Universidade Federal do Pampa
Dirceu A. Maraschin Junior

Definição Intervalar da Função Densidade de Probabilidade com Distribuição Beta: Aritmética de Moore e Aritmética Multidimensional RDM

Alegrete
2016

Dirceu A. Maraschin Junior

Definição Intervalar da Função Densidade de Probabilidade com Distribuição Beta: Aritmética de Moore e Aritmética Multidimensional RDM

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Ciência da Computação da Universidade Federal do Pampa como requisito parcial para a obtenção do título de Bacharel em Ciência da Computação.

Orientador: Profa. Ma. Alice Fonseca Finger

Alegrete
2016

Dirceu A. Maraschin Junior

Definição Intervalar da Função Densidade de Probabilidade com Distribuição Beta: Aritmética de Moore e Aritmética Multidimensional RDM

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Ciência da Computação da Universidade Federal do Pampa como requisito parcial para a obtenção do título de Bacharel em Ciência da Computação.

Trabalho de Conclusão de Curso defendido e aprovado em 2 de dezembro de 2016.

Banca examinadora:

Alice Fonseca Finger

Profa. Ma. Alice Fonseca Finger
Orientadora

Aline Brum Loreto

Profa. Dra. Aline Brum Loreto
UFSM

Prof. Dr. Marcelo Resende Thielo
UNIPAMPA

Agradecimentos

Primeiramente a Deus, agradeço por toda a força que encontrei nos momentos difíceis enfrentados ao longo desse período para me manter firme perante os meus objetivos. Agradeço pela minha família, pai, mãe e irmã por me fortalecerem a cada dia e fornecerem uma base sólida de apoio e condições para que eu pudesse manter o foco sem precisar sair do conforto da própria casa. À minha namorada Luísa, agradeço por todas as palavras de apoio, à amizade e ao amor que construímos ao longo de todo esse tempo em que estamos juntos, nos mantendo fortes para superar os momentos difíceis das escolhas que fizemos pensando no futuro. Obrigado por seres a pessoa maravilhosa que sempre foi, espero que eu possa retribuir sempre. Te amo! Não basta apenas agradecer pela orientação excepcional que tive em todos os trabalhos dos quais pude participar até o momento, agradeço também pela amizade construída. Prof. Alice, muito obrigado por tudo! Obrigado por todas as reuniões, revisões, conversas informais, eventos participados onde levamos o nosso trabalho para ser posto à prova da qualidade do que fizemos e pela confiança que dedicastes a mim. Gostaria e te convido para que sejamos parceiros nessa caminhada profissional durante muito tempo ainda. Às amizades construídas durante a graduação digo obrigado sem citar nomes. Estas foram de grande importância, seja compartilhando conhecimento, seja nos momentos de descontração muito importantes para aliviar a carga enfrentada dia após dia dentro da universidade. Estas levarei por toda a vida. A todos estes, agradeço pela paciência, por acreditarem e ajudarem a desenvolver ideias, por contribuírem tanto no que consegui até agora. Obrigado!

"Não vos amoldeis às estruturas deste mundo,
mas transformai-vos pela renovação da mente,
a fim de distinguir qual é a vontade de Deus:
o que é bom, o que lhe é agradável, o que é perfeito.
(Bíblia Sagrada, Romanos 12:2)

Resumo

Quando trabalhamos com cálculos numéricos em ambientes computacionais, operamos sobre números de ponto flutuante. Dessa forma, o resultado é apenas uma aproximação de um valor real e erros gerados por arredondamentos ou truncamentos podem levar a resultados incorretos, não podendo ser afirmada a exatidão das respostas estimadas sem o auxílio de uma análise de erro. Ao utilizar-se intervalos para representação de valores reais, torna-se possível controlar a propagação desses erros, pois resultados intervalares carregam consigo a segurança de sua qualidade. Para obter o valor numérico das funções densidade de probabilidade das variáveis aleatórias contínuas com distribuições se faz necessário o uso da integração numérica. Sendo o resultado obtido por aproximação, este é afetado por erros. Nesse contexto, o presente trabalho possui o objetivo de definir a função densidade de probabilidade da variável aleatória contínua com distribuição Beta de forma intervalar utilizando o método de extensão intervalar, posteriormente, realizar-se a implementação dessas utilizando linguagem de programação com suporte ao tipo intervalo. Duas abordagens acerca da aritmética intervalar serão consideradas: a aritmética intervalar de Moore e a aritmética intervalar multidimensional RDM. A fim de verificar a qualidade dos resultados intervalares obtidos, será feita uma análise de erro através do cálculo das métricas de erro relativo, erro absoluto e diâmetro dos intervalos. Depois, de forma a complementar o trabalho, ainda será feita análise de complexidade dos algoritmos desenvolvidos. Seguindo tal objetivo, deseja-se justificar que, ao utilizar aritmética intervalar para o cálculo da função densidade de probabilidade com distribuição Beta, é possível ter um controle automático de erros com limites confiáveis e, além disso, manter o esforço computacional relativo aos cálculos com entradas reais.

Palavras-chave: Aritmética intervalar. RDM Intervalar. Análise de erros. Esforço computacional. Distribuições de probabilidade.

Abstract

When working with numerical calculations in computational environments, we operate on floating point numbers. Thus, the result is only an approximation of a real value and errors created by rounding or truncation can be lead to incorrect results, which cannot be considered as accurates without an error analysis. When using ranges to represent real values, it becomes possible to control the spread of these errors, since interval results have quality assurance. To obtain the numerical value of density functions of continuous random variables with distributions is required the use of numerical integration. If the result is obtained by approximation, it is affected by errors. In this context, this work has the objective of define the probability density function with Beta distribution using the interval extension method, by following some criteria of choose, and subsequently, to implements this functions using programming language with support to interval type. Two approaches to interval arithmetic will be considered: The interval arithmetic of Moore and the multidimensional RDM interval-arithmetic. In order to verify the quality of the interval results obtained, an error analysis will be done by calculating the relative error, absolute error and interval diameter metrics. Then, in order to complement the work, a complexity analysis of the developed algorithms will be done. The goal is to verify that, when using interval arithmetic to calculate probability density function of continuous random variables with distributions, it is possible to has an automatic error control with reliable limits and also, maintain the computation effort for the calculations with real entries.

Key-words: Interval arithmetic. Interval RDM. Error analysis. Computational effort. Probability distributions.

Lista de ilustrações

Lista de tabelas

Sumário

1 Introdução

Uma solução numérica exata requer muitas vezes uma sequência infinita de operações matemáticas exatas para ser calculada (MOORE; YANG, 1959). No entanto, o conjunto de números representáveis em qualquer máquina é finito, portanto discreto. Isso se reflete diretamente na possibilidade de serem constatadas fontes de erros, tais como: simplificação no modelo matemático; erro de truncamento; erro de arredondamento; propagação dos erros nos dados de entrada, entre outros (FRANCO, 2006).

Os sistemas de computador atuais são padronizados para operar com formatos e métodos da aritmética de ponto flutuante, conforme o padrão IEEE (ZURAS et al., 2008), de modo que resultados numéricos e exceções como valores não normalizados ou especiais como o infinito, por exemplo, sejam determinados a partir dos dados de entrada do usuário. Desse modo, números reais são representados por aproximação através de um subconjunto chamado números de máquina. Devido a essa representação são gerados dois tipos de erros: o primeiro ocorre quando um valor real de entrada é aproximado para um número de máquina; o segundo é causado pelos resultados intermediários gerados na execução de cada operação e que vão se acumulando (LORETO, 2006).

A análise de erros numéricos é um processo importante para se obter resultados confiáveis e precisos em processos computacionais. A relevância de tal processo se reflete quando falhas envolvendo sistemas de computador ocorrem. Mesmo que, para a maioria das pessoas, um software seja algo que opera de maneira invisível e pareça altamente seguro, erros em sistemas podem exercer um grave impacto na sociedade (TAMAI, 2009).

A análise intervalar surgiu com o objetivo inicial de controlar a propagação de erros numéricos em procedimentos computacionais. Através da sua utilização tem-se um controle automático de erros com limites confiáveis, além de provas de existência e não existência de solução de diversos problemas (LORETO, 2006). Justificando-se, assim, o emprego de técnicas intervalares para contornar erros gerados pela aritmética de ponto flutuante, fazendo-se possível obter resultados mais exatos com menor erro contido.

A matemática intervalar fornece uma alternativa aos erros ocasionados por aproximações utilizando intervalos, os quais foram definidos com o propósito de automatizar a análise do erro computacional. Na aritmética intervalar, um valor real x é aproximado por um intervalo fechado $\mathbf{x}$ que possui limites inferior e superior, da forma $\mathbf{x}$=[$\underline{x}$, $\bar{x}$], de modo que este intervalo contenha o valor real x (MOORE; KEARFOTT; CLOUD, 2009). Dessa forma, todos os cálculos reais precisam ser substituídos por cálculos que utilizem aritmética intervalar, além de que o comprimento deste intervalo pode ainda ser utilizado como medida para avaliar a qualidade do intervalo solução (RATSCHEK; ROKNE, 1988).

Intervalos podem ser aplicados em diversas áreas, tais como: programação matemática, manipulação de equações, projeto de circuitos elétricos, estatística, processamento de imagens, cálculo diferencial, física, engenharia química e muitos outros (TAKAHASHI; BEDREGAL; LYRA, 2005)(MOORE; KEARFOTT; CLOUD, 2009).

No estudo das variáveis aleatórias contínuas aplicadas no conjunto dos reais, $\mathbb{R}$, um problema enfrentado é o cálculo de probabilidades, visto que se faz necessário resolver uma integral definida, referente à função densidade de probabilidade (f.d.p.). Sendo esta resolvida normalmente de forma analítica, seu valor resultante é dado por aproximação e pode ser afetado por erros de arredondamento ou truncamento (FINGER, 2014).

Dentre as diversas soluções de implementação possíveis para um dado problema, é preciso avaliar qual é a mais adequada e, depois disso, julgar quanto à otimalidade da solução desenvolvida. Para isso, existe a análise de algoritmos, a qual tem por objetivo melhorar, quando possível, seu desempenho e qualificá-lo em termos de sua eficiência. Existem diversos critérios para a avaliação de um algoritmo, dentre elas: quantidade de memória ocupada, exatidão da saída, processamento requerido e tempo de processamento (TOSCANI; VELOSO, 2009).

O objetivo deste trabalho é definir a função densidade de probabilidade da variável aleatória contínua com distribuição Beta, acrescentando às já existentes na literatura definidas desta forma. Utilizando o método de extensão intervalar para realizar essa definição, o passo seguinte é aplicar duas abordagens diferentes como modo de computar essa distribuição utilizando aritmética intervalar: a aritmética intervalar tradicional de Moore (MOORE, 1966) e a aritmética multidimensional RDM, de Piegat e Landowski (PIEGAT; LANDOWSKI, 2012). Para tanto, a implementação da função densidade será desenvolvida utilizando o pacote de extensão para programação intervalar da linguagem de programação Python, denominado IntPy (BARRETO, 2016), o qual dispõe definido o tipo intervalo e operações sobre o tipo. Após a implementação, será analisada a qualidade dos intervalos obtidos a fim de comparar resultados com a forma real da distribuição, apurando sua exatidão.

Com a preocupação se a quantidade de processamento necessário ao utilizar-se intervalos aumenta, pretende-se realizar a análise da complexidade dos algoritmos elaborados para computar a função densidade de probabilidade das variáveis aleatórias contínuas com distribuições, nas formas real e intervalar. Por meio desta análise será possível saber se o cálculo utilizando intervalos para tal função oferece, além do controle automático de erros com limites confiáveis, um esforço computacional aceitável quando comparado ao cálculo da função densidade com valores reais.

1.1 Objetivos

O objetivo principal deste trabalho é utilizar a matemática intervalar para criar a definição intervalar para a função densidade de probabilidade da variável aleatória contínua com distribuição Beta. Além disso, utilizar duas abordagens diferentes a respeito da aritmética intervalar para computar a função densidade com distribuição Beta utilizando um ambiente de desenvolvimento intervalar.

A fim de atingir o objetivo principal, os objetivos específicos a serem executados são listados e detalhados logo abaixo:

- Definir a função densidade de probabilidade da variável aleatória contínua com distribuição Beta para entradas intervalares. Na literatura podem ser encontradas as definições das funções densidade de probabilidade das variáveis aleatórias contínuas Uniforme, Exponencial, Normal, Gama e Pareto nas quais é aplicado um método da aritmética intervalar, chamado método de extensão intervalar, por meio do qual se faz possível definir, a partir de uma função real, sua correspondente função intervalar.

- Aplicar o método de extensão intervalar à função densidade de probabilidade da variável aleatória contínua com distribuição Beta de modo que todas as entradas que antes comporiam entradas reais, passem a receber entradas intervalares para os parâmetros da função densidade.

- Depois de ser definida na forma intervalar, implementar a função densidade de probabilidade com distribuição Beta, tanto real quanto intervalar. Utilizar, para isso, a linguagem de programação Python (PYTHON, 2016), operando com o pacote IntPy, desenvolvido por Barreto (2016), o qual possibilita a programação utilizando intervalos.

- Utilizar duas abordagens diferentes da aritmética intervalar para as implementações intervalares. A aritmética intervalar desenvolvida por Moore (1966), como sendo um método já conhecido na literatura, portanto já aplicado inclusive nas distribuições de probabilidade definidas de forma intervalar na literatura e, a aritmética intervalar multidimensional RDM desenvolvida por Piegat e Landowski (2012), sendo esta uma nova abordagem a respeito da aritmética intervalar, onde até mesmo as operações aritméticas básicas estão definidas apenas conceitualmente.

