



Universidade Federal de Pernambuco - UFPE
Centro de Ciências Exatas e da Natureza

Doutorado em Matemática Computacional

Probabilidades Autovalidáveis para as Variáveis
Aleatórias Exponencial, Normal e Uniforme

Maria das Graças dos Santos

Tese de Doutorado

Recife

27 de abril de 2010

Universidade Federal de Pernambuco - UFPE
Centro de Ciências Exatas e da Natureza

Maria das Graças dos Santos

Probabilidades Autovalidáveis para as Variáveis
Aleatórias Exponencial, Normal e Uniforme

*Trabalho apresentado ao Programa de Doutorado em
Matemática Computacional do Centro de Ciências Exatas
e da Natureza da Universidade Federal de Pernambuco -
UFPE como requisito parcial para obtenção do grau de
Doutor em Matemática Computacional.*

Orientadora: *Prof^a. Dr^a. Marcilia Andrade Campos*

Recife

27 de abril de 2010

Catálogo na fonte
Bibliotecária Jane Souto Maior, CRB4-571

Santos, Maria das Graças dos

**Probabilidades autovalidáveis para as variáveis aleatórias
exponencial, normal e uniforme - Recife: O**

Autor, 2010.

xiii, 114 p. : il., fig., tab.

Orientador: Marcilia Andrade Campos.

**Tese (doutorado) - Universidade Federal de Pernambuco.
CCEN, Matemática Computacional, 2010.**

Inclui bibliografia e apêndice.

**1. Análise Intervalar. 2. Computação Científica. 3. Aritmética
e Exatidão Máxima. 4. Probabilidade. I. Campos, Marcilia Andrade
Campos (orientadora). II. Título.**

511.42

CDD (22. ed.)

MEI2011 - 080

PARECER DA BANCA EXAMINADORA DE DEFESA DE TESE DE DOUTORADO

MARIA DAS GRAÇAS DOS SANTOS

*"PROBABILIDADES AUTOVALIDÁVEIS PARA AS VARIÁVEIS
ALEATÓRIAS EXPONENCIAL, NORMAL E UNIFORME"*

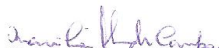
A Banca composta pelos Professores: MARCÍLIA ANDRADE CAMPOS, do Centro de Informática da UFPE; SÍLVIO DE BARROS MELO, do Centro de Informática da UFPE; LEANDRO CHAVES RÊGO, do Departamento de Estatística da UFPE; GRAÇALIZ PEREIRA DIMURO, da Pós-Graduação em Modelagem Computacional da FURG; e RAMÓN ORESTE MENDOZA AHUMADA, do Departamento de Matemática da UFPE, considera a Tese da candidata:

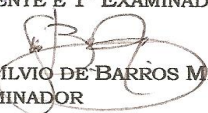
☒ APROVADA

☐ EM EXIGÊNCIA

☐ REPROVADA

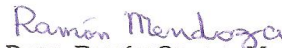
Secretaria do Programa de Pós-Graduação em Matemática Computacional do Centro de Ciências Exatas e da Natureza da Universidade Federal de Pernambuco, aos 27 dias do mês de abril de 2010.


PROF^a. MARCÍLIA ANDRADE CAMPOS
PRESIDENTE E 1º EXAMINADOR


PROF. SÍLVIO DE BARROS MELO
2º EXAMINADOR


PROF. LEANDRO CHAVES RÊGO
3º EXAMINADOR


PROF^a. GRAÇALIZ PEREIRA DIMURO
4º EXAMINADOR


PROF. RAMÓN ORESTE MENDOZA AHUMADA
5º EXAMINADOR

*À memória de meus Pais, Manoel Paulo e Maria José,
que com dedicação, amor e trabalho iluminaram e
encaminharam todos os passos da minha vida.*

Agradecimentos

Agradeço a Deus, que por sua presença, luz e força, sempre me abençoa e capacita para tudo aquilo que Ele me destina.

Aos meus Pais que morreram na certeza que eu continuaria minha luta, a minhas irmãs e irmão, sobrinhos em geral que tanto torceram por mim, acreditando na minha força.

