



Definição intervalar e análise de qualidade da função densidade de probabilidade da variável aleatória contínua com distribuição Beta

Interval definition and quality analysis of the probability density function of the continuous random variable with Beta distribution

D. Maraschin^{1*}; A. Finger²; A. Loreto³

¹Programa de Pós-Graduação em Computação/CDTec, UFPel, 96010-610, Pelotas-RS, Brasil

²Unipampa, 97546-550, Alegrete-RS, Brasil

³UFSM, 96506-001, Cachoeira do Sul-RS, Brasil

*dmaraschin@inf.ufpel.edu.br

(Recebido em 13 de abril de 2017; aceito em 22 de maio de 2017)

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 sem o auxílio da 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 com distribuições se faz necessário o uso da integração numérica. Sendo o resultado obtido por aproximação, este é afetado por erros. Neste contexto, este trabalho possui o objetivo de definir a função densidade de probabilidade com distribuição Beta de forma intervalar utilizando o método de extensão intervalar para, posteriormente, implementar esta função usando linguagem de programação. A fim de verificar a qualidade dos resultados obtidos, uma análise de erro é realizada através do cálculo das métricas de erro relativo, erro absoluto e diâmetro dos intervalos. O objetivo é verificar e justificar que, ao usar a aritmética intervalar para calcular a função densidade de probabilidade com distribuição Beta, é possível ter um controle automático de erros com limites confiáveis. Somando-se a isso, pretende-se complementar a literatura, uma vez que já existem distribuições definidas em termos de intervalos. Através da análise de qualidade dos intervalos calculados, obteve-se resultados satisfatórios, com diâmetro pequeno e com erros absoluto e relativo que garantem a qualidade do intervalo.

Palavras-chave: aritmética intervalar, análise de erro, distribuição de probabilidade

When you work with numerical calculations in computing environments, we operate on floating point numbers. In this way, the result is only an approximation of a real value and errors generated by rounding or truncation can lead to incorrect results, cannot be affirmed the accuracy of the answers without the aid of an error analysis. When using intervals for representation of real values, it becomes possible to control the spread of these errors, because interval results carry with them the security of your quality. To get the numeric value of the probability density functions with distributions it is necessary the use of numerical integration. Being the result obtained by approximation, this one is affected by errors. In this context, this work has the objective of defining the probability density function with Beta distribution in an interval form using the interval extension method to later implement this function using programming language. In order to check the quality of the results obtained, an error analysis it will be performed by calculating metrics of relative error, absolute error and diameter of the intervals. The goal is to verify and justify that, when using the interval arithmetic to calculate the probability density function with Beta distribution, it is possible to have an automatic control of errors with reliable limits. Adding to this, it is intended to complement the literature, once it already exist distributions defined in terms of intervals. Through the analysis of quality of the calculated intervals, obtained satisfactory results, with small diameter and with absolute and relative errors that guarantee the quality of the interval.

Keywords: interval arithmetic, error analysis, distribution of probability

1. INTRODUÇÃO

Uma solução numérica requer muitas vezes uma sequência infinita de operações matemáticas exatas para ser calculada [15]. 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 [6]. Os sistemas de computador atuais são padronizados para operar com formatos e métodos da aritmética de ponto flutuante, conforme o padrão IEEE [22], 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 [10].

A análise de erros numéricos é um processo importante para verificar se os resultados numéricos são confiáveis e precisos em processos computacionais. A relevância de tal processo se reflete quando falhas envolvendo sistemas de computador ocorrem. Pois 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 [20].

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 [10]. Justificando-se, assim, o emprego de técnicas intervalares para contornar erros gerados pela aritmética de ponto flutuante.

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 [14]. 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 da aproximação obtida [17].

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 [14, 19].

No estudo das variáveis aleatórias contínuas aplicadas sobre valores 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, como citado, pode ser afetado por erros de arredondamento ou truncamento [5].