- Apresentar os resultados de qualidade para os intervalos obtidos através da computação da função densidade de probabilidade com distribuição Beta intervalar, utilizando-se medidas de erro absoluto e erro relativo, bem como o cálculo do diâmetro do intervalo, o que se faz importante para avaliar a qualidade e precisão dos intervalos.

- Realizar a análise de complexidade dos algoritmos implementados a fim de analisar o esforço computacional despendido para calcular a função densidade de probabilidade da variável aleatória contínua com distribuição Beta, nas formas real e intervalar. Com a análise da complexidade de algoritmos sendo realizada para cada forma da distribuição, será possível justificar o controle da propagação de erros, mantendo ainda um esforço computacional aceitável necessário para executar os cálculos utilizando intervalos.

1.2 Organização do Trabalho

O presente trabalho organiza-se da seguinte forma:

O Capítulo 2 apresenta uma fundamentação teórica sobre matemática intervalar. As duas abordagens para a matemática intervalar que serão utilizadas neste trabalhos estão descritas neste capítulo. Conceitos e definições das operações, bem como o modo em que essas operações são calculadas respeitando cada uma dessas abordagens são especificados em seções distintas. Exemplos são empregados de forma a esclarecer melhor esses conceitos.

No capítulo seguinte, Capítulo 3, alguns trabalhos já desenvolvidos, relacionados ao tema matemática intervalar e probabilidade e estatística, são apresentados de modo que são contribuintes em conceitos e auxiliam em enriquecer o presente trabalho, alguns resultados julgados importantes sobre tais trabalhos também são apresentados sucintamente dentro de cada respectiva seção.

O Capítulo 4 apresenta um embasamento teórico em relação à representação numérica em sistemas de computador, os erros ocasionados por tal representação e também as medidas utilizadas para mensurar esses erros. A forma de implementação de um algoritmo utilizando aritmética intervalar e o conceito da aritmética de exatidão máxima são abordados neste mesmo capítulo. Dando sequência, a definição intervalar para a função densidade de probabilidade com distribuição Beta é apresentada na seção 4.1, englobando também uma descrição matemática para a composição desta função. Na seção seguinte, desenvolve-se a análise numérica onde são apresentados os resultados obtidos a partir da computação da função densidade de probabilidade com distribuição Beta nas formas real e intervalar, onde, para a forma intervalar os resultados são dispostos para ambas as abordagens da aritmética intervalar utilizadas neste trabalho. A análise de qualidade a partir das métricas de erros são apresentadas nessa mesma seção com o objetivo de apurar a qualidade dos resultados intervalares obtidos.

O Capítulo 5 é reservado para a análise de complexidade dos algoritmos desenvolvidos para a computação da função densidade de probabilidade com distribuição Beta. Separadamente, cada algoritmo é analisado em termos de suas operações necessárias para calcular a função da distribuição Beta, apurando-se, a partir disso, suas respectivas ordens de complexidade.

As conclusões obtidas no presente trabalho são apresentadas no Capítulo 6. Reservou-se este espaço para também apresentar as perspectivas de trabalhos futuros, os quais se deseja desenvolver a partir do conhecimento adquirido no presente trabalho e dar continuidade em trabalhos relacionados..

Por fim, as referências bibliográficas são apresentadas.

2 Aritmética Intervalar

Para muitos cientistas, a aritmética intervalar não representa grande importância, sobretudo a exatidão dos resultados na resolução de problemas. Porém, na medida em que se torna possível obter resultados exatos e com menor erro, este conceito ganha grande relevância com o desenvolvimento da teoria da incerteza.

Desenvolvida por Liu (2007) em 2007, a teoria da incerteza trata de dados incertos, para os quais não se pode afirmar sua exatidão, podendo ser deduzida com base em três conceitos fundamentais: o primeiro é a medida de incerteza, a qual pode ser usada para dimensionar o grau de crença em um dado evento incerto; o segundo conceito trata do uso de uma variável incerta para representar quantidades não precisas; o último se refere à distribuição de incerteza, o que descreve variáveis incertas de forma incompleta mas simples de serem utilizadas (LIU, 2010). Dessa forma, a teoria da incerteza propõe como uma das soluções um modelo matemático para o trabalho com fenômenos incertos, a aritmética intervalar.

A teoria de matemática intervalar é descrita seguindo os conceitos desenvolvidos inicialmente por Moore (MOORE; KEARFOTT; CLOUD, 2009) (MOORE, 1976) (MOORE; YANG, 1959), sendo baseada no uso de intervalos fechados $[\underline{x}, \bar{x}]$ de números reais como elementos básicos e sua ideia do ponto de vista computacional, basicamente, é: dada uma função $f(x)$ de variável real x pertencente a um intervalo $\mathbf{x}$=$[\underline{x}, \bar{x}]$, onde $\underline{x}$, $\bar{x} \in \mathbb{R}$, a imagem da função f é dada por:

$$f(x) = \{y|y = f(x), \underline{x} \leq x \leq \bar{x}\},$$

onde nem sempre é representada exatamente, mas é possível determinar um intervalo $\mathbf{y}$, da forma $\mathbf{y}$=$[\underline{y}, \bar{y}]$, tal que $f\,(x) \subseteq \mathbf{y}$, logo: $\underline{y} \leq f(x) \leq \bar{y}$. A partir disso, pode-se então definir uma função F associada a f pela transformação do intervalo $[\underline{x}, \bar{x}]$ no intervalo $[\underline{y}, \bar{y}]$, de modo que:

$$f(x) \subseteq F(x) = \mathbf{y}.$$

Definição 1 (função intervalar): *Seja $F : x \to y$ uma função. Se $x = Dom(f) \subseteq \mathbb{R}$ e $y = Cod(f) \subseteq \mathbb{R}$, então, dizemos que F é uma função intervalar de uma variável intervalar.*

A função F, chamada de extensão intervalar de f, deve ser a função que se afasta o mínimo possível da diferença da imagem $f(x)$. A forma de apurar o erro obtido ao se calcular $f(x)$ a partir do intervalo $\mathbf{x}$ é através cálculo do seu diâmetro.

Definição 2 (diâmetro do intervalo): *O diâmetro de um intervalo* $\mathbf{x} = [\underline{x}, \bar{x}] \in \mathbb{R}$ *é denotado por* $diam(\mathbf{x})$ *ou* $w(\mathbf{x})$*, como sendo o número real não negativo* $(x \in \mathbb{R}^+)$ *tal que* $diam(\mathbf{x}) = w(\mathbf{x}) = \bar{x} - \underline{x}$*.*

O cálculo de expressões utilizando aritmética intervalar consiste basicamente na extensão das operações aritméticas junto a um conjunto de operações "*standard*", ou padrão.

Na aritmética intervalar é utilizado um arredondamento especial, o chamado arredondamento direcionado. Isso significa que os valores dos intervalos que contêm o resultado das operações são arredondados para o menor e para o maior número de máquina, obtendo-se assim um intervalo de máquina, com diâmetro mínimo, no qual a solução se situa.

Uma distinção importante na matemática intervalar é entre a imagem intervalar e a avaliação intervalar da uma função $f(x)$.

Definição 3 (imagem da função intervalar): *A imagem intervalar de uma função* f*, contínua no intervalo* $\boldsymbol{x}$*, é definida como sendo o intervalo limitado pelo mínimo e pelo máximo da imagem de* $f(x)$*, sendo* x *um elemento do intervalo* $\boldsymbol{x}$*, da seguinte forma:*

$$Im(f(x)) = \{min[f(x)], max[f(x)] \mid x \in \mathbf{x}\}.$$

Na computação intervalar, segundo Kreinovich et al. (2013), pode-se obter o intervalo solução, $\mathbf{y} = [\bar{y}, \underline{y}] = f(x_1, \ldots, x_n)$, por meio de métodos de aproximação, técnicas de otimização, extensão intervalar e, ainda, por métodos considerados sofisticados como, por exemplo, encontrar o melhor algoritmo, o qual estima intervalos mais estreitos, para calcular a mesma função.

No presente trabalho será utilizado o método de extensão intervalar para definir de forma intervalar a função densidade de probabilidade com distribuição Beta. O método da extensão intervalar, segundo Ferson et al. (2002), com base nos conceitos de Moore é, historicamente, o primeiro método para computar o intervalo solução de uma função. Extensões intervalares foram propostas por Moore como forma de generalizar funções reais em termos de intervalos. A principal abordagem de tal método tem como princípio de que em um computador, todo algoritmo processa os dados de entrada executando operações elementares, lógicas e aritméticas.

Para cada operação elementar sobre uma função $f(x, y)$, se são conhecidos os intervalos de $\mathbf{x}$ e $\mathbf{y}$ para x e y, pode-se então calcular a imagem exata $f(x, y)$ por meio da aritmética intervalar definida por Moore, Kearfott e Cloud (2009). Com isso, é possível repetir a computação, utilizando a extensão intervalar, formando o programa f passo-a-passo, sendo necessário substituir cada operação elementar de números reais pela correspondente operação em aritmética intervalar.

Segundo Santiago, Bedregal e Acióly (2006), a extensão intervalar proposta por Moore pode ser definida da seguinte forma:

Definição 4: *A função $F : I(\mathbb{R}) \to I(\mathbb{R})$ é uma extensão intervalar de uma função $f : \mathbb{R} \to \mathbb{R}$, se para todo $x \in \mathbb{R}$, $F([x, x]) = [f(x), f(x)]$.*

Pode-se também avaliar a corretude de uma função intervalar F em relação à uma função real f, por via de regra:

Definição 5: *Uma função intervalar F está correta em relação à f, real, se satisfaz à propriedade:*

$$x \in [\underline{x}, \bar{x}] \Rightarrow f(x) \in F([\underline{x}, \bar{x}])$$

É possível dizer ainda que F representa f ou que F é uma representação intervalar de f. Um algoritmo intervalar A(x) está correto em relação à f, ou a representa, se este pode ser interpretado em uma função intervalar parcial F que representa f (SANTIAGO; BEDREGAL; ACIÓLY, 2006).

2.1 Aritmética Intervalar de Moore

Na aritmética intervalar desenvolvida por Moore é possível calcular valores através de operações elementares conhecidas da matemática básica, tais como: adição, subtração, multiplicação e divisão. Porém, ao invés de se operar sobre valores reais, $\mathbb{R}$, tais cálculos são realizados sobre intervalos da forma $[\underline{x}, \bar{x}]$.

Assim, considerando $\mathbb{IR}$, chamado conjunto de intervalos reais, o conjunto que representa todos os intervalos fechados de números reais, pode-se expressar um intervalo $\mathbf{x}$ como:

$$\{\mathbf{x} = [x_1, x_2] | x_1, x_2 \in \mathbb{R}, x_1 \leq x_2\},$$

onde x_1 e x_2 representam, respectivamente o limite inferior $\underline{x}$ e o limite superior $\bar{x}$ do intervalo.

O ponto-chave das definições de Moore, as quais serão expostas logo a seguir, é que a computação sobre intervalos ocorre sobre conjuntos, ou seja, quando se faz a adição de dois intervalos, por exemplo, o intervalo resultante é um novo conjunto contendo as somas de todos os pares de números das duas séries iniciais. Nessa aritmética, quando o método de extensão intervalar é aplicado, apenas as “fronteiras” do intervalo são consideradas, ou seja, somente os valores que representam os limites do intervalo são utilizadas nos cálculos.

As operações básicas da aritmética intervalar serão explicitadas sobre o conjunto de intervalos $\mathbf{x} = [\underline{x}, \bar{x}]$ e $\mathbf{y} = [\underline{y}, \bar{y}]$ de valores reais quaisquer.

A operação de adição entre dois conjuntos, $\mathbf{x}$ e $\mathbf{y}$, se dá do seguinte modo:

$$\mathbf{x} + \mathbf{y} = [\underline{x} + \underline{y}, \bar{x} + \bar{y}].$$

De forma semelhante à adição, a diferença entre dois conjuntos, $\mathbf{x}$ e $\mathbf{y}$, é apresentada:

$$\mathbf{x} - \mathbf{y} = [\underline{x} - \bar{y}, \bar{x} - \underline{y}].$$

O produto entre $\mathbf{x}$ e $\mathbf{y}$ utiliza as operações de *min* e *max*, de modo a considerar a possibilidade de valores negativos nos intervalos, desse modo, a operação de multiplicação é dada por:

$$\mathbf{x} \cdot \mathbf{y} = [min\{\underline{x}\underline{y}, \underline{x}\bar{y}, \bar{x}\underline{y}, \bar{x}\bar{y}\}, max\{\underline{x}\underline{y}, \underline{x}\bar{y}, \bar{x}\underline{y}, \bar{x}\bar{y}\}].$$

Se ambos os intervalos forem de valores positivos, pode-se realizar a operação de multiplicação de forma simplificada:

$$\mathbf{x} \cdot \mathbf{y} = [\underline{x}\underline{y}, \bar{x}\bar{y}]$$

O quociente entre $\mathbf{x}$ e $\mathbf{y}$ é dado de forma semelhante à multiplicação. Considerando a possibilidade de valores negativos, pode-se operar da seguinte forma:

$$\mathbf{x}/\mathbf{y} = [min\{\underline{x}/\underline{y}, \underline{x}/\bar{y}, \bar{x}/\underline{y}, \bar{x}/\bar{y}\}, max\{\underline{x}/\underline{y}, \underline{x}/\bar{y}, \bar{x}/\underline{y}, \bar{x}/\bar{y}\}]$$

Porém, se ambos os intervalos forem positivos, a forma simplificada para a operação de divisão pode ser expressa por:

$$\mathbf{x}/\mathbf{y} = [\underline{x}/\bar{y}, \bar{x}/\underline{y}]$$

Assume-se que $\mathbf{y}$ possui elementos não nulos $\{\underline{y} \neq 0, \bar{y} \neq 0\}$, para ambas as formas de operação dispostas.