Agradeço à minha orientadora, Professora Dr^a. Marcilia Andrade Campos, que sempre demonstrou acreditar no meu potencial, pela oportunidade oferecida, pela orientação, dedicação, paciência e principalmente pelo excelente convívio ao longo de todos estes anos.

Agradeço aos membros da banca examinadora da tese Professores Dr. Sílvio de Barros Melo, Dr^a Graçaliz Pereira Dimuro, Dr. Leandro Chaves Rêgo e Dr. Ramón Oreste Mendoza Ahumada por terem aceitado participar da avaliação deste trabalho e contribuíram com importantes e enriquecedoras sugestões.

À coordenação, professores e funcionários do Doutorado em Matemática Computacional da UFPE.

Às minhas grandes e inesquecíveis amigas Glória e Mirele, por fazerem parte, sem dúvida alguma, dos melhores momentos desta jornada.

A todos que, direta ou indiretamente, contribuíram para a realização deste trabalho.

Resumo

No estudo das variáveis aleatórias contínuas um dos problemas é o cálculo de probabilidades, visto que é necessário resolver uma integral definida da função densidade que, na maioria das vezes, não possui primitiva explícita ou cuja primitiva não é simples de obter. Embora integrais de funções densidade de probabilidade como a exponencial e a uniforme sejam resolvidas analiticamente seu valor numérico no computador é dado por aproximação, e portanto afetado por erros de arredondamento ou truncamento. Outras funções densidade como a normal ou gama, por exemplo, não possuem primitivas na forma analítica, sendo necessário o uso de integração numérica onde erros de arredondamentos e truncamentos são propagados devido às operações aritméticas no computador.

O objetivo desta tese é utilizar a Matemática Intervalar e a Aritmética de Exatidão Máxima para calcular intervalos encapsuladores, ou probabilidades autovalidáveis ou probabilidades encapsuladas ou ainda probabilidades intervalares para as variáveis Exponencial, Normal Padrão e Uniforme. No caso da Exponencial e Normal Padrão, o método proposto usou Simpson Intervalar. A Uniforme, devido ao fato de ter derivada de ordem quatro nula, teve uma forma diferente de encapsular probabilidades. A metodologia aqui proposta foi implementada no IntLab. Resultados numéricos ilustraram os teóricos. Adicionalmente, são mostrados como cálculos autovalidáveis podem ser usados em probabilidade condicional e independência.

Palavras-chave: <Aritmética de Exatidão Máxima, Matemática Intervalar, Probabilidade>

Abstract

In the study of continuous random variables we commonly have problems to calculate probabilities since it is necessary to solve definite integrals of density functions that most frequently do not have an explicit primitive or that primitive is not easy to be obtained. Although integrals related to Exponential and Uniform density functions can be analytically solved, computers only approximated their numerical values which are then affected by roundoff or truncation errors. The Normal density function, for example, doesn't have analytical form for the primitive, so it is necessary to use numerical integration to solve the integrals and for this reason roundoff and truncation errors are propagated due to the arithmetical operations in the computer.

Our main goal is to use interval mathematics and high accuracy arithmetic to obtain encapsulated intervals or encapsulated probabilities or interval probabilities to the Exponential, Standard Normal and Uniform distributions. As for the Exponential and Standard Normal distributions we use interval Simpson method. We have used a different method to encapsulate probabilities with the Uniform distribution because its fourth order derivative is zero. The presented methodology is implemented in IntLab. Numerical results illustrated the theoretical ones. Besides we show as autovalidated calculations can be used in conditional probabilities and independence.

