sqrt(h));
155 %          gstrajdois = (1+(ldois*((jstradois)^2))); % difusion
          coefficient
156
157
158
159 %          kstrazero(i+1) = kstrazero(i)+(fstrazero*h)+(gstrazero*

```

```

    dW(i)) + ...
160 %          +((0.5*(1/sqrt(h))*(gstrajzero-gstrazero))*((dW(i)
    ^2)));
161
162 %          kstraum(i+1) = kstraum(i) + (fstraum*h) + (gstraum*dW(i)
    )) + ...
163 %          +((0.5*(1/sqrt(h))*(gstrajum-gstraum))*((dW(i)^2)))
    ;
164
165 %          kstradois(i+1) = kstradois(i)+(fstradois*h)+(gstradois*
    dW(i)) + ...
166 %          +((0.5*(1/sqrt(h))*(gstrajdois-gstradois))*((dW(i)
    ^2)));
167 %      end
168 %return
169
170
171
172
173 %

```

Algoritmo 9 - FUNÇÃO QUE GERA UMA SOLUÇÃO NUMÉRICA USANDO O MÉTODO DE
RUNGE-KUTTA

```

1  %


---


2  % sde_numericmethod_rungekutta.m
3  %
4  % This function generates solution numeric using
5  % the Stochastic Runge–Kutta.
6  %
7  % Input:
8  %
9  % nm – mesh number of nodes
10 % f – drift coefficient
11 % g – diffusion coefficient
12 % h – step size
13 % dW – white noise
14 % n – number of elements
15 % l – lambda squared
16 % freq – oscillator frequency
17 %
18 % Output:
19 % r – vector with numeric solution of numeric method Runge–
20 % Kutta


---


21 % programmer: Sara Cristina Quintanilha Valente
22 % saravalenteq@gmail.com
23 %
24 % last update: May 21, 2019
25 %


---


26
27 %


---



```

```

28
29 % aditive noise
30
31 function r = sde_numericmethod_rungekutta(mm,h,dW,n,l,freq)
32     r=zeros(1,mm); % vector pre-allocation for results
33     r(1)=1.0; % initial condition for Runge-Kutta
34
35     for i=1:n
36         g = (1+(l*(r(i)^2))); % difusion coefficient
37         f = (((-0.5*freq*r(i))*(g^2))-(l*r(i)*g)); % drift
           coefficient
38
39         g_linha = 2*r(i)*l; % derived from g
40         g_duaslinhas= 2*r(i); % second derived from g
41
42         %%%%f_linha = (-0.5*freq*r(i)^2) + (-l*r(i)^3*freq) +
           ...
43         %%%%      + (-0.5*l^2*freq*r(i)^5); %derived from f
44         %%%%f_duaslinhas = -freq + (-6*l*freq*r(i)) + ...
45         %%%%      + (-10*l^2*freq*r(i)^3); %second derived from f
46
47         f_linha = ((-0.5*freq*((1+(4*l*r(i)^2))+(5*(l^2)*(r(i)
           ^4))))- ...
48         (l*(1+3*l*r(i)^2))); %derived from f
49
50         f_duaslinhas =((-0.5*freq*((8*(l^2)*r(i))-(20*(l^4)*(r(i)
           ^3))))-...
51         (6*(l^2)*r(i))); %second derived from f
52
53         deltaZ = dW(i)*h; %vari vel aleatoria
54
55         r(i+1) = r(i) + (f*h) + (g*dW(i)) + ...
56             + ((0.5*g*g_linha)*((dW(i)^2)-h)) + ...
57             + (f_linha*g*deltaZ) + (0.5*((f*f_linha)+ ...
58             (0.5*g^2*f_duaslinhas))*h^2) + ...
59             + ((f*g_linha)+(0.5*g^2*g_duaslinhas))*((dW(i)*h)-
           deltaZ) + ...
60             +(0.5*g*(g*g_duaslinhas)+...
61             +(g_linha)^2)*((1/3*(dW(i)^2)-h)*dW(i));

```

```

62     end
63     return
64
65     % multiplicative noise
66
67     %function [rzero ,rum,rdois] = ...
68     %     sde_numericmethod_rungekutta(nm,h,dW,n,lzero ,lum ,ldois ,freq
69     %     )
70
71     %     rzero=zeros(1,nm); % vector pre-allocation for results
72     %     rum=zeros(1,nm); % vector pre-allocation for results
73     %     rdois=zeros(1,nm); % vector pre-allocation for results
74
75     %     rzero(1)=1.0; % initial condition for Runge-Kutta
76     %     rum(1)=1.0; % initial condition for Runge-Kutta
77     %     rdois(1)=1.0; % initial condition for Runge-Kutta
78
79     %     for i=1:n
80     %         gzero = (1+(lzero*(rzero(i)^2))); % difusion
81     %         gum = (1+(lum*(rum(i)^2))); % difusion coefficient
82     %         gdois = (1+(ldois*(rdois(i)^2))); % difusion
83     %         fzero = ((-0.5*freq*rzero(i))*(gzero^2)); % drift
84     %         fum = ((-0.5*freq*rum(i))*(gum^2)); % drift coefficient
85     %         fdois = ((-0.5*freq*rdois(i))*(gdois^2)); % drift
86     %         coefficient
87     %         gzero_linha = 2*rzero(i)*lzero; % derived from g
88     %         gum_linha = 2*rum(i)*lum; % derived from g
89     %         gdois_linha = 2*rdois(i)*ldois; % derived from g
90
91     %         gzero_duaslinhas= 2*rzero(i); % second derived from g
92     %         gum_duaslinhas= 2*rum(i); % second derived from g
93     %         gdois_duaslinhas= 2*rdois(i); % second derived from g
94
95

```