Em vista disso, alguns trabalhos já foram desenvolvidos nesse contexto utilizando intervalos para obter maior exatidão nos resultados e contornar o problema do erro computacional. Finger (2014) [5] desenvolveu, em sua dissertação, as definições das funções densidade de probabilidade das variáveis aleatórias contínuas com distribuições Gama e Pareto com entradas intervalares utilizando o método de extensão intervalar. São apresentadas duas formas de obtenção dos intervalos para as funções densidade citadas, uma utilizando o método de Simpson Intervalar e outra a partir da primitiva da função. Além disso, acrescenta a análise de complexidade dos algoritmos desenvolvidos para computar as funções densidade com distribuições Uniforme, Exponencial, Normal, Pareto e Gama a partir de resultados obtidos com as implementações utilizando o pacote IntPy. Finger (2014) [5] afirma que as melhores soluções foram obtidas através da aplicação da primitiva da função em ambas as formas real e intervalar. E, além disso, concluiu que a utilização de intervalos nessas distribuições mantém o mesmo esforço computacional que a forma real.

O trabalho desenvolvido por Loreto (2006) [10] traz como tema a complexidade dos problemas de cálculo de indicadores estatísticos com entradas intervalares. Para isso, foram definidos alguns

indicadores estatísticos através do método de extensão intervalar, tais como as separatrizes: coeficiente de variação intervalar, mediana intervalar, quartil, decil e percentil intervalares. Adicionalmente, reuniu ainda outros indicadores estatísticos já definidos na forma intervalar, tais como: média intervalar, moda intervalar, variância intervalar, desvio padrão intervalar, covariância intervalar e coeficiente de correlação intervalar, obtendo resultados que justificam o uso de intervalos.

Varjão (2011) [21] desenvolveu funções matemáticas específicas para a linguagem de programação Python, tais como: potência, raiz quadrada, logaritmo e funções trigonométricas. Estendendo, assim, as operações possíveis ao utilizar-se o pacote IntPy e possibilitando a programação utilizando o tipo intervalo.

Dentre o que já foi realizado em outros trabalhos, tais como os citados, ainda restam distribuições de probabilidade para as quais não foram criadas definições intervalares bem como trabalhadas utilizando intervalos. Nesta perspectiva, o presente trabalho vem como uma proposta para complementar a literatura nesse sentido.

O objetivo deste trabalho é definir a função densidade de probabilidade da variável aleatória contínua com distribuição Beta para entradas intervalares. Para tanto, utiliza-se do método de extensão intervalar. Além disso, desenvolve-se a implementação da função densidade de probabilidade com distribuição Beta utilizando o pacote para programação intervalar da linguagem de programação Python, denominado IntPy [2], o qual dispõe o tipo intervalo e operações sobre o tipo. Após a implementação e obtenção dos resultados, uma análise de qualidade dos intervalos obtidos permite comparar os resultados com a forma real da distribuição, apurando sua exatidão.

2. FUNDAMENTAÇÃO TEÓRICA

Nesta seção serão apresentados conceitos relacionados à matemática intervalar, juntamente com suas definições, os quais serviram como base teórica ao desenvolvimento deste trabalho.

A teoria da matemática intervalar é descrita seguindo os conceitos desenvolvidos por Moore [13, 14, 15]. Além disso, são apresentadas algumas definições que serviram de base para compreender os conceitos básicos acerca da aritmética intervalar. Apresenta-se, nesse mesmo contexto, o método de extensão intervalar, o qual é utilizado para definir de forma intervalar a função densidade de probabilidade da variável aleatória contínua com distribuição Beta.

Acrescentando-se a isso, descreve-se a computação de cálculos numéricos em ambientes computacionais com suporte intervalar. Tipo de ambiente no qual a codificação e testes realizados neste trabalho foram executados tanto para a forma real quanto intervalar da função densidade de probabilidade Beta, distribuição para a qual fora desenvolvida a modelagem utilizando linguagem de programação.

2.1 Fundamentos da Matemática Intervalar

O conceito de matemática intervalar desenvolvido por Moore [13, 14, 15], baseia-se 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} e $\bar{x} \in \mathbb{R}$. Logo, a imagem da função f é representada na Eq. (1).

$$f(\mathbf{x}) = \{y \mid y = f(x), \underline{x} \leq x \leq \bar{x}\} \quad (1)$$

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

Definição 1 (Função Intervalar): Seja $F: x \rightarrow y$ uma função. Se $x = \text{Dom}(f) \subseteq \mathbb{R}$ e $y = \text{Cod}(f) \subseteq \mathbb{R}$, 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 $f(x)$. Uma forma de apurar o erro obtido ao se calcular $f(x)$ a partir do intervalo x é através do diâmetro do intervalo.

