

UNIVERSIDADE FEDERAL DO PAMPA

Paloma Ribeiro dos Santos

**Definição intervalar do método de
integração dos Trapézios**

Alegrete
2017

Paloma Ribeiro dos Santos

Definição intervalar do método de integração dos Trapézios

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

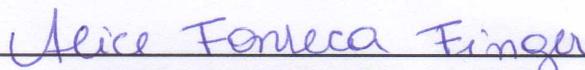
Alegrete
2017

Paloma Ribeiro dos Santos

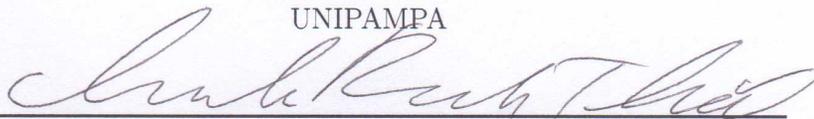
Definição intervalar do método de integração dos Trapézios

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 29 de novembro de 2017
Banca examinadora:



Profa. Ma. Alice Fonseca Finger
Orientadora
UNIPAMPA



Prof. Dr. Marcelo Resende Thielo
UNIPAMPA



Prof. Me. Eliezer Soares Flores
UNIPAMPA

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. Uma das alternativas de contornar os erros causados pela máquina é a utilização da aritmética intervalar, que torna possível obter resultados intervalares com maior exatidão e com menor erro contido. Métodos numéricos utilizam uma sequência finita de operações aritméticas básicas. Estes algoritmos estão sujeitos a erros que são propagados devido ao grande número de cálculos aritméticos realizados no computador. Quando não é possível encontrar a primitiva da função na forma analítica, não podemos usar outros métodos que não os numéricos. Sendo a integração numérica obtida por aproximação, o resultado é afetado por erros. Na literatura existem métodos de integração intervalares, como a Integral de Moore, Integral de Bedregal, Integral de Rall e Simpson Intervalar, porém ainda não foi definida a forma intervalar do método dos Trapézios. Nesse contexto, o presente trabalho possui o objetivo de definir de forma intervalar o método de integração dos Trapézios, utilizando o método de extensão intervalar, para que seja possível realizar a implementação do método em um ambiente de programação com suporte ao tipo intervalo. A fim de verificar os resultados obtidos, foi realizada uma análise numérica através do cálculo do diâmetro do intervalo, e dos erros relativo e absoluto. Além disso, foi realizado um comparativo com o método de Simpson intervalar. Ao utilizar o método dos Trapézios Intervalar, foi possível obter resultados intervalares com erros aceitáveis e manter um tempo de processamento menor que ao utilizar o método de Simpson Intervalar.

Palavras-chave: Aritmética intervalar. Métodos numéricos. Integração numérica. Integração intervalar. Método dos Trapézios.

ABSTRACT

When working with numerical calculation in a computational system, we operate on floating-point numbers. This way, the result is only an approximation and errors generated by arredondations or truncations can lead to incorrect results. One of the alternatives to bypass the errors caused by the machine is the utilization of intervals arithmetic which makes possible to obtain intervals results with greater accuracy and with a lower contained error. Numerical methods look for develop algorithms using a finite sequence of basic arithmetic operations. When it is not possible to find the primitive of a function in its analitic form, we can not use other method than the numerical method. Being the numerical integration obtained through aproximation the result is affected by errors. In the literature there are interval integration methods, such as the Moore Integral, Bedregal Integral, Rall Integral, and Interval Simpson Method, but still the interval form of Trapezoidal method is not defined. In this context, the present work has as objective to define the interval form of the Trapezoidal method, using the interval extension method, to, posteriorly, implementate the method in a programation environment that supports the interval type. In order to verify the results obtained, a numerical analysis was performed by calculating the diameter of the interval, and the relative and absolute errors. In addition, a comparison was made with the Simpson interval method. When using the interval trapezoidal method, it was possible to obtain interval results with acceptable errors and to mantain a lower processing time than that obtained when using the interval Simpson method.

Key-words: Interval arithmetic. Numerical methods. Numerical integration. Interval integration. Trapezoidal method.

LISTA DE TABELAS

Tabela 1 – Número de iteração e tempo de CPU.	23
Tabela 2 – Comparação entre as regras de Trapézio e Simpson.	24
Tabela 3 – Esforço computacional das distribuições.	25
Tabela 4 – Erro Absoluto.	26
Tabela 5 – Erro Relativo.	26
Tabela 6 – Regra dos Trapézios com verificação	27
Tabela 7 – Regra 1/3 de Simpson com verificação	27
Tabela 8 – Valores reais e intervalares	28
Tabela 9 – Resultados obtidos no exemplo f1.	37
Tabela 10 – Resultados obtidos no exemplo f2.	37
Tabela 11 – Resultados obtidos no exemplo f3.	38
Tabela 12 – Resultados obtidos no exemplo f4.	38
Tabela 13 – Resultados obtidos no exemplo f5.	39
Tabela 14 – Resultados obtidos no exemplo f6.	39
Tabela 15 – Resultados obtidos no exemplo f7.	40
Tabela 16 – Resultados obtidos no exemplo f8.	40
Tabela 17 – Resultados obtidos no exemplo f9.	41
Tabela 18 – Resultados obtidos no exemplo f10.	41
Tabela 19 – Resultados obtidos no exemplo f11.	42
Tabela 20 – Resultados obtidos no exemplo f12.	42
Tabela 21 – Resultados obtidos no exemplo f13.	43
Tabela 22 – Diâmetro, erros absoluto e relativo da função f1.	43
Tabela 23 – Diâmetro, erros absoluto e relativo da função f2.	44
Tabela 24 – Diâmetro, erros absoluto e relativo da função f3.	45
Tabela 25 – Diâmetro, erros absoluto e relativo da função f4.	45
Tabela 26 – Diâmetro, erros absoluto e relativo da função f5.	46
Tabela 27 – Diâmetro, erros absoluto e relativo da função f6.	46
Tabela 28 – Diâmetro, erros absoluto e relativo da função f7.	47
Tabela 29 – Diâmetro, erros absoluto e relativo da função f8.	47
Tabela 30 – Diâmetro, erros absoluto e relativo da função f9.	48
Tabela 31 – Diâmetro, erros absoluto e relativo da função f10.	48
Tabela 32 – Diâmetro, erros absoluto e relativo da função f11.	49
Tabela 33 – Diâmetro, erros absoluto e relativo da função f12.	49
Tabela 34 – Diâmetro, erros absoluto e relativo da função f13.	50
Tabela 35 – Tempo de execução em segundos para a função f1.	51
Tabela 36 – Tempo de execução em segundos para a função f2	51
Tabela 37 – Tempo de execução em segundos para a função f3.	51
Tabela 38 – Tempo de execução em segundos para a função f4.	51

Tabela 39 – Tempo de execução em segundos para a função f5.	52
Tabela 40 – Tempo de execução em segundos para a função f6.	52
Tabela 41 – Tempo de execução em segundos para a função f7.	52
Tabela 42 – Tempo de execução em segundos para a função f8.	52
Tabela 43 – Tempo de execução em segundos para a função f9.	53
Tabela 44 – Tempo de execução em segundos para a função f10.	53
Tabela 45 – Tempo de execução em segundos para a função f11.	53
Tabela 46 – Tempo de execução em segundos para a função f12.	53
Tabela 47 – Tempo de execução em segundos para a função f13.	54
Tabela 48 – Classificação por tipo de função	54

SUMÁRIO

1	INTRODUÇÃO	13
1.1	Objetivos	14
1.2	Organização deste trabalho	14
2	FUNDAMENTAÇÃO TEÓRICA	17
2.1	Matemática Intervalar	17
2.1.1	Operações Aritméticas Intervalares Básicas	19
2.2	IntPy: pacote de aritmética intervalar	20
3	TRABALHOS RELACIONADOS	23
3.1	Avaliação de Esquemas Numéricos para Problemas Difusivos 2D Resolvidos com Volumes Finitos	23
3.2	Análise Numérica para a Solução de Integrais não Elementares	24
3.3	Extensão intervalar para as variáveis aleatórias com distribui- ções Uniforme, Normal, Gama, Exponencial e Pareto	24
3.4	Comparando Diferentes Métodos de Integração Intervalar . .	25
3.5	Integração Intervalar Revisitada	27
3.6	Critérios para Análise e Escolha de Ambientes Intervalares .	28
4	MÉTODOS DE INTEGRAÇÃO NUMÉRICA E DEFINIÇÃO DO MÉTODO DOS TRAPÉZIOS INTERVALAR	31
4.1	Método dos Trapézios	31
4.2	Método de Simpson	33
5	ANÁLISE DA QUALIDADE DOS INTERVALOS ENCAP- SULADORES	35
5.1	Análise numérica	36
6	CONSIDERAÇÕES FINAIS	57
6.1	Trabalhos futuros	58
	REFERÊNCIAS	59
	APÊNDICES	61
	APÊNDICE A – IMPLEMENTAÇÕES DOS MÉTODOS . .	63
A.1	Trapézio Real	63
A.2	Trapézio Intervalar	63
A.3	Simpson Real	64

A.4	Simpson Intervalar	65
-----	------------------------------	----

1 INTRODUÇÃO

Na aritmética de ponto flutuante, números reais são aproximados por um subconjunto chamado números de máquina. Devido a essa representação são gerados alguns tipos de erros. Tais erros ocorrem quando: um valor real de entrada é aproximado para um número de máquina; resultados intermediários gerados na execução de cada operação e que vão se acumulando; e ainda quando os dados de entrada são incertos. Pode-se encontrar também erros devido à conversões realizadas pela máquina. Os valores de entrada normalmente são fornecidos na base decimal, necessitando que sejam convertidos para a base binária, a qual pode ser computada pelo sistema, e depois, convertidos novamente para gerar uma saída ao usuário (RUGGIERO; LOPES, 1996).

A análise numérica de erros é um processo importante para se obter resultados confiáveis e precisos em processos computacionais. O emprego de técnicas intervalares pode ser aplicado para contornar esses erros, tornando possível obter resultados mais exatos e com menor erro contido. Através da análise intervalar, tem-se um controle automático de erros com limites confiáveis.

Na matemática intervalar, em vez de se aproximar um valor real x para um número de máquina, ele é aproximado por um intervalo \mathbf{x} , que possui como limite inferior e superior números de máquina de forma que o intervalo contenha x . O tamanho deste intervalo pode ser usado como medida para avaliar a qualidade de aproximação, quanto menor o tamanho do intervalo melhor é sua qualidade (RATSCHEK; ROKNE, 1988).

O cálculo numérico de integrais aparece constantemente na resolução de diversos problemas que não podem ser resolvidos analiticamente, ou requerem muito esforço humano, e seu procedimento será facilitado por métodos computacionais. Quando não é possível encontrar primitivas das funções na forma analítica, não podemos usar outros métodos que não os numéricos. As aplicações imediatas de integrais definidas residem em vários domínios da engenharia, física, teoria da probabilidade, entre outras (NÓBREGA, 2012).

Os métodos numéricos são algoritmos que utilizam uma sequência finita de operações aritméticas básicas. Estes algoritmos estão sujeitos a erros de arredondamento e truncamento, que são propagados devido ao grande número de cálculos intermediários realizados no computador (MONTEIRO, 2012). Dentre os métodos numéricos de integração, a simplicidade da fórmula do método dos Trapézios é um dos fatores que o tornam atrativo para solução de problemas, pois é de fácil aplicação. Além disso, o método dos Trapézios pode ser aplicado a qualquer número de subintervalos n .

Na literatura existem diversos métodos de integração intervalares, como a Integral de Moore (MOORE; STROTHER; T.YANG, 1960), Integral de Bedregal (NOBREGA G. A. S; BEDREGAL, 2010), Integral de Rall (RALL, 1982) e Simpson Intervalar (CAPRANI; MADSEN; NIELSEN, 2002).

O objetivo deste trabalho é definir de forma intervalar o método dos Trapézios,

a fim de complementar a literatura, já que o mesmo é inexistente. Após a definição do método com entradas intervalares, será feita sua implementação na linguagem Python, utilizando o pacote Intpy (BARRETO, 2016), que possui definido o tipo intervalo e operações sobre o tipo. Para validar os resultados do método intervalar, este será aplicado em funções da literatura por seus valores exatos serem conhecidos. Será feita a comparação com o método de Simpson Intervalar (CAPRANI; MADSEN; NIELSEN, 2002), para analisar a qualidade do intervalo solução retornado pelo método dos Trapézios. Adicionalmente, calculou-se o tempo de processamento para os dois métodos de integração, a fim de verificar se o tempo de execução do método dos Trapézios justifica seu uso.

1.1 Objetivos

O principal objetivo deste trabalho é definir de forma intervalar o método de integração dos Trapézios, utilizando as propriedades da matemática intervalar. A fim de atingir o objetivo principal, os objetivos específicos a serem executados são listados e detalhados logo abaixo:

- Definir com entradas intervalares o método de integração dos Trapézios: Na literatura encontram-se alguns métodos de integração intervalar como o método de Simpson Intervalar, Integral de Moore, Integral de Bedregal e Integral de Rall. Será aplicado o método de extensão intervalar para definir a forma intervalar do método composto de Trapézio.
- Após ser definida a forma intervalar, implementar o método dos Trapézios, tanto real quanto intervalar. Utilizar, para isso, a linguagem de programação Python (PYTHON, 2017), juntamente com a biblioteca IntPy, desenvolvido por Barreto e Campos (2008), a qual possibilita a programação com entradas intervalares.
- Analisar a qualidade do intervalo encapsulador: O intervalo resultante para o método dos Trapézios intervalar será analisado usando as medidas de erro absoluto e relativo e o cálculo de diâmetro do intervalo, para assim mostrar a qualidade do intervalo solução encontrado.
- Comparar o método intervalar dos Trapézios com o método de Simpson Intervalar: Serão analisados os resultados obtidos pelo método dos Trapézios e o método de Simpson, a fim de comparar os intervalos solução retornados e, assim, poder avaliar qual retorna intervalo com melhor aproximação e melhor tempo de processamento.

