



Waldir Jesus de Araujo Lobão

**Solução de Equações Diferenciais Ordinárias,
Parciais e Estocásticas por Programação Genética
e Diferenciação Automática**

Tese de Doutorado

Tese apresentada ao programa de Pós-Graduação em Engenharia Elétrica do Departamento de Engenharia Elétrica da PUC-Rio como parte dos requisitos parciais para obtenção do título de Doutor em Engenharia Elétrica.

Orientador: Prof. Marco Aurélio Cavalcanti Pacheco

Rio de Janeiro

Abril de 2015



Waldir Jesus de Araujo Lobão

Solução de Equações Diferenciais Ordinárias, Parciais e Estocásticas por Programação Genética e Diferenciação Automática

Tese apresentada como requisito parcial para obtenção do grau de Doutor pelo Programa de Pós-Graduação em Engenharia Elétrica do Departamento de Engenharia Elétrica do Centro Técnico Científico da PUC-Rio Aprovada pela Comissão Examinadora abaixo assinada.

Prof. Marco Aurélio Cavalcanti Pacheco

Orientador

Departamento de Engenharia Elétrica – PUC-Rio

Prof. Douglas Mota Dias

Co-Orientador

Departamento de Engenharia Elétrica – PUC-Rio

Prof. André Vargas Abs da Cruz

UEZO

Profa. Carla Silva Oliveira

Instituto Brasileiro de Geografia e Estatística

Prof. José Andre de Moura Brito

Escola Nacional de Ciências e Estatísticas

Prof. Valmir Carneiro Barbosa

UFRJ

Prof. Iury Steiner de Oliveira Bezerra

Instituto de Resseguros do Brasil, IIRB_FORN, Brasil

Prof. Tara Keshar Nanda Baidya

Universidade do Grande Rio

Prof. José Eugenio Leal

Coordenador Setorial do Centro

Técnico Científico – PUC-Rio

Rio de Janeiro, 09 de abril de 2015

Todos os direitos reservados. É proibida a reprodução total ou parcial do trabalho sem autorização da universidade, do autor e do orientador.

Waldir Jesus de Araujo Lobão

Economista, Mestre em Estatística pela UFRJ. Pesquisador concursado da F.IBGE, lotado na Escola Nacional de Ciências Estatística - ENCE, onde atua como professor e pesquisador. Professor concursado da Escola Naval da Marinha do Brasil, onde leciona e coordena a cadeira de Estatística. Com interesse nas áreas acadêmicas de Computação Evolucionária; Programação Genética; Estatística; Processos Estocásticos; Econometria de Séries Temporais; Economia Aplicada e Finanças. Autor de trabalhos publicados nas áreas de Demanda por Energia Elétrica; Tarifação e Demanda por Água; Crime e Violência Urbana.

Ficha Catalográfica

Lobão, Waldir Jesus de Araujo

Solução de equações diferenciais ordinárias, parciais e estocásticas por programação genética e diferenciação automática / Waldir Jesus de Araujo Lobão; orientador: Marco Aurélio Cavalcanti Pacheco. – 2015.

171 f. ; 30 cm

Tese (doutorado) – Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Engenharia Elétrica, 2015.

Inclui bibliografia

1. Engenharia elétrica – Teses. 2. Computação evolucionária. 3. Programação genética. 4. Algoritmo. 5. Diferenciação automática. 6. Equações diferenciais ordinárias, parciais e estocásticas. I. Pacheco, Marco Aurélio Cavalcanti. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Engenharia Elétrica. III. Título.

CDD: 621.3

“Em Memória de Meu Pai.”

Agradecimentos

Em especial, ao meu orientador, Prof. Dr. Marco Aurélio C. Pacheco, e ao meu co-orientador, Prof. Dr. Douglas Mota Dias, pelos ensinamentos e orientações recebidas e pela forma cordial e amigável que sempre me trataram.

Aos Professores membros da banca examinadora pela aceitação ao convite de participação, pela gentileza de ler e avaliar este extenso trabalho, e pelas importantes sugestões que contribuíram significativamente para a melhoria e aprimoramento desta tese.

À PUC-Rio, à F.IBGE-ENCE e à Escola Naval da Marinha do Brasil, pelos auxílios concedidos, sem os quais este trabalho não poderia ter sido realizado.

Aos professores, colegas e funcionários da PUC-Rio pelos ensinamentos, amizade e ajuda recebida de todos.

Ao meu grande amigo Daniel Ricardo de Castro Cerqueira, parceiro em diversos trabalhos, que sempre me deu forças e acreditou que seria possível.

A todas as pessoas que, direta ou indiretamente, apoiaram e contribuíram para realização deste trabalho.

Por fim, agradeço à minha família pela compreensão e pelo apoio recebido ao longo desta difícil jornada.

Resumo

Lobão, Waldir Jesus de Araujo Lobão; Pacheco, Marco Aurélio Cavalcanti (Orientador). **Solução de equações diferenciais ordinárias, parciais e estocásticas por programação genética e diferenciação automática.** Rio de Janeiro, 2015. 171p. Tese de Doutorado - Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro.

O presente trabalho teve como objetivo principal investigar o potencial de algoritmos computacionais evolutivos, construídos a partir das técnicas de programação genética, combinados com diferenciação automática, na obtenção de soluções analíticas, exatas ou aproximadas, para problemas de equações diferenciais ordinárias (EDO), parciais (EDP) e estocásticas. Com esse intuito, e utilizando-se o ambiente de programação Matlab, diversos algoritmos foram elaborados e soluções analíticas de diferentes tipos de equações diferenciais foram determinadas. No caso das equações determinísticas, EDOs e EDPs, foram abordados problemas de diferentes graus de dificuldade, do básico até problemas complexos como o da equação do calor e a equação de Schrödinger para o átomo de hélio. Os resultados obtidos são promissores, com soluções exatas para a grande maioria dos problemas tratados e que atestam, empiricamente, a consistência e robustez da metodologia proposta. Com relação às equações estocásticas, o trabalho apresenta uma nova proposta de solução e metodologia alternativa para a precificação de opções europeias, de compra e de venda, e realiza algumas aplicações para o mercado brasileiro, com ações da Petrobras e da Vale. Além destas aplicações, são apresentadas as soluções de alguns modelos clássicos, usualmente utilizados na modelagem de preços e retornos de ativos financeiros, como, por exemplo, o movimento Browniano geométrico. De uma forma geral, os resultados obtidos nas aplicações indicam que a metodologia proposta nesta tese pode ser uma alternativa eficiente na modelagem de problemas científicos complexos.

Palavras-chave

Computação evolucionária; programação genética; algoritmo; diferenciação automática; equações diferenciais ordinárias, parciais e estocásticas.

Abstract

Lobão, Waldir Jesus de Araujo; Marco Aurélio c. Pacheco (Advisor). **Solution of ordinary, partial and stochastic differential equations by genetic programming and automatic differentiation.** Rio de Janeiro, 2015. 171p. PhD Thesis – Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro.

The main objective of this work was to investigate the potential of evolutionary algorithms, built from genetic programming techniques and combined with automatic differentiation, in obtaining exact or approximate analytical solutions for problems of ordinary (ODE), partial (PDE), and stochastic differential equations. To this end, and using the Matlab programming environment, several algorithms were developed and analytical solutions of different types of differential equations were determined. In the case of deterministic equations, ODE and PDE problems of varying degrees of difficulty were discussed, from basic to complex problems such as the heat equation and the Schrödinger equation for the helium atom. The results are promising, including exact solutions for the vast majority of the problems treated, which attest empirically the consistency and robustness of the proposed methodology. Regarding the stochastic equations, the work presents a new proposal for a solution and alternative methodology for European options pricing, buying and selling, and performs some applications for the Brazilian market, with stock prices of Petrobras and Vale. In addition to these applications, there are presented solutions of some classical models, usually used in the modeling of prices and returns of financial assets, such as the geometric Brownian motion. In a general way, the results obtained in applications indicate that the methodology proposed in this dissertation can be an efficient alternative in modeling complex scientific problems.

Keywords

Evolutionary computation; genetic programming; algorithm; automatic differentiation; ordinary differential equations, stochastic and partial.

Sumário

1	Introdução	14
1.1.	Motivação e Objetivos	14
1.2.	Descrição do Trabalho e Contribuições	15
1.3.	Revisão da Literatura	16
1.4.	Organização do Trabalho	20
2	Fundamentos Teóricos	21
2.1.	Introdução	21
2.2.	Programação Genética: Aspectos Gerais	21
2.2.1.	Formas de Representação	24
2.2.2.	Funções e Terminais	25
2.2.3.	Fechamento e Suficiência	26
2.2.4.	População Inicial	27
2.2.5.	Função de Aptidão e Avaliação dos Indivíduos	29
2.2.6.	Métodos de Seleção para Reprodução	32
2.2.7.	Operadores Genéticos	34
2.2.8.	Métodos de Reprodução	36
2.2.9.	Parâmetros e Critério de Parada	37
2.3.	Diferenciação Automática	38
2.3.1.	A Regra da Cadeia e os modos <i>Forward</i> e <i>Reverse</i>	39
2.3.2.	Softwares e Linguagens de Diferenciação Automática	41
2.4.	Equações Diferenciais	44
2.4.1.	Introdução	44
2.4.2.	Equações Diferenciais Ordinárias e Parciais	45
2.4.3.	Equações Diferenciais Estocásticas	48
3	Algoritmo de Programação Genética e Diferenciação Automática	54
3.1.	Introdução	54
3.2.	Estrutura e Etapas de Funcionamento do algoritmo de PGDA	54
3.2.1.	Parametrização do Algoritmo	55
3.2.2.	População Inicial e Avaliação dos Indivíduos	56
3.2.3.	Reprodução e Geração da Nova População	58
3.2.4.	Critério de Parada e a Solução de PGDA	59
4	Aplicações e Análise dos Resultados	61
4.1.	Solução de Equações Diferenciais por PGDA	61
4.1.1.	Solução de Equações Diferenciais Ordinárias	62
4.1.2.	Solução de Equações Diferenciais Parciais	71
4.1.3.	Solução da Equação de Schrödinger para o Átomo de He	75
4.1.4.	Solução de Modelos com Especificação Arbitrária	86
4.2.	Solução de Equações Diferenciais Estocásticas por PGDA	90
4.2.1.	Algoritmo de PGDA para Solucionar EDEs	90
4.2.2.	Solução da EDE de MBG por PGDA - Exemplo 1	96
4.2.3.	Solução da EDE de MBG por PGDA - Exemplo 2	112
4.2.4.	Modelagem dos Preços da Ação PETR4 - Exemplo 3	114
4.2.5.	Modelo de Precificação de Opções Europeias por PGDA	126

5 Conclusões e Trabalhos Futuros	136
6 Referências bibliográficas	140
7 Apêndice	147
7.1. Codes Matlab Utilizados nas Simulações Realizadas	147
7.1.1. Code para Simulação da EDE de MBG	147
7.1.2. Code de Solução da EDE de MBG por Aproximação de Euler	148
7.2. Code Matlab de Bootstrap para Cálculo de Erro Padrão	149
7.3. Resultados Adicionais dos Modelos Estimados para a PETR4	150
7.3.1. Análise dos Resíduos da Solução de PGDA para a PETR4	150
7.3.2. Resultados do Modelo ARCH para a PETR4	152
7.3.3. Resultados do Modelo GARCH para a PETR4	153
7.3.4. Resultados do Modelo EGARCH para a PETR4	154
7.4. Resultados dos Modelos Estimados para a VALE5 - Exemplo 4	155
7.4.1. Resultados da Solução de PGDA para a VALE5	155
7.4.2. Resultados do Modelo ARCH para a VALE5	161
7.4.3. Resultados do Modelo GARCH para a VALE5	162
7.4.4. Resultados do Modelo EGARCH para a VALE5	163
7.5. Fórmulas de Cálculo do Prêmio de Calls e Puts	164
7.6. Identificação das Opções Negociadas na BM&FBovespa	168

Lista de figuras

Figura 1: Fluxograma de um algoritmo tradicional de PG	23
Figura 2: Exemplo de Representação em forma de Árvore	24
Figura 3: Árvore produzida com a Inicialização <i>Grow</i>	28
Figura 4: Árvore produzida com a Inicialização <i>Full</i>	28
Figura 5: Exemplo de Crossover <i>one-point</i>	35
Figura 6: Exemplo de Mutação Padrão	36
Figura 7: Aplicação de DA em Code Matlab, utilizando myAD	42
Figura 8: Aplicação de DA em Code Matlab, utilizando myA2D	43
Figura 9: Exemplo 1 - Resultados da EDO1	63
Figura 10: Exemplo 2 - Resultados da EDO2	64
Figura 11: Exemplo 3 - Resultados da EDO3	65
Figura 12: Exemplo 4 - Resultados da EDO4	66
Figura 13: Exemplo 5 - Resultados da EDO5	67
Figura 14: Exemplo 6 - Resultados da EDO6	68
Figura 15: Exemplo 7 - Resultados da EDONL1	69
Figura 16: Exemplo 8 - Resultados da EDONL2	70
Figura 17: Exemplo 9 - Solução da EDP1	72
Figura 18: Exemplo 10 – Solução da EDP2	73
Figura 19: Exemplo 11 – Solução da EDP3	74
Figura 20: Coordenadas do Átomo de Hélio	75
Figura 21: Resultados do Caso 1 da Equação de Schrödinger	81
Figura 22: Resultados do Caso 2 da Equação de Schrödinger	82
Figura 23: Resultados do Caso 3 da Equação de Schrödinger	83
Figura 24: Resultados da Estimação do Modelo por PGDA	89
Figura 25: Comparação entre os Ajustes dos Modelos Estimados	89
Figura 26: Trajetória do Processo Simulado de MBG	97
Figura 27: Especificação Inicial da EDE de MBG por PGDA	99
Figura 28: Trajetória de Ruído Branco e Teste de Normalidade	101
Figura 29: Correlograma e Testes de Ruído Branco	102
Figura 30: Trajetória do Processo de Wiener Estimada por PGDA	102

Figura 31: Especificação Final da EDE de MBG por PGDA	104
Figura 32: Solução da EDE transformada de MBG por PGDA	105
Figura 33: Trajetória Simulada e Soluções por PGDA e SimByEuler	108
Figura 34: Diferenças entre a Trajetória Simulada e as Soluções	108
Figura 35: Comparações das Tendências e Volatilidades	110
Figura 36: Trajetórias das Soluções por PGDA e SimByEuler	113
Figura 37: Comparações entre Tendências e Volatilidades	114
Figura 38: Preços Nominais Diários de Fechamento da PETR4	116
Figura 39: Média Condicional dos Log-Retornos da PETR4	118
Figura 40: Volatilidade Estimada para os Log-Retornos da PETR4	119
Figura 41: Solução de PGDA para a PETR4	121
Figura 42: Tendência e Volatilidade Total para a PETR4	122
Figura 43: Estimativas dos Modelos para os Preços da PETR4	124
Figura 44: Estimativas das Volatilidades dos Modelos para a PETR4	125

Lista de quadros

Quadro 1: DA - Exemplo do Modo <i>Forward</i>	40
Quadro 2: Funções Elementares Utilizadas pelo Algoritmo de PG	56
Quadro 3: Parametrização do Algoritmo PGDA	61
Quadro 4: Estimativa da Energia E_0 para o Átomo de He	85
Quadro 5: Medidas de Erro e da Qualidade das Estimativas	108
Quadro 6: Medidas das Estimativas da Tendência e da Volatilidade	110
Quadro 7: Erros Padrão e Testes para Tendência e Volatilidade	111
Quadro 8: Estatísticas da Qualidade das Estimativas e Testes	114
Quadro 9: Estatísticas de Erro Padrão e Testes – PETR4	123
Quadro 10: Estatísticas das Estimativas dos Modelos – PETR4	125
Quadro 11: Estimativas de Prêmios Justos de Opções de Compra	134
Quadro 12: Estimativas de Prêmios Justos de Opções de Venda	135

Siglas

ARCH: Autoregressive Conditional Heteroskedasticity

DA: Diferenciação Automática

GARCH: Generalized Autoregressive Conditional Heteroskedasticity

Gplab: Genetic Programming Toolbox for MATLAB

EDE: Equação Diferencial Estocástica

EDO: Equação Diferencial Ordinária

EDP: Equação Diferencial Parcial

EGARCH: Exponential GARCH

MAE: Erro Absoluto Médio

MAPE: Erro Percentual Absoluto Médio

Matlab: The language of technical computing

MBG: Movimento Browniano Geométrico

Petrobras: Empresa Petróleo Brasileiro S. A.

PETR4: Ação Preferencial Nominativa da Petrobras

PG: Programação Genética

PGDA: Programação Genética com Diferenciação Automática

REQM: Raiz do Erro Quadrático Médio

VALE: Empresa VALE S. A.

VALE5: Ação Preferencial Nominativa da VALE

1

Introdução

1.1

Motivação e Objetivos

É de amplo conhecimento que um número expressivo de projetos de pesquisa realizados por diversas áreas do conhecimento científico, como a engenharia, economia, física, química, biologia, etc., utilizam-se de modelos matemáticos que são formulados e especificados, parcialmente ou totalmente, por equações diferenciais. No entanto, dada à complexidade dos modelos propostos, os seus formuladores quase sempre se deparam com problemas de difícil solução e com solução analítica desconhecida. Nesta situação, o procedimento usual é a obtenção de uma solução aproximada através de métodos numéricos.

Contudo, grande parte destes projetos objetivam resultados maiores e usam as equações diferenciais para tentar conhecer e entender o comportamento dinâmico de variáveis que são objetos centrais destas pesquisas. Nestes casos, certamente, a solução numérica não é suficientemente completa para dar aos formuladores as respostas requeridas para as suas avaliações. Faz-se necessário o conhecimento da função solução na sua forma literal, pois esta permite a realização de importantes e diferentes tipos de análises, tais como: estáticas comparativas; conhecimento da magnitude de efeitos parciais e elasticidades; determinação de tendências, ciclos e volatilidades; estudos de estabilidade e estacionariedade; realização de prognósticos e avaliação de cenários futuros; etc.

Outra questão importante, comumente verificada na prática, ocorre quando os modelos são especificados de forma arbitrária (*ad hoc*), isto é, sem critérios e compromissos metodológicos. Este tipo de procedimento pode levar o modelo a produzir resultados completamente espúrios que, quando avaliados com testes adequados, não apresentam nenhuma evidência empírica. Então, a geração de soluções analíticas para estes modelos, identificando uma forma funcional mais adequada, deverá corrigir o problema e propiciar avanços significativos nos resultados da modelagem.

Com a motivação de contribuir com novos resultados e metodologias que ajudem a dirimir os problemas acima colocados; o objetivo principal desta tese é investigar o potencial de algoritmos computacionais evolutivos, construídos a partir da combinação das técnicas de programação genética e diferenciação automática (PGDA), na obtenção de soluções analíticas, exatas ou aproximadas, para problemas de equações diferenciais ordinárias (EDOs), parciais (EDPs) e estocásticas (EDEs).

1.2

Descrição do Trabalho e Contribuições

Com a finalidade de realizar o objetivo proposto, e utilizando o ambiente de programação Matlab, diversos algoritmos de PGDA foram elaborados e soluções analíticas para diferentes problemas de EDOs, EDPs e EDEs foram determinadas.

Inicialmente o trabalho abordou as equações determinísticas, EDOs e EDPs, derivadas de diversos problemas, com diferentes graus de dificuldade, e que abrangem grande parte da teoria existente. Os estudos iniciaram-se com as EDOs mais simples, do tipo linear e de primeira ordem, até chegar às EDPs mais complexas, como a equação do calor e a equação de Schrödinger para o átomo de He. Os resultados obtidos são promissores, com soluções exatas para a grande maioria dos problemas abordados e que atestam, empiricamente, a consistência e robustez da metodologia proposta.

Com relação às equações estocásticas, abordadas na segunda fase da pesquisa, o trabalho apresenta uma metodologia de solução analítica em tempo contínuo, para EDEs representadas na forma tradicional, onde o algoritmo de PGDA é construído com base no cálculo estocástico de Itô. A metodologia foi testada através de simulações do modelo clássico de movimento browniano geométrico, usualmente utilizado na modelagem de preços e retornos de ativos financeiros. O trabalho ainda propõe um modelo alternativo para precificação de opções europeias, de compra e de venda, e realiza algumas aplicações para o mercado brasileiro, com ações da Petrobras e da Vale. Em geral, os resultados obtidos nas aplicações indicam que os métodos propostos são uma alternativa eficiente na solução de EDEs e precificação de ativos financeiros.

1.3

Revisão da Literatura

Para comparar os estudos realizados na tese aos do mesmo gênero existentes na literatura, bem como deixar clara a contribuição do trabalho, uma revisão bibliográfica foi realizada e os principais resultados são comentados a seguir.

Um grande número de trabalhos que tratam de temas relacionados à tese foi pesquisado, mas, embora exista uma vasta literatura sobre PG e DA, a grande maioria dos trabalhos abordam e aplicam estas técnicas de forma separada. Além disso, dentre os que desenvolvem PG e DA conjuntamente são raros aqueles que apresentam estudos sobre a solução de equações diferenciais e, especificamente no caso das EDEs, nenhum trabalho foi encontrado. Dentre eles destacam-se os artigos de (Imae, 2004; Tsoulus, 2006; Figueira, 2011).

Em Imae (2004), um algoritmo de PG com DA é proposto para resolver equações de Hamilton-Jacobi-Bellman (HJB); Hamilton-Jacobi-Isaacs (HJI); e Francis-Byrnes-Isidori (FBI) em sistemas dinâmicos não lineares de problemas de controle ótimo. O algoritmo é aplicado em equações com soluções analíticas já conhecidas e obtém soluções aproximadas com baixo erro de estimativa em todos os casos simulados. A metodologia utiliza a técnica de PG convencional, combinada com método de DA desenvolvido originalmente pelos próprios autores. As conclusões do artigo são de que, dada a complexidade dos problemas abordados, a metodologia apresenta bons resultados e é uma boa alternativa para a solução de problemas de controle ótimo.

No trabalho de Figueira (2011) os dois métodos são combinados para resolver duas equações diferenciais ordinárias, cujas soluções analíticas já eram conhecidas a priori. A equação do Oscilador Harmônico Simples, do sistema massa-mola; e a equação de Schrödinger para o átomo de Hidrogênio. O trabalho obtém soluções exatas para as duas equações propostas e utiliza os toolboxes GPLAB e myAD para a PG e DA, respectivamente. O autor conclui o trabalho sugerindo que essa combinação de técnicas é de grande utilidade quando aplicadas à resolução analítica de equações diferenciais.

Dentre os artigos pesquisados, o de Tsoulus (2006) é o que apresenta o mais diversificado e maior número de exemplos de soluções de equações diferenciais utilizando a metodologia de PG com DA. São apresentadas soluções para

problemas de EDOs e EDPs, lineares e não lineares, de primeira e segunda ordem. Soluções analíticas exatas são encontradas para a grande maioria dos problemas propostos. A metodologia do trabalho utiliza evolução gramatical para o algoritmo de PG, combinada com diferentes métodos de DA, para maiores detalhes sobre os métodos de DA utilizados, veja (Griewank, 1989; Bischof, 1992; Cusdin, 2003; Stauning, 2003). Os autores concluem o trabalho afirmando que a metodologia é muito adequada para solucionar problemas de EDO e EDP e que os resultados obtidos são muito encorajadores.

Embora não se tenha encontrado na pesquisa bibliográfica realizada nenhum trabalho que utilize a metodologia de PGDA para solucionar EDEs, deve-se informar que existem outras metodologias e uma extensa e sólida literatura que estuda especificamente as EDEs. Neste sentido, inúmeros são os artigos e livros que tratam da matéria e, necessariamente, do problema daquelas que não possuem solução analítica conhecida. Este é um problema que atinge diretamente a modelagem quantitativa de séries financeiras, em particular os modelos de precificação de opções, que é um dos temas de interesse desta tese. Diversos são os métodos desenvolvidos com a finalidade de obter soluções para EDEs, onde, os mais conhecidos e utilizados baseiam-se em estimação estatística, métodos numéricos e métodos de simulação estocástica. Dentre as referências consultadas e utilizadas na tese, embora existam outras tão boas quanto, destacam-se (Black, 1973; Arnoud, 1974; Kloeden, 1992; Campbell et al., 1997; Bollerslev, 1986, 1994; Nicolau, 2000).

Os trabalhos de Arnoud (1974) e Kloeden & Platen (1992) são livros amplamente conhecidos que tratam, com extremo rigor e formalismo matemático, da teoria das EDEs. Nestes livros, além da apresentação da teoria de cálculo estocástico, encontram-se também capítulos destinados a métodos de estimação em tempo contínuo, baseados em inferência estatística e métodos numéricos, para os parâmetros de EDEs discretizadas.

Em Nicolau (2000) é apresentada, também, uma descrição da teoria de cálculo estocástico, mas se concentra mais fortemente nos aspectos teóricos da modelagem e estimação estatística de séries financeiras. O objetivo principal do trabalho é apresentar novos métodos de estimação, em tempo contínuo, para os parâmetros de modelos especificados por EDEs. O trabalho ainda apresenta duas

aplicações com séries financeiras, uma para a modelagem da taxa de câmbio (dólar/marco) e outra para um índice de bolsa de valores.

O artigo de Black & Scholes (1973) é pioneiro em finanças e apresenta o famoso modelo de precificação de opções que leva o nome dos autores. Neste artigo, a EDE utilizada para determinar a dinâmica dos preços do ativo-objeto é o conhecido movimento browniano geométrico (MBG). O livro de (Campbell et al., 1997) é referência básica e um dos mais completos sobre econometria aplicada a finanças. O livro reúne grande parte das técnicas de mensuração e métodos quantitativos clássicos aplicados a finanças. Inclusive, trata da estimação EDEs em tempo discreto e contínuo, com volatilidade constante e estocástica. Métodos baseados em estimação de máxima verossimilhança, momentos generalizados, simulações estocásticas, aproximações numéricas, também são apresentados.

Os artigos de Bollerslev (1986, 1994) são de grande importância, pois introduzem na literatura os conhecidos modelos ARCH e GARCH¹. A partir da divulgação destes artigos uma família de modelos derivados destes foi criada. Na literatura atual de séries temporais e econometria aplicada a finanças os modelos da família ARCH são os mais utilizados para modelar séries financeiras em tempo discreto. Isso ocorre devido ao bom desempenho destes modelos, principalmente, na estimação da volatilidade da série temporal de interesse. Contudo, são especificados em tempo discreto e, portanto, não possuem as propriedades teóricas necessárias para representar a dinâmica de processos estocásticos em estados e tempo contínuo. Deve-se ressaltar que, a estimação estatística dos parâmetros de uma EDE discretizada é apenas uma aproximação da verdadeira solução.

Comparando os resultados da revisão bibliográfica com os estudos realizados na tese, verificam-se as seguintes similitudes e ineditismos:

i) Os exercícios e exemplos de soluções de EDOs e EDPs realizados na tese são, em parte, semelhantes aos apresentados no artigo de Tsoulus (2006). Isto ocorreu propositalmente, pois foram utilizados três exemplos deste trabalho como base de comparação para avaliar a eficiência e a eficácia computacional dos algoritmos desenvolvidos. Contudo, as semelhanças são apenas na forma funcional das equações diferenciais. Era necessário resolver exercícios idênticos

¹ARCH e GARCH são siglas que significam, respectivamente, (Autoregressive Conditional Heteroskedasticity) e (Generalized Autoregressive Conditional Heteroskedasticity).

para poder comparar efetivamente os resultados. Estas comparações foram fundamentais, pois possibilitaram perseguir melhores resultados e desenvolver algoritmos levemente mais eficientes e eficazes. Os demais aspectos da modelagem e da construção dos algoritmos são bem diferentes. Inclusive, por que se utiliza a técnica de PG tradicional, em forma de *árvore*, enquanto que o artigo citado trabalha com a técnica de *evolução gramatical*. Os algoritmos de DA também diferem.

ii) As similaridades com os artigos de (Imae, 2004; Figueira, 2011) estão diretamente relacionadas com o fato de ambos os trabalhos se destinarem a solucionar EDOs e EDPs usando a mesma técnica de PG, em forma *árvore*. No entanto, os exemplos desenvolvidos são bem diferentes e isso implica em modelagens e programações diferentes. Exemplos de soluções de EDOs ou EDPs diferentes exigem programações diferentes, com especificidades e detalhes de programação que tornam únicos cada um dos programas. Portanto, admite-se que similaridades existem, mas as diferenças entre os trabalhos são bem maiores.

iii) Os trabalhos de (Arnoud, 1974; Kloeden, 1992; Nicolau, 2000) serviram como referências teóricas para o cálculo estocástico de Itô utilizado na construção do algoritmo de PGDA elaborado para solucionar EDEs. De acordo com a pesquisa bibliográfica realizada, nenhum trabalho desta natureza foi encontrado na literatura consultada. Diante disso, acredita-se que este estudo da tese possa ser realmente inédito.

iv) Com relação às referências de (Black, 1973; Bollerslev, 1986, 1994; Campbell, 1997), estas foram usadas como base teórica, de finanças, para a elaboração do modelo de precificação de opções proposto pelo trabalho. Em virtude de utilizar o algoritmo de PGDA, citado no item anterior, e da pesquisa bibliográfica não ter encontrado nenhum modelo de precificação com estas características, acredita-se, também, que este estudo possa ser inédito.

v) Dentre os problemas tratados na tese, um grande dificuldade foi a construção de um algoritmo de PGDA para tentar solucionar a complexa equação de Schrödinger para o átomo de He. Embora não se tenha conseguido a solução exata, foi possível obter soluções aproximadas de baixo erro para a equação de onda. De acordo com a pesquisa bibliográfica realizada, também não foram encontrados na literatura consultada estudos de PGDA sobre este problema.

1.4

Organização do Trabalho

Além desta introdução, o trabalho apresenta-se organizado em outras seis seções. A segunda descreve os fundamentos teóricos e o referencial necessário para o desenvolvimento e entendimento das metodologias criadas e utilizadas na tese. A terceira seção apresenta os aspectos metodológicos que constituem a base conceitual dos algoritmos de PGDA desenvolvidos para solucionar os diversos problemas de equações diferenciais. Na quarta seção são apresentadas as metodologias específicas de cada estudo realizado e os resultados obtidos no trabalho. Na quinta seção são apresentadas as conclusões e sugestões de trabalhos futuros. A sexta e a sétima seção apresentam, respectivamente, as referências bibliográficas e o apêndice do trabalho.

2

Fundamentos Teóricos

2.1

Introdução

Nesta seção são apresentados os aspectos teóricos básicos necessários para o entendimento conceitual de alguns procedimentos técnicos adotados nas metodologias dos estudos e na análise dos resultados. Três subseções descrevem com detalhes e exemplos simples as técnicas de Programação Genética, Diferenciação Automática e Equações Diferenciais. As descrições baseiam-se fortemente nos trabalhos de (Boyce, 1977; Kloeden, 1992; Koza, 1992; Fink, 2007). O objetivo da seção é fornecer aos leitores menos familiarizados com os respectivos temas, os elementos técnicos indispensáveis para o entendimento do trabalho. Portanto, a leitura da seção é totalmente dispensável para aqueles que já dominam estes conhecimentos.

2.2

Programação Genética: Aspectos Gerais

A solução de problemas que envolvem elevado nível de complexidade computacional tem sido um desafio constante para pesquisadores de diversas áreas do conhecimento científico. Comumente se defrontam com problemas onde a solução ótima, por meio de métodos tradicionais, está limitada à instâncias de pequeno porte. Estas dificuldades passaram a ser dirimidas a partir do momento em que se passou a associar estas técnicas às da inteligência computacional. As quais se caracterizam por uma maior flexibilidade de programação e com menor custo computacional. A reunião destes métodos propiciou o surgimento dos chamados métodos inteligentemente flexíveis e dentre eles se destacam os métodos da computação evolucionária.

A computação evolucionária é a área da inteligência computacional que engloba um conjunto de métodos computacionais, onde a estrutura básica de funcionamento é inspirada na teoria evolucionista de Charles Darwin (Darwin,

2000). Segundo a teoria, a probabilidade de sobrevivência de um indivíduo na natureza depende fundamentalmente da sua capacidade de adaptação ao meio ambiente. Portanto, sobrevivem em maior número os mais aptos (fortes) e, conseqüentemente, as suas características genéticas são repassadas em maior proporção para as gerações seguintes. Assim, provavelmente, a nova geração será composta de indivíduos com material genético melhor que o da população anterior.

A programação genética (PG), cujo desenvolvimento é atribuído a John Koza (Koza, 1989), é uma meta-heurística estocástica de uso geral, sendo uma técnica da Computação Evolucionária. A PG destina-se à evolução de programas de computador em linguagens arbitrárias, no sentido da otimização de estruturas funcionais capazes de realizar operações lógicas, aritméticas, condicionais e de desvios, que normalmente mapeiam entradas em saídas.

De forma semelhante aos algoritmos genéticos, submete-se uma população de indivíduos, programas ou expressões matemáticas criadas aleatoriamente, ao processo artificial de seleção e evolução natural. Neste processo iterativo, figuram a cada geração: a seleção de indivíduos promissores para reprodução, a criação de seus descendentes por meio de operações genéticas como crossover e mutação, e finalmente a inserção destes novos indivíduos na população, marcando-se o início de uma nova geração. Muito embora não existam garantias de obtenção de soluções ótimas globais ao longo das gerações, esta dinâmica tende a produzir respostas satisfatórias e aceitáveis para a solução do problema de interesse. O fluxograma de um algoritmo tradicional de PG pode ser visto na Figura 1 a seguir.

Destacam-se como características típicas da programação genética as seguintes qualidades:

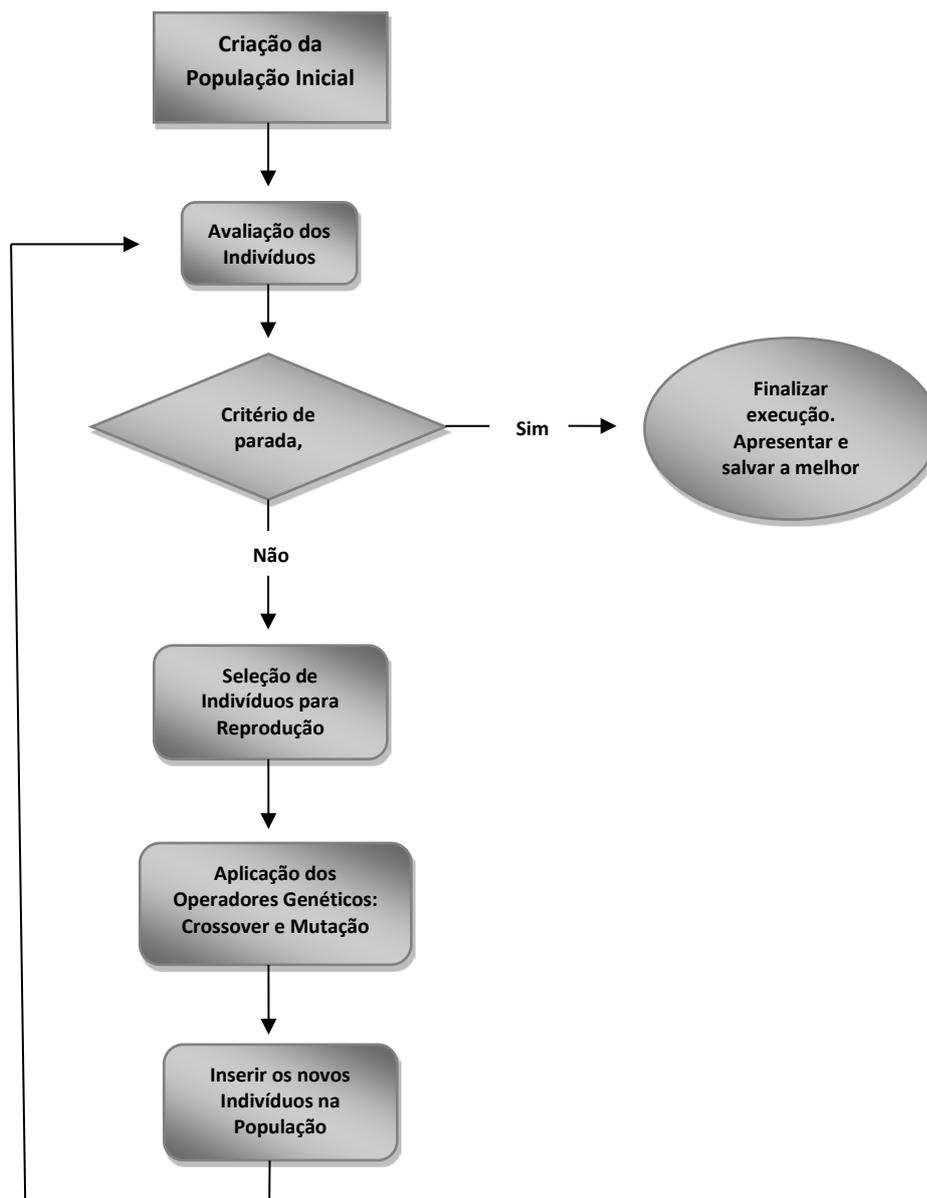


Figura 1: Fluxograma de um algoritmo tradicional de PG.

- Possui paralelismo natural: as demandas computacionais da PG podem ser particionadas em vários níveis, com o emprego ou não de modelos distribuídos bioinspirados, permitindo assim redução no tempo de execução e escalabilidade para problemas mais complexos.

- Produz soluções simbólicas: em virtude de o processo evolutivo atuar diretamente no nível simbólico das estruturas, as soluções obtidas pela PG são imediatamente legíveis e interpretáveis.

- É facilmente extensível e modificável: a PG é uma meta-heurística muito versátil e, por isso, permite hibridizações (desenvolvimento conjunto com outras técnicas), integração de outros métodos e as mais diversas representações e linguagens para os programas (como a representação formal por gramáticas).

2.2.1

Formas de Representação

Existem diversas formas capazes de representar programas de computador no âmbito da programação genética. As três principais estruturas conceituais de representação mais utilizadas são: por árvore, linear e evolução gramatical.

A despeito do extenso leque de opções, utiliza-se nesta tese a estrutura em forma de árvore para representar os resultados de PG. Nesta representação, o cromossoma de um indivíduo apresenta um desenho gráfico como exemplificado na Figura 2. Considere o exemplo simples de um indivíduo da população que tenha a forma da função, $f(x, y) = x^2 + y$. A sua representação em forma de árvore de sintaxe é dada por:

$$f(x,y) = (+ (* x x) y)$$

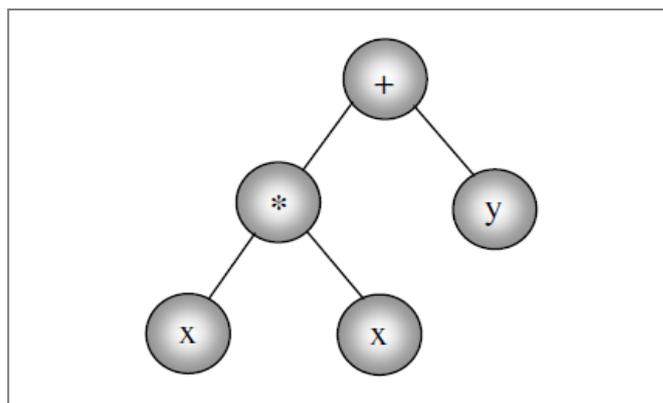


Figura 2: Exemplo de Representação em forma de Árvore

Uma árvore possui duas características básicas, a sua profundidade (*depth*) e os seus nós (*nodes*). No exemplo acima, a árvore apresenta profundidade de tamanho 3 e 5 nós.

Dentre as estruturas de representação, a em forma de árvore é a mais conhecida e utilizada em virtude da sua tradição na programação genética, talvez pela sua relativa simplicidade em expressar os resultados de um programa.

Todavia, esta forma de representação é conceitual e difere da representação computacional. Muito embora a árvore seja uma estrutura naturalmente hierárquica, ela pode ser implementada e operada linearmente sem que haja violação de suas propriedades. Ainda, um terceiro nível de classificação pode ser introduzido, a representação semântica. Neste, a representação conceitual é complementada com regras e restrições que governam como a estrutura em questão pode ser operada; pode-se citar como membros desta classificação, a representação formal por gramática sobre a estrutura árvore e sobre a estrutura linear.

2.2.2

Funções e Terminais

Os conjuntos de funções (F) e terminais (T) são elementos essenciais para o funcionamento da programação genética. Eles definem as ferramentas primitivas que estarão à disposição do processo evolucionário para a construção de estruturas mais complexas, isto é, definem a linguagem onde a evolução poderá explorar.

O conjunto F é responsável pelo fornecimento de funções que requerem argumentos, como: operadores matemáticos (+, -, ×, ÷); e funções matemáticas básicas (*sen, cos, exp, log, x^y, tan etc.*). O conjunto T provê os operandos, isto é, variáveis e atributos (*x, y, renda, idade etc.*); e constantes (*números reais, números complexos, número π , números aleatórios (rand) etc.*). O conjunto T fornece um valor para o sistema, enquanto que o conjunto F processa os valores no sistema. Juntos, os conjuntos de funções e terminais representam a profundidade e os nós da árvore de representação.

Os conjuntos de funções e terminais podem ser definidos livremente, e de forma recursiva são combinados. Portanto, todo indivíduo da população, em qualquer estágio do processo evolutivo, nada mais é do que uma combinação particular entre elementos de F e T.

Embora a escolha de F e T seja livre, esta etapa da modelagem é de crucial importância para a especificação de um bom algoritmo de PG. Caso os conjuntos

sejam escolhidos com um número muito pequeno de elementos, provavelmente, por falta de informações significantes e reduzido espaço de busca, o algoritmo de PG não será capaz de obter uma solução (ótimo local) de boa qualidade para o problema. Por outro lado, se os conjuntos são escolhidos com um número excessivo de elementos, o espaço de busca poderá ficar muito extenso, dificultando a determinação da solução ótima, caso exista, e provocando um grande e desnecessário esforço computacional. O mais aconselhável é estudar o problema e identificar um conjunto mínimo de funções e terminais que seja indispensável à solução do problema. Daí, iniciar a modelagem com este conjunto mínimo e, caso não consiga soluções satisfatórias, que estejam de acordo com a avaliação mínima desejável, ir adicionando novas funções e terminais ao processo de modelagem, até a obtenção de uma solução satisfatória.

2.2.3

Fechamento e Suficiência

As propriedades de fechamento (*closure*) e de suficiência (*sufficiency*) foram definidas por (Koza, 1992), com o objetivo de garantir soluções viáveis que satisfaçam a todas as restrições do problema.

A propriedade de fechamento garante que qualquer função do conjunto F deve ser capaz de operar com todos os valores recebidos como entrada. Isto garante que sejam geradas árvores sintaticamente corretas e viáveis. Um exemplo comum de uma função que não atende à propriedade de fechamento é a divisão por zero. O operador de divisão não aceita zero como entrada. A divisão por zero faz com que o programa apresente erro e pare de funcionar. Assim, ao invés de se trabalhar com o operador comum, define-se um novo operador, denominado de divisão protegida, que no caso em que aparece uma divisão por zero, o operador retorna o valor um. Para outras funções que apresentam este tipo de problema, (como por exemplo: raiz quadrada e logaritmo), a solução é dada de forma semelhante, definindo-se um novo operador protegido².

A propriedade de suficiência diz que, os elementos dos conjuntos F e T devem ser suficientes para descrever a solução esperada em pelo menos uma

² Utiliza-se a notação (%) para representar o operador de divisão protegida.

combinação específica. Entretanto, na prática, a propriedade de suficiência não é tão simples de ser atendida. Principalmente nos casos de problemas reais de grande porte e complexidade, onde pouco se conhece sobre os aspectos gerais e específicos da possível solução. Nesta situação, pouco se pode cogitar a respeito da formação dos conjuntos de funções e terminais e, diante disso, torna-se tentadora a possibilidade de trabalhar com grandes conjuntos de funções e terminais. Contudo, tal procedimento aumenta consideravelmente o espaço de busca e pode afetar fortemente o desempenho do algoritmo, tanto em termo de tempo de processamento como na qualidade das soluções.

2.2.4

População Inicial

O primeiro passo na inicialização de uma PG é definir a sua população inicial, ou seja, deve-se criar uma população de N indivíduos ou estruturas de programas para posterior evolução. Usualmente, a população inicial é criada de forma aleatória, elementos dos conjuntos de funções (F) e terminais (T) são selecionados aleatoriamente e combinados, dando origem aos indivíduos que compõem a população inicial.

Um importante parâmetro para a construção de algoritmos de PG com representação em árvore é exatamente o tamanho máximo da árvore. O tamanho de uma árvore é definido pelo seu número de nós (*nodes*) e sua profundidade (*depth*). Mas, geralmente, os algoritmos controlam a profundidade, que consiste do número de ramificações da árvore, partindo-se do nó raiz até o último e mais profundo nó.

Existem vários métodos para inicializar uma população em estrutura de árvore, os mais utilizados são: Full, Grow, ramped-half-and-half, random-branch e uniform, que são brevemente descritos a seguir.

Método Grow: Os nós são selecionados aleatoriamente dos conjuntos F e T , por este motivo o método produz árvores de formatos irregulares. Se uma ramificação contém um nó terminal, esta ramificação pára, mesmo que a profundidade máxima não tenha sido atingida. Na Figura 3 o nó d tem profundidade de tamanho 3, embora a árvore tenha profundidade de tamanho 4.

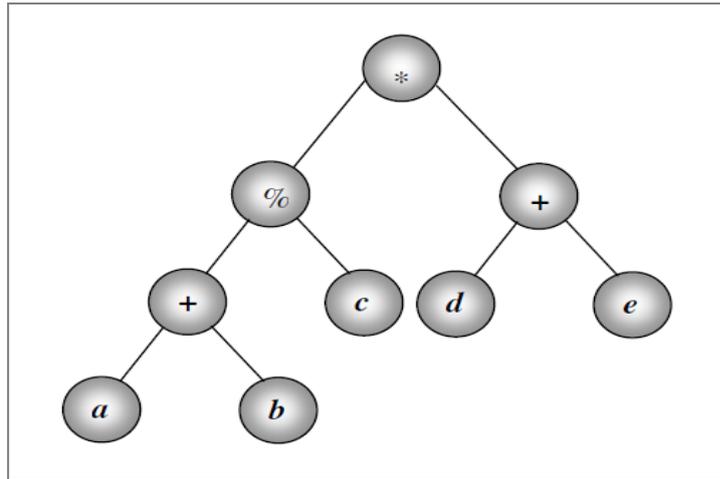


Figura 3: Árvore produzida com a Inicialização *Grow*

Método Full: Ao invés de escolher aleatoriamente os nós do conjunto de F e T , o método escolhe somente funções até que um nó de profundidade máxima seja selecionado, então ele passa a escolher somente terminais (Banzhaf, 1998). O resultado disso é que todas as árvores atingem a profundidade máxima. A Figura 4 mostra o exemplo uma árvore inicializada pelo método *Full*.

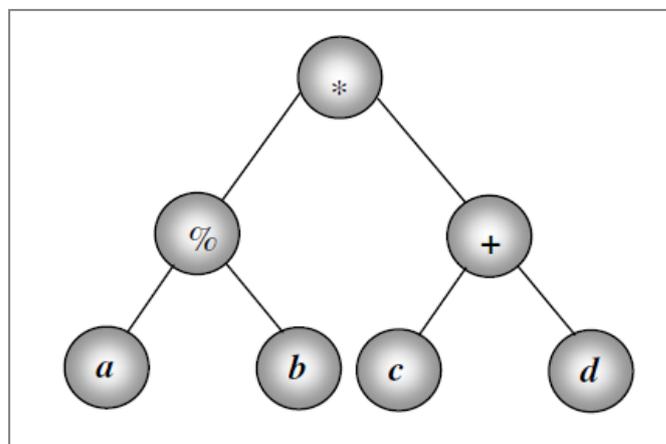


Figura 4: Árvore produzida com a Inicialização *Full*

Método Half-and-half: É uma combinação dos métodos *Grow* e *Full*, ou seja, utiliza o método *Full* em 50% das vezes e o método *Grow* nas outras 50%, com o objetivo gerar um número igual de árvores para cada profundidade (Koza, 1992). Supondo uma árvore com profundidade máxima de tamanho seis, a população é igualmente dividida em árvores com profundidades de tamanhos dois,

três, quatro, cinco e seis, ou seja, 20% terão profundidade dois, 20% terão profundidade três e assim por diante. Em cada grupo, metade das árvores são geradas pelo método *Full* e a outra metade pelo método *Grow*. As desvantagens deste método, segundo Luke (2000), são citadas a seguir:

- Se o conjunto de funções é maior que o conjunto de terminais, a tendência será de gerar a maior árvore possível;
- A escolha do parâmetro de profundidade máxima da árvore é realizada de forma proporcional e não aleatória;
- A faixa de profundidade é fixa (usualmente entre 2 e 6), independente do tamanho da árvore e dependendo do número de argumentos (aridade) de cada função, mesmo tendo a mesma profundidade, as árvores geradas podem ter tamanhos muito diferentes.

Método Random-Branch: Neste método, ao invés de se informar a profundidade máxima da árvore é informado seu tamanho máximo em nós. O método é menos restritivo, apesar de ter forte complexidade linear (Luke, 2001).

Método Uniform: O método foi desenvolvido com o objetivo de criar árvores uniformes, geradas a partir do conjunto de todas as árvores possíveis (Bohm, 1996). O algoritmo calcula várias vezes quantas árvores poderão ser geradas para cada tamanho desejado, por este motivo o método possui um elevado custo computacional.

2.2.5

Função de Aptidão e Avaliação dos Indivíduos

A função de aptidão é uma medida quantitativa que avalia o êxito e a qualidade com que os indivíduos realizam a tarefa alvo. A definição da função depende fundamentalmente do domínio da aplicação e o seu resultado determina o nível de aptidão do indivíduo.