O cálculo de expressões utilizando aritmética intervalar consiste basicamente na extensão das operações aritméticas junto a um conjunto de operações “standard”, ou padrão. Na computação intervalar, segundo Kreinovich et al. (2013) [7], pode-se obter o intervalo solução, $y = [\underline{y}, \bar{y}] = f(x_1, \dots, x_n)$, 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.

O método da extensão intervalar, segundo Ferson et al. (2002) [4], 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 x e y para x e y , pode-se então calcular a imagem exata $f(x, y)$ por meio da aritmética intervalar definida por Moore et al. (1966) [14]. 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 et al. (2006) [18], a extensão intervalar proposta por Moore pode ser definida da seguinte forma:

Definição 2: A função $F: I(\mathbb{R}) \rightarrow I(\mathbb{R})$, de variável intervalar $X \in I(\mathbb{R})$, é uma extensão intervalar de uma função $f: \mathbb{R} \rightarrow \mathbb{R}$, se para todo $x \in \mathbb{R}$, $F([x, x]) = [f(x), f(x)]$, $x \in X$.

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

Definição 3: Uma função intervalar F está correta em relação a f , real, se satisfaz a propriedade: $x \in [\underline{x}, \bar{x}] \Rightarrow f(x) \in F([x, x])$.

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 a f , ou a representa, se este pode ser interpretado em uma função intervalar parcial F que representa f [18].

2.2 Computação Matemática em Ambientes Intervalares

Na maioria das linguagens de programação modernas de alto nível as operações aritméticas são realizadas através de símbolos, de forma semelhante à encontrada, normalmente, na notação usual científica. Isso permite certa facilidade ao programador na hora de escrever expressões, fórmulas ou funções.

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 (2014) [9]. 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 [8, 9] propuseram que a implementação da aritmética intervalar seja realizada utilizando a chamada aritmética de exatidão máxima, o que significa a busca para que resultados numéricos ou sejam um número de ponto flutuante ou estejam entre dois números em ponto flutuante consecutivos [1].

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 Scientific Computations*) [11].

Aplicada à matemática intervalar, 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 [9], por exemplo, uma operação de adição entre dois intervalos, \mathbf{x} e \mathbf{y} , se dá da seguinte forma: $\mathbf{x} + \mathbf{y} = [\nabla(\underline{x} + \underline{y}), \Delta(\bar{x} + \bar{y})]$, onde ∇ constitui o arredondamento direcionado para baixo e Δ se refere ao arredondamento direcionado para cima.

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].

Existem diversos ambientes computacionais para o desenvolvimento de algoritmos para cálculos numéricos utilizando matemática intervalar na sua estrutura, no entanto, o presente trabalho fará uso do pacote IntPy [2], adicionado na linguagem Python [3], para implementar a função densidade de probabilidade da variável aleatória contínua com distribuição Beta. A escolha da linguagem se deu a partir de comparações realizadas entre alguns ambientes de programação intervalar realizado por Balboni et al. (2014) [1], onde foi apontado que o IntPy oferece melhor precisão sobre os demais ambientes comparados.

O IntPy, como citado, é um pacote para programação intervalar na linguagem de programação Python desenvolvido por Barreto (2016) [2]. Tanto o pacote IntPy como subpacotes relacionados que o compõe são implementados na linguagem Python como software livre (GPL - *General Public License*) [2].

3. FUNÇÃO DENSIDADE DE PROBABILIDADE DA VARIÁVEL ALEATÓRIA CONTÍNUA COM DISTRIBUIÇÃO BETA NAS FORMAS REAL E INTERVALAR

Com base nas definições encontradas em Naghettini et al. (2007) [16] e Milone (2004) [12], esta seção apresenta a definição da função densidade de probabilidade da variável aleatória contínua com distribuição Beta nas formas real e intervalar.

3.1 Distribuição Beta na Forma Real

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 Eq.(2).

$$P(a < X < b) = \int_a^b f(x)dx \quad (2)$$

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 $f_X : \mathbb{R} \rightarrow (0, +\infty]$, que satisfaz as propriedades:

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

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 $[0, 1]$.