1.2 Organização deste trabalho

Esse trabalho organiza-se da seguinte maneira:

O [Capítulo 2](#) apresenta uma fundamentação teórica sobre matemática intervalar, citando os principais conceitos, o método da extensão intervalar e as operações aritméticas básicas. Após, descreve-se o ambiente IntPy e o conceito de exatidão máxima oferecido por tal pacote.

O [Capítulo 3](#) apresenta trabalhos já desenvolvidos, na área da matemática intervalar e integração numérica, os quais serviram de referência para o desenvolvimento deste trabalho.

No [Capítulo 4](#) são apresentados dois métodos de integração numérica, o método dos Trapézios e o método de Simpson, nas formas com valores reais e intervalares.

O [Capítulo 5](#) apresenta os resultados obtidos neste trabalho, através da análise numérica da qualidade dos intervalos encapsulares dos métodos de integração dos Trapézios Intervalar e Simpson Intervalar.

O [Capítulo 6](#) apresenta as considerações obtidas no presente trabalho, e as perspectivas de trabalhos futuros, os quais se deseja desenvolver a partir do conhecimento adquirido no presente trabalho.

Por fim são apresentadas as referências bibliográficas.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo serão apresentados conceitos da matemática intervalar, suas definições e operações básicas que servirão como base teórica para o desenvolvimento da monografia. Também será explicado o método da extensão intervalar, que será aplicado no método dos Trapézios. Por fim, será apresentado o ambiente computacional com suporte ao tipo intervalo, IntPy (BARRETO; CAMPOS, 2008), o qual será utilizado para implementar o método dos Trapézios na forma intervalar.

2.1 Matemática Intervalar

A matemática intervalar é uma ferramenta para a solução de problemas relacionados com erros numéricos, com base em um sistema algébrico formado por todos os intervalos fechados e operações definidas sobre ele.

Segundo Ratschek e Rokne (1988), os computadores empregam aritméticas de ponto flutuante. Nessa aritmética, números reais são aproximados por um subconjunto dos reais, chamado números de máquina. 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, 1996). Devido a representação de ponto flutuante, são gerados alguns tipos de erros. Define-se por erro a diferença entre o valor obtido por meio de aproximação e o valor exato.

Os erros ocorrem quando um valor real de entrada é aproximado para um número de máquina, como no caso do número $\pi = 3,1415\dots$, o qual é um número irracional, não pode ser representado por um número de dígitos decimais finitos, podendo ser aproximado, por exemplo, para o número de máquina 3,14. Neste caso, o erro depende exclusivamente da aproximação escolhida para π , podendo ser um erro de arredondamento ou truncamento. Também ocorrem erros devido aos resultados intermediários gerados na execução de operações aritméticas que vão se acumulando. Outro tipo de erro ocorre quando os dados de entrada são incertos, o que geralmente ocorre em experimentos. Pode-se encontrar também erros devido a conversões realizadas pela máquina.

Uma das alternativas de contornar erros causados pela máquina é a utilização de intervalos para representar valores reais, o que torna possível o controle automático de erros com limites confiáveis (LORETO, 2006), onde o intervalo solução contém o valor real.

Segundo Moore (1966), a matemática intervalar utiliza intervalos fechados de números reais, conforme a definição abaixo.

Definição 1 (*intervalo de reais*): *Seja \mathbb{R} o conjunto dos números reais e sejam $\underline{x}, \bar{x} \in \mathbb{R}$, tais que $\underline{x} \leq \bar{x}$. Então, o conjunto $\{x \in \mathbb{R} | \underline{x} \leq x \leq \bar{x}\}$ é um intervalo de reais ou simplesmente um **intervalo** e será denotado por $\mathbf{x} = [\underline{x}, \bar{x}]$.*

Se $\underline{x} = \bar{x}$, o intervalo $[\underline{x}, \bar{x}]$ é denominado intervalo degenerado. O conjunto dos intervalos de extremos reais \mathbb{R} \underline{x} e \bar{x} é denotado de \mathbb{IR} .

Do ponto de vista computacional, 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:

$$f(\mathbf{x}) = \{y | y = f(x), (\underline{x} \leq x \leq \bar{x})\}, \quad (2.1)$$

onde é possível determinar um intervalo $\mathbf{y} = [\underline{y}, \bar{y}]$ tal que $f(\mathbf{x}) \subseteq \mathbf{y}$, isto é $\underline{y} \leq f(x) \leq \bar{y}$. A partir disso, pode-se então definir uma função intervalar F associada a f pela transformação do intervalo $[\underline{x}, \bar{x}]$ em $[\underline{y}, \bar{y}]$, de modo que

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

Definição 2 (função intervalar): Seja $f : x \rightarrow y$ uma função. Se $\mathbf{x} = \text{Dom}(f) \in \mathbb{R}$ e $\mathbf{y} = \text{Cod}(f) \in \mathbb{R}$, então, dizemos que F é uma função intervalar de uma variável intervalar.

A função F , chamada extensão intervalar de f , deve ser a função que se afasta o mínimo possível da diferença da imagem $f(x)$.

O erro obtido no cálculo de $f(x)$ a partir do intervalo \mathbf{x} é dado pelo diâmetro do intervalo.

Definição 3 (diâmetro de um intervalo): O diâmetro de um intervalo $\mathbf{x} = [\underline{x}, \bar{x}] \in \mathbb{R}$ é denotado por $\text{diam}(\mathbf{x})$ ou $w(\mathbf{x})$, como sendo o número real não negativo ($x \in \mathbb{R}^+$) tal que $w(\mathbf{x}) = \bar{x} - \underline{x}$.

O cálculo de expressões na aritmética intervalar consiste em usar extensão das operações aritméticas com um conjunto de funções padrão. O suporte para a realização de operações matemáticas sobre intervalos é o uso de arredondamentos direcionados (KULISCH; MIRANKER, 1981), onde o extremo inferior \underline{x} é arredondado para baixo (∇) e o extremo superior \bar{x} é arredondado para cima (Δ), resultando em um intervalo com diâmetro mínimo, que contém a solução.

Segundo Kreinovich (2003), o intervalo solução, $\mathbf{y} = [\underline{y}, \bar{y}]$, pode ser obtido por meio de métodos de aproximação, técnicas de otimização, extensão intervalar, assim como métodos para encontrar o melhor algoritmo, o qual estima intervalos mais estreitos, para calcular a mesma função.

Na aritmética intervalar, uma distinção importante, é entre imagem intervalar de uma função e avaliação intervalar da função.

Definição 4 (imagem da função intervalar): A imagem intervalar de uma função f , contínua no intervalo \mathbf{x} , é definida como o intervalo limitado pelo mínimo da imagem de $f(x)$ e pelo máximo da imagem de $f(x)$, sendo x um elemento encapsulado pelo intervalo \mathbf{x} , de modo que:

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

O método de extensão intervalar proposto por Moore (1966), também conhecido como avaliação intervalar (OLIVEIRA; DIVERIO; CLAUDIO, 1997), está baseado no fato que em um computador, todo algoritmo consiste de operações elementares (aritméticas e lógicas). Para cada operação elementar $f(a, b)$, se são conhecidos os intervalos \mathbf{a} e \mathbf{b} para a e b , pode-se computar a imagem exata $f(a, b)$. Na extensão intervalar, repete-se a computação formando o programa f passo-a-passo, substituindo cada operação elementar de números reais pela correspondente operação da aritmética intervalar.

Em Santiago, Bedregal e Acióly (2006) a extensão intervalar do método é definida a seguir.

Definição 5 (extensão intervalar): A função $F : \mathbb{IR} \rightarrow \mathbb{IR}$ é uma extensão intervalar de uma função $f : \mathbb{R} \rightarrow \mathbb{R}$, se para todo $x \in \mathbb{R}$, $F([x, x]) = [f(x), f(x)]$.

Portanto, uma extensão intervalar de f é uma função intervalar que tem valor real quando os argumentos são todos reais (intervalos degenerados) e coincide com f .

Exemplo Seja $f(x) = 3x + x$ e um intervalo degenerado $\mathbf{x} = [2, 2] \subseteq \text{Dom}(f) = \mathbb{R}$. Substituindo x real por \mathbf{x} intervalo, tem-se: $F(\mathbf{x}) = [3, 3] \times [2, 2] + [2, 2] = [6, 6] + [2, 2] = [8, 8]$. Sabendo que $f(2) = 6 + 2 = 8$, tem-se então $F([2, 2]) = [f(2), f(2)]$. Logo, F é uma extensão intervalar da função real f (DUARTE, 2007).

Pode-se avaliar a corretude de uma função intervalar F em relação a uma função real f , pela seguinte regra:

Definição 6 Uma função intervalar F está correta em relação à função real f , se satisfaz à propriedade: $x \in [\underline{x}, \bar{x}] \rightarrow f(x) \in F([\underline{x}, \bar{x}])$.

É possível dizer que F é uma representação intervalar de f . Um algoritmo intervalar $A(\mathbf{x})$ 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).

2.1.1 Operações Aritméticas Intervalares Básicas

A aritmética intervalar foi desenvolvida principalmente por Moore (MOORE, 1966)(MOORE, 1979)(MOORE; KEARFOTT; CLOUD, 2009), e substitui cálculos reais por cálculos sobre intervalos. A seguir serão mostradas as operações de adição, subtração, multiplicação e divisão nos intervalos $\mathbf{x} = [\underline{x}, \bar{x}]$ e $\mathbf{y} = [\underline{y}, \bar{y}]$.

- Adição: a operação de adição entre dois intervalos \mathbf{x} e \mathbf{y} , é dada por:

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

Exemplo: Sejam os intervalos reais \mathbf{x} e \mathbf{y} , sendo $\mathbf{x} = [2, 4]$ e $\mathbf{y} = [3, 7]$, tem-se então $\mathbf{x} + \mathbf{y} = [5, 11]$.

- Subtração: a diferença entre dois intervalos, \mathbf{x} e \mathbf{y} , é dada como:

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

Exemplo: Sejam os intervalos $\mathbf{x}=[2,4]$ e $\mathbf{y}=[3,7]$, tem-se então $\mathbf{x} - \mathbf{y} = [-5,1]$.

- Multiplicação: 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, sendo representado por:

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

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

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

Exemplo: Sejam os intervalos $\mathbf{x} = [2, 4]$ e $\mathbf{y} = [1, 3]$, obtem-se o intervalo $\mathbf{x} \cdot \mathbf{y} = [2, 12]$.

- Divisão: o quociente entre dois intervalos, considerando a possibilidade de valores negativos, pode-se representado da seguinte forma:

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

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

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

Exemplo: Sejam os intervalos $\mathbf{x} = [2, 4]$ e $\mathbf{y} = [1, 2]$. O resultado da divisão será o intervalo $\mathbf{x}/\mathbf{y} = [2, 2]$.

2.2 IntPy: pacote de aritmética intervalar

Existem diversas linguagens de programação que permitem o desenvolvimento de algoritmos para cálculos numéricos com o tipo intervalo e operações sobre intervalos definidas, as quais são usualmente denominadas linguagens XSC (eXtended Scientific Computations) (MESQUITA, 2004). O presente trabalho fará uso do pacote IntPy (BARRETO, 2016), que é uma biblioteca para a linguagem de programação Python (PYTHON, 2017).

A implementação de operações e algoritmos de forma intervalar em computadores é realizada por meio do critério de semimorfismo, o qual foi proposto por Kulisch e Miranker

(1981). 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 (1981) e Kulisch (2008) 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.

A biblioteca IntPy foi desenvolvida por Barreto e Campos (2008), e fornece um *framework* para a matemática intervalar com exatidão máxima, desenvolvido como software livre usando a linguagem de programação Python.

Aplicada à matemática intervalar, a aritmética de exatidão máxima redefine a aritmética intervalar de modo que possuam um arredondamento do sistema de ponto flutuante, chamado de arredondamento direcionado (BARRETO; CAMPOS, 2008).

$$\mathbf{x} + \mathbf{y} = [\nabla(\underline{x} + \underline{y}), \Delta(\bar{x} + \bar{y})].$$

$$\mathbf{x} \times \mathbf{y} = [\min(\nabla\sigma), \max(\Delta\sigma)].$$

onde ∇ constitui o arredondamento direcionado para baixo e Δ o arredondamento direcionado para cima. σ representa o resultado da multiplicação.

A última versão do pacote implementa intervalos com arredondamentos direcionados, reconhece entrada de intervalos como *strings* e realiza as operações sobre intervalos.

Varjão (2011) desenvolveu extensões intervalares para complementar o pacote IntPy a fim de compor um ambiente auto validável em Python para operações envolvendo intervalos. As extensões intervalares implementadas por ele foram as funções, exponencial (exp), logarítmica (log), potência (pow), raiz quadrada (sqrt) e também funções trigonométricas seno (sin), cosseno(cos) e tangente (tan).

No presente capítulo foram apresentados os principais conceitos da matemática intervalar importantes para o entendimento do trabalho. Também foi apresentado o pacote IntPy, o qual será utilizado na implementação do trabalho. O capítulo a seguir apresenta os métodos numéricos de integração, regra dos Trapézios e Simpson, assim como, os métodos de integração intervalares existentes na literatura.

3 TRABALHOS RELACIONADOS

Este capítulo apresenta os trabalhos considerados relevantes e de maior relação com o que está sendo proposto na monografia. Além disso, são apresentados alguns resultados obtidos em cada um desses, os quais serviram de referência para o desenvolvimento dos objetivos apresentados.

3.1 Avaliação de Esquemas Numéricos para Problemas Difusivos 2D Resolvidos com Volumes Finitos

O trabalho desenvolvido por [Carvalho e Marchi \(2009\)](#) avalia o desempenho de alguns esquemas numéricos para resolver problemas difusivos bidimensionais com o método de volumes finitos. Para aproximar a integral da derivada da variável dependente nas faces dos volume de controle, foi usada diferença central (CDS) com integração numérica obtida pelas regras do Retângulo, Trapézio, Simpson e função biquadrática com integral analítica. São apresentados resultados do erro numérico e sua ordem efetiva comparado com o tamanho dos volumes de controle para malhas com 2x2 a 512x512 volumes e três variáveis de interesse. Entre os esquemas avaliados e para uma mesma malha, a função biquadrática resultou no menor erro numérico para todas as variáveis testadas.