Assim, as operações (adição, subtração, multiplicação e divisão) podem ser executadas e, como resultado, obtém-se então um intervalo com elementos resultantes da operação computada sobre as respectivas posições dos elementos dos conjuntos que compõem a operação sobre $\mathbf{x}$ e $\mathbf{y}$.

Exemplo: *Sejam os intervalos reais* $\mathbf{x}$ *e* $\mathbf{y}$*, sendo* $\mathbf{A} = [2, 7]$ *e* $\mathbf{B} = [3, 4]$*, tem-se então as operações definidas na aritmética de Moore:*

$$A + B = [2 + 3, 7 + 4] = [5, 11]$$

$$A - B = [2 - 4, 7 - 3] = [-2, 4]$$

$$A \cdot B = [2 \cdot 3, 7 \cdot 4] = [6, 28]$$
$$A/B = \left[\frac{2}{4}, \frac{7}{3}\right] = [0.5, 2.3]$$

2.2 Aritmética Intervalar Multidimensional

A aritmética intervalar multidimensional foi desenvolvida principalmente por Piegat e Landowski (PIEGAT; LANDOWSKI, 2012)(PIEGAT; LANDOWSKI, 2013a) com base na teoria da incerteza desenvolvida por Liu (2010), como sendo uma alternativa para algumas limitações encontradas na aritmética intervalar de Moore, tais como: problema do excesso de largura dos intervalos, dependência de dados e solicitação para introduzir entropia negativa em sistemas (LANDOWSKI, 2015).

Como na prática qualquer valor, parâmetro ou variável, está sujeito a ser calculado com um fator de erro, estes não podem ser precisamente conhecidos. Sendo que todo cálculo numérico pode ser expresso na forma de intervalos, assim como a aritmética intervalar desenvolvida por Moore, a aritmética intervalar multidimensional de Piegat também pode ser utilizada como base para outras ciências que buscam solucionar problemas relacionados à incerteza (PIEGAT; LANDOWSKI, 2013b).

A abreviatura RDM (do inglês, *Relative Distance Measure*) significa Medida da Distância Relativa, sendo caracterizada como “multidimensional” por permitir perceber que cada novo parâmetro de incerteza de um sistema, aumenta a sua dimensionalidade. De modo geral, RDM possui essa descrição quando dado um valor x, pertencente a um intervalo $\mathbf{x} = [\underline{x}, \bar{x}]$, é descrito utilizando uma variável RDM α_x da seguinte forma:

$$x = \underline{x} + \alpha_x(\bar{x} - \underline{x}), \text{ onde } \alpha_x \in [0, 1].$$

Por definição, o intervalo $\mathbf{x} = [\underline{x}, \bar{x}]$ é descrito em notação RDM do seguinte modo:

$$\mathbf{x} = \{x : x = \underline{x} + \alpha_x(\bar{x} - \underline{x}), \alpha_x \in [0, 1]\}.$$

A variável RDM α_x dá a possibilidade de se obter qualquer valor entre as bordas do intervalo, ou seja, entre o limite inferior $\underline{x}$ e o limite superior $\bar{x}$ do intervalo $\mathbf{x}$. Dado que $\alpha_x \in [0, 1]$, se $\alpha_x = 0$, o valor do intervalo $\mathbf{x}$ é igual a $\underline{x}$ e, caso $\alpha_x = 1$, o valor de $\mathbf{x}$ é igual a $\bar{x}$. Por exemplo, supondo que o valor x pertence ao intervalo $[2, 4]$, utilizando-se a notação RDM, o valor x pode ser expressado por $x = 2 + 2\alpha_x$. A partir disso fica claro que, quando $\alpha_x = 0$, x assume o valor 2, para $\alpha_x = 1$ o valor de x é 4.

A Figura 1 ilustra o intervalo $\mathbf{x} = [\underline{x}, \bar{x}]$ e o significado da variável RDM α_x, dado que $\underline{x} \leq \bar{x}$.

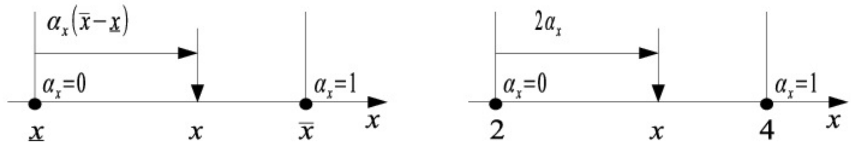

Figura 1 – Intervalo **x** e o significado da variável RDM

Fonte: (PIEGAT; LANDOWSKI, 2013b).

Para dois intervalos, $\mathbf{x} = [\underline{x}, \bar{x}]$ e $\mathbf{y} = [\underline{y}, \bar{y}]$, com limites superior e inferior quaisquer, o intervalo solução para as operações básicas RDM pode ser obtido por:

$$\mathbf{x} + \mathbf{y} = [min\{\mathbf{x} + \mathbf{y}\}, max\{\mathbf{x} + \mathbf{y}\}]$$
$$\mathbf{x} - \mathbf{y} = [min\{\mathbf{x} - \mathbf{y}\}, max\{\mathbf{x} - \mathbf{y}\}]$$
$$\mathbf{x} \cdot \mathbf{y} = [min\{\mathbf{x} \cdot \mathbf{y}\}, max\{\mathbf{x} \cdot \mathbf{y}\}]$$
$$\mathbf{x}/\mathbf{y} = [min\{\mathbf{x}/\mathbf{y}\}, max\{\mathbf{x}/\mathbf{y}\}]$$

Assim como na aritmética proposta por Moore, na aritmética RDM as operações também são efetuadas sobre intervalos, obtendo-se um resultado intervalar. A aritmética multidimensional possui as quatro operações matemáticas básicas definidas na literatura: soma, subtração, multiplicação e divisão.

Dependendo do número de variáveis nos cálculos numéricos aos quais se aplica RDM, a solução intervalar obtida é um espaço multidimensional. Em contrapartida à aritmética de Moore, a qual oferece soluções unidimensionais (LANDOWSKI, 2015).

Em vista disso, nesta subseção as operações básicas definidas na arimética multidimensional RDM serão explicitadas sobre os conjuntos $\mathbf{x} = [\underline{x}, \bar{x}]$ e $\mathbf{y} = [\underline{y}, \bar{y}]$, sendo que, de acordo com as definições apresentadas por Landowski (2015), estes intervalos são representados da seguinte forma:

$$\mathbf{x} = [\underline{x}, \bar{x}] = \{x : x = \underline{x} + \alpha_x(\bar{x} - \underline{x}), \alpha_x \in [0, 1]\};$$
$$\mathbf{y} = [\underline{y}, \bar{y}] = \{y : y = \underline{y} + \alpha_y(\bar{y} - \underline{y}), \alpha_y \in [0, 1]\}.$$

Sobre esses dois intervalos, define-se a operação de soma da seguinte forma:

$$\mathbf{x} + \mathbf{y} = \{\underline{x} + \alpha_x(\bar{x} - \underline{x}) + \underline{y} + \alpha_y(\bar{y} - \underline{y}); \alpha_x, \alpha_y \in [0, 1]\}.$$

O cálculo da subtração obedece à seguinte forma de operação:

$$\mathbf{x} - \mathbf{y} = \{\underline{x} + \alpha_x(\bar{x} - \underline{x}) - \underline{y} - \alpha_y(\bar{y} - \underline{y}); \alpha_x, \alpha_y \in [0, 1]\}.$$

O produto entre os intervalos x e y é dado por:

$$\mathbf{x} \cdot \mathbf{y} = \{[\underline{x} + \alpha_x(\bar{x} - \underline{x})] \cdot [\underline{y} + \alpha_y(\bar{y} - \underline{y})]; \alpha_x, \alpha_y \in [0,1]\}.$$

O cálculo do quociente pela aritmética RDM se dá da seguinte forma:

$$\mathbf{x}/\mathbf{y} = \{[\underline{x} + \alpha_x(\bar{x} - \underline{x})]/[\underline{y} + \alpha_y(\bar{y} - \underline{y})]; \alpha_x, \alpha_y \in [0,1], 0 \notin Y\}.$$

Exemplo: *Para demonstrar a multidimensionalidade da aritmética RDM, serão utilizados os intervalos* $A = [1,2]$ *e* $B = [3,4]$*. Considerando o uso das variáveis RDM* α_a *e* α_b *pertencentes ao intervalo [0, 1].*

A solução para os intervalos A e B obedecem à seguinte regra:

$$A = [1,2] = \{a : a = 1 + \alpha_a, \alpha_a \in [0,1]\}$$
$$B = [3,4] = \{b : b = 3 + \alpha_b, \alpha_b \in [0,1]\}$$

Para a obtenção dos intervalos solução, as operações são dispostas a seguir:

$$A + B = \left[\min_{\alpha_a,\alpha_b \in [0,1]}(4 + \alpha_a + \alpha_b), \max_{\alpha_a,\alpha_b \in [0,1]}(4 + \alpha_a + \alpha_b)\right] = [4,6]$$
$$A - B = \left[\min_{\alpha_a,\alpha_b \in [0,1]}(-2 + \alpha_a - \alpha_b), \max_{\alpha_a,\alpha_b \in [0,1]}(-2 + \alpha_a - \alpha_b)\right] = [-3,-1]$$
$$A \cdot B = \left[\min_{\alpha_a,\alpha_b \in [0,1]}(3 + 3 \cdot \alpha_a + \alpha_b + \alpha_a \cdot \alpha_b), \max_{\alpha_a,\alpha_b \in [0,1]}(3 + 3 \cdot \alpha_a + \alpha_b + \alpha_a \cdot \alpha_b)\right] = [3,8]$$
$$A/B = \left[\min_{\alpha_a,\alpha_b \in [0,1]}[(1 + \alpha_a)/(3 + \alpha_b)], \max_{\alpha_a,\alpha_b \in [0,1]}(4 + \alpha_a + \alpha_b)\right] = \left[\frac{1}{4}, \frac{2}{3}\right]$$

A aritmética multidimensional RDM difere da aritmética de Moore apresentada na seção 2.1, tanto em suas definições para as operações aritméticas quanto para o cálculo das mesmas, levando em consideração o uso da variável RDM α. Ao utilizar diferentes métodos para obter valor numérico de um mesmo cálculo, em geral, pretende-se que o resultado seja o mesmo. Entretanto, o cálculo utilizando aritmética RDM pode nem sempre retornar valores de resultado iguais quando comparado à aritmética de Moore justamente pela ideia de considerar os valores entre os limites do intervalo, porém tais resultados ou serão iguais ou bem próximos.

3 Trabalhos Relacionados

O presente capítulo apresenta os trabalhos considerados relevantes e de maior relação com o que está sendo proposto no presente trabalho. Cada seção apresentará, de forma sucinta, diversos conceitos que contribuíram com o entendimento dos conceitos e vêm a enriquecer a base teórica desenvolvida. Além disso, são apresentados alguns resultados obtidos em cada um desses, os quais serviram de referência para o desenvolvimento dos objetivos apresentados.

3.1 Extensão intervalar para as variáveis aleatórias com distribuições Uniforme, Normal, Gama, Exponencial e Pareto

Ao se implantar o uso de intervalos em cálculos computacionais uma das preocupações quando se deseja o controle automático de erros é manter o mesmo esforço computacional quando se utiliza a aritmética com valores reais de entrada. Em vista disso, Finger (2014) desenvolveu sua dissertação com objetivos semelhantes aos apresentados neste trabalho. Definiu a função densidade de probabilidade das variáveis aleatórias contínuas com distribuições Gama e Pareto com entradas intervalares utilizando o método de extensão intervalar. Além disso, ela acrescenta a análise de complexidade dos algoritmos desenvolvidos para computar as funções densidade de probabilidade das variáveis aleatórias com distribuições Uniforme, Exponencial, Normal, Pareto e Gama a partir de resultados obtidos com as implementações utilizando o pacote IntPy.

São apresentadas, quando possível, duas formas de obtenção dos intervalos para as funções densidade citadas, uma utilizando o método de Simpson Intervalar, definido por Caprani, Madsen e Nielsen (2002), e outra a partir da primitiva da função. Acrescendo-se a isso, são apresentados os resultados de qualidade para cada função de acordo com as medidas de erro absoluto e erro relativo, bem como o cálculo do diâmetro do intervalo.

A análise de complexidade de um algoritmo representa a quantidade de trabalho por ele gasto para resolver um determinado problema. O resultado da análise foi obtido a partir dos algoritmos implementados para cada distribuição utilizando a primitiva da função ou método de Simpson, os quais podem ser observados na Tabela 1 apresentada logo a seguir.