Nos algoritmos clássicos de PG, como os problemas de regressão e outros onde cabe avaliação semelhante, geralmente a medida de aptidão é definida em função do erro de predição do indivíduo. Ou seja, a diferença entre os valores produzidos pela solução proposta e os valores reais observados para o problema.

Como exemplo, suponha em que os valores reais observados para o problema sejam dados pelo vetor Y de dimensão n , e os valores produzidos pela solução proposta sejam dados pelo vetor \hat{Y} , também de dimensão n . Ou seja,

$$Y = (y_1, y_2, \dots, y_n) \quad e \quad \hat{Y} = (\hat{y}_1, \hat{y}_2, \dots, \hat{y}_n)$$

Então, o erro de predição de um indivíduo é dado pelo vetor diferença,

$$E = (e_1, e_2, \dots, e_n) = Y - \hat{Y} = (y_1 - \hat{y}_1, y_2 - \hat{y}_2, \dots, y_n - \hat{y}_n).$$

Como existem n erros possíveis e estes erros podem assumir valores positivos e negativos, as medidas de avaliação são geralmente definidas de forma a ressaltar todos os n erros e em termos positivos. Pois, o objetivo da função de avaliação é medir a distância entre os valores reais observados e os propostos pela solução.

Com base nestas informações são construídas e usualmente utilizadas como função de aptidão (*fitness*) as seguintes medidas:

1. Soma de Quadrados dos Erros (SQE):

$$SQE = \sum_{i=1}^n e_i^2 = \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

2. Raiz do Erro Quadrático Médio (REQM):

$$SQE = \sqrt{\frac{1}{n} \sum_{i=1}^n e_i^2} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$$

3. Erro Absoluto Médio (EAM ou MAE em inglês):]

$$EAM = \frac{1}{n} \sum_{i=1}^n |e_i| = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

4. Erro Percentual Absoluto Médio (EPAM ou MAPE em inglês):

$$EPAM = \frac{1}{n} \sum_{i=1}^n \left| \frac{e_i}{y_i} \right| = \frac{1}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right|$$

Logo, quanto menor for o valor computado por cada uma das medidas acima, melhor será a avaliação do indivíduo. Portanto, o que se procura e se deseja encontrar é o indivíduo ou a solução que minimiza a função de aptidão ou a função objetivo do problema.

No entanto, muitas funções de aptidão, inclusive as apresentadas acima, possuem a desvantagem de não ponderar a aptidão do indivíduo em função de sua complexidade e tamanho. O que permitiria a obtenção de soluções mais compactas, legíveis e com maior poder de generalização.

O controle da complexidade e do tamanho das soluções é um dos grandes problemas da PG e tem demandado expressivo esforço de pesquisa. Por basear-se em estruturas de tamanho variável e ilimitado, existe o risco iminente de crescimento inútil e desordenado do tamanho médio destas estruturas no curso da evolução; este fenômeno é conhecido como *bloat*.

A despeito do esforço teórico direcionado ao entendimento do *bloat*, há uma mobilização concomitante em torno do controle prático da complexidade das soluções. As abordagens propostas podem ser categorizadas em três tipos:

- (i) controle rígido do tamanho e profundidade das estruturas;
- (ii) operadores genéticos especificamente desenvolvidos; e
- (iii) pressão de seleção contrária à complexidade.

No primeiro método, o controle rígido de tamanho e profundidade, limites máximos são impostos sobre todas as operações que criam ou modificam as estruturas dos programas, isto é, usualmente inicialização, cruzamento e mutação. Estes limites são facilmente implementados na inicialização, porém, são inadequados quando impostos sobre a mutação e o cruzamento. Nestes operadores a existência de limites esbarra na natureza estocástica da programação genética, cujas regiões de atuação dos operadores são escolhidas aleatoriamente, culminando no fracasso (desperdício) de diversas tentativas que infringiriam os limites. Ademais, o controle rígido transfere ao modelador a responsabilidade de garantir rigorosamente que o limite máximo estabelecido é suficiente para descrever a solução do problema.

Aplicar operadores genéticos especializados é outra variação de controle da complexidade. Estes operadores atuam no âmbito do cruzamento e mutação e asseguram que um limite pré-estabelecido não seja transgredido ou, em alguns casos, que não seja transgredido além de certa margem. Incluem, por exemplo, operadores de cruzamento que restringem a recombinação apenas às partes de tamanhos compatíveis, ou operadores de mutação que encolhem as soluções candidatas ou proíbem alterações que acresçam complexidade. Muito embora o método de especialização dos operadores genéticos ofereça uma alternativa

aprimorada ao controle rígido de complexidade, a operação de cruzamento ainda é restrita e induz um viés na dinâmica de recombinação do material genético. Os tamanhos das estruturas a serem recombinadas devem ser semelhantes.

A terceira abordagem adota uma postura mais flexível e natural, e delega o controle da complexidade exclusivamente à função de seleção. Agora, ser um indivíduo apto significa, dentre outros quesitos, ser parcimonioso; a solução ideal é aquela que melhor conjuga qualidade com complexidade e tamanho. Assim, os indivíduos muito complexos de baixa e média avaliação tendem a desaparecer sob pressão seletiva. O controle da complexidade por pressão parcimoniosa dispensa quaisquer modificações no que diz respeito aos operadores genéticos, logo, tais operações realizam-se livres de restrições.

Várias teorias foram elaboradas na tentativa de prover explicações para o problema de *bloat*. No entanto, nenhuma das teorias propostas foi capaz de tratar o problema de maneira segura e completa. Para maiores informações sobre o problema de *bloat*, sugere-se os trabalhos de (Langdon, 1997; Luke, 2006; Poli, 2008).

2.2.6

Métodos de Seleção para Reprodução

O processo de seleção de indivíduos para reprodução é o instrumento pelo qual os algoritmos evolucionários conduzem a busca para as regiões mais promissoras do espaço e que potencializa o processo de otimização.

A seleção é responsável por escolher os indivíduos da população que participarão do processo de reprodução genética e que produzirão descendentes para as gerações seguintes. Portanto, é natural esperar que os métodos de seleção favoreçam os indivíduos mais aptos, pois a transmissão de bom material genético contribui para o desenvolvimento de melhores soluções.

A diferença entre a probabilidade de seleção dos melhores indivíduos e a probabilidade de seleção dos piores denomina-se *pressão de seleção*. Quando a pressão de seleção é baixa, os indivíduos com menor aptidão têm mais chance de disseminar seus genes; por outro lado, quando a pressão é elevada, somente os indivíduos muito bem avaliados têm a oportunidade de serem selecionados. Embora possa parecer que, a elevada pressão de seleção é sempre vantajosa, na

realidade ela poderá levar o processo evolucionário à indesejável convergência prematura. Isto provavelmente ocorrerá devido à degradação da diversidade da população, uma vez que um pequeno e seletivo grupo de indivíduos dominará o processo.

Existem diferentes métodos de seleção e a escolha de qual método será utilizado na PG é uma tarefa de difícil decisão. Dada a inexistência de um método ótimo, o que se faz na prática é testar diferentes métodos e escolher aquele que apresenta o melhor desempenho para a PG do problema em questão. Alguns dos métodos mais utilizados são descritos a seguir.

- Seleção Proporcional (*Proportional Selection*): este método de seleção é aplicado em algoritmos evolutivos e especifica a probabilidade de que cada indivíduo seja selecionado para a próxima geração. Para o indivíduo i , a probabilidade de ser selecionado para a próxima geração é dada pela equação,

$$p_i = \frac{f_i}{\sum_{k=1}^N f_k}.$$

Onde f_i é o valor da aptidão do i -ésimo indivíduo e N o número de indivíduos. O indivíduo com maior aptidão tem maior probabilidade de ser selecionado e, em geral, é incluído na população seguinte. Este procedimento é conhecido como *elitismo*, e tem por objetivo privilegiar a melhor solução, de forma que este indivíduo propague suas características para a população seguinte.

- Truncamento (*truncation selection*): este é o segundo método mais popular, utilizado para seleção e provém dos algoritmos de Estratégias Evolucionárias (Schwefel, 1995). Com base em um valor de limiar (*Threshold*) Th , um número entre 0 e 1, a seleção é feita aleatoriamente entre os $Th\%$ melhores indivíduos. Por exemplo, se $Th = 0,6$, isto significa que a seleção é feita entre os 60% melhores indivíduos e os demais são descartados.

- Ranqueamento (*Ranking Selection*): no método de seleção por ranking os indivíduos são ordenados de forma crescente de acordo com seu valor de aptidão. Assim, a cada indivíduo, é atribuído um número inteiro de acordo com sua posição no ranking, quanto melhor o ranking do indivíduo, melhor sua aptidão em relação aos demais indivíduos da população e, portanto, melhores são suas chances de ser sorteado. Pode ser utilizado o ranking linear ou exponencial. Para o ranking linear, a probabilidade é uma função linear dada pela equação:

$$p_i = \frac{1}{N} \left[R_1 + (R_N - R_1) \frac{(i-1)}{(N-1)} \right]; \quad i = 1, \dots, N$$

Onde, $p_1 = \frac{R_1}{N}$ é a probabilidade do pior indivíduo ser selecionado, e $p_N = \frac{R_N}{N}$ é a probabilidade do melhor indivíduo ser selecionado.

Para o *ranking* exponencial, a probabilidade exponencialmente ponderada é dada por:

$$p_i = \frac{(C-1)C^{N-i}}{C^N - 1}; \quad 0 < C < 1 \quad e \quad i \in \{1, \dots, N\}$$

- Torneio: a seleção por torneio, dada a sua facilidade de aplicação e eficiência, é comumente utilizada em programação genética. O método consiste em selecionar aleatoriamente (n) indivíduos da população e comparar as suas aptidões. O indivíduo com melhor aptidão vence o torneio. Quanto maior for número de selecionados (n) maior será a competição e, conseqüentemente, a pressão seletiva.

2.2.7

Operadores Genéticos

Após os indivíduos terem sido selecionados por um dos métodos de seleção, os operadores genéticos são aplicados a estes indivíduos para então gerar a nova população. Segundo (Koza 1992), diversos operadores genéticos foram criados, mas os mais considerados e usualmente utilizados são os operados de cruzamento (crossover) e o de mutação, os quais são apresentados a seguir:

Operador de Crossover: este operador executa o cruzamento e recombinação de partes de estruturas previamente selecionadas pelo processo de seleção, com o intuito de derivar estruturas ainda melhores. O crossover é um operador primário na programação genética e, geralmente, é aplicado com taxa variável de probabilidade.

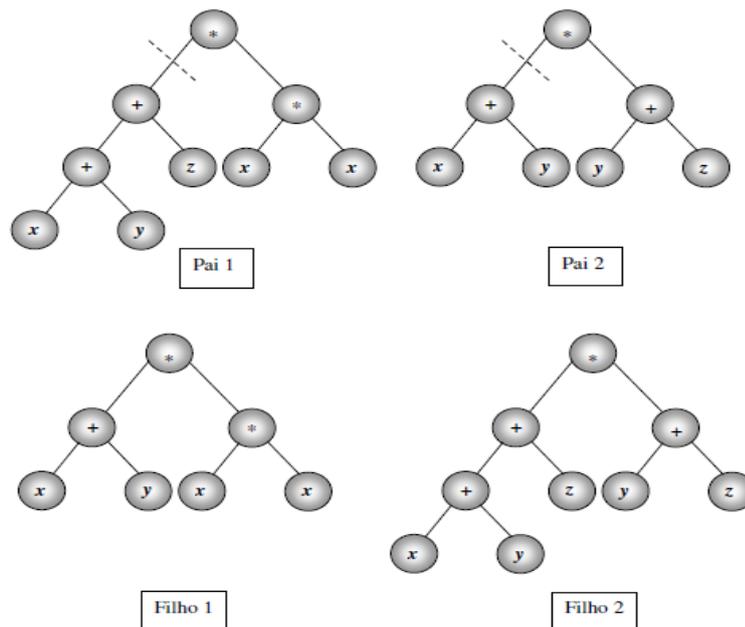


Figura 5: Exemplo de Crossover *one-point*

Uma das formas mais simples de cruzamento é o de um ponto de corte (*one-point crossover*). Nesta forma, dois indivíduos, denominados de genitores ou pais, são selecionados da população com base em suas boas aptidões. Então, um ponto de corte é selecionado aleatoriamente e as sub-árvores são combinadas gerando dois novos descendentes ou filhos. Um exemplo deste método de cruzamento é apresentado na Figura 5.

- **Mutação:** este operador efetua alterações em um indivíduo somente, ou seja, nenhum novo indivíduo é gerado. Nesta operação, seleciona-se aleatoriamente o indivíduo que sofrerá a mutação e o ponto de corte. Neste ponto substitui-se a sub-árvore existente por uma nova gerada aleatoriamente. Esta nova sub-árvore está sujeita às mesmas limitações de profundidade e tamanho que as árvores geradas na população inicial. Em seguida, o indivíduo transformado é reinserido na população.

A função deste operador é inserir diversidade na população, fazendo com que os novos indivíduos explorem novas áreas do espaço de busca. Porém ao se inserir muita diversidade numa população, a mesma poderá não convergir para um ótimo global, ou mesmo poderá não convergir, oscilando indefinidamente, por este motivo a taxa de mutação deve ser baixa. Um exemplo do funcionamento do operador de mutação é apresentado na Figura 6, onde a árvore da esquerda mostra

o indivíduo em sua forma original e a da direita o indivíduo após a mutação da sub-árvore com cinco nós. Geralmente, a operação de mutação é realizada com uma taxa variável de probabilidade.

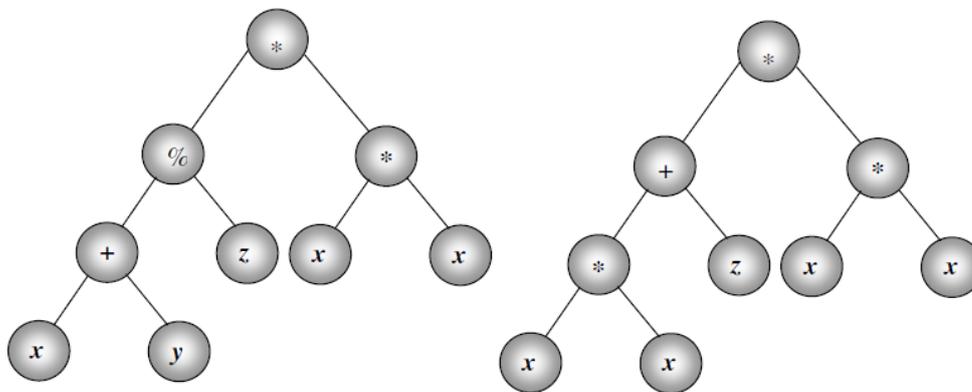


Figura 6: Exemplo de Mutação Padrão

2.2.8

Métodos de Reprodução

Os métodos de reprodução dizem respeito à forma sob a qual os descendentes recém-criados são introduzidos no processo evolucionário. Existem dois métodos básicos: *o geracional* e *o steady-state*.

O método geracional foi o primeiro a surgir na literatura dos algoritmos de computação evolucionária; o seu funcionamento consiste nos seguintes passos: a cada geração os descendentes recém-criados são alocados em uma população temporária; quando o número de indivíduos criados se iguala ao da população principal a população temporária substitui a principal e assim todos os indivíduos da geração passada são descartados. Uma variação muitas vezes útil é preservar os melhores indivíduos da geração passada (*elitismo*).

Uma desvantagem do método geracional, que pode ser decisiva em certas aplicações, é a necessidade de se alocar uma segunda população durante o processo evolutivo. Por outro lado, o método de reprodução *steady-state* insere os novos descendentes imediatamente na população atual, logo, eles passam a coexistir com seus antecedentes. Quando um novo descendente é inserido, um indivíduo da geração atual deve ser eliminado para que o tamanho da população

permaneça constante. Habitualmente, escolhe-se para ser substituído um indivíduo com baixa aptidão (*ou o menos apto*), dessa forma, a evolução descarta soluções pouco promissoras e obtém-se, naturalmente, o efeito de manutenção dos melhores indivíduos, análogo ao *elitismo* da reprodução geracional.

2.2.9

Parâmetros e Critério de Parada

A definição dos parâmetros a serem utilizados em um algoritmo de PG pode ter grande influência nos resultados obtidos, e caso não sejam bem definidos podem provocar excessivo tempo de processamento e grande custo computacional. Os parâmetros a serem definidos são:

- *Tamanho da População*: este parâmetro deve informar o número de indivíduos que compõem a população. A sua escolha deve ser feita de forma criteriosa, pois dela depende a qualidade dos resultados. Uma população muito pequena restringe o espaço de busca, e se for muito grande poderá provocar um esforço computacional excessivo e sem grandes ganhos na qualidade dos resultados.

- *Taxa de cruzamento (crossover)*: define a taxa de aplicação do operador de cruzamento em cada geração. Se esta taxa for alta, pode haver uma convergência prematura do algoritmo. Por outro lado, se for muito pequena, o algoritmo poderá levar muito tempo para obter uma boa solução para o problema.

- *Taxa de mutação*: define o percentual de mutações que deverão ocorrer em cada geração, se for muito alta, poderá tornar a busca completamente aleatória.

- *Número de gerações*: define o número de vezes que o processo evolutivo será executado.

Com relação ao critério de parada, o ideal seria que a execução da PG terminasse somente quando uma solução ótima fosse encontrada. Infelizmente, dependendo da complexidade do problema e dos recursos computacionais disponíveis, isto nem sempre é possível. Nesses casos, a alternativa é obter soluções factíveis, “suficientemente boas”, e que sirvam ao propósito e objetivo do trabalho em questão. A implementação desta estratégia de modelagem exige a definição de critérios de parada que atendam às limitações acima citadas. São comumente utilizados os seguintes:

- Aptidão dentro da faixa aceitável: o processo é interrompido quando uma solução aproximada é obtida de acordo com a avaliação mínima desejada.
- Número máximo de gerações ou avaliações: a execução é interrompida ao se atingir um número pré-estabelecido de gerações ou avaliações.
- Tempo limite de execução: a execução é interrompida quando o tempo pré-estabelecido de execução é atingido.
- Estagnação do processo: a execução termina quando é detectada a estagnação do processo evolutivo, isto é, quando o algoritmo da PG converge para uma solução qualquer.

2.3

Diferenciação Automática

O cálculo diferencial é um dos mais importantes conhecimentos científicos desenvolvidos pela Matemática e amplamente aplicado em diversas áreas. Muitos problemas da Engenharia, Física, Economia, Biologia e outras ciências dependem deste conhecimento para suas resoluções. Neste trabalho, o cálculo diferencial é utilizado para, junto com a programação genética, solucionar problemas oriundos da sua própria formulação e extensão, que são as equações diferenciais ordinárias, parciais e estocásticas.

Na construção de algoritmos de PG que cumpram com este objetivo, o método de diferenciação de funções ocupa posição de destaque, pois é dele a tarefa de verificar matematicamente as soluções propostas pela PG. Portanto, é parte principal da função objetivo do algoritmo (*fitness*) e componente essencial para o seu funcionamento. Daí, surge a necessidade metodológica de se ter métodos de diferenciação que atendam às exigências do algoritmo e que possam ser implementados computacionalmente.

Como solução para o problema foram criados os métodos de diferenciação automática (DA) que, dada a sua grande precisão e rapidez de execução, é a alternativa mais indicada. Segundo Griewank (2008) a DA, também chamada de diferenciação algorítmica ou diferenciação computacional, consiste de um conjunto de técnicas computacionais, construídas com base na teoria do cálculo diferencial, que avaliam numericamente a derivada de uma função especificada por um programa de computador.

Em muitos problemas de computação científica, utilizam-se funções matemáticas definidas através de programas que constroem as funções a partir de uma série de operações matemáticas elementares. O método de DA explora o fato de que qualquer programa deste tipo executa uma sequência de operações aritméticas elementares (*adição, subtração, multiplicação, divisão etc.*) e de funções elementares (*exp, log, sen, cos etc.*). Deste ponto de vista, um programa representa uma composição de várias funções primitivas, as quais possuem derivadas conhecidas. Com isso, computar a derivada de uma função composta torna-se um exercício de aplicar a regra da cadeia criteriosamente. Além disso, derivadas de ordem arbitrária podem ser calculadas automaticamente e com elevada precisão.

Dois outros métodos também servem ao propósito da diferenciação de funções: o método de diferenças finitas (ou, diferenciação numérica) e a diferenciação simbólica. Contudo, não são em geral tão eficazes quanto os métodos de DA.

2.3.1

A Regra da Cadeia e os modos *Forward* e *Reverse*

O método de diferenciação automática baseia-se fortemente na decomposição dos diferenciais definidos pelo teorema da regra da cadeia. Para relembrar este conceito, considere a seguinte função composta:

$$f(x) = g(h(x))$$

Aplicando a regra da cadeia, tem-se:

$$\frac{df(x)}{dx} = \frac{dg(h(x))}{dh(x)} \frac{dh(x)}{dx}$$

A diferenciação automática utiliza-se de duas formas distintas para aplicar a regra da cadeia, os modos *forward* e *reverse*. O *forward* especifica a regra da cadeia da direita para a esquerda (ou seja, primeiro calcula dh/dx e, em seguida, dg/dh), enquanto o modo *reverse* a especifica da esquerda para a direita.

Para exemplificar o cálculo da derivada pelo modo *forward*, considere a seguinte função:

$$f(x_1, x_2) = x_1x_2 + \text{sen}(x_1)$$

A função é interpretada por um computador como uma sequência de operações elementares das variáveis w_i e, para realizar as derivadas parciais de $f(x_1, x_2)$, o modo *forward* segue os passos apresentados no Quadro 1.

Observe que, primeiramente, é necessário definir $w_1 = x_1$ e $w_2 = x_2$ com $w'_1 = 1$ e $w'_2 = 0$ para que o modo possa identificar que a derivada é com relação a x_1 , e $w'_1 = 0$ e $w'_2 = 1$ para identificar que a derivada é com relação a x_2 . Note que, embora o quadro apresente as formas simbólicas das derivadas, no programa o que fica armazenado é sempre o valor avaliado (numérico). Veja que nas linhas 3 e 4 são realizadas, respectivamente, a derivada do produto de $w_3 = w_1 w_2$ e a regra da cadeia para a derivada de $w_4 = \text{sen}(w_1)$. Na quinta linha o modo *forward* chega aos resultados finais das derivadas parciais. O modo *reverse* realiza operações semelhantes às apresentadas no Quadro 1, mas com a diferença de que o processo de diferenciação é realizado da esquerda para a direita (*backward propagation*).

Quadro 1: DA - Exemplo do Modo *Forward*

<i>Argumentos Originais</i>	<i>Argumentos de $\left(\frac{\partial f}{\partial x_1}\right)$</i>	<i>Argumentos de $\left(\frac{\partial f}{\partial x_2}\right)$</i>
$w_1 = x_1$	$w'_1 = 1$	$w'_1 = 0$
$w_2 = x_2$	$w'_2 = 0$	$w'_2 = 1$
$w_3 = w_1 w_2$	$w'_3 = w'_1 w_2 + w_1 w'_2 = 1x_2 + x_1 0 = x_2$	$w'_3 = 0x_2 + x_1 1 = x_1$
$w_4 = \text{sen}(w_1)$	$w'_4 = \cos(w_1)w'_1 = \cos(x_1)1 = \cos(x_1)$	$w'_4 = \cos(x_1)0 = 0$
$w_5 = w_3 + w_4$	$w'_5 = w'_3 + w'_4 = x_2 + \cos(x_1) = \frac{\partial f}{\partial x_1}$	$w'_5 = w'_3 + w'_4 = x_1 = \frac{\partial f}{\partial x_2}$

Com os resultados do Quadro 1 torna-se simples e direto o cálculo do gradiente de $f(x_1, x_2)$. Ou seja,

$$\frac{\partial f(\vec{x})}{\partial \vec{x}} = \left[\frac{\partial f(x_1, x_2)}{\partial x_1}, \frac{\partial f(x_1, x_2)}{\partial x_2} \right] = [x_2 + \cos(x_1), x_1].$$

Para o exemplo acima, com funções do tipo $f: \mathbb{R}^n \rightarrow \mathbb{R}$, onde $n > 1$, o modo *reverse* apresenta desempenho computacional superior ao do modo *forward*, pois executa um número menor de derivadas e de etapas para realizar o programa de diferenciação. No entanto, embora não seja o nosso caso, necessita armazenar algumas variáveis de trabalho (w_i), o que pode ocasionar um problema

significativo de memória. No caso oposto, do tipo $f: \mathbb{R} \rightarrow \mathbb{R}^m$, onde $m > 1$, o modo *forward* apresenta desempenho computacional superior ao do modo *reverse*.

Os modos *forward* e *reverse* também são utilizados para solucionar problemas mais gerais e complexos de DA aplicados a funcionais que envolvem várias variáveis de entrada e saída, da forma $f: \mathbb{R}^n \rightarrow \mathbb{R}^m$, onde as soluções são obtidas através de cálculos de Jacobianos e, para ordem mais elevada de derivadas, matrizes Hessianas. Para maiores detalhes sobre estas metodologias, sugere-se consultar (Rall, 1981; Griewank, 2008; Neidinger, 2010).

2.3.2

Softwares e Linguagens de Diferenciação Automática

Diversos são os softwares desenvolvidos para a aplicação da diferenciação automática. As linguagens de programação mais utilizadas são: C/C++, Matlab, Python, Fortran, Java, Haskell, Clojure e Octave. O método de diferenciação automática é geralmente implementado no modo *forward* e através de duas estratégias ou abordagens de programação: a (SCT-Source Code Transformation) e a (OO-Operator Overloading).

Neste trabalho foram utilizados os programas myAD e myA2D, de autoria de Martin Fink (Fink, 2007), elaborados em linguagem Matlab e com modo *forward*. Os programas são específicos e calculam, respectivamente, derivadas de primeira e segunda ordem. A seguir são apresentados dois exemplos simples que objetivam mostrar como esses programas realizam a DA.

Exemplo 1: Aplicação de DA em Code Matlab, utilizando myAD.

Suponha que se tenha o interesse em calcular as derivadas de primeira ordem do funcional $f: \mathbb{R}^2 \rightarrow \mathbb{R}^2$; tal que

$$f(x_1, x_2) = \begin{bmatrix} f_1(x_1, x_2) \\ f_2(x_1, x_2) \end{bmatrix} = \begin{bmatrix} e^{x_1 x_2} \\ -\cos(x_1 - x_2) \end{bmatrix}$$

e avaliar no ponto $(x_1 = 0, x_2 = 1)$. A Figura 7 mostra o desenvolvimento deste cálculo com DA. Utilizando as técnicas tradicionais do cálculo diferencial pode-se mostrar que a matriz jacobiana é idêntica à apresentada na Figura 7. Ou seja,

$$f'(x_1, x_2)|_{(0,1)} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{bmatrix}_{(0,1)} = \begin{bmatrix} x_2 e^{x_1 x_2} & x_1 e^{x_1 x_2} \\ \text{sen}(x_1 - x_2) & -\text{sen}(x_1 - x_2) \end{bmatrix}_{(0,1)} =$$

$$= \begin{bmatrix} 1e^0 & 0e^0 \\ \text{sen}(-1) & -\text{sen}(-1) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -0,845 & 0,845 \end{bmatrix}.$$

#

The screenshot shows the MATLAB Editor window with the following code:

```

1
2 % Exemplo de Aplicação de AD usando o Code: myAD (Fink, 2007)
3
4 % f(x1, x2)=[exp(x1*x2); -cos(x1-x2)]; Função de Interesse.
5
6 % f'(x1, x2)=[df1/dx1 df1/dx2; df2/dx1 df2/dx2]; Matriz Jacobiana.
7
8 % (x1=0; x2=1). Ponto de avaliação das derivadas.
9
10 function Exemplo_myAD % Code que calcula as primeiras derivadas de f.
11
12     X=[0; 1]; % Ponto onde são avaliadas as derivadas.
13
14     DX=myAD(X); % Converte X em uma variável AD.
15
16     ADX=fun(DX); % Avalia a função na variável convertida para AD.
17
18     dfdx=getderivs(ADX); % Cálculo da Matriz Jacobiana.
19
20
21 function f=fun(X) % Declaração da função que será diferenciada.
22
23     f=[exp(X(1)*X(2)); -cos(X(1)-X(2))]; % Função de Interesse.
24
25     %%%

```

The Command Window shows the following output:

```

ADX:
Values =
    1.0000
   -0.5403

Derivatives =
    1.0000    0
   -0.8415    0.8415

dfdx =
    1.0000    0
   -0.8415    0.8415

fx >> |

```

Figura 7: Aplicação de DA em Code Matlab, utilizando myAD

Exemplo 2: Aplicação de DA em Code Matlab, utilizando myA2D.

Suponha que, agora, o interesse seja de calcular as derivadas de segunda ordem do funcional $f: \mathbb{R}^2 \rightarrow \mathbb{R}$, tal que

$$f(x_1, x_2) = \frac{1}{\sqrt{x_1^2 + x_2^2}},$$

e avalia-las no ponto $(x_1 = 1, x_2 = -1)$. Calculando-se as derivadas parciais chega-se à seguinte matriz hessiana H:

$$\begin{aligned}
 H(x_1, x_2)|_{(1,-1)} &= \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_2 \partial x_1} \\ \frac{\partial^2 f}{\partial x_1 \partial x_2} & \frac{\partial^2 f}{\partial x_2^2} \end{bmatrix}_{(1,-1)} = \\
 &= \begin{bmatrix} 3x_1^2(x_1^2 + x_2^2)^{-5/2} - (x_1^2 + x_2^2)^{-3/2} & 3x_1x_2(x_1^2 + x_2^2)^{-5/2} \\ 3x_2x_1(x_1^2 + x_2^2)^{-5/2} & 3x_2^2(x_1^2 + x_2^2)^{-5/2} - (x_1^2 + x_2^2)^{-3/2} \end{bmatrix}_{(1,-1)} = \\
 &= \begin{bmatrix} 3 \cdot 2^{-5/2} - 2^{-3/2} & -3 \cdot 2^{-5/2} \\ -3 \cdot 2^{-5/2} & 3 \cdot 2^{-5/2} - 2^{-3/2} \end{bmatrix} = \begin{bmatrix} 0,1768 & -0,5303 \\ -0,5303 & 0,1768 \end{bmatrix}.
 \end{aligned}$$

Portanto,

$$H(x_1, x_2)|_{(1,-1)} = \begin{bmatrix} 0,1768 & -0,5303 \\ -0,5303 & 0,1768 \end{bmatrix}$$

#

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
% Exemplo de Aplicação de AD usando o Code: myA2D (Fink, 2007)
% f(x1, x2)=1/((x1^2 + x2^2)^(1/2)); Função de Interesse.
% (x1=1; x2=-1). Ponto de avaliação das derivadas.
function Exemplo_myA2D % Code que calcula as derivadas de 2ª ordem de f.
    X=[1; -1]; % Ponto onde são avaliadas as derivadas.
    D2X=myA2D(X); % Converte X em uma variável AD.
    AD2X=fun(D2X); % Avalia a função na variável convertida para AD.
    d2fdx=getsecderiv(AD2X) % Cálculo da Matriz Hessiana.
function f=fun(X) % Declaração da função que será diferenciada.
    f=1./((X(1)^2 + X(2)^2)^(1/2)); %Função de Interesse.
    %%%
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
d2fdx(:, :, 1) =
    0.1768    -0.5303
d2fdx(:, :, 2) =
   -0.5303     0.1768
fx >>
    