No trabalho são avaliados o erro numérico da solução quando se utilizam diferentes técnicas de integração, e a convergência dos métodos, assim como o tempo de execução.

A seguir são apresentados alguns resultados visando mensurar o esforço computacional necessário para se resolver os problemas com os métodos de aproximações numéricas. A Tabela 1 apresenta os resultados do número de iterações necessárias para o método convergir e o tempo de CPU para obter as soluções numéricas pelos métodos: Retângulos, Trapézios, Simpson e a Função Biquadrática.

Tabela 1 – Número de iteração e tempo de CPU.

Interpolação	Iterações	Tempo de CPU(s)	Razão
Retângulo	125.342	672	1,00
Trapézio	65.713	684	1,02
Simpson	105.316	1.123	1,67
Função Biquadrática	114.785	1.223	1,82

Nos resultados encontrados por [Carvalho e Marchi \(2009\)](#), mostrados Tabela 1, podemos ver que tanto o número de iterações quanto o tempo de CPU do método dos Trapézios foi menor que os demais.

Os autores concluíram que a média do erro de discretização do esquema pela regra do Retângulo é menor do que a dos esquemas pelas regras do Trapézio e Simpson na solução da equação de Laplace. Em relação ao esquema que converge no menor tempo de CPU e para uma mesma malha, a função biquadrática gasta 82% a mais de tempo de

CPU para atingir o mesmo critério de convergência no caso da equação de Laplace, e 88% a mais no caso da equação de advecção-difusão.

3.2 Análise Numérica para a Solução de Integrais não Elementares

Neste trabalho é apresentado um estudo sobre integração numérica para resolver integrais definidas de funções contínuas que não possuem primitivas, ou cuja primitiva seja de difícil obtenção. Foram aplicados e comparados os métodos de Trapézio e de Simpson, e estimado o erro ao usar tais métodos de aproximação.

O trabalho mostra a aplicação de métodos numéricos para encontrar o perímetro de uma região delimitada por uma elipse. O cálculo foi realizado utilizando a Regra do Trapézio, Primeira e Segunda Regra de Simpson. A Tabela 2 apresenta os resultados obtidos e a estimativa de erro.

Tabela 2 – Comparação entre as regras de Trapézio e Simpson.

Técnica	Fórmula	Perímetro	Qtd. Arame	Erros
Regra do Trapézio	Simples	141,3	423,9	0,145
	Composta	141,8	425,4	0,004
Primeira Regra de Simpson	Simples	141,85	425,55	0,59
	Composta	141,69	425,07	0,007
Segunda Regra de Simpson	Simples	141,82	425,46	0,263
	Composta	141,69	425,07	0,016

Analisando os resultados, [Nóbrega \(2012\)](#) concluiu que para esta aplicação o melhor resultado será dado pela Regra do Trapézio na sua fórmula composta, devido ao erro encontrado ter sido o menor dentre os métodos aplicados.

Tendo em vista a aplicabilidade dos métodos numéricos nos diversos problemas da vida real, o trabalho apresentou alguns modelos conhecidos, obtendo resultados coerentes com aproximações ideais, sendo o modelo numérico muitas vezes a única maneira de descrever sistemas reais em diversas áreas como na engenharia civil, na matemática e na teoria das probabilidades.

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

O trabalho desenvolvido por [Finger \(2014\)](#) teve como objetivo definir com entradas intervalares as funções densidade de probabilidade com distribuição de Pareto e Gama, utilizando o método de extensão intervalar. Além disso, ela acrescenta a análise do esforço computacional dos algoritmos desenvolvidos, a partir dos resultados obtidos com as implementações utilizando o pacote IntPy.

Para obter o valor numérico das funções densidade de probabilidade das variáveis aleatórias contínuas com distribuições Uniforme, Exponencial, Normal, Gama e Pareto se

faz necessário o uso de integração numérica, uma vez que a primitiva da função nem sempre é simples de se obter. Usando integração numérica o resultado é obtido por aproximação e, portanto, afetado por erros de arredondamento ou truncamento. Utilizando intervalos para representar os números reais, é possível controlar a propagação desses erros, pois os resultados intervalares carregam consigo a segurança de sua qualidade.

Os intervalos foram obtidos utilizando o método de Simpson Intervalar, definido por [Caprani, Madsen e Nielsen \(2002\)](#), e a partir da primitiva da função. Além disso, são apresentados os resultados de qualidade para cada função de acordo com as medidas de erro absoluto, erro relativo, e 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, quando possível e o método de Simpson, os quais podem ser observados na [Tabela 3](#) apresentada logo a seguir.

Tabela 3 – 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	$O(1)$	$O(2^n)$	$O(1)$	$O(2^n)$

A partir da análise da complexidade computacional dos algoritmos, verificou-se que, ao utilizar intervalos para calcular a função densidade de probabilidade das variáveis aleatórias contínuas, a quantidade de trabalho despendido pelo algoritmo aumenta em relação a forma real. Em ambas as formas, real e intervalar, as melhores soluções foram obtidas através da primitiva da função. Constatou que o esforço computacional se mantém o mesmo, tanto na forma real quanto na intervalar.

3.4 Comparando Diferentes Métodos de Integração Intervalar

Considerando que métodos numéricos podem ser usados para o cálculo de integrais, é necessário que estes sejam suportados pela matemática intervalar e pela aritmética de exatidão máxima.

O trabalho desenvolvido por [BALBONI et al. \(2015\)](#) compara os métodos de integração intervalar para obter o método que retorna a melhor solução com exatidão máxima. Os métodos comparados foram: Simpson Intervalar, Integral de Bedregal e Integral de Moore. A comparação entre os métodos foi realizada pela validação do resultado e da

análise da qualidade dos intervalos solução através de medidas de Erro Absoluto e Erro Relativo.

As medidas de erro aplicadas no trabalho foram:

- Erro Absoluto (EA):

$$EA = |x - m(\mathbf{x})| < \frac{w(\mathbf{x})}{2},$$

onde x é o valor real, $m(\mathbf{x})$ é o ponto médio do intervalo \mathbf{x} e $w(\mathbf{x})$ é o diâmetro do intervalo \mathbf{x} .

- Erro Relativo (ER):

$$ER = \left| \frac{x - m(\mathbf{x})}{x} \right| < \frac{w(\mathbf{x})}{2\min|\mathbf{x}|},$$

onde os parâmetros são exatamente os mesmos do Erro absoluto, com exceção da expressão $\min|\mathbf{x}|$ que consiste no módulo limite inferior do intervalo.

Os métodos da Integral de Moore, Integral de Bedregal e Simpson Intervalar, foram implementados usando a biblioteca IntPy. Foi considerada uma precisão de 14 casas decimais, e foram usadas funções simples para comparar os métodos. A comparação dos resultados é vista na [Tabela 4](#) e [Tabela 5](#).

Tabela 4 – Erro Absoluto.

Métodos	Exemplo 1	Exemplo 2
Simpson Caprani	6.2172489379e-15 < 9.72555369572e-14	3.3784086639e-13 < 5.12256903562e-13
Integral de Bedregal	8.881784197e-15 < 1.02140518266e-14	0.0 < 9.99200722163e-15
Integral de Moore	8.881784197e-15 < 1.02140518266e-14	0.0 < 9.99200722163e-15

Tabela 5 – Erro Relativo.

Métodos	Exemplo 1	Exemplo 2
Simpson Caprani	1.24987804277e-14 <= 1.97080922405e-14	4.032678289887e-13 <= 6.11461637512e-13
Integral de Bedregal	1.79982577539e-15 <= 2.0697996417e-15	0.0 <= 1.19270800555e-14
Integral de Moore	1.79982577539e-15 <= 2.0697996417e-15	0.0 <= 1.19270800555e-14

Analisando os resultados dos intervalos solução e medidas de qualidade, erros absoluto e relativo, os autores concluíram que os métodos da Integral de Bedregal e da Integral de Moore obtêm um intervalo solução melhor com máxima exatidão. [BALBONI et al. \(2015\)](#) afirmam que um dos problemas de utilizar o método de Simpson Intervalar é a alta complexidade, tal que para se obter resultados melhores o seu custo também é aumentado devido ao número de subdivisões no intervalo de integração.

3.5 Integração Intervalar Revisitada

O trabalho apresenta uma visão geral das abordagens para autovalidação de fórmulas de integração e um algoritmo de integração numérica verificado com uma adaptação estratégica. No trabalho, [Galdino \(2012\)](#) mostra dois métodos de integração intervalar: Regra de Trapézio e Regra 1/3 de Simpson aplicados a um conjunto de funções, a fim de comparar o algoritmo adaptativo proposto. Este trabalho mostra uma definição intervalar para a Regra de Trapézio, sendo esta a regra simples do método. O método de Simpson adaptativo consiste em calcular resultado delimitando uma tolerância para o erro.

O principal mecanismo para obter um intervalo estreito de resultados é para aumentar o número de subdivisões. Em seguida, consider-se a progressão do erro de discretização à medida que aumenta-se n .

Tabela 6 – Regra dos Trapézios com verificação

Subdivisões	Resultado Verificado
10	0,479421870930105, 6,395545854416262e-004
100	0,479428226049601, 16,395545891768606e-007
1000	0,479428226688739, 6,395906027023557e-010
10000	0,479428226688802, 1,002808946992673e-012
100000	0,479428226687608, 3,629152534045943e-012
1000000	0,479428226671977, 3,620936883663717e-011

Tabela 7 – Regra 1/3 de Simpson com verificação

Subdivisões	Resultado Verificado
10	0,479428217025575, 2,131848625963606e-007
100	0,479428226688792, 2,139066701545289e-012
1000	0,479428226688796, 7,244205235679146e-014
10000	0,479428226688723, 7,243095012654521e-013
100000	0,479428226688001, 7,241873767327434e-012
1000000	0,479428226680795, 7,241879318442557e-011

Observa-se que se o número de subdivisões for extremamente grande isso pode levar a um arredondamento significativo (mais termos na soma, cada um com um erro de arredondamento para contribuir). A regra trapezoidal verificada com 10^5 subdivisões é maior do que 10^4 subdivisões e verificada Simpson 1/3 regra com 10^4 as subdivisões são mais largas do que 10^3 .

O autor concluiu que a técnica adaptativa do intervalo geralmente é mais eficiente do que o intervalo da regra de Simpson com resultados de intervalos estreitos, sendo possível controlar o erro, mas essa técnica aumenta o comprimento do intervalo, reduzindo a precisão.

3.6 Critérios para Análise e Escolha de Ambientes Intervalares

Dentre os vários ambientes computacionais intervalares, pode ser difícil a decisão de qual é o mais adequado ou o melhor para se trabalhar na execução de um trabalho futuro. 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. A análise desses critérios foi realizada para os seguintes ambientes: Maple Intervalar, IntLab, IntPy, C-XSC, Fortran-XSC, PascalXSC e Java-XSC.

Como ainda não existem na literatura critérios para avaliar ambientes intervalares, e pelos diferentes tipos de ambientes encontrados, que podem ser softwares, linguagens de programação ou bibliotecas, o trabalho utiliza como critérios para avaliar o melhor ambiente intervalar: avaliação para linguagens de programação, qualidade de software e qualidade do resultado intervalar.

A fim de verificar qual ambiente retorna o intervalo solução com melhor qualidade (intervalo com menor diâmetro), foi apresentada a análise da qualidade dos intervalos obtidos na linguagem C-XSC, no pacote IntPy, na biblioteca IntLab e no software Maple Intervalar. A análise foi realizada através da aplicação dos indicadores estatísticos descritivos intervalares: Média, Mediana, Variância, Amplitude Total, Desvio Padrão, Coeficiente de Variação, Covariância e Coeficiente de Correlação. A Tabela 8 apresenta um quadro onde são mostrados os resultados intervalares obtidos a partir de valores reais aplicados aos indicadores citados.

Tabela 8 – Valores reais e intervalares

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	21.2331	[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]

Concluiu-se, dentre as análises realizadas, que a linguagem de programação Python apresentou os melhores resultados 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.

4 MÉTODOS DE INTEGRAÇÃO NUMÉRICA E DEFINIÇÃO DO MÉTODO DOS TRAPÉZIOS INTERVALAR

O presente capítulo apresenta uma breve introdução sobre integração numérica, as fórmulas e definições, assim como as implementações, para os métodos de integração explorados nesta monografia. A [seção 4.1](#) apresenta o método dos Trapézios e a definição intervalar do método, realizada neste trabalho. Esta seção também mostra a implementação do método intervalar, usando a linguagem Python e o pacote IntPy. A [seção 4.2](#) apresenta o método de Simpson e sua definição intervalar, a qual é existente na literatura.

A utilização de métodos numéricos se faz necessária quando: a expressão de f é complexa de resolver, não se conhece uma primitiva de f , ou cuja primitiva não seja fácil de obter; a expressão de f é conhecida apenas em alguns pontos.

Se não possuímos a expressão analítica de f , não podemos usar outro método que não o numérico. A integração numérica pode trazer bons resultados quando falham outros métodos, e seus procedimentos são facilmente implementados.

Existem dois métodos principais para desenvolver uma definição intervalar de uma função: o método da imagem intervalar e o método da extensão intervalar. Partindo dos trabalhos de indicadores estatísticos intervalares feitos por [Loreto \(2006\)](#) e extensão intervalar para variáveis aleatórias com distribuições Uniforme, Normal, Gama, Exponencial e Pareto, desenvolvido por [Finger \(2014\)](#), e a extensão intervalar do método de Simpson Intervalar, desenvolvida por [Caprani, Madsen e Nielsen \(2002\)](#), os quais obtiveram resultados satisfatórios usando o método de extensão intervalar, deu-se a escolha deste método para a definição intervalar do método dos Trapézios na sua regra composta.

O método de Simpson Intervalar ([CAPRANI; MADSEN; NIELSEN, 2002](#)) foi utilizado para análise neste trabalho por ser um método que dispõe de uma forma real, sendo possível um comparativo com o método dos Trapézios na forma real e intervalar.