Nesse caso, para $0 \leq x \leq 1$, $\alpha > 0$, $\beta > 0$, a função densidade de probabilidade Beta [16] é expressa pela Eq.(3):

$$f_X(x) = \frac{1}{B(\alpha, \beta)} \int_0^1 x^{\alpha-1} (1-x)^{\beta-1} dx \quad (3)$$

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

$$B(\alpha, \beta) = \frac{\Gamma(\alpha) \Gamma(\beta)}{\Gamma(\alpha + \beta)} \quad (4)$$

Para a Eq.(4), 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 Eq.(5), onde y representa um parâmetro de entrada dessa função.

$$\Gamma(n) = \int_0^{\infty} y^{n-1} e^{-y} dy = (n-1)! \quad (5)$$

De modo geral, para valores iguais de ambos os parâmetros, a função densidade de probabilidade Beta é simétrica; contrariamente, a distribuição Beta é assimétrica. Se ambos os parâmetros, são superiores a 1, $\alpha > 1$ e $\beta > 1$, a distribuição Beta é unimodal. A Fig.(1) apresenta um gráfico descrevendo o comportamento da função densidade Beta para diferentes valores de seus parâmetros.

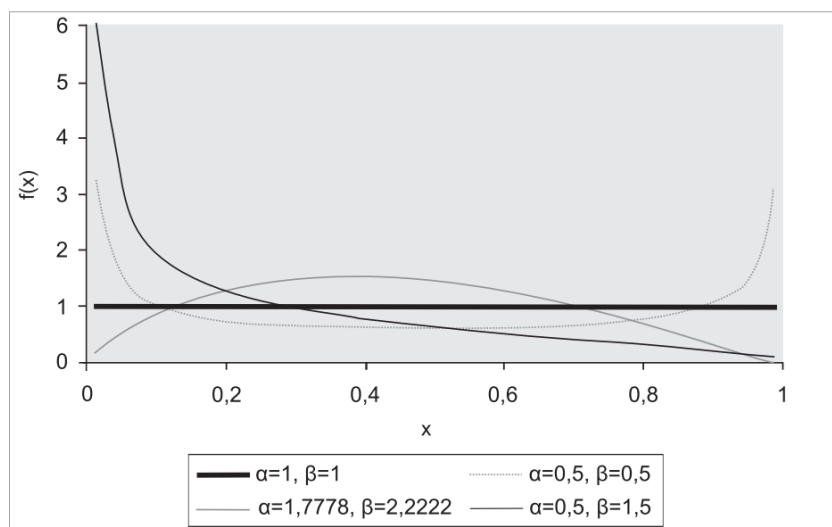


Figura 1: Função densidade de probabilidade com distribuição Beta

Por produzir distribuições probabilísticas sobre intervalos, a distribuição de probabilidades Beta é adequada ao estudo de variáveis cujos valores dos parâmetros sabidamente se restringem a

intervalos. Além disso, a variedade de formas dessa distribuição faz com que esta seja de múltiplas utilidades para a modelagem de variáveis com limites à direita e à esquerda. Alguns exemplos de aplicações para a distribuição de probabilidades com distribuição Beta são: modelos de probabilidade pluvial, modelagem de probabilidade de direção e potência do vento, probabilidade para aceleração de projetos e simulações utilizando técnicas de PERT-CPM, na engenharia química para estimar concentrações, na física, dentre outros.

3.2 Distribuição Beta na Forma Intervalar

Sendo a função densidade de probabilidade com distribuição Beta um modelo probabilístico para uma variável aleatória contínua X , cujos valores possíveis são limitados superior e inferiormente, a variável X é definida no intervalo $[0, 1]$ [16].

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 Eq.(6).

$$f_{B_{int}}(x, \alpha, \beta) = \frac{1}{B(\alpha, \beta)} \int_0^a x^{\alpha-1} (1-x)^{\beta-1} dx \quad (6)$$

Seguindo tal método para se criar a definição intervalar da função densidade da distribuição Beta, fez-se necessário substituir todas as entradas de x da Eq.(3), a qual antes comportava valores reais, \mathbb{R} , para entradas intervalares.