```

Figura 8: Aplicação de DA em Code Matlab, utilizando myA2D

A Figura 8 apresenta o desenvolvimento deste cálculo em DA, onde é utilizado o programa myA2D implementado em code matlab. Observe que os resultados chegam à mesma matriz hessiana (H) obtida acima. Na Figura 8, a matriz H é dada por dois vetores linha e é composta da seguinte forma:

$$H(x_1, x_2)|_{(1,-1)} = \begin{bmatrix} \mathbf{d^2f dx}(:, :, 1) \\ \mathbf{d^2f dx}(:, :, 2) \end{bmatrix} = \begin{bmatrix} 0,1768 & -0,5303 \\ -0,5303 & 0,1768 \end{bmatrix}$$

Portanto, verifica-se nos Exemplos 1 e 2 que a diferenciação automática é um método muito eficiente, pois produz derivadas pontuais exatas para o problema de interesse.

2.4

Equações Diferenciais

2.4.1

Introdução

O objetivo desta seção é o de apresentar de forma sucinta os aspectos básicos da teoria de equações diferenciais ordinárias, parciais e estocásticas que são necessários para o entendimento dos estudos apresentados na tese. A seção está dividida em duas subseções, onde a primeira trata das equações determinísticas, EDOs e EDPs e a segunda das equações estocásticas, EDEs. Antes, porém, apresenta-se um breve resumo das áreas e projetos onde as equações diferenciais são aplicadas.

Embora os resultados finais de estudos e projetos não evidenciem a sua utilização, é de amplo conhecimento técnico que as equações diferenciais são indispensáveis e usualmente são utilizadas para modelar problemas concretos e de grande importância em diversas áreas do conhecimento técnico e científico. Para ilustrar como isso ocorre na prática, enumera-se, a seguir, algumas atividades, projetos e pesquisas que, em alguma etapa ou fase de suas elaborações, utilizam-se destas técnicas.

- i) *Engenharia Elétrica*: na construção de circuitos elétricos; modelagem da dilatação de cabos de redes; aplicações do filtro digital de “*Butterworth*”, na área de processamento de sinais; etc.;

- ii) *Engenharia Civil*: na modelagem dos problemas de vigas em geral; deflexão e vibração;
- iii) *Engenharia Química*: no balanço mássico de um reator químico;
- iv) *Engenharia Nuclear, Química, Arqueologia e Geologia*: na determinação da taxa de desintegração ou decaimento de um elemento radioativo, tipo carbono-14;
- v) *Economia*: em modelos econométricos; microeconomia; macroeconomia; teoria de crescimento econômico; economia dinâmica e as equações de “Bellman”; e modelos de equilíbrio geral;
- vi) *Finanças*: na modelagem de preço e retornos de ativos financeiros; modelos de opções e derivativos em geral; e em todo cálculo estocástico aplicado a finanças;
- vii) *Probabilidade e Estatística*: na teoria de probabilidade e processos estocásticos; nos métodos de estimação de máxima verossimilhança e mínimos quadrados; na teoria assintótica;
- viii) *Engenharia Mecânica*: no planejamento de projetos de suspensões automotivas, definindo o sistema mecânico de amortecimento; e projetos industriais em geral;
- ix) *Engenharia Aeroespacial*: nos projetos de planejamento e construção de aeronaves, turbinas, motores e navegação;
- x) *Física*: na mecânica quântica; mecânica relativista; mecânica clássica; ótica, estática, dinâmica etc.

2.4.2

Equações Diferenciais Ordinárias e Parciais

Segundo alguns relatos históricos mencionados em (Ince, 1956; Boyce, 1977; Eves, 1990; Boyer, 1996; Courant, 2000), os estudos das equações diferenciais iniciam-se no século XVII quando Newton, Leibniz e Bernoulli resolveram algumas equações simples surgidas em problemas de geometria e mecânica. Estas primeiras descobertas pareciam sugerir que as soluções de todas as equações diferenciais podiam ser expressas por funções elementares do cálculo.

Durante o século XVIII métodos mais sistemáticos de resolução de equações diferenciais foram desenvolvidos por Euler, Lagrange e Laplace,

começando a ficar claro que poucas equações diferenciais podiam ser resolvidas por métodos elementares. A partir deste ponto a grande preocupação passou a ser encontrar condições de existência e unicidade para as soluções e deduzir propriedades da solução através da análise da própria equação diferencial.

Liouville, em 1841, provou que, em certos casos, principalmente para equações não lineares, não é possível obter a solução de uma equação diferencial por métodos elementares, mesmo sabendo-se que a solução existe e é única.

As equações lineares são muito mais conhecidas e estudadas que as não lineares. No entanto, muitos fenômenos de grande importância prática e teórica são descritos por equações não lineares. Grande parte dos matemáticos atuais considera que o estudo das equações diferenciais não lineares é uma área de pesquisa aberta em matemática. Resultados significativos têm sido obtidos na atualidade, em particular, na pesquisa de sistemas de equações, que podem exibir comportamentos surpreendentes. Devido às dificuldades deste estudo, até recentemente, tudo o que se podia fazer era estudar os casos lineares ou aproximações lineares de sistemas caóticos. Espera-se que o grande esforço que vem sendo feito nesta área se frutifique em praticamente todas as áreas de aplicação. Daí, a necessidade de maior desenvolvimento da análise quantitativa e dos métodos computacionais para a solução de equações diferenciais.

Definições Elementares e Classificações das Equações Diferenciais³

1) Definição de equação diferencial

Uma equação que contém as derivadas ou diferenciais de uma ou mais variáveis dependentes em relação a uma ou mais variáveis independentes é chamada de *equação diferencial*.

2) Classificação por tipo EDO e EDP:

Se uma equação contiver somente derivadas ordinárias de uma ou mais variáveis dependentes em relação a uma única variável independente, ela será denominada de *equação diferencial ordinária* (EDO). Exemplos:

³ As definições básicas e classificações das equações diferenciais foram retiradas de (Zill, 2012).

$$\frac{dy}{dx} + 5y = 3x^2 \quad \text{e} \quad \frac{dx}{dt} + \frac{dy}{dt} = 4x + y$$

Uma equação que envolve as derivadas parciais de uma ou mais variáveis dependentes em relação a duas ou mais variáveis independentes é denominada de *equação diferencial parcial* (EDP). Exemplos:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial u}{\partial y} = 2 \quad \text{e} \quad \frac{\partial^2 u}{\partial x^2} = \frac{\partial u}{\partial y} + 5 \frac{\partial v}{\partial x}$$

3) Classificação por ordem:

A ordem de uma equação diferencial (EDO ou EDP) é a ordem da maior derivada que aparece na equação. Exemplos:

$$\frac{d^4 y}{dx^4} + 5y^5 = \frac{dy}{dx} - 4x \quad (\text{EDO de quarta ordem})$$

$$\frac{\partial^3 u}{\partial x^3} = \left(\frac{\partial u}{\partial y}\right)^4 - 2 \frac{\partial^2 v}{\partial x^2} \quad (\text{EDP de terceira ordem})$$

4) Classificação por linearidade:

Uma equação diferencial de ordem n é dita linear se for expressa por uma combinação linear de todas as derivadas presentes na equação, caso contrário, será não linear. Exemplos:

$$2x \frac{d^5 y}{dx^5} + 3x^2 \frac{d^2 y}{dx^2} - 4y = 0 \quad (\text{EDO linear de quinta ordem})$$

$$4x^4 \frac{d^5 y}{dx^5} - 2x^3 \frac{d^2 y}{dx^2} - 3y^2 = 0 \quad (\text{EDO não linear de quinta ordem})$$

$$3 \frac{\partial^4 u}{\partial x^4} + 2x \frac{\partial^2 u}{\partial x^2} - 4y \frac{\partial^2 v}{\partial y^2} + u = 0 \quad (\text{EDP linear de quarta ordem})$$

$$uv \frac{\partial^3 u}{\partial x^3} + 4xy \frac{\partial^2 v}{\partial y^2} + (u + v)^4 = 0 \quad (\text{EDP não linear de terceira ordem})$$

5) Definição de Solução:

Toda função h , definida em um intervalo I , que tem pelo menos n derivadas contínuas em I , as quais quando substituídas em uma equação diferencial de

ordem n reduzem a equação a uma identidade é denominada uma solução da equação diferencial no intervalo I . Exemplos:

$$F(x, h(x), h^{(1)}(x), \dots, h^{(n)}(x)) = 0 \quad \text{para todo } x \in I.$$

Uma forma de verificar a solução é observar, depois de substituir, se ambos os lados da equação são iguais para cada x no intervalo. Considere a seguinte EDO,

$$\frac{dy}{dx} = \frac{x}{\sqrt{y}}, \text{ onde } x \in (-\infty, \infty)$$

Logo, $y(x) = x^4/16$ é a solução da EDO, pois

$$\frac{dy}{dx} = \frac{4x^3}{16} = \frac{x^3}{4} = \frac{x}{\sqrt{y}}, \text{ para todo } x \in (-\infty, \infty)$$

As soluções de EDPs também são verificadas da mesma forma.

2.4.3

Equações Diferenciais Estocásticas

Para um melhor entendimento das metodologias e aplicações que serão apresentadas na seção 4.2, faz-se necessário uma breve introdução sobre os aspectos teóricos das equações diferenciais estocásticas (EDEs) e do cálculo estocástico de Itô. Para maiores detalhes teóricos sobre EDEs, sugere-se consultar as referências bibliográficas utilizadas na confecção deste texto, que são: (Arnold, 1974; Schuss, 1980; Kloeden, 1992; Øksendal, 2003; (Braumann, 2005).

De acordo com a literatura pesquisada, os primeiros estudos formais sobre equações diferenciais estocásticas (EDEs) surgem no início do século passado com os trabalhos de Bachelier, Einstein, Ornstein, Uhlenbeck, Wiener e Brown, dentre outros. Dada a relevância do tema, a teoria evoluiu significativamente ao longo do tempo, ganhou importância prática e despertou o interesse de muitas áreas do conhecimento científico, em particular, da economia e finanças. Sabe-se, que a evolução de fenômenos econômicos definem trajetórias de processos estocásticos que, por definição, dependem do tempo e da incerteza. Tais variáveis influenciam fortemente e são essenciais para se entender o comportamento e as decisões dos agentes econômicos. Por esta razão é que o estudo e as soluções de

EDEs são de fundamental importância para a análise econômica e financeira moderna.

As EDEs são basicamente equações diferenciais determinísticas acrescidas de um termo estocástico perturbatório. O termo determinístico, comum nas equações diferenciais exatas, descreve o comportamento dinâmico médio e o termo estocástico descreve o ruído, ou seja, as perturbações aleatórias que influenciam o fenômeno em análise. Formalmente, define-se uma EDE unidimensional, na sua forma geral tradicional, pela seguinte expressão:

$$dX(t) = a(t, X(t)) dt + b(t, X(t)) dW(t); \quad t \in [0, T] \quad (1)$$

onde,

$\{X(t); t \in [0, T]\}$: é o processo estocástico de interesse, definido em um espaço de probabilidade real (Ω, \mathcal{F}, P) ;

$a(t, X(t))$: é o coeficiente da tendência (*drift*), momento infinitesimal que mede a velocidade média de X no instante t ;

$b(t, X(t))$: é o coeficiente de difusão (volatilidade), momento infinitesimal que mede as flutuações e velocidade da variância de X no instante t ;

$W(t)$: processo de Wiener (ou movimento Browniano padrão).

Em geral, as expressões matemáticas do *drift* e da volatilidade dependem de um conjunto de parâmetros. Se estes parâmetros são constantes no tempo, denota-se que o processo é homogêneo, caso contrário, heterogêneo.

Uma forma equivalente de representar o problema é na sua forma integral:

$$X(t) = X(0) + \int_0^t a(s, X(s)) ds + \int_0^t b(s, X(s)) dW(s) \quad (2)$$

A primeira integral pode ser interpretada como uma integral de Riemann-Stieltjes, mas a segunda não. Pelo fato da trajetória do processo de Wiener ser de variação ilimitada em qualquer intervalo de tempo finito, a segunda integral, em geral, não pode ser vista e solucionada como uma integral de Riemann-Stieltjes, a menos que $b(s, X_s)$ seja uma constante ou uma função de comportamento suave e não estocástica.

Esta integral estocástica é conhecida como a integral de Itô, e a sua definição para a classe de funções $b \in H_2$, tal que b é contínua em média quadrática, é dada pelo seguinte limite:

$$\int_0^t b(s, X_s) dW_s = \text{l.i.m.}_{\delta_n \rightarrow 0} \sum_{i=1}^n b(t_{i-1}, X_{t_{i-1}}) (W_{t_i} - W_{t_{i-1}}) \quad (3)$$

onde, $0 = t_0 < t_1 < \dots < t_n = t$; $\delta_n = \max\{t_i - t_{i-1}\}$

l.i.m. é a notação de limite em média quadrática; H_2 é um espaço de Banach com norma

$$\|b\| = \left(\int_0^t E(b(s, \cdot)) ds \right)^{1/2},$$

onde E é o operador de valor esperado.

A integral de Itô tem outras definições baseadas em diferentes formas de convergência, mas os elementos necessários e essenciais requeridos são, quase sempre, os mesmos: funções não antecipativas, aproximação de $b(t, X(t))$ por uma sequência de funções simples, etc. Uma das dificuldades em se trabalhar com esta forma de integral é que a mesma não segue as regras usuais do cálculo, conforme mostra o exemplo abaixo:

$$\int_0^t W_s dW_s = \text{l.i.m.}_{\delta_n \rightarrow 0} \sum_{i=1}^n W_{t_{i-1}} (W_{t_i} - W_{t_{i-1}}) = \frac{W_t^2}{2} - \frac{t}{2}.$$

Observe que, um termo adicional $(-t/2)$ aparece na solução da integral de Itô e que difere do resultado que teríamos pela integral de Riemann. Além disso, as regras usuais de diferenciação também não funcionam, veja o simples exemplo:

$$d(W_t^2) = 2 W_t dW_t + dt, \quad \text{que é diferente de } (2 W_t dW_t).$$

Pelo fato das regras usuais de integração e diferenciação não funcionarem, a solução de EDEs demandou o desenvolvimento de uma nova teoria de cálculo diferencial e integral, conhecida como cálculo estocástico, onde uma das principais contribuições é o teorema de Itô, apresentado a seguir.

Teorema de Itô (fórmula de Itô)

Seja $\{X(t); t \in [0, T]\}$ um processo de Itô. Ou seja, um processo estocástico que satisfaça a equação diferencial (1) ou a integral (2), com $X(0) = X_0$. Seja $Y(t) = U(t, X(t))$ uma função contínua com derivadas parciais U_t , U_x e U_{xx} contínuas. Então, $Y(t)$ também é um processo de Itô, com condição inicial $Y(0) = U(0, X(0)) = Y_0$, com equação diferencial dada pela *fórmula de Itô*:

$$dY(t) = \left(U_t + U_x a(t, X(t)) + \frac{1}{2} U_{xx} b^2(t, X(t)) \right) dt + U_x b(t, X(t)) dW(t) \quad (4)$$

ou

$$dY(t) = \tilde{a}(t, X(t)) dt + \tilde{b}(t, X(t)) dW(t) \quad (4)$$

onde,

$$U_t = \frac{\partial U(t, X(t))}{\partial t}, \quad U_x = \frac{\partial U(t, X(t))}{\partial x} \quad \text{e} \quad U_{xx} = \frac{\partial^2 U(t, X(t))}{\partial x^2}.$$

Com forma integral,

$$Y(t) = Y_0 + \int_0^t \left(\frac{\partial U(s, X(s))}{\partial s} + \frac{\partial U(s, X(s))}{\partial x} a(s, X(s)) + \frac{1}{2} \frac{\partial^2 U(s, X(s))}{\partial x^2} b^2(s, X(s)) \right) ds + \int_0^t \frac{\partial U(s, X(s))}{\partial x} b(s, X(s)) dW(s) \quad (5)$$

ou

$$Y(t) = Y_0 + \int_0^t \tilde{a}(s, X(s)) ds + \int_0^t \tilde{b}(s, X(s)) dW(s).$$

#

Deve-se ressaltar que o teorema de Itô é de fundamental importância para a solução de EDEs, principalmente para aquelas que na sua formulação inicial dependem da solução de complexas integrais estocásticas. A fórmula de Itô estabelece uma regra de diferenciação para funções compostas que pode ser interpretada como uma *regra da cadeia* para diferenciais estocásticos. O resultado primordial é que, através da sua aplicação a uma função $U(t, X(t))$, adequadamente escolhida, pode-se transformar o problema original $dX(t)$ em um novo problema $dY(t)$, sem se alterar o diferencial $dW(t)$. Consequentemente, este procedimento pode ser repetido várias vezes até que se obtenha uma EDE com solução conhecida ou de equacionamento mais simples.

Observando a integral de Itô da expressão (5), tem-se que

$$\int_0^t \tilde{b}(s, X(s)) dW(s) = \int_0^t \frac{\partial U(s, X(s))}{\partial x} b(s, X(s)) dW(s).$$

Portanto, se for possível escolher uma função U , com as condições exigidas pelo teorema, de tal forma que se tenha

$$\frac{\partial U(s, X(s))}{\partial x} = \frac{1}{b(s, X(s))},$$

então a integral estará solucionada. Pois,

$$\int_0^t \tilde{b}(s, X(s)) dW(s) = \int_0^t dW(s) = [W(t) - W(0)] = W(t).$$

Onde, por definição do processo de Wiener, $W(0) = 0$.

Este procedimento é uma decorrência natural e imediata do teorema de Itô, conhecido na literatura especializada como a transformação de Lamperti, veja (Zenghu Li, 2012; Giesecke, 2013;). A idéia central é transformar a equação original $dX(t)$ em $dY(t)$, isolando por completo o coeficiente $b(t, X(t))$ do diferencial estocástico. Ou seja, transformar

$$dX(t) = a(t, X(t))dt + b(t, X(t)) dW(t) \text{ em } dY(t) = f(t, X(t)) dt + dW(t).$$

Quando pode ser utilizado, facilita consideravelmente a solução do problema, pois a integral estocástica resultante da transformação é de simples solução. No entanto, o procedimento não pode ser aplicado a todos os casos de EDEs, pois requer, além das exigências do teorema, que a transformação $Y(t) = U(t, X(t))$ seja invertível e que $[1/b(t, X(t))]$ seja uma função contínua e diferenciável sob o sinal da integral.

Para exemplificar o uso do teorema de Itô na solução de EDEs, utilizaremos a equação clássica de Movimento Browniano Geométrico (MBG), muito utilizada para modelar a dinâmica dos preços de ativos financeiros e que é a base do conhecido modelo de precificação de opções de Black & Scholes, ver (Black, 1973; Campbell, 1997; Hull, 1997).

Seja $\{X(t): t \in [0, T]\}$ um processo de Itô que satisfaz a seguinte equação diferencial:

$$dX(t) = \mu X(t) dt + \sigma X(t) dW(t) \quad (6)$$

onde,

$$a(t, X(t)) = \mu X(t), \quad b(t, X(t)) = \sigma X(t), \text{ tal que } \sigma > 0 \text{ e } X(t) > 0 \quad \forall t \in [0, T].$$

Defina a transformação $Y(t) = U(t, X(t))$, tal que $U_x = 1/b(t, X(t))$. Então,

$$Y(t) = U(t, X(t)) = \frac{\ln(X(t))}{\sigma} \Rightarrow \frac{\partial U(t, X(t))}{\partial x} = \frac{1}{\sigma X(t)} = \frac{1}{b(t, X(t))}.$$

Aplicando a fórmula de Itô em $Y(t)$, expressão (4), tem-se que:

$$\begin{aligned} dY(t) &= \left(U_t + U_x a(t, X(t)) + \frac{1}{2} U_{xx} b^2(t, X(t)) \right) dt + U_x b(t, X(t)) dW(t) = \\ &= \left(\frac{1}{\sigma X(t)} \mu X(t) - \frac{1}{2} \frac{1}{\sigma [X(t)]^2} [\sigma X(t)]^2 \right) dt + \frac{1}{\sigma X(t)} \sigma X(t) dW(t) = \\ &= \left(\frac{\mu}{\sigma} - \frac{\sigma}{2} \right) dt + dW(t). \end{aligned}$$

Então, $Y(t)$ é um novo processo de Itô, mais simplificado, com EDE

$$dY(t) = \tilde{a}(t, X(t)) dt + \tilde{b}(t, X(t)) dW_t$$

$$\text{onde, } \tilde{a}(t, X(t)) = \left(\frac{\mu}{\sigma} - \frac{\sigma}{2}\right) \text{ e } \tilde{b}(t, X(t)) = 1.$$

Agora, aplicando o operador de integral em $dY(t)$, chega-se as soluções para $Y(t)$ e $X(t)$. Ou seja,

$$\int_0^t dY(s) = \int_0^t \tilde{a}(s, X(s)) ds + \int_0^t \tilde{b}(s, X(s)) dW(s) =$$

$$Y(t) - Y(0) = \int_0^t \left(\frac{\mu}{\sigma} - \frac{\sigma}{2}\right) ds + \int_0^t 1 dW(s).$$

$$\text{Portanto, } Y(t) = Y(0) + \left(\frac{\mu}{\sigma} - \frac{\sigma}{2}\right)t + W(t).$$

Onde $Y(0) = \ln(X(0))/\sigma$ é a condição inicial de $Y(t)$ e $X(0)$, uma constante positiva, é a condição inicial de $X(t)$. Em termos probabilísticos, decorre da normalidade de $W(t)$ que $Y(t)$ também é normalmente distribuído. Isto é,

$$Y(t) \sim \text{Normal}(\mu_{Y_t}, \sigma_{Y_t}^2), \text{ onde } \mu_{Y_t} = Y(0) + \left(\frac{\mu}{\sigma} - \frac{\sigma}{2}\right)t \text{ e } \sigma_{Y_t}^2 = t.$$

Por inversão de $Y(t)$, pode-se agora obter a solução do processo $X(t)$. Então,

$$Y(t) = \frac{\ln(X(t))}{\sigma} \Rightarrow X(t) = \exp\{\sigma Y(t)\} = \exp\left\{\sigma \left[Y(0) + \left(\frac{\mu}{\sigma} - \frac{\sigma}{2}\right)t + W(t)\right]\right\}.$$

Logo,

$$X(t) = X(0) \exp\left\{\left(\mu - \frac{\sigma^2}{2}\right)t + \sigma W(t)\right\} \sim \text{LogNormal}(\mu_{X_t}, \sigma_{X_t}^2).$$

#

3

Algoritmo de Programação Genética e Diferenciação Automática

3.1

Introdução

Nesta seção são apresentados os procedimentos metodológicos gerais adotados na elaboração do algoritmo de programação genética (PG) com diferenciação automática (DA), desenvolvido especificamente para solucionar problemas de equações diferenciais, que denominamos de PGDA. Antes da apresentação, deve-se ressaltar que a elaboração desta metodologia não foi de fácil realização, pois, embora exista uma vasta literatura sobre os temas, na grande maioria dos trabalhos pesquisados, os assuntos são tratados de forma separada e independente. Portanto, muito do que foi desenvolvido para combinar PG com DA não aparece na literatura e é fruto deste trabalho.

Em linhas gerais, a menos de algumas especificidades introduzidas para a implementação da DA e otimização do tamanho de árvore, a metodologia aqui descrita assemelha-se à apresentada na seção 2.2. É importante registrar que os algoritmos desenvolvidos foram programados em ambiente Matlab e utilizam-se, em parte, dos programas: Gplab, (Sliva, 2009), para a elaboração do algoritmo de PG; e myAD e myA2D, (Fink, 2007), para a DA.

3.2

Estrutura e Etapas de Funcionamento do algoritmo de PGDA

Conforme citado anteriormente, o algoritmo de PGDA foi construído com o objetivo específico de solucionar problemas de equações diferenciais. A sua estrutura e etapas de funcionamento é muito semelhante a do algoritmo tradicional apresentado na seção 2.2.

A estrutura de representação dos indivíduos é na forma de árvore e a sua utilização é relativamente simples e amigável. No entanto, ainda é necessário

conhecer um pouco de programação em ambiente Matlab para entender a sua lógica e dominar plenamente a sua execução. Em linhas gerais, o PGDA é constituído de dois programas básicos elaborados pelo usuário, que funcionam simultaneamente, e que executam os programas `gplab.m`, `myAD` e `myA2D`. O primeiro é responsável pela execução da programação genética e contém toda a parametrização adicional discordante dos *defaults* do GPLAB. O segundo é o programa que executa a diferenciação automática e avalia a aptidão (*fitness*) de cada indivíduo gerado pelo primeiro.

3.2.1

Parametrização do Algoritmo

A parametrização do algoritmo de PG é muito flexível e permite ao usuário escolher e definir um amplo conjunto de parâmetros importantes para a sua execução. O usuário menos familiarizado com este tipo de modelagem poderá utilizar os parâmetros pré-definidos em *defaults*. A lista abaixo mostra algumas das possibilidades de parametrização, mas não esgota o total permitido.

- Tipos e formas de árvores;
- Profundidade mínima da árvore inicial;
- Profundidade máxima da árvore (ou número de *depth*);
- Controle de nós (ou *nodes*);
- Seleção do conjunto de funções elementares e terminais que deseja-se trabalhar;
- Percentual ou número de indivíduos e diferentes métodos de seleção para reprodução;
- Taxas de crossover e mutação fixas ou variáveis;
- Tamanho da população;
- Número de gerações e outros critérios de parada.

Todas as parametrizações do modelo são declaradas e registradas na função `“availableparams.m”`, a qual é lida pelo code `“gplab.m”` quando da execução do algoritmo de PG.

Quadro 2: Funções Elementares Utilizadas pelo Algoritmo de PG

Nome	Função	Argumentos	Saída
Soma	plus	(a, b)	a+b
Subtração	minus	(a, b)	a-b
-um	uminus	(a)	-a
Divisão	rdivide	(a, b)	a/b
Multiplicação	times	(a, b)	a*b
Raiz quadrada	sqrt	(a)	a^(1/2)
Exponencial	exp	(a)	exp(a)
Logaritmo Natural	log	(a)	ln(a)
Potência	power	(a, b)	a^b
Seno	sin	(a)	sen(a)
Coseno	cos	(a)	cos(a)
Tangente	tan	(a)	tan(a)
Secante	sec	(a)	sec(a)
(*) Divisão	mydivide	(a, b)	a/b se b ≠ 0; e 1 se b=0.
(*) Raiz quadrada	mysqrt	(a)	a^(1/2) se a ≥ 0; e 1 se a < 0.
(*) Logaritmo Natural	mylog	(a)	ln(abs(a)) se a ≠ 0; e 1 se a = 0.
(*) Potência	mypower	(a, b)	a^b se a>0; e 1 se a ≤ 0.

(*) Este símbolo representa que a função é protegida.

O conjunto de funções e terminais é essencial para o funcionamento do algoritmo, pois implica diretamente na criação da população de indivíduos e, conseqüentemente, na solução ótima dada ao problema. Juntos, funções e terminais definem o tamanho da árvore de representação. O conjunto de terminais é formado de números aleatórios (gerados pela função *rand*), de constantes (*números reais e complexos, número π etc.*) e das variáveis que compõem a equação diferencial. Algumas das funções utilizadas são apresentadas no Quadro 2. Além das funções disponíveis, o algoritmo ainda permite que o usuário mais treinado e experiente construa as suas próprias funções.

3.2.2

População Inicial e Avaliação dos Indivíduos

A primeira tarefa que o algoritmo realiza é a criação da população inicial. Funções e terminais são selecionados aleatoriamente e combinados, dando origem aos indivíduos que compõem a população. No caso, cada indivíduo criado é uma

função matemática e, portanto, uma possível solução da equação diferencial de interesse.

Três métodos são utilizados pelo PGDA para inicializar a população em estrutura de árvore: *Full*, *Grow* e *ramped*. As descrições destes métodos já foram apresentadas e podem ser vistas na seção 2.2.4. As funções que executam estes métodos são nomeadas pelo GPLAB de *Fullinit*, *Growinit* e *rampedinit*. Uma vez inicializada a população é realizada a avaliação da aptidão dos indivíduos.

Para a avaliação dos indivíduos recém-criados é necessária a definição de uma medida de aptidão (*fitness*) que avalie e compare as qualidades de cada indivíduo. Em virtude de o objetivo ser a solução de equações diferenciais, surge, neste ponto, a necessidade da diferenciação automática para, via cálculo de derivadas, avaliar e verificar a qualidade da solução proposta. Para exemplificar como isso é feito no algoritmo de PGDA, suponha que se queira solucionar o seguinte problema de EDO:

$$y'(x) + 2 y(x) = \exp(-2x)$$

$$\text{com } x \in \mathbb{R}; \text{ e condição inicial } y(0) = 3.$$

Suponha, também, que a função a seguir seja um indivíduo ou solução proposta pela PG e, portanto, a sua aptidão deve ser avaliada.

$$f(x) = 2 + \exp(-2x)$$

Então, a diferenciação automática calcula a derivada de $f(x)$ em cada ponto do seu domínio, definido pelo algoritmo (geralmente uma grade com 30 ou mais pontos de x), e gera uma medida de *fitness* que compara os resultados obtidos com o modelo de EDO. Então, de acordo com o exemplo, segue que:

$$f'(x) = -2 \exp(-2x)$$

Aplicando este resultado na EDO, teríamos:

$$f'(x) + 2 f(x) - \exp(-2x) = -2 \exp(-2x) + 2[2 + \exp(-2x)] - \exp(-2x) =$$

$$= 4 - \exp(-2x) = \text{erro} \neq 0.$$

Observe que $f(x)$ apresenta erro diferente de zero e, portanto, não é a solução exata da EDO proposta, embora satisfaça a condição inicial,

$$f(0) = 2 + \exp(-2 \times 0) = 3.$$

Contudo, o algoritmo não trabalha com a diferenciação simbólica da solução proposta, como feito acima, e sim com a derivada avaliada em cada ponto do domínio. Nesta situação, uma medida natural de *fitness* seria a média ou a soma

dos erros absolutos, ou ainda as medidas sugeridas na seção 2.2.5, acrescida de uma penalidade para o erro da condição inicial ou de outras restrições exigidas pelo problema, se for o caso. Um exemplo seria,

fitness = erro médio absoluto + erro de restrições.

$$fitness = \left\{ \frac{1}{n} \sum_{i=1}^n |f'(x_i) + 2f(x_i) - \exp(-2x_i)| \right\} + |f(0) - 3|.$$

Portanto, o que o algoritmo de PGDA faz é buscar o indivíduo ou a solução $f(x)$ que minimize o *fitness* ou maximize a medida de aptidão. Quando a função obtida for a solução exata do problema, necessariamente, o *fitness* será nulo.

3.2.3

Reprodução e Geração da Nova População

A próxima etapa do PGDA é a seleção de indivíduos para reprodução. Esta etapa é de fundamental importância, pois potencializa o processo de otimização. Seleciona os indivíduos da população que participarão do processo de reprodução genética e que deixarão descendentes para as gerações futuras. Portanto, é natural esperar que os métodos de seleção escolham os indivíduos mais aptos. Com isso, a transmissão de bom material genético contribuirá para o desenvolvimento de melhores soluções.

Embora se disponha de cinco métodos de seleção para reprodução, que são: *lexictour*, *doubletour*, *sus*, *roulette*, *tournament*, o algoritmo de PGDA apresenta melhores resultados, em termos de parcimônia de árvore e de *fitness*, quando utiliza o método *lexictour*. Por este motivo ele foi adotado como padrão e foi utilizado em grande parte das aplicações realizadas.

O método *lexictour* (Luke, 2002) foi desenvolvido com o conceito de *pressão lexicográfica parcimoniosa*. Neste método, um número de indivíduos é selecionado aleatoriamente da população e o de melhor avaliação ou mais apto é escolhido. Mas, caso dois ou mais indivíduos tenham a mesma avaliação, o método seleciona o indivíduo de menor tamanho. Ou seja, o que possui a árvore com o menor número de nós (*nodes*). Segundo diversos autores, este método tem se mostrado bastante eficiente no controle do tamanho excessivo das árvores.

Os operadores genéticos, de *Crossover e Mutação*, usados no PGDA, trabalham com taxas variáveis e são os mesmos descritos na seção 2.2.7. O método de reprodução utilizado é o *steady-state*, descrito na seção 2.2.8. A validação dos novos indivíduos é feita em função da profundidade e do número de nós da árvore. Caso o tamanho do novo indivíduo exceda ao tamanho máximo parametrizado no algoritmo, este não será validado.

Depois da geração da população, o algoritmo define a sobrevivência e a nova população. Quatro são as formas de *elitismo* disponíveis, *keepbest*, *halfelitism*, *totalelitism* e *replace*. O PGDA utiliza na maioria das aplicações o método de *totalelitism*, onde os indivíduos sobrevivem em função das suas aptidões, independentemente de serem pais ou filhos.

3.2.4

Critério de Parada e a Solução de PGDA

Após a reprodução e a geração da nova população o processo se repete até que o critério de parada seja atingido. O critério de parada utilizado no algoritmo é o número de gerações que, dependendo da complexidade do problema, pode ser muito variado. Para as diversas aplicações realizadas no trabalho foram utilizados diferentes números de gerações, variando entre 20 e 100.

Outro aspecto importante da metodologia são os critérios que definem e elegem a solução analítica de uma equação diferencial pelo algoritmo de PGDA. De acordo com o critério de parada utilizado, o algoritmo funciona até que a última geração seja avaliada e, ao término das gerações, seleciona a melhor solução encontrada, ou seja, aquela que minimiza a medida de erro (*fitness*). A partir deste ponto, uma solução encontrada é validada como uma solução final de PGDA se satisfaz os seguintes requisitos:

- i) Se apresenta *fitness* abaixo de 0,00001;
- ii) se satisfaz as condições iniciais e condições de contorno do problema proposto; e
- iii) se quando diferenciada, no caso de EDO ou EDP, pelas regras do cálculo usual, e no caso de EDE, pelo cálculo estocástico, se aproxima ou se iguala à equação diferencial proposta em todo o domínio do problema.

Caso uma solução validada apresente *fitness* nulo e satisfaça os demais requisitos com perfeita exatidão, a mesma é considerada como uma *solução analítica exata* do problema proposto. Pois, esta solução, encontrada por PGDA, é idêntica à obtida pelo cálculo diferencial usual, no caso de EDO ou EDP, e estocástico, no caso de EDE. Caso contrário, a solução encontrada é considerada como uma solução analítica aproximada.

4

Aplicações e Análise dos Resultados

4.1

Solução de Equações Diferenciais por PGDA

Nesta seção são apresentados os resultados das diversas aplicações realizadas para solucionar EDOs e EDPs. No total, mais de 30 problemas foram tratados com o objetivo de testar a eficiência e eficácia do algoritmo de PGDA no sentido da obtenção de soluções analíticas exatas. Com a preocupação de que a exposição não ficasse muito extensa e cansativa decidiu-se apresentar apenas onze destes resultados, que se acredita que são suficientes para representar o embasamento de todos os tipos de equações trabalhadas.

De uma forma geral, o algoritmo foi parametrizado de acordo com os parâmetros indicados no Quadro 3.

Quadro 3: Parametrização do Algoritmo PGDA

Parâmetros Básicos da Programação Genética			
Gerações	20 a 100	Inicialização	Growinit
População (indivíduos)	30 a 600	Seleção e Reprodução	Lexictour
Funções Elementares	8 a 15	Elitismo	totalelitism
Terminais	5 a 15	Critério de Parada	Nº de Gerações
Variáveis	2 a 6	Taxa de Crossover	Variável
Modo de DA (fixo)	<i>forward</i>	Taxa de Mutação	Variável

O número de gerações de 20 a 100 e o tamanho da população de 30 a 600, dependendo do número de variáveis, da ordem de diferenciação, e do tamanho e complexidade da equação diferencial. O número de funções e operações matemáticas elementares variou de 8 a 15, e o número de terminais, variáveis mais constantes, variou de 5 a 15. A inicialização da população, a seleção para reprodução e a técnica de elitismo, para a maioria das aplicações, foram utilizados, respectivamente, os métodos *growinit*, *lexictour*, e *totalelitism*. As

taxas de crossover e mutação são variáveis e seguem o default do GPLAB, iniciando com 50% para cada operador. O critério de parada utilizado foi o número de gerações. Foram utilizados conjuntos ou grades de 20, 30 e 50 pontos do domínio da função para a realização dos exercícios.

A seção está organizada em quatro subseções que, respectivamente, apresentam os resultados dos problemas de EDOs; EDPs; o caso especial da Equação de Schrödinger para o átomo de He; e o problema da especificação arbitrária em modelos de regressão.

4.1.1

Solução de Equações Diferenciais Ordinárias

Nesta subseção são apresentados os resultados das resoluções de oito EDOs, das quais três são lineares de primeira ordem, três lineares de segunda ordem, uma não linear de primeira ordem e uma não linear de segunda ordem. Para todos os casos, na árvore da PG, a variável independente (x) é representada por ($x = x_1$).

Exemplo 1: EDO linear, de 1ª ordem, com coeficiente variável.

EDO1:	$y'(x) = \frac{dy(x)}{dx} = \frac{2x - y(x)}{x} ; \text{ se } x \neq 0$
--------------	---

com $x \in \mathbb{R} - \{0\}$; e $y(x) = -y(x)$. Ou seja, $y(x)$ é ímpar.

Solução analítica exata:	$y(x) = x + \frac{2}{x}$
---------------------------------	--------------------------

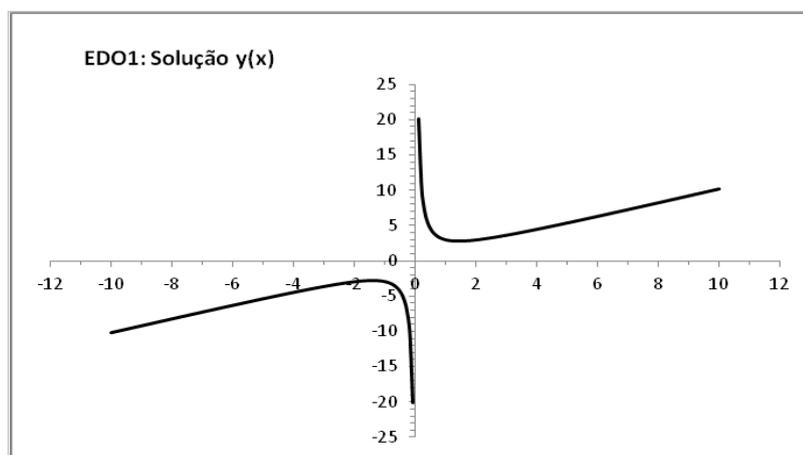
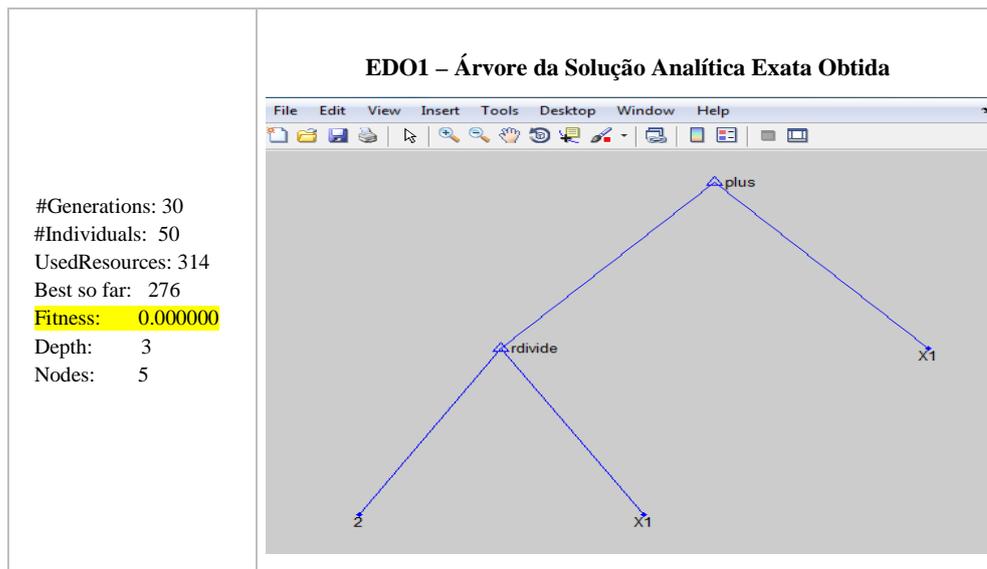


Figura 9: Exemplo 1 - Resultados da EDO1

A solução da EDO1 foi obtida no indivíduo 276 (bestsofar), para uma população de 50 indivíduos e 30 gerações. A solução encontrada é a exata e o fitness é nulo. A árvore de representação obtida é a mais parcimoniosa possível, com 5 nós (nodes) e profundidade 3 (depth). Este exemplo foi também desenvolvido por Tsoulis (2006), que obteve solução analítica exata, mas com um número médio de 653 gerações.

Exemplo 2: EDO linear, de 1ª ordem, com coeficiente constante.

EDO2:	$y'(x) + 2 y(x) = \exp(-x)$
--------------	-----------------------------

com $x \in \mathbb{R}$; e condição inicial $y(0) = 3$.

Solução analítica exata:	$y(x) = \exp(-x) + 2 \exp(-2x)$
---------------------------------	---------------------------------

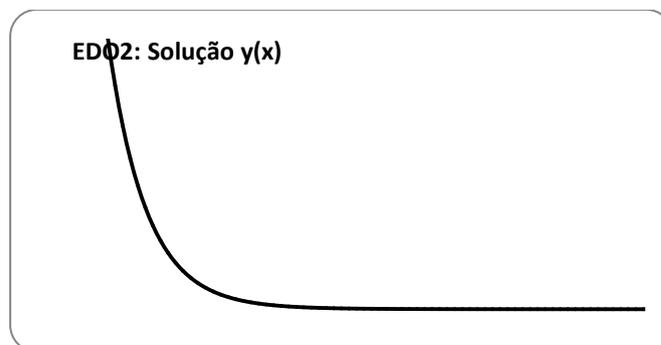
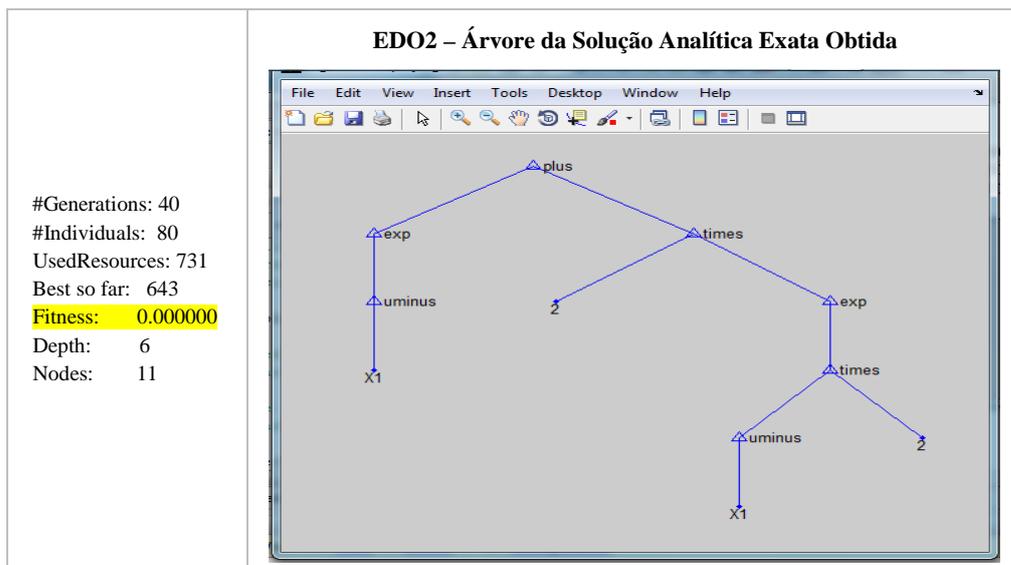


Figura 10: Exemplo 2 - Resultados da EDO2

A solução da EDO2 foi obtida no indivíduo 643 (bestsofar), para uma população de 80 indivíduos e 40 gerações. A solução analítica encontrada é a exata e o fitness é nulo. A árvore de representação obtida é a mais parcimoniosa possível, com 11 nós (nodes) e profundidade 6 (depth).

Exemplo 3: EDO linear, de 1ª ordem, com coeficiente variável.

EDO3:	$y'(x) + x y(x) = 10 x \exp\left(-\frac{x^2}{2}\right)$
--------------	---

com $x \in \mathbb{R}$; e condição inicial $y(0) = 0$.

Solução analítica exata:	$y(x) = 5 x^2 \exp\left(-\frac{x^2}{2}\right)$
---------------------------------	--

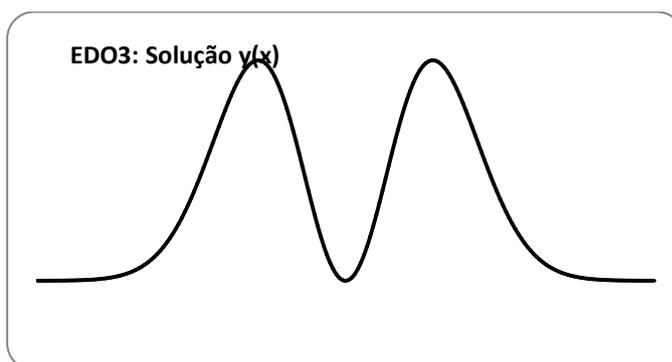
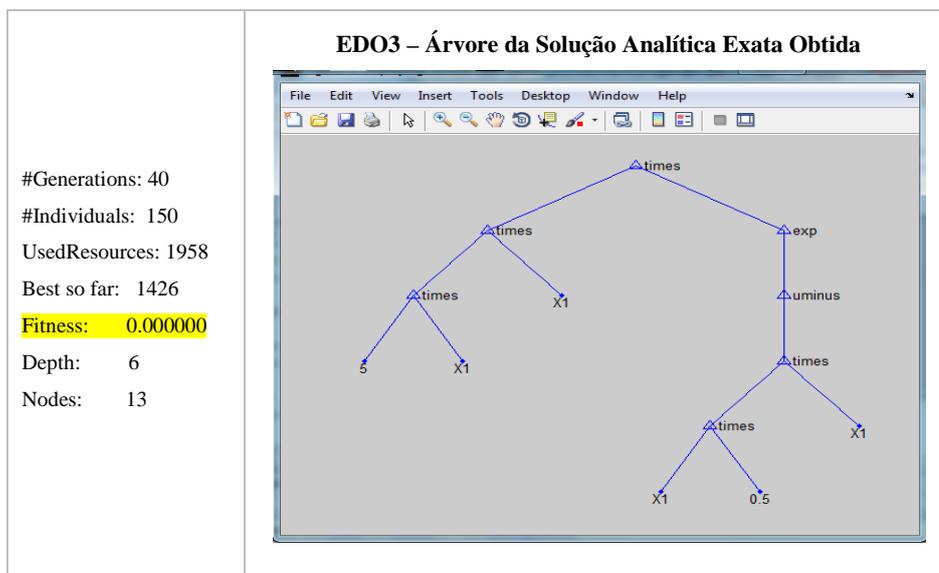


Figura 11: Exemplo 3 - Resultados da EDO3

A solução da EDO3 foi obtida no indivíduo 1426 (bestsofar), para uma população de 150 indivíduos e 40 gerações. A solução analítica encontrada é a exata e o fitness é nulo. A árvore de representação obtida é a mais parcimoniosa possível, com profundidade 6 (depth) e 13 nós (nodes).

Exemplo 4: EDO linear, de 2ª ordem, com coeficientes constantes.

EDO4:	$y''(x) + y'(x) + y(x) =$ $-1,6 + 0,4x + 0,2x^2 - 8 \operatorname{sen}(4x) - 30 \operatorname{cos}(4x)$
--------------	---

com $x \in \mathbb{R}$; e condições iniciais $y(0) = 0$ e $y'(0) = 0$.

Solução analítica exata:	$y(x) = -2 + 0,2x^2 + 2 \operatorname{cos}(4x)$
---------------------------------	---

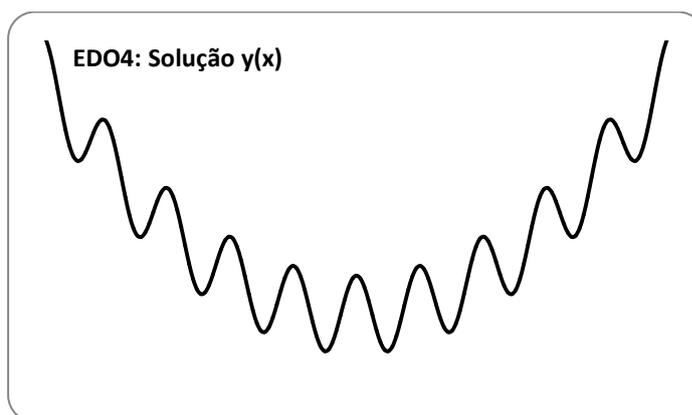
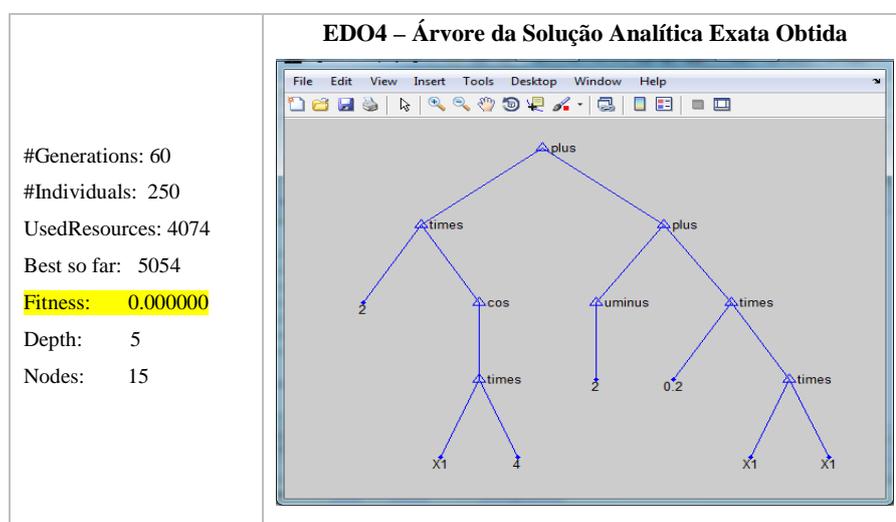


Figura 12: Exemplo 4 - Resultados da EDO4

A solução da EDO4 foi obtida no indivíduo 5054 (bestsofar), para uma população de 250 indivíduos e 60 gerações. A solução analítica encontrada é a exata e o fitness é nulo. A árvore de representação obtida é a mais parcimoniosa possível, com profundidade 5 (depth) e 15 nós (nodes).

Exemplo 5: EDO linear, de 2ª ordem, com coeficientes constantes.

EDO5:	$y''(x) + 0,3 y'(x) + 25 y(x) =$ $25,12 + 10x - 1,5 \cos(5x) \exp(-0,3x)$
--------------	---

com $x \in \mathbb{R}$; e condições iniciais $y(0) = 1$ e $y'(0) = 5,4$.

Solução analítica exata:	$y(x) = 1 + 0,4x + \text{sen}(5x) \exp(-0,3x)$
---------------------------------	--

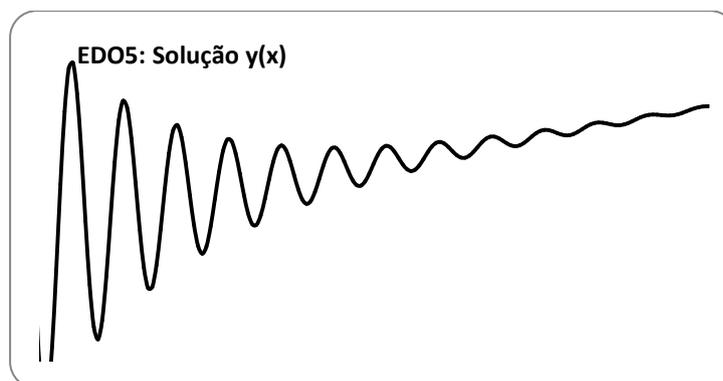
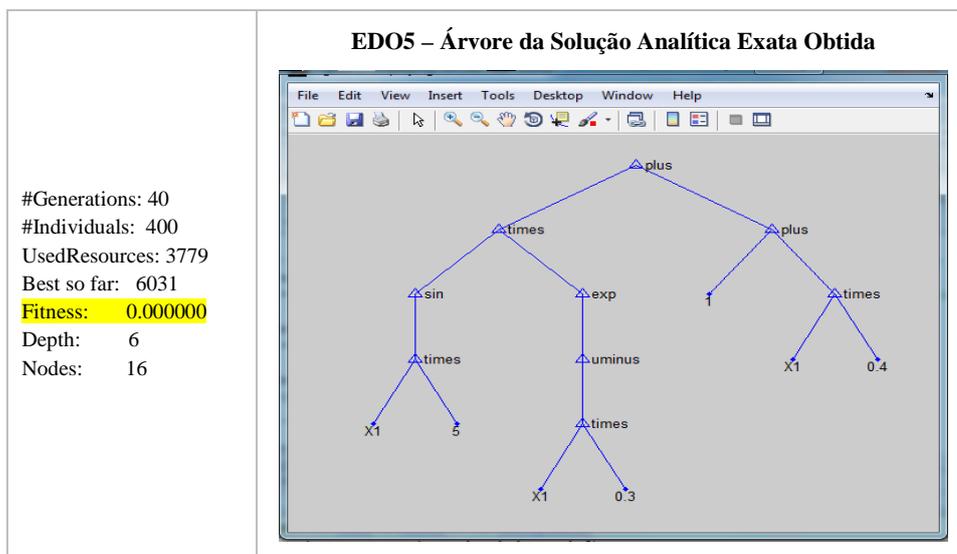


Figura 13: Exemplo 5 - Resultados da EDO5

A solução da EDO5 foi obtida no indivíduo 6031 (bestsofar), para uma população de 400 indivíduos e 40 gerações. A solução analítica encontrada é a exata e o fitness é nulo. A árvore de representação obtida é a mais parcimoniosa possível, com profundidade 6 (depth) e 16 nós (nodes).

Exemplo 6: EDO linear, de 2ª ordem, com coeficientes variáveis.

EDO6:	$x^2 y''(x) + x y'(x) + 4y(x) = 6 \operatorname{sen}(\ln(x))$
--------------	---

com $x \in \mathbb{R}$, tal que $x > 0$ e condições iniciais $y(1) = 3$ e $y'(1) = 2$.

Solução analítica exata:	$y(x) = 3 \cos(2 \ln(x)) + 2 \operatorname{sen}(\ln(x))$
---------------------------------	--

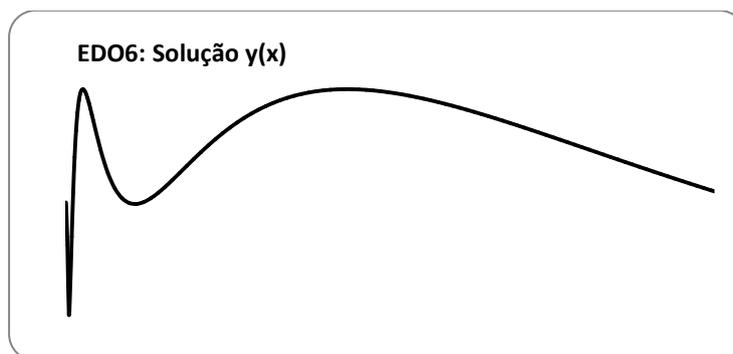
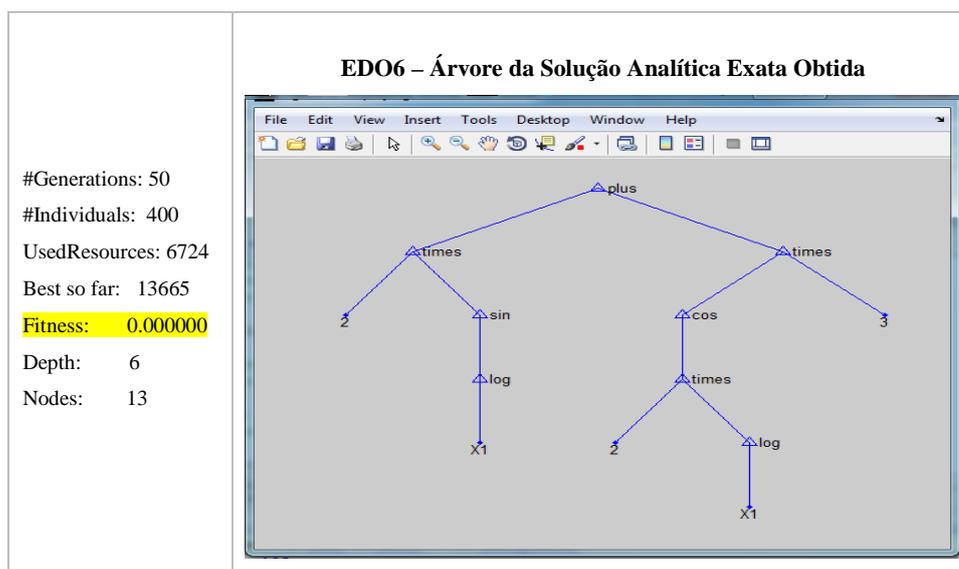


Figura 14: Exemplo 6 - Resultados da EDO6

A solução da EDO6 foi obtida no indivíduo 13665 (bestsofar), para uma população de 400 indivíduos e 50 gerações. A solução analítica encontrada é a exata e o fitness é nulo. A árvore de representação obtida é a mais parcimoniosa possível, com profundidade 6 (depth) e 13 nós (nodes).

Exemplo 7: EDO Não Linear, de 1ª ordem, com coeficiente constante.

EDONL1:	$(y'(x))^2 + \log(y(x)) = (1 + \cos(x))^2 + \log(x + \sin(x))$
----------------	--

com $x \in \mathbb{R}$, tal que $x > 0$; e condição inicial $y(1) = 1 + \sin(1)$.

Solução analítica exata:	$y(x) = x + \sin(x)$
---------------------------------	----------------------

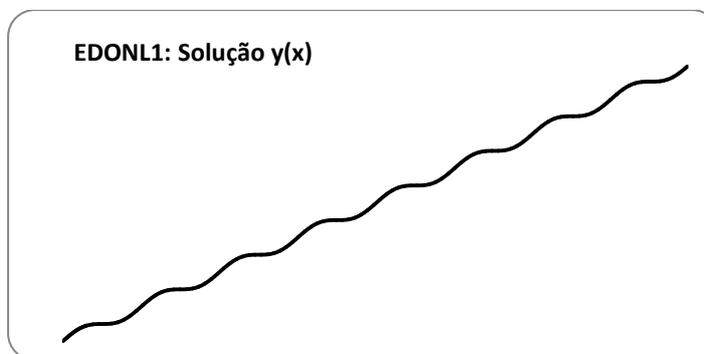
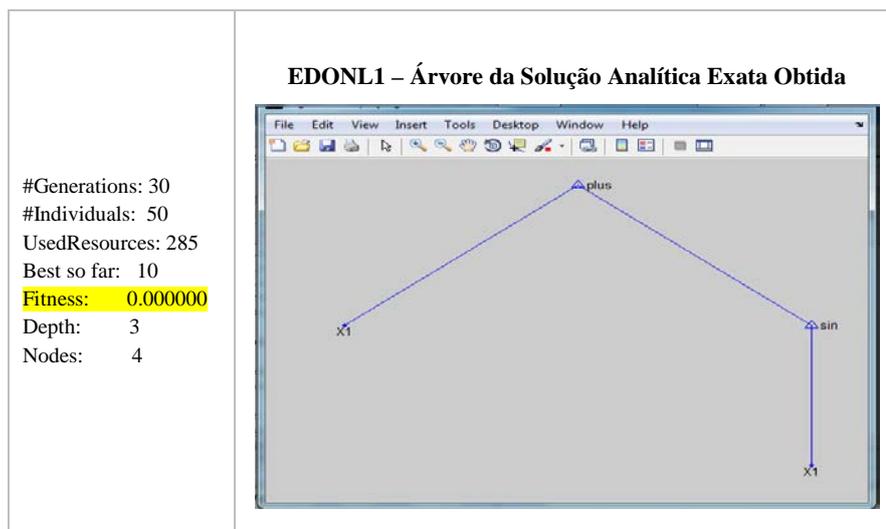


Figura 15: Exemplo 7 - Resultados da EDONL1

A solução da EDONL1 foi obtida no indivíduo 10 (bestsofar), para uma população de 50 indivíduos e 30 gerações. A solução analítica encontrada é a exata e o fitness é nulo. A árvore de representação obtida é a mais parcimoniosa possível, com profundidade 3 (depth) e 4 nós (nodes). Este exemplo também foi desenvolvido em Tsoulus (2006), que também obteve solução analítica exata, mas com um número médio de 86 gerações.

Exemplo 8: EDO Não Linear, de 2ª ordem, com coeficientes variáveis.

EDONL2:	$x^2 y''(x) + (x y'(x))^2 = -\frac{1}{\log(x)}$
----------------	---

com $x \in \mathbb{R}$, tal que $x > e$; com condições iniciais $y(e) = 0$; $y'(e) = 1/e$.

Solução analítica exata:	$y(x) = \log(\log(x))$
---------------------------------	------------------------

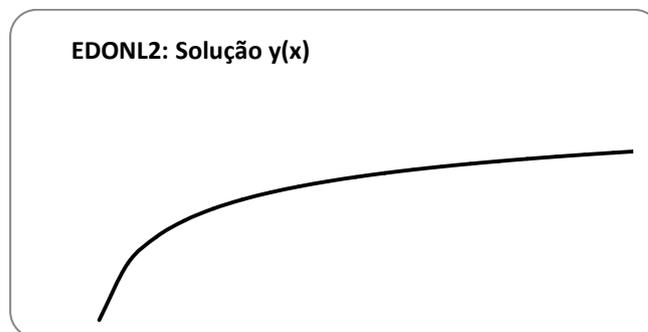
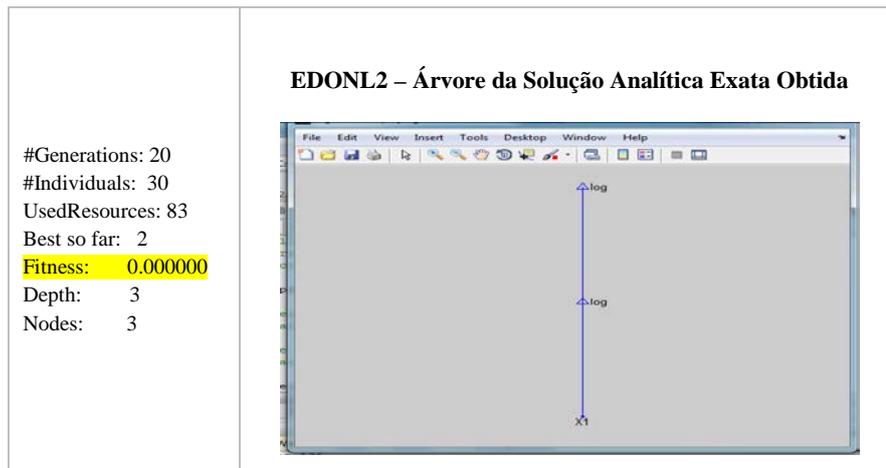


Figura 16: Exemplo 8 - Resultados da EDONL2

A solução da EDONL2 foi obtida no indivíduo 2 (bestsofar), para uma população de 30 indivíduos e 20 gerações. A solução analítica encontrada é a exata e o fitness é nulo. A árvore de representação obtida é a mais parcimoniosa possível, com profundidade 3 (depth) e 3 nós (nodes). Este exemplo foi também desenvolvido por Tsoulis (2006), que também obteve solução analítica exata, mas com um número médio de 161 gerações.

4.1.2

Solução de Equações Diferenciais Parciais

Nesta subseção são apresentados os resultados das resoluções de três EDPs lineares, das quais uma é de primeira ordem e duas de segunda ordem. Na árvore da PG para os três exemplos, as variáveis independentes x e t são representadas, respectivamente, pelas variáveis ($x = x_1$) e ($t = x_2$).

Exemplo 9: EDP linear, de 1ª ordem, com coeficientes variáveis

EDP1:	$xf_x + 2f_t = x\cos(x) + 2\sin(x)$
--------------	-------------------------------------

*com $(x, t) \in \mathbb{R}^2$, onde $f_x = \frac{\partial f(x, t)}{\partial x}$; $f_t = \frac{\partial f(x, t)}{\partial t}$;
e as condições $f(0, t) = 0$ e $f(x, 0) = x^2$.*

Solução analítica exata:	$f(x, t) = x^2 e^{-t} + t \sin(x)$
---------------------------------	------------------------------------

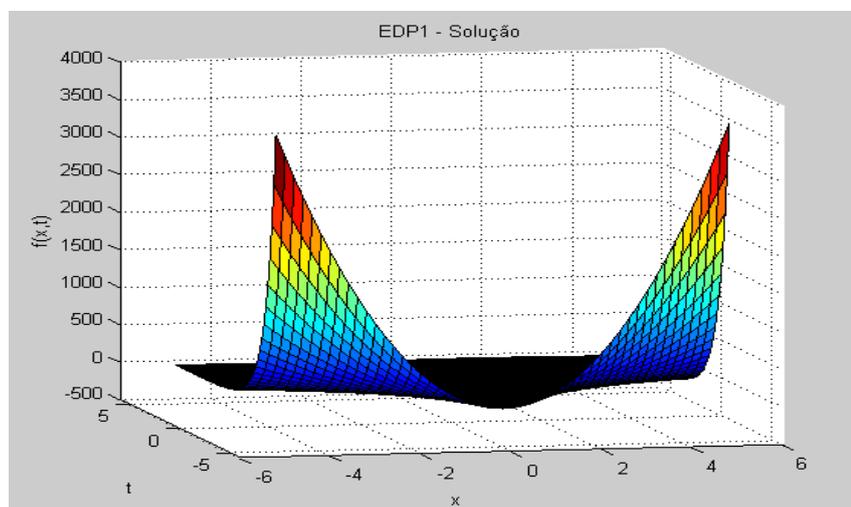
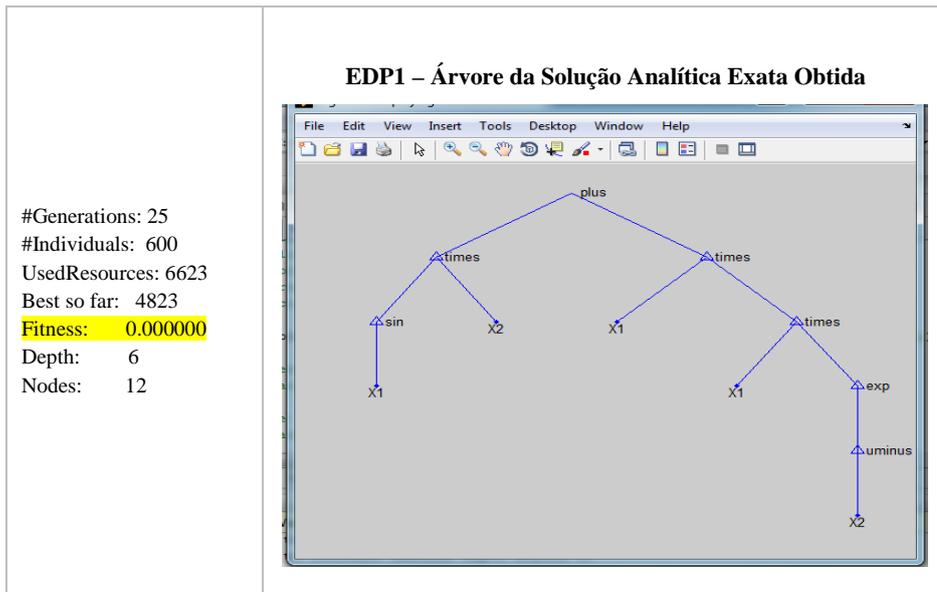


Figura 17: Exemplo 9 - Solução da EDP1

A solução da EDP1 foi obtida no indivíduo 4823 (bestsofar), para uma população de 600 indivíduos e 25 gerações. A solução analítica encontrada é a exata e o fitness é nulo. A árvore de representação obtida é a mais parcimoniosa possível, com profundidade 6 (depth) e 12 nós (nodes).

Exemplo 10: EDP de 2ª ordem – Caso Particular da Equação do Calor

EDP2:	$f_t - f_{xx} = [x \cos(t) - 4(x - 1) \sin(t)] e^{-2x}$
--------------	---

com $0 \leq x \leq 1$; $0 \leq t \leq 1$; onde $f_{xx} = \frac{\partial^2 f(x, t)}{\partial x^2}$; $f_t = \frac{\partial f(x, t)}{\partial t}$; e

as condições $f(0, t) = 0$; $f(x, 0) = 0$; e $f_x(1, t) + f(1, t) = 0$.

Solução analítica exata:	$f(x, t) = \text{sen}(t) x e^{-2x}$
---------------------------------	-------------------------------------

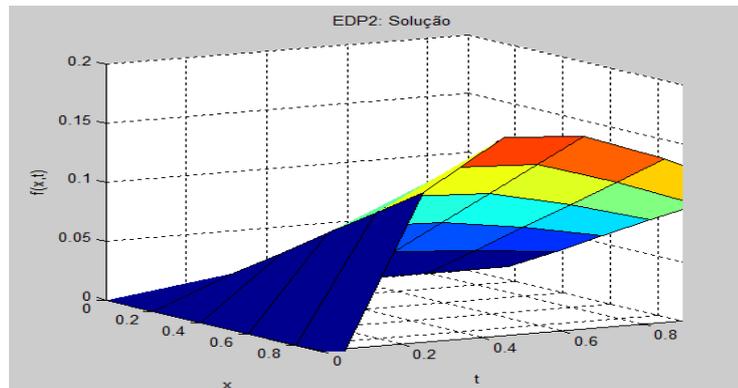
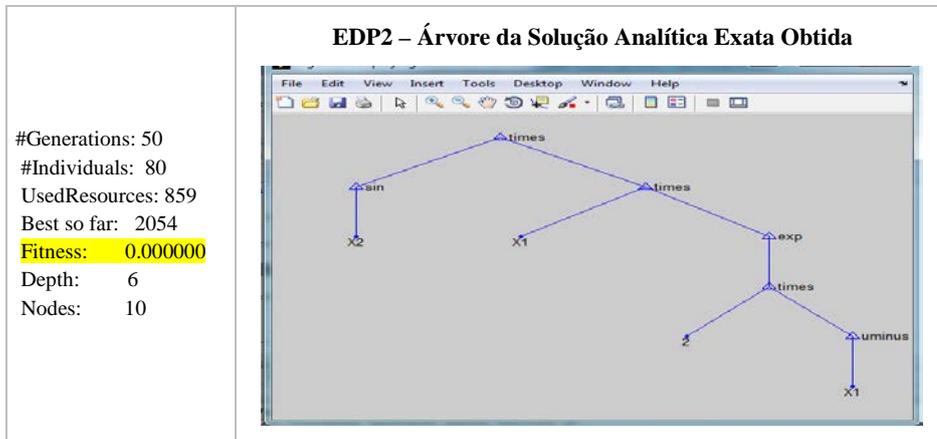


Figura 18: Exemplo 10 – Solução da EDP2

A solução da EDP2 foi obtida no indivíduo 2054 (bestsofar), para uma população de 80 indivíduos e 50 gerações. A solução analítica encontrada é a exata e o fitness é nulo. A árvore de representação obtida é a mais parcimoniosa possível, com profundidade 6 (depth) e 10 nós (nodes).

Exemplo 11: EDP de 2ª ordem – Caso Particular da Equação do Calor

EDP3:	$f_t - f_{xx} = [x - t(4x^3 - 6x)] e^{-x^2}$
--------------	--

com $0 \leq x \leq 1$; $0 \leq t \leq 1$; e condições
 $f(0, t) = 0$; $f(x, 0) = 0$; e $f_x(1, t) + f(1, t) = 0$.

Solução analítica exata:

$$f(x, t) = t x e^{-x^2}$$

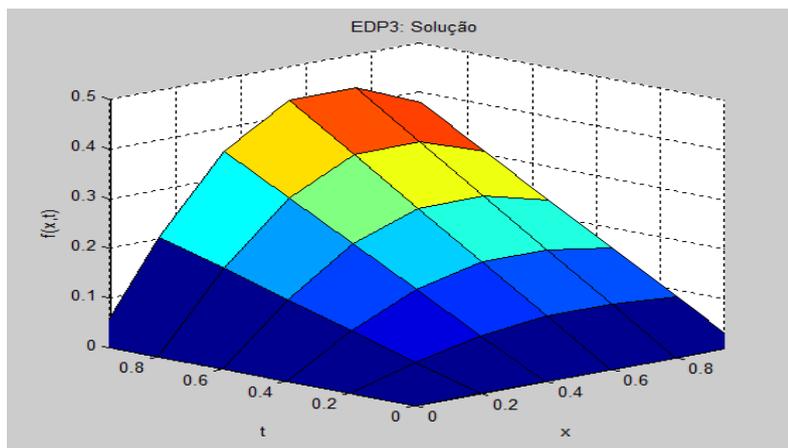
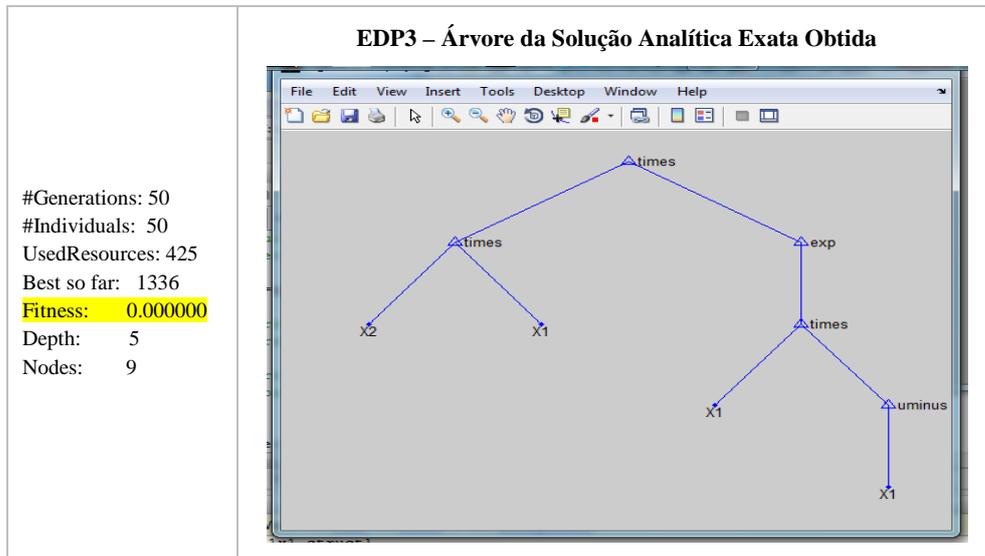


Figura 19: Exemplo 11 – Solução da EDP3

A solução da EDP3 foi obtida no indivíduo 1336 (bestsofar), para uma população de 50 indivíduos e 50 gerações. A solução analítica encontrada é a exata e o fitness é nulo. A árvore de representação obtida é a mais parcimoniosa possível, com profundidade 5 (depth) e 9 nós (nodes).

Deve-se ressaltar que todas as equações diferenciais apresentadas nos exemplos de 1 a 11 foram testadas na sub-rotina *DSolver* do programa *Mathematica* e nas *ODE-PDE-Solver Functions* do programa *Matlab*. As seis primeiras EDOs lineares, dos exemplos 1 a 6, foram solucionadas corretamente e apresentaram solução analítica. No entanto, os programas não apresentam soluções exatas e analíticas para as EDOs e EDPs dos exemplos de 7 a 11, apenas

produzem soluções numéricas. Estes resultados são importantes, pois mostram a potencialidade do método PGDA.

4.1.3

Solução da Equação de Schrödinger para o Átomo de He

4.1.3.1

Introdução ao Problema

A teoria da Mecânica Quântica teve em sua formulação uma grande contribuição do físico austríaco *Erwin Schrödinger*. De acordo com Schrödinger, em decorrência do caráter dual da matéria (onda-partícula), mesmo que uma partícula se mova em uma trajetória definida ela sempre estará distribuída em todo o espaço como uma onda. Assim, uma onda na mecânica quântica equivaleria ao conceito de trajetória na mecânica clássica e seria representada por uma função de onda, ψ . A teoria de Schrödinger é expressa de forma geral pela seguinte equação que leva o seu próprio nome:

$$-\frac{\hbar^2}{2m}\nabla^2\Psi(u,t) + V(r)\Psi(u,t) = i\hbar\frac{\partial\Psi(u,t)}{\partial t}$$

No caso do átomo de He, que possui dois elétrons, conforme mostra a Figura 20, temos a seguinte descrição dos componentes da equação geral:

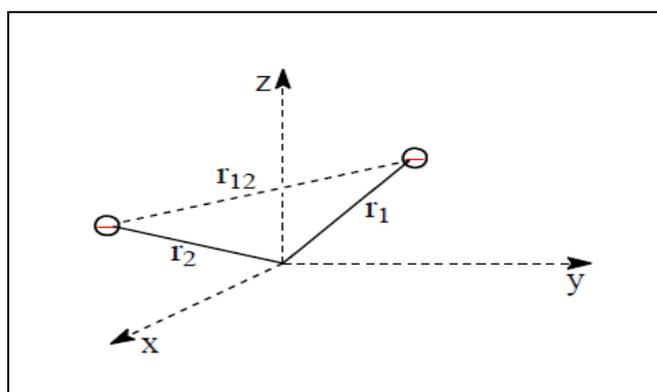


Figura 20: Coordenadas do Átomo de Hélio

Onde,

\hbar : Constante normalizada de “Planck”;

m : massa do elétron;

(x_1, y_1, z_1) e (x_2, y_2, z_2) : coordenadas dos elétrons 1 e 2;

$$u = (x_1, y_1, z_1, x_2, y_2, z_2)$$

$r = (r_1, r_2, r_{12})$: vetor de distâncias;

$r_1 = \sqrt{(x_1^2 + y_1^2 + z_1^2)}$, distância do elétron 1 ao núcleo;

$r_2 = \sqrt{(x_2^2 + y_2^2 + z_2^2)}$, distância do elétron 2 ao núcleo;

$$r_{12} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2};$$

r_{12} : distância do elétron 1 ao elétron 2;

Ψ : função de onda; t : tempo; $i = \sqrt{-1}$: número complexo;

$\nabla^2\Psi = \nabla_1^2\Psi + \nabla_2^2\Psi$: laplaciano total

$\nabla_1^2\Psi = \frac{\partial^2\Psi}{\partial x_1^2} + \frac{\partial^2\Psi}{\partial y_1^2} + \frac{\partial^2\Psi}{\partial z_1^2}$: laplaciano das coord. do elétron 1;

$\nabla_2^2\Psi = \frac{\partial^2\Psi}{\partial x_2^2} + \frac{\partial^2\Psi}{\partial y_2^2} + \frac{\partial^2\Psi}{\partial z_2^2}$: laplaciano das coord. do elétron 2;

$V(r) = \frac{e^2}{4\pi\epsilon_0} \left(-\frac{Z}{r_1} - \frac{Z}{r_2} + \frac{1}{r_{12}} \right)$: energia potencial;

e : carga do elétron; $Z = 2$: número atômico do He; ϵ_0 : permissividade.

Assumindo que as funções são separáveis, tal que

$$\Psi(u, t) = \Psi(u)E^{-iEt/\hbar}$$

chega-se a forma estacionária da equação de Schrödinger, independente do tempo:

$$-\frac{\hbar^2}{2m}\nabla^2\Psi(u) + V(r)\Psi(u) = E\Psi(u)$$

Transformando-se os valores das diversas constantes em unidades atômicas (u.a.), pode-se obter uma forma simplificada:

$$\left(-\frac{1}{2}\right)\nabla^2\Psi(u) + V(r)\Psi(u) = E\Psi(u)$$

onde,

$\Psi = \Psi(u) = \Psi(x_1, y_1, z_1, x_2, y_2, z_2)$, função de onda;

$\nabla^2\Psi = \nabla^2\Psi(u) = \frac{\partial^2\Psi}{\partial x_1^2} + \frac{\partial^2\Psi}{\partial y_1^2} + \frac{\partial^2\Psi}{\partial z_1^2} + \frac{\partial^2\Psi}{\partial x_2^2} + \frac{\partial^2\Psi}{\partial y_2^2} + \frac{\partial^2\Psi}{\partial z_2^2}$, laplaciano total;

$V(r) = \left(-\frac{2}{r_1} - \frac{2}{r_2} + \frac{1}{r_{12}}\right)$: energia potencial em u.a. (Hartree);

E : energia total. Onde, no estado fundamental, $E_0 \cong -2,90372437711 \dots$

Fazendo,

$$H = \left(\frac{-1}{2}\right) \nabla^2 + V(r) \quad (\text{operador Hamiltoniano})$$

e aplicando o operador hamiltoniano à função de onda Ψ , tem-se a conhecida expressão da equação de Schrödinger na forma estacionária:

$$H\Psi = E\Psi$$

Propriedades e Condições de Contorno da Função de Onda Ψ

- i) $\Psi(0) = 0$; a função tende a se anular quando os elétrons tendem ao núcleo.
- ii) $\lim_{u \rightarrow \pm\infty} \Psi(u) = 0$; a função se anula no infinito.
- iii) Ψ é antissimétrica para a troca de elétrons.
- iv) Ψ é contínua e diferenciável, pelo menos de classe C^2 .

Portanto, o problema matemático que se coloca é obter a função de onda ψ que soluciona esta equação diferencial parcial, de segunda ordem, com seis variáveis, onde não é possível aplicar o método de separação de variáveis.

Além da notória dificuldade matemática para a resolução do problema, existem entraves conceituais que aumentam a sua complexidade. Segundo Brito (2012), pelo *princípio de exclusão de Pauli e o princípio da incerteza de Heisenberg* elétrons são indistinguíveis, podendo ser trocados, e assim *torna-se impossível uma solução exata da equação de Schrödinger* para átomos multieletrônicos ou moléculas, devido à repulsão intereletrônica, devendo-se tentar soluções aproximadas. Soluções aproximadas da equação de Schrödinger indicam a energia de um estado particular, por exemplo, para o *estado fundamental* ou para os *estados excitados*, em um átomo como um todo, mas não para um específico elétron.

4.1.3.2

Solução Analítica da Equação de Schrödinger para o Átomo de Hélio e Estimativa da Energia

Nesta subseção são apresentados os resultados obtidos pelo algoritmo PGDA na tentativa de solucionar a equação de Schrödinger para o átomo de hélio. Muitas tentativas foram feitas, mas, infelizmente, não foi possível obter uma solução exata para o problema. Contudo, algumas soluções analíticas aproximadas foram encontradas e estes resultados são reportados a seguir.

Para facilitar a aplicação do algoritmo e produzir representações em árvore mais parcimoniosas, foi necessário realizar uma transformação de coordenadas que reduz o problema de seis para três variáveis.⁴ Após a transformação o problema é agora representado pelas distâncias de cada elétron ao núcleo (r_1 e r_2) e pela distância entre os elétrons (r_{12}), conhecido como termo de repulsão. Aplicando-se a mudança de variáveis, o laplaciano original da equação de Schrödinger pode ser igualado a seguinte expressão:

$$\begin{aligned} \nabla^2 \Psi &= \frac{\partial^2 \Psi}{\partial x_1^2} + \frac{\partial^2 \Psi}{\partial y_1^2} + \frac{\partial^2 \Psi}{\partial z_1^2} + \frac{\partial^2 \Psi}{\partial x_2^2} + \frac{\partial^2 \Psi}{\partial y_2^2} + \frac{\partial^2 \Psi}{\partial z_2^2} = \\ &= \frac{\partial^2 \Psi}{\partial r_1^2} + \frac{2}{r_1} \frac{\partial \Psi}{\partial r_1} + \left(\frac{r_1^2 - r_2^2 + r_{12}^2}{r_1 r_{12}} \right) \frac{\partial^2 \Psi}{\partial r_1 \partial r_{12}} + \frac{\partial^2 \Psi}{\partial r_2^2} + \\ &+ \frac{2}{r_2} \frac{\partial \Psi}{\partial r_2} - \left(\frac{r_2^2 - r_1^2 + r_{12}^2}{r_1 r_{12}} \right) \frac{\partial^2 \Psi}{\partial r_2 \partial r_{12}} + \frac{4}{r_{12}} \frac{\partial \Psi}{\partial r_{12}} + 2 \frac{\partial^2 \Psi}{\partial r_{12}^2}. \end{aligned}$$

Substituindo-se o laplaciano acima na equação original, obtém-se a seguinte expressão para a equação de Schrödinger transformada:

$$\begin{aligned} H\Psi &= \left(\frac{-1}{2} \right) \nabla^2 \Psi + V(r)\Psi = E\Psi \Rightarrow \\ H\Psi &= \left(\frac{-1}{2} \right) \left[\frac{\partial^2 \Psi}{\partial r_1^2} + \frac{2}{r_1} \frac{\partial \Psi}{\partial r_1} + \left(\frac{r_1^2 - r_2^2 + r_{12}^2}{r_1 r_{12}} \right) \frac{\partial^2 \Psi}{\partial r_1 \partial r_{12}} + \frac{\partial^2 \Psi}{\partial r_2^2} \right] + \end{aligned}$$

⁴ Não se trata de coordenadas polares e nem esféricas, o desenvolvimento completo da transformação de coordenadas pode ser visto em (David, C. W., 2009, p. 8.).

$$\left(\frac{-1}{2}\right) \left[\frac{2}{r_2} \frac{\partial \Psi}{\partial r_2} - \left(\frac{r_2^2 - r_1^2 + r_{12}^2}{r_1 r_{12}} \right) \frac{\partial^2 \Psi}{\partial r_2 \partial r_{12}} + \frac{4}{r_{12}} \frac{\partial \Psi}{\partial r_{12}} + 2 \frac{\partial^2 \Psi}{\partial r_{12}^2} \right] +$$

$$\left[-\frac{2}{r_1} - \frac{2}{r_2} + \frac{1}{r_{12}} \right] \Psi = E \Psi$$

Aparentemente o problema parece mais complexo, entretanto, agora se reduziu a três variáveis, r_1 , r_2 e r_{12} .

Esta nova forma da equação de Schrödinger para o He foi implementada no algoritmo PGDA de duas diferentes formas:

i) Na primeira, definiu-se uma medida de erro absoluto pela diferença entre o lado esquerdo e o direito da expressão, calculando-se o seguinte vetor:

$$\text{Erro Absoluto} = EA = \left| \frac{\left(\frac{-1}{2}\right) \nabla^2 \Psi}{\Psi} + V(r) - E \right| = 0$$

Caso se obtenha a verdadeira Ψ , este vetor será nulo, por definição. Contudo, se desconhece o autovalor E . Uma opção sugerida em (BRITO, 2012) e em outros artigos é trabalhar com o valor experimental da energia em seu estado fundamental, $E_0 \cong -2,90372437711 \dots$. Esta sugestão foi aceita e assim tornou-se factível a definição de uma medida de aptidão. Assim, o processo ocorre ao longo das gerações, da seguinte maneira: o algoritmo de PG gera as funções (indivíduos) candidatas a Ψ e a DA realiza os cálculos das diversas derivadas do laplaciano. Em seguida, o vetor de erros absolutos pode ser calculado e os indivíduos avaliados. As medidas de avaliação (*fitness*) utilizadas, não necessariamente juntas, são as seguintes:

- Soma dos Erros absolutos

$$\text{Fit(SAE)} = \sum_{i=1}^n \left| \left\{ \left[\left(\frac{-1}{2} \right) \nabla^2 \Psi(r_i) \right] / \Psi(r_i) \right\} + V(r_i) - E \right|$$

- Erro Absoluto Médio

$$\text{Fit(MAE)} = \frac{1}{n} \sum_{i=1}^n \left| \left[\left(-\frac{1}{2} \right) \nabla^2 \Psi(r_i) / \Psi(r_i) \right] + V(r_i) - E \right|$$

- Raiz do Erro Quadrático Médio

$$\text{Fit(EQM)} = \sqrt{\frac{1}{n} \sum_{i=1}^n \left\{ \left[\left(-\frac{1}{2} \right) \nabla^2 \Psi(r_i) / \Psi(r_i) \right] + V(r_i) - E \right\}^2}$$

ii) Na segunda forma, calcula-se primeiramente a estimativa do vetor de energia. Isto é,

$$\text{Vetor de Energia} = VE = \frac{\left(-\frac{1}{2} \right) \nabla^2 \Psi}{\Psi} + V(r) = E$$

Caso se obtenha a verdadeira Ψ , este vetor será constante com todos os elementos iguais a E . Portanto, o desvio padrão do vetor de energia será nulo. Logo, esta era a medida de avaliação natural a ser usada. Ou seja,

- Desvio Padrão do Vetor de Energia

$$\text{Fit(DPVE)} = \sqrt{\frac{1}{n-1} \sum_{i=1}^n \left\{ \left[\left(-\frac{1}{2} \right) \nabla^2 \Psi(r_i) / \Psi(r_i) \right] + V(r_i) - \bar{E} \right\}^2}$$

Onde, \bar{E} é a média aritmética dos elementos do vetor de energia estimado (VE).

A seguir são apresentados os resultados das soluções analíticas obtidas para a EDP de Schrödinger para o átomo de hélio. Na árvore de representação da solução de todos os casos, as variáveis independentes r_1 , r_2 e r_{12} são representadas pelas variáveis X_1 , X_2 e X_3 .

• **Caso 1: Solução da EDP de Schrödinger para He**

Caso 1 - Solução Analítica Obtida por PGDA

$$\Psi(r_1, r_2, r_{12}) = \text{sen} \left(\frac{1}{\exp[2 r_1 + \exp(r_2) + r_{12}/\exp[\tan(r_1) (r_1 + \exp(r_2))]]} \right) \\ \times \exp[\sqrt{r_2 + 0,5} - \exp(r_{12})] \times \frac{1}{10r_2}$$

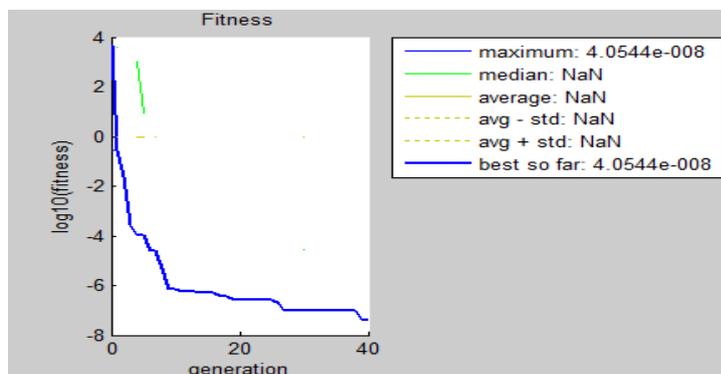
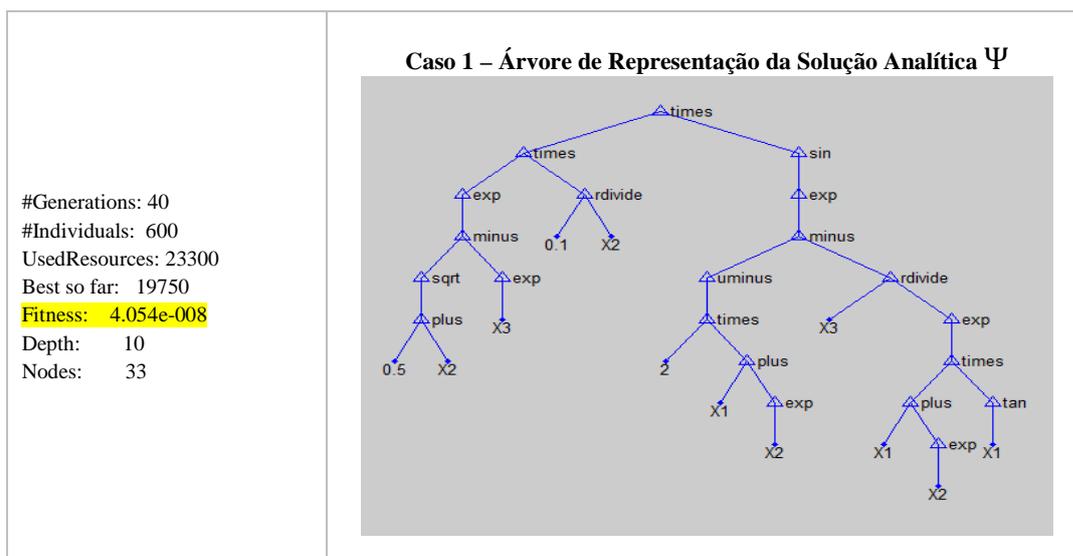


Figura 21: Resultados do Caso 1 da Equação de Schrödinger

A melhor solução da EDP foi obtida no indivíduo 19750 (bestsofar), para uma população de 600 indivíduos e 40 gerações. A solução analítica encontrada é aproximada, com fitness baixíssimo de 0,00000004054. A árvore de representação obtida é pouco parcimoniosa, com profundidade 10 (depth) e 33 nós (nodes).

• **Caso 2: Solução da EDP de Schrödinger para He**

Caso 2 - Solução Analítica Obtida por PGDA

$$\Psi(r_1, r_2, r_{12}) = \frac{0,000135 \log\left(\frac{(\sqrt{2}(r_1 + \sqrt{2}))}{2 r_{12} r_1}\right) \exp(-r_2)}{(r_1 + r_2) r_2 \sqrt{r_1}}$$

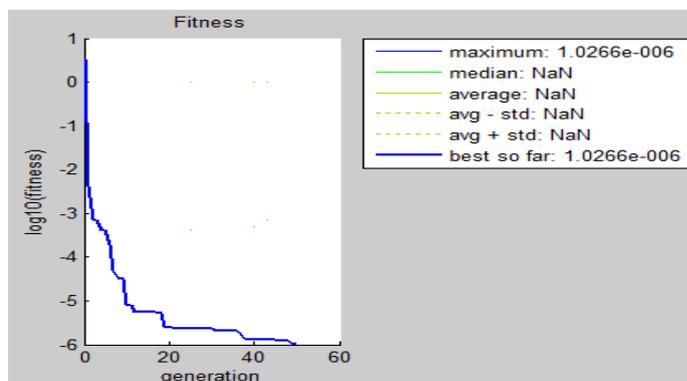
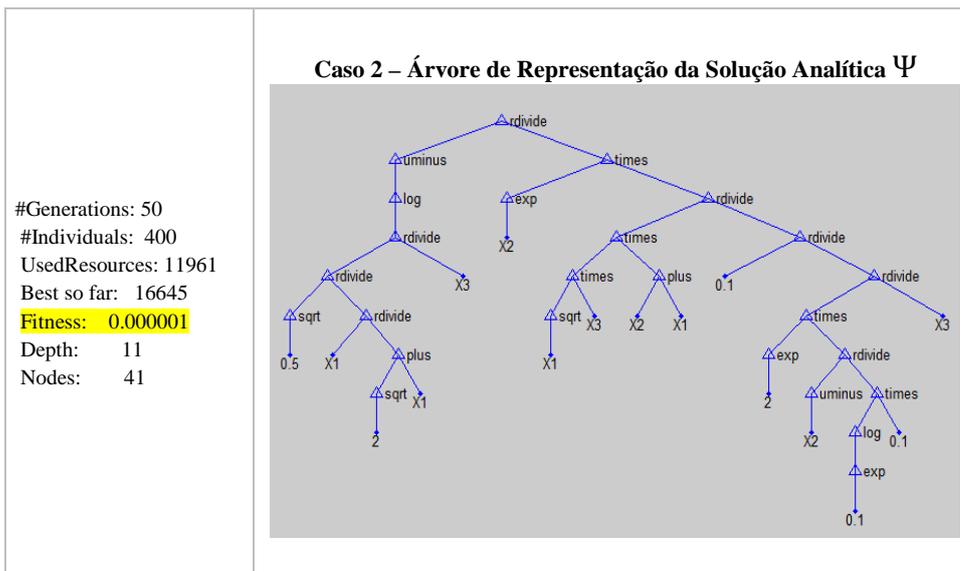


Figura 22: Resultados do Caso 2 da Equação de Schrödinger

A melhor solução da EDP foi obtida no indivíduo 16645 (bestsofar), para uma população de 400 indivíduos e 50 gerações. A solução analítica encontrada é aproximada, com baixo fitness de 0,000010266. A árvore de representação obtida é de grande tamanho, com profundidade 11 (depth) e 41 nós (nodes).

• **Caso 3: Solução da EDP de Schrödinger para He**