4.1 Método dos Trapézios

- Forma real

A regra do trapézio é a versão mais simples dos métodos de *Newton-Cotes*, com $n = 1$, aproximada por um polinômio interpolador de primeiro grau, na qual usa-se apenas dois pontos. Como a regra do Trapézio requer pelo menos um intervalo para cada aplicação, o inteiro n pode ser tanto ímpar quanto par. A fórmula da regra simples é dada através da seguinte equação:

$$\int_a^b f(x)dx = \frac{h}{2} [f(x_0) + f(x_1)]. \quad (4.1)$$

Essa equação pode ser comparada pela fórmula da área de trapézio, na qual h seria a altura, $f(x_0)$ e $f(x_1)$ são as bases menores e maiores, respectivamente.

A regra simples do trapézio produz um erro de truncamento, dado por:

$$E_T = -\frac{h^3}{12}f''(\xi), \quad (4.2)$$

em que $x_0 < \xi < x_1$, sendo ξ a média das derivadas de segunda ordem.

Como o termo de erro para a regra do trapézio envolve f'' , a regra fornece o resultado exato quando aplicada a qualquer função cuja segunda derivada seja identicamente zero, ou seja, qualquer polinômio de grau menor ou igual a 1 (BURDEN; FAIRES, 2008). Em questão do erro, esse método pode ser utilizado quando o intervalo $[a,b]$ é pequeno. Assim, a aproximação é considerada razoável.

Ao usar a regra dos trapézios simples obteremos um resultado impreciso (CLÁUDIO; MARINS, 1994), logo se faz necessário aumentar a quantidade de intervalos para melhorar a precisão do método. Se subdividimos o intervalo $[a,b]$ em n subintervalos e em cada um deles aproximamos f por uma reta, ou seja, aplicamos novamente a regra simples dos trapézios, obtém-se a regra composta dos trapézios, vista na Equação (4.3).

Os valores de cada um dos pontos x_i das subdivisões podem ser obtidos a partir da expressão: $x_i = x_0 + ih$.

A fórmula composta dos trapézios será:

$$\int_a^b f(x)dx = \frac{h}{2} \left[f(x_0) + f(x_n) + 2 \sum_{i=1}^{n-1} f(x_i) \right]. \quad (4.3)$$

O erro de truncamento será dado por:

$$E_{Tc} = -\frac{(b-a)h^2}{12}f''(\xi). \quad (4.4)$$

- Forma intervalar

O método composto dos Trapézios ainda não possui definição intervalar na literatura, foi utilizado o método de extensão intervalar para realizar a definição. A fórmula do método dos Trapézios composto para entradas intervalares é definida como:

$$\int_a^b f(\mathbf{x})dx = \frac{w(\mathbf{x})}{2} \left[f(\mathbf{x}_0) + f(\mathbf{x}_n) + 2 \sum_{i=1}^{n-1} f(\mathbf{x}_i) \right].$$

O método possui valores intervalares, sendo o h substituído pelo diâmetro do intervalo $w(\mathbf{x})$.

- Implementação do método dos Trapézios Intervalar

def trapezio_intervalar (n, a, b):

```

x = IReal(a, b)           #entrada do tipo intervalar
h = (x.sup - x.inf) / float(n)   #tamanho do intervalo
aux = x.inf + h

```

```

soma = IReal(0)
result = IReal(0)
listaInt = [] #lista de intervalos
for i in range (n-1):
    listaInt = listaInt + [IReal(aux, aux + h)]
    aux = aux + h
for i in range (n-1):
    soma += IReal(2) * F(listaInt[i])
soma += F(listaInt[i].inf) + F(listaInt[i].sup)
soma = soma * ((listaInt[i].sup) - (listaInt[i].inf))/2)
return soma

```

4.2 Método de Simpson

- Forma real

O método de integração de Simpson aproxima f por um polinômio de grau 2. Para se interpolar f por um polinômio de grau 2 precisa-se de três pontos para a construção da fórmula da regra simples (RUGGIERO; LOPES, 1996). Resolvendo a integral definida do polinômio p , tem-se a regra simples de Simpson:

$$\int_a^b f(x)dx = \frac{h}{3} [f(x_0) + 4f(m) + f(x_1)], \quad (4.5)$$

onde m é o ponto médio, dado por $\frac{(a+b)}{2}$.

O erro de truncamento da regra simples de Simpson é dado por:

$$E_S = -\frac{h^5}{90} f^4(\xi), \quad (4.6)$$

onde $\xi \in [a, b]$.

A regra composta de Simpson consiste em subdividirmos o intervalo $[a, b]$ em n subintervalos de amplitude h , onde n é um número par de subintervalos, pois cada parábola utilizará 3 pontos consecutivos. A regra de Simpson composta é dada por:

$$\int_a^b f(x)dx = \frac{h}{6} [f(x_0) + 4f(m) + f(x_n)]. \quad (4.7)$$

O erro de truncamento na regra composta é dado por:

$$E_{Sc} = \frac{h^5}{2880} f^4(\xi). \quad (4.8)$$

Como o termo de erro envolve a quarta derivada de f , a regra de Simpson fornece resultados exatos quando aplicada a qualquer polinômio de grau menor ou igual a 3.

- Forma intervalar

O método de Simpson Intervalar (CAPRANI; MADSEN; NIELSEN, 2002) é uma extensão intervalar do método de Simpson 1/3.

A definição intervalar do método é apresentada pela Equação (4.9):

$$S_i = \frac{w(A_i)}{6}(F(\underline{x}) + 4F(m(A_i)) + f(\bar{x})) - \frac{w(A_i)^5}{2880}G(A_i). \quad (4.9)$$

Assumindo que G é uma extensão intervalar linear de f^4 , tem-se

$$w(S_i) = \frac{1}{2880}w(x_i)^5w(G(x_i)) \leq \frac{1}{2880}w(x_i)^5K_Gw(x_i) = K_2w(x_i)^6,$$

onde $w(S_i) = \frac{K_3}{n^5}$.

A simplicidade da fórmula do método dos Trapézios é um dos fatores que a tornam atraente e utilizada na solução de problemas, apesar do erro de truncamento ser maior do que ao utilizar o método de Simpson. Além disso, o método dos Trapézios pode ser aplicado a qualquer número de subintervalos n , enquanto que o método de Simpson necessita que o n seja múltiplo de 2 ou de 3, dependendo do polinômio interpolador que foi utilizado na aproximação.

5 ANÁLISE DA QUALIDADE DOS INTERVALOS ENCAPSULADORES

O objetivo deste capítulo é analisar os intervalos encapsuladores obtidos pelo método de Trapézio na forma intervalar. Para isso foram computados exemplos aplicando os métodos dos Trapézios e de Simpson, tanto na forma real quanto intervalar, a fim de compará-los. Serão comparados os resultados de diferentes funções para classificar o melhor método para resolver cada tipo de função. A [seção 5.1](#) apresenta os resultados obtidos utilizando os métodos dos Trapézios e de Simpson.

Para comparar os resultados encontrados e a exatidão dos mesmos fez-se uso de um ambiente computacional de desenvolvimento intervalar, sendo implementados os métodos de integração para entradas com valores reais e valores intervalares, utilizando a linguagem de programação Python e o pacote com suporte intervalar IntPy ([BARRETO, 2016](#)). A escolha deu-se através de trabalhos vistos na literatura que fazem um comparativo entre ambientes intervalares, o pacote IntPy foi o que demonstrou os melhores resultados, retornando intervalos com qualidade e em tempos de processamento menores.

Após a implementação foi realizada a análise numérica através da comparação dos resultados reais com os intervalares. Para verificar a qualidade dos intervalos obtidos, fez-se a análise de erros, utilizando métricas de qualidade de intervalos, as quais são diâmetro do intervalo, erro absoluto e erro relativo. O diâmetro do intervalo $w(\mathbf{x})$, é o comprimento do intervalo e pode ser usado como medida para qualidade de aproximação, quanto mais estreito o intervalo, maior sua precisão. O erro absoluto refere-se a diferença entre o valor exato de um número e seu valor aproximado, e erro relativo, o qual está relacionado ao erro percentual podendo ser calculado pelo quociente entre o erro absoluto e o valor aproximado. As estimativas de erro para computação com intervalos estão descritas a seguir:

- Diâmetro: $w(\mathbf{x}) = \bar{x} - \underline{x}$.
- Erro Absoluto (EA): $EA = |x - m(\mathbf{x})| < \frac{w(\mathbf{x})}{2}$, onde x é o valor real, $m(\mathbf{x})$ é o ponto médio do intervalo \mathbf{x} e $w(\mathbf{x})$ é o diâmetro do intervalo \mathbf{x} .
- Erro Relativo (ER): $ER = \left| \frac{x - m(\mathbf{x})}{x} \right| < \frac{w(\mathbf{x})}{2\min|\mathbf{x}|}$, se $0 \notin x$;

No contexto da matemática intervalar, a interpretação usualmente aceita para um intervalo 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. Dessa forma, o valor exato estaria limitado pelo intervalo apresentado, entre seu limite inferior \underline{x} e seu limite superior \bar{x} ([SUNAGA, 2009](#))

Estas medidas foram aplicadas aos intervalos solução obtidos na computação das funções de exemplo pelos métodos dos Trapézios e Simpson, com o objetivo de verificar a qualidade de aproximação dos resultados intervalares obtidos.

O conjunto de funções utilizadas nos cálculos foram retiradas do trabalho desenvolvido por [Galdino \(2012\)](#), por seus resultados exatos serem conhecidos, a fim de validar os valores intervalares encontrados pelo método dos Trapézios Intervalar. Para o cálculo das funções foram utilizados os valores 10, 100, 1000 e 10000 subdivisões.

Com o objetivo de adicionar o tempo de processamento dos métodos de integração, ao critério de classificação, também serão apresentados os tempos de execução dos métodos com entradas reais e intervalares, onde o mesmo foi calculado através de uma média do tempo obtido em 10 execuções do cálculo de cada função.

Para todos os resultados foi utilizado o mesmo computador com as seguintes configurações: processador Intel Core i3 CPU M 330 @ 2.13GHz x 4; memória RAM 3 GB DDR3 1066 MHz.

Todos os resultados intervalares e reais utilizaram o sistema de ponto flutuante $F(10, 17, -17, 17)$.

5.1 Análise numérica

Para obter os resultados numéricos foram utilizadas as funções a seguir, retiradas do trabalho de [Galdino \(2012\)](#), por seus valores serem conhecidos e com isso possibilitar a validação dos resultados obtidos neste trabalho.

$$\begin{aligned}
 f_1 &= \int_0^1 e^x dx = 1.7182818284590452354 \\
 f_2 &= \int_{-1}^1 \left(\frac{23}{25} \cosh(x) - \cos(x) \right) dx = 0.47942822668880166736 \\
 f_3 &= \int_{-1}^1 (x^4 + x^2 + 0.9)^{-1} dx = 1.5822329637296729331 \\
 f_4 &= \int_0^1 (1 + x^4)^{-1} dx = 0.86697298733991103757 \\
 f_5 &= \int_0^1 2(2 + \sin(10\pi x))^{-1} dx = 1.154700538379251529 \\
 f_6 &= \int_0^1 (1 + x)^{-1} dx = 0.69314718055994530942 \\
 f_7 &= \int_0^1 (1 + e^x)^{-1} dx = 0.37988549304172247537 \\
 f_8 &= \int_{0.1}^1 \frac{\sin(100\pi x)}{(\pi x)} dx = 0.0090986375391668429156 \\
 f_9 &= \int_0^{10} \sqrt{50} e^{-50\pi^2} dx = 0.5 \\
 f_{10} &= \int_0^{10} 25e^{-25x} dx = 1.0 \\
 f_{11} &= \int_0^{10} 50(\pi(2500x^2))^{-1} dx = 0.49936338107645674464 \\
 f_{12} &= \int_{-1}^1 (1.005 + x^2)^{-1} dx = 1.5643964440690497731 \\
 f_{13} &= \int_0^1 (1 + (230x - 30)^2)^{-1} dx = 0.013492485649467772692
 \end{aligned}$$

A seguir, apresentam-se os resultados obtidos através da aplicação do método dos Trapézios, definido de forma intervalar no presente trabalho, e de Simpson, já presente na literatura, ambos aplicados nas 13 funções de exemplo.

A [Tabela 9](#) apresenta os resultados obtidos no cálculo da função f_1 utilizando os métodos dos Trapézios e de Simpson, para entradas reais e intervalares.

Tabela 9 – Resultados obtidos no exemplo f1.

f1				
n	Trapézio Real	Trapézio Intervalar	Simpson Real	Simpson Intervalar
10	1.719713491389314	[1.622096290833137, 1.768056601948833]	1.7182827819248232	[1.718281888103855, 1.718281888103857]
100	1.718296147450418	[1.709570847156018, 1.726483191879523]	1.718281828554504	[1.718281828465007, 1.718281828465018]
1000	1.718281971649196	[1.717421473631122, 1.719137038536556]	1.7182818284590564	[1.718281828458991, 1.718281828459105]
10000	1.718281829890868	[1.718195902209320, 1.718367703211823]	1.7182818284589612	[1.718281828458222, 1.7182818284593644]

Analisando a tabela, é possível observar que os valores reais obtidos estão contidos nos intervalos solução encontrados pelo método dos Trapézios intervalar para todos os n subintervalos utilizados. Os intervalos resultantes pelo método de Simpson com 10 e 100 subintervalos não encapsulam os valores obtidos pelo método na forma real.

A [Tabela 10](#) apresenta os resultados dos métodos com entradas reais e intervalares para a função $f2$.

Tabela 10 – Resultados obtidos no exemplo f2.

f2				
n	Trapézio Real	Trapézio Intervalar	Simpson Real	Simpson Intervalar
10	0.49224487664969757	[0.27353439547931196, 0.5881474593623484]	0.4794323255909411	[0.4794284904949504, 0.4794284904949512]
100	0.47955640365446817	[0.45999697298380626, 0.4976090191469075]	0.4794282271147981	[0.47942822671543306, 0.47942822671543694]
1000	0.47942950845951443	[0.4775070111483984, 0.48133665587585517]	0.4794282266888455	[0.47942822668878793, 0.47942822668882445]
10000	0.4794282395064301	[0.47923633468916954, 0.4796199905424556]	0.47942822668872376	[0.47942822668848545, 0.4794282266888479]

Como mostra a tabela, os intervalos solução do método dos Trapézios Intervalar encapsulam os valores reais para todos os subintervalos testados. Utilizando o método de Simpson, apenas o intervalo resultante para 10000 subintervalos encapsula o valor real obtido pelo método.