Keywords: <High Accuracy Arithmetic, Interval Mathematics, Probability>

Sumário

1	Introdução	1
2	Fundamentos	4
2.1	Espaço dos Intervalos	4
2.1.1	Aritmética Intervalar	5
2.1.1.1	Adição	5
2.1.1.2	Subtração	6
2.1.1.3	Multiplicação	6
2.1.1.4	Divisão	7
2.1.1.5	Negativo de um Intervalo	8
2.1.1.6	Recíproco de um Intervalo	9
2.1.2	Intervalos como Conjuntos	10
2.1.2.1	Intersecção de dois intervalos	10
2.1.2.2	União de dois intervalos	11
2.1.2.3	União Convexa de dois intervalos	11
2.1.3	A Topologia de IIR	13
2.1.4	Conceitos Básicos da Análise Intervalar	21
2.2	Aritmética de Exatidão Máxima	27
2.3	Uma Definição de Probabilidade Intervalar	28
2.3.1	Noções Básicas de Probabilidade	29
2.3.2	Um Espaço de Probabilidade Intervalar	31
2.4	O Método de Newton Intervalar	32
2.5	O Método de Simpson Intervalar para Integração Numérica	35
2.5.1	O Método de Simpson Real	36
2.5.2	O Método de Simpson Intervalar	37
3	Probabilidade Condicional Intervalar. Independência	39
3.1	Probabilidade Condicional	39
3.2	Independência	45

4	Intervalos Encapsuladores para Probabilidades de Variáveis Aleatórias Contínuas	47
4.1	Cálculo de Probabilidades de Variáveis Aleatórias Contínuas	47
4.1.1	Intervalo Encapsulador para a Probabilidade de uma Variável Aleatória com Distribuição Exponencial	48
4.1.2	Distribuição Normal	51
4.1.3	Distribuição Uniforme	61
5	Conclusões, Contribuições e Trabalhos Futuros	65
	Referências Bibliográficas	67
A	Implementação Intervalares no Maple	70
A.1	Representação geométrica das operações aritméticas entre intervalos	70
	geosoma.msw	70
	geosub.msw	72
	geomult.msw	73
	geodiv.msw	74
	geoposto.msw	75
	geoinv.msw	76
A.2	Representação geométrica das operações entre intervalos como conjuntos	77
	geoint.msw	77
	geouni.msw	79
	geounicon.msw	80
A.3	Representação geométrica de conceitos topológicos	81
	geobaberta.msw	81
	geobfechada.msw	83
	geoesfera.msw	85
B	Implementação dos Métodos Intervalares no MatLab/IntLab	87
B.1	Conceitos Básicos do IntLab	87
B.2	Implementações dos intervalos encapsuladores no MatLab/IntLab	92
B.2.1	Implementação da fórmula de Simpson intervalar	92
B.2.2	Implementação do intervalo encapsulador para a densidade Exponencial	93
B.2.3	Implementação do Método de Newton Intervalar	95
B.2.4	Implementação da extensão intervalar inclusão monotônica derivada de ordem 4 da função $\exp(-x^2/2)$	96
B.2.5	Implementação do intervalo encapsulador para a Normal Padrão	111

B.2.6	Implementação do intervalo encapsulador para a densidade Uniforme	114
--------------	--	------------

Lista de Figuras

2.1	Representação geométrica de IIR.	5
2.2	Representação geométrica da Adição.	6
2.3	Representação geométrica da Subtração.	7
2.4	Representação geométrica da Multiplicação.	7
2.5	Representação geométrica da Divisão.	8
2.6	Representação geométrica do Negativo de um Intervalo.	9
2.7	Representação geométrica do Recíproco de um Intervalo.	10
2.8	Representação geométrica da Intersecção.	11
2.9	Representação geométrica da União.	12
2.10	Representação geométrica da União Convexa.	12
2.11	Representação geométrica da Bola Aberta de centro P e raio r.	14
2.12	Representação geométrica da Bola Aberta de centro $P = [-2, 4]$ e raio $r = 1.5$.	15
2.13	Representação geométrica da Bola Aberta de centro $P = [-1, \sqrt{3.5}]$ e raio $r = 4$.	15
2.14	Representação geométrica da Bola Fechada de centro P e raio r.	16
2.15	Representação geométrica da Bola Fechada de centro $P = [-1, \sqrt{3.5}]$ e raio $r = 4$.	17
2.16	Representação geométrica da Bola Fechada de centro $P = [-2, 7]$ e raio $r = 2$.	17
2.17	Representação geométrica da Esfera de centro P e raio r.	18
2.18	Representação geométrica da Esfera de centro $P = [-1, \sqrt{3.5}]$ e raio $r = 4$.	18
2.19	Representação geométrica da Esfera de centro $P = [-2, 7]$ e raio $r = 2$.	19