Caso 3 - Solução Analítica Obtida por PGDA

$$\Psi(r_1, r_2, r_{12}) = \frac{\text{sen}\left(\frac{1}{10(r_1 + 2)}\right) \exp\left[-\frac{(r_1 + 2,945)(\pi + r_1 + \exp(r_1))}{r_2}\right]}{\exp(r_{12}) + r_1 \exp(r_1)}$$

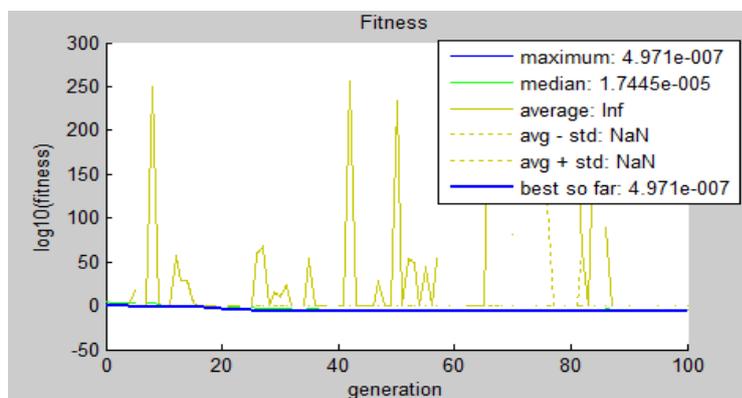
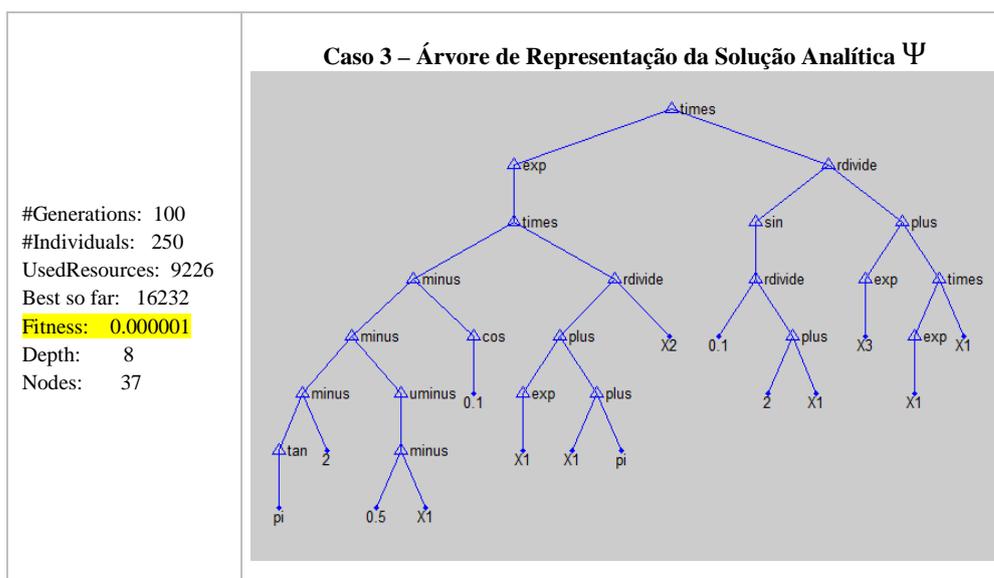


Figura 23: Resultados do Caso 3 da Equação de Schrödinger

A melhor solução encontrada foi obtida com o indivíduo 16232 (bestsofar), com população de 250 indivíduos e 100 gerações. A solução analítica é aproximada e apresentou fitness de 0,000000497. A árvore de representação é pouco parcimoniosa, com profundidade 8 (depth) e 37 nós (nodes).

Embora as três soluções encontradas tenham registrado baixíssimas medidas de erro (os *fitness*), nenhuma delas é uma autofunção da equação de Schrödinger, são apenas boas aproximações analíticas. Além disso, não satisfazem todas as propriedades e condições de contorno exigidas pelo problema. Contudo, foram as que apresentaram as melhores performances e deveriam ser apresentadas.

- **Cálculo da Energia Total para Átomo de He**

Uma das finalidades da obtenção da função de onda (Ψ) que soluciona a EDP de Schrödinger é poder calcular a energia total do sistema E . Uma vez que o sistema esteja em estado estacionário, cada autovalor E define uma autofunção e uma possível solução de equilíbrio para o sistema. É importante entender que ($E=E_n$) é uma sequência de números reais e não uma função contínua, daí nasce o termo *quantum* da Mecânica Quântica. Portanto, não é qualquer número real que pode representá-lo. Por esta razão a estimativa da energia deve ser baseada em algum valor conhecido de E . No caso do átomo de hélio, o único valor de referência conhecido é o valor experimental ($E_0 \cong -2,90372437711\dots$) para a energia em seu estado fundamental. Ou seja, o valor da energia total mínima do sistema, que também é o menor autovalor associado a uma autofunção (Ψ).

Isso implica dizer que, se procuramos uma função de onda que atenda aos requisitos da equação de Schrödinger estacionária através do cálculo da sua energia, então, a função de onda candidata deverá recuperar um valor de energia mínima (E_0) muito próximo do valor experimental acima citado.

Com base nestes princípios, estimamos a energia mínima usando as funções de onda resultantes da solução da equação de Schrödinger por PGDA. Contudo, para nossa surpresa, as três melhores soluções, apresentadas anteriormente, não produziram as melhores estimativas da energia mínima. Outra solução, ainda não reportada, apresentou estimativa bem superior às demais em termos de menor erro relativo médio e baixo desvio padrão. Os resultados podem ser vistos no Quadro 4 mostrado a seguir:

Função de Onda Utilizada na Estimativa da Energia E_0

$$\Psi(r_1, r_2, r_{12}) = (r_1 r_2)^{0,253086} r_{12}^{0,373892} \times \exp[-2,264805 (r_1^{0,879495} + r_2^{0,879495}) + 0,353655 (r_{12}^{-0,655396})]$$

Quadro 4: Estimativa da Energia E_0 para o Átomo de He

Valor da Energia Média Estimada \hat{E}_0	Valor Experimental de E_0	Erro Relativo Médio	Desvio Padrão da Energia
-2,90372437541	-2,90372437711	0,0000001%	0,21676

Este resultado foi obtido através da minimização do desvio padrão do vetor de energia para 3.000 pontos, selecionados aleatoriamente, das coordenadas $(x_1, y_1, z_1, x_2, y_2, z_2)$ dos dois elétrons.

$$\text{Vetor de Energia} = \nabla E = \frac{\left(\frac{-1}{2}\right) \nabla^2 \Psi}{\Psi} + V(r) = E$$

Primeiramente derivou-se o laplaciano da equação de Schrödinger usando a função de onda (ψ) estimada pelo algoritmo de PGDA, com parâmetros genéricos. Em seguida, utilizou-se os 3.000 pontos de $(x_1, y_1, z_1, x_2, y_2, z_2)$ e os parâmetros da função de onda estimada para calcular o vetor de energia inicial. O próximo passo foi então realizar o processo de otimização. Ou seja, obter o vetor de parâmetros da função de onda (Ψ) que minimizasse o desvio padrão do vetor de energia, sujeito a restrição de que a média das energias fosse maior ou igual ao valor da energia mínima experimental $E_0 = -2,903744$.

O resultado da estimativa média de (E_0) apresentou erro relativo quase nulo. No entanto, o desvio padrão ainda é muito elevado para que a função de onda estimada seja uma boa aproximação da verdadeira autofunção (Ψ) associada ao autovalor (E_0) .

4.1.4

Solução de Modelos com Especificação Arbitrária

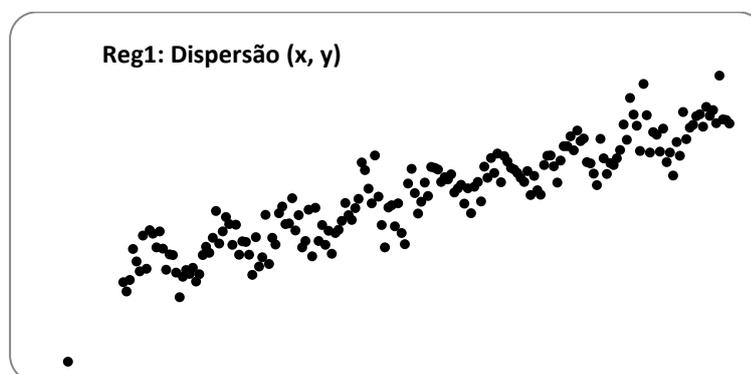
Uma metodologia estatística usualmente utilizada para estimar modelos teóricos concebidos em diversas áreas do conhecimento é denominada de modelos de regressão. Comumente, em virtude do desconhecimento da verdadeira forma funcional que determina a equação comportamental de interesse, os modelos são especificados na forma linear clássica e são estimados pelo conhecido método de mínimos quadrados ordinários. Este procedimento de modelagem, na maioria das vezes não é recomendável e não deve ser aplicado. Pois, as suas hipóteses básicas são muito fortes e quase sempre não são verificadas pelos testes de diagnósticos. Conseqüentemente, a sua indevida utilização pode levar a erros grosseiros de estimação estatística e a análises causais completamente equivocadas.

Esta breve seção apresenta um exemplo de como a metodologia de programação genética com diferenciação automática pode ser usada para auxiliar na solução do problema e ser muito útil a pesquisadores e usuários deste tipo de modelagem.

Suponha que a verdadeira relação estocástica entre as variáveis Y e X seja dada pelo seguinte modelo linear:

Modelo Verdadeiro:	$Y_t = 5 + 0,2 X_t + 1,2 \text{ sen}(X_t) + \varepsilon_t ; \quad \varepsilon_t \sim N(0, \sigma^2)$
---------------------------	--

Simulando-se 200 observações para este modelo, tem-se o seguinte gráfico de dispersão dos dados:



Sem o conhecimento do verdadeiro modelo, o gráfico de dispersão sugere ao analista uma especificação linear simples. Procedimento este, usualmente adotado em modelagem estatística de regressão. A partir desta avaliação visual, o analista menos experiente define e arbitra o seguinte modelo:

$$Y_t = \alpha + \beta X_t + \varepsilon_t; \quad \varepsilon_t \sim N(0, \sigma^2)$$

Então, uma vez estimados os coeficientes α e β o problema estará resolvido e a sua análise poderá ser realizada. Para a estimação estatística dos parâmetros, geralmente é utilizado o conhecido método de mínimos quadrados ordinários, que consiste na minimização da soma de quadrados dos resíduos do modelo, ou seja:

$$\min_{\hat{\alpha}, \hat{\beta}} \sum_{t=1}^n (Y_t - \hat{\alpha} - \hat{\beta} X_t)^2$$

Isto é, obter $\hat{\alpha}$ e $\hat{\beta}$ que minimizem a soma de quadrados dos resíduos da estimação. Este problema de otimização é resolvido calculando-se as derivadas parciais da expressão acima em relação à $\hat{\alpha}$ e $\hat{\beta}$ e igualando-as a zero, com a restrição de que a matriz de segundas derivadas seja positiva definida.

Este método está implementado em programas estatísticos e em planilhas de cálculos. Abaixo são apresentados os resultados da estimação de mínimos quadrados realizadas pelo programa estatístico EViews.

Modelo Linear Estimado:	$\hat{Y} = 5,18 + 0,195 X$
--------------------------------	----------------------------

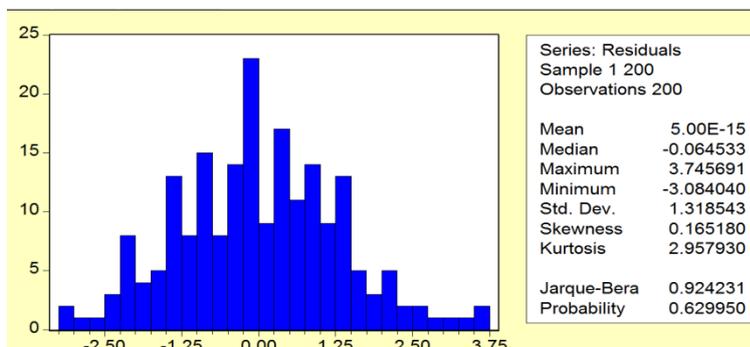
Dependent Variable: Y
Method: Least Squares
Sample: 1 200
Included observations: 200

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	5.184600	0.210017	24.68655	0.0000
X	0.195031	0.005397	36.13982	0.0000
R-squared	0.868359	Mean dependent var		11.98144
Adjusted R-squared	0.867694	S.D. dependent var		3.634111
S.E. of regression	1.321869	Akaike info criterion		3.405919
Sum squared resid	345.9726	Schwarz criterion		3.438903
F-statistic	1306.087	Prob(F-statistic)		0.000000

Os resultados do teste F indicam que o modelo estimado é altamente significativo, com nível de significância inferior a 1%. Os resultados das medidas R^2 e R^2 -adjusted corroboram com o teste e apresentam boa qualidade de ajuste. O teste Jarque-Bera de normalidade dos resíduos não rejeita esta hipótese mesmo

com níveis de significância bastante elevados. Portanto, pela ótica da análise estatística de regressão o modelo estimado estaria aprovado e pronto para servir a uma eventual análise de causalidade entre as variáveis.

Teste de Normalidade dos Resíduos



No entanto, o modelo foi estimado com a arbitrariedade e imposição de uma forma funcional por falta de opção metodológica. Teoricamente, o modelo deve ser especificado de forma geral, ou seja:

$$Y_t = f(X_t; \theta) + \varepsilon_t; \quad \varepsilon_t \sim N(0, \sigma^2),$$

onde f é uma função desconhecida e θ um vetor de parâmetros de valores e dimensão, ambos desconhecidos. O problema agora é obter as estimativas \hat{f} e $\hat{\theta}$ que minimizem a soma de quadrados de resíduos. Isto é,

$$\min_{(\hat{f}, \hat{\theta})} \sum_{t=1}^n (Y_t - \hat{f}(X_t; \hat{\theta}))^2 = \min_{(\hat{f}, \hat{\theta})} \sum_{t=1}^n e_t^2$$

A solução é dada pela resolução da seguinte equação diferencial parcial:

$$\frac{\partial \sum_{t=1}^n e_t^2}{\partial \hat{\theta}} = 0.$$

Nesta situação o algoritmo de PGDA pode auxiliar na solução do problema, obtendo a \hat{f} e $\hat{\theta}$ que solucionam a EDP acima. De fato, seguem os resultados:

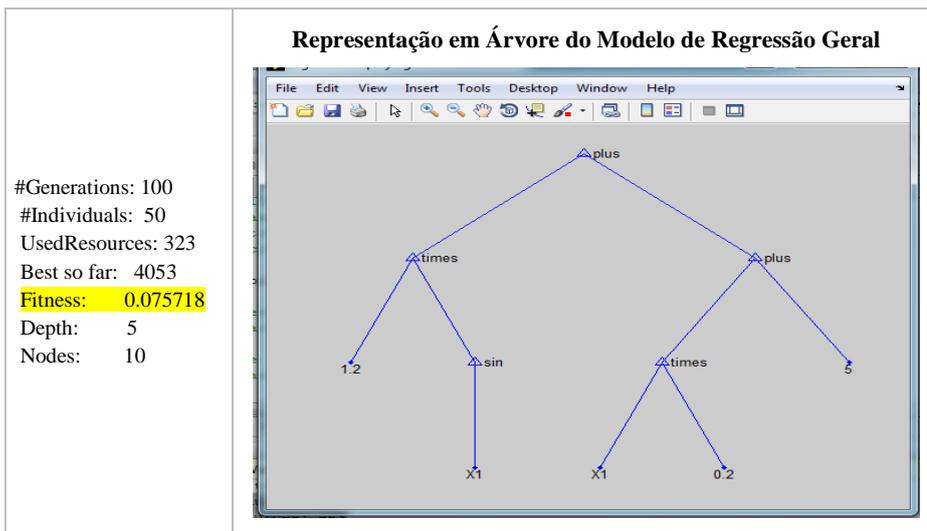


Figura 24: Resultados da Estimação do Modelo por PGDA

Modelo Especificado por PGDA:	$Y_{PG} = 5 + 0,2 X + 1,2 \text{sen}(X)$
--------------------------------------	--

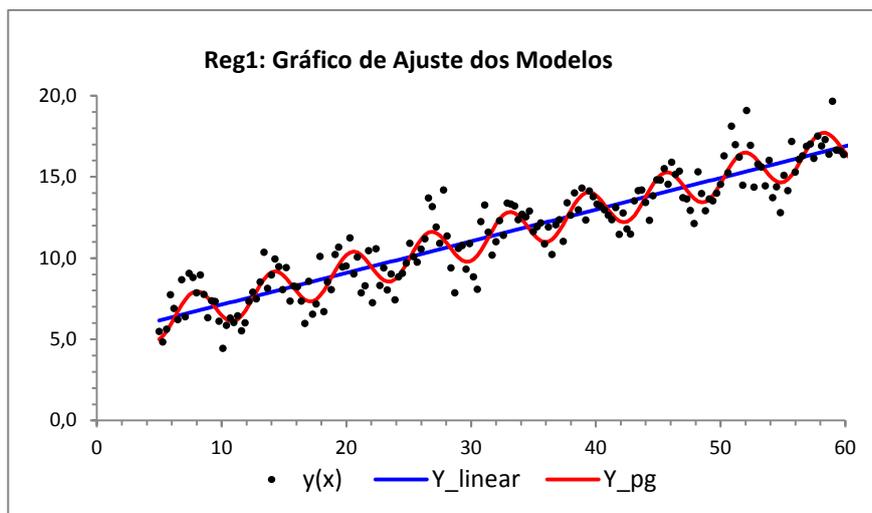


Figura 25: Comparação entre os Ajustes dos Modelos Estimados

Observe que o modelo simulado foi recuperado na forma idêntica como foi concebido, com a mesma forma funcional e os mesmos parâmetros. A melhor solução encontrada foi obtida com o indivíduo 4053 (bestsofar), com população de 50 indivíduos e 100 gerações. A solução é aproximada, pois o modelo original é estocástico. A árvore de representação é a mais parcimoniosa possível, com profundidade 5 (depth) e 10 nós (nodes).

O gráfico acima mostra claramente a eficiência do método PGDA que capturou o verdadeiro modelo na íntegra. Foram realizados os mesmos testes estatísticos realizados para o modelo linear simples apresentado na primeira parte do exemplo, e os resultados confirmam a significância do modelo estimado pela metodologia PGDA.

Diante do resultado obtido, conclui-se este exercício com a convicção de que a PGDA é uma metodologia alternativa para ser usada na especificação e estimação de modelos de regressão.

4.2

Solução de Equações Diferenciais Estocásticas por PGDA

Esta seção tem a finalidade de apresentar um algoritmo computacional, construído a partir das técnicas de programação genética e diferenciação automática, que é uma nova alternativa para solucionar equações diferenciais estocásticas (EDEs). O algoritmo foi desenvolvido com base nas premissas teóricas do cálculo estocástico de Itô e utiliza a PGDA para obter soluções analíticas para os problemas de EDEs. Dada à complexidade do tema, as equações tratadas no texto se limitam a EDEs clássicas, univariadas, e que satisfazem aos teoremas de existência e unicidade de solução. A seção apresenta-se dividida em duas subseções, onde a primeira descreve a estrutura de funcionamento do algoritmo e apresenta exemplos de soluções de EDEs tradicionalmente utilizadas na modelagem de ativos financeiros. Na segunda subseção, utilizando-se o algoritmo de PGDA para determinar a dinâmica dos preços do ativo-objeto, um modelo de precificação de opções europeias é proposto e algumas aplicações são realizadas para o mercado brasileiro, com ações da Petrobras e da Vale.

4.2.1

Algoritmo de PGDA para Solucionar EDEs

O algoritmo apresentado nesta seção foi construído com a mesma concepção metodológica e estrutura geral de funcionamento descrita no Capítulo 3. Utiliza-se do cálculo estocástico de Itô, visto anteriormente, para gerar soluções analíticas de

EDEs representadas na forma tradicional, abaixo apresentada. No entanto, com algumas modificações e adaptações algoritmos similares poderão ser elaborados para modelar outros tipos de EDEs.

$$dX(t) = a(t, X(t)) dt + b(t, X(t)) dW(t); \quad t \in [0, T] \quad (1)$$

Embora a estrutura geral de funcionamento do algoritmo já seja conhecida, faz-se necessário descrever os detalhes técnicos e procedimentos utilizados na construção da solução de uma EDE. O processo de modelagem é realizado em duas etapas que se descreve a seguir:

i) A primeira etapa consiste na especificação da EDE de interesse que, diante de um problema real de modelagem, defronta-se com duas situações distintas. A primeira, que facilita consideravelmente o processo de solução, ocorre quando se conhece a priori a forma funcional e as demais propriedades matemáticas das funções que definem os coeficientes infinitesimais $a(t, X(t))$ e $b(t, X(t))$. A segunda situação, comumente verificada na prática, é quando não se tem nenhuma informação sobre o modelo de EDE e apenas se conhece uma específica trajetória do processo estocástico $\{X(t)\}$.

Portanto, com a finalidade de oferecer uma metodologia de especificação menos arbitrária, baseada somente nos dados observados e na teoria de cálculo estocástico, o algoritmo foi construído a partir desta segunda situação, mas também permite que a modelagem seja realizada partindo-se de uma especificação conhecida.

Com esse intuito foram elaborados dois codes matlab que funcionam de forma simultânea, onde o primeiro executa a PG, gerando possíveis especificações para a EDE (indivíduos), e o segundo, utilizando-se da DA para a construção das medidas de avaliação, avalia as aptidões (*fitness*) e seleciona a especificação de melhor desempenho. Em virtude da técnica de diferenciação usual não poder ser aplicada diretamente na equação de interesse e dada a impossibilidade de se trabalhar com dados efetivamente em tempo contínuo, os codes trabalham com dados e equações discretizadas em tempo contínuo, obtidas através do método de Euler ou Euler-Maruyama, ver (Pederson, 1995; Nicolau, 2000; Ait-Sahalia, 2002), e buscam soluções analíticas exatas que satisfaçam ao teorema de Itô. O processo de especificação segue os seguintes passos: inicialmente, considera-se a EDE de interesse na sua forma discretizada:

$$X_{t_i} - X_{t_{i-1}} = a(t_{i-1}, X_{t_{i-1}}) \Delta + b(t_{i-1}, X_{t_{i-1}}) (W_{t_i} - W_{t_{i-1}}) \quad (7)$$

Onde, $0 = t_0 < t_1 < \dots < t_n$

= T é uma partição de $[0, T]$; e o intervalo detempo, Δ

= $(t_i - t_{i-1}) = \frac{T}{n}$, é suposto constante em toda a partição.

Em seguida, com base nos pressupostos e exigências do teorema de Itô, os codes realizam a escolha da transformação $Y_t = U(X_t)$, de maneira que se tenha a seguinte forma discretizada para a EDE transformada:

$$Y_{t_i} - Y_{t_{i-1}} = \tilde{a}(t_{i-1}, X_{t_{i-1}}) \Delta + \tilde{b}(t_{i-1}, X_{t_{i-1}}) (W_{t_i} - W_{t_{i-1}}) \quad (8)$$

onde, pela fórmula de Itô,

$$\begin{aligned} \tilde{a}(t_i, X_{t_i}) &= \frac{\partial U(t_i, X_{t_i})}{\partial t_i} + \frac{\partial U(t_i, X_{t_i})}{\partial X_{t_i}} a(t_i, X_{t_i}) + \frac{1}{2} \times \frac{\partial^2 U(t_i, X_{t_i})}{\partial X_{t_i}^2} b^2(t_i, X_{t_i}) \text{ e} \\ \tilde{b}(t_i, X_{t_i}) &= \frac{\partial U(t_i, X_{t_i})}{\partial X_{t_i}} b(t_i, X_{t_i}). \end{aligned}$$

Observe que, os coeficientes (\tilde{a} e \tilde{b}) e a trajetória do processo de Wiener (W_{t_i}) da equação (8) não são conhecidos. No entanto, decorre das propriedades deste processo o seguinte resultado:

$$(W_{t_i} - W_{t_{i-1}}) \sim \text{Normal}(0, \Delta) \Leftrightarrow \frac{(W_{t_i} - W_{t_{i-1}})}{\sqrt{\Delta}} = \varepsilon_{t_i} \sim \text{Normal}(0, 1).$$

Substituindo-se esta relação na equação (8), obtém-se que

$$\frac{Y_{t_i} - Y_{t_{i-1}} - \tilde{a}(t_{i-1}, X_{t_{i-1}}) \Delta}{\tilde{b}(t_{i-1}, X_{t_{i-1}}) \sqrt{\Delta}} = \varepsilon_{t_i} \sim \text{Normal}(0, 1), \quad \forall t_i. \quad (9)$$

Este resultado é de fundamental importância, pois permite que os coeficientes infinitesimais (\tilde{a} e \tilde{b}) sejam estimados pelo método de máxima verossimilhança, através maximização da função de log-verossimilhança de ruídos normais padrão, independentes, (ε_i). Ou seja,

$$(\hat{\tilde{a}}, \hat{\tilde{b}}) = \underset{(\tilde{a}, \tilde{b})}{\text{argmax}} \{ \mathcal{L}(\tilde{a}, \tilde{b} \mid t, Y, X) \},$$

onde

$$\mathcal{L}(\tilde{a}, \tilde{b} \mid t, Y, X) = -\frac{n \log(2\pi)}{2} - \sum_{i=1}^n \frac{(Y_{t_i} - Y_{t_{i-1}} - \tilde{a}(t_{i-1}, X_{t_{i-1}}) \Delta)^2}{2 \tilde{b}(t_{i-1}, X_{t_{i-1}})^2 \Delta},$$

sujeito às restrições:

$$\tilde{a}(t_i, X_{t_i}) = \frac{\partial U(t_i, X_{t_i})}{\partial t_i} + \frac{\partial U(t_i, X_{t_i})}{\partial X_{t_i}} a(t_i, X_{t_i}) + \frac{1}{2} \times \frac{\partial^2 U(t_i, X_{t_i})}{\partial X_{t_i}^2} b^2(t_i, X_{t_i}) e$$

$$\tilde{b}(t_i, X_{t_i}) = \frac{\partial U(t_i, X_{t_i})}{\partial X_{t_i}} b(t_i, X_{t_i}).$$

Substituindo-se as restrições nos lugares de (\tilde{a} e \tilde{b}), tem-se um novo problema de maximização, onde os objetos da otimização são agora os coeficientes infinitesimais da EDE original. Isto é,

$$(\hat{a}, \hat{b}) = \underset{(a,b)}{\operatorname{argmax}} \{ \mathcal{L}(a, b \mid t, Y, X) \},$$

onde,

$$\mathcal{L}(a, b \mid t, Y, X) = -\frac{n \log(2\pi)}{2} - \frac{\sum_{i=1}^n \left[Y_{t_i} - Y_{t_{i-1}} - \left(\frac{\partial U(t_{i-1}, X_{t_{i-1}})}{\partial t_{i-1}} + \frac{\partial U(t_{i-1}, X_{t_{i-1}})}{\partial X_{t_{i-1}}} a(t_{i-1}, X_{t_{i-1}}) + \frac{\partial^2 U(t_{i-1}, X_{t_{i-1}})}{2 \partial X_{t_{i-1}}^2} b^2(t_{i-1}, X_{t_{i-1}}) \right) \Delta \right]^2}{2 \left[\frac{\partial U(t_i, X_{t_i})}{\partial X_{t_i}} b(t_i, X_{t_i}) \right]^2 \Delta}.$$

Deve-se ressaltar que a estimação de máxima verossimilhança⁵ realizada pelos codes de PGDA é uma valiosa ferramenta para a modelagem de equações estocásticas, pois possibilita que equações com especificações parcialmente ou totalmente desconhecidas sejam estimadas criteriosamente. Como é caso acima, onde a transformação $Y_t = U(X_t)$ e os coeficientes infinitesimais $a(t, X_t)$ e $b(t, X_t)$ são funções contínuas totalmente desconhecidas. Além disso, o uso do método de máxima verossimilhança garante à metodologia de estimação importantes propriedades estatísticas assintóticas, como consistência, eficiência e normalidade.

$$\frac{Y_{t_i} - Y_{t_{i-1}} - \hat{a}(t_{i-1}, X_{t_{i-1}}) \Delta}{\hat{b}(t_{i-1}, X_{t_{i-1}}) \sqrt{\Delta}} = \hat{\varepsilon}_{t_i} \sim \text{Normal}(0, 1), \quad \forall t_i. \quad (10)$$

Outra restrição que se coloca ao processo de estimação é que a hipótese de normalidade, definida na expressão (9), deve ser atendida. Ou seja, os ruídos estimados, de acordo com a expressão (10), devem apresentar distribuição normal padrão, sob pena de invalidação das estimativas obtidas. Para garantir que esta hipótese seja cumprida, o code de (*fitness*) exige que as seguintes medidas de avaliação sejam satisfeitas:

⁵ Para maiores informações sobre estimadores de máxima verossimilhança para EDEs, veja (Kloden, 1992; Hamilton, 1994; Campbell, 1997).

$$\bar{\varepsilon} = \frac{1}{n} \sum_{i=1}^n \hat{\varepsilon}_{t_i} = 0; \text{ (m\u00e9dia amostral)}$$

$$S_{\varepsilon} = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (\hat{\varepsilon}_{t_i} - \bar{\varepsilon})^2} = 1; \text{ (desvio padr\u00e3o amostral)}$$

$$JB = \frac{n}{6} \left(A^2 + \frac{(K-3)^2}{4} \right) < \chi_{\text{tab}}^2(5\%; 2\text{gl}) = 5,9915. \text{ (teste de normalidade)}$$

A estat\u00edstica JB refere-se ao teste de normalidade de Jarque-Bera⁶, realizado ao n\u00edvel de signific\u00e2ncia de 5%. O teste indica que a hip\u00f3tese nula de normalidade de (ε_{t_i}) n\u00e3o ser\u00e1 rejeitada, ao n\u00edvel de 5%, caso a estat\u00edstica calculada JB seja inferior ao valor tabelado de 5,9915, obtido de uma distribui\u00e7\u00e3o qui-quadrado, com n\u00edvel de signific\u00e2ncia de 5% e 2 graus de liberdade. As estat\u00edsticas A e K s\u00e3o, respectivamente, as estimativas amostrais dos coeficientes de assimetria e curtose dos ru\u00eddos estimados.

Uma vez que as exig\u00eancias acima sejam atendidas, e sabendo-se das propriedades do processo de Wiener que $(W_{t_i} - W_{t_{i-1}}) \sim \text{Normal}(0, \Delta)$, estimativas consistentes da trajet\u00f3ria deste processo podem ser obtidas a partir da trajet\u00f3ria estimada para o processo de ru\u00eddo branco. Isto \u00e9, dado que $(W_{t_i} - W_{t_{i-1}}) = \sqrt{\Delta} \varepsilon_{t_i}$, implica na forma recursiva $\hat{W}_{t_i} = \sqrt{\Delta} \hat{\varepsilon}_{t_i} + \hat{W}_{t_{i-1}}$. Ou, dado que $(W_0 = 0)$, substituindo-se na forma recursiva tem-se a estimativa do processo como soma de ru\u00eddos (erros), que \u00e9 uma forma cl\u00e1ssica de representa\u00e7\u00e3o de W_{t_i} :

$$\hat{W}_{t_i} = \sqrt{\Delta} \sum_{k=1}^i \hat{\varepsilon}_{t_k}$$

Uma vez escolhida a transforma\u00e7\u00e3o $Y_t = U(X_t)$ e estimada a trajet\u00f3ria de W_{t_i} , os coeficientes infinitesimais $(\tilde{a}$ e $\tilde{b})$ s\u00e3o reestimados para confirmar e revisar as estimativas iniciais, usando-se, agora, a forma da equa\u00e7\u00e3o discretizada (8). Com isso, todos os componentes das EDEs original e transformada estar\u00e3o identificados e a etapa de especifica\u00e7\u00e3o estar\u00e1 conclu\u00edda com as seguintes equa\u00e7\u00f5es estimadas:

⁶ Para maiores detalhes sobre o teste de normalidade de Jarque-Bera, indica-se (Bera, 1981; Johnston, 1997; Morettin, 2004). No entanto, este \u00e9 um teste cl\u00e1ssico e pode ser encontrado na maioria dos livros texto de Econometria e S\u00e9ries Temporais.

$$X_{t_i} - X_{t_{i-1}} = \hat{a}(t_{i-1}, X_{t_{i-1}}) \Delta + \hat{b}(t_{i-1}, X_{t_{i-1}}) (\hat{W}_{t_i} - \hat{W}_{t_{i-1}}) \quad (11)$$

$$Y_{t_i} - Y_{t_{i-1}} = \hat{\tilde{a}}(t_{i-1}, X_{t_{i-1}}) \Delta + \hat{\tilde{b}}(t_{i-1}, X_{t_{i-1}}) (\hat{W}_{t_i} - \hat{W}_{t_{i-1}}) \quad (12)$$

ii) A segunda etapa consiste em obter a solução analítica da EDE original (1), através da solução da EDE transformada pela fórmula de Itô, utilizando-se a transformação $Y_t = U(X_t)$ e os coeficientes infinitesimais estimados na etapa anterior. Ou seja, o problema agora é obter a solução analítica da seguinte EDE:

$$dY(t) = \hat{\tilde{a}}(t, X(t)) dt + \hat{\tilde{b}}(t, X(t)) dW_t.$$

A solução do problema é obtida pela resolução da integral estocástica

$$\int_0^t dY(s) = \int_0^t \hat{\tilde{a}}(s, X(s)) ds + \int_0^t \hat{\tilde{b}}(s, X(s)) dW(s).$$

A solução da integral do lado esquerdo, dado que $Y(t)$ é uma função contínua e diferenciável, é simples e dada pela expressão:

$$\int_0^t dY(s) = Y(t) - Y(0) = U(X(t)) - U(X(0)).$$

Onde $X(0) = X_0$ é a condição inicial, supostamente, uma constante conhecida.

As soluções das integrais estocásticas do lado direito, dada a complexidade do cálculo, principalmente o da integral de Itô, são obtidas em forma analítica, mas de maneira que satisfaçam às seguintes aproximações:

$$\int_0^{t_i} \hat{\tilde{a}}(s, X(s)) ds \cong \sum_{s=1}^i \frac{\hat{\tilde{a}}(t_s, X_{t_s}) + \hat{\tilde{a}}(t_{s-1}, X_{t_{s-1}})}{2} \Delta ;$$

$$\int_0^{t_i} \hat{\tilde{b}}(s, X(s)) dW(s) \cong \sum_{s=1}^i \hat{\tilde{b}}(t_{s-1}, X_{t_{s-1}}) (\hat{W}_{t_s} - \hat{W}_{t_{s-1}}).$$

Logo, a solução aproximada para $Y(t)$ é dada na forma discretizada por:

$$Y_{t_i} \cong Y_0 + \sum_{s=1}^i \frac{\hat{\tilde{a}}(t_s, X_{t_s}) + \hat{\tilde{a}}(t_{s-1}, X_{t_{s-1}})}{2} \Delta + \sum_{s=1}^i \hat{\tilde{b}}(t_{s-1}, X_{t_{s-1}}) (\hat{W}_{t_s} - \hat{W}_{t_{s-1}}).$$

No entanto, a solução final gerada pelo algoritmo é analítica e em tempo contínuo,

$$Y(t) = Y(0) + f(t, X(t), \hat{W}_t),$$

onde $f(t, X(t), \hat{W}_t)$ é uma função contínua que determina e aproxima a solução dos somatórios acima, de tal forma que se obtenha uma solução exata para $Y(t)$. Em

seguida, a solução para $X(t)$ é obtida através da função inversa de $Y(t) = U(X(t))$, conforme mostra o cálculo abaixo:

$$X(t) = U^{-1}(Y(t)) = U^{-1}(Y(0) + f(t, X(t), \widehat{W}_t)) = h(t, X(s), \widehat{W}_t, X(0)).$$

Portanto, a solução exata da EDE é dada por:

$$X(t) = h(t, X(s), \widehat{W}_t, X(0)).$$

#

Com a finalidade de testar a metodologia acima descrita e de ilustrar o algoritmo de solução, foram realizados alguns experimentos para modelos de EDEs utilizados na modelagem de preços e retornos de ativos financeiros. Os exemplos são apresentados a seguir.

4.2.2

Solução da EDE de MBG por PGDA - Exemplo 1

Neste primeiro exemplo apresenta-se um exercício completo, com dados simulados, que mostra todas as etapas da solução de uma EDE por PGDA. O modelo escolhido para a simulação foi o Movimento Browniano Geométrico (MBG), pela sua grande importância na modelagem de séries financeiras. Este modelo já foi apresentado na seção 2.4.3 e é especificado pela EDE (6), abaixo.

$$dX(t) = \mu X(t) dt + \sigma X(t) dW(t) \quad (6)$$

Onde, relacionando-se com a forma tradicional de EDE, tem-se:

$$a(t, X(t)) = \mu X(t), \quad b(t, X(t)) = \sigma X(t), \quad \text{tal que } \sigma > 0 \text{ e } X(t) > 0 \quad \forall t \in [0, T].$$

Embora a solução desta equação já seja conhecida, desenvolve-se a sua solução, em rápidos passos, para facilitar o entendimento da aplicação que será realizada posteriormente.

Então, escolhendo adequadamente $Y(t) = \text{Log}(X(t))$ e aplicando-se a fórmula de Itô, chega-se a seguinte EDE transformada:

$$dY(t) = \left(\mu - \frac{\sigma^2}{2} \right) dt + \sigma dW(t).$$

Onde,

$$\tilde{a}(t, X(t)) = \left(\mu - \frac{\sigma^2}{2} \right) \text{ e } \tilde{b}(t, X(t)) = \sigma.$$

Integrando $dY(t)$, no intervalo $[0, t]$, determina-se a solução para $Y(t)$:

$$Y(t) = Y(0) + \left(\mu - \frac{\sigma^2}{2} \right) t + \sigma W(t).$$

Agora, aplicando-se a função inversa ($Y^{-1}(t) = \text{Log}^{-1}(X(t)) = e^{Y(t)}$), obtém-se a solução final para $X(t)$,

$$X(t) = X(0) \exp \left\{ \left(\mu - \frac{\sigma^2}{2} \right) t + \sigma W(t) \right\},$$

onde $X(0) = e^{Y(0)}$ é a condição inicial do problema.

Portanto, a priori, já se conhece a solução analítica da EDE do MBG. Contudo, o exercício é realizado com a finalidade de oferecer uma alternativa de modelagem que seja útil para tratar o problema na forma crua como ele se apresenta na prática. Ou seja, sem nenhum conhecimento sobre a especificação da EDE que representa a sua dinâmica. Geralmente, o que se conhece, são apenas os dados observados de uma única trajetória do processo de interesse $X(t)$.

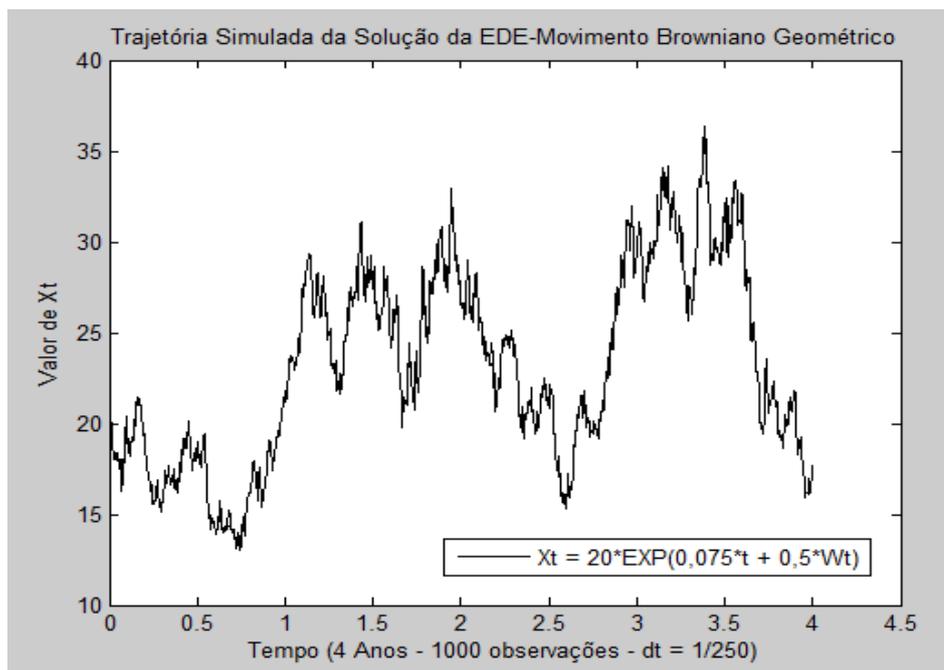


Figura 26: Trajetória do Processo Simulado de MBG

Para a realização do exercício os dados foram simulados por code Matlab, através da função “simBySolution”,⁷ aplicada especificamente ao modelo de

⁷ Os codes Matlab utilizados nas simulações podem ser vistos na seção 7.1 do apêndice.

MBG. No processo de simulação foram definidos e utilizados os seguintes parâmetros para a EDE geradora dos dados:

- i) Modelo de MBG, EDE simulada: $dX(t) = 0,2 X(t) dt + 0,5 X(t) dW(t)$
com, condição inicial $X(0) = 20$; $\mu = 0,2$ e $\sigma = 0,5$;
- ii) Número de períodos ou observações: $n = 1000$;
- iii) Variação infinitesimal do tempo: $dt = \Delta = 1/250 = 0,004$; (constante)
- iv) Intervalo de tempo considerado: $[0, T] = [0, n \cdot dt] = [0, 4]$. (quatro anos de dias úteis, aproximadamente)

Os dados simulados para $X(t)$ são apresentados no gráfico da Figura 26.

Etapa1: Especificação da EDE

Uma vez os dados conhecidos, a primeira etapa da solução da EDE por PGDA é a identificação e estimação dos componentes que especificam a equação, obtendo-se uma solução que satisfaça à expressão (10).

$$\frac{Y_{t_i} - Y_{t_{i-1}} - \hat{a}(t_{i-1}, X_{t_{i-1}}) \Delta}{\hat{b}(t_{i-1}, X_{t_{i-1}}) \sqrt{\Delta}} = \hat{\varepsilon}_{t_i} \sim \text{Normal}(0, 1), \quad \forall t_i. \quad (10)$$

Isto é, identificar e estimar a função $Y(t) = U(X(t))$ e os coeficientes (\hat{a} e \hat{b}), com $\Delta = 0,004$, de tal forma que se tenha, pela razão, uma trajetória do ruído branco normal padrão. Com esta finalidade, dois codes de PGDA foram elaborados e executados.⁸ Os resultados são apresentados a seguir:

O resultado do code de PGDA, apresentado na árvore da Figura 27, indica que a melhor solução para estimar a trajetória do processo de ruído branco normal padrão é dada pela expressão:

$$\hat{\varepsilon}_{t_i} = \log \left[\left(\frac{X_1}{1,000288 X_2} \right)^{31,4971} \right] = \log \left[\left(\frac{X_{t_i}}{1,000288 X_{t_{i-1}}} \right)^{31,4971} \right].$$

Onde, (Power = potenciação), ($X_1 = X_{t_i}$) e ($X_2 = X_{t_{i-1}}$). Desenvolvendo esta expressão e igualando o resultado à equação (10), tem-se que:

⁸ Os dois codes elaborados para esta etapa de especificação da EDE são denominados de PGDA_ESPCEDE e FITNESS_ESPCEDE. O primeiro realiza o programa de PG e o segundo avalia e seleciona o melhor indivíduo ou solução.

$$\hat{\varepsilon}_{t_i} = 31,4971[\log(X_{t_i}) - \log(X_{t_{i-1}}) - \log(1.000288)] \Leftrightarrow$$

$$\hat{\varepsilon}_{t_i} = \frac{[\log(X_{t_i}) - \log(X_{t_{i-1}}) - 0.000288]}{0,031749} = \frac{Y_{t_i} - Y_{t_{i-1}} - \hat{a}(t_{i-1}, X_{t_{i-1}}) \Delta}{\hat{b}(t_{i-1}, X_{t_{i-1}}) \sqrt{\Delta}}$$

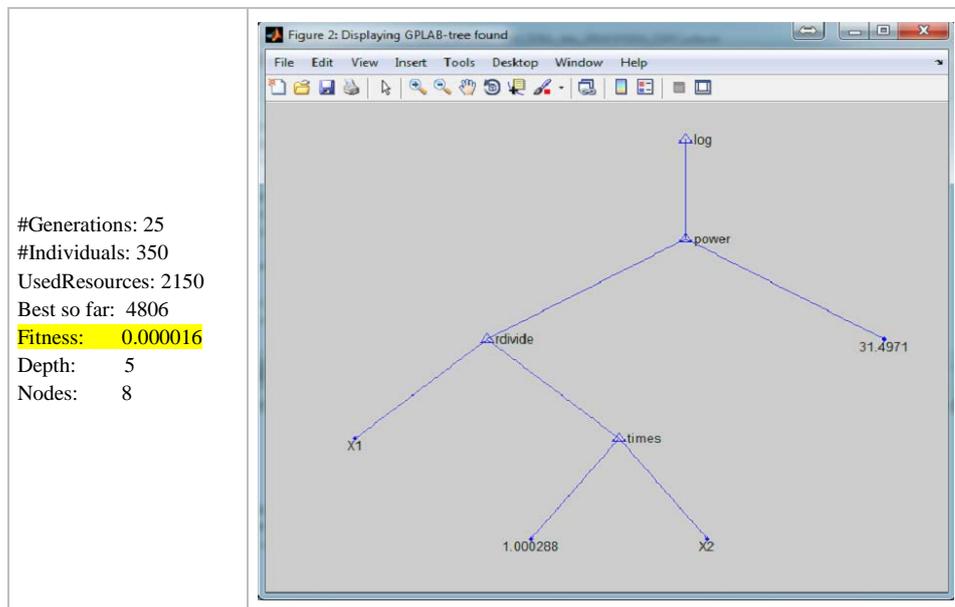


Figura 27: Especificação Inicial da EDE de MBG por PGDA

Portanto, a igualdade acima sugere como estimativas iniciais para os componentes da EDE transformada, $dY(t)$, os seguintes elementos:

$$\begin{aligned} \text{i) } Y(t) &= U(t) = \text{Log}(X(t)); \\ \text{ii) } \hat{a}(t, X(t)) &= \frac{0.000288}{\Delta} = \frac{0.000288}{0.004} = 0,072; \\ \text{iii) } \hat{b}(t, X(t)) &= \frac{0.031749}{\sqrt{\Delta}} = \frac{0.031749}{\sqrt{0.004}} = 0,501996. \end{aligned}$$

Aplicando o diferencial de Itô, com os resultados acima, determina-se também as estimativas iniciais para os coeficientes infinitesimais da EDE original, $dX(t)$. Então, sabe-se que,

$$\tilde{a}(t, X(t)) = \frac{\partial U(t, X(t))}{\partial t} + \frac{\partial U(t, X(t))}{\partial X(t)} a(t, X(t)) + \frac{1}{2} \times \frac{\partial^2 U(t, X(t))}{\partial X(t)^2} b^2(t, X(t)) \text{ e}$$

$$\tilde{b}(t, X(t)) = \frac{\partial U(t, X(t))}{\partial X(t)} b(t, X(t)).$$

Aplicando as derivadas parciais e substituindo as estimativas acima obtidas, chega-se ao seguinte sistema:

$$\begin{cases} 0,072 = \frac{1}{X(t)} a(t, X(t)) - \frac{1}{2 X(t)^2} b^2(t, X(t)) \\ 0,501996 = \frac{1}{X(t)} b(t, X(t)) \end{cases}$$

Resolvendo o sistema, obtêm-se as seguintes estimativas para os coeficientes:

$$\hat{a}(t, X(t)) = 0,198 X(t) \quad \text{e} \quad \hat{b}(t, X(t)) = 0,501996 X(t).$$

Portanto, as especificações iniciais estimadas para as EDEs $dX(t)$ e $dY(t)$ são:

$$dX(t) = 0,198 X(t) dt + 0,501996 X(t) dW(t)$$

$$dY(t) = 0,072 dt + 0,501996 dW(t).$$

Agora, apresenta-se a estimativa da trajetória do processo de ruído branco normal padrão (ε_t), obtida através da expressão

$$\hat{\varepsilon}_{t_i} = \log \left[\left(\frac{X_{t_i}}{1,000288 X_{t_{i-1}}} \right)^{31,4971} \right] = \frac{[\log(X_{t_i}) - \log(X_{t_{i-1}})] - 0,00288}{0,031749}.$$

Conforme mostra a Figura 28, a trajetória obtida apresenta valores aleatórios e com configuração gráfica típica de uma distribuição normal padrão. A hipótese de normalidade é confirmada pelo teste estatístico de Jarque-Bera, que aparece ao lado do histograma. Observe que a estatística calculada do teste é de 2,459581, com p-valor (probability) de 0,292354, o que indica que a hipótese de normalidade da série estimada não pode ser rejeitada mesmo a níveis de significância bem elevados. Além do teste estatístico formal, verifica-se que o histograma e as estatísticas amostrais, média, mediana, desvio padrão, assimetria e curtose, corroboram com a hipótese de uma distribuição $N(0,1)$. Contudo, estes testes não garantem a hipótese de ruído branco, para isso é necessário testar a estrutura de autocorrelação da série estimada.

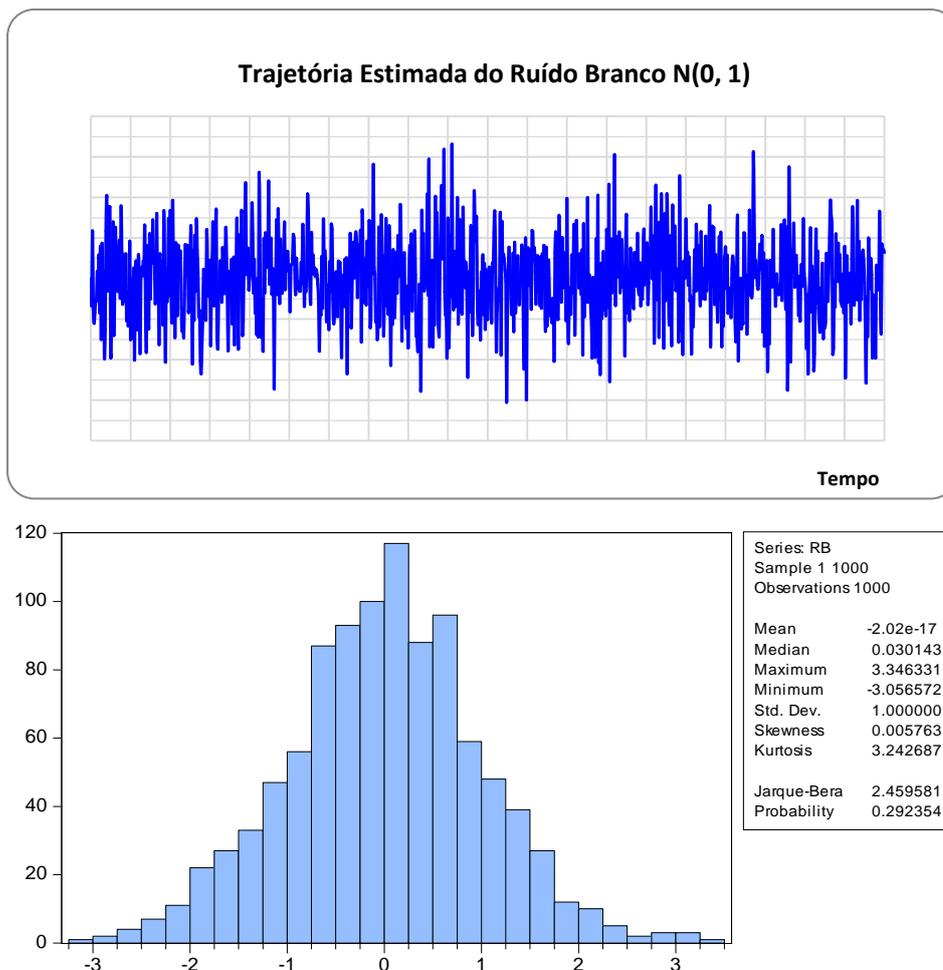


Figura 28: Trajetória de Ruído Branco e Teste de Normalidade

Teoricamente, um processo de ruído branco possui função de autocorrelação nula em todas as suas defasagens, pois, por definição, é uma sequência de variáveis aleatórias independentes e identicamente distribuídas. A Figura 29 exhibe os resultados da função de autocorrelação dos ruídos estimados e as estatísticas (Q-Stat) do teste de Ljung-Box⁹, usualmente utilizado para testar a hipótese de nulidade de autocorrelações acumuladas ao longo de diversas defasagens. Os resultados dos testes, aos níveis de significância usuais, de 1% e 5%, não rejeitam a hipótese de nulidade das autocorrelações, individuais ou acumuladas, em todas as defasagens apresentadas no gráfico. Em outras palavras, os testes indicam que não há evidência empírica para rejeitar a hipótese de ruído branco para a trajetória estimada.

⁹ Para maiores detalhes sobre o teste de Ljung-Box, indica-se (Hamilton, 1994; Morettin, 2004). No entanto, este é um teste clássico e pode ser encontrado na maioria dos livros texto de Séries Temporais.

Sample: 1 1000
Included observations: 1000

Autocorrelation	Partial Correlation	AC	PAC	Q-Stat	Prob
		1 -0.018	-0.018	0.3264	0.568
		2 0.054	0.053	3.2140	0.200
		3 -0.021	-0.019	3.6408	0.303
		4 -0.030	-0.034	4.5444	0.337
		5 -0.009	-0.008	4.6190	0.464
		6 -0.011	-0.008	4.7358	0.578
		7 -0.012	-0.012	4.8719	0.676
		8 0.031	0.030	5.8334	0.666
		9 0.034	0.036	7.0087	0.636
		10 0.008	0.004	7.0667	0.719
		11 -0.006	-0.009	7.1030	0.791
		12 0.042	0.044	8.8749	0.714
		13 -0.029	-0.024	9.7344	0.715
		14 0.014	0.009	9.9312	0.767
		15 -0.029	-0.023	10.758	0.770
		16 0.023	0.023	11.316	0.790
		17 -0.041	-0.041	13.037	0.734
		18 0.013	0.008	13.204	0.779
		19 -0.031	-0.026	14.177	0.773
		20 -0.019	-0.025	14.564	0.801
		21 0.018	0.017	14.892	0.828
		22 -0.006	-0.003	14.928	0.865
		23 -0.026	-0.030	15.629	0.871
		24 0.038	0.034	17.111	0.844
		25 -0.040	-0.032	18.741	0.809
		26 -0.024	-0.032	19.333	0.822
		27 -0.034	-0.028	20.526	0.808
		28 -0.066	-0.065	25.034	0.626
		29 0.027	0.031	25.779	0.637
		30 -0.019	-0.020	26.134	0.668

Figura 29: Correlograma e Testes de Ruído Branco

Uma vez testada e aceita, utilizou-se esta estimativa para estimar a trajetória do processo de Wiener (W_t), através da relação abaixo. Os resultados obtidos são mostrados na Figura 30.

$$\widehat{W}_{t_i} = \sqrt{\Delta} \sum_{k=1}^i \widehat{\varepsilon}_{t_k} = \widehat{W}_{t_i} = \sqrt{0,004} \sum_{k=1}^i \widehat{\varepsilon}_{t_k}$$



Figura 30: Trajetória do Processo de Wiener Estimada por PGDA

Uma vez identificada a função $Y_{t_i} = \text{Log}(X_{t_i})$ e estimada a trajetória do processo de Wiener (\widehat{W}_{t_i}), o próximo passo foi obter a especificação final para a EDE transformada $dY(t)$, através da estimação da sua forma discretizada. Isto é,

$$Y_{t_i} - Y_{t_{i-1}} = \tilde{a}(t_{i-1}, X_{t_{i-1}}) \Delta + \tilde{b}(t_{i-1}, X_{t_{i-1}}) (\widehat{W}_{t_i} - \widehat{W}_{t_{i-1}})$$

Esta segunda estimativa tem a finalidade de confirmar ou revisar as estimativas iniciais dos coeficientes (\tilde{a} e \tilde{b}) e concluir a etapa de especificação.

Fazendo $DY_{t_i} = (Y_{t_i} - Y_{t_{i-1}})$ e $D\widehat{W}_{t_i} = (\widehat{W}_{t_i} - \widehat{W}_{t_{i-1}})$ pode-se escrever a equação discretizada da seguinte forma:

$$DY_{t_i} = \tilde{a}(t_{i-1}, X_{t_{i-1}}) \times 0,004 + \tilde{b}(t_{i-1}, X_{t_{i-1}}) D\widehat{W}_{t_i} = G(t_{i-1}, X_{t_{i-1}}, D\widehat{W}_{t_i}).$$

Para solucionar o problema, aplicaram-se os codes de PGDA de forma a obter a função G que minimiza a soma de quadrados dos erros de estimação. Ou seja,

$$\min_{(G)} \sum_{i=1}^n [DY_{t_i} - G(t_{i-1}, X_{t_{i-1}}, D\widehat{W}_{t_i})]^2.$$

A função G estimada pelos codes de PGDA é dada pela árvore da Figura 31 e apresenta soma de quadrados dos erros quase nula, conforme pode ser observado nos resultados do Fitness (Best so far). A variável $X1$, que aparece na árvore, representa a variável $D\widehat{W}_{t_i}$. Assim, a função G estimada tem a seguinte forma analítica:

$$G(t_{i-1}, X_{t_{i-1}}, D\widehat{W}_{t_i}) = 0,07240 \times 0,004 + 0,50198 D\widehat{W}_{t_i}$$

Daí, as novas estimativas para os coeficientes infinitesimais de $dY(t)$ são:

$$\hat{a}(t, X(t)) = 0,0724 \quad \text{e} \quad \hat{b}(t, X(t)) = 0,50198.$$

Comparando com as estimativas iniciais, as diferenças são muito pequenas e certamente não afetam os resultados obtidos anteriormente. Revisando as estimativas dos coeficientes da EDE original, obtém-se:

$$\hat{a}(t, X(t)) = 0,19839 X(t) \quad \text{e} \quad \hat{b}(t, X(t)) = 0,50198 X(t).$$

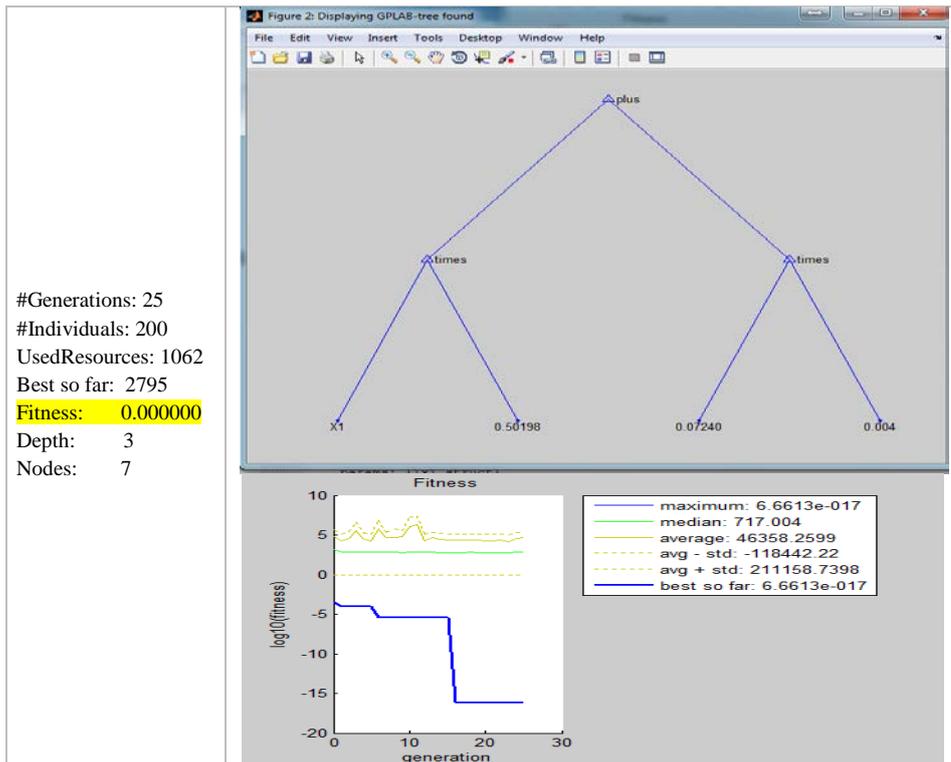


Figura 31: Especificação Final da EDE de MBG por PGDA

Portanto, conclui-se esta etapa com as seguintes especificações para as EDEs em tempo contínuo:

$$\begin{aligned} dX(t) &= 0,19839 X(t) dt + 0,50198 X(t) dW(t) \\ dY(t) &= 0,0724 dt + 0,50198 dW(t). \end{aligned}$$

#

Etapa 2: Solução da EDE especificada por PGDA

Nesta etapa é obtida a solução analítica exata da EDE especificada na etapa anterior, seguindo-se a metodologia descrita na seção (4.2.1). Em linhas gerais, aplica-se a integração estocástica na especificação estimada para $dY(t)$, obtendo-se a solução analítica para $Y(t)$ e, em seguida, por inversão, obtém-se a solução final para $X(t)$. Para isso, foram elaborados dois codes de PGDA que executam o seguinte procedimento para a solução:

Obter a função $f(t, X(t), \widehat{W}(t))$ que soluciona o seguinte problema,

$$\min_{(f)} \sum_{i=1}^n [Y_{t_i} - Y_0 - f(t_i, X_{t_i}, \widehat{W}_{t_i})]^2$$

onde,

$$f(t_i, X_{t_i}, \widehat{W}_{t_i}) \cong \sum_{s=1}^i \frac{\widehat{a}(t_s, X_{t_s}) + \widehat{a}(t_{s-1}, X_{t_{s-1}})}{2} \Delta + \sum_{s=1}^i \widehat{b}(t_{s-1}, X_{t_{s-1}}) (\widehat{W}_{t_s} - \widehat{W}_{t_{s-1}})$$

Uma vez obtida a função $f(t, X(t), \widehat{W}(t))$, a solução para $X(t)$ é obtida pela função inversa de $Y(t) = U(X(t))$, conforme mostra o cálculo abaixo:

$$X(t) = U^{-1}(Y(t)) = U^{-1}(Y(0) + f(t, X(t), \widehat{W}_t)) = h(t, X(s), \widehat{W}_t, X(0)).$$

Este é um procedimento geral e que pode ser usado para obter a solução de qualquer EDE tradicional com especificação conhecida. Aplicando os codes de PGDA ao exemplo em desenvolvimento, com $\Delta = 0,004$, $Y_{t_i} = \log(X_{t_i})$, $Y(0) = \log(20)$, $\widehat{a}(t, X(t)) = 0,0724$, $\widehat{b}(t, X(t)) = 0,50198$ e a trajetória estimada de \widehat{W}_{t_i} , obtém-se os resultados apresentados na Figura 32.

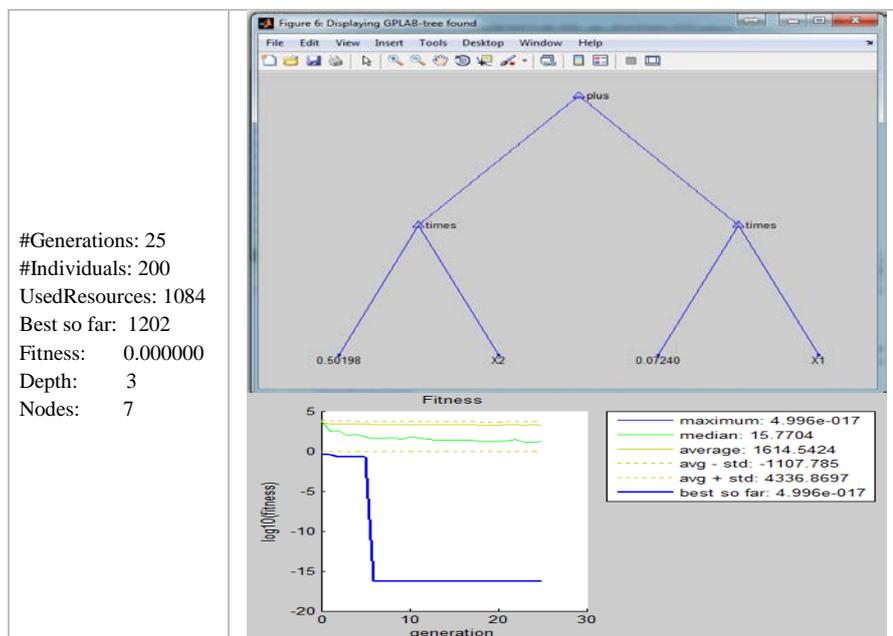


Figura 32: Solução da EDE transformada de MBG por PGDA

Conforme os resultados acima, onde $X1 = t$ e $X2 = \widehat{W}(t)$, a função f é estimada com baixíssimo nível de erro e apresenta a seguinte forma linear:

$$f(t, X(t), \widehat{W}(t)) = 0,0724 t + 0,50198 \widehat{W}(t).$$

Portanto, a solução exata estimada para $Y(t)$, em tempo contínuo, está determinada e é dada por

$$\widehat{Y}(t) = Y(0) + f(t, X(t), W(t)) = \log(20) + 0,0724 t + 0,50198 W(t).$$

Dado que $Y(t) = \log(X(t))$, aplicando a função inversa, chega-se a solução analítica e exata para a EDE original especificada,

$$\hat{X}(t) = 20 \exp\{0,0724 t + 0,50198 W(t)\}.$$

Portanto, aqui se encerra a etapa 2, com a apresentação da solução final do problema de interesse, onde se mostrou todas as fases da modelagem de PGDA na solução de uma EDE tradicional. No entanto, deve-se ressaltar que, esta etapa somente deve ser realizada quando a EDE especificada na etapa anterior não possuir solução analítica conhecida, caso contrário, ela é totalmente dispensável.

#

Dando prosseguimento ao exemplo, apresenta-se a seguir algumas análises pertinentes sobre os resultados obtidos. Inicialmente, comparando a solução obtida por PGDA com a solução teórica da EDE utilizada para a simulação dos dados, dada por $X(t) = 20 \exp\{0,075 t + 0,5 W(t)\}$, verificam-se pequenas diferenças entre os parâmetros propostos na simulação e os estimados por PGDA. No caso do coeficiente da tendência, o erro absoluto é de 0,0026, o que representa um erro relativo de aproximadamente 3,5%. Para o parâmetro da volatilidade, o erro absoluto é de 0,00198 e o erro relativo é de, aproximadamente, 0,4%. Dois fatores podem ter contribuído para estes erros: o primeiro diz respeito ao processo de simulação da variável $X(t)$ que, embora tenha sido realizada com base nos verdadeiros parâmetros da solução exata proposta, não necessariamente preserva estes valores na trajetória simulada. Isto ocorre, principalmente, em virtude da difícil tarefa de simular uma trajetória para o processo de Wiener que satisfaça a todas as suas propriedades básicas.

O segundo fator tem haver com o tamanho do intervalo de tempo (dt) e o número de observações ou pontos utilizados na modelagem, que, no caso da estimação de processos em tempo contínuo, são aspectos determinantes da qualidade das estimativas. No entanto, acredita-se que este não seja o problema, pois em toda a modelagem trabalhou-se com o pequeno intervalo de ($dt = 0,004$) e amostra com 1000 observações ou pontos.

Portanto, o mais provável é que o problema seja proveniente dos dados simulados para $X(t)$, mas devemos considerar que este é um risco que se corre quando se necessita modelar um fenômeno de dinâmica temporal com o conhecimento de apenas uma trajetória do processo estocástico. Em termos práticos, muitas vezes não se tem a opção de ter dados de boa qualidade

estatística, produzidos por fontes responsáveis, e em número suficiente que garantam uma boa modelagem. Entretanto, as diferenças entre os parâmetros são pequenas e não devem afetar substancialmente os resultados.

Para avaliar a qualidade das estimativas obtidas, realizou-se uma análise comparativa entre os dados simulados, os resultados da solução de PGDA, e os resultados obtidos por estimação numérica, realizada através do método de aproximação de Euler. Para obtenção dos resultados da solução numérica utilizou-se o code disponível no programa Matlab, próprio para estimação de EDE de MBG, e que executa o método de aproximação de Euler através da função “SimByEuler”.¹⁰

A Figura 33 apresenta a trajetória da solução simulada e as trajetórias das soluções obtidas por PGDA e pelo método numérico com aproximação de Euler. As trajetórias são muito próximas, se sobrepõem, que em termos visuais não é possível perceber as diferenças e avaliar os tamanhos dos erros. Para facilitar a análise, foi elaborada a Figura 34 que mostra os gráficos das diferenças entre trajetória simulada e cada uma das estimativas realizadas.

Observa-se que a solução de PGDA, por ser uma solução exata, apresenta erros mais bem comportados e bem inferiores aos registrados pela aproximação de Euler. Para avaliar as suas magnitudes, algumas medidas de erro foram calculadas e são apresentadas no Quadro 5.

¹⁰ Um resumo do code Matlab é apresentado na seção 7.1 do apêndice. O método de aproximação de Euler é um método clássico e de referência na estimação de equações estocásticas em tempo contínuo. Para maiores detalhes, sugere-se consultar as referências indicadas anteriormente para o método de discretização.

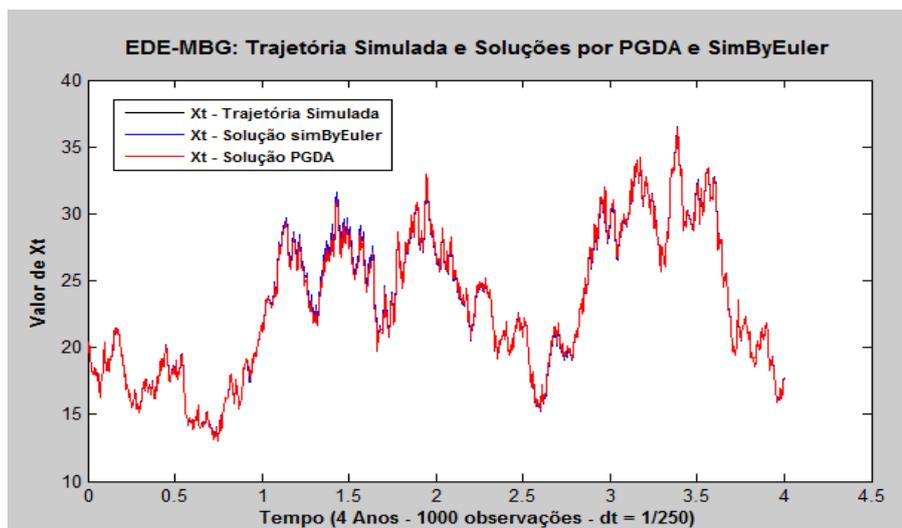


Figura 33: Trajetória Simulada e Soluções por PGDA e SimByEuler

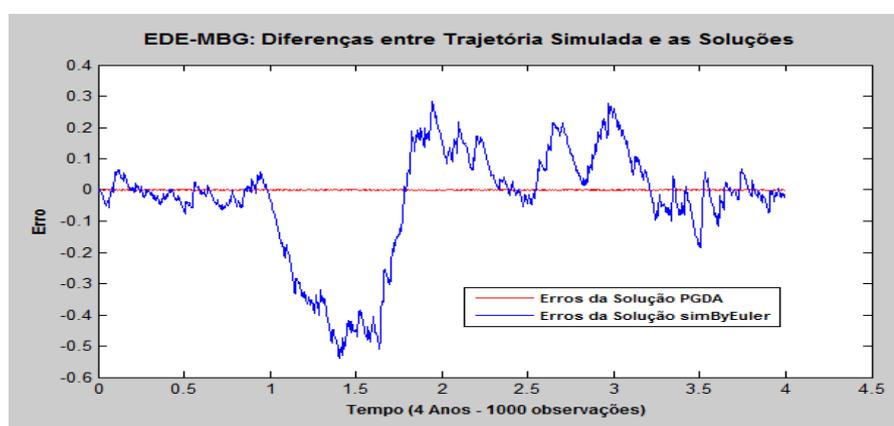


Figura 34: Diferenças entre a Trajetória Simulada e as Soluções

Quadro 5: Medidas de Erro e da Qualidade das Estimativas

Medida de Qualidade da Estimativa ¹¹	X(t) - Trajetória estimada por PGDA	X(t) - Trajetória estimada por simByEuler
MAPE (%)	1,154E-08	0,485481
MAE	2,551E-09	0,120871
REQM	2,927E-09	0,180460
R ²	0,9999999	0,998902

¹¹ Embora já sejam conhecidas, faz-se necessário informar o significado das abreviaturas apresentadas na primeira coluna do quadro: MAPE, erro percentual absoluto médio; MAE, erro absoluto médio; REQM, raiz do erro quadrático médio; e R², coeficiente de determinação entre a trajetória simulada e a respectiva estimativa.

De uma forma geral, os resultados mostram que as duas soluções apresentam boa qualidade de estimativa, com MAPE abaixo de 0,5% e R^2 muito próximo da unidade, o que indica uma forte aderência dos dados estimados aos produzidos pela simulação. Cabe destacar que, as medidas de erro derivadas da estimação por PGDA são bem inferiores às derivadas da estimação por aproximação de Euler.

No entanto, esta análise é muito agregada, focada somente no erro da estimativa final, e não avalia outros aspectos fundamentais da modelagem. Para melhor entender e avaliar a dinâmica de fenômenos estocásticos deve-se considerar que a solução de uma EDE é dividida em duas partes, uma determinística e outra aleatória. A parte determinística representa a tendência (drift) do processo e a parte aleatória a sua incerteza ou volatilidade. No exemplo que estamos tratando a solução obtida é a de um modelo multiplicativo e, nestes casos, as componentes de tendência e volatilidade aparecem multiplicadas na solução. Decompondo as soluções da EDE simulada e a obtida por PGDA, tem-se:

$$X(t) = Trends(t) \times Vs(t)$$

onde,

$Trends(t) = 20 \exp\{0,075 t\}$, tendência da solução da EDE simulada, $X(t)$; e

$Vs(t) = \exp\{0,5 W(t)\}$, volatilidade da solução da EDE simulada, $X(t)$.

$$\hat{X}(t) = Trend(t) \times V(t)$$

onde,

$Trend(t) = 20 \exp\{0,0724 t\}$, tendência da solução de PGDA, $\hat{X}(t)$; e

$V(t) = \exp\{0,50198 W(t)\}$, volatilidade da solução de PGDA, $\hat{X}(t)$.

A Figura 35 apresenta dois gráficos que comparam e mostram a evolução das tendências e volatilidades ao longo do tempo. Diferente da Figura 33, onde os efeitos destas componentes se misturam e se compensam, gerando erros quase nulos e imperceptíveis, nestes gráficos observa-se claras diferenças, mesmo que pequenas, entre as componentes simuladas e as estimadas por PGDA. Portanto, a forma mais adequada de comparar os resultados da solução simulada aos da solução obtida por PGDA é através da análise isolada de cada componente, a qual é apresentada a seguir.

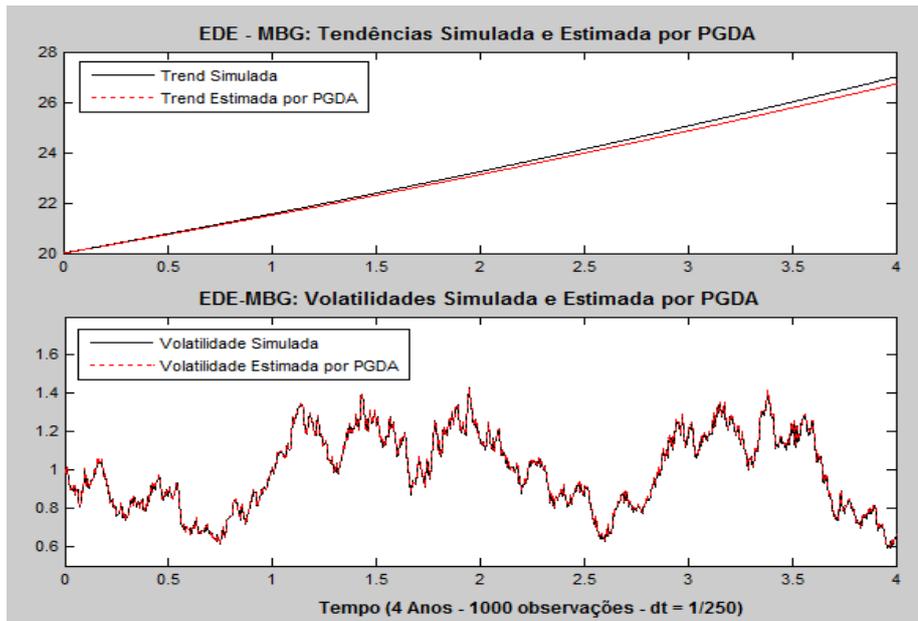


Figura 35: Comparações das Tendências e Volatilidades

Comparando a tendência do modelo simulado, supostamente verdadeira, com a tendência estimada por PGDA verifica-se uma pequena diferença de inclinação, resultante da diferença de 0,0026 entre o coeficiente proposto (0,075) e o coeficiente estimado (0,0724). Em virtude das tendências terem a forma exponencial, a diferença entre elas tende a aumentar ao longo do tempo. No entanto, a diferença entre os coeficientes não chega a afetar significativamente os resultados. Conforme mostra o Quadro 6, a diferença entre as tendências apresenta medidas de erro dentro de padrões aceitáveis, com MAPE de 0,52%, MAE de 0,13 e REQM de 0,15.

Quadro 6: Medidas das Estimativas da Tendência e da Volatilidade

Medidas de Qualidade da Estimativa	Comparação entre Tendências	Comparação entre Volatilidades
MAPE (%)	0,518201	0,532822
MAE	0,126901	0,005215
REQM	0,150333	0,005969
R2	0,999998	0,999805

Comparando as volatilidades, verifica-se que as trajetórias apresentam comportamentos com dinâmicas muito semelhantes, a estimativa de PGDA consegue capturar com grande similaridade a volatilidade simulada, o que é

confirmado pelo elevado R^2 de 0,9998. Além disso, as medidas de erro são muito baixas e certificam a boa qualidade da estimativa, com MAPE de 0,53%, MAE de 0,005 e REQM de 0,006.

Outra questão importante na análise deste tipo de modelagem é poder testar a significância dos parâmetros estimados. No entanto, as propriedades estatísticas do estimador de PGDA não são conhecidas e, por esta razão, não há como determinar o erro padrão das suas estimativas a partir da abordagem clássica da inferência estatística, como geralmente é feito em metodologias de estimação paramétrica.

Porém, neste tipo de situação, ou em casos de difícil determinação da distribuição de probabilidade do estimador ou do erro padrão, a alternativa mais indicada é o método de Bootstrap, desenvolvido por (EFRON, 1993). O objetivo principal do método, embora seja utilizado para outros fins, é estimar a variabilidade e o erro padrão de um estimador reamostrando, com reposição, observações da própria amostra utilizada na estimação. Este método está disponível no programa Matlab, com função de mesmo nome, e foi utilizado para calcular os erros padrão e gerar as estatísticas de teste que possibilitaram testar os coeficientes da tendência e da volatilidade da solução estimada por PGDA¹².

Quadro 7: Erros Padrão e Testes para Tendência e Volatilidade

Componente da Solução de PGDA	Coefficiente Estimado	Erro Padrão por Bootstrap	Estatística-t por Bootstrap	P-Valor
Tendência	0,07240	0,00287	25,24158	0,00000
Volatilidade	0,50198	0,00645	77,79998	0,00000

O Quadro 7 apresenta os resultados dos erros padrão, calculados pela técnica de Bootstrap, onde se utilizou 10.000 reamostragens dos erros de estimação de cada uma das componentes. Com estes resultados, foram calculadas as estatísticas-t e o P-valor normal para cada um dos testes. Analisando os resultados, verifica-se que os coeficientes estimados são fortemente significativos, com baixos erros padrão, onde a hipótese de nulidade de cada um é rejeitada pelo respectivo teste-t, com baixíssima probabilidade de erro, quase nula, conforme

¹² Um exemplo do code Matlab para Bootstrap é apresentado na seção 7.1 do apêndice.

mostra o p-valor. Aproveitando os erros padrão, foram calculados intervalos de confiança de (95%) para os coeficientes de cada componente, os resultados são os seguintes: I.C. de 95% para o coeficiente da tendência [0,06678 0,07802]; e I.C. de 95% para o coeficiente da volatilidade [0,48933 0,51463]. Com estes intervalos, temos um teste formal, com nível de significância de 5%, que não rejeita a hipótese de igualdade entre os coeficientes usados na simulação e os coeficientes estimados pela solução de PGDA. Basta observar que os intervalos contêm os respectivos coeficientes usados na simulação, que são 0,075 e 0,5. Portanto, as diferenças entre os coeficientes não são estatisticamente significativas e são, provavelmente, provenientes da aleatoriedade dos dados.

Com base nos resultados obtidos e nas análises realizadas, concluímos este exemplo com a convicção de que a metodologia desenvolvida pode ser uma possível alternativa para a solução de EDEs desta natureza. Contudo, para tentar chegar a uma conclusão mais definitiva e robusta sobre a sua qualidade, foram desenvolvidos mais alguns exercícios, cujos resultados são apresentados a seguir.

4.2.3

Solução da EDE de MBG por PGDA - Exemplo 2

Neste segundo exemplo apresenta-se um exercício similar ao realizado no Exemplo 1, mas com algumas modificações que, teoricamente, devem produzir resultados menos expressivos que os obtidos anteriormente. O objetivo do exercício é testar o desempenho da metodologia quando, com o mesmo problema de EDE, se aumenta intervalo de tempo (Δt) e se diminui o número de pontos ou períodos observados.

Para a realização do exercício, utilizou-se a mesma EDE solucionada no Exemplo 1: $dX(t) = 0,2 X(t) dt + 0,5 X(t) dW(t)$; com a mesma condição inicial $X(0) = 20$; definida no mesmo intervalo $[0, T] = [0, 4]$; mas com intervalo de tempo maior, o dobro do anterior, $\Delta t = \Delta = 1/125 = 0,008$; e com menor número de observações, $n = 500$, que é metade do utilizado anteriormente. Para que os resultados pudessem ser comparáveis, os dados utilizados são uma subamostra de 500 observações, selecionadas sequencialmente, uma a cada duas observações, da amostra de 1000 observações simuladas para o Exemplo 1.

A solução teórica da EDE é a mesma, $X(t) = 20 \exp\{0,075 t + 0,5 W(t)\}$, e os resultados obtidos por PGDA são os seguintes:

i) EDE especificada: $d\hat{X}(t) = 0,21102 X(t) dt + 0,52894 X(t) d\hat{W}(t)$;

ii) Solução obtida: $\hat{X}(t) = 20 \exp\{0,07113 t + 0,52894 W(t)\}$.

As Figuras 36 e 37 mostram, graficamente, a boa qualidade das estimativas, mesmo quando realizadas com maior intervalo (dt) e menor número de observações. O Quadro 8 apresenta as medidas de erro, as quais são levemente superiores às medidas apresentadas no exemplo anterior. Contudo, a perda de qualidade da solução geral e das componentes de tendência e volatilidade é muito pequena. Verifica-se, como no Exemplo 1, um melhor desempenho da solução de PGDA em relação à solução por aproximação de Euler, com erros bem menores. Os erros padrão dos coeficientes infinitesimais da tendência e da volatilidade são maiores que os obtidos no exemplo anterior, mas são suficientemente pequenos para garantir a significância destes coeficientes, com elevado nível de confiança. Portanto, o exemplo mostra que o aumento do intervalo (dt) e a redução do número de observações afetam a qualidade da solução de PGDA, mas de maneira bem mais suave do que era esperado. Isto talvez ocorra pelo fato da metodologia trabalhar com derivadas que independem do tamanho do intervalo.

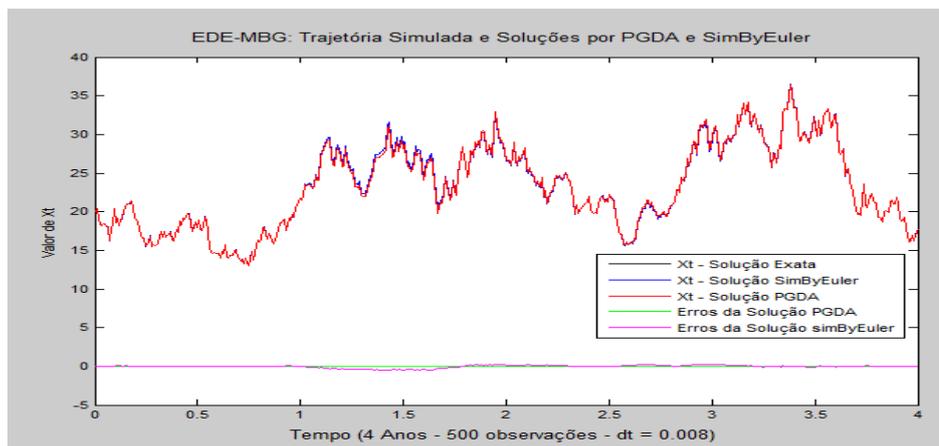


Figura 36: Trajetórias das Soluções por PGDA e SimByEuler

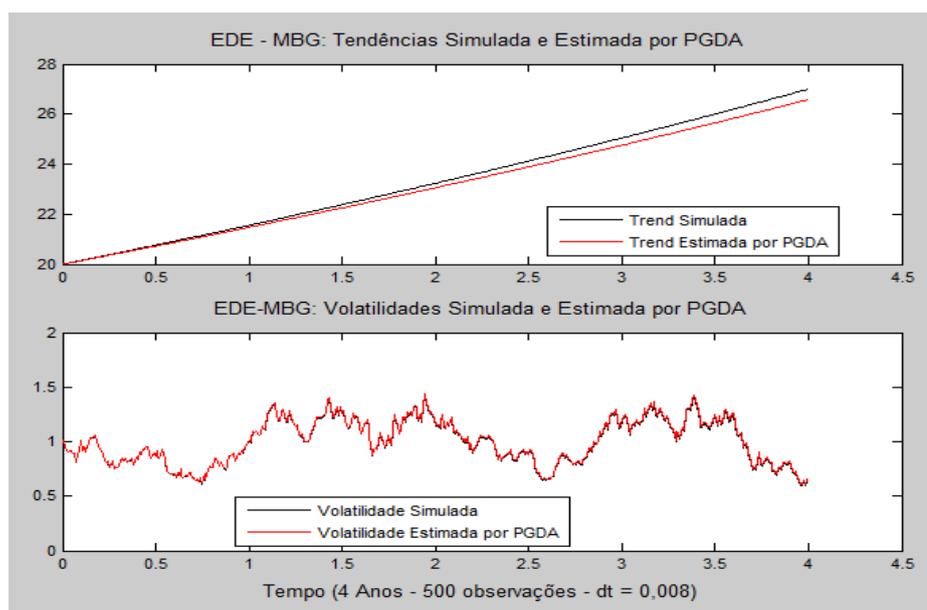


Figura 37: Comparações entre Tendências e Volatilidades

Quadro 8: Estatísticas da Qualidade das Estimativas e Testes

Componente da Solução de PGDA	Coefficiente Estimado	Erro Padrão por Bootstrap	Estatística-t por Bootstrap	P-Valor
Tendência	0,07113	0,00402	17,70653	0,00000
Volatilidade	0,52894	0,00948	55,81136	0,00000
Medida de Qualidade da Estimativa	Trajectoria estimada por PGDA	Trajectoria estimada por simByEuler	Comparação das Tendências	Comparação das Volatilidades
MAPE (%)	3,181E-05	0,483684	0,770018	0,778021
MAE	7,056E-06	0,120543	0,188580	0,007863
REQM	8,185E-06	0,179917	0,223408	0,009209
R2	1,000E+00	0,998912	0,999996	0,999566

4.2.4

Modelagem dos Preços da Ação PETR4 - Exemplo 3

Neste exemplo, diferente dos anteriores, apresenta-se uma aplicação da metodologia de PGDA a um problema real, onde o modelo de EDE é totalmente desconhecido. A aplicação é desenvolvida na tentativa de se obter uma especificação e a solução analítica para a EDE que representa a dinâmica dos preços de uma particular ação do mercado financeiro. A série temporal escolhida para o exercício foi a de preços nominais da PETR4, ação preferencial nominativa

(PN), pertencente a Petróleo Brasileiro S.A. – Petrobras, e uma das principais ações negociadas na Bolsa de Valores de São Paulo – BM&FBovespa.



Os dados utilizados na modelagem são preços nominais de fechamento (ajustado), com periodicidade diária, coletados junto ao banco de dados da BM&FBovespa, e disponíveis para o período de 03-jun-2000 a 30-dez-2014, conforme mostra o gráfico acima¹³. Contudo, em virtude dos diferentes momentos de conjuntura econômica e política, nacional e internacional, ocorridas ao longo deste período de 15 anos, passando pela crise internacional de 2008, a série disponível, completa, apresenta evidências claras de mudanças estruturais do processo estocástico, que alteraram a sua tendência e a sua volatilidade. Por estes motivos, que complicam demasiadamente o processo de modelagem, decidiu-se trabalhar com dados mais atuais, referentes ao período dos últimos cinco anos, de 04-jan-2010 a 30-dez-2014, com um total de 1195 observações diárias, conforme mostra a Figura 38.