A [Tabela 11](#) mostra os resultados obtidos para o cálculo da função $f3$ utilizando os métodos dos Trapézios e de Simpson.

Tabela 11 – Resultados obtidos no exemplo f3.

f3				
n	Trapézio Real	Trapézio Intervalar	Simpson Real	Simpson Intervalar
10	1.577479324165713	[1.4785128296436527, 1.7514066111758124]	1.582221117071613	[1.5822322924808947, 1.5822322924808963]
100	1.58218540z1532032	[1.5673007144169104, 1.5976615190112837]	1.582232962812423	[1.5822329636726349, 1.5822329636726484]
1000	1.582232488105438	[1.580704214554543, 1.5837664897575343]	1.582232963729582	[1.5822329637296035, 1.5822329637297308]
10000	1.582232958973474	[1.5820797450869903, 1.582386229955491]	1.582232963729715	[1.582232963728984, 1.582232963730252]

Analisando a tabela, é possível perceber que os intervalos encontrados pelo método dos Trapézios Intervalar contêm os valores reais em todos subintervalos testados. Os intervalos encontrados pelo método de Simpson Intervalar para 10, 100 e 1000 subintervalos, não contêm os valores reais do método.

A Tabela 12 apresenta os resultados do cálculo da função $f4$ utilizando os métodos dos Trapézios e Simpson com entradas reais e intervalares.

Tabela 12 – Resultados obtidos no exemplo f4.

f4				
n	Trapézio Real	Trapézio Intervalar	Simpson Real	Simpson Intervalar
10	0.8661388208039201	[0.8568012899845178, 0.8964184628605529]	0.8669763082948306	[0.8336415287970268, 0.8336415287970277]
100	0.8669646539232446	[0.8646158574367337, 0.8695153676117121]	0.8669729876732439	[0.863639654044074, 0.8636396540440818]
1000	0.8669729040065693	[0.8667244052520171, 0.8672234047530935]	0.8669729873399439	[0.8666396540065425, 0.8666396540066166]
10000	0.8669729865066089	[0.8669480015074877, 0.866997991507728]	0.8669729873399433	[0.866939654006164, 0.8669396540069008]

É possível perceber que os intervalos solução do método dos Trapézios encapsulam os valores reais obtidos pelo método na forma real para todos subintervalos testados. Os intervalos solução do método de Simpson não encapsularam os valores obtidos pelo método real em nenhum dos subintervalos testados para esta função, com mais subdivisões o método deve convergir.

A Tabela 13 apresenta os resultados do cálculo da função $f5$ pelos métodos dos Trapézios e Simpson, na formas reais e intervalares.

Tabela 13 – Resultados obtidos no exemplo f5.

f5				
n	Trapézio Real	Trapézio Intervalar	Simpson Real	Simpson Intervalar
10	1.0000000000000002	[0.999999999999953, 1.000000000000044]	1.000000000000013	[1.22222222222212, 1.22222222222228]
100	1.1547005383876563	[1.0902858971227785, 1.2222809250424858]	1.1547020069846476	[1.1547005383764453, 1.1547005383764557]
1000	1.1547005383792506	[1.1480573122341902, 1.1613751830330465]	1.1547005383792521	[1.1547005383792082, 1.1547005383792983]
10000	1.1547005383792677	[1.1540341072084166, 1.1553672837096554]	1.1547005383792721	[1.1547005383787161, 1.154700538379603]

Observa-se que os valores obtidos pelo método dos Trapézios real para os subintervalos testados, estão contidos nos intervalos solução do método na forma intervalar. Os resultados reais pelo método de Simpson para 10 e 100 subintervalos não estão contidos no intervalo solução do método intervalar, já os resultados reais para 1000 e 10000 subintervalos estão contidos nos intervalos.

A [Tabela 14](#) apresenta os resultados reais e intervalares da função $f6$, utilizando os métodos dos Trapézios e de Simpson.

Tabela 14 – Resultados obtidos no exemplo f6.

f6				
n	Trapézio Real	Trapézio Intervalar	Simpson Real	Simpson Intervalar
10	0.693771403175428	[0.6728649704268898, 0.7202333914795219]	0.6931502306889303	[0.6931473746651158, 0.6931473746651166]
100	0.6931534304818241	[0.6906912458211438, 0.6956661201930094]	0.6931471808723669	[0.6931471805794723, 0.6931471805794791]
1000	0.6931472430599372	[0.6928976183726872, 0.6933973682476898]	0.6931471805599752	[0.6931471805599139, 0.6931471805599787]
10000	0.6931471811849604	[0.6931221849349699, 0.6931721824354565]	0.6931471805599584	[0.6931471805595771, 0.6931471805602243]

Como mostra a tabela, para cada subintervalo testado, os valores obtidos pelo método dos Trapézios real estão contidos nos intervalos resultantes pela forma intervalar. Os intervalos solução encontrados pelo método de Simpson Intervalar para 10 e 100 subintervalos, não contêm os valores encontrados no método real. Para 1000 e 10000 subintervalos os valores reais estão contidos.

A [Tabela 15](#) mostra os resultados encontrados na integração da função $f7$, utilizando os métodos dos Trapézios e de Simpson, nas formas reais e intervalares.

Tabela 15 – Resultados obtidos no exemplo f7.

f7				
n	Trapézio Real	Trapézio Intervalar	Simpson Real	Simpson Intervalar
10	0.3799299955621036	[0.37143672530597277, 0.39253167556847357]	0.3798854430223608	[0.37988548992491167, 0.37988548992491217]
100	0.3798859379435242	[0.3787602501432208, 0.3810511293667175]	0.3798854930367404	[0.3798854930414095, 0.37988549304141295]
1000	0.3798854974907277	[0.3797702632328416, 0.3800011251541501]	0.3798854930417217	[0.3798854930417056, 0.37988549304174013]
10000	0.3798854930862224	[0.37987394310638195, 0.37989704699839727]	0.3798854930417313	[0.3798854930415245, 0.3798854930418686]

É possível notar que os intervalos obtidos pelo método dos Trapézios Intervalar, encapsulam os resultados reais para cada subintervalo testado. Enquanto que, para 10 e 100 subintervalos, os intervalos solução do método de Simpson não encapsulam o valores obtidos no método com entradas reais.

A [Tabela 16](#) apresenta os resultados para o cálculo da função $f8$ utilizando os métodos dos Trapézios e de Simpson, reais e intervalares

Tabela 16 – Resultados obtidos no exemplo f8.

f8				
n	Trapézio Real	Trapézio Intervalar	Simpson Real	Simpson Intervalar
10	1.8318124672719585e-15	[-3.8746859945135635e-14, 4.192840969540898e-14]	-1.2006013107942834e-14	[0.0742271259826426, 0.07422712598265166]
100	0.0020325646511704	[-0.4635541846582267, 0.47214586786105306]	0.0260953389177536	[0.009328164458373115, 0.009328164458393273]
1000	0.0090378069913575	[-0.05759926154770708, 0.07546084343087929]	0.0090989621640819	[0.009083935377322844, 0.009083935377344353]
10000	0.0090980300311234	[0.0024185918447622748, 0.015775174677278538]	0.0090986375713253	[0.009097143429513482, 0.009097143429550099]

Na tabela, observa-se que nos intervalos resultantes pelo método dos Trapézios Intervalar, apenas o valor com 10 subintervalos não contém o valor obtido pelo método real. Nos demais subintervalos, os intervalos solução pelo método dos Trapézios intervalar encapsularam os valores reais. Para a função $f8$, os intervalos obtidos pelo método de Simpson Intervalar não encapsulam os valores reais do método com os subintervalos utilizados no trabalho.

A [Tabela 17](#) apresenta os resultados do cálculo da função f_9 , pelos métodos dos Trapézios e de Simpson.

Tabela 17 – Resultados obtidos no exemplo f_9 .

f9				
n	Trapézio Real	Trapézio Intervalar	Simpson Real	Simpson Intervalar
10	3.5355339059327378	[4.272354783661453e-68, 7.071067811865477]	2.3570226039551585	[1.178511301977579, 1.1785113019775801]
100	0.5018674427438695	[0.14831405215059554, 0.8554208333371466]	0.4325740089778648	[0.49937751910159184, 0.4993775191015933]
1000	0.49999999999999994	[0.46464466094067075, 0.5353553390593411]	0.49999999999999983	[0.49999999999999867, 0.50000000000000205]
10000	0.49999999999999994	[0.4964644660940543, 0.5035355339060422]	0.49999999999999917	[0.4999999999999879, 0.50000000000002038]

Como mostra a tabela, todos os intervalos solução do método dos Trapézios encapsularam os valores pelo método real para cada subintervalo utilizado. Nos intervalos resultantes pelo método de Simpson, os valores reais para 10 e 100 subintervalos, não foram encapsulados pelo método intervalar.

A [Tabela 18](#) apresenta os resultados obtidos pelos métodos dos Trapézios e de Simpson, na formas reais e intervalares, para a função f_{10} dos exemplos.

Tabela 18 – Resultados obtidos no exemplo f_{10} .

f10				
n	Trapézio Real	Trapézio Intervalar	Simpson Real	Simpson Intervalar
10	12.500000000347198	[3.471985966289228e-10, 25.00000000034723]	8.333333333796265	[4.1667287776686, 4.1667287776686095]
100	1.47356372458463	[0.22356372458462953, 2.723563724584664]	1.1201122079356658	[1.0113972892934764, 1.0113972892934988]
1000	1.0052029160469496	[0.880202916046939, 1.130202916047081]	1.0000215409731326	[1.0000013538179773, 1.0000013538181936]
10000	1.000052082790805	[0.9875520827907268, 1.0125520827918622]	1.0000000021699724	[1.0000000001355303, 1.0000000001376481]

Observa-se que os intervalos resultantes pelo método dos Trapézios contêm os valores reais para todos os subintervalos utilizados. Nas subdivisões utilizadas, os intervalos resultantes pelo método não encapsularam os valores reais, necessitando de um subintervalo maior do que 10000 para convergir.

A [Tabela 19](#), mostra os resultados obtidos para a computação da função f_{11} , utilizando os métodos dos Trapézios e de Simpson, com entradas reais e intervalares.

Tabela 19 – Resultados obtidos no exemplo f11.

f11				
n	Trapézio Real	Trapézio Intervalar	Simpson Real	Simpson Intervalar
10	7.967578697188159	[0.009888744663722113, 15.925304459207752]	5.316741888759356	[2.6763519389027843, 2.676351938902789]
100	0.8972014191639541	[0.1014300826669538, 1.692973018155106]	0.648624779021752	[0.4842738329423143, 0.4842738329423201]
1000	0.4993668683248717	[0.4197897170021226, 0.5789440222017181]	0.49874438534959414	[0.4993622186411384, 0.49936221864119407]
10000	0.4993633810753954	[0.49140566577020883, 0.5073210964055008]	0.4993633810764549	[0.4993633810761791, 0.4993633810767338]

Podemos observar na tabela, que para todas subdivisões testados, os intervalos solução pelo método dos Trapézios encapsularam os valores reais. Já para os intervalos obtidos pelo método de Simpson, o valor real está contido apenas para o subintervalo 10000.

A Tabela 20 contém os resultados obtidos para o cálculo da função $f12$ com entradas reais e intervalares, utilizando o método dos Trapézios e de Simpson.

Tabela 20 – Resultados obtidos no exemplo f12.

f12				
n	Trapézio Real	Trapézio Intervalar	Simpson Real	Simpson Intervalar
10	1.5610797449331955	[1.4961251166251206, 1.6728038963200205]	1.5643956358449842	[1.5643964326874134, 1.5643964326874151]
100	1.564363276776222	[1.5547412930785713, 1.574391180170637]	1.564396444080993	[1.5643964440698492, 1.5643964440698623]
1000	1.5643961123961527	[1.5634065588854837, 1.565389653901808]	1.5643964440690479	[1.5643964440689877, 1.5643964440691156]
10000	1.5643964407523498	[1.5642972162556046, 1.5644957050577728]	1.5643964440690812	[1.564396444068334, 1.5643964440696023]

É possível perceber na tabela, que os resultados intervalares obtidos com o método dos Trapézios encapsulam o valores reais para todos os subintervalos testados. O resultados intervalares obtidos com o método de Simpson para 10 e 100 subintervalos não encapsulam os valores reais.

A Tabela 21, apresenta os resultados obtidos no cálculo da função $f13$, utilizando os métodos dos Trapézios e de Simpson.

Tabela 21 – Resultados obtidos no exemplo f13.

f13				
n	Trapézio Real	Trapézio Intervalar	Simpson Real	Simpson Intervalar
10	0.0025732063562327392	[0.0006311543900482234, 0.004516974769787307]	0.0031078446328193145	[0.004335138673289401, 0.004335138673289421]
100	0.015309127743213507	[0.005413820910196639, 0.02520444648489766]	0.01775778187756945	[0.013019088261908951, 0.013019088261909142]
1000	0.01349248422805536	[0.012502952743292113, 0.014482015828210789]	0.01349248348300922	[0.013492485649474249, 0.01349248564947581]
10000	0.01349248563525495	[0.013392548779558828, 0.013592422492098516]	0.013492485649468904	[0.013492485649460168, 0.013492485649475373]

Como mostra a tabela, para todos subdivisões utilizadas os intervalos resultantes pelo método dos Trapézios encapsulam os valores reais. Nos intervalos obtidos pelo método de Simpson, apenas para o subintervalo 10000, o valor real está contido.

Observamos que o método dos Trapézios Intervalar encapsula o valor real em todos os casos apresentados. O método de Simpson Intervalar para as funções $f1$, $f5$, $f6$, $f7$ e $f9$ e $f12$, contém o valor real usando 1000 e 10000 subintervalos, para as funções $f2$, $f3$, $f11$ e $f13$ o método encapsula o valor real apenas com 1000 subintervalos, e não encapsula os valores reais para a funções $f4$, $f8$ e $f10$, com as subdivisões utilizadas.