Tabela 1 – Esforço computacional das distribuições

Distribuição	**Complexidade Real**		**Complexidade Intervalar**	
	Primitiva da Função	Simpson	Primitiva da Função	Simpson
Uniforme	$O(1)$	-	$O(1)$	-
Exponencial	$O(1)$	$O(2^n)$	$O(1)$	$O(2^n)$
Normal	-	$O(2^n)$	-	$O(2^n)$
Gama	-	$O(2^n)$	-	$O(2^n)$
Pareto	$O(1)$	$O(2^n)$	$O(1)$	$O(2^n)$

Como conclusão de tais resultados, a autora afirma que utilizando a primitiva da função como solução, os algoritmos podem ser executados com menor esforço para computar o resultado das funções com entradas reais, ou seja, em menor complexidade. Para as funções na forma intervalar, a mesma observação é feita, chegando à conclusão de que em ambas as formas, real e intervalar, as melhores soluções podem ser obtidas através da aplicação da primitiva da função. E, além disso, concluiu que a utilização de intervalos nessas distribuições mantém o mesmo esforço computacional que a forma real.

3.2 Estatística Descritiva Intervalar

O trabalho desenvolvido por Loreto (2006) traz como tema a complexidade dos problemas de cálculo de indicadores estatísticos com entradas intervalares. Para isso, foram definidos, com uma abordagem intervalar, alguns indicadores estatísticos através do método de extensão intervalar, tais como: as separatrizes coeficiente de variação intervalar, mediana intervalar, quartil, decil e percentil intervalares. Ela reuniu ainda outros indicadores estatísticos já definidos na forma intervalar, como a média intervalar, moda intervalar, variância intervalar, desvio padrão intervalar, covariância intervalar e coeficiente de correlação intervalar.

A Figura 2 contém uma relação de todas as expressões dos indicadores intervalares contidos na tese de Loreto.

Média Intervalar:	$\mathbf{ME}_v = [\underline{me}, \overline{me}] = \frac{1}{n}[\sum_{i=1}^{n} \underline{x}_i, \sum_{i=1}^{n} \overline{x}_i]$
Mediana Intervalar:	$\mathbf{MD}_v = [\underline{md}, \overline{md}] = \begin{cases} \frac{(\mathbf{x}_{(\frac{n}{2})} + \mathbf{x}_{(\frac{n}{2}+1)})}{2}, & \text{se } n \text{ for par} \\ (\mathbf{x}_{(\frac{n+1}{2})}), & \text{se } n \text{ for ímpar.} \end{cases}$
Moda Intervalar:	$\mathbf{MO}_v = [\underline{mo}, \overline{mo}] = [mo_{1 \leq i \leq n}\{\underline{x}_i\}, mo_{1 \leq i \leq n}\{\overline{x}_i\}]$
Amplitude total Intervalar:	$\mathbf{AT}_v = [\underline{at}, \overline{at}] = \begin{cases} [\underline{ma} - \overline{mi}, \overline{ma} - \underline{mi}], & \text{se } \underline{ma} \geq \overline{mi} \\ [0, \overline{ma} - \underline{mi}], & \text{se } \underline{ma} < \overline{mi}. \end{cases}$
Variância Intervalar:	$\mathbf{VA}_v = [\underline{va}, \overline{va}] = \frac{1}{n-1} \sum_{i=1}^{n} (\mathbf{x_i} - \mathbf{ME})^2$
Desvio padrão Intervalar:	$\mathbf{DP}_v = [\underline{dp}, \overline{dp}] = \sqrt{\mathbf{VA}} = +\sqrt{[\underline{va}, \overline{va}]} = [+\sqrt{\underline{va}}, +\sqrt{\overline{va}}]$
Coeficiente de variação Intervalar:	$\mathbf{CV}_v = [\underline{cv}, \overline{cv}] = \frac{\mathbf{DP}}{\mathbf{ME}} = \frac{[\underline{dp}, \overline{dp}]}{[\underline{me}, \overline{me}]}$
Covariância Intervalar:	$\mathbf{CO}_v = [\underline{co}, \overline{co}] = \frac{1}{n-1} \sum_{i=1}^{n} (\mathbf{x_i} - \mathbf{ME_X})(\mathbf{y_i} - \mathbf{ME_Y})$
Coeficiente de correlação Intervalar:	$\mathbf{CC}_v = [\underline{cc}, \overline{cc}] = \frac{\mathbf{CO}}{\mathbf{DP_X DP_Y}} = (\frac{[\underline{co}, \overline{co}]}{[\underline{dp}_X, \overline{dp}_X][\underline{dp}_Y, \overline{dp}_Y]})$
Separatrizes Intervalares:	$\mathbf{Q}_v = [\underline{q}, \overline{q}], \mathbf{D}_v = [\underline{d}, \overline{d}], \mathbf{P}_v = [\underline{p}, \overline{p}], pos = (n-1)\alpha + 1$

Figura 2 – Indicadores Estatísticos Intervalares

Fonte: (LORETO, 2006).

A fim de mostrar a qualidade de aproximação nos intervalos solução para os indicadores estatísticos intervalares, foi realizado um estudo de caso onde comparou-se os resultados reais com os intervalares para cada um dos indicadores, avaliando se os intervalos resultantes englobam a resposta real exata. Para esse fim, ela utilizou exemplos numéricos através da aplicação de valores de índice de massa corporal de alunos do ensino fundamental. A efetiva certificação foi feita por meio do cálculo da medida de erro, onde, em todos os exemplos considerados, verificou-se que a qualidade no intervalo solução foi mantida.

3.3 Critérios para Análise e Escolha de Ambientes Intervalares

Na existência de vários ambientes computacionais intervalares, muitas vezes é difícil a decisão de qual é o mais adequado ou o melhor para se trabalhar na execução de um futuro trabalho. Pensando nisso, Balboni et al. (2014) desenvolveram uma análise quanto aos critérios de qualidade de software, critérios de linguagens de programação e critérios de qualidade do intervalo solução. Realizaram a análise desses critérios para os seguintes ambientes: Maple Intervalar, IntLab, IntPy, C-XSC, Fortan-XSC, PascalXSC e Java-XSC.

Salientando que não existem na literatura critérios para avaliar ambientes intervalares, mas encontram-se ambientes que são softwares e outros linguagens de programação, o trabalho adota como critérios de avaliação para o melhor ambiente intervalar: avaliação para linguagens de programação, qualidade de software e qualidade do resultado intervalar.

Para a avaliação de linguagens de programação, segundo Sebesta e Mukherjee

(2002), existem critérios que as diferenciam umas das outras, os quais afetam diretamente na finalidade para que a linguagem será projetada, como: legibilidade, manutebilidade, simplicidade, ortogonalidade, suporte a abstração, verificação de tipos e manipulação de exceções. Tais critérios são explicitados detalhadamente no trabalho desenvolvido por Balboni et al. (2014).

Os objetivos de qualidade de um software determinam as propriedades gerais que o produto deve possuir (ROCHA; CAMPOS, 1993). Para Pressman (2011), a qualidade de um software pode ser definida, de modo geral, como uma gestão de qualidade efetiva aplicada de modo a criar um produto útil que forneça valor mensurável para aqueles que o produzem e o utilizam. Foram consideradas algumas características para um software de boa qualidade, as quais são: manutenibilidade, operacionalidade, portatilidade, reutilizabilidade, eficiência, rentabilidade e avaliabilidade.

A fim de verificar qual ambiente retorna o intervalo solução com melhor qualidade (intervalo com menor diâmetro), apresenta-se uma análise da qualidade dos intervalos obtidos na linguagem C-XSC, no pacote IntPy, na biblioteca IntLab e no software Maple Intervalar, os quais possibilitam a programação utilizando operações definidas na matemática intervalar. A análise dá-se através da aplicação dos indicadores estatísticos descritivos intervalares: Media, Mediana, Variância, Amplitude Total, Desvio Padrão, Coeficiente de Variação, Covariância e Coeficiente de Correlação. A Figura 3 apresenta um quadro onde são mostrados os resultados intervalares obtidos a partir de valores reais aplicados aos indicadores citados.

Indicador	**Valores Reais**	**C-XSC**	**IntLab**	**Maple**	**IntPy**
Média	21.2404	[21.235428,21.245429]	[21.2354,21. 2455]	[21.2344,21.2464]	[21.2403757142, 21.2403957142]
Mediana	21.4331	[21.428499, 21.438501]	[21.4284, 21.4386]	[21.4281,21.4381]	[21.4331399999, 21.4331600000]
Ampl. Total	12.7571	[12.746999, 12.767001]	[12.7469, 12.7671]	[12.7471,12.7671]	[12.7570799999, 12.7571200000]
Variância	14.3066	[14.2964, 14.3168]	[14.2442, 14.3677]	[14.2226,14.3906]	[14.3065269110, 14.306772815]
Desvio Padrão	3.78241	[3.774184, 3.790438]	[3.7741, 3.7904]	[3.77361, 3.79121]	[3.78239697956, 3.78242948580]
Coef. Variação	0.178076	[0.177646, 0.178496]	[0.1776, 0.1785]	[0.17764, 0.17850]	[0.17807563618, 0.17807733425]
Covariância	5.151523	[5.105631, 5.197465]	[5.0946, 5.2040]	[5.08052, 5.22252]	[5.15140819363, 5.15163820792]
Coef. Correlação	0.645051	[0.644621, 0.645481]	[0.6345, 0.6556]	[0.63763,0.65247]	[0.64502883893, 0.64507392065]

Figura 3 – Valores reais e intervalares

Fonte: (BALBONI et al., 2014).

Objetivando escolher o melhor ou mais adequado ambiente para se desenvolver trabalhos que requerem controle de erros e garantia de soluções exatas na resolução de problemas, conclui-se, dentre as análises realizadas, que a linguagem de programação Python apresentou os melhores resultados em se tratando de exatidão, verificada com os critérios de qualidade do intervalo, em comparação a C-XSC e aos pacotes Maple Intervalar e IntLab. Além disso, salienta-se que o IntPy é de fácil utilização e instalação, possui manual, é um software gratuito e apresenta exatidão de resultados, sendo este adotado em trabalhos futuros.

3.4 Computação Científica Auto Validável em Python

Varjao (2011) desenvolveu funções matemáticas específicas para a linguagem de programação Python, estendendo as operações possíveis ao utilizar-se o pacote IntPy (BARRETO, 2016).

O cálculo de probabilidades intervalares para as variáveis aleatórias contínuas também foi desenvolvido. Após, realizaram-se comparações de desempenho sobre rotinas que já haviam sido implementadas em outro pacote de desenvolvimento intervalar, o chamado IntLab.

O suporte para a realização de operações matemáticas sobre intervalos é o uso de arredondamentos direcionados (KULISCH; MIRANKER, 2014) nos extremos dos intervalos, ou seja, o limite inferior $\underline{x}$ é arredondado para baixo ($\bigtriangledown$) e o limite superior $\bar{x}$ é arredondado para cima ($\bigtriangleup$). Por exemplo, o intervalo $[2.71828, 3.14159]$ se arredondado utilizando a técnica de arredondamento direcionado, para quatro casas decimais, o intervalo resultante torna-se $[2.7183, 3.1416]$.

A lista completa de funções implementadas por Varjão é apresentada na Tabela 2, a qual contém um quadro apresentando os atributos referentes às operações e sua descrição de retorno de resultado.

Tabela 2 – Funções Implementadas por Varjão

Atributo	**Retorno**
`pow`	Potência do intervalo
`sqrt`	Raiz Quadrada do intervalo
`log`	Logaritmo do intervalo
`exp`	Exponencial do intervalo
`sin`	Seno do intervalo
`cos`	Cosseno do intervalo
`tan`	Tangente do intervalo
`asin`	Arco Seno do intervalo
`acos`	Arco Cosseno do intervalo
`atan`	Arco Tangente do intervalo
`sinh`	Seno Hiperbólico do intervalo
`cosh`	Cosseno Hiperbólico do intervalo
`tanh`	Tangente Hiperbólica do intervalo

Tais funções estão contidas na classe `stdfunc` (*standard functions*), a qual herda o atributo $\mathbb{IR}$eal e, como característica principal, possui a extensão das operações listadas acima sobre o atributo $\mathbb{IR}$eal.

Ele conclui que que o IntPy obteve resultados satisfatórios quando comparado aos resultados encontrados no Intlab, em tempo de processamento e em precisão de máquina, não encontrando dificuldades para realizar as implementações das funções intervalares em Python.

3.5 Diferenças entre Aritmética de Moore e Aritmética Intervalar RDM

Partindo do conceito de que a teoria da incerteza busca solucionar problemas numéricos envolvendo dados incertos, a computação numérica intervalar se faz necessária para a computação desses problemas.

Sendo a aritmética intervalar desenvolvida por Moore a mais conhecida e também, aparentemente, a mais utilizada pela comunidade científica, o trabalho desenvolvido por Landowski (2015) faz um comparativo entre a aritmética de Moore e a aritmética intervalar multidimensional RDM. Para isso, ele apresenta tanto as operações aritméticas básicas de soma, subtração, multiplicação e divisão definidas na aritmética de Moore, quanto na forma definida na aritmética multidimensional RDM. Além disso, são utilizados exemplos numéricos como meio comparativo entre as duas metodologias apresentadas.

No trabalho de Landowski são apresentadas ilustrações que permitem perceber como os valores dos intervalos são dispostos no plano, salientando que os valores entre os extremos, inferior e superior, do intervalo são considerados na arimética multidimensional. A ilustração construída por Landowski é apresentada na Figura 4. Nesta ilustração são apresentadas quatro disposições no plano tridimensional, na qual são demonstradas cada uma das operações sob a abordagem multidimensional RDM. Dois intervalos foram utilizados nos cálculos: $A = [1, 2]$ e $B = [3, 4]$.