¹³ Preços de fechamento ajustados são preços que sofrem ajustes em função do volume de operações realizadas, onde são ponderados os efeitos de possíveis sobrevalorizações ou desvalorizações especulativas para casos de pagamentos de dividendos ou de operações de desdobramento e agrupamento.

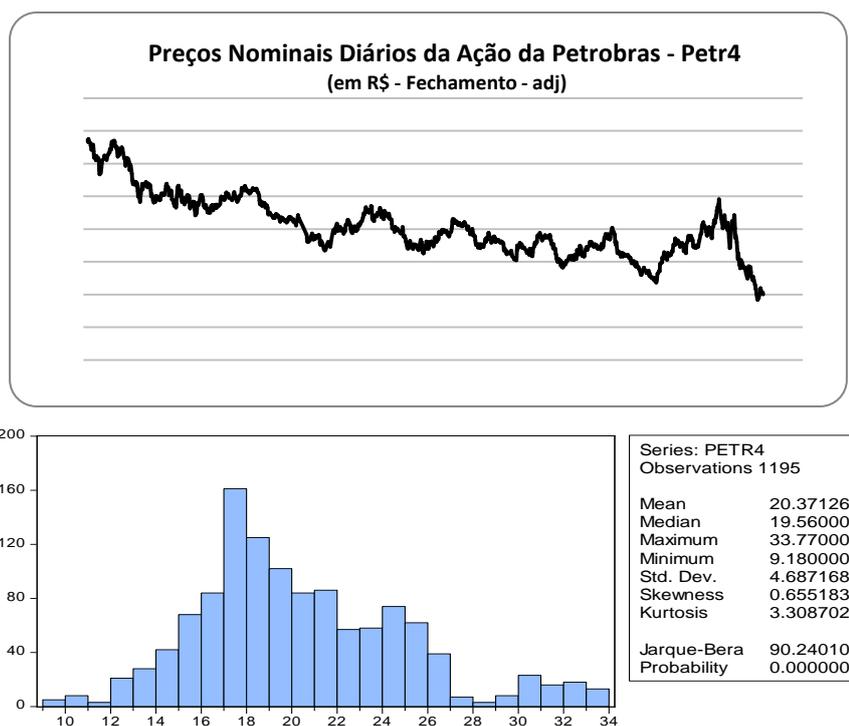


Figura 38: Preços Nominais Diários de Fechamento da PETR4

Tratando-se de um fenômeno com comportamento desconhecido, faz-se necessário obter algum conhecimento empírico que dê suporte analítico aos resultados obtidos no processo de modelagem. Por isso, uma breve análise estatística dos dados é necessária.

Analisando o gráfico da série e as estatísticas descritivas, para o período considerado, verifica-se uma clara tendência de queda dos preços, decaindo do valor máximo de R\$33,77, registrado em 6-jan-10, para o valor mínimo de R\$9,18, registrado em 15-dez-14. O que representa em termos relativos, uma queda de 72,8% no valor da ação. O preço médio é de R\$20,37, com desvio padrão de R\$4,69, o que implica em um elevado coeficiente de variação de 23%. Os coeficientes de assimetria e curtose, de 0,655 e 3,309, respectivamente, retratam a não normalidade dos dados, a qual é confirmada pelo teste Jarque-Bera, com estatística de 90,24, rejeitando a hipótese com elevado nível de confiança.

Embora o processo estocástico de interesse seja o preço da ação, na maioria dos casos, principalmente para aqueles que atuam no mercado de ações, o maior interesse é pela modelagem do retorno ou, como usualmente se trabalha, do $\log(\text{retorno})$. Pois, com isso, é possível avaliar o risco e a rentabilidade esperada

quando se aplica na específica ação. Contudo, ao se obter a solução da EDE do preço, naturalmente se terá a equação do retorno.

Para a realização do exercício foram definidos e utilizados os seguintes parâmetros para a solução da EDE na forma tradicional:

i) Modelo de EDE: $dX(t) = a(t, X(t)) dt + b(t, X(t)) dW(t)$

onde, $X(t)$: é o preço da ação PETR4 no tempo t ;

ii) Condição inicial: $X(0) = 33,6$; preço da ação em 04-jan-2010;

iii) Número de períodos ou observações: $n = 1195$;

iv) Variação infinitesimal do tempo: $t_i - t_{i-1} = \Delta = 0,004$; (constante)

v) Intervalo de tempo: $[0, T] = [0, 5]$ (aproximadamente, cinco anos de dias úteis).

Os procedimentos utilizados na modelagem seguiram os mesmos passos da metodologia apresentada nos exemplos anteriores, ou seja, primeiramente se realiza a especificação da EDE e, em seguida, determina-se a solução final.

Aplicando-se os algoritmos de PGDA aos dados observados, obteve-se na primeira etapa, após a estimação das trajetórias do ruído branco e do processo de Wiener, a seguinte estimativa para a especificação discretizada da EDE transformada, que é conhecida como a equação dos log-retornos:

$$Y_{t_i} - Y_{t_{i-1}} = \hat{a}_{t_{i-1}} \Delta + \hat{b}_{t_{i-1}} (\hat{W}_{t_i} - \hat{W}_{t_{i-1}})$$

onde,

$$Y_{t_i} = \log(X_{t_i}); \hat{a}_{t_i} = -0,01043 t_i^{-0,48195}; \Delta = 0,004;$$

$$\hat{b}_{t_i} = \sqrt{0,0000144 + [0,07331 (\hat{W}_{t_{i-1}} - \hat{W}_{t_{i-2}})^2 + 0,90405] \hat{b}_{t_{i-1}}^2};$$

com $\hat{b}_0 = 0,000186$; e \hat{W}_{t_i} : é a estimativa do processo de Wiener.

Interpretando os coeficientes estimados, observa-se que o coeficiente da parte determinística, $\hat{a}_{t_i} = -0,01043 t_i^{-0,48195}$, indica que a média condicional dos log-retornos não é nula, como esperado, é negativa, mas converge para zero ao longo do tempo, conforme mostra o gráfico abaixo.

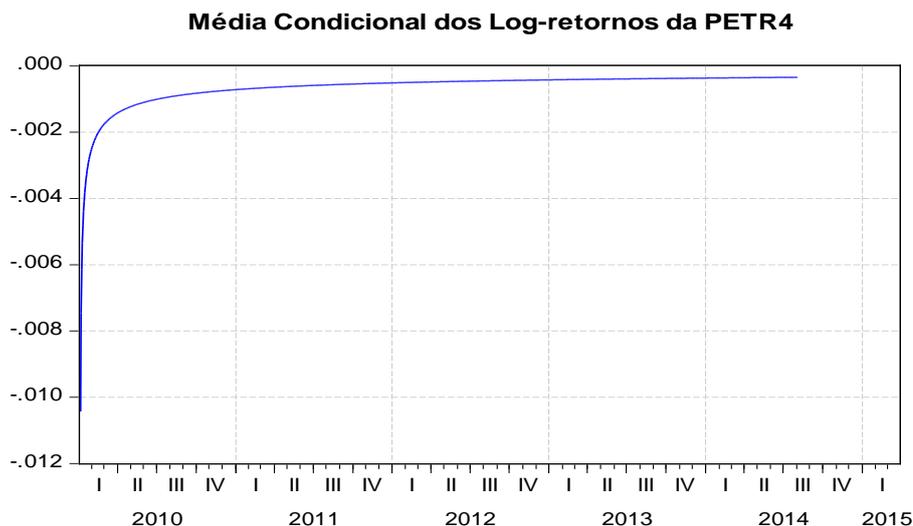


Figura 39: Média Condicional dos Log-Retornos da PETR4

O coeficiente estimado da parte aleatória, \hat{b}_{t_i} , é determinado, em seu quadrado, por uma forma auto-regressiva com o seu passado imediato. Onde, a sua atualização no tempo depende da variação passada ocorrida no processo de Wiener. Desenvolvendo-se a equação, desde o seu valor inicial, pode-se mostrar que, de forma ponderada, o coeficiente acumula toda informação das variações ocorridas no processo de Wiener até o tempo imediatamente anterior. Além disso, quando aplicado à equação, adiciona a variação do processo ocorrida no mesmo momento do tempo.

A especificação e estimação deste coeficiente, na sua forma recursiva, como foi obtida, é de fundamental importância na explicação das variações ocorridas na série de dados, tanto no logaritmo dos retornos, quanto nos preços da PETR4. A determinação deste coeficiente na modelagem de PGDA não foi de simples obtenção e foi perseguido com muita expectativa, não por fatores técnicos da método de solução, pois o algoritmo, com poucas gerações e população razoavelmente pequena, rapidamente obtém solução de baixo erro. O problema foi obter uma solução justificável e que atendesse aos aspectos teóricos mínimos da modelagem de preços e retornos em finanças. A expressão obtida é consistente e compatível com as usadas nos diversos modelos do gênero, ver (Bollerslev, 1986, 1994; Hamilton, 1994; Campbell et al., 1997).

Ao calcular a variância condicional da equação de log-retornos, verifica-se que este coeficiente ganha grande importância. Pois,

$$\sigma_{t_i}^2 = \text{Var}[(\log(X_{t_i}) - \log(X_{t_{i-1}})) | X_{t_{i-1}}] = (\tilde{b}_{t_i})^2 \text{Var}(W_{t_i} - W_{t_{i-1}}) = \tilde{b}_{t_i}^2 \Delta$$

ou,

$$\hat{\sigma}_{t_i}^2 = 0,0000144 \Delta + [0,07331 (\widehat{W}_{t_{i-1}} - \widehat{W}_{t_{i-2}})^2 + 0,90405] \widehat{b}_{t_{i-1}}^2 \Delta; \Leftrightarrow$$

$$\hat{\sigma}_{t_i}^2 = 0,0000144 \Delta + [0,07331 (\widehat{W}_{t_{i-1}} - \widehat{W}_{t_{i-2}})^2 + 0,90405] \hat{\sigma}_{t_{i-1}}^2.$$

Ou seja, a menos do intervalo infinitesimal de tempo, os coeficientes $\widehat{b}_{t_i}^2$ e $\hat{\sigma}_{t_i}^2$ são, respectivamente, a variância e o desvio padrão condicional dos log-retornos, denotados em finanças de volatilidade. A Figura 40 apresenta o comportamento da volatilidade estimada por PGDA ao longo do período em análise. Uma das suas funções é apontar os “fatos estilizados” ou pontos que registram valores bem mais elevados e destoantes dos demais. O conhecimento destes pontos é de fundamental importância para a análise financeira, pois permite identificar e avaliar os efeitos de choques, positivos e negativos, sobre o log-retorno dos preços da ação-objeto. Consequentemente, toda medida de análise de risco de retorno desta ação deverá está associada à sua volatilidade.

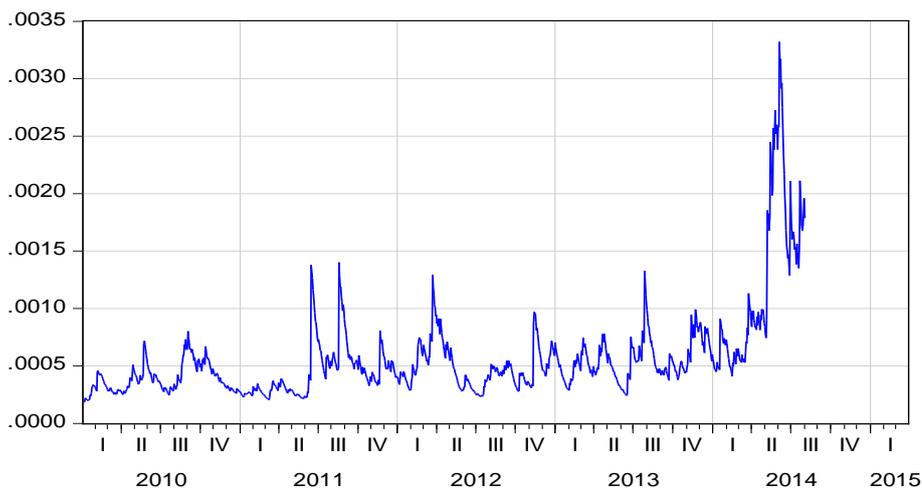


Figura 40: Volatilidade Estimada para os Log-Retornos da PETR4

Este parâmetro ainda será analisado, mais adiante, e comparado com as volatilidades estimadas por outros modelos.

Com relação à estimativa do processo de Wiener (\widehat{W}_{t_i}), esta pode ser vista na seção 7.3 do apêndice.

Uma vez concluída a etapa de especificação, a próxima etapa foi a aplicação do algoritmo de PGDA para obter a solução contínua da EDE. Ou seja, obter a solução do seguinte problema:

$$dY(t) = \hat{a}(t) dt + \hat{b}(t) dW(t)$$

Aplicando-se integração estocástica, a solução teórica do problema é dada por:

$$\int_0^t dY(s) = \log(X(t)) - \log(33,6) = \int_0^t \hat{a}(s) ds + \int_0^t \hat{b}(s) dW(s).$$

A primeira integral não necessita ser solucionada por PGDA, pois é uma integral de Riemann de simples solução. Logo,

$$\int_0^t \hat{a}(s) ds = \int_0^t -0,01043 s^{-0.48195} ds = -0,02014 t^{0.51805}.$$

A segunda integral é um caso particular da integral de Itô que não possui solução fechada conhecida. Portanto, o resultado apresentado a seguir é a melhor aproximação contínua que se conseguiu por PGDA para a integral. Ou seja,

$$\int_0^t \hat{b}(s) dW(s) \cong \sqrt{0,00015 + 0,00037 \left(\frac{1 - 0,904045^t}{t} \right)} \hat{W}(t).$$

Somando os resultados das duas integrais e aplicando a função exponencial à $Y(t)$, chega-se a seguinte expressão para a solução contínua, aproximada, da EDE:

$$\hat{X}(t) = X(0) \exp \left\{ \beta_1 t^{\beta_2} + \sqrt{\alpha_1 + \alpha_2 \left(\frac{1 - \alpha_3^t}{t} \right)} W(t) \right\} \quad (13)$$

onde,

$$X_0 = 33,6 \text{ (CI)}; \beta_1 = -0,02014; \beta_2 = 0,51805; \alpha_1 = 0,00015; \alpha_2 = 0,00037; \text{ e} \\ \alpha_3 = 0,904045.$$

Além desta solução, pode-se obter uma solução discretizada para a EDE através da fórmula aproximada da integral de Itô. Segue que,

$$\hat{X}(t) = X(0) \exp \left\{ \beta_1 t^{\beta_2} + \int_0^t \hat{b}(s) dW(s) \right\},$$

onde,

$$\int_0^t \hat{b}(s) dW(s) \cong \sum_{i=1}^n \hat{b}_{t_{i-1}} (\hat{W}_{t_i} - \hat{W}_{t_{i-1}})$$

Da EDE especificada na forma discretizada, tem-se que:

$$\hat{b}_{t_i} = \sqrt{0,0000144 + [0,07331 (\hat{W}_{t_{i-1}} - \hat{W}_{t_{i-2}})^2 + 0,90405] \hat{b}_{t_{i-1}}^2} .$$

Logo, a solução da EDE na forma discretizada é dada por:

$$\hat{X}_{t_i} = 33.6 \exp \left\{ -0,02014 t_i^{0,51805} + \sum_{k=1}^i \hat{b}_{t_{k-1}} (\hat{W}_{t_k} - \hat{W}_{t_{k-1}}) \right\} \quad (14)$$

onde, $i = 1, 2, \dots, n$.

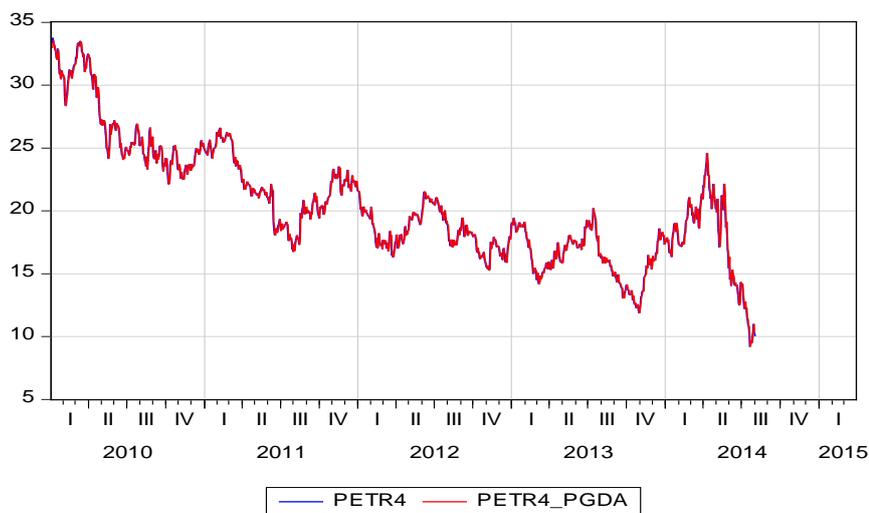


Figura 41: Solução de PGDA para a PETR4

A vantagem de se ter e utilizar a solução discretizada é que ela captura com maior precisão as fortes variações registradas pela série de dados, pois estes são observados de forma discreta. Contudo, fora dos pontos da discretização, esta solução não pode ser usada e a solução contínua é a melhor alternativa para predição e previsão dos dados, embora apresente comportamento bem mais suave que o da solução discretizada.

A Figura 41 compara o resultado da solução discretizada obtida com os dados originais dos preços da ação PETR4. O resultado apresenta uma boa qualidade de ajuste, com baixas medidas de erros, conforme será mostrado mais adiante. Contudo, é necessário decompor a solução em suas componentes de tendência e volatilidade para se avaliar melhor a qualidade do resultado obtido. Usando as notações de T , para a tendência, e de V , para a volatilidade total, a solução discretizada foi decomposta e as expressões e os resultados são apresentados a seguir.

$$\hat{X}_{t_i} = \hat{T}_{t_i} \times \hat{V}_{t_i}$$

onde,

$$\hat{T}_{t_i} = 33,6 \exp\{-0,02014 t_i^{0,51805}\}; e$$

$$\hat{V}_{t_i} = \exp\left\{\sum_{k=1}^i \hat{b}_{t_{k-1}} (\hat{W}_{t_k} - \hat{W}_{t_{k-1}})\right\}$$

onde,

$$\hat{b}_{t_k} = \sqrt{0,0000144 + \left[0,07331 (\hat{W}_{t_{k-1}} - \hat{W}_{t_{k-2}})^2 + 0,90405\right] \hat{b}_{t_{k-1}}^2}$$

com $\hat{b}_0 = 0,000186$; e $i = 1, 2, \dots, n$.

Conforme mostra a Figura 42, a tendência estimada tem a forma exponencial e decai fortemente ao longo da série. Apresenta uma boa qualidade de ajuste, posicionando-se bem ao centro dos dados e indica ser uma boa representação para a média incondicional dos preços da PETR4.

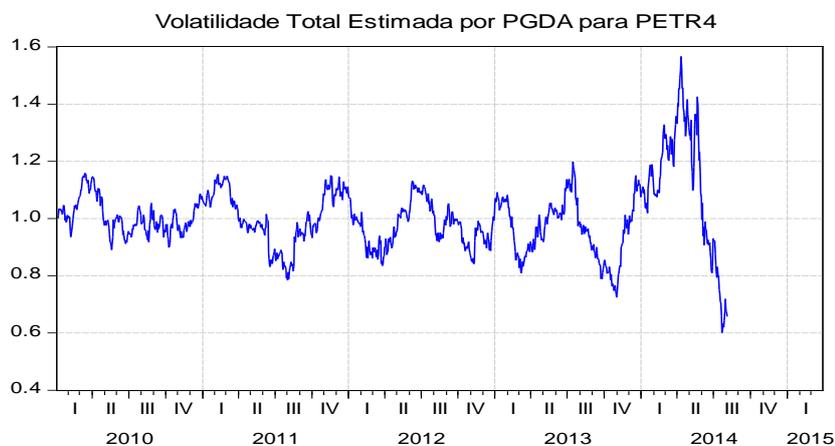


Figura 42: Tendência e Volatilidade Total para a PETR4

Retirando-se a tendência dos dados, que é a parte determinística da solução, sobra a parte aleatória e de maior complexidade de modelagem. Observe que, pelo fato do modelo ser multiplicativo, a amplitude de variação da volatilidade total estimada é bem inferior ao da série como um todo. No entanto, ela mostra-se claramente não estacionária na variância e, por ser expressa em exponencial, oscilando crescentemente em torno da média 1. Isto corrobora com a especificação do modelo obtido, pois ao aplicarmos o logaritmo aos dados ainda teremos um comportamento crescente e oscilante, mas agora em torno da média zero. Isto é, muito semelhante ao comportamento de um processo de Wiener.

A próxima etapa da modelagem foi testar a significância dos coeficientes estimados para as componentes da solução, o Quadro 9 apresenta estes resultados.

Quadro 9: Estatísticas de Erro Padrão e Testes – PETR4

Componente do Modelo	Estimativa do Coeficiente	Erro Padrão por Bootstrap	Estatística-z por Bootstrap	P-Valor
Tendência				
β_1	-0,02014	0,00136	-14,77918	0,00000
β_2	0,51805	0,01045	49,56432	0,00000
Volatilidade Total				
b_0	0,000186	0,0000541	3,43634	0,00060
δ_1	0,000014	0,0000031	4,66733	0,00000
δ_2	0,073311	0,0113750	6,44488	0,00000
δ_3	0,904045	0,0131200	68,90649	0,00000

De acordo com as estatísticas acima, todos os coeficientes das equações da tendência e da volatilidade total são significativamente diferentes de zero, com elevado nível de confiança, conforme atestam as estatísticas-z e os p-valores. Com isso, as equações estimadas e os resultados obtidos possuem significância estatística¹⁴.

Após a validação do modelo estimado, o próximo passo foi comparar a solução obtida por PGDA com outras metodologias. Três modelos usualmente utilizados na modelagem de preços de ativos financeiros foram estimados para a

¹⁴ Os erros padrão calculados por Bootstrap foram obtidos nas específicas equações de tendência e volatilidade em logaritmos.

comparação. Os modelos são os conhecidos ARCH, GARCH e EGARCH¹⁵, ver (Bollerslev, 1986, 1994; Campbell et al., 1997; Morettin, 2004).

Os resultados das estimativas dos três modelos e mais a solução de PGDA são apresentados na Figura 43. Observa-se que as trajetórias das estimativas se confundem com a série observada, mostrando que os quatro ajustes são de boa qualidade e conseguem reproduzir com baixo erro o comportamento dos preços da PETR4.

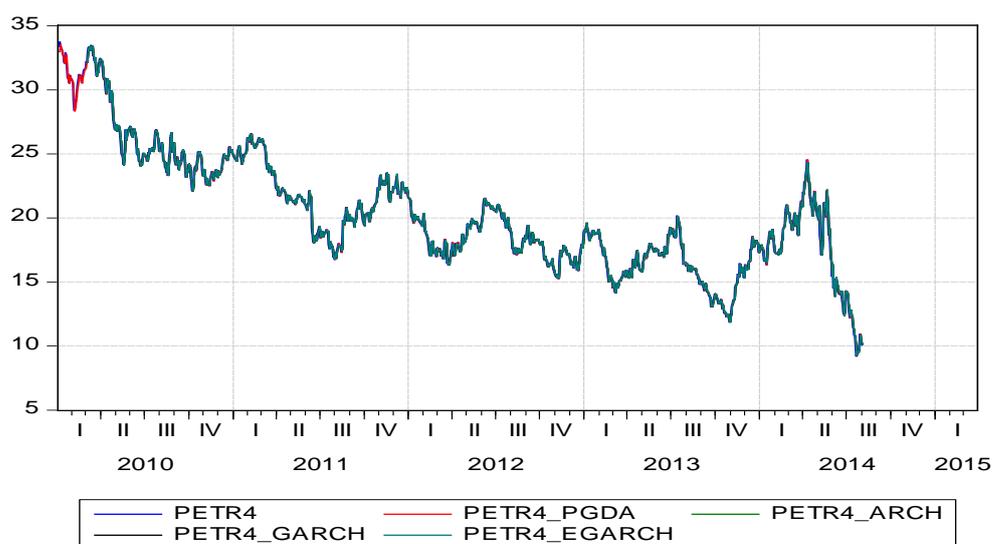


Figura 43: Estimativas dos Modelos para os Preços da PETR4

O Quadro 10 confirma a similaridade entre as estimativas dos modelos, onde se verifica medidas de erro muito baixas e com magnitudes muito próximas. O coeficiente de determinação (R^2), entre cada uma das estimativas e a série da PETR4, mostram a forte correlação das estimativas com os dados originais e confirmam a boa qualidade dos ajustes. Os resultados das estimativas e testes realizados na estimação dos modelos ARCH, GARCH e EGARCH são apresentados na seção 7.3 do apêndice.

¹⁵ As siglas ARCH, GARCH e EGARCH significam, respectivamente, (Autoregressive Conditional Heteroskedasticity), (Generalized Autoregressive Conditional Heteroskedasticity) e (Exponential Generalized Autoregressive Conditional Heteroskedasticity)

Quadro 10: Estatísticas das Estimativas dos Modelos – PETR4

Estatística	Modelo			
	PGDA	ARCH	GARCH	EGARCH
MAPE (%)	1,73465	1,74883	1,74836	1,74813
MAE	0,33737	0,33543	0,33530	0,33521
REQM	0,45607	0,45497	0,45480	0,45483
R ² com PETR4	0,99053	0,98865	0,98862	0,98863

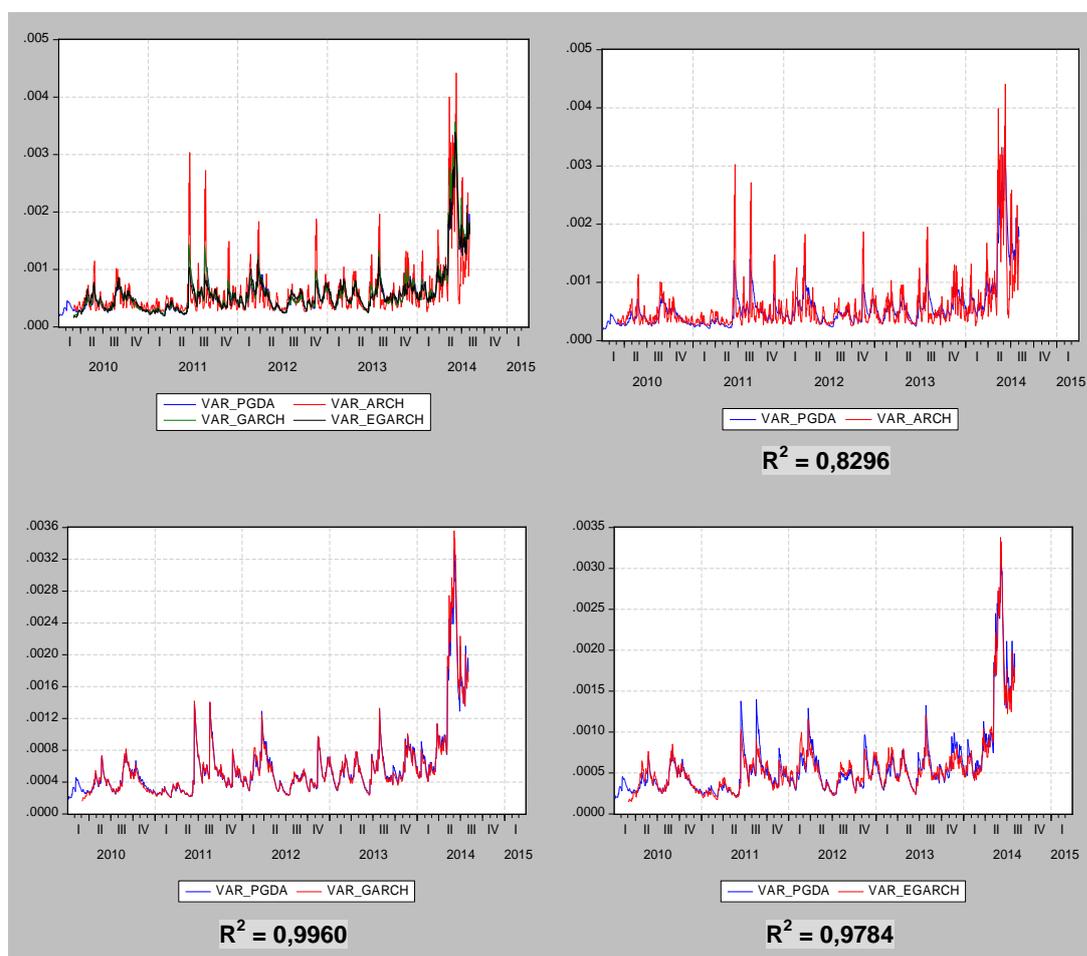


Figura 44: Estimativas das Volatilidades dos Modelos para a PETR4

Para finalizar o exemplo, estimou-se a variância condicional dos log-retornos ou volatilidade pelos três modelos e comparou-se com a volatilidade estimada por PGDA. Os resultados são apresentados na Figura 44 e mostram forte correlação entre as volatilidades estimadas pelos modelos e a volatilidade

estimada por PGDA. Nos casos dos modelos GARCH e EGARCH os coeficientes R^2 são de 0,9960 e 0,9784, respectivamente. Apenas a volatilidade do modelo ARCH é que difere um pouco, mesmo assim apresenta R^2 de 0,8296.

Portanto, com base nos resultados apresentados, concluímos o exemplo com a certeza de que a solução de PGDA é uma alternativa bastante satisfatória para ser usada na modelagem dos preços da ação PETR4. Pois, além de apresentar baixas medidas de erro, coeficientes significativos e boa qualidade de ajuste, seus resultados são equiparáveis aos dos reconhecidos modelos ARCH, GARCH e EGARCH.

#

Além dos três exemplos apresentados, um quarto exemplo foi desenvolvido, onde se obtém a solução por PGDA para os preços da ação da VALE5, pertencente à empresa VALE S.A.. Embora os resultados obtidos sejam significativos, a modelagem é repetitiva e segue os mesmos passos da realizada para a PETR4, por isso, decidiu-se não apresentar o exemplo nesta subseção e reportar os seus resultados na seção 7.4 do apêndice.

Considerando a qualidade dos resultados obtidos nos exemplos, concluímos esta subseção acreditando que a metodologia desenvolvida por PGDA é consistente, promissora e uma boa alternativa para a solução de problemas de EDEs do tipo abordado.

4.2.5

Modelo de Precificação de Opções Europeias por PGDA

Nesta seção é apresentada uma breve aplicação da metodologia de PGDA na tentativa da precificação ou apreçamento de opções europeias. O objetivo do exercício não é o de se aprofundar e discutir questões teóricas de finanças sobre o tema, mas sim de tentar mostrar que a metodologia de PGDA pode ser uma alternativa viável para este fim. O modelo que será proposto difere dos demais modelos existentes, basicamente, na forma de definição e especificação da equação diferencial que rege a dinâmica dos preços do ativo-objeto. Para testar a metodologia proposta, foram realizados alguns exercícios com as ações PETR4 e VALE5.

4.2.5.1

Conceitos Básicos sobre Opções

Nesta seção é apresentada uma breve introdução dos conceitos básicos e definições sobre contratos de opções, necessários ao entendimento do modelo de precificação proposto. Os conceitos apresentados e os demais utilizados na seção foram retirados de (Hull, 1997; Marins, 2004; AIUBE, 2013).

Um importante segmento dos mercados financeiros é conhecido como mercado de derivativos. Contudo, o que são derivativos?

Definição 1: Derivativos

Derivativos são contratos ou ativos negociados no mercado a vista, em bolsas de valores, cujos valores *derivam* do comportamento ou posição de outro ativo subjacente, denominado de ativo-objeto. São classificados em contratos futuros, a termo, *opções*, operações de swaps etc.

Definição 2: Opção

Um importante segmento do mercado de derivativos é o mercado de opções. Conceitualmente, uma opção é um contrato que dá ao seu detentor um direito sobre o objeto da negociação, mas não uma obrigação. Um contrato de seguro é um bom exemplo de opção. O segurado tem o direito de ser ressarcido de um sinistro, mas não tem qualquer obrigação, a não ser o pagamento do prêmio do seguro, o qual é efetuado no ato da contratação. As negociações de opções sobre ações e outros tipos de ativos são realizadas, no Brasil, na BM&FBovespa. As opções são classificadas em opções de compra (call) e de venda (put). Dependendo da forma de negociação, são classificadas nos tipos europeias, americanas e exóticas. No Brasil temos, além das citadas, um tipo especial, misto de opção europeia com americana, chamada de opção do tipo brasileiro. Na tese nos limitamos a trabalhar com as opções do tipo europeia.

Definição 3: Opções de Compra (Calls)

Tipo Europeia: Neste tipo, o comprador do contrato de opção, denominado de Titular, tem o direito de comprar o ativo-objeto pelo preço de exercício

contratado, mas somente na data de vencimento do contrato. No entanto, não é obrigado a exercer o seu direito.

Preço de Exercício (PE): preço contratado do ativo-objeto, no ato da negociação, pelo qual o Titular pode exercer o seu direito.

Preço a Vista (PVt): preço do ativo-objeto no tempo t , varia de valor no período do contrato $[0, T]$. Em $t = 0$ é chamado de preço inicial e em $t = T$ é chamado de preço a vista no vencimento do contrato. É de fundamental importância, pois define a rentabilidade da opção e influencia nas decisões dos Titulares.

Tipo Americana: Neste tipo o Titular tem o direito, mas não a obrigação, de comprar o ativo-objeto, pelo preço de exercício contratado, e em qualquer data entre o início e o vencimento do contrato.

Definição 4: Opções de Venda (Puts)

Tipo Europeia: O Titular do contrato de opção, tem o direito de vender o ativo-objeto pelo preço de exercício contratado, mas somente na data de vencimento do contrato. No entanto, não é obrigado a exercer o seu direito.

Tipo Americana: O Titular tem o direito, mas não a obrigação, de vender o ativo-objeto, pelo preço de exercício contratado, e em qualquer data entre o início e o vencimento do contrato.

Definição 5: Preço da Opção

O preço da opção é o valor ou prêmio que o Titular paga ao vendedor (Lançador) da opção, Call ou Put, pelo direito de exercer a compra ou a venda do ativo-objeto. Observe que, ao se comprar uma opção, não se está comprando o ativo-objeto e sim um direito de compra.

Lançador: é o vendedor do contrato de opção. Recebe o prêmio, pago pelo Titular, e passa a ter somente a obrigação de vender o ativo-objeto (no caso de Call), pelo preço de exercício, e de comprar (no caso de Put).

Definição 6: Ganhos e Perdas na Negociação de Calls e Puts – Opções Dentro, No e Fora do Dinheiro

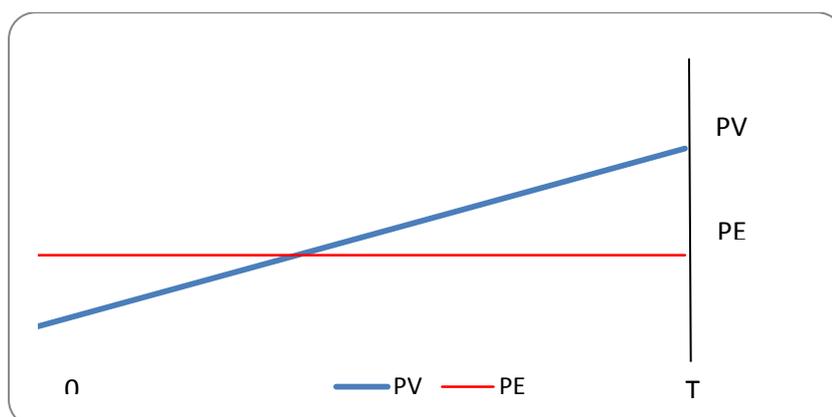
Dentro do Dinheiro (In-the-Money): são opções que possibilitam ao Titular exercer com ganho o seu direito de compra ou de venda da opção. Ou seja, Calls com PV acima do PE e Puts com PV abaixo do PE.

No Dinheiro (At-the-Money): são opções que tornam o Titular indiferente em exercer ou não o seu direito de compra ou de venda da opção. Ou seja, Calls e Puts quando $PV = PE$.

Fora do Dinheiro (Out-of-the-Money): são opções em que o Titular terá prejuízo se exercer o seu direito de compra ou de venda da opção. Ou seja, Calls com $PV < PE$ e Puts com $PV > PE$.

#

O Problema da Precificação de Opções



Conforme mostra o gráfico acima, o problema da precificação de uma opção pode ser visto como um problema de previsão do preço a vista, feita na data inicial do contrato, com horizonte para a data de vencimento. Isto é, qual será o PV do ativo-objeto na data de vencimento da opção? Qual será a diferença entre PV e PE na data do vencimento? Portanto, quanto se deve pagar, hoje, por um direito que será realizado no futuro e ao preço PE? Qual é o valor justo e como determiná-lo? Estas são questões que o modelo de precificação tentará respondê-las.

4.2.5.2

Modelo de Precificação de Opções Europeias

A formulação do modelo que será apresentado segue as mesmas premissas teóricas de finanças, adotadas na maioria dos modelos precificação de opções, mas com diferente hipótese para a EDE que determina a dinâmica dos preços do ativo-objeto. Esta diferença torna o modelo mais geral, mas pode ocasionar maior dificuldade no processo de estimação. No entanto, poderá produzir resultados mais realistas e precisos. O modelo é definido sob as seguintes hipóteses:

Hipóteses do Modelo

H₁. A dinâmica dos preços do ativo-objeto é determinada pela seguinte EDE na forma tradicional:

$$dP(t) = a(P(t), t) dt + b(P(t), t) dW(t)$$

onde,

$P(t)$: preço nominal da ação-objeto, evidentemente, positivo para todo t ;

$dP(t)$: variação instantânea do preço da ação-objeto;

$a(P(t), t)$: tendência ou Drift do modelo;

$b(P(t), t)$: coeficiente do processo de difusão $dW(t)$, positivo para todo t ;

$W(t)$: é o processo de Wiener.

H₂. O retorno esperado do ativo-objeto é igual a taxa de juros livre de risco. Considerada conhecida e constante durante todo o período de maturação. Ou seja, neutralidade ao risco.

H₃. Não há custos de transação ou impostos. Também não há pagamento de dividendos durante o período de maturação da opção.

H₄. Os ativos são infinitamente divisíveis e as transações ocorrem de forma contínua ao longo da vida da opção.

H₅. O mercado é eficiente no sentido de não admitir oportunidade de arbitragem sem risco, conforme sugere a hipótese 2.

H₆. A opção precificada é do tipo europeia.

Com base nas hipóteses acima, a fórmula de cálculo do prêmio justo de uma opção de compra do tipo europeia é dada pela seguinte expressão:

$$C_0(P(T), T, PE) = e^{-rT} E(\text{Max}\{[P(T) - PE], 0\})$$

Pode-se entender a fórmula acima como sendo: o valor da opção de compra, com taxa de juros livre de risco (r), contratada na data inicial (0), com vencimento na data (T) e preço de exercício (PE). Usando a paridade existente entre calls e puts, pode-se expressar a valor da opção de venda da seguinte forma:

$$V_0(P(T), T, PE) = C_0(P(T), T, PE) - [P_0 - PE e^{-rT}]$$

Que tem leitura semelhante a da call, ou seja, valor da opção de venda europeia, com taxa de juros livre de risco (r), contratada na data inicial (0), com vencimento na data (T) e preço de exercício (PE).

Analisando as fórmulas definidas para a call e a put, verifica-se que ambas dependem da equação do preço do ativo-objeto ($P(T)$) para que sejam factíveis de cálculo. Isto é, cada equação determina uma específica fórmula de cálculo para calls e puts. Portanto, não existe um modelo geral que possa ser usado, corretamente, para todos os tipos de comportamentos de preços de ativos.

Utilizando as soluções contínuas obtidas por PGDA para os preços das ações PETR4 e VALE5, equações (13) e (15), impondo as hipóteses do modelo às equações de preço e modificando a notação de (X para P), chega-se as seguintes expressões para as fórmulas das calls e puts¹⁶:

- i) Fórmulas dos valores da call e da put para PETR4, utilizando a equação contínua, eq (13), estimada por PGDA:

$$C_0(P(T), T, PE) = P_0 e^{\frac{\sigma^2(T)T}{2}} \Phi(z_1) - PE e^{-rT} \Phi(z_2).$$

$$\text{onde, } \sigma^2(T) = 0,00015 + 0,00037 \left(\frac{1 - 0,904045^T}{T} \right)$$

$$z_1 = \frac{\ln\left(\frac{P_0}{PE}\right) + (rT + \sigma^2(T)T)}{\sigma(T)\sqrt{T}} \quad \text{e} \quad z_2 = \frac{\ln\left(\frac{P_0}{PE}\right) + rT}{\sigma(T)\sqrt{T}},$$

¹⁶ O desenvolvimento matemático da determinação das fórmulas de calls e puts é apresentado na seção 7.5 do apêndice.

Φ : é a função de distribuição acumulada da $N(0, 1)$. Abaixo, usando a paridade, tem-se a fórmula da put.

$$V_0(P(T), T, PE) = PE e^{-rT} \Phi(-z_2) - P_0 \left[1 - e^{\frac{\sigma^2(T)T}{2}} \Phi(z_1) \right].$$

ii) Fórmulas dos valores da call e da put para VALE5, utilizando a equação contínua, eq. (15), estimada por PGDA: (as fórmulas são semelhantes as obtidas para a PETR4, com diferença somente na fórmula da variância).

$$C_0(P(T), T, PE) = P_0 e^{\frac{\sigma^2(T)T}{2}} \Phi(z_1) - PE e^{-rT} \Phi(z_2).$$

$$\text{onde, } \sigma^2(T) = 0,000152 + 0,00025 \left(\frac{1 - 0,86878^T}{T} \right)$$

$$z_1 = \frac{\ln\left(\frac{P_0}{PE}\right) + (rT + \sigma^2(T)T)}{\sigma(T)\sqrt{T}} \quad \text{e} \quad z_2 = \frac{\ln\left(\frac{P_0}{PE}\right) + rT}{\sigma(T)\sqrt{T}},$$

Φ : é a função de distribuição acumulada da $N(0, 1)$. Abaixo, usando a paridade, tem-se a fórmula da put.

$$V_0(P(T), T, PE) = PE e^{-rT} \Phi(-z_2) - P_0 \left[1 - e^{\frac{\sigma^2(T)T}{2}} \Phi(z_1) \right].$$

#

A seguir, utilizando as fórmulas de calls e puts, acima definidas, apresenta-se um exercício de precificação de opções, para opções europeias cujo ativo-objeto são as ações PETR4 e VALE5. Foram selecionadas do banco de dados da BM&FBovespa vinte opções, das quais, dez são opções da Petrobras e dez da Vale, sendo cinco opções de compra e cinco opções de venda de cada empresa. Para comparar os resultados obtidos pelo modelo de precificação, que utiliza as equações de preço estimadas por PGDA, utilizou-se modelo de Black & Scholes na sua forma tradicional, com volatilidade constante, e uma variante que utiliza volatilidade variável e estimada pelo modelo GARCH. Como Proxy para a taxa de juros livre de risco (r), utilizou-se a taxa de juros Selic de 12,25% a.a que, junto com o CDI, são as taxas frequentemente utilizadas neste tipo de exercício.

Os resultados são apresentados nos Quadros 11 e 12, um para a precificação de opções de compra e o outro para opções de venda. A parte superior de cada quadro apresenta as informações básicas do contrato de opção, necessárias para o

seu entendimento e para a aplicação dos modelos de precificação. Cada opção é denotada pela sua própria sigla, cuja decodificação é encontrada na seção 7.5 do apêndice. Conforme mostram os quadros, as opções possuem diferentes valores, datas de contratação, preços de exercício, tempo de maturidade e data de vencimento.

Antes de comentar os resultados, deve-se lembrar de que a finalidade do modelo de precificação não é a de prever o valor que a opção será negociada no mercado, mas sim de determinar um valor justo que sirva de parâmetro para a negociação do ativo. Por exemplo: se um lançador emite uma opção de compra, com preço de exercício muito acima do preço que o ativo-objeto está registrando no mercado, e com tendência de manutenção ou de queda, provavelmente, grande parte dos titulares não estará disposta a pagar um elevado valor pelo direito de comprar este ativo. Com isso, o preço justo deste direito, que é o prêmio justo da opção, tende a ser baixo, pois a probabilidade do titular não exercer o seu direito de compra é muito elevada. A mesma interpretação pode ser dada ao prêmio justo da opção de venda, neste caso, basta considerar o preço de exercício muito abaixo do preço do ativo-objeto. Contudo, muitos apostam na reversão das tendências, principalmente, no longo prazo, e acabam realizando ganhos expressivos.

Esta racionalidade econômica verifica-se nos Quadros 11 e 12, onde os três modelos, com metodologias diferentes, estimam baixos valores para as opções de compra e venda. Isso ocorre, pois, na maioria dos casos de opções de compra, o preço de exercício (PE) está acima do preço inicial (P0) e, na maioria dos casos de opções de venda, está abaixo. Mesmo trazendo o (PE) a preço presente, com a taxa de desconto (r), o valor justo da opção ainda é muito baixo. Em termos gerais, o que os modelos indicam é que os valores de mercado das opções estão muito elevados e não condizem com o comportamento claro de tendência de queda que apresentam os ativos subjacentes, PETR4 e VALE5. E nesse aspecto, as estimativas do modelo de PGDA, cujas volatilidades são estimadas conjuntamente com as tendências dos preços, são ainda mais conservadoras e realistas que as dos modelos de Black & Scholes e GARCH.

Quadro 11: Estimativas de Prêmios Justos de Opções de Compra

Resultados dos Exercícios de Precificação de Opções Europeias						
1) Opções de Compra da Petrobras e da Vale, com Ativo-Objeto: PETR4 e VALE5						
Sigla da Opção	Valor da Opção na Data do Contrato	Preço de Exercício PE	Preço do Ativo-Objeto na Data da Inicial - P0	Maturidade em dias úteis - T	Data do Contrato de Opção	Data de Vencimento do Contrato
PETRK15E	2,86	14,91	16,61	18	22/10/14	17/11/14
PETRL17E	2,39	16,91	16,61	38	22/10/14	15/12/14
PETRC1E	0,14	12,41	10,00	29	03/02/15	16/03/15
PETRF70E	1,61	10,75	10,20	93	04/02/15	15/06/15
PETRL12E	0,72	12,25	8,18	232	30/01/15	21/12/15
VALEC71E	1,07	17,60	17,82	28	04/02/15	16/03/15
VALED75E	0,07	24,10	18,05	54	03/02/15	20/04/15
VALEF78E	1,24	18,75	16,96	128	17/12/14	15/06/15
VALEH91E	0,89	21,86	18,05	139	03/02/15	17/08/15
VALEH5E	0,98	26,36	16,60	433	17/12/14	15/08/16
Sigla da Opção	Estimativa do Valor Justo da Opção (em R\$)					
	Modelo de Black-Scholes	Modelo GARCH		Modelo de PGDA		
PETRK15E	1,93	2,23		1,85		
PETRL17E	0,83	1,82		0,54		
PETRC1E	0,0371	0,0776		0,0004		
PETRF70E	0,91	0,98		0,48		
PETRL12E	0,419	0,472		0,054		
VALEC71E	0,93	0,95		0,75		
VALED75E	0,0268	0,0295		0,0013		
VALEF78E	1,12	1,14		0,72		
VALEH91E	0,730	0,75		0,34		
VALEH5E	1,09	1,12		0,50		

Quadro 12: Estimativas de Prêmios Justos de Opções de Venda

Resultados dos Exercícios de Precificação de Opções Europeias						
2) Opções de Venda da Petrobras e da Vale, com Ativo-Objeto: PETR4 e VALE5						
Sigla da Opção	Valor da Opção na Data do Contrato	Preço de Exercício PE	Preço do Ativo-Objeto na Data da Inicial - P0	Maturidade em dias úteis - T	Data do Contrato de Opção	Data de Vencimento do Contrato
PETRO14E	0,28	8,21	10,20	28	04/02/15	16/03/15
PETRP40E	1,31	10,66	10,20	53	04/02/15	20/04/15
PETRS64E	1,29	14,16	18,35	207	03/10/14	20/07/15
PETRU62E	1,30	13,25	12,60	219	17/11/14	21/09/15
PETRX40E	1,50	10,50	12,88	473	26/02/14	21/12/15
VALEO15E	0,22	15,35	18,05	29	03/02/15	16/03/15
VALET18E	1,57	18,11	18,05	139	03/02/15	17/08/15
VALEM86E	0,95	16,00	17,82	248	04/02/15	18/01/16
VALEQ71E	1,10	17,60	17,82	73	04/02/15	18/05/15
VALEX3E	1,79	25,11	25,12	387	27/06/14	21/12/15
Sigla da Opção	Estimativa do Valor justo da Opção (em R\$)					
	Modelo de Black-Scholes	Modelo GARCH	Modelo de PGDA			
PETRO14E	0,016	0,041	0,023			
PETRP40E	0,82	0,91	0,50			
PETRS64E	0,41	0,49	0,31			
PETRU62E	1,41	1,50	0,76			
PETRX40E	0,61	0,69	0,52			
VALEO15E	0,024	0,028	0,044			
VALET18E	1,01	1,03	0,71			
VALEM86E	0,56	0,58	0,50			
VALEQ71E	0,72	0,74	0,48			
VALEX3E	1,56	1,60	1,37			

Diante dos resultados obtidos no exercício e por oferecer uma forma mais geral de especificação da EDE dos preços, acredita-se que o modelo de PGDA, embora necessite ser mais testado, é uma alternativa promissora para ser utilizada na precificação de opções europeias.

5

Conclusões e Trabalhos Futuros

O projeto de tese, de acordo com os seus objetivos, procurou investigar o potencial de algoritmos computacionais evolutivos, construídos a partir das técnicas de programação genética, combinadas com programas de diferenciação automática, na tentativa de obtenção de soluções analíticas, exatas ou aproximadas, para problemas de equações diferenciais ordinárias, parciais e estocásticas. Com essa finalidade, utilizando-se o ambiente de programação Matlab, diversos algoritmos foram elaborados e a metodologia proposta foi testada em diferentes aplicações que envolvem os três tipos de equações diferenciais.

Os primeiros exercícios realizados, apresentados nas seções (4.1.1) e (4.1.2), tiveram o objetivo de testar a metodologia proposta na resolução de EDOs e EDPs. Muitos exercícios, de diferentes graus de dificuldade, foram realizados com este fim, mas reportou-se apenas onze resultados que se considerou ser o suficiente para representar o conjunto de equações abordadas. Os resultados obtidos foram muito animadores, pois se obteve solução analítica exata em todos os casos. Inclusive, para cinco dos exemplos desenvolvidos não se encontrou solução analítica pelos módulos de solução simbólica dos programas Mathematica e Matlab.

Um caso especial de EDP foi tratado na seção (4.1.3), problema de elevada complexidade e de difícil solução, a equação de Schrödinger para o átomo de He. Infelizmente, não foi possível obter a solução exata da função de onda do problema, mas três soluções analíticas aproximadas, com baixíssimos erros, foram encontradas. Além disso, estimativas da energia no seu estado fundamental (E_0), autovalor do problema, foram obtidas em termos médios, com baixo nível de erro, mas com desvio padrão elevado para o resultado que o problema requer. Acredita-se que as soluções aproximadas obtidas para o problema, dada a sua complexidade, são razoáveis e podem ser úteis para pesquisadores que trabalham com o tema. Tem-se a certeza de que o problema da equação de Schrödinger requer uma melhor formulação matemática, que defina de forma mais clara e

específica as propriedades e as condições de contorno da função de onda na sua forma estacionária. Com isso, acredita-se que a tentativa de solução do problema se torne mais fácil de ser abordada e, provavelmente, melhores resultados serão produzidos.

Na seção (4.1.4) é desenvolvido um exercício que se acredita ser de grande interesse, principalmente, para a estimação estatística e econométrica de modelos de regressão. O exercício mostra como a metodologia de PGDA pode ser extremamente útil na etapa de especificação do melhor modelo a ser estimado. Os resultados obtidos são muito promissores, pois o verdadeiro modelo simulado é totalmente recuperado e a metodologia identifica corretamente a forma funcional que deve ser utilizada na modelagem. Este resultado é importante, pois oferece uma forma alternativa de especificação de modelos de regressão, que, quase sempre, são especificados de forma arbitrária e com forma funcional linear.

Na seção (4.2) é proposto um algoritmo de PGDA para obter soluções analíticas de EDEs definidas na forma tradicional. O algoritmo foi desenvolvido com base no cálculo estocástico de Itô e determina a solução do problema em duas etapas. Quatro exercícios foram desenvolvidos para avaliar a eficiência do algoritmo. Os dois primeiros, apresentados nas seções (4.2.2) e (4.2.3), foram desenvolvidos com dados simulados para a EDE do movimento browniano geométrico. Os resultados são muito promissores, pois, nos dois exemplos, o algoritmo especificou corretamente a forma da EDE simulada e obteve a solução analítica exata da EDE especificada.

Nos dois últimos exemplos, apresentados nas seções (4.2.4) e (7.4), foram modelados os preços das ações PETR4 e VALE5. O algoritmo produziu soluções analíticas discretizadas e contínuas para as EDEs especificadas para os log-retornos destas ações. Contudo, as soluções obtidas são aproximadas, pois a solução exata de cada equação depende de uma integral de Itô com solução desconhecida. No entanto, as estimativas obtidas apresentam boa qualidade de ajuste, com baixas medidas de erros, e com dinâmicas de comportamento, para preços e volatilidades, semelhantes às estimadas pelos modelos ARCH, GARCH e EGARCH, usados para efeito de comparação.

A última aplicação realizada, apresentada na seção (4.2.5), propõe uma nova metodologia de precificação de opções europeias que difere dos demais modelos existentes, basicamente, na forma de definição e especificação da equação

diferencial que rege a dinâmica dos preços do ativo-objeto. Utilizando as equações de preços obtidas nas seções (4.2.4) e (7.4), foram realizados exercícios de precificação de vinte opções europeias da Petrobras e da Vale, cujos ativos subjacentes são, respectivamente, as ações PETR4 e VALE5. Os resultados obtidos são consistentes com a tendência atual de queda dos preços dos ativos subjacentes, onde o modelo precifica valores justos para as opções, abaixo do preço oferecido pelo mercado. O mesmo ocorre com as precificações realizadas pelos modelos de Black & Scholes e GARCH, embora as precificações do modelo de PGDA sejam ainda mais conservadoras e mais próximas da realidade atual destas empresas e das expectativas de crescimento destes mercados.

Portanto, um número expressivo e variado de aplicações da metodologia de PGDA foi desenvolvido com a finalidade de solucionar problemas que envolvem equações diferenciais. Com base nos resultados obtidos e nas análises realizadas, concluímos esta tese com a convicção de que a metodologia proposta pode ser uma alternativa eficiente na abordagem de problemas científicos complexos.

Como sugestões de trabalhos futuros, temos as seguintes:

- 1) Embora o algoritmo de PGDA apresente bom desempenho em ambiente de programação Matlab, para eliminar algumas limitações e melhorar ainda mais a sua performance, sugere-se que o algoritmo seja desenvolvido em linguagem computacional mais adequada, do tipo C++ ou Fortran.
- 2) Tentar generalizar o algoritmo de DA, de formas que o mesmo possa calcular derivadas de qualquer ordem ou de ordens mais elevadas;
- 3) Desenvolver uma formulação matemática mais completa para a função de onda da equação de Schödinger para o He, que facilite a abordagem da EDP, e que seja consistente com as exigências teóricas da Física do problema. Com isso, como dito anteriormente, acredita-se que a tentativa de solução do problema se torne mais factível e, provavelmente, melhores resultados serão alcançados.
- 4) Através de simulações, tentar demonstrar que o algoritmo de PGDA, quando aplicado na resolução de modelos paramétricos, com *fitness* que minimiza a

soma de quadrados de resíduos ponderados, equivale a uma generalização do método de mínimos quadrados, mas com a vantagem de não necessitar arbitrar a priori a forma funcional do modelo. Esta é uma extensão importante, pois formaliza o algoritmo de PGDA como um método estatístico de estimação e com as mesmas propriedades dos métodos tradicionais. Com isso, seus resultados poderão ser avaliados e testados por critérios estatísticos.

- 5) Aplicação do algoritmo de PGDA na resolução de sistemas de equações diferenciais estocásticas. Por exemplo: em finanças, para a solução de modelos de volatilidade estocástica, os quais envolvem duas EDEs; outra aplicação importante é em problemas que envolvem cópulas.

Referências bibliográficas

AIT-SAHALIA, Y. **Maximum-Likelihood Estimation of Discretely-Sampled Diffusions: A Closed-Form Approximation Approach.** *Econometrica*, 70, 223–262. 2002.

AIUBE, F. A. L. **Modelos Quantitativos em Finanças com enfoque em commodities.** Porto Alegre: Bookman, 2013.

ARNOLD, L. **Stochastic Differential Equations: Theory and Applications.** Wiley, New York, 1974.

BAALOUSHA, H. and KÖNGETER, J. **Stochastic modelling and risk analysis of groundwater pollution using FORM coupled with automatic differentiation.** *Advances in Water Resources*, 29 (March - 2006) 1815–1832.

BANZHAF; W. NORDIN, P.; KELLER, R. E. and FRANCONI, F. D., **Genetic Programming an introduction.** Morgan Kaufmann, 1998.

BANZHAF, W. and LANGDON, W. B. **Some Considerations on the Reason for Bloat.** *Genetic Programming and Evolvable Machines*, v. 3, n. 1, pp. 81–91, 2002.

BARBOSA, H. J. C. **Algoritmos genéticos para otimização em engenharia: uma introdução.** In: IV Seminários sobre Elementos Finitos e Métodos Numéricos em Engenharia, 1996, Juiz de Fora, MG. Anais. . . [S.l.: s.n.], 1996.

BERA, A. K. and JARQUE, C. M. **An efficient large sample test for normality of observations and regression residuals.** Working Paper in Econometrics, 40, Australian National University, Canberra. 1981.

BISCHOF, C. et. al. **ADIFOR – generating derivative codes from fortran programs.** *Scientific Programming*, no. 1, pp. 1–29, 1992.

BLACK, F. and SCHOLES, M. **The pricing of options and corporate liabilities.** *J. Political Economy* 81: 637-654. (1973).

BOHM, W. and GEYER-SCHULZ, **A Exact uniform initialization for genetic program.** *Foundations of Genetic Algorithms IV.* p 379-407. Morgan Kaufmann, 1996.

BOLLERSLEV, T. **Generalized Autoregressive Conditional Heteroskedasticity.** *Journal of Econometrics*, 31, 307-327, 1986.

BOLLERSLEV, T.; Engle, R. and Nelson, D. **ARCH Models.** In R. Engle and D. McFadden(eds). *Handbook of Econometrics*, Vol. IV, Elsevier, Amsterdam, 1994.

BOYCE W. E. and DIPRIMA R. C. **Elementary Differential Equations and Boundary Value Problems.** John Wiley & Sons, Inc. 1977.

BOYER, Carl. **História da Matemática.** Edgard Blucher, São Paulo, 1996.

BRAUMANN, C. A. **Introdução às Equações Diferenciais Estocásticas e Aplicações**. XIII Congresso Anual da Sociedade Portuguesa de Estatística, Ericeira, Edições SPE, 2005.

BRAUN, M. **Differential Equations and Their Applications**, Springer-Verlag, 1975.

BRITO, M. A. **QMC 3111 – Química Inorgânica Avançada**. UFSC – Departamento de Química. 2012.

BROTO, C. and E., R. **Estimation methods for stochastic volatility methods: A survey**. Journal of Economic Surveys, v. 18(5), p. 613–649, 2004.

BURGESS, G. **Finding approximate analytic solutions to differential equations using genetic programming**. Surveillance Systems Division, Electronics and Surveillance Research Laboratory, Department of Defense, Australia, 1999.

CAO, H. et. al. **Evolutionary modeling of systems of ordinary differential equations with genetic programming**. Genetic Programming and Evolvable Machines, vol. 1, pp. 309–337, 2000.

CAMPBELL, J. Y.; LO, A. W. and MacKINLAY, A. C. **The Econometrics of Financial Markets**. Princeton University Press, New Jersey. 1997.

CHELLAPILLA, K. **Evolutionary programming with tree mutations: Evolving computer programs without sub-tree crossover**. Genetic Programming. In: Proc. Second Annual Conference of San Francisco, CA. Morgan Kauffmann, p. 432-438, 1997.

COLEMAN, T. and VERMA, A. **ADMAT: An automatic differentiation toolbox for MATLAB**. In Proceedings of the SIAM Workshop on Object Oriented Methods for Inter-Operable Scientific and Engineering Computing, SIAM, Philadelphia, PA. 1998a.

COURANT, R; Robbins, H. **O Que é a Matemática?** Ciência Moderna, Rio de Janeiro, 2000.

CUSDIN, P. and MULLER, J. **Automatic differentiation and sensitivity analysis methods for CFD**. QUB School of Aeronautical Engineering, 2003.

DARWIN, C. **A origem das Espécies e a Seleção Natural**. Ed. Hemus, 5ª Ed., 2000.

DAVID, C. W. **A Review of Helium Hamiltonians**. Chemistry Education Materials. Paper 78. 2009. <http://digitalcommons.uconn.edu>.

DIAS, DOUGLAS M. and PACHECO, MARCO A. C. **Quantum-Inspired Linear Genetic Programming as a Knowledge Management System**. The Computer Journal 01/2013; 56(9):1043-1062. · 0.76 Impact Factor.

EVES, Howard. **Introdução à História da Matemática**. Editora Unicamp, Campinas 1990.

EFRON, B., and TIBSHIRANI, R. **An Introduction to the Bootstrap**. London: Chapman and Hall. 1993.

FIGUEIRA, A. F. **Utilização de programação genética para achar soluções analíticas em equações diferenciais.** Departamento de Engenharia Elétrica – PUC-Rio. 2011.

FIGUEIREDO, DJAIRO G. DE. **Análise de Fourier e equações diferenciais parciais.** IMPA, CNPQ, (Projeto Euclides). Rio de Janeiro, 1977.

FINK, M. **Automatic Differentiation for MATLAB, 2007.** MATLAB Central, www.mathworks.com/matlabcentral. (Martin Fink, Director of Hewlett-Packard - HP Labs research.)

GEYER-SCHULTZ, A. **Fuzzy rule-based expert systems and genetic machine learning.** Heidelberg: Physica-Verlag, 1997.

GHYSELS, E.; HARVEY, A. C.; and RENAULT, E. **Statistical methods in finance.** In: . [S.l.]: North Holland, 1996. cap. Stochastic Volatility, p. 221–238.

GIESECKE, K. and SMELOV, D. **Exact Sampling of Jump Diffusions.** Operations Research 61(4):894-907. (2013).

GOLDBERG, D. E. **Genetic algorithms in search, optimization & machine learning.** [S.l.]: Addison-Wesley, 1989.

GPTIPS: An Open Source Genetic Programming Toolbox For Multigene Symbolic Regression.

GRIEWANK, A. **On automatic differentiation** in Mathematical programming: Recent Developments and Applications, M. Iri and K. Tanabe (eds.), Kluwer Academic Publishers, Amsterdam, pp. 83–108, 1989.

GRIEWANK, A. and CORLISS, G. F. **Automatic Differentiation of Algorithms: Theory, Implementation, and Application.** SIAM, Philadelphia, PA. Eds. 1991.

GRIEWANK, A. and WALTHER, A. **Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation.** (<http://www.ec-securehost.com/SIAM/OT105.html>). Other Titles in Applied Mathematics 105 (2nd ed.). SIAM. ISBN 978-0-89871-659-7. 2008.

HAMILTON, J. D. **Time Series Analysis.** Princeton University Press. Princeton. New Jersey. 1994.

HOLLAND, J. H. **Adaptation in natural and artificial systems.** Univ. of Michigan Press, AnnArbor, 1975.

HULL, J. **Options, Futures and Other Derivatives.** Prentice-Hall. 1997.

HYLLERAAS, E. A. (1939) Z Phys 54, 347.

IBA, H. and SAKAMOTO, E. **Inference of differential equation models by genetic programming.** Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2002), 2002, pp. 788–795.

IMAE, J. et al. **Design of nonlinear control systems by means of differential genetic programming.** 43rd IEEE, Conference on Decision and Control. Atlantis, Paradise Island, Bahamas. December 14-17, 2004.

INCE, E. L. **Ordinary Differential Equations.** Longmans, Green, London, 1927; Dover, New York, 1956.

ITÔ, K. **On Stochastic Differential Equations**. American Mathematical Society Memoirs No. 4, New York, 1951.

JINJUN, H. **Finding Approximate Analytic Solutions to Differential Equations by Seed Selection Genetic Programming**. 2007.

JOHNSTON, J. and DiNARDO, J. **Econometric Methods**. McGraw Hill Higher Education; 4th edition (July 1, 1997).

JUNGBACKER, B. and KOOPMAN, S. J. **Handbook of financial time series**. In: [S.l.]: Springer, 2009. cap. Parameter Estimation and Practical Aspects of Modeling Stochastic Volatility, p. 313–344.

KARATZAS, I. and SHREVE, S. E. **Methods of Mathematical Finance**. Springer, New York, 1998.

KOZA, J. R. **Hierarchical genetic algorithms operating on populations of computer programs**. Proceedings of the 11th International Joint Conference on Artificial Intelligent (IJCAI-89). Detroit, MI. Pp 768-774. Morgan Kaufmann, 1989.

KLOEDEN, P. E. and PLATEN, E. **Numerical Solution of Stochastic Differential Equations**. Springer, Berlin, 1992.

KLOEDEN, P. E., PLATEN, E. and SCHURZ, H. **Numerical Solution of SDE Through Computer Experiments**. Springer, Berlin, 1994.

KOZA, R. J. **Genetic Programming: On the Programming of Computers by Means of Natural Selection (Complex Adaptive Systems)**. 1992, Massachusetts Institute Technology. ISBN 0-262-11170-5.

KOZA, R. J. **Genetic Programming II: Automatic Discovery of Reusable Programs (Complex Adaptive Systems)**. 1994, Massachusetts Institute Technology. ISBN 978-0-262-11189-8. A Bradford Book, The MIT Press, Cambridge, Massachusetts, London, England.

LANGDON, W. B. and POLI, R. **Fitness Causes Bloat**. In: Second On-line World Conference on Soft Computing in Engineering Design and Manufacturing, pp. 13–22, July 1997.

LANGDON, W. B. et al. **The evolution of size and Shape**. In: Advances in genetic programming: volume 3, chap. 8, pp. 163–190, Cambridge, MA, USA, MIT Press, 1999.

LANGDON, W. B. and POLI, R. **Foundations of Genetic Programming**. Springer – Verlag, 2002. 260+xv pp. ISBN 3-540-42451-2.

LAURINI, M. P. e HOTA, L. K. **Inferência indireta em modelos fracionários de taxas de juros de curto prazo**. Insper Working Paper WPE: 128/2008.

LAURINI, M. P. e HOTA, L. K. **Estimação de Modelos de Volatilidade Estocástica usando Métodos de Verossimilhança Empírica/Mínimo Contraste Generalizados**. Insper Working Paper WPE: 192/2009.

LUKE, S. **Two fast tree-creation algorithms for genetic programming**. IEEE Transactions in Evolutionary Computation, v. 4, n.3, p. 274-283. IEEE Press. September, 2000.

LUKE, S. and PANAIT, L. **Lexicographic parsimony pressure**. In Langdon, W.B. et al., editors, Proceedings of GECCO-2002. Morgan Kaufmann (2002) 829–836

LUKE, S. and PANAIT, L. **A survey and comparison of tree generation algorithms**. Proceedings of the 6th Annual Conference in Genetic Programming (GECCO 2001). Springer-Verlag, 2001.

LUKE, S. and PANAIT, L. **A comparison of bloat control methods for genetic Programming**. *Evol. Comput.*, v. 14, n. 3, pp. 309–344, 2006.

MARINS, A. **Mercados Derivativos e Análise de Risco**. 2000.

MAKAROV, D. et al. **Using Genetic Programming To Solve the Schrödinger Equation**. 2000.

MCPHEE, N. F. and MILLER, J. D. **Accurate Replication in Genetic Programming**. In: Proceedings of the 6th International Conference on Genetic Algorithms, pp. 303–309, San Francisco, CA, USA, Morgan Kaufmann Publishers Inc., 1995.

MICHALEWICZ, Z. **Genetic algorithms + data structures = evolution programs**. Berlin: Springer-Verlag, 1999.

MILSTEIN, G. **Numerical Integration of Stochastic Differential Equations**. Mathematics and Its Applications. Springer, 1994.

MORETTIN, P. A. e TOLOI, C. M. C. **Análise de Séries Temporais**. ABE-Projeto Fisher. Editora Edgard Blücher. 2004.

MERTON, R. **Theory of rational option pricing**. *Bell J. Economics and Management Science* 4: 141-183. 1973.

NEIDINGER, R. D. H. **Introduction to Automatic Differentiation and MATLAB Object-Oriented Programming**. *Siam Review*, Society for Industrial and Applied Mathematics. Vol. 52, No. 3, pp. 545–563, 2010.

NICOLAU, J. C. H. C. **Modelação e Estimação de Séries Financeiras através de Equações Diferenciais Estocásticas**. Instituto Superior de Economia e Gestão Universidade Técnica de Lisboa. Tese de Doutorado. 2000.

ØKSENDAL, B. **Stochastic Differential Equations. An Introduction with Applications (sixth edition)**. Springer, Berlin, 2003.

PACHECO, M. A. C. **Algoritmos Genéticos: Princípios e Aplicações**. ICA: Laboratório de Inteligência Computacional Aplicada – DEE – PUC-RJ, 1999.

PACHECO, M. A. C. e VELASCO, M. M. B. R., **Sistemas Inteligentes de Apoio à Decisão: Análise Econômica de Projetos de Desenvolvimento de Campos de Petróleo sob Incerteza**. Série Business Intelligence. ISBN: 978-85-7193-172-5 (brochura), ISBN: 978-85-7193-173-2 (cartonada), 300 págs, Série Business Intelligence, Ed. Interciência e Ed. PUC-Rio, Junho 2007.

PEDERSON, A. R. **A New Approach to Maximum Likelihood Estimation for Stochastic**. *Scandinavian Journal of Statistics*, 22, 55-71. (1995).

PÉREZ SERRADA, A. **Una introducción a la computación evolutiva.** Disponível em <http://www.geocities.com/igoryepes/spanish.zip>. (Setembro de 2000).

POLI, R.; LANGDON, W. B. & McPhee, N. M. **A Field Guide to Genetic Programming.** 2008. ISBN 978-1-4092-0073-4 (softcover).

POLI, R. and MCPHEE, N. F. **Covariant Parsimony Pressure for Genetic Programming.** Tech. Rep. CES-480, Department of Computing and Electronic Systems, University of Essex, UK, 2008.

RALL, L. B. **Automatic Differentiation: Techniques and Applications.** Lecture Notes in Computer Science 120. Springer. ISBN 3-540-10861-0. (1981).

RICH, L. C. and HILL, D. R. 1992. **Automatic differentiation in MATLAB.** Appl. Numer. Math.9, 1 (Jan. 1992), 33–43.

SARSON, D. P. et al. **Equação de Schrödinger: Resolução analítica e simulação para o átomo de hidrogênio.** Proceedings of the International MultiConference of Engineers and Computer Scientists, 2010 – Vol I – IMECS – March – 2010, Hong Konk.

SCHWEFEL, H. **Evolution and optimum seeking.** Sixth-Generation Computer Technology Series. John Wiley & Sons. New York, 1995.

SCHUSS Z. **Theory and Applications of Stochastic Differential Equations.** John Wiley & Sons. 1980.

SHEPARD, N. and ANDERSEN, T. **Handbook of financial time series.** In: [S.l.]: Springer, 2009. cap. Stochastic Volatility: Origins and Overview, p. 233–254.

SILVA, S. **A Genetic Programming Toolbox for MATLAB, 2009.** Universidade Nova de Lisboa, PO Box 127, 2780-156 Oeiras, Portugal. sara@itqb.unl.ptl.

SOULE, T. and FOSTER, J. A. **Removal Bias: a New Cause of Code Growth in Tree Based Evolutionary Programming.** In: 1998 IEEE International Conference on Evolutionary Computation, pp. 781–186, Anchorage, Alaska, USA, IEEE Press, 5-9 May 1998.

STAUNING, O. **Flexible automatic differentiation using templates and operator overloading in C++.** Talk presented at the Automatic Differentiation Workshop at Shrivensham Campus, Cranfield University, June 6, 2003.

TSOULOS, I. G. and LAGARIS, I. E. **Solving differential equations with genetic programming.** Genetic Program Evolvable March (2006) 7: 33-54.

WALL, M. GALIB: **A C++ library of genetic algorithm components.** Massachusetts, EUA: [s.n.], 2000. Manual disponível em <http://lancet.mit.edu/ga/> (Agosto de 2000).

WHITLEY, D. **A genetic algorithm tutorial.** Setembro de 2000, disponível em http://www.geocities.com/igoryepes/ga_tutorial.zip.

YOUNG, E. C. **Partial Differential Equations: An Introduction.** Allyn and Bacon, Boston, 1972.

ZENGHU LI. **Stochastic Equations and Lamperti transformations.** The 8th Workshop on Markov Processes and Related Topics. Beijing and Wuyishan, July 16–21, 2012. (Beijing Normal University)

ZILL, D. G. **Equações Diferenciais com aplicações em modelagens.** Tradução da nona edição norte-americana. CENGAGE Learning, São Paulo, 2012.

7

Apêndice

7.1

Codes Matlab Utilizados nas Simulações Realizadas

7.1.1

Code para Simulação da EDE de MBG

O code abaixo foi utilizado na simulação dos dados da EDE de movimento browniano para o exemplo 1 da seção 4.2.1. O code utiliza a função “simBySolution” para obter os dados da solução $X(t)$. Para os demais exemplos de MBG foram desenvolvidos codes semelhantes, com mudanças apenas nos parâmetros.