A seguir são apresentadas as tabelas com os cálculos de diâmetro e erros absoluto e relativo para cada função. Para os casos em que o intervalo encontrado pelo método não encapsula o valor real, não foram calculados os erros absoluto, relativo e o diâmetro.

A tabela 22 apresenta os valores obtidos para estimativa de erros dos intervalos resultantes da aplicação a função de exemplo $f1$.

Tabela 22 – Diâmetro, erros absoluto e relativo da função f1.

f1	n	Diâmetro	Erro Absoluto	Erro Relativo
Trapézio Intervalar	10	1.45960311111569618e-01	0.02320538206806 < 0.072980155557848	0.013504991837614076 <= 0.044991259748435895
	100	1.691234472350467e-02	0.000254808941274 < 0.008456172361752	0.000148292868523598 <= 0.004946371410006038
	1000	1.7155649054336752e-03	2.57237520573e-06 < 0.000857782452716	1.497062451065283e-06 <= 0.000499459489640151
	10000	1.7180100250246966e-04	2.57484735755e-08 < 8.59005012512e-05	1.498501185838383e-08 <= 4.999459091991824e-05
Simpson Intervalar	10	-	-	-
	100	-	-	-
	1000	1.1435297153639112e-13	2.88657986402e-15 < 5.71764857681e-14	1.679922243380814e-15 <= 3.327538289773641e-14
	10000	1.142197447734361e-12	2.52020626589e-13 < 5.71098723867e-13	1.4667013432594032e-13 <= 3.3236615461365573e-13

Com os valores mostrados na tabela, observa-se que os valores de diâmetro apresentados pelo método dos Trapézios são maiores que os valores apresentados pelo método de Simpson. Para 10 e 100 subintervalos o método dos Trapézios apresenta diferença respectivamente a partir da 1ª e 2ª casa decimal, enquanto pelo método de Simpson a diferença está na 15ª e 14ª casa decimal, porém é possível observar que para esta quan-

tidade de subintervalos o método de Simpson não satisfaz os critérios de estimativa de erros absoluto e relativo, pois o valor real não está contido nos intervalos resultantes. Os demais intervalos solução obtidos respeitam os critério de estimativas de erros, sendo o intervalo resultante pelo método de Simpson com 1000 subintervalos, o que resultou no menor erro dentre eles.

A Tabela 23 mostra o diâmetro e os erros absoluto e relativo dos resultados retornados pelo cálculo da função de exemplo f_2 .

Tabela 23 – Diâmetro, erros absoluto e relativo da função f_2 .

f_2	n	Diâmetro	Erro Absoluto	Erro Relativo
Trapézio Intervalar	10	0.3146130638830362	3.828981047519499 < 1.573065319415181	0.1013442608574434 <= 0.5750886708995822
	100	0.0376120461631012	0.000625230623444 < 0.018806023081550	0.0013041172560132 <= 0.0408829279018161
	1000	0.0038296447274567	6.393176674834e-06 < 0.001914822363728	1.333500265303e-05 <= 0.0040100403114988
	10000	0.0003836558532860	6.407298908106e-08 < 0.0001918279266430	1.336445906065e-07 <= 0.0004002783444361
Simpson Intervalar	10	-	-	-
	100	-	-	-
	1000	-	-	-
	10000	3.6243230638888e-13	1.349476086431e-13 < 1.812161531944e-13	2.814761441461e-13 <= 3.779839047987e-13

Na tabela podemos observar que o diâmetro dos intervalos encontrados pelo método de Simpson possuem uma melhor qualidade de intervalo, já que o método dos Trapézios possui um maior comprimento de intervalo. É possível notar que nos intervalos resultantes pelo método de Simpson com 10, 100 e 1000 subintervalos, não satisfazem os critérios de erro absoluto e erro relativo, isso acontece pois o valor real não está contido no intervalo.

A Tabela 24, contém o diâmetro dos intervalos e os erros absoluto e relativo para a função f_3

Tabela 24 – Diâmetro, erros absoluto e relativo da função f_3 .

f_3	N	Diâmetro	Erro Absoluto	Erro Relativo
Trapézio Intervalar	10	0.2728937815321597	0,032726756680059976 < 0,133644689076607985	0,02068390523410394 <= 0,09228657880430154
	100	0.030360804594373292	0,000248152984425376 < 0,015180402297186646	0,00015683719788040 <= 0,00968569857561397
	1000	0.0030622752029914047	2.38842636690606e-06 < 0.001531137601495702	1.5095288883856e-06 <= 0.00096864270202960
	10000	0.0003064848685006005	2.37913932998878e-08 < 0,000153242434250300	1.5036593122043e-08 <= 9.6861384343107e-05
Simpson Intervalar	10	-	-	-
	100	-	-	-
	1000	-	-	-
	10000	1.2678746941219288e-12	5.50670620214077e-14 < 6.33937347060964e-13	3.4803384383803e-14 <= 4.00659929095972e-13

Analisando a tabela, é possível observar que o método dos Trapézios Intervalar satisfaz os critérios de estimativas de erro para os subintervalos testados. O método de Simpson Intervalar resulta em um comprimento menor de intervalo do que o método de Trapézios, porém para os subintervalos 10, 100 e 1000 os critérios de erro absoluto e erro relativo não foram satisfeitos. O intervalo com melhor qualidade e menores estimativas de erro para a função f_3 foi utilizando o método de Simpson com 1000 subintervalos.

A [Tabela 25](#), apresenta as estimativas de erro para a função f_4 dos exemplos.

Tabela 25 – Diâmetro, erros absoluto e relativo da função f_4 .

f_4	n	Diâmetro	Erro Absoluto	Erro Relativo
Trapézio Intervalar	10	0.03961717287603517	0.0096368890826243 < 0.0198085864380175	0.0111155586429431 <= 0.0231192304091600
	100	0.004899510174978361	9.262518431196e-05 < 0.0024497550874891	0.0001068374512984 <= 0.0028333450820018
	1000	0.000498999501076347	9.176626443174e-07 < 0.0002494997505381	1.058467400620e-06 <= 0.0002878651495519
	10000	4.999000024030753e-05	9.167696934930e-09 < 2.499500012015e-05	1.0574374367833e-08 <= 2.8831025709375e-05
Simpson Intervalar	10	-	-	-
	100	-	-	-
	1000	-	-	-
	10000	-	-	-

Como mostra a tabela, o método dos Trapézios satisfaz os critérios de estimativas de erro no subintervalos testado. O método de Simpson possui um menor comprimento de intervalo, mas não satisfaz os critérios de erro absoluto e erro relativo. Para a função f_4 , o menor erro obtido foi utilizando o método dos Trapézios com 10000 subintervalos.

A Tabela 26 mostra as estimativas de erro para a função f_5 .

Tabela 26 – Diâmetro, erros absoluto e relativo da função f_5 .

f_5	n	Diâmetro	Erro Absoluto	Erro Relativo
Trapézio Intervalar	10	9.1038288019262e-15	0.1547005383792518 < 4.551914400963e-15	0.1339745962155616 <= 4.5519144009631e-15
	100	0.1319950279197073	0.0015828727033817 < 0.0659975139598536	0.0013708079720855 <= 0.0605323008708251
	1000	0.0133178707988550	1.5709254239659e-05 < 0.00665893539942752	1.3604613246053e-05 <= 0.00580017681039769
	10000	0.0013331765012385	1.5707965750344e-07 < 0.00066658825061927	1.3603497381574e-07 <= 0.00057761572769439
Simpson Intervalar	10	-	-	-
	100	-	-	-
	1000	9.015010959956e-14	1.77635683940025e-15 < 4.50750547997813e-14	1.53837014910685e-15 <= 3.90361425335878e-14
	10000	8.868461520705e-13	9.19264664389629e-14 < 4.43423076035287e-13	7.96106552162795e-14 <= 3.84015648470975e-13

Analisando a tabela, observa-se que o método dos Trapézios não obedece os critérios de erro absoluto e erro relativo, pois o resultado intervalar não contém o valor exato da função. Para os outros subintervalos os critérios de estimativa são satisfeitos pelo método dos Trapézios. O método de Simpson não satisfaz os critérios de erro absoluto e relativo utilizando 10 e 100 subintervalos, pois o intervalo solução não contém o valor exato. O intervalos com melhor qualidade para a função f_5 foi utilizando o método de Simpson com 1000 subintervalos.

A tabela Tabela 27, apresenta as estimativas de erro para a função f_6 .

Tabela 27 – Diâmetro, erros absoluto e relativo da função f_6 .

f_6	n	Diâmetro	Erro Absoluto	Erro Relativo
Trapézio Intervalar	10	0.04736842105263217	0.003402000393260551 < 0.023684210526316085	0.004908049096459301 <= 0.035199054144979446
	100	0.00497487437186561	3.150244713134853e-05 < 0.002487437185932806	4.54484242522632e-05 <= 0.003601373552918801
	1000	0.00049974987500267	3.127502432587192e-07 < 0.000249874937501337	4.51203224986171e-07 <= 0.000360623172710831
	10000	4.9997500486620e-05	3.125267933334896e-09 < 2.499875024331022e-05	4.50880854887155e-09 <= 3.60668736142901e-05
Simpson Intervalar	10	-	-	-
	100	-	-	-
	1000	6.4837024638109e-14	9.992007221626409e-16 < 3.241851231905457e-14	1.44154192671671e-15 <= 4.67700269556999e-14
	10000	6.4714900105400e-13	4.451994328746878e-14 < 3.235745005270018e-13	6.42287014014891e-14 <= 4.66819327268676e-13

Na tabela, observa-se que todos intervalos retornados pelo método dos Trapézios, nos subintervalos testados, obedecem os critério de erro absoluto e relativo, pois o valor exato da função está contido nos intervalos. Os intervalos com 10 e 100 subintervalos, retornados pelo método de Simpson não contém o valor exato da função. O intervalo

com menores erros para a função $f6$ foi obtido com o método de Simpson com 1000 subintervalos.

A tabela [Tabela 28](#), mostra as estimativas de erro para a função $f7$.

Tabela 28 – Diâmetro, erros absoluto e relativo da função $f7$.

f7	n	Diâmetro	Erro Absoluto	Erro Relativo
Trapézio Intervalar	10	0.0210949502625008	0.00209870739550072 < 0.0105474751312504	0.005524578942713723 <= 0.028396425050758962
	100	0.00229087922349668	2.01967132466562e-05 < 0.00114543961174834	5.31652658935257e-05 <= 0.003024181157645816
	1000	0.00023086192130850	2.01151773349117e-07 < 0.000115430960654250	5.295063302852293e-07 <= 0.0003039494447817738
	10000	2.3103892015319e-05	2.01066713279018e-09 < 1.15519460076596e-05	5.292824205238497e-09 <= 3.0409945765678802e-05
Simpson Intervalar	10	-	-	-
	100	-	-	-
	1000	3.4527936065842e-14	3.88578058618804e-16 < 1.72639680329211e-14	1.022882067718568e-15 <= 4.544518900864126e-14
	10000	3.4405811533133e-13	2.59237076249974e-14 < 1.72029057665668e-13	6.82408465177959e-14 <= 4.528445039802134e-13

Na tabela, é possível notar que o método dos Trapézios atende os critérios das estimativas de erros, os intervalos obtidos encapsulam o valor exato da função. Os intervalos resultantes pelo método de Simpson com 10 e 100 subintervalos não obedecem aos critérios de erro absoluto e erro relativo. O intervalo solução com melhor qualidade foi obtido pelo método de Simpson utilizando 1000 subintervalos.

A [Tabela 29](#), apresenta as estimativas de erro para a função $f8$.

Tabela 29 – Diâmetro, erros absoluto e relativo da função $f8$.

f8	n	Diâmetro	Erro Absoluto	Erro Relativo
Trapézio Intervalar	10	8.0675269640544e-14	0.009098637539165252 < 4.03376348202723e-14	0.9999999999998251 <= 1.041055581726859
	100	0.9357000525192805	0.004802795937753623 < 0.46785002625964023	0.5278588049121707 <= 1.0092671833058318
	1000	0.1330601049785881	0.000167846597580612 < 0.06653005248929408	0.0184474430218903 <= 1.1550504416482688
	10000	0.01335658283252138	1.75427814294928e-06 < 0.00667829141626069	0.0001928066851105 <= 2.761231263853263
Simpson Intervalar	10	-	-	-
	100	-	-	-
	1000	-	-	-
	10000	-	-	-

Analisando a tabela, observa-se que o método dos Trapézios utilizando 10 subintervalos não atende ao critério de erro absoluto, o qual deve ser menor que metade do diâmetro do intervalo. Os intervalos retornados pelo método de Simpson não satisfazem os critérios de erro absoluto e erro relativo. É possível que o método de Simpson Intervalar convirja utilizando mais subintervalos. O intervalo com melhor qualidade foi retornado pelo método dos Trapézios, utilizando 1000 subintervalos.

A tabela [Tabela 30](#), apresenta as estimativas de erro para a função f_9 .

Tabela 30 – Diâmetro, erros absoluto e relativo da função f_9 .

f_9	n	Diâmetro	Erro Absoluto	Erro Relativo
Trapézio Intervalar	10	7.071067811865477	$3.035533905932739 <$ 3.5355339059327386	$6.071067811865478 <=$ $8.275375255476e+67$
	100	0.707106781186548	$0.00186744274386929 <$ 0.35355339059327423	$0.0037348854877385 <=$ 2.3838158655006167
	1000	0.070710678118668	$4.71844785465691e-15 <$ 0.035355339059334384	$9.4368957093138e-15 <=$ 0.07609113378760972
	10000	0.007071067811984	$2.28816965375244e-13 <$ 0.003535533905992038	$4.57633930750489e-13 <=$ 0.00712142388318320
Simpson Intervalar	10	-	-	-
	100	-	-	-
	1000	$2.187139358511e-14$	$9.65894031423886e-15 <$ $1.09356967925577e-14$	$1.931788062847772e-14 <=$ $2.187139358511564e-14$
	10000	$2.159383782895e-13$	$9.59232693276135e-14 <$ $1.07969189144796e-13$	$1.918465386552270e-13 <=$ $2.159383782895981e-13$

Na tabela, vemos que todos os intervalos solução do método dos Trapézios obedecem aos critérios de erro absoluto e relativo. Os intervalos resultantes pelo método de Simpson para 10 e 100 subintervalo não obedecem aos critério de erros. Para a função f_9 , o intervalo com melhor qualidade foi obtido pelo método de Simpson com 1000 subintervalos.