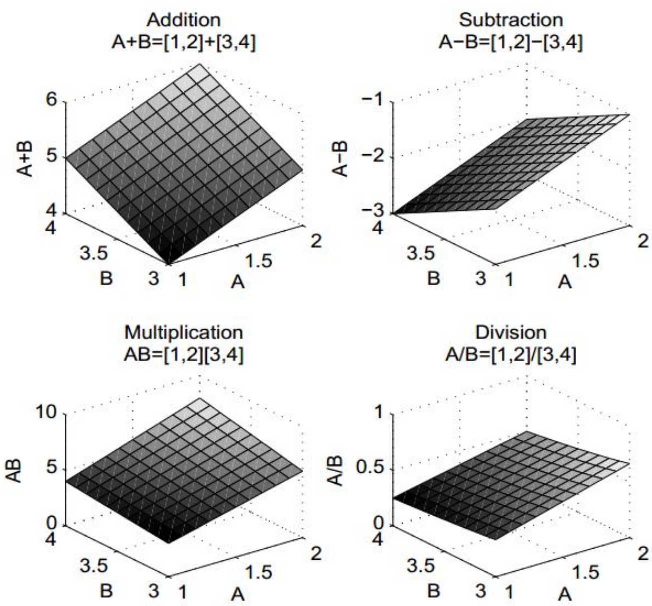

Figura 4 – Plano tridimensional para as operações elementares utilizando aritmética RDM

Fonte: (LANDOWSKI, 2015).

Além das operações básicas, o trabalho apresenta propriedades como comutatividade, associatividade, elementos neutros da multiplicação e divisão, inversão de elementos, sub distributivo e cancelamento.

Por fim, Landowski conclui que a aritmética de Moore não resolve problemas mais complexos e que suas soluções são unidimensionais, não apresentando uma solução completa. Por outro lado, a aritmética multidimensional RDM apresenta os mesmos resultados para diferentes formas de uma mesma equação.

3.6 Duas Interpretações para a Aritmética Multidimensional RDM - Multiplicação e Divisão

O trabalho desenvolvido por Piegat e Landowski (2013b) é na verdade continuação de outro desenvolvido anteriormente, onde tratou-se das operações de soma e subtração. Neste, são apresentadas duas interpretações e formas de realização para as operações de multiplicação e divisão utilizando intervalos. Uma delas utilizando o que chamam de interpretação possibilística, onde as operações são realizadas da forma como são apresentadas da aritmética multidimensional RDM convencional, a qual os autores apontam ser muito importante no contexto de sistemas fuzzy. A segunda interpretação é chamada de probabilística condicional, na qual são utilizadas distribuições de densidade de probabilidade, no caso, uma distribuição Uniforme. Ambas são exemplificadas no decorrer do trabalho sob conceito da aritmética multidimensional RDM.

Os autores ressaltam que ambas as abordagens não são contraditórias, mas sim complementares e, além disso, expõem que a possibilidade de realizar cálculos numéricos utilizando intervalos de duas maneiras distintas é um argumento científico na proposta de duas abordagens diferentes para a modelagem da teoria da incerteza: a possibilística e a probabilística.

Como forma de colocar à prova a aritmética intervalar de Moore, neste trabalho, Piegat e Landowski levantam a questão de que a aritmética de Moore trata dos valores envolvidos no cálculo anonimamente, não levando em consideração se as variáveis pertencem ao mesmo contexto do problema. Além de que apenas as "fronteiras" do intervalo são levadas em consideração, ou seja, apenas os limites superior e inferior, não considerando-se os valores entre esses limites. Cabe citar outro trabalho destes mesmos autores, intitulado "*Is the conventional interval arithmetic correct?*" (PIEGAT; LANDOWSKI, 2012), onde mais uma vez a aritmética de Moore é questionada quanto à sua corretude para todas as aplicações.

Como conclusão, Piegat e Landowski salientam que a versão possibilística para as operações, ou usual, possui um grande significado prático pois, em problemas práticos, normalmente não se tem conhecimento aprofundado sobre distribuições de probabilidade para que sejam aplicadas, apenas se conhecem os limites do intervalo representando os valores de aproximação para valores desconhecidos exatamente.

4 Definição Intervalar da Função Densidade de Probabilidade com Distribuição Beta

O presente capítulo tem como objetivo apresentar a definição intervalar para a função densidade de probabilidade da variável aleatória contínua com distribuição Beta desenvolvida a partir do método de extensão intervalar, sendo esta computada sob duas abordagens diferentes: a aritmética intervalar definida por Moore (1966) e a aritmética multidimensional RDM, de Piegat e Landowski (2012). Com o intuito de comparar e apurar a exatidão dos resultados obtidos, fez-se uso de um ambiente computacional de desenvolvimento intervalar, realizando-se a implementação dos algoritmos para a função densidade com distribuição Beta utilizando a linguagem de programação Python (PYTHON, 2016) e o pacote intervalar IntPy (BARRETO, 2016), tanto para entradas de valores reais quanto intervalares. A partir disso, tornou-se possível realizar uma análise numérica dos resultados, o que será feito a partir da comparação dos valores de resultados intervalares com os reais aplicando-se métricas de erros definidas na aritmética intervalar para estimar a qualidade dos intervalos.

Existem diversos ambientes computacionais para o desenvolvimento de algoritmos para cálculos numéricos que utilizam matemática intervalar na sua estrutura. No entanto, como mencionado, o presente trabalho fará uso da linguagem Python e o pacote IntPy para codificar os algoritmos. Essa escolha se deu a partir de comparações já realizadas anteriormente entre os ambientes intervalares mais utilizados na literatura, onde o IntPy demonstrou ser o mais eficiente, retornando intervalos com melhor qualidade e com menor tempo de processamento (BALBONI et al., 2014).

A implementação de operações e algoritmos de forma intervalar em computadores é realizado por meio do critério de semimorfismo, o qual foi proposto por Kulisch e Miranker (2014). Considerando que o problema do controle de erro numérico pode ser feito por meio do uso de intervalos ao invés de números reais, Kulisch e Miranker propuseram que a implementação da aritmética intervalar seja realizada utilizando a chamada aritmética de exatidão máxima, a qual também foi desenvolvida por Kulisch e Miranker, significando a busca para que resultados numéricos ou sejam um número de ponto flutuante ou estejam entre dois números em ponto flutuante consecutivos. Isso se dá pelo fato de que qualquer tentativa de implementação através da aritmética intervalar em sistemas de computador, o espaço de intervalos não está definido sobre o conjunto dos números reais, $\mathbb{R}$, mas sobre o sistema de ponto flutuante (BARRETO; CAMPOS, 2008).

Na busca por maior exatidão de resultados, os cálculos numéricos precisam ser suportados, concomitantemente, pela matemática intervalar e pela aritmética de exatidão máxima. Nesse contexto, implica que, em computadores, esses cálculos sejam computados através de linguagens de programação ou bibliotecas que possuam o tipo intervalo e operações sobre intervalos definidas, as quais são usualmente denominadas linguagens XSC (*eXtended Scientifc Computations*) (MESQUITA, 2002).

A aritmética de exatidão máxima redefine a aritmética intervalar de modo que os limites inferior e superior de um intervalo possuam um arredondamento do sistema de ponto flutuante, chamado de arredondamento direcionado (KULISCH; MIRANKER, 2014), por exemplo, uma operação de adição entre dois intervalos, **x** e **y**, se dá da seguinte forma:

$$\mathbf{x} + \mathbf{y} = [\bigtriangledown(\underline{x} + \underline{y}), \bigtriangleup(\bar{x} + \bar{y})],$$

onde, $\bigtriangledown$ constitui o arredondamento direcionado para baixo e $\bigtriangleup$ se refere ao arredondamento direcionado para cima.

Em um computador de uso geral, quando são realizados cálculos numéricos, os valores de entrada normalmente são fornecidos na base decimal (RUGGIERO; LOPES, 1997), necessitando que estas informações sejam convertidas para a base binária, a qual pode ser computada pelo sistema, e depois, convertida novamente para gerar uma saída ao usuário. Tais transformações podem ser fontes de erros.

A representação numérica utilizada nos sistemas de computador é chamada de Notação de Ponto Flutuante, sendo que, em qualquer máquina, apenas um subconjunto dos números reais, $\mathbb{R}$, é representado exatamente, e, portanto, a apresentação de um número real será realizada através de truncamento ou arredondamento. A representação de um número depende da base escolhida ou disponível na máquina em uso e do número máximo de dígitos usados em sua representação (RUGGIERO; LOPES, 1997).

No sistema em ponto flutuante, um número é representado por quatro valores, na forma $F(\beta, t, m, M)$, onde: β refere-se à base utilizada, t o tamanho ou número de dígitos da mantissa (parte fracionária), m é o menor expoente possível para a notação e, M o maior expoente para a notação. Em relação à esta representação, quanto maior o número de dígitos na mantissa, maior será a precisão dos números representáveis e menor será o intervalo entre dois números pertencentes ao sistema de ponto flutuante. Para o expoente, quanto maior o número de bits disponíveis nesta representação, maior serão os limites dos números.

Devido a essa representação, podem ocorrer dois tipos de erros: um relacionado à quantidade de bits disponíveis para representar o expoente e outro em relação ao tamanho da mantissa. Quanto ao limite do expoente, quando uma operação aritmética produz um expoente maior que o máximo de M, ocorre um erro de armazenamento, gerando uma exceção chamada *overflow*. Analogamente, operações que resultam em expoentes menores que o expoente mínimo, m, geram uma exceção denominada *underflow*.

No processo de resolução de problemas através da aplicação de métodos numéricos nem sempre se obtêm resultados exatos como pretendido. Segundo Ratschek e Rokne (1988), na aritmética de ponto flutuante, números reais são aproximados por um subconjunto dos reais, chamados números de máquina. Define-se por erro a diferença entre o valor obtido por meio de aproximação e o valor exato. Nesse contexto, as fontes de erros podem ser devido a vários fatores, os quais se pode especificar mais detalhadamente:

- **Erros inerentes aos dados iniciais:** um modelo matemático não contém apenas equações e relações, mas também é constituído de dados e parâmetros para os quais se atribuem valores cuja fonte pode ser de uma medida experimental, e, portanto, aproximados. Isso caracteriza uma fonte de erros externa ao processo matemático computado, estando relacionado com a incerteza dos dados de entrada;
- **Erros inerentes ao modelo:** um modelo matemático raramente oferece uma representação exata do problema que se deseja solucionar, visto que as vezes se faz necessária uma simplificação do problema para torná-lo solucionável. Em vista disso, um bom modelo seria o que inclui características do problema real, através do qual se pode reproduzir os erros com um nível aceitável (VALENÇA, 1996);
- **Erro de arredondamento:** como para cálculos realizados em computador existe uma aritmética de precisão finita na qual se pode levar em consideração um número finito de dígitos, após uma operação ser realizada em ponto flutuante, o resultado é normalizado e o arredondamento pode ser necessário. Chama-se então de erro de arredondamento o erro gerado devido a desprezar-se números de casas decimais adiante e arredondar o valor;
- **Erro de truncamento:** na existência de problemas que não podem ser solucionados por um número finito de operações aritméticas, porém podem ser aproximados por problemas cuja solução é obtida executando uma sequência finita desse tipo de operações, erros de truncamento são causados ao se truncar sequências infinitas de operações aritméticas após um número finito de etapas.

Nesse contexto, percebe-se a importância de técnicas intervalares, pois a aritmética intervalar fornece uma ferramenta para estimar e controlar esses erros automaticamente. No lugar de aproximar um valor real x por um número de máquina, esse valor, usualmente desconhecido, é aproximado por um intervalo $\mathbf{x}$ contendo números de máquina como extremos inferior e superior. O intervalo $\mathbf{x}$ contém o valor x.

4.1 Definição Intervalar

A distribuição de probabilidades de uma variável aleatória X é uma descrição do conjunto das probabilidades associadas com os valores possíveis para X. Similarmente, uma função densidade de probabilidade $f(x)$ pode ser usada para descrever a distribuição de probabilidades de uma variável aleatória contínua X onde a probabilidade de X estar entre dois valores, a e b, é determinada pela integral definida dessa função no intervalo de a até b. A função densidade de uma variável aleatória contínua usada para determinar probabilidades segue conforme a Equação 4.1:

$$P(a < X < b) = \int_a^b f(x)dx \tag{4.1}$$

Seja X uma variável aleatória. Supondo que o contradomínio ($\mathbb{R}_x$ de X seja um intervalo ou uma coleção de intervalos. Então diremos que X é uma variável aleatória contínua se existe uma função $f_X : \mathbb{R} \to [0, +\infty)$, que satisfaz às propriedades:

1. $f_X(x) \geq 0$, para todo $x \in \mathbb{R}_x$
2. $\displaystyle\int_{-\infty}^{\infty} f(x)dx = 1$

Nesse caso, a variável aleatória X pode assumir uma quantidade não enumerável e a função f_X é denominada função densidade de probabilidade.

As variáveis aleatórias contínuas podem ser classificadas pela qualidade e pelo número de propriedades ou características de interesse dos entes estudados. Se S é o espaço amostral de um experimento qualquer E; partindo do princípio que S de E é constituído de N pontos e X de S, de n pontos, e que tanto N quanto n podem ser finitos ou infinitos, enumeráveis ou não enumeráveis, a variável aleatória associada aos eventos S de E e X de S é dita contínua (inumerável ou intervalar) se tais pontos não podem ser contados.