```
% Code Matlab dos Modelos Desenvolvidos no Exemplo 1:  
% Simulação da Solução da EDE de Movimento Browniano Geométrico  
% MBG - EDE:  $dX(t) = 0,2X(t)dt + 0,5X(t)dW(t)$ :  
%           com  $T=4$ ;  $dt=0,004$  e  $n=1000$ .  
  
% Simulação da Solução do MBG pela Função simBySolution:  
nPeriods = 1000; % número de observações ou pontos;  
dt = 0.004; % intervalo de tempo fixo ==>  $T = 4$   
obj = gbm(0.2, 0.5, 'StartState', 20);  
strm = RandStream('mt19937ar','Seed',001);  
RandStream.setDefaultStream(strm);  
[X,T]=obj.simBySolution(nPeriods,'DeltaTime',dt,'nSteps',1);
```

7.1.2

Code de Solução da EDE de MBG por Aproximação de Euler

O code abaixo descrito foi utilizado para determinar a solução da EDE de MGB apresentada no exemplo 1 da seção 4.2.1. O code utiliza a função “simByEuler” para obter a solução de $X(t)$ pelo método de aproximação de Euler. Para os demais exemplos de MBG foram desenvolvidos codes semelhantes para a solução pelo método de Euler, com mudanças apenas nos parâmetros.