A [Tabela 31](#), mostra as estimativas de erro para a função f_{10} .

Tabela 31 – Diâmetro, erros absoluto e relativo da função f_{10} .

f_{10}	n	Diâmetro	Erro Absoluto	Erro Relativo
Trapézio Intervalar	10	25.0000000000000036	$11.500000000347216 <$ 12.5000000000000018	$11.500000000347216 <=$ 36002449668.19295
	100	2.50000000000000253	$0.4735637245846416 <$ 1.2500000000000127	$0.4735637245846416 <=$ 5.591246980351826
	1000	0.2500000000001368	$0.0052029160469886 <$ 0.1250000000000684	$0.0052029160469886 <=$ 0.1420127083439532
	10000	0.02500000000112157	$5.208279074042e-05 <$ 0.0125000000005607	$5.2082790740426e-05 <=$ 0.0126575602627903
Simpson Intervalar	10	-	-	-
	100	-	-	-
	1000	-	-	-
	10000	-	-	-

Analisando a tabela, observa-se o método dos Trapézios satisfaz os critério de erro absoluto e reativo nos casos testados. Os intervalos obtidos pelo método de Simpson, não satisfazem os critério de erro absoluto e erro relativo. Para a função f_{10} , o intervalo com melhor qualidade foi obitido pelo método dos Trapézios com 10000 subintervalos.

A [Tabela 32](#), apresenta as estimativas de erros para a função f_{11} .

Tabela 32 – Diâmetro, erros absoluto e relativo da função f_{11} .

f11	n	Diâmetro	Erro Absoluto	Erro Relativo
Trapézio Intervalar	10	15.91541571454403	7.468233220859281 < 7.957707857272015	14.955508360986189 <= 804.7237670586937
	100	1.5915429354881523	0.3978381693345731 < 0.7957714677440761	0.7966907154404675 <= 7.845517294479547
	1000	0.1591543051995955	3.4885254636685e-06 < 0.07957715259979778	6.9859456977976e-06 <= 0.18956432084160701
	10000	0.0159154306352919	1.1398104682314e-11 < 0.0079577153176459	2.2825271363999e-11 <= 0.016193780153457134
Simpson Intervalar	10	-	-	-
	100	-	-	-
	1000	-	-	-
	10000	5.546674231027e-13	2.77555756156289e-16 < 2.773337115513641e-13	5.55819202357156e-16 <= 5.55374546995579e-13

Como mostrado na tabela, os intervalos solução obtidos pelo método dos Trapézios contém os resultados do método real para todos os subintervalos testados, contendo o valor exato da função. Nos intervalos obtidos pelo método de Simpson, apenas o intervalo solução para o subintervalo 10000 contém o valor exato para a função f_{11} , para os demais subintervalos o valor exato não está contido. O intervalo com melhor qualidade para a função f_{11} foi obtito utilizando o método de Simpson com 10000 subintervalos.

A [Tabela 33](#), apresenta as estimativas de erro para a função f_{12} dos exemplos utilizados.

Tabela 33 – Diâmetro, erros absoluto e relativo da função f_{12} .

f12	n	Diâmetro	Erro Absoluto	Erro Relativo
Trapézio Intervalar	10	0.17667877969489987	0.020068062403521214 < 0.08833938984744993	0.012827990295940256 <= 0.059045456069022624
	100	0.01964988709206583	0.00016979255555714 < 0.009824943546032916	0.000108535503388180 <= 0.006319343024959709
	1000	0.00198309501632421	1.662324597306863e-06 < 0.000991547508162105	1.06259804131432e-06 <= 0.000634222430836516
	10000	0.00019848880216821	1.658746651145293e-08 < 9.92444010841087e-05	1.06031093169122e-08 <= 6.344344287825109e-05
Simpson Intervalar	10	-	-	-
	100	-	-	-
	1000	1.27897692436818e-13	1.776356839400250e-15 < 6.394884621840902e-14	1.135490205270400e-15 <= 4.087764738973605e-14
	10000	1.2683187833317788e-12	8.171241461241152e-14 < 6.341593916658894e-13	5.223254944243843e-14 <= 4.053700032817186e-13

Podemos observar que os intervalos resultantes pelo método dos Trapézios satisfazem aos critérios de erro absoluto e erro relativo, encapsulando o valor exato. O método de Simpson não obedece os critérios de estimativas de erro com 10 e 100 subintervalos. O intervalo com melhor qualidade para a função f_{12} deu-se pelo método de Simpson com 1000 subintervalos.

A Tabela 34, mostra as estimativas de erro para a função f_{13} .

Tabela 34 – Diâmetro, erros absoluto e relativo da função f_{13} .

f13	n	Diâmetro	Erro Absoluto	Erro Relativo
Trapézio Intervalar	10	0.003885820379739083	0.010918421069550007 < 0.001942910189869541	0.8092223592604448 <= 3.07834377848674
	100	0.019790625574701022	0.001816648048079376 < 0.009895312787350511	0.1346414660186083 <= 1.8277872414865486
	1000	0.001979063084918676	1.36371632215859e-09 < 0.000989531542459338	1.0107228257177e-07 <= 0.07914382808414802
	10000	0.000199873712539688	1.36391002658609e-11 < 9.99368562698442e-05	1.0108663903895e-09 <= 0.007462123746181815
Simpson Intervalar	10	-	-	-
	100	-	-	-
	1000	-	-	-
	10000	1.52048512669367e-14	1.734723475976807e-18 < 7.602425633468357e-15	1.2856959948260e-16 <= 5.634562697328144e-13

Como mostra a tabela, o método dos Trapézios satisfaz os critérios de estimativas de erro, encapsulando o valor exato da função em todos os subintervalos testados, contendo um comprimento de intervalo grande. O método de Simpson não satisfaz os critérios de erro absoluto e erro relativo para 10, 100 e 1000 subintervalos. O método de Simpson possui um comprimento de intervalo menor em relação ao método dos Trapézios. O intervalo com melhor qualidade para a função f_{13} foi obtido pelo método de Simpson com 10000 subintervalos.

Nas tabelas acima, foi possível observar que o método dos Trapézios contém um comprimento de intervalo maior que o método de Simpson. O método dos Trapézios Intervalar não encapsulou o valor exato nas funções f_5 e f_8 com 10 subintervalos. O método de Simpson Intervalar não satisfaz os critérios de erro relativo e erro absoluto utilizando 10 e 100 subintervalos, sendo necessário um número maior de subintervalos para conter o valor exato da função. Para as funções f_4 , f_8 e f_{10} o método de Simpson não encapsulou o valor exato para os casos de subintervalos testados.

A seguir serão apresentados os tempos de processamento para os métodos de integração testados no presente trabalho. O tempos foram obtidos através da média de 10 execuções pra cada caso.

A [Tabela 35](#), mostramos o tempo de processamento para a função $f1$.

Tabela 35 – Tempo de execução em segundos para a função $f1$.

f1				
n	Trapézio Real	Trapézio Intervalar	Simpson Real	Simpson Intervalar
10	0.000007	0.000567	0.000008	0.001271
100	0.000044	0.004720	0.000054	0.013125
1000	0.000420	0.050853	0.000534	0.132028
10000	0.004171	1.048503	0.005139	1.267505

A [Tabela 36](#), mostra o tempo de processamento para a função $f2$.

Tabela 36 – Tempo de execução em segundos para a função $f2$

f2				
n	Trapézio Real	Trapézio Intervalar	Simpson Real	Simpson Intervalar
10	0.000012	0.001269	0.000013	0.002674
100	0.000078	0.011360	0.000089	0.026515
1000	0.000848	0.111728	0.000837	0.265066
10000	0.007272	1.731262	0.008251	2.630413

A [Tabela 37](#), apresenta o tempo de processamento para a função $f3$.

Tabela 37 – Tempo de execução em segundos para a função $f3$.

f3				
n	Trapézio Real	Trapézio Intervalar	Simpson Real	Simpson Intervalar
10	0.000011	0.001368	0.000013	0.003021
100	0.000090	0.011505	0.000102	0.028631
1000	0.000830	0.126366	0.000933	0.129961
10000	0.007999	1.780067	0.009262	2.978886

A [Tabela 38](#), apresenta o tempo de processamento para a função $f4$.

Tabela 38 – Tempo de execução em segundos para a função $f4$.

f4				
n	Trapézio Real	Trapézio Intervalar	Simpson Real	Simpson Intervalar
10	0.000009	0.000915	0.000010	0.003072
100	0.000552	0.009222	0.000460	0.026570
1000	0.000659	0.088244	0.000755	0.263899
10000	0.006792	1.433183	0.007646	2.625534

A [Tabela 39](#), apresenta o tempo de processamento para a função $f5$.

Tabela 39 – Tempo de execução em segundos para a função $f5$.

f5				
n	Trapézio Real	Trapézio Intervalar	Simpson Real	Simpson Intervalar
10	0.000190	0.014243	0.000202	0.038297
100	0.000913	0.126977	0.001277	0.406838
1000	0.008624	1.306816	0.009217	3.890229
10000	0.079715	1.860170	0.090570	3.945204

A [Tabela 40](#), apresenta o tempo de processamento para a função $f6$.

Tabela 40 – Tempo de execução em segundos para a função $f6$.

f6				
n	Trapézio Real	Trapézio Intervalar	Simpson Real	Simpson Intervalar
10	0.000007	0.007677	0.000008	0.021089
100	0.000429	0.072454	0.000533	0.211335
1000	0.004125	0.780664	0.005225	2.075485
10000	0.040745	1.301292	0.051620	2.100994

A [Tabela 41](#), apresenta o tempo de processamento para a função $f7$.

Tabela 41 – Tempo de execução em segundos para a função $f7$.

f7				
n	Trapézio Real	Trapézio Intervalar	Simpson Real	Simpson Intervalar
10	0.000003	0.000979	0.000002	0.002460
100	0.000008	0.008246	0.000100	0.024421
1000	0.000643	0.083532	0.000732	0.236376
10000	0.006124	1.446521	0.007165	2.352012

A [Tabela 42](#), apresenta o tempo de processamento para a função $f8$.

Tabela 42 – Tempo de execução em segundos para a função $f8$.

f8				
n	Trapézio Real	Trapézio Intervalar	Simpson Real	Simpson Intervalar
10	0.000002	0.001541	0.000004	0.003747
100	0.000008	0.012361	0.000159	0.038022
1000	0.000617	0.129088	0.000832	0.428644
10000	0.005967	1.755685	0.007347	3.996024

A [Tabela 43](#), apresenta o tempo de processamento para a função f_9 .

Tabela 43 – Tempo de execução em segundos para a função f_9 .

f9				
n	Trapézio Real	Trapézio Intervalar	Simpson Real	Simpson Intervalar
10	0.000003	0.001498	0.000004	0.004189
100	0.000137	0.013288	0.000212	0.042201
1000	0.001126	0.151615	0.001222	0.407332
10000	0.011069	1.928587	0.012470	4.001552

A [Tabela 44](#), apresenta o tempo de processamento para a função f_{10} .

Tabela 44 – Tempo de execução em segundos para a função f_{10} .

f10				
n	Trapézio Real	Trapézio Intervalar	Simpson Real	Simpson Intervalar
10	0.000002	0.001101	0.000003	0.002840
100	0.000115	0.009663	0.000102	0.027997
1000	0.000604	0.095844	0.000703	0.277265
10000	0.005909	1.495558	0.006836	2.772321

A [Tabela 45](#), apresenta o tempo de processamento para a função f_{11} .

Tabela 45 – Tempo de execução em segundos para a função f_{11} .

f11				
n	Trapézio Real	Trapézio Intervalar	Simpson Real	Simpson Intervalar
10	0.000002	0.001528	0.000002	0.004209
100	0.000135	0.014323	0.000149	0.041625
1000	0.000834	0.135899	0.000910	0.409657
10000	0.007887	1.877128	0.008867	4.060335

A [Tabela 46](#), apresenta o tempo de processamento para a função f_{12} .

Tabela 46 – Tempo de execução em segundos para a função f_{12} .

f12				
n	Trapézio Real	Trapézio Intervalar	Simpson Real	Simpson Intervalar
10	0.0000002	0.000994	0.0000002	0.002626
100	0.0001189	0.009543	0.0001051	0.027853
1000	0.0005278	0.087126	0.0006599	0.260475
10000	0.0051660	1.392478	0.0064132	2.574657

A [Tabela 47](#), apresenta o tempo de processamento para a função f_{13} .

Tabela 47 – Tempo de execução em segundos para a função f_{13} .

f13				
n	Trapézio Real	Trapézio Intervalar	Simpson Real	Simpson Intervalar
10	0.000002	0.001399	0.0000003	0.003823
100	0.000009	0.013032	0.0001089	0.037389
1000	0.000777	0.123301	0.0008661	0.366049
10000	0.007376	1.758221	0.0084440	3.646942

Nas tabelas acima, foram apresentados os tempos de execução para todas as funções de exemplo. Foi possível observar que para todas as funções seguiu-se o padrão de aumento de tempo conforme aumenta do número de subintervalos utilizados para realizar o cálculo da função, o que era o esperado. Os métodos com entradas reais obtiveram tempos de execução parecidos, percebendo-se pouca diferença entre o tempo de execução e sendo o método dos Trapézios real um pouco melhor. Para os métodos nas formas intervalares, para todas as funções calculadas, o método dos Trapézios Intervalar obteve melhor tempo de processamento.

Com todos os resultados analisados pelo método de Simpson e dos Trapézios, realizou-se uma análise quanto ao tipo de função que cada integral estava resolvendo nos 13 exemplos e qual o melhor método que a resolve de forma intervalar. Assim, na [Tabela 48](#), apresenta-se um resumo dessa análise.