```

96 %      fzero_linha = ((-0.5*freq*((1+(4*lzero*rzero(i)^2))+...
97 %          +(5*(lzero^2)*(rzero(i)^4))))- ...
98 %          - (lzero*(1+3*lzero*rzero(i)^2))); %derived from f
99
100 %      fum_linha = ((-0.5*freq*((1+(4*lum*rum(i)^2))+...
101 %          +(5*(lum^2)*(rum(i)^4))))- ...
102 %          - (lum*(1+3*lum*rum(i)^2))); %derived from f
103
104 %      fdois_linha = ((-0.5*freq*((1+(4*ldois*rdois(i)^2))+...
105 %          +(5*(ldois^2)*(rdois(i)^4))))- ...
106 %          - (ldois*(1+3*ldois*rdois(i)^2)));%derived from f
107
108
109 %      fzero_duaslinhas = ((-0.5*freq*((8*(lzero^2)*rzero(i))
- ...
110 %          -(20*(lzero^4)*(rzero(i)^3)))) -...
111 %          - (6*(lzero^2)*rzero(i))); %second derived from f
112
113 %      fum_duaslinhas = ((-0.5*freq*((8*(lum^2)*rum(i)) -...
114 %          -(20*(lum^4)*(rum(i)^3)))) -...
115 %          - (6*(lum^2)*rum(i))); %second derived from f
116
117 %      fdois_duaslinhas = ((-0.5*freq*((8*(ldois^2)*rdois(i))
- ...
118 %          -(20*(ldois^4)*(rdois(i)^3)))) -...
119 %          - (6*(ldois^2)*rdois(i))); %second derived from f
120
121 %      deltaZ = dW(i)*h; %vari vel aleatoria
122
123 %      rzero(i+1) = rzero(i) + (fzero*h) + (gzero*dW(i)) + ...
124 %          + ((0.5*gzero*gzero_linha)*(dW(i)^2)-h) + ...
125 %          + (fzero_linha*gzero*deltaZ) + (0.5*((fzero*
fzero_linha)+ ...
126 %          + (0.5*gzero^2*fzero_duaslinhas))*h^2) + ...
127 %          + ((fzero*gzero_linha)+...
128 %          + (0.5*gzero^2*gzero_duaslinhas))*((dW(i)*h)-deltaZ)
+ ...
129 %          + (0.5*gzero*(gzero*gzero_duaslinhas)+...
130 %          + (gzero_linha)^2)*((1/3*(dW(i)^2)-h)*dW(i));

```



```

131
132 %      rum(i+1) = rum(i) + (fum*h) + (gum*dW(i)) + ...
133 %          + ((0.5*gum*gum_linha)*((dW(i)^2)-h)) + ...
134 %          + (fum_linha*gum*deltaZ) + (0.5*((fum*fum_linha)+
...
135 %          + (0.5*gum^2*fum_duaslinhas))*h^2) + ...
136 %          + ((fum*gum_linha) + ...
137 %          + (0.5*gum^2*gum_duaslinhas))*((dW(i)*h)-deltaZ) +
...
138 %          +(0.5*gum*(gum*gum_duaslinhas) + ...
139 %          +(gum_linha)^2)*((1/3*(dW(i)^2)-h)*dW(i));
140
141 %      rdois(i+1) = rdois(i) + (fdois*h) + (gdois*dW(i)) + ...
142 %          + ((0.5*gdois*gdois_linha)*((dW(i)^2)-h)) + ...
143 %          + (fdois_linha*gdois*deltaZ) + (0.5*((fdois*
fdois_linha)+ ...
144 %          + (0.5*gdois^2*fdois_duaslinhas))*h^2) + ...
145 %          + ((fdois*gdois_linha) + ...
146 %          + (0.5*gdois^2*gdois_duaslinhas))*((dW(i)*h)-deltaZ)
+ ...
147 %          +(0.5*gdois*(gdois*gdois_duaslinhas) + ...
148 %          +(gdois_linha)^2)*((1/3*(dW(i)^2)-h)*dW(i));
149 %      end
150 %return
151
152
153
154 %

```