A partir da definição intervalar, foi realizada a implementação da função densidade de probabilidade da variável aleatória contínua com distribuição Beta na linguagem de programação Python, utilizando o pacote de extensão intervalar IntPy, com o objetivo de executar testes e verificar se a aritmética com intervalos retorna resultados satisfatórios. Para isso, foi preciso codificar a distribuição Beta com entradas reais, expressa pela Eq.(3), e intervalares, Eq.(6), para que fosse possível comparar os resultados obtidos.

4. RESULTADOS E ANÁLISE DE QUALIDADE DO INTERVALO

Para computar a função densidade de probabilidade da variável aleatória contínua com distribuição Beta, tanto na forma real quanto intervalar, utiliza-se a fórmula da densidade de probabilidade desta distribuição para realizar seus 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, Memória RAM de 4GB DDR3 1600MHz, armazenamento HD Sata 1TB modelo ST1000LM024 HN-M101MBB, sistema operacional Linux Ubuntu 16.04 LTS.

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

Como forma de obter os resultados numéricos, foram utilizados quatro conjuntos de dados de entrada como parâmetros para as funções real e intervalar da distribuição Beta, tais dados se referem a valores de exemplo 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 Beta. A Tab.(1) apresenta os resultados obtidos com a implementação no IntPy, das formas real e intervalar da distribuição de probabilidade. De maneira complementar, também é apresentado o tempo de execução, em segundos, de cada conjunto de valores computado:

- **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 $a = 1$.

Tabela 1: Aplicação dos parâmetros à função densidade nas formas real e intervalar

Exemplo	Resultado	
	Real	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

Analisando a Tab.(1), é possível perceber que o valor real está contido no intervalo solução em todos os exemplos aplicados.

Para verificar a qualidade dos resultados, realiza-se a análise de erros sobre os dados 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 as possíveis formas de apurar a qualidade do resultado obtido, o diâmetro do intervalo é uma métrica para mensurar a qualidade de aproximação. As fórmulas para tais métricas são descritas a seguir.

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

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} [5].

A Tabela(2) apresenta os valores obtidos para a estimativa de erro absoluto e erro relativo em relação aos intervalos resultantes da aplicação dos dados dos exemplos da distribuição Beta. A seguir, na Tab.(3), apresenta-se a medida do diâmetro dos intervalos obtidos a partir do cálculo computado.

Tabela 2: Qualidade dos intervalos para a variável aleatória com distribuição Beta

Exemplo	Erros	
	Absoluto	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	$0.0 < 2.22044604925 \times 10^{-16}$	$0.0 \leq 1.07854707001 \times 10^{-16}$

Tabela 3: Diâmetro dos intervalos para a variável aleatória Beta

Exemplo	Diâmetro do intervalo
Exemplo 1	0.0000000000000000222
Exemplo 2	0.0000000000000000222
Exemplo 3	0.0000000000000000111
Exemplo 4	0.0000000000000000444

No erro absoluto, verifica-se que a diferença entre o valor exato e o ponto médio do intervalo ocorre, em relação aos exemplos, na 16^a casa decimal. Além de que, em todos os resultados, constata-se que a condição $|x - m(\mathbf{x})|$ 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 15^a 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.

4 CONCLUSÃO

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

Neste trabalho, criou-se a definição intervalar para a função densidade de probabilidade da variável aleatória contínua com distribuição Beta seguindo o método da extensão intervalar. Para isso, fez-se necessário substituir todas as entradas reais da função para entradas intervalares, constituindo uma nova função que comporta não só parâmetros de entrada intervalares, mas que também fornece uma solução intervalar.

A implementação da função com entradas reais foi desempenhada na linguagem de programação Python. Para a forma intervalar da distribuição Beta, também se utilizou dessa mesma linguagem juntamente com pacote de extensão IntPy, o qual fornece suporte para programação com intervalos. A escolha desse ambiente de desenvolvimento se deu pela análise comparativa a outras linguagens, onde apontou-se que o pacote IntPy e a linguagem Python apresentaram boa precisão nos resultados [1].

Sob o sistema de ponto flutuante F(10, 18, -17, 17), com precisão definida em dezoito casas decimais, os valores reais obtidos na computação da função densidade de probabilidade estão contidos nos intervalos solução resultantes da função densidade intervalar. Além disso, verificou-se, por meio das métricas de erros (relativo e absoluto) e diâmetro, a qualidade dos intervalos.