Tabela 48 – Classificação por tipo de função

Tipo de função	Exemplos	Melhor método
Exponencial	f1	Simpson Intervalar
	f7	Simpson Intervalar
	f9	Simpson Intervalar
	f10	Trapézio Intervalar
Polinomial	f3	Simpson Intervalar
	f4	Trapézio Intervalar
	f6	Simpson Intervalar
	f11	Simpson Intervalar
	f12	Simpson Intervalar
	f13	Simpson Intervalar
Trigonométrica	f2	Simpson Intervalar
	f5	Simpson Intervalar
	f8	Trapézio Intervalar

O método dos Trapézios Intervalar possui um maior comprimento de intervalo, possuindo menor precisão que o método de Simpson Intervalar. O método de Simpson Intervalar resulta intervalos mais estreitos, mas necessita de um número de subdivisões maior que o método dos Trapézios para encapsular o valor exato, necessitando um maior tempo de processamento. O principal mecanismo para obter um intervalo estreito é aumentar o número de subintervalos, conforme o aumento de subdivisões se torna muito grande, acontece um arredondamento significativo. O método dos Trapézios Intervalar pôde ser aplicado para calcular todas as funções utilizadas de exemplo, encapsulando o valor obtido pelo método real e o valor exato para cada uma das funções.

Neste capítulo foram apresentados os resultados obtidos da computação de exemplos aplicando os métodos de integração dos Trapézios e Simpson nas formas reais e intervalares. Realizou-se uma análise numérica a fim de justificar o uso do método dos Trapézios Intervalar, quanto a qualidade dos intervalos resultantes, e as métricas de erro absoluto e relativo e ao tempo de processamento demandado. O método dos Trapézios Intervalar, mostrou possuir um comprimento de intervalo maior que o método de Simpson Intervalar, sendo menos preciso, mas possui um menor erro contido, além de convergir com menos subdivisões e demandar menos tempo de processamento.

6 CONSIDERAÇÕES FINAIS

Visto que o cálculo numérico de integrais é realizado através de aproximações, seu resultado é afetado por erros de arredondamento e truncamento, que podem ser propagados devido às operações aritméticas realizadas no computador. Quando não é possível encontrar a primitiva da função na forma analítica, não podemos usar outro método que não o numérico. Um dos fatores de maior importância quando se trabalha com computação numérica é a exatidão dos resultados. A matemática intervalar, surge com o objetivo de realizar um controle automático de erros dos cálculos, retornando respostas com maior exatidão possível.

Na literatura é possível encontrar funções com integração que são resolvidas aplicando o método dos trapézios.

Neste trabalho, foi realizada a definição intervalar do método de integração dos Trapézios, utilizando o método de extensão intervalar. O método definido foi implementado em um ambiente de programação com suporte ao tipo intervalo, utilizando a linguagem Python e o pacote Intpy. Para que fosse possível comparar os resultados intervalares aos reais, a forma real do método também foi implementada utilizando a linguagem Python. A fim de validar os resultados obtidos pelo método dos Trapézios Intervalar, foram utilizadas algumas funções, as quais possuem seu valor exato conhecido. Foi realizada uma análise numérica utilizando as métricas de erro estabelecidas na aritmética intervalar, comparando o método dos Trapézios ao método de Simpson, por este já possuir uma definição intervalar. Por fim foi verificado o tempo de processamento demandado por ambos os métodos.

Verificou-se que para todas as funções computadas, os resultados reais apresentados pelo método dos Trapézios Real estão contidos nos valores intervalares apresentados pelo método dos Trapézios Intervalar.

Como resultados da análise numérica, foi possível concluir que, apesar de na maioria das vezes o método de Simpson retornar um intervalo solução com diâmetro menor, o método dos Trapézios sempre converge para uma resposta com um menor número de subdivisões. Em alguns casos isso é preferido ao invés de um diâmetro menor, uma vez que influencia diretamente no esforço computacional de resolver o problema, já que quanto maior o número de divisões, maior será o tempo de processamento e o esforço para computar o resultado. Se pudermos resolver um problema com uma ordem exponencial de 2^{10} , isso é melhor do que ter que resolver o mesmo problema com ordem 2^{10000} .

O principal objetivo deste trabalho foi utilizar a matemática intervalar para realizar a definição intervalar do método dos Trapézios composto, por este não possuir definição na literatura. A partir da implementação do método dos Trapézios Intervalar, foi possível realizar uma análise de qualidade para os valores intervalares obtidos, além do tempo de processamento do método.

A definição intervalar do método dos Trapézios pode ser adicionada aos demais

métodos de integração intervalar existentes na literatura, visto que a integração numérica tem seus resultados afetados por erros gerados pela aritmética de ponto flutuante.

6.1 Trabalhos futuros

A partir da definição intervalar e das comparações feitas no presente trabalho, ficam alguns itens como trabalhos futuros:

- Aplicar os métodos em outros tipos de funções;
- Automatizar a escolha do método a ser utilizado em cada função, possibilitando que o usuário entre com a função a ser integrada e o algoritmo já resolve com o melhor método;
- Testar essa definição com a aritmética RDM, a fim de verificar se o resultado solução pode melhorar.

REFERÊNCIAS

- BALBONI, M. D. C. et al. Comparando diferentes métodos de integração intervalar. **Vetor**, 2015. Citado 2 vezes nas páginas 25 e 26.
- BALBONI, M. D. C. et al. Critérios para análise e escolha de ambientes intervalares. **Revista Jr de Iniciação Científica em Ciências Exatas e Engenharia**, Rio Grande - RS, n. 7, 2014. Citado na página 28.
- BARRETO, R. M. Intpy 0.1.3. 2016. Disponível em: <<https://pypi.python.org/pypi/IntPy/0.1.3>>, 2016. Citado 3 vezes nas páginas 14, 20 e 35.
- BARRETO, R. M.; CAMPOS, M. A. Intpy: Um framework intervalar em python. **VIII ERMAC-R3**, 2008. Citado 3 vezes nas páginas 14, 17 e 21.
- BURDEN, R. L.; FAIRES, J. D. **Análise Numérica**. [S.l.: s.n.], 2008. Citado na página 32.
- CAPRANI, O.; MADSEN, K.; NIELSEN, H. B. Introduction to interval analysis. **IMM - Informatics and Mathematical Modelling**, 2002. Citado 5 vezes nas páginas 13, 14, 25, 31 e 34.
- CARVALHO, N. F. de; MARCHI, C. H. Avaliação de esquemas numéricos para problemas difusivos 2d resolvidos com volumes finitos. **CILAMCE**, 2009. Citado na página 23.
- CLÁUDIO, D. M.; MARINS, J. M. **Cálculo Numérico Computacional: Teoria e Prática**. [S.l.: s.n.], 1994. Citado na página 32.
- DUARTE, G. C. P. **Introdução à matemática e à administração financeira intervalar**. Natal: [s.n.], 2007. Citado na página 19.
- FINGER, A. F. **Extensão intervalar para as variáveis aleatórias com distribuições Uniforme, Normal, Gama, Exponencial e Pareto**. mathesis, 2014. Citado 2 vezes nas páginas 24 e 31.
- GALDINO, S. Interval integration revisited. **Open Journal of Applied Sciences**, 2012. Citado 2 vezes nas páginas 27 e 36.
- KREINOVICH, V. Probabilities. intervals, what next? optimization problems related to extension of interval computations to situation with partial information about probabilities. **Global Optimization**, v. 29, n. 3, p. 265–280, 2003. Citado na página 18.
- KULISCH, U.; MIRANKER, L. **Computer Arithmetic in Theory and Practice**. 1th. ed. [S.l.]: Academic Press, 1981. Citado 2 vezes nas páginas 18 e 21.
- KULISCH, U. W. Complete interval arithmetic and its implementation on the computer. In: **Numerical Validation in Current Hardware Architectures**. [S.l.]: Springer, 2008. v. 5492. Citado na página 21.
- LORETO, A. B. **Análise da Complexidade Computacional de Problemas de Estatística Descritiva com Entradas Intervalares**. Tese (Doutorado) — Universidade Federal do Rio Grande do Sul, Porto Alegre, 2006. Citado 2 vezes nas páginas 17 e 31.

- MESQUITA, M. Matemática intervalar: princípios e ferramenta c-xsc. **Online**, 2004. Citado na página 20.
- MONTEIRO, M. T. T. **Métodos Numéricos: exercícios resolvidos aplicados à Engenharia e outras Ciências**. [S.l.: s.n.], 2012. Citado na página 13.
- MOORE, R.; KEARFOTT, M.; CLOUD, J. **Introduction to Interval Analysis**. Philadelphia: Studies in Applied and Numerical Mathematics (SIAM), 2009. 213 p. Citado na página 19.
- MOORE, R. E. **Interval Analysis**. Englewood Cliffs, NJ: Prentice Hall, 1966. 159 p. Citado 2 vezes nas páginas 17 e 19.
- MOORE, R. E. **Methods and Applications of Interval Analysis**. 2. ed. Madison, Wisconsin: Studies in Applied and Numerical Mathematics (SIAM), 1979. 200 p. Citado na página 19.
- MOORE, R. E.; STROTHER, W.; T.YANG, C. Interval integrals. 1960. Citado na página 13.
- NÓBREGA, B. S. da. **Análise Numérica para a Solução de Integrais não Elementares**. mathesis, 2012. Citado 2 vezes nas páginas 13 e 24.
- NOBREGA G. A. S; BEDREGAL, R. C. B. B. R. C. Integrais de linha intervalar: Fundamentos e aplicações. **Universidade Federal do Rio Grande do Norte**, 2010. Citado na página 13.
- OLIVEIRA, P.; DIVERIO, T.; CLAUDIO, D. **Fundamentos de Matemática Intervalar**. Porto Alegre, Brasil: Sagra-Luzzato, 1997. 93 p. Citado na página 19.
- PYTHON. 2017. [Http://www.python.org](http://www.python.org). Citado 2 vezes nas páginas 14 e 20.
- RALL, L. B. Integration of interval functions ii. the finite case*. **Society for Industrial and Applied Mathematics**, 1982. Citado na página 13.
- RATSCHEK, H.; ROKNE, J. **New Computer Methods for Global Optimization**. Chichester, United Kingdom: Limited, 1988. 229 p. Citado 2 vezes nas páginas 13 e 17.
- RUGGIERO, M. A. G.; LOPES, V. L. R. **Cálculo numérico: aspectos teóricos e computacionais**. 2. ed. São Paulo: Pearson Makron Books, 1996. Citado 3 vezes nas páginas 13, 17 e 33.
- 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-Verlag, v. 18, p. 231–243, 2006. Citado na página 19.
- 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 35.
- VARJÃO, F. R. G. **IntPy: Computação Científica Auto Validável em Python**. Dissertação (Mestrado) — Universidade Federal de Pernambuco, Recife, 2011. Citado na página 21.

Apêndices

APÊNDICE A – IMPLEMENTAÇÕES DOS MÉTODOS

Todas as implementações descritas foram realizadas utilizando o ambiente de desenvolvimento intervalar composto pela linguagem de programação Python e o pacote de extensão intervalar IntPy. Estão demonstradas as implementações dos métodos para a função $f1$.

A.1 Trapézio Real

```

from scipy import integrate
from scipy import inf
from math import exp, cos, cosh, sin, sqrt
from intpy import*
from intpy.support.stdfunc *
import time
import timeit

def trapezio__real(n,a,b):
    e = 2.718281828459045
    h = (b - a)/float(n)
    soma = 0
    soma = (e**a) + (e**b)
    x = a + h
    for i in range(n-1):
        x = x + h
        soma = soma + 2 * (e**x)
    soma = (h/2) * soma
    return soma

```

A.2 Trapézio Intervalar

```

from scipy import integrate
from scipy import inf
from math import exp, cos, cosh, sin, sqrt
from intpy import*
from intpy.support.stdfunc *
import time
import timeit

def trapezio__intervalar(n,a,b):

```

```

e = IReal(2.718281828459045)
x = IReal(a,b)
h = (x.sup - x.inf)/float(n)
aux = x.inf + h
soma = IReal(0)
result = IReal(0)
listaInt = []
for i in range (n-1):
    listaInt = listaInt + [IReal(aux, aux + h)]
    aux = aux + h
for i in range (n-1):
    soma += IReal(2) * powI(e, listaInt[i])
soma += (powI(e, IReal(listaInt[i].inf))
soma += powI(e, IReal(listaInt[i].sup)))
soma = soma * ((listaInt[i].sup) - (listaInt[i].inf))/2)
return soma

```

A.3 Simpson Real

```

from scipy import integrate
from scipy import inf
from math import exp, cos, cosh, sin, sqrt
from intpy import*
from intpy.support.stdfunc *
import time
import timeit

def simpson__real(n,a,b):
    e = 2.718281828459045
    total = 0
    h = (b - a)/float(n)
    total = (e**a) + (e**b)
    x = a
    m = (a + b)/2
    for i in range(n-1):
        if i%2 == 0:
            x = x + h
            total = total + 4*(e**m)
        else:
            x = x + h

```

```

        total = total + 2 *(e**m)
    total = (h/6) * total
    return total

```

A.4 Simpson Intervalar

```

from scipy import integrate
from scipy import inf
from math import exp, cos, cosh, sin, sqrt
from intpy import*
from intpy.support.stdfunc *
import time
import timeit

def simpson__intervalar(n, a, b):
    listaInt = []
    e = 2.718281828459045
    total = 0
    x = IReal(a,b)
    h = (x.sup-x.inf)/float(n)
    aux = x.inf
    for i in range(n):
        listaInt += [ IReal(aux, aux+h) ]
        aux = aux+h
    for i in range(n):
        totalaux = IReal(0)
        totalaux += (powI(IReal((e),(listaInt[i].inf)))
        totalaux += powI(IReal((e), (listaInt[i].sup)))
        m = (listaInt[i].inf+listaInt[i].sup)/2
        totalaux += 4 * (e**m)
    totalaux = totalaux * ((listaInt[i].sup-listaInt[i].inf)/6)
    total = totalaux + total
    return total

```