```
% Code Matlab dos Modelos Desenvolvidos no Exemplo 1:  
% Solução da EDE de MBG por Aproximação de Euler  
% MBG - EDE:  $dX(t) = 0,2X(t)dt + 0,5X(t)dW(t)$ :  
%           com  $T=4$ ;  $dt=0,004$  e  $n=1000$ .  
  
% Solução da EDE de MBG pela Função simByEuler:  
nPeriods = 1000; % número de observações ou pontos;  
dt = 0.004; % intervalo de tempo fixo ==> T = 4  
obj = gbm(0.2, 0.5, 'StartState', 20);  
strm = RandStream('mt19937ar','Seed',001);  
RandStream.setDefaultStream(strm);  
[Y,T]=obj.simByEuler(nPeriods,'DeltaTime',dt,'nSteps',1);
```

7.2

Code Matlab de Bootstrap para Cálculo de Erro Padrão

O code abaixo foi utilizado para calcular os erros padrão das estimativas dos coeficientes infinitesimais, da tendência e da volatilidade, da solução de PGDA para a EDE do exemplo 1. O code utiliza a função “bootstrap”, com dez mil reamostragens, para obter estimativas dos erros padrão. Para os coeficientes infinitesimais dos demais exemplos foram desenvolvidos codes semelhantes, mas com algumas modificações de equações e parâmetros.

```

% Code Matlab para o Cálculo dos Erros Padrão dos
% Coeficientes Infinitesimais da Tendência e da
% Volatilidade das Soluções de PGDA das EDEs de MBG:
% Solução da EDE de MBG por Aproximação de Euler
% Exemplo 1 - EDE:  $dX(t) = 0,2X(t)dt + 0,5X(t)dW(t)$ :
% com T=4; dt=0,004 e n=1000.