O objetivo deste trabalho foi mostrar a importância de se utilizar a matemática intervalar para se alcançar soluções exatas e com o menor erro possível contido nestas. Justificou-se, por meio do cálculo da função densidade de probabilidade com distribuição Beta que é possível alcançar a exatidão nos resultados, contornando o problema do erro computacional, por meio do emprego de técnicas intervalares em utilização de um ambiente computacional intervalar e na aritmética de exatidão máxima.

6. REFERÊNCIAS BIBLIOGRÁFICAS

1. Balboni M, Tortelli L, Lorini M, Furlan V, Finger A, Loreto A. Critérios para análise e escolha de ambientes intervalares. *Revista Jr. de Iniciação Científica em Ciências Exatas e Engenharia*. 2014 Fev;1(7):2236-93.
2. Barreto RM. Intpy 0.1.3. URL <https://pypi.python.org/pypi/IntPy/0.1.3>; 2016.
3. Documentation Python. URL <https://docs.python.org/2/>; 2016.
4. Ferson S, Ginzburg L, Kreinovich V, Lopez J. Absolute bounds on the mean of sum, product, max, and min: a probabilistic extension of interval arithmetic. *Applied Mathematical Sciences*. 2002 Mai;1(9):395-440.
5. Finger AF. Extensão intervalar para as variáveis aleatórias com distribuições Uniforme, Normal, Gama, Exponencial e Pareto. [dissertation]. Pelotas (RS): Universidade Federal de Pelotas; 2014. 68 p.
6. Franco, NB. Cálculo numérico. Rio de Janeiro: Pearson; 2006. 494 p.
7. Kreinovich V, Lakeyev AV, Rohn J, Kahl P. Computational complexity and feasibility of data processing and interval computations. *Springer Science & Business Media*; 2013, doi: 10.1007/978-1-4757-2793-7.
8. Kulisch UW. Complete interval arithmetic and its implementation on the computer. *Numerical Validation in Current Hardware Architectures*. 2009;5492:7-26, doi: 10.1007/978-3-642-01591-5_2.
9. Kulisch UW, Miranker WL. *Computer arithmetic in theory and practice*. London: Academic press; 2014.
10. Loreto AB. Análise da Complexidade Computacional de Problemas de Estatística Descritiva com Entradas Intervalares. [tese]. Porto Alegre (RS): Universidade Federal do Rio Grande do Sul; 2006. 93 p.
11. Mesquita MP. Matemática intervalar: princípios e ferramentas c-xsc. [dissertation]. Lavras (MG). Universidade Federal de Lavras; 2002.
12. Milone G. *Estatística: geral e aplicada*. São Paulo: Pioneira Thomson Learning; 2004.
13. Moore RE. On computing the range of a rational function of n variables over a bounded region. *Computing*. 1976 Mar;16(1):1-15 p, doi:10.1007/BF02241975.
14. Moore RE, Kearfott RB, Cloud MJ. *Introduction to interval analysis*. USA: Prentice-Hall; 1966.
15. Moore RE, Yang C. *Interval analysis i*. Sunnyvale; CA; USA. 1959. LMSD-285875
16. Naghettini M, Pinto E. *Hidrologia Estatística*. Belo Horizonte: CPRM; 2007.
17. Ratschek H, Rokne J. *New computer methods for global optimization*. Horwood Chichester; 1988. 236 p.
18. Santiago RHN, Bedregal BRC, Acióly BM. Formal aspects of correctness and optimality of interval computations. *Formal Aspects of Computing*. 2006;18(2):231-243, doi:10.1007/s00165-006-0089-x.
19. Takahashi A, Bedregal BRC, Lyra A. Uma versão intervalar do método de segmentação de imagens utilizando o k-means. *Trends in Applied and Computational Mathematics*. 2005;6(2):315-24.
20. Tamai T. Social impact of information system failures. *Computer*. 2009 Jun;42(6):58-65.
21. Varjao F. IntPy: Computação científica auto validável em Python. [dissertation]. Recife: Universidade Federal de Pernambuco; 2011.
22. Zuras D, Cowlshaw M, Aiken A, Applegate M, Bailey D, Bass S, Bhandarkar D, Bhat M, Bindel D, Boldo S. et al. Ieee standard for floating-point arithmetic. *IEEE Std 754-2008*; 2008. 70 p.