A distribuição Beta é um modelo probabilístico para uma variável aleatória contínua X, cujos valores possíveis são limitados superior e inferiormente. Na forma da distribuição Beta padronizada, a variável X é definida no intervalo $[0, 1]$, podendo ser usada para modelagem de proporções ou modelagem de objetos que pertencem a esse intervalo.

Devido à grande versatilidade de uma variável aleatória com distribuição Beta para modelar funções densidade de probabilidade no intervalo $[0, 1]$, além da possibilidade de generalizar essa função para qualquer variável aleatória restrita a um intervalo finito (m, n), a função densidade de probabilidade com distribuição Beta possui inúmeras aplicações para representar quantidades físicas cujos valores estejam restritos a um intervalo identificável. Alguns exemplos de aplicação da função densidade com distribuição

Beta podem ser citados: ajuste de velocidade em uma rede de computadores; em modelos para calcular velocidade e direção predominante do vento, intensidade pluvial ou índices de umidade em sistemas meteorológicos; modelos para aceleração de atividades em projetos entre outros.

Nesse caso, para $0 \leq x \leq 1$, $\alpha > 0$, $\beta > 0$, a função densidade de probabilidade Beta é expressa pela Equação 4.2:

$$f_X(x) = \frac{1}{B(\alpha, \beta)} \int_0^a x^{\alpha-1}(1-x)^{\beta-1} dx \tag{4.2}$$

na qual, α e β são parâmetros e $B(\alpha, \beta)$ representa a função Beta completa, a qual é descrita por:

$$B(\alpha, \beta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)} \tag{4.3}$$

Para a Equação 4.3, o termo $\Gamma(n)$ representa a função Gama, a qual se reduz ao fatorial de n, sendo n um valor numérico. A função Gama é descrita a seguir pela Equação 4.4, onde y representa um parâmetro de entrada dessa função.

$$\Gamma(n) = \int_0^\infty y^{n-1} e^{-y} dy = (n-1)! \tag{4.4}$$

De modo geral, para valores iguais para ambos os parâmetros, a função densidade de probabilidade com distribuição Beta é simétrica; contrariamente, a função é assimétrica. Caso os parâmetros sejam superiores a 1, $\alpha > 1$ e $\beta > 1$, então a função densidade com distribuição Beta é unimodal [1]. A Figura 5 ilustra algumas formas possíveis para a função densidade de probabilidade com distribuição Beta.

[1] Distribuição probabilística que apresenta apenas um ponto de mínimo ou máximo.

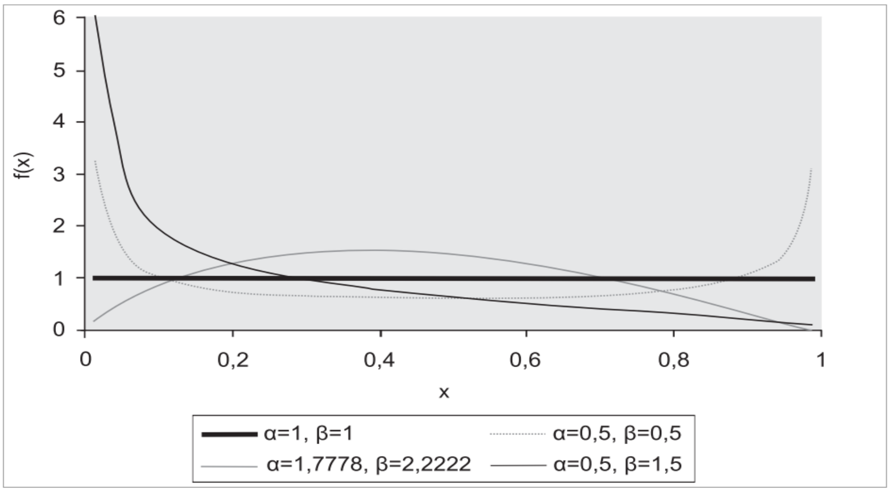

Figura 5 – Exemplos de funções densidades de probabilidade da Distribuição Beta

Fonte: (NAGHETTINI; PINTO, 2007).

Na Figura 5, pode-se observar que a distribuição de probabilidade uniforme é um caso particular da distribuição Beta, ocorrendo quando os parâmetros são iguais a 1 ($\alpha = 1$ e $\beta = 1$). O parâmetro α controla os valores da densidade Beta em correspondência ao limite inferior da variável. Analogamente, o parâmetro β controla os valores da densidade Beta em correspondência ao limite superior.

Utilizando o método da extensão intervalar, a fórmula da função densidade de probabilidade da variável aleatória contínua com distribuição Beta é expressa a seguir na Equação 4.5.

$$f_{B_{int}}(\mathbf{x}, \alpha, \beta) = \frac{1}{B(\alpha, \beta)} \int_0^a \mathbf{x}^{\alpha-1}(1-\mathbf{x})^{\beta-1} dx \tag{4.5}$$

Seguindo tal método para se criar a definição intervalar da função densidade com distribuição Beta, fez-se necessário substituir todas as entradas de x da Equação 4.2, a qual antes comportava valores reais, $\mathbb{R}$, para entradas intervalares.

Para computar a função densidade de probabilidade da variável aleatória contínua com distribuição Beta, tanto na forma real quanto intervalar, utilizou-se apenas a fórmula da densidade de probabilidade desta distribuição para realizar seus cálculos.

4.2 Análise Numérica

Como forma de obter os resultados numéricos, foram utilizados cinco conjuntos de dados de entrada como parâmetros para as funções real e intervalar da distribuição Beta, tais dados se referem a valores de exemplos arbitrários escolhidos para que fosse possível computar os dados e obter as soluções pretendidas. Os dados de entrada se referem aos parâmetros x, α e β da função densidade com distribuição Beta. Abaixo são apresentados os conjuntos de dados utilizados na computação da função:

- **exemplo 1:** $x = 0.2222$, $\alpha = 1.7778$, $\beta = 2.2222$ e $a = 1$;
- **exemplo 2:** $x = 0.322$, $\alpha = 1.777$, $\beta = 3.228$ e $a = 1$;
- **exemplo 3:** $x = 0.750$, $\alpha = 1.760$, $\beta = 2.760$ e $a = 1$;
- **exemplo 4:** $x = 0.188$, $\alpha = 5.031$, $\beta = 10.411$ e $a = 1$;
- **exemplo 5:** $x = 0.13$, $\alpha = 3.07$, $\beta = 5.82$ e $a = 1$.

Tais valores procedem de exemplos práticos, como cálculos químicos para concentração de gases e líquidos, ou valores escolhidos arbitrariamente desde que respeitassem as características da função densidade com distribuição Beta.

Seguindo o objetivo deste trabalho, duas abordagens diferentes da aritmética intervalar são aplicadas como métodos para calcular os valores intervalares de resultado a partir da computação da função densidade com distribuição Beta, sendo a aritmética intervalar de Moore e a aritmética intervalar multidimensional RDM.

A Tabela 3 apresenta os resultados obtidos através da computação da função densidade com distribuição Beta utilizando a aritmética intervalar de Moore e implementação no IntPy. Nela, são apresentados os valores obtidos a partir de cada exemplo aplicado, tanto na forma real quanto intervalar dessa função e, de forma complementar, apresenta-se o tempo de execução de cada computação, expresso em segundos.

Tabela 3 – Aplicação dos parâmetros à função densidade com distribuição Beta nas formas real e intervalar sob Aritmética Intervalar de Moore

	Valor Real	**Valor Intervalar**
Exemplo 1	1.326767382571450105	[1.326767382571449883, 1.326767382571450105]
Tempo(s)	0.00029	0.00032
Exemplo 2	1.827952566032753579	[1.827952566032753356, 1.827952566032753579]
Tempo(s)	0.00018	0.00029
Exemplo 3	0.560744430584336095	[0.560744430584335984, 0.560744430584336095]
Tempo(s)	0.00021	0.00034
Exemplo 4	2.058738196035631685	[2.058738196035631240, 2.058738196035631685]
Tempo(s)	0.00006	0.00016
Exemplo 5	1.262740949463830420	[1.262740949463830198, 1.262740949463830420]
Tempo(s)	0.00019	0.00038

Analisando a Tabela 3, é possível perceber que o valor real está contido no intervalo solução em todos os exemplos aplicados. Os valores de tempo de processamento apresentados foram coletados a partir de dez execuções para cada cálculo computado, fazendo-se a média. Pode-se observar que tais valores são próximos aos apresentados para o cálculo com valores reais, indicando que o uso da aritmética intervalar não interfere significativamente no tempo de processamento.

A Tabela 4 apresenta os resultados obtidos através da computação da função densidade com distribuição Beta utilizando a aritmética intervalar multidimensional RDM, também com implementação no IntPy. Nela, são apresentados os valores obtidos a partir da aplicação dos mesmos exemplos anteriores. Para esta coleta de resultados, apenas os valores intervalares e o tempo de execução (em segundos) de cada computação serão apresentados.

Tabela 4 – Aplicação dos parâmetros à função densidade com distribuição Beta intervalar sob Aritmética Multidimensional RDM

	Valor Intervalar
Exemplo 1	[1.326767382571449883, 1.326767382571450105]
Tempo(s)	0.00041
Exemplo 2	[1.827952566032753356, 1.827952566032753579]
Tempo(s)	0.00034
Exemplo 3	[0.560744430584335984, 0.560744430584336095]
Tempo(s)	0.00034
Exemplo 4	[2.058738196035631240, 2.058738196035631685]
Tempo(s)	0.00020
Exemplo 5	[1.262740949463830198, 1.262740949463830420]
Tempo(s)	0.00022

Como análise da Tabela 4, com valores apresentados a partir da computação aplicando-se a aritmética intervalar multidimensional RDM, pode-se perceber que os valores de resultado intervalares são iguais aos apresentados quando aplicada a aritmética intervalar de Moore. Em vista disso, é possível se chegar a duas conclusões: a primeira em relação à exatidão dos resultados intervalares, visto que era esperado que os valores ou fossem iguais ou muito próximos; a segunda, em relação direta à aritmética multidimensional RDM, é que a função densidade com distribuição Beta possui apenas uma variável do escopo do problema a ser computado, sendo os demais valores, α e β, parâmetros com valores predefinidos, o que faz a solução ser unidimensional, assim como na aritmética de Moore aplicada. Para os valores de tempo de processamento apresentados, pode-se verificar que são próximos ou até mesmo iguais aos apresentados na tabela com os valores obtidos através da aritmética de Moore, o que indica que o uso da aritmética multidimensional RDM também não interfere de forma significativa no tempo de processamento dos cálculos.

Para todos os resultados dos testes, fora utilizado o mesmo computador com as seguintes configurações: processador Intel Core i5-4210U @ 2.40GHz Quad-Core, L1 Cache 128Kb, L2 Cache 512Kb, L3 Cache 3Mb, Memoria RAM de 4GB DDR3 1600MHz, armazenamento HD Sata 1TB modelo ST1000LM024 HN-M101MBB, sistema operacional Linux Ubuntu 16.04 LTS.

Todos os resultados apresentados, tanto reais quanto intervalares, utilizaram o sistema de ponto flutuante F(10, 18, -17, 17).

Para verificar a qualidade dos resultados, realiza-se a análise de erros sobre os intervalos resultantes. A computação utilizando intervalos fornece duas estimativas principais para o erro: erro Absoluto, referente a diferença entre o valor exato de um número e seu valor aproximado; e erro Relativo, estando relacionado ao erro percentual podendo ser calculado pelo quociente entre o erro absoluto e o valor aproximado. Acrescendo-se às possíveis formas de apurar a qualidade do resultado obtido, o diâmetro do intervalo é uma métrica para mensurar a qualidade de aproximação. Sendo que na computação utilizando aritmética intervalar os cálculos são executados utilizando intervalos e, consequentemente, a aritmética real é substituída pela aritmética intervalar, as estimativas para o erro oferecidas pela aritmética intervalar são descritas a seguir:

- **Erro Absoluto:** $\mid x - m(\mathbf{x}) \mid < \frac{w(\mathbf{x})}{2}$, onde $m(\mathbf{x}) = \left(\frac{\underline{x}+\bar{x}}{2}\right)$ é o ponto médio do intervalo $\mathbf{x}$;
- **Erro Relativo:** $\left|\frac{x-m(\mathbf{x})}{x}\right| \leq \frac{w(\mathbf{x})}{2min|\mathbf{x}|}$, se $0 \notin \mathbf{x}$;
- **Diâmetro:** $diam(\mathbf{x}) = w(\mathbf{x}) = \bar{x} - \underline{x}$.

Nas medidas de erros, utiliza-se o ponto médio $m(\mathbf{x})$ do intervalo $\mathbf{x}$ para medir a distância do valor real em relação ao valor pontual do intervalo. Uma interpretação usual no contexto da matemática intervalar é a de envoltória intervalar de um número real. Esta semântica sugere a representação dos intervalos na forma $m(\mathbf{x}) \pm \frac{w(\mathbf{x})}{2}$, aludindo à ideia de que o ponto médio seria o número real "medido" e o raio, $\frac{w(\mathbf{x})}{2}$, indicaria a incerteza gerada pelas restrições de precisão existentes. De tal forma, o valor exato estaria limitado pelo intervalo apresentado, entre seu limite inferior $\underline{x}$ e seu limite superior $\bar{x}$ (SUNAGA, 2009).

Tais medidas de erros foram aplicadas nos valores de resultados intervalares obtidos na computação da função densidade com distribuição Beta com o objetivo de verificar a qualidade do intervalo solução, além de que o cálculo do diâmetro forneceu uma medida de qualidade de aproximação do intervalo solução em relação ao valor real.