% T: Variável tempo (0, 0,004 0,008 ... 4)
% P1=0.0724
% P2=0.50198
% WT: Trajetória estimada do processo de Wiener;
x = [T,WT];% Matriz de variáveis explicativas;
yobs = log(X/20);% variável dependente, Y(t)-Y(0);
b = regress(yobs,x);%[P1, P2]';% estimativa dos coeficientes
yfit = (x*b);% Estimativas dos valores observados;
resid = yobs - yfit; % Resíduos das Estimativa;
se=std(bootstrap(10000,@(bootr)regress(yfit+bootr,x),resid));
% se: erros padrão das estimativas, realizadas com 10000
% reamostragens dos resíduos.

```

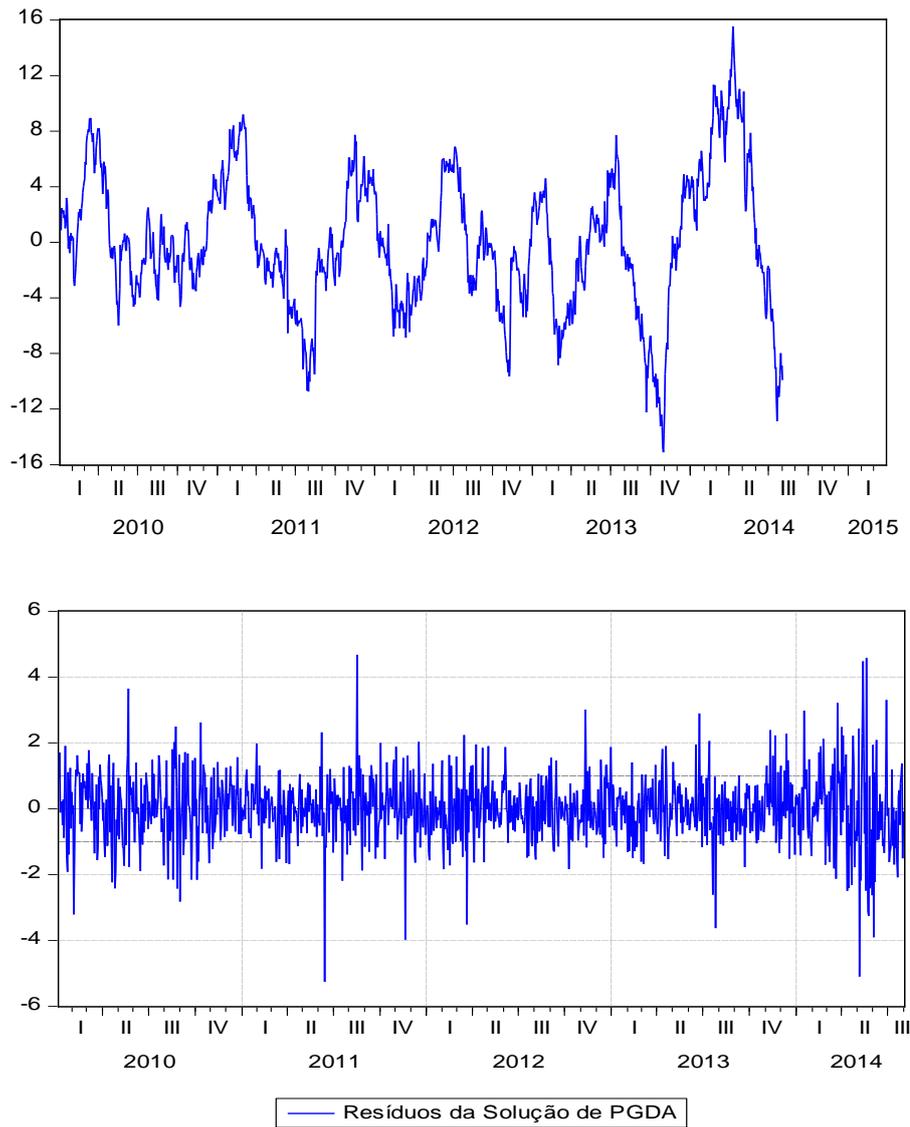
7.3

Resultados Adicionais dos Modelos Estimados para a PETR4

7.3.1

Análise dos Resíduos da Solução de PGDA para a PETR4

Processo de Wiener Estimado por PGDA para PETR4



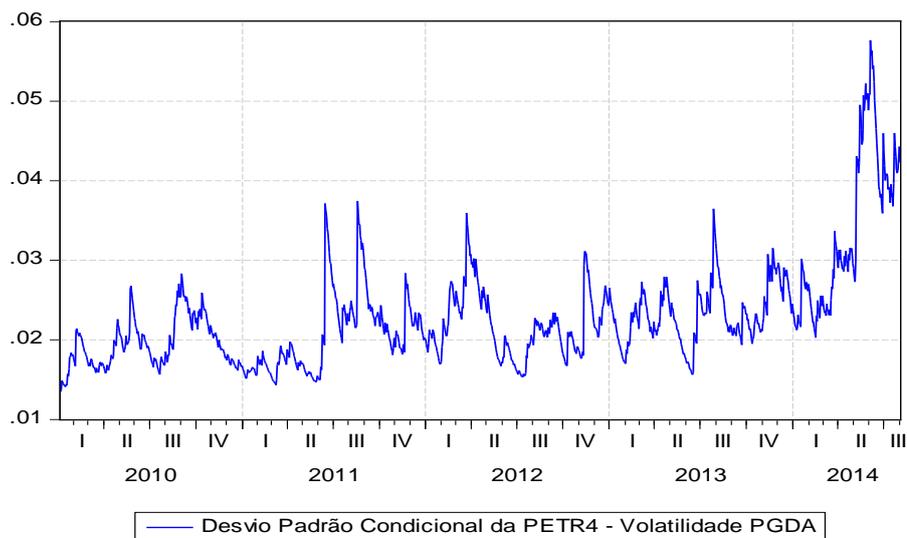
Correlograma dos Resíduos

Included observations: 1194

	Autocorrelation	Partial Correlation	AC	PAC	Q-Stat	Prob
1			0.029	0.029	1.0005	0.317
2			-0.009	-0.010	1.0919	0.579
3			-0.013	-0.012	1.2824	0.733
4			-0.004	-0.004	1.3066	0.860
5			-0.008	-0.008	1.3844	0.926
6			-0.043	-0.043	3.6165	0.728
7			0.014	0.016	3.8565	0.796
8			-0.017	-0.019	4.2142	0.837
9			0.004	0.004	4.2325	0.895
10			0.017	0.017	4.5866	0.917
11			0.021	0.019	5.1029	0.926
12			0.015	0.013	5.3771	0.944
13			-0.077	-0.076	12.573	0.481
14			0.026	0.030	13.377	0.497
15			0.024	0.022	14.047	0.522
16			-0.017	-0.018	14.380	0.570
17			-0.032	-0.029	15.590	0.553
18			0.007	0.009	15.644	0.617
19			0.053	0.046	19.020	0.456
20			0.000	0.001	19.020	0.521
21			-0.039	-0.042	20.856	0.468
22			0.013	0.015	21.054	0.517
23			-0.016	-0.016	21.373	0.558
24			-0.011	-0.008	21.520	0.608
25			0.002	0.006	21.525	0.663
26			0.027	0.019	22.448	0.664
27			0.032	0.035	23.706	0.647
28			-0.005	-0.001	23.736	0.695
29			0.005	-0.002	23.770	0.740
30			0.014	0.009	23.999	0.772
31			0.028	0.031	24.935	0.771
32			-0.017	-0.008	25.284	0.794

Heteroskedasticity Test: ARCH

F-statistic	0.583589	Prob. F(30,1133)	0.9648
Obs*R-squared	17.71298	Prob. Chi-Square(30)	0.9630



7.3.2

Resultados do Modelo ARCH para a PETR4

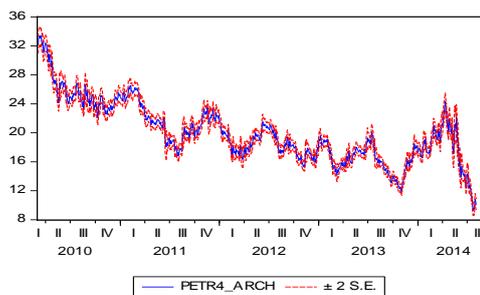
Dependent Variable: D(LOG(PETR4))
 Method: ML - ARCH (Marquardt) - Normal distribution
 Included observations: 1153 after adjustments
 Convergence achieved after 12 iterations
 Presample variance: backcast (parameter = 0.7)
 GARCH = C(3) + C(4)*RESID(-1)^2 + C(5)*RESID(-2)^2 + C(6)*RESID(-3)^2
 + C(7)*RESID(-4)^2 + C(8)*RESID(-5)^2

Variable	Coefficient	Std. Error	z-Statistic	Prob.
AR(13)	-0.103983	0.023190	-4.484041	0.0000
AR(41)	0.096657	0.025223	3.832107	0.0001

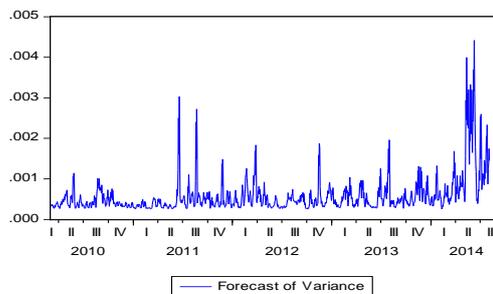
Variance Equation				
C	0.000246	2.07E-05	11.90873	0.0000
RESID(-1)^2	0.045996	0.020105	2.287728	0.0222
RESID(-2)^2	0.154912	0.039912	3.881386	0.0001
RESID(-3)^2	0.084803	0.026672	3.179499	0.0015
RESID(-4)^2	0.182764	0.026095	7.003880	0.0000
RESID(-5)^2	0.122746	0.026042	4.713442	0.0000

R-squared	0.006654	Mean dependent var	-0.001014
Adjusted R-squared	0.005791	S.D. dependent var	0.024144
S.E. of regression	0.024074	Akaike info criterion	-4.753138
Sum squared resid	0.667066	Schwarz criterion	-4.718098
Log likelihood	2748.184	Hannan-Quinn criter.	-4.739913
Durbin-Watson stat	1.955727		

PUC-Rio - Certificação Digital Nº 1121535/CA



Forecast: PETR4_ARCH
 Actual: PETR4
 Forecast sample: 1/04/2010 3/30/2015
 Adjusted sample: 3/03/2010 8/20/2014
 Included observations: 1153
 Root Mean Squared Error 0.454968
 Mean Absolute Error 0.335431
 Mean Abs. Percent Error 1.748827
 Theil Inequality Coefficient 0.011135
 Bias Proportion 0.001842
 Variance Proportion 0.000103
 Covariance Proportion 0.998055



7.3.3

Resultados do Modelo GARCH para a PETR4

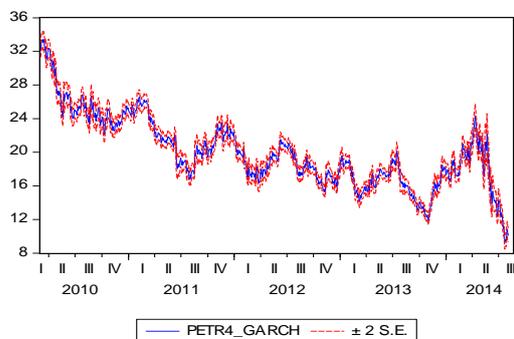
Dependent Variable: D(LOG(PETR4))
 Method: ML - GARCH (Marquardt) - Normal distribution
 Included observations: 1153 after adjustments
 Convergence achieved after 13 iterations
 Presample variance: backcast (parameter = 0.7)
 GARCH = C(3) + C(4)*RESID(-1)^2 + C(5)*GARCH(-1)

Variable	Coefficient	Std. Error	z-Statistic	Prob.
AR(13)	-0.102605	0.028409	-3.611678	0.0003
AR(41)	0.088160	0.026774	3.292681	0.0010

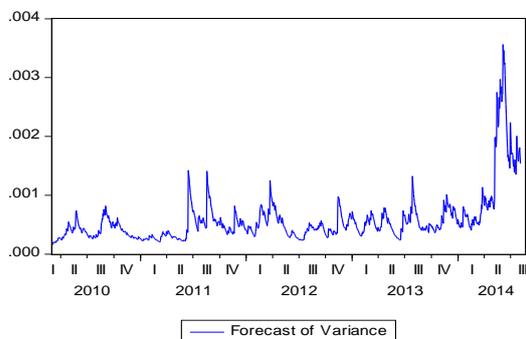
Variance Equation				
C	1.48E-05	3.27E-06	4.526941	0.0000
RESID(-1)^2	0.076552	0.012398	6.174260	0.0000
GARCH(-1)	0.900451	0.013780	65.34584	0.0000

R-squared	0.006861	Mean dependent var	-0.001014
Adjusted R-squared	0.005998	S.D. dependent var	0.024144
S.E. of regression	0.024071	Akaike info criterion	-4.767152
Sum squared resid	0.666927	Schwarz criterion	-4.745252
Log likelihood	2753.263	Hannan-Quinn criter.	-4.758886
Durbin-Watson stat	1.956433		

PUC-Rio - Certificação Digital Nº 1121535/CA



Forecast: PETR4_GARCH	
Actual: PETR4	
Forecast sample: 1/04/2010 3/30/2015	
Adjusted sample: 3/03/2010 8/20/2014	
Included observations: 1153	
Root Mean Squared Error	0.454860
Mean Absolute Error	0.335301
Mean Abs. Percent Error	1.748364
Theil Inequality Coefficient	0.011132
Bias Proportion	0.001856
Variance Proportion	0.000100
Covariance Proportion	0.998045



7.3.4

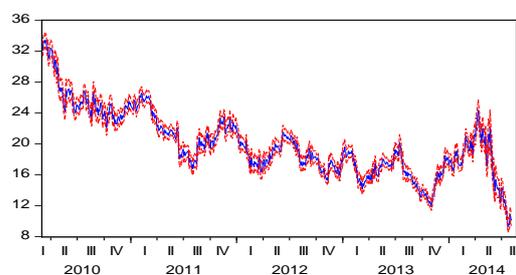
Resultados do Modelo EGARCH para a PETR4

Dependent Variable: D(LOG(PETR4))
 Method: ML - EGARCH
 Included observations: 1153 after adjustments
 Convergence achieved after 19 iterations
 Presample variance: backcast (parameter = 0.7)
 $LOG(GARCH) = C(3) + C(4)*ABS(RESID(-1)/@SQRT(GARCH(-1))) + C(5)*RESID(-1)/@SQRT(GARCH(-1)) + C(6)*LOG(GARCH(-1))$

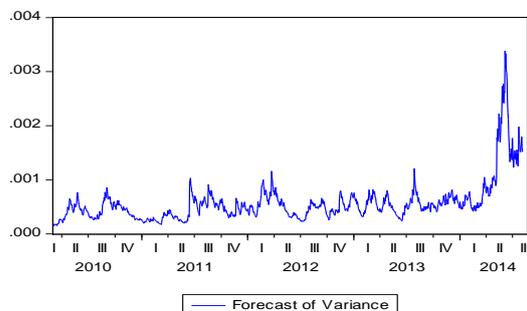
Variable	Coefficient	Std. Error	z-Statistic	Prob.
AR(13)	-0.104104	0.027762	-3.749869	0.0002
AR(41)	0.075617	0.028295	2.672402	0.0075

Variance Equation				
C(3)	-0.355693	0.058929	-6.035930	0.0000
C(4)	0.162501	0.026345	6.168294	0.0000
C(5)	-0.023318	0.015057	-1.548676	0.1215
C(6)	0.969274	0.006240	155.3379	0.0000

R-squared	0.006636	Mean dependent var	-0.001014
Adjusted R-squared	0.005773	S.D. dependent var	0.024144
S.E. of regression	0.024074	Akaike info criterion	-4.768222
Sum squared resid	0.667077	Schwarz criterion	-4.741942
Log likelihood	2754.880	Hannan-Quinn criter.	-4.758303
Durbin-Watson stat	1.957217		



Forecast: PETR4_EGARCH	
Actual: PETR4	
Forecast sample: 1/04/2010 3/30/2015	
Adjusted sample: 3/03/2010 8/20/2014	
Included observations: 1153	
Root Mean Squared Error	0.454834
Mean Absolute Error	0.335212
Mean Abs. Percent Error	1.748128
Theil Inequality Coefficient	0.011131
Bias Proportion	0.001884
Variance Proportion	0.000093
Covariance Proportion	0.998023



7.4

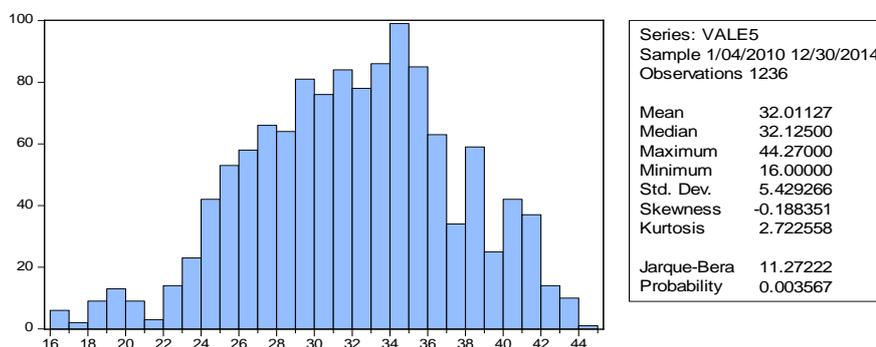
Resultados dos Modelos Estimados para a VALE5 - Exemplo 4

7.4.1

Resultados da Solução de PGDA para a VALE5

Neste exemplo apresenta-se uma aplicação da metodologia de PGDA na modelagem dos preços nominais da VALE5, ação preferencial nominativa (PN), pertencente a empresa VALE S.A, que, assim como a PETR4, é uma das ações mais negociadas na Bolsa de Valores de São Paulo – BM&FBovespa.

Os dados utilizados na modelagem são preços nominais de fechamento (ajustado), com periodicidade diária, coletados junto ao banco de dados da BM&FBovespa e, pelos mesmos motivos apresentados na modelagem da PETR4, referentes ao período dos últimos cinco anos, de 04-jan-2010 a 30-dez-2014, com um total de 1236 observações diárias, conforme mostra a Figura abaixo.



Para a realização do exercício foram definidos e utilizados os seguintes parâmetros para a solução da EDE na forma tradicional:

i) Modelo de EDE: $dX(t) = a(t, X(t)) dt + b(t, X(t)) dW(t)$

onde, $X(t)$: é o preço da ação VALE5 no tempo t ;

ii) Condição inicial: $X(0) = 35,47$; preço da ação em 04-jan-2010;

iii) Número de períodos ou observações: $n = 1236$;

iv) Variação infinitesimal do tempo: $dt = t_i - t_{i-1} = \Delta = 0,004$; (constante)

v) Intervalo de tempo: $[0, T] = [0, 5]$ (aproximadamente, cinco anos de dias úteis).

Aplicando-se os algoritmos de PGDA aos dados observados, obteve-se na primeira etapa, a seguinte estimativa para a especificação discretizada da EDE log-retornos:

$$Y_{t_i} - Y_{t_{i-1}} = \hat{a}_{t_{i-1}} \Delta + \hat{b}_{t_{i-1}} (\hat{W}_{t_i} - \hat{W}_{t_{i-1}})$$

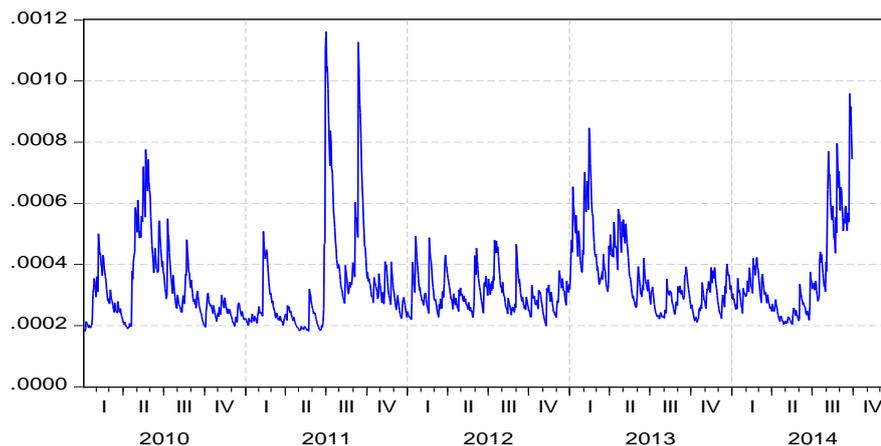
onde,

$$Y_{t_i} = \log(X_{t_i}); \hat{a}_{t_i} = -0,00000068 t_i; \Delta = 0,004;$$

$$\hat{b}_{t_k} = \sqrt{0,00002 + [0,07371 (\hat{W}_{t_{k-1}} - \hat{W}_{t_{k-2}})^2 + 0,86878]} \hat{b}_{t_{k-1}}^2$$

com $\hat{b}_0 = 0,000185$; e \hat{W}_{t_i} : é a estimativa do processo de Wiener.

Volatilidade Estimada por PGDA para VALE5



Volatilidade Estimada para os Log-Retornos da VALE5

Solução analítica da EDE:

$$\widehat{X}(t) = X(0) \exp \left\{ \beta_1 + \beta_2 t^2 + \sqrt{\alpha_1 + \alpha_2 \left(\frac{1 - \alpha_3^t}{t} \right)} W(t) \right\} \quad (15)$$

onde,

$$X_0 = 35,47 \text{ (CI)}; \beta_1 = -0,06039; \beta_2 = -0,00000034; \alpha_1 = 0,000152;$$

$$\alpha_2 = 0,00025; \text{ e } \alpha_3 = 0,86878.$$

Solução discretizada da EDE:

$$\widehat{X}_{t_i} = 35,47 \exp \left\{ 0,06039 - 0,00000034 t_i^2 + \sum_{k=1}^i \widehat{b}_{t_{k-1}} (\widehat{W}_{t_k} - \widehat{W}_{t_{k-1}}) \right\}; \quad (16)$$

onde,

$$\widehat{b}_{t_k} = \sqrt{0,00002 + \left[0,07371 (\widehat{W}_{t_{k-1}} - \widehat{W}_{t_{k-2}})^2 + 0,86878 \right] \widehat{b}_{t_{k-1}}^2}$$

Para $i=1, 2, \dots, n$.



Gráfico da Solução de PGDA para a VALE5

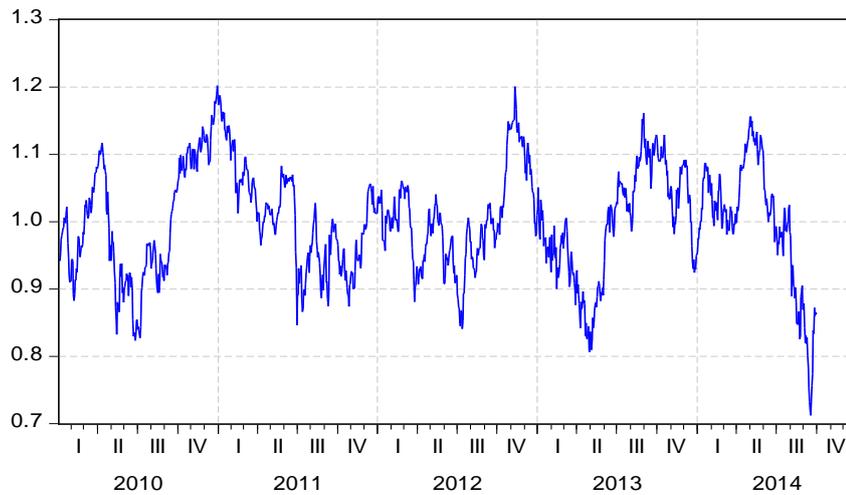
Equações e Gráficos da Tendência e da Volatilidade Total

$$\widehat{T}_{t_i} = 35,47 \exp\{0,06039 - 0,000000344 t^2\}$$

$$\hat{V}_{t_i} = \exp \left\{ \sum_{k=1}^i \hat{b}_{t_{k-1}} (\hat{W}_{t_k} - \hat{W}_{t_{k-1}}) \right\}; i = 1, 2, \dots, n.$$



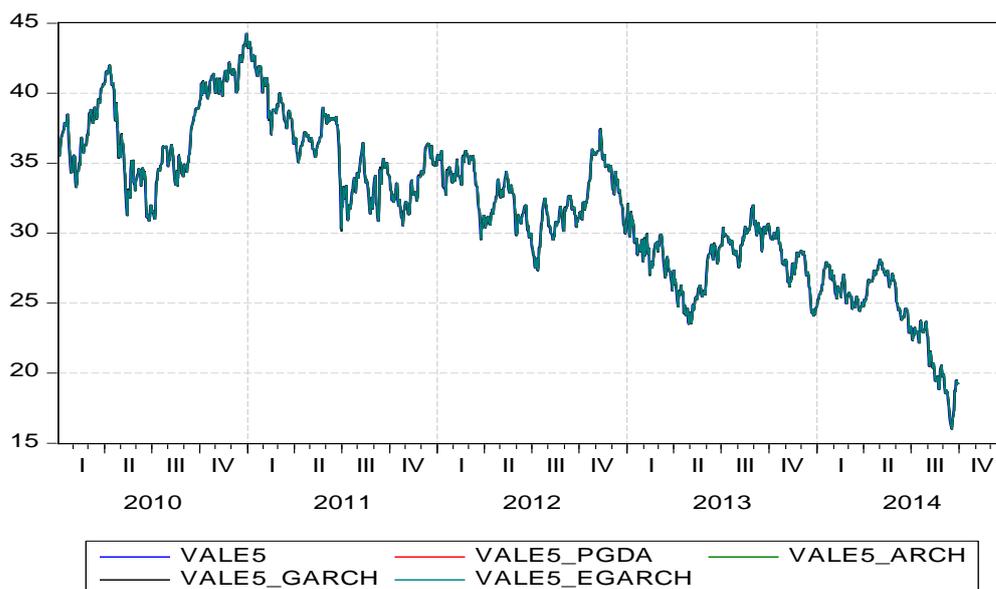
Volatilidade Total Estimada por PGDA para VALE5



Estadísticas de Erros Padrão e Testes da Estimação de PGDA para VALE5

Componente do Modelo	Estimativa do Coeficiente	Erro Padrão por Bootstrap	Estatística-z por Bootstrap	P-Valor
Tendência				
β_1	0,06039	0,00319	18,94565	0,00000
β_2	-3,44E-07	5,76E-09	-59,68582	0,00000
Volatilidade Total				
α_0	0,000185	0,0000482	3,83817	0,00012
α_1	0,000020	0,0000050	3,97063	0,00010
α_2	0,073711	0,0116910	6,30493	0,00000
α_3	0,868786	0,0232500	37,36671	0,00000

Comparação dos Resultados da Estimação de PGDA com os dos Modelos ARCH, GARCH e EGARCH

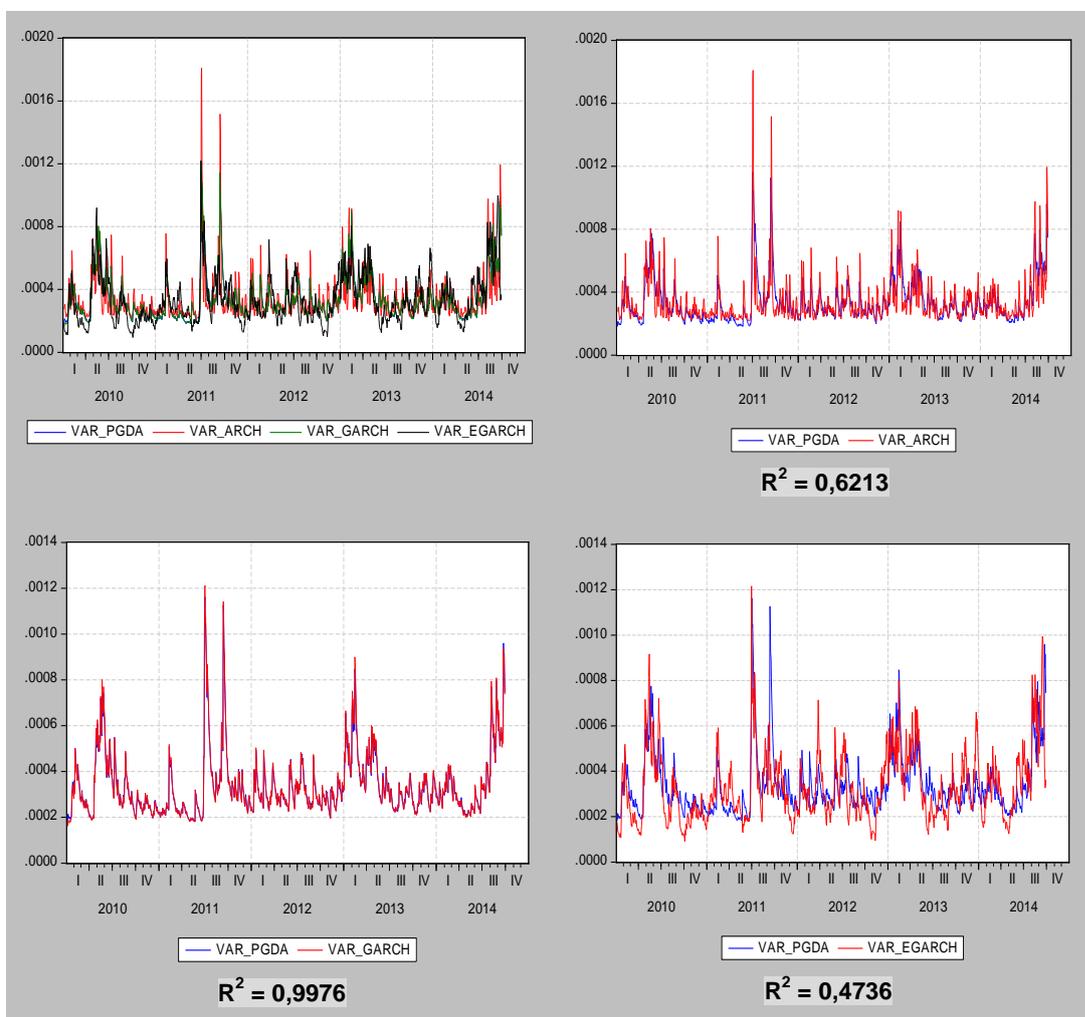


Os modelos apresentam resultados muito semelhantes e com boa qualidade de ajuste.

Estadísticas da Qualidade do Ajuste dos Modelos Estimados

Estatística	Modelos Estimados – VALE5			
	PGDA	ARCH	GARCH	EGARCH
MAPE (%)	1,34949	1,34817	1,35210	1,34817
MAE	0,41980	0,41942	0,42042	0,41942
REQM	0,56679	0,56759	0,56750	0,56759
R ² com VALE5	0,989100	0,989088	0,989096	0,989088

Estimativas dos Modelos para a Volatilidade do Log-Retorno da VALE5



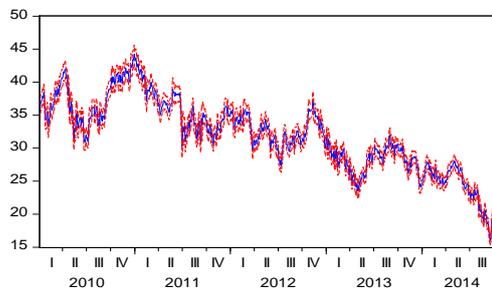
7.4.2

Resultados do Modelo ARCH para a VALE5

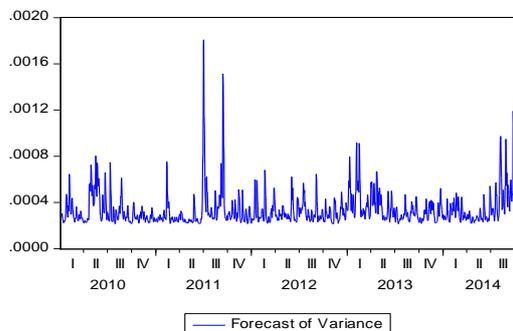
Dependent Variable: DLOG(VALE5)
 Method: ML - ARCH (Marquardt) - Normal distribution
 Included observations: 1235 after adjustments
 Convergence achieved after 11 iterations
 Presample variance: backcast (parameter = 0.7)
 GARCH = C(1) + C(2)*RESID(-1)^2 + C(3)*RESID(-2)^2 + C(4)*RESID(-3)^2
 + C(5)*RESID(-4)^2 + C(6)*RESID(-5)^2

Variable	Coefficient	Std. Error	z-Statistic	Prob.
Variance Equation				
C	0.000213	1.35E-05	15.78498	0.0000
RESID(-1)^2	0.012042	0.027164	0.443316	0.6575
RESID(-2)^2	0.131498	0.026923	4.884294	0.0000
RESID(-3)^2	0.118284	0.032494	3.640205	0.0003
RESID(-4)^2	0.065306	0.025563	2.554700	0.0106
RESID(-5)^2	0.045796	0.014803	3.093773	0.0020
R-squared	-0.000732	Mean dependent var		-0.000496
Adjusted R-squared	0.000079	S.D. dependent var		0.018335
S.E. of regression	0.018334	Akaike info criterion		-5.203407
Sum squared resid	0.415139	Schwarz criterion		-5.178538
Log likelihood	3219.104	Hannan-Quinn criter.		-5.194052
Durbin-Watson stat	1.936786			

PUC-Rio - Certificação Digital Nº 1121535/CA



Forecast: VALE5F_ARCH	
Actual: VALE5	
Forecast sample: 1/04/2010 12/30/2014	
Adjusted sample: 1/05/2010 9/30/2014	
Included observations: 1235	
Root Mean Squared Error	0.567591
Mean Absolute Error	0.419417
Mean Abs. Percent Error	1.348172
Theil Inequality Coefficient	0.008740
Bias Proportion	0.000537
Variance Proportion	0.000397
Covariance Proportion	0.999066



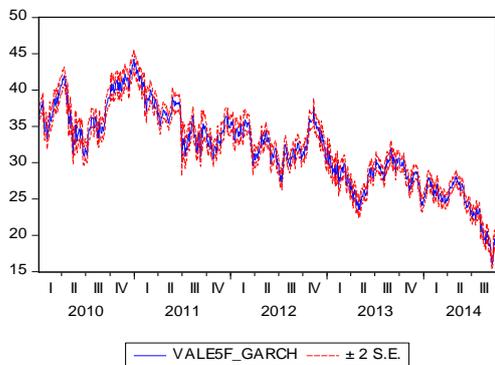
7.4.3

Resultados do Modelo GARCH para a VALE5

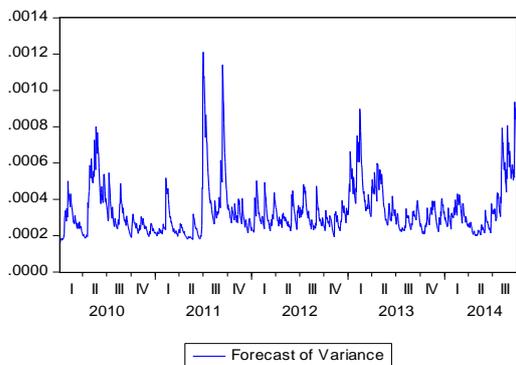
Dependent Variable: DLOG(VALE5)
 Method: ML - ARCH (Marquardt) - Normal distribution
 Included observations: 1234 after adjustments
 Convergence achieved after 10 iterations
 Presample variance: backcast (parameter = 0.7)
 GARCH = C(2) + C(3)*RESID(-1)^2 + C(4)*GARCH(-1)

Variable	Coefficient	Std. Error	z-Statistic	Prob.
AR(1)	0.060184	0.031818	1.891468	0.0586
Variance Equation				
C	2.04E-05	5.10E-06	3.993666	0.0001
RESID(-1)^2	0.077175	0.012323	6.262729	0.0000
GARCH(-1)	0.863962	0.023928	36.10617	0.0000
R-squared	-0.000613	Mean dependent var	-0.000507	
Adjusted R-squared	-0.000613	S.D. dependent var	0.018338	
S.E. of regression	0.018344	Akaike info criterion	-5.214553	
Sum squared resid	0.414888	Schwarz criterion	-5.197963	
Log likelihood	3221.379	Hannan-Quinn criter.	-5.208313	
Durbin-Watson stat	2.056923			
Inverted AR Roots	.06			

PUC-Rio - Certificação Digital Nº 1121535/CA



Forecast: VALE5F_GARCH
 Actual: VALE5
 Forecast sample: 1/04/2010 12/30/2014
 Adjusted sample: 1/06/2010 9/30/2014
 Included observations: 1234
 Root Mean Squared Error 0.567495
 Mean Absolute Error 0.420427
 Mean Abs. Percent Error 1.352097
 Theil Inequality Coefficient 0.008739
 Bias Proportion 0.000533
 Variance Proportion 0.000225
 Covariance Proportion 0.999243



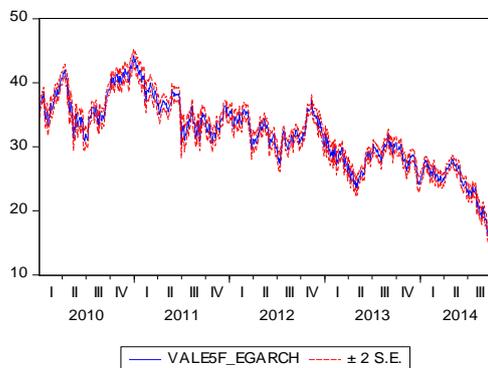
7.4.4

Resultados do Modelo EGARCH para a VALE5

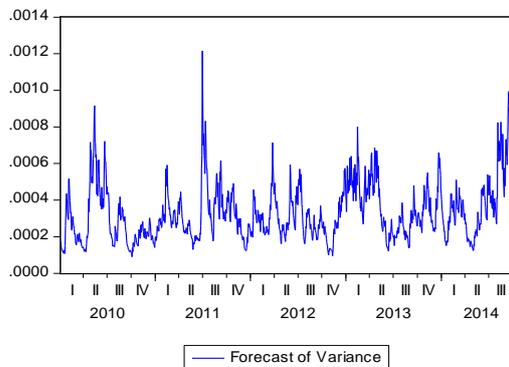
Dependent Variable: DLOG(VALE5)
 Method: ML - ARCH (Marquardt) - Normal distribution
 Included observations: 1235 after adjustments
 Convergence achieved after 12 iterations
 Presample variance: backcast (parameter = 0.7)
 $LOG(GARCH) = C(1) + C(2)*ABS(RESID(-1)/@SQRT(GARCH(-1))) + C(3)*RESID(-1)/@SQRT(GARCH(-1)) + C(4)*LOG(GARCH(-1))$

Variable	Coefficient	Std. Error	z-Statistic	Prob.
Variance Equation				
C(1)	-0.462624	0.072882	-6.347557	0.0000
C(2)	0.056003	0.019105	2.931357	0.0034
C(3)	-0.134002	0.012792	-10.47568	0.0000
C(4)	0.948685	0.008174	116.0579	0.0000
R-squared	-0.000732	Mean dependent var		-0.000496
Adjusted R-squared	0.000079	S.D. dependent var		0.018335
S.E. of regression	0.018334	Akaike info criterion		-5.269427
Sum squared resid	0.415139	Schwarz criterion		-5.252848
Log likelihood	3257.871	Hannan-Quinn criter.		-5.263191
Durbin-Watson stat	1.936786			

PUC-Rio - Certificação Digital Nº 1121535/CA



Forecast: VALE5F_EGARCH	
Actual: VALE5	
Forecast sample: 1/04/2010 12/30/2014	
Adjusted sample: 1/05/2010 9/30/2014	
Included observations: 1235	
Root Mean Squared Error	0.567591
Mean Absolute Error	0.419417
Mean Abs. Percent Error	1.348172
Theil Inequality Coefficient	0.008740
Bias Proportion	0.000537
Variance Proportion	0.000397
Covariance Proportion	0.999066



7.5

Fórmulas de Cálculo do Prêmio de Calls e Puts

Nesta seção do apêndice apresenta-se o desenvolvimento matemático utilizado para determinar as fórmulas de cálculo dos prêmios das opções de compra e de venda, do tipo europeia, para as ações-objeto PETR4 e VALE5, utilizadas pelo modelo de precificação de opções da seção (4.2.5.2).

Utilizando a solução contínua obtida por PGDA para os preços da ação PETR4, equação (13) da seção 4.2.4., determina-se, a seguir, as fórmulas da call e da put para esta ação.

Impondo a restrição da hipótese H_2 do modelo, onde se assume que o valor esperado dos log-retornos é igual a taxa livre de risco (r), substituindo-se esta restrição na equação e modificando a notação de (X para P), tem-se a seguinte expressão para a equação dos preços:

$$P(T) = P_0 \exp \left\{ rT + \sqrt{\alpha_1 + \alpha_2 \left(\frac{1 - \alpha_3^T}{T} \right)} W(T) \right\}$$

onde,

$$P_0 = \text{condição inicial}; \quad \alpha_1 = 0,00015; \quad \alpha_2 = 0,00037; \quad \text{e } \alpha_3 = 0,904045.$$

Para simplificar a notação, faremos $\sigma(T) = \sqrt{\alpha_1 + \alpha_2 \left(\frac{1 - \alpha_3^T}{T} \right)}$, então teremos:

$$P(T) = P_0 \exp\{rT + \sigma(T) W(T)\}.$$

Defina, $\text{Max}\{[P(T) - PE], 0\} = [P(T) - PE]^+$. Segue que,

$$C_0(P(T), T, PE) = e^{-rT} E(\text{Max}\{[P(T) - PE], 0\}) = E\{e^{-rT}[P(T) - PE]^+\}$$

Das propriedades do processo de Wiener, tem-se que $W(T) \sim N(0, T)$. Isso implica que,

$$S = rT + \sigma(T) W(T) \sim N(rT, \sigma^2(T) T).$$

Daí,

$$E\{e^{-rT} [P(T) - PE]^+\} = e^{-rT} \int_{-\infty}^{+\infty} [P_0 e^s - PE]^+ f(s) ds.$$

Onde,

$$f(s) = \frac{1}{\sqrt{2\pi \sigma^2(T) T}} e^{-\frac{(s-rT)^2}{2 \sigma^2(T) T}} ds$$

Logo, se $[P_0 e^s - PE]^+ = 0$, a integral acima é nula e o valor da call é zero.

Caso contrário, $[P_0 e^s - PE]^+ > 0$, que ocorre se e somente se $P_0 e^s > PE$. Ou seja, se $s > \ln\left(\frac{PE}{P_0}\right)$. Então,

$$\begin{aligned} \int_{-\infty}^{+\infty} [P_0 e^s - PE]^+ f(s) ds &= \int_{\ln\left(\frac{PE}{P_0}\right)}^{+\infty} [P_0 e^s - PE] \times f(s) ds = \\ &= \int_{\ln\left(\frac{PE}{P_0}\right)}^{+\infty} P_0 e^s f(s) ds - \int_{\ln\left(\frac{PE}{P_0}\right)}^{+\infty} PE f(s) ds = \left(\text{fazendo } z = \frac{s-rT}{\sigma(T)\sqrt{T}} \right) = \\ &= \int_{\ln\left(\frac{PE}{P_0}\right)}^{+\infty} P_0 e^s f(s) ds - \int_{\frac{\ln\left(\frac{PE}{P_0}\right)-rT}{\sigma(T)\sqrt{T}}}^{+\infty} PE \frac{e^{-\frac{z^2}{2}}}{\sqrt{2\pi}} dz = I_1 - I_2. \end{aligned}$$

Onde,

$$\begin{aligned} I_2 &= \int_{\frac{\ln\left(\frac{PE}{P_0}\right)-rT}{\sigma(T)\sqrt{T}}}^{+\infty} PE \frac{e^{-\frac{z^2}{2}}}{\sqrt{2\pi}} dz = PE \left[1 - \Phi\left(\frac{\ln\left(\frac{PE}{P_0}\right) - rT}{\sigma(T)\sqrt{T}}\right) \right] = \\ &= PE \Phi\left(-\frac{\ln\left(\frac{PE}{P_0}\right) - rT}{\sigma(T)\sqrt{T}}\right) = PE \Phi\left(\frac{\ln\left(\frac{P_0}{PE}\right) + rT}{\sigma(T)\sqrt{T}}\right) \end{aligned}$$

Φ : é a função de distribuição acumulada da $N(0, 1)$.

$$I_1 = \int_{\ln\left(\frac{PE}{P_0}\right)}^{+\infty} P_0 e^s f(s) ds = P_0 \int_{\ln\left(\frac{PE}{P_0}\right)}^{+\infty} e^s \frac{e^{-\frac{(s-rT)^2}{2 \sigma^2(T) T}}}{\sqrt{2\pi \sigma^2(T) T}} ds =$$

$$\begin{aligned}
&= P_0 \int_{\ln\left(\frac{PE}{P_0}\right)}^{+\infty} e^s \frac{e^{-\frac{(s-rT)^2}{2\sigma^2(T)T}}}{\sqrt{2\pi\sigma^2(T)T}} ds \\
&= P_0 e^{rT+\frac{\sigma^2(T)T}{2}} \int_{\ln\left(\frac{PE}{P_0}\right)}^{+\infty} \frac{e^{-\frac{(s-(rT+\sigma^2(T)T))^2}{2\sigma^2(T)T}}}{\sqrt{2\pi\sigma^2(T)T}} ds. \\
&\left(\text{fazendo } u = \frac{s-(rT+\sigma^2(T)T)}{\sigma(T)\sqrt{T}}\right) \Rightarrow \int_{\ln\left(\frac{PE}{P_0}\right)}^{+\infty} \frac{e^{-\frac{(s-(rT+\sigma^2(T)T))^2}{2\sigma^2(T)T}}}{\sqrt{2\pi\sigma^2(T)T}} ds = \\
&\int_{\frac{\ln\left(\frac{PE}{P_0}\right)-(rT+\sigma^2(T)T)}{\sigma(T)\sqrt{T}}}^{+\infty} \frac{e^{-\frac{u^2}{2}}}{\sqrt{2\pi}} du = 1 - \Phi\left(\frac{\ln\left(\frac{PE}{P_0}\right)-(rT+\sigma^2(T)T)}{\sigma(T)\sqrt{T}}\right) = \\
&= \Phi\left(\frac{\ln\left(\frac{P_0}{PE}\right) + (rT+\sigma^2(T)T)}{\sigma(T)\sqrt{T}}\right).
\end{aligned}$$

Logo,

$$I_1 = P_0 e^{rT+\frac{\sigma^2(T)T}{2}} \Phi\left(\frac{\ln\left(\frac{P_0}{PE}\right) + (rT+\sigma^2(T)T)}{\sigma(T)\sqrt{T}}\right).$$

Então,

$$I_1 - I_2 = P_0 e^{rT+\frac{\sigma^2(T)T}{2}} \Phi(z_1) - PE \Phi(z_2)$$

$$\text{onde, } z_1 = \frac{\ln\left(\frac{P_0}{PE}\right) + (rT+\sigma^2(T)T)}{\sigma(T)\sqrt{T}} \text{ e } z_2 = \frac{\ln\left(\frac{P_0}{PE}\right) + rT}{\sigma(T)\sqrt{T}}$$

Portanto, a fórmula da call para opções europeias da PETR4 é dada por:

$$C_0(P(T), T, PE) = e^{-rT} \left[P_0 e^{rT+\frac{\sigma^2(T)T}{2}} \Phi(z_1) - PE \Phi(z_2) \right] =$$

$$C_0(P(T), T, PE) = P_0 e^{\frac{\sigma^2(T)T}{2}} \Phi(z_1) - PE e^{-rT} \Phi(z_2).$$

Onde,

$$\sigma^2(T) = 0,00015 + 0,00037 \left(\frac{1 - 0,904045^T}{T} \right)$$

Aplicando a paridade, determina-se também a fórmula da put:

$$V_0(P(T), T, PE) = C_0(P(T), T, PE) - [P_0 - PE e^{-rT}] =$$

$$V_0(P(T), T, PE) = PE e^{-rT} \Phi(-z_2) - P_0 \left[1 - e^{\frac{\sigma^2(T)T}{2}} \Phi(z_1) \right].$$

O mesmo tipo de desenvolvimento foi aplicado à equação discretizada da PETR4, equação (14), e as equações contínua e discretizada da VALE5, equações (15) e (16), apresentadas na seção 7.4 deste apêndice. Os resultados obtidos possuem a mesma forma, com diferenças somente na expressão da variância.

- i) Fórmulas da call e put para PETR4, utilizando a equação discretizada estimada por PGDA:

$$C_0(P(T), T, PE) = P_0 e^{\frac{\sigma^2(T)T}{2}} \Phi(z_1) - PE e^{-rT} \Phi(z_2).$$

$$\text{onde, } \sigma^2(T) = \sum_{k=0}^{T-1} \hat{b}_k^2$$

$$\text{e } \hat{b}_k^2 = 0,0000144 + \left[0,07331 (\hat{W}_{k-1} - \hat{W}_{k-2})^2 + 0,90405 \right] \hat{b}_{k-1}^2.$$

$$V_0(P(T), T, PE) = PE e^{-rT} \Phi(-z_2) - P_0 \left[1 - e^{\frac{\sigma^2(T)T}{2}} \Phi(z_1) \right].$$

- ii) Fórmulas da call e put para VALE5, utilizando a equação contínua, eq. (15), estimada por PGDA:

$$C_0(P(T), T, PE) = P_0 e^{\frac{\sigma^2(T)T}{2}} \Phi(z_1) - PE e^{-rT} \Phi(z_2).$$

$$\text{onde, } \sigma^2(T) = 0,000152 + 0,00025 \left(\frac{1 - 0,86878^T}{T} \right)$$

$$V_0(P(T), T, PE) = PE e^{-rT} \Phi(-z_2) - P_0 \left[1 - e^{\frac{\sigma^2(T)T}{2}} \Phi(z_1) \right].$$

- iii) Fórmulas da call e put para VALE5, utilizando a equação discretizada estimada por PGDA:

$$C_0(P(T), T, PE) = P_0 e^{\frac{\sigma^2(T)T}{2}} \Phi(z_1) - PE e^{-rT} \Phi(z_2).$$

$$\text{onde, } \sigma^2(T) = \sum_{k=0}^{T-1} \hat{b}_k^2$$

$$\text{e } \hat{b}_k^2 = 0,00002 + \left[0,0737 (\hat{W}_{k-1} - \hat{W}_{k-2})^2 + 0,86878 \right] \hat{b}_{k-1}^2.$$

$$V_0(P(T), T, PE) = PE e^{-rT} \Phi(-z_2) - P_0 \left[1 - e^{\frac{\sigma^2(T)T}{2}} \Phi(z_1) \right].$$

7.6

Identificação das Opções Negociadas na BM&FBovespa

O código das opções negociadas no Mercado Bovespa é composto por 5 (cinco) letras e 2 (dois) números. Esta estrutura composta por 7 (sete) caracteres indica qual o ativo-objeto, o mês de vencimento e o preço de exercício da opção negociada.

As quatro primeiras letras do código das opções Bovespa referem-se ao ativo-objeto do contrato de opção. São as mesmas letras utilizadas na estrutura do código de negociação destes ativos no mercado à vista. O quinto caractere identifica se este é uma opção de compra ou de venda e qual é o mês de vencimento deste contrato. Veja os quadros abaixo para saber quais letras identificam as opções de compra (call) e as opções de venda (put) de acordo com seu mês de vencimento. Os dois números do código das opções Bovespa determinam o preço pelo qual a ação-objeto será negociada, em caso de exercício desta opção.

Regra da formação do código

Descrição	Código
Exemplo:	PETRI34
Regra	AAAAMNN
Ativo-objeto	AAAA
Mês de vencimento	M
Preço de Exercício	NN

Exemplos: i) **PETRI34**: opção de compra de ação da Petrobras, com vencimento em setembro (I) e preço de exercício de R\$34,00 (NN); ii) **VALEO48**: opção de venda de ação da Vale com vencimento em Março (O) e preço de exercício de R\$48,00 (NN); iii) **BBDCF26**: opção de compra de ação do Bradesco com vencimento em Junho (F) e preço de exercício de R\$ 26,00 (NN); iv) **BVMFW14**: opção de venda de ação da BM&FBovespa com vencimento em Novembro (W) e preço de exercício de R\$ 14,00 (NN).

Quadro de Identificação do Mês de Vencimento da Opção

Mês de vencimento: mês de expiração da validade do contrato de opção.		
Mês de vencimento	Série da Opção de Compra	Série da Opção de Venda
	(CALL)	(PUT)
Janeiro	A	M
Fevereiro	B	N
Março	C	O
Abril	D	P
Maio	E	Q
Junho	F	R
Julho	G	S
Agosto	H	T
Setembro	I	U
Outubro	J	V
Novembro	K	W
Dezembro	L	X

Informações:

- Ativo-objeto: valor mobiliário de referência da opção.
- O código do mercado de opções Bovespa utiliza as mesmas letras utilizadas pelo mercado à vista para definição do ativo-objeto.
- O código do mercado de opções Bovespa não indica se o ativo-objeto é uma ação preferencial (PN) ou uma ação ordinária (ON).
- No mercado de opções Bovespa, o dia de vencimento das opções ocorre toda terceira segunda-feira de cada mês.
- O código do mercado de opções Bovespa não indica o ano de vencimento da opção.
- Preço de exercício (strike): preço pelo qual a ação-objeto será negociada no exercício da opção.
- O código Bovespa não indica eventuais ajustes do preço de exercício em caso de juros, dividendos, subscrições, grupamentos, bonificações, fracionamentos, reorganizações e demais proventos em dinheiro sobre a

ação-objeto ou o seu emissor. Assim, o número descrito no código Bovespa da opção pode, eventualmente, representar o valor aproximado do real valor do preço de exercício (strike) da opção.

Exemplo de Exercício		
Série	Vencimento	Dias úteis
K	17/11/2014	26
Próximo Exercício		
Série	Vencimento	Dias úteis
L	15/12/2014	54