Lista de Símbolos

$\mathbb{R}$	conjunto dos números reais
$\mathbb{IIR}$	espaços dos intervalos de números reais
Ω	espaço de resultados elementares, espaço amostral
ω	evento simples, evento elementar
A, B	eventos aleatórios, eventos
X, Y, T	variáveis aleatórias
$X = [x_1, x_2], A = [\underline{a}, \bar{a}]$	intervalos
$m(X)$	ponto médio do intervalo X
$ X $	módulo do intervalo X
$w(X)$	diâmetro do intervalo X
$r(X)$	raio do intervalo
$B_r(P)$	bola aberta de centro P e raio r
$BF_r(P)$	bola fechada de centro P e raio r
$E_r(P)$	esfera de centro P e raio r
$\tilde{f}$	extensão unida da função f
$\tilde{f}$	cobertura intervalar da função f
$f _Y$	restrição da função f ao conjunto Y
$\min(Y)$	o mínimo do conjunto Y
$\max(Y)$	o máximo do conjunto Y
$D(f)$	domínio da função f
$d(X, Y)$	distância entre os intervalos X e Y
$\mathbb{Y}$	subconjunto do espaço $\mathbb{IIR}$
$\triangle x = \sup \inf x$	arredondamento monótono direcionado para cima
$\nabla x = \inf \sup x$	arredondamento monótono direcionado para baixo
sup	supremo
inf	ínfimo
$P(A)$	probabilidade (real) do evento A
$P_{\mathbb{IIR}}(A)$	probabilidade intervalar do evento A
$P_{\mathbb{IIR}}(A B)$	probabilidade condicional intervalar do evento A quando o evento B acontece

Lista de Tabelas

3.1	Número de Máquinas.	41
------------	----------------------------	-----------

Introdução

O cálculo das probabilidades fundamenta-se nas analogias entre medida de um conjunto e integral de uma função com probabilidade de um evento e esperança matemática de uma variável aleatória, respectivamente. A formalização do conceito de probabilidade começa com um espaço abstrato e desenvolve-se a partir desse. O conceito de variável aleatória porém, transforma-se rapidamente em um tema central da teoria. A razão para isto vem do fato que na maioria das situações experimentais o interesse é atribuir um número real ao resultado do experimento.

Computar probabilidades em situações práticas envolve números e, conseqüentemente, problemas numéricos [Campos97, CDCAD02, SDBCRC06, WanKen94]. Problemas numéricos na computação científica originam-se primordialmente da impossibilidade de se operar com números reais diretamente, pois tem-se que representar uma grandeza contínua de forma discreta e finita. O sistema de ponto flutuante é uma aproximação prática dos números reais. Uma alternativa bastante difundida para solucionar problemas numéricos, não só de imprecisão do dado em si como erros de arredondamento, truncamento e a propagação do erro numa sequência de operações aritméticas, é a matemática intervalar. A matemática intervalar [Moore66, Moore79, Sunaga58] fornece uma ferramenta para estimar e controlar erros numéricos automaticamente. Ao invés de aproximar um número real x por um número de máquina, x é aproximado por um intervalo X , tendo números de máquina como limites inferior e superior, o intervalo X contém (encapsula) o valor de x . A amplitude do intervalo pode ser usada como uma medida da qualidade da aproximação. Os cálculos entretanto tem de ser efetuados usando intervalos ao invés de números reais e as operações aritméticas têm de ser substituídas pelas operações sobre intervalos. A aritmética intervalar com os arredondamentos direcionados fornece limites rigorosos para o contradomínio de operações e funções. Há uma diferença qualitativa em computação científica desde que os resultados são intervalos nos quais os resultados exatos estão obrigatoriamente incluídos.

No estudo das variáveis aleatórias contínuas sobre o conjunto dos números reais, $\mathbb{R}$, um dos problemas é o cálculo de probabilidades, visto que é necessário resolver uma integral

definida da função densidade que, na maioria das vezes, não possui primitiva explícita ou cuja primitiva não é simples de se obter. Embora integrais de funções densidade de probabilidade como a exponencial e a uniforme, sejam resolvidas analiticamente, seu valor numérico no computador é dado por aproximação, e portanto afetado por erros de arredondamento ou truncamento. Outras funções densidade como a normal ou gama, por exemplo, não possuem primitivas na forma analítica, sendo necessário o uso de integração numérica onde erros de arredondamentos e truncamentos são propagados devido às operações aritméticas realizadas no computador.