A Tabela 5 apresenta os valores obtidos para a estimativa de erro absoluto e erro relativo em relação aos intervalos resultantes da aplicação dos dados dos exemplos à distribuição Beta. A seguir, na Tabela 6, apresenta-se a medida do diâmetro dos intervalos obtidos a partir do cálculo computado.

Tabela 5 – Qualidade dos intervalos para a variável aleatória contínua com distribuição Beta

	Erro Absoluto	**Erro Relativo**
Exemplo 1	$0.0 < 1.11022302463 \times 10^{-16}$	$0.0 \leq 8.36788000074 \times 10^{-17}$
Exemplo 2	$0.0 < 1.11022302463 \times 10^{-16}$	$0.0 \leq 6.07358771368 \times 10^{-17}$
Exemplo 3	$0.0 < 5.55111512313 \times 10^{-17}$	$0.0 \leq 9.89954571166 \times 10^{-17}$
Exemplo 4	$0.0 < 2.22044604925 \times 10^{-16}$	$0.0 \leq 1.07854707001 \times 10^{-16}$
Exemplo 5	$0.0 < 1.11022302463 \times 10^{-16}$	$0.0 \leq 8.79216774507 \times 10^{-17}$

Tabela 6 – Diâmetro dos intervalos para a variável aleatória com distribuição Beta

	Diâmetro do Intervalo
Exemplo 1	0.000000000000000222
Exemplo 2	0.000000000000000222
Exemplo 3	0.000000000000000111
Exemplo 4	0.000000000000000444
Exemplo 5	0.000000000000000222

No erro absoluto, verifica-se que a diferença entre o valor exato e o ponto médio do intervalo ocorre, em relação aos exemplos, na 16ª casa decimal. Além de que, em todos os resultados, constata-se que a condição $\mid x - m(\mathbf{x}) \mid$ é menor ou igual que a metade do diâmetro, $w(\mathbf{x})$, do intervalo. Isso mostra que este critério de estimativa para o erro é satisfeito.

Para o erro relativo, a desigualdade se manteve válida em todos os resultados obtidos, indicando a qualidade do intervalo. Em vista dos valores de diâmetro apresentados, observa-se que até a 15ª casa decimal para todos os exemplos o diâmetro dos intervalos é nulo. Verificando-se, dessa forma, que os intervalos possuem qualidade de aproximação do valor real.

Cabe salientar ainda que as fórmulas de medidas de erros absoluto, erro relativo e diâmetro dos intervalos também foram codificadas na linguagem Python, possibilitando a computação e obtenção de tais estimativas.

Neste capítulo foram apresentados os valores de resultado a partir da computação da função densidade de probabilidade da variável aleatória contínua com distribuição

Beta, tanto para a forma real quanto intervalar. Para tanto, fez-se necessário a implementação de algoritmos, os quais foram desenvolvidos no ambiente computacional de desenvolvimento intervalar composto pela linguagem de programação Python e o pacote IntPy, sendo que, para a forma intervalar da função densidade com distribuição Beta, duas abordagens da aritmética intervalar foram utilizadas neste trabalho: a aritmética intervalar desenvolvida por Moore e a aritmética multidimensional RDM idealizada por Piegat e Landowski. Após, a fim de comparar os resultados intervalares aos reais e, principalmente, justificar quanto à qualidade dos intervalos obtidos, realizou-se uma análise de qualidade utilizando as métricas de erros definidas na aritmética intervalar.

Com a finalidade de complementar o trabalho realizado e cumprir mais um dos objetivos iniciais, o próximo capítulo apresenta a análise de complexidade para cada implementação desenvolvida em vista da função densidade com distribuição Beta, real e intervalar.

5 Análise de Complexidade

Segundo Toscani e Veloso (2009), a complexidade de um algoritmo reflete o esforço computacional requerido para executá-lo. Tal esforço não pode ser descrito apenas por um número, porque o número de operações efetuadas na computação de um problema, em geral, não é o mesmo para qualquer entrada, mas depende do tamanho desta e ainda, para entradas de mesmo tamanho, a quantidade de operações pode variar.

Um algoritmo é um procedimento, consistindo em um conjunto de regras não ambíguas, as quais especificam, para cada entrada, uma sequência finita de operações, terminando com uma saída correspondente (TOSCANI; VELOSO, 2009). Acrescendo-se a isso, diz-se que um algoritmo resolve, teoricamente, um problema produzindo uma resposta correta para qualquer entrada, sendo concedidos tempo e memória suficientes para sua execução.

Diversos critérios podem ser utilizados na escolha de um algoritmo para solucionar um problema. Na maioria das vezes, essa escolha é feita através de critérios subjetivos como: facilidade de compreensão, codificação e depuração, eficiência na utilização dos recursos do computador e rapidez (RITT; BURIOL; PRESTES, 2009).

Em aplicações práticas, o fato de um algoritmo resolver um problema, teoricamente, gerando uma saída, não significa que seja aceitável, pois recursos de tempo e espaço em memória necessários para a execução do algoritmo têm grande relevância. Para isso, existe a análise de algoritmos, a qual busca classificar os algoritmos de acordo com a quantidade de trabalho necessário para realizar computações seguindo critérios de avaliação para suas características de implementação e execução, processo que resulta em definir a complexidade do algoritmo.

Questões relativas à complexidade de um algoritmo em termos do tempo de computação e espaço de memória são determinantes para o julgamento da eficiência do mesmo (LORETO, 2006). Essa eficiência depende diretamente de quantos passos computacionais são necessários para que o algoritmo chegue à solução do problema, seguindo um modelo matemático denominado Máquina de Turing (MENEZES, 1998). Podendo-se, dessa forma, determinar se o algoritmo é razoável ou não.

Para Kreinovich et al. (2013), um algoritmo é considerado razoável quando obtém a solução de um problema em tempo polinomial. A caracterização de algoritmo de tempo polinomial baseia-se em duas noções fundamentais: tamanho da entrada e tempo de execução. Em geral, esses problemas são considerados tratáveis, mas por razões filosóficas, e não matemáticas (LEISERSON et al., 2002).

Existem vários aspectos a considerar na determinação da complexidade de um algoritmo, como medidas de desempenho como tempo e espaço requeridos por um algoritmo, relacionadas à velocidade e à quantidade de memória, respectivamente. Para mensurar a quantidade de trabalho realizado por um algoritmo, costuma-se escolher uma operação, chamada operação fundamental. A operação escolhida como fundamental deve ser tal que a contagem do número de vezes que ela é executada expresse a quantidade de trabalho do algoritmo, dispensando outras medidas (TOSCANI; VELOSO, 2009).

As notações utilizadas para descrever o tempo de execução assintótica de um algoritmo são definidas em termos de funções cujos domínios são o conjunto dos números naturais $\mathbb{N}$. Tais notações são convenientes para descrever a função do tempo de execução no pior caso $T(n)$, que em geral, é definida somente sobre tamanhos de entrada inteiros (LEISERSON et al., 2002).

A seguir são apresentadas as análises de complexidade para cada um dos algoritmos implementados para computar a função densidade com distribuição Beta, tanto sob as diferentes abordagens da aritmética intervalar utilizadas, quanto para a sua forma real. Primeiramente, apresenta-se a análise da implementação para entradas reais. Em seguida, é apresentada a análise para a forma intervalar da função Beta codificada seguindo os aspectos da aritmética intervalar de Moore e, posteriormente, esta análise é desenvolvida a partir do uso da aritmética intervalar multidimensional RDM, de Piegat e Landowski, para implementação dessa função. Todos os algoritmos analisados podem ser encontrados no Anexo A.

- **Função Densidade com Distribuição Beta para Entradas Reais:** Como valores de entrada, é necessário fornecer ao algoritmo os valores correspondentes aos parâmetros da função densidade com distribuição Beta para que seja possível computar o cálculo, os quais são x, α e β. O cálculo da função requer basicamente operações aritméticas de potência, multiplicação, subtração e divisão. Uma integral definida precisa ser resolvida, a qual é feita por meio de uma função própria da linguagem Python chamada "*integrate*", para a qual foi atribuído o método "*quad*", responsável por integrar uma função de uma variável entre dois pontos, possuindo ordem de complexidade constante. Como o desenvolvimento do algoritmo se deu através da implementação dessas operações, a ordem de complexidade pode ser dada como constante, ou seja, $O(1)$.

- **Função Densidade com Distribuição Beta para Entradas Intervalares utilizando a Aritmética de Moore:** O algoritmo intervalar implementa a mesma equação da função densidade de probabilidade Beta, utilizando as mesmas entradas que do algoritmo real. Entretanto, todas as operações aritméticas antes realizadas na forma real, foram substituídas por operações intervalares. Sendo assim, a complexidade para resolver tais operações continua com ordem constante, como no algoritmo real. Dessa forma, a complexidade para computar a função densidade de probabilidade da variável aleatória contínua com distribuição Beta de forma intervalar permanece com ordem de complexidade constante $O(1)$.

- **Função Densidade com Distribuição Beta para Entradas Intervalares utilizando a Aritmética Multidimensional RDM:** Para computar a função densidade com distribuição Beta utilizando a aritmética multidimensional RDM, fez-se necessário implementar todas as operações aritméticas básicas de soma, subtração, multiplicação, divisão e potência acordo com essa abordagem, pois não estavam definidas em termos de linguagem de programação ou biblioteca para uso direto. Isso não afeta na complexidade do algoritmo desenvolvido, pois todas essas operações podem ser computadas em complexidade de ordem constante. Para a função densidade com distribuição Beta, a implementação ocorre da mesma maneira que no algoritmo intervalar utilizando aritmética de Moore, porém, utilizando as operações definidas como mencionado anteriormente. Isso indica que o algoritmo também possui complexidade de ordem constante, então se pode concluir que é de ordem $O(1)$.

A complexidade de um algoritmo representa a quantidade de trabalho despendido para resolver um determinado problema em função de recursos computacionais e do tempo necessário. No presente trabalho, efetuou-se a análise de complexidade para os algoritmos desenvolvidos para computar a função densidade de probabilidade com distribuição Beta em vista de que foram utilizadas duas abordagens para a implementação utilizando aritmética intervalar e também a forma real da função. A Tabela 7 apresenta um resumo das complexidades analisadas anteriormente.

Tabela 7 – Esforço computacional da distribuição Beta

Complexidade Distribuição Beta	
Real	$O(1)$
Intervalar Moore	$O(1)$
Intervalar RDM	$O(1)$

A partir da Tabela 7 é possível observar de forma mais clara que em todas as implementações a complexidade se manteve a mesma, sendo esta constante. Com isso, conclui-se que utilizando a aritmética intervalar para computar a função densidade de probabilidade com distribuição Beta processada sob duas abordagens diferentes, no caso do presente trabalho, a complexidade se manteve igual a do algoritmo com entradas reais. Assim sendo, pode-se justificar o uso de técnicas intervalares para computar cálculos numéricos, obtendo maior exatidão nos resultados com controle automático de erros além de manter o esforço computacional em um nível aceitável, se mantendo igual, neste caso.

6 Conclusão

No trabalho com computação numérica, um dos fatores de maior importância é a exatidão de resultados dos cálculos. No estudo de variáveis aleatórias sobre o conjunto dos reais, $\mathbb{R}$, um dos problemas está no cálculo da função densidade de probabilidade, visto que seu valor numérico resultante no computador é dado por aproximação devido à representação em aritmética de ponto flutuante, portanto, afetado por erros de arredondamento ou truncamento. O que se procura então são resultados mais exatos com um menor erro contido neles. Para isso, a matemática intervalar surge com o objetivo principal de realizar um controle automático de erros em cálculos numéricos, retornando respostas com maior exatidão.

No presente trabalho, definiu-se de forma intervalar a função densidade de probabilidade da variável aleatória contínua com distribuição Beta, utilizando para isso o método de extensão intervalar onde as entradas reais foram substituídas para comportar valores de entrada intervalares.

Duas abordagens diferentes foram aplicadas como modo de calcular a função densidade de probabilidade com distribuição Beta utilizando aritmética intervalar: a aritmética intervalar tradicional de Moore (1966) e a aritmética multidimensional RDM, de Piegat e Landowski (2012). Desse modo, para que fosse possível computar a função densidade de probabilidade com distribuição Beta, realizou-se a implementação dessa função, sendo a codificação na linguagem de programação Python em conjunto com o pacote de extensão intervalar IntPy (BARRETO, 2016). Para que fosse possível comparar os resultados intervalares aos reais, a forma real dessa distribuição de probabilidades também foi implementada utilizando a linguagem Python.

Com o sistema de ponto flutuante F(10, 18, -17, 17), definido para uma precisão de dezoito casas decimais, verificou-se por meio das métricas de erro absoluto e erro relativo, estabelecidas na aritmética intervalar, que todos os resultados reais apresentados estão contidos nos valores intervalares apresentados a partir da computação da função densidade de probabilidade com distribuição Beta com entradas intervalares, tanto sob a aritmética de Moore quanto sob a aritmética intervalar multidimensional RDM. Calculou-se ainda a medida do diâmetros dos intervalos solução, por meio da qual foi possível verificar a qualidade dos valores de resultados intervalares obtidos.

Para ambas as abordagens da aritmética intervalar utilizadas neste trabalho, os intervalos solução para a função densidade de probabilidade com distribuição Beta apresentaram-se iguais, o que podia ser esperado visto que a função possui apenas uma variável. Dessa forma, ambas as soluções intervalares permanecem unidimensionais, justificando a igualdade dos resultados.

De forma complementar, a complexidade dos algoritmos implementados foi analisada. Apresentando-se para todos os algoritmos uma complexidade constante, uma vez que são realizadas apenas operações aritméticas simples no cálculo da função densidade com distribuição Beta. Para a resolução da integral definida necessária, é possível se obter uma complexidade também constante utilizando a linguagem Python, a qual define uma função própria para esse cálculo.

O objetivo deste trabalho foi mostrar a importância do uso da matemática intervalar na busca por exatidão em resultados numéricos para a função densidade com distribuição Beta, o que se fez por meio do uso de duas abordagens diferentes para a aritmética intervalar. A partir da implementação dessa função em linguagem de programação, com base na aritmética intervalar e na aritmética de exatidão máxima, tornou-se possível realizar uma análise de qualidade para os valores intervalares obtidos além da realização da análise de complexidade dos algoritmos desenvolvidos, justificando o seu uso.

Como pretensões futuras de trabalho, o estudo de outros métodos para a resolução de integrais, sobretudo no escopo das distribuições de probabilidade, visto que este cálculo é afetado por erros gerados pela aritmética de ponto flutuante. Além disso, aprofundar a pesquisa sobre algumas definições existentes na literatura acerca de integrais intervalares, podendo ser obtidos resultados melhores e ampliar as possíveis aplicações pra a aritmética intervalar envolvendo computação numérica.

A definição intervalar para a função densidade de probabilidade com distribuição Beta pode ser adicionada às demais distribuições existentes na literatura definidas desta forma. Junto a isso, definir de forma intervalar o cálculo das medidas de média, variância, desvio padrão e esperança para esta mesma distribuição.

Referências

BALBONI, M. et al. Critérios para análise e escolha de ambientes intervalares. *Revista Jr. de Iniciação Científica em Ciências Exatas e Engenharia, FURG, nº7*, Rio Grande - RS, 2014. Citado 3 vezes nas páginas 24, 25 e 30.

BARRETO, R. M. *IntPy 0.1.3*. 2016. Disponível em: <https://pypi.python.org/pypi/IntPy/0.1.3>. Citado 5 vezes nas páginas 11, 12, 26, 30 e 45.

BARRETO, R. M.; CAMPOS, M. A. Intpy: Um framework intervalar em python. *VIII Encontro Regional de Matemática Aplicada e Computacional-R3 (ERMAC 2008-R3)*, 2008. Citado na página 30.

CAPRANI, O.; MADSEN, K.; NIELSEN, H. *Introduction to interval analysis*. [S.l.: s.n.], 2002. Citado na página 22.

FERSON, S. et al. Absolute bounds on the mean of sum, product, max, and min: a probabilistic extension of interval arithmetic. 2002. Citado na página 16.

FINGER, A. F. *Extensão intervalar para as variáveis aleatórias com distribuições Uniforme, Normal, Gama, Exponencial e Pareto*. Dissertação (Mestrado) — Universidade Federal de Pelotas, 2014. Citado 2 vezes nas páginas 11 e 22.

FRANCO, N. B. *Cálculo numérico*. Rio de Janeiro, Brazil: Pearson, 2006. Citado na página 10.

KREINOVICH, V. et al. *Computational complexity and feasibility of data processing and interval computations*. [S.l.]: Springer Science & Business Media, 2013. v. 10. Citado 2 vezes nas páginas 16 e 41.

KULISCH, U. W.; MIRANKER, W. L. *Computer arithmetic in theory and practice*. London, Reino Unido: Academic press, 2014. Citado 3 vezes nas páginas 26, 30 e 31.

LANDOWSKI, M. Differences between moore and rdm interval arithmetic. In: *Intelligent Systems' 2014*. Springer, 2015. p. 331–340. Disponível em: <https://www.researchgate.net/publication/271966134_Differences_between_Moore_and_RDM_Interval_Arithmetic>. Citado 3 vezes nas páginas 19, 20 e 27.

LEISERSON, C. E. et al. Algoritmos: teoria e prática. *Editora Campus*, 2002. Citado 2 vezes nas páginas 41 e 42.

LIU, B. *Uncertainty theory*. [S.l.]: Springer, 2007. 205–234 p. Citado na página 15.

LIU, B. *Uncertainty theory*. Heidelberg, New York: Springer, 2010. 1–79 p. Citado 2 vezes nas páginas 15 e 19.

LORETO, A. B. *Análise da Complexidade Computacional de Problemas de Estatística Descritiva com Entradas Intervalares*. Tese (Doutorado) — doutorado em Ciências da Computação, Universidade Federal do Rio Grande do Sul. Porto Alegre, 2006. Citado 4 vezes nas páginas 10, 23, 24 e 41.

MENEZES, P. B. *Linguagens formais e autômatos*. [S.l.]: Sagra-Dcluzzato, 1998. Citado na página 41.

MESQUITA, M. P. *Matemática intervalar: princípios e ferramentas c-xsc*. Minas Gerais, Brasil: Online, 2002. Citado na página 31.

MOORE, R. E. *Interval analysis*. Englewood Clifis, New Jersey: Prentice-Hal, 1966. v. 4. Citado 4 vezes nas páginas 11, 12, 30 e 45.

MOORE, R. E. On computing the range of a rational function of n variables over a bounded region. *Computing*, Springer, v. 16, n. 1, p. 1–15, 1976. Citado na página 15.

MOORE, R. E.; KEARFOTT, R. B.; CLOUD, M. J. *Introduction to interval analysis*. Philadelphia: Siam, 2009. Citado 4 vezes nas páginas 10, 11, 15 e 16.

MOORE, R. E.; YANG, C. Interval analysis i. *Technical Document LMSD-285875, Lockheed Missiles and Space Division, Sunnyvale, CA, USA*, DTIC Document, 1959. Citado 2 vezes nas páginas 10 e 15.

NAGHETTINI, M.; PINTO, E. *Hidrologia Estatística*. Belo Horizonte, Brasil: CPRM, Serviço Geológico do Brasil, 2007. Citado na página 35.

PIEGAT, A.; LANDOWSKI, M. Is the conventional interval-arithmetic correct? *Journal of Theoretical and Applied Computer Science*, v. 6, n. 2, p. 27–44, 2012. Citado 6 vezes nas páginas 11, 12, 19, 28, 30 e 45.

PIEGAT, A.; LANDOWSKI, M. Correctness-checking of uncertain-equation solutions on example of the interval-modal method. *Modern Approaches in Fuzzy Sets, Intuitionistic Fuzzy Sets, Generalized Nets and Related Topics*, v. 1, p. 159–170, 2013. Citado na página 19.

PIEGAT, A.; LANDOWSKI, M. Two interpretations of multidimensional rdm interval arithmetic-multiplication and division. *International Journal of Fuzzy Systems*, v. 15, n. 4, p. 486–496, 2013. Citado 3 vezes nas páginas 19, 20 e 28.

PRESSMAN, R. S. Engenharia de software: uma abordagem profissional. 7ª edição. *Ed: McGraw Hill*, 2011. Citado na página 25.

PYTHON, . 2016. Disponível em: <https://docs.python.org/2/>. Citado 2 vezes nas páginas 12 e 30.

RATSCHEK, H.; ROKNE, J. *New computer methods for global optimization*. [S.l.]: Horwood Chichester, 1988. Citado 2 vezes nas páginas 10 e 32.

RITT, M.; BURIOL, L.; PRESTES, E. Algoritmos e complexidade notas de aula. 2009. Citado na página 41.

ROCHA, A. R.; CAMPOS, G. H. B. d. Avaliação da qualidade de software educacional. *Aberto, ano*, v. 12, p. 32–44, 1993. Citado na página 25.

RUGGIERO, M. A. G.; LOPES, V. L. d. R. *Cálculo numérico: aspectos teóricos e computacionais*. [S.l.]: Makron Books do Brasil, 1997. Citado na página 31.

SANTIAGO, R. H. N.; BEDREGAL, B. R. C.; ACIÓLY, B. M. Formal aspects of correctness and optimality of interval computations. *Formal Aspects of Computing*, Springer, v. 18, n. 2, p. 231–243, 2006. Citado na página 17.

SEBESTA, R. W.; MUKHERJEE, S. *Concepts of programming languages*. [S.l.]: Addison-Wesley Reading, 2002. v. 281. Citado na página 25.

SUNAGA, T. Theory of an interval algebra and its application to numerical analysis. *Japan Journal of Industrial and Applied Mathematics*, Springer, v. 26, n. 2, p. 125–143, 2009. Citado na página 38.

TAKAHASHI, A.; BEDREGAL, B. R. C.; LYRA, A. Uma versão intervalar do método de segmentação de imagens utilizando o k-means. *Trends in Applied and Computational Mathematics*, v. 6, n. 2, p. 315–324, 2005. Citado na página 11.

TAMAI, T. Social impact of information system failures. *Computer*, IEEE Computer Society, v. 42, n. 6, p. 58–65, 2009. Citado na página 10.

TOSCANI, L.; VELOSO, P. *Complexidade de Algoritmos: Série Livros Didáticos Informática UFRGS - Vol. 13*. [S.l.]: Bookman, 2009. ISBN 9788540701397. Citado 3 vezes nas páginas 11, 41 e 42.

VALENÇA, M. R. *Análise Numérica, Universidade Aberta*. [S.l.]: Lisboa, 1996. Citado na página 32.

VARJAO, F. *IntPy: Computaçao cientıfica auto validável em Python*. Dissertação (Mestrado) — Master's thesis, Universidade Federal de Pernambuco, Recife, 2011. Citado na página 26.

ZURAS, D. et al. Ieee standard for floating-point arithmetic. *IEEE Std 754-2008*, IEEE Computer Society, p. 1–70, 2008. Citado na página 10.

Anexos

ANEXO A – Implementações para a Função Densidade de Probabilidade com Distribuição Beta

Todas as implementações descritas no decorrer deste anexo foram realizadas utilizando o ambiente de desenvolvimento intervalar composto pela linguagem de programação Python e o pacote de extensão intervalar IntPy, para essa mesma linguagem.

A.1 Aritmética Real

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
from scipy import integrate
from intpy import *
import math
import time

def BetaReal(x, alpha, beta):
    begin_time = time.time()

    z = x**(alpha-1)*(1-x)**(beta-1)
    f = lambda x, alpha, beta: x**(alpha-1)*(1-x)**(beta-1)
    y, err = integrate.quad(f, 0, 1, args=(alpha,beta))
    resul = z/y

    end_time = time.time()

    print "Real=[" + '%.18f' %resul + "]"
    print "Tempo de processamento:", end_time - begin_time
```

A.2 Aritmética Intervalar de Moore

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
from scipy import integrate
from intpy import *
import math
import time

def BetaIntervalar(x, alpha, beta):
    begin_time = time.time()

    z = x**(alpha-1)*(1-x)**(beta-1)
    f = lambda x, alpha, beta: x**(alpha-1)*(1-x)**(beta-1)
    y, err = integrate.quad(f, 0, 1, args=(alpha,beta))

    alpha = IReal(alpha)
    beta = IReal(beta)
    x = IReal(x)
    z = IReal(z)
    y=IReal(y)

    resul = z/y
    end_time = time.time()

    print "Intervalar=[" + '%.18f' %(resul.inf) + ", " + '%.18f' %(resul.sup) + "]"
    print "Tempo de processamento:", end_time-begin_time
```

A.3 Aritmética Intervalar RDM

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
from scipy import integrate
from intpy import *
import math
from scipy import inf
import time

def somaRDM(a, b):
    x = (min(a.inf+b.inf, a.inf+b.sup, a.sup+b.inf, a.sup+b.sup))
    y = (max(a.inf+b.inf, a.inf+b.sup, a.sup+b.inf, a.sup+b.sup))
    s = IReal(x, y)
    return s

def subtRDM(a, b):
    x = (min(a.inf-b.inf, a.inf-b.sup, a.sup-b.inf, a.sup-b.sup))
    y = (max(a.inf-b.inf, a.inf-b.sup, a.sup-b.inf, a.sup-b.sup))
    sub = IReal(x, y)
    return sub

def multRDM(a, b):
    x = (min(a.inf*b.inf, a.inf*b.sup, a.sup*b.inf, a.sup*b.sup))
    y = (max(a.inf*b.inf, a.inf*b.sup, a.sup*b.inf, a.sup*b.sup))
    m = IReal(x, y)
    return m

def divRDM(a, b):
    x = (min(a.inf/b.inf, a.inf/b.sup, a.sup/b.inf, a.sup/b.sup))
    y = (max(a.inf/b.inf, a.inf/b.sup, a.sup/b.inf, a.sup/b.sup))
    d = IReal(x, y)
    return d

def powI(num, exp):
    pot = IReal(num.inf**exp.inf, num.sup**exp.sup)
    return pot

def BetaIntervalar(x, alpha, beta):
    begin_time = time.time()

    f = lambda x, alpha, beta: x**(alpha-1)*(1-x)**(beta-1)
    y, err = integrate.quad(f, 0, 1, args=(alpha,beta))

    x = IReal(x)
    alpha = IReal(alpha)
```

```
beta = IReal(beta)
um = IReal(1)
y=IReal(y)

z = multRDM(powI(x,(subtRDM(alpha,um))), powI((subtRDM(um,x)), \\
    (subtRDM(beta,um))))

resul = (z/y)
end_time = time.time()

print "\nIntervalar=[" + '%.18f' %(resul.inf) + ", " + '%.18f' %(resul.sup) + "]"
print "Tempo_de_processamento:", end_time-begin_time
```