Existem na literatura [BurFai03, RugLop96] vários métodos de integração numérica, como os métodos conhecidos como fórmulas de Newton-Cotes, quadratura Gaussiana, entre outros. Como o cálculo da integral definida é aproximado, faz-se necessário computar o erro nessa aproximação que também é um valor aproximado. A matemática intervalar fornece uma alternativa para resolver este problema, ou seja, o controle automático do erro numérico, pois encontra um intervalo que encapsula o valor da integral. Estudos e propostas relacionados com integrais intervalares podem ser encontrados em Caprani et al. [CMN02, CMR81], Moore [Moore66, Moore79], Moore et al. [MSY60] e Rall [Rall82]. Para outras aplicações ver [KeaKre95].

O objetivo principal desta tese é utilizar a Matemática Intervalar e a Aritmética de Exatidão Máxima para calcular intervalos encapsuladores, isto é, intervalos que contém o valor real ou probabilidades autovalidáveis¹ ou probabilidades encapsuladas, isto é, o intervalo ou ainda probabilidades intervalares [Campos97] para as variáveis Exponencial, Normal Padrão e Uniforme. Adicionalmente, são mostrados como cálculos autovalidáveis podem ser usados em probabilidade condicional e independência.

O computador no qual os programas foram executados possui a seguinte plataforma:

- (a) Memória RAM: 1GB.
- (b) Disco Rígido: 60GB.
- (c) Processador: AMD Sempron(tm) de 1.72GHz.
- (d) Sistema Operacional: WindowsXP.

¹Palavra usual na literatura da computação científica (baseada na matemática intervalar e na aritmética de exatidão máxima) para designar que o cálculo numérico realizado agrega o controle do erro numérico.

Esta tese está organizada em capítulos, sumarizados a seguir:

O Capítulo 2, Fundamentos, é dedicado a noções básicas sobre análise intervalar, probabilidade intervalar, aritmética de exatidão máxima e o método de Simpson intervalar. A definição do espaço dos intervalos com sua aritmética, topologia e análise intervalar é a proposta por Moore [Moore66, Moore79] e Moore et al. [MSY60], porém existem outras que, de certa forma, são extensões dessas; a probabilidade intervalar, usada na definição de uma probabilidade condicional intervalar, é a proposta por Campos [Campos97]; a aritmética de exatidão máxima está definida em Kulish e Miranker [KulMir81, KulMir86] e o método de Simpson intervalar para integração numérica para o cálculo de um intervalo encapsulando uma integral definida é o proposto em Caprani et al. [CMN02]. São implementadas as representações geométricas das operações aritméticas, intersecção, união, união convexa e dos elementos topológicos utilizando-se o Maple 8.0 [Maple02].

O Capítulo 3, Probabilidade Condicional Intervalar. Independência, é uma contribuição original desta tese pois apresenta uma definição de probabilidade condicional intervalar e uma propriedade da probabilidade intervalar de eventos independentes.

O Capítulo 4, Computando Probabilidades Autovalidáveis para Variáveis Aleatórias Contínuas, é o central desta tese. Nele define-se, usando-se o Método de Simpson Intervalar, um intervalo encapsulador para probabilidade para variáveis aleatórias contínuas com distribuição Exponencial e Normal. É construída uma tabela intervalar para a distribuição $N(0, 1)$, denominada $N_{\text{IIR}}(0, 1)$. O intervalo encapsulador para a Uniforme foi definido a partir da função densidade. Os algoritmos para obtenção dos intervalos encapsuladores foram implementados no Matlab [MatLab04], usando o IntLab [Rump06].

No Capítulo 5, Conclusões, Contribuições e Trabalhos futuros, apresentam-se as contribuições desta tese, bem como propostas de trabalhos futuros.

Por fim estão as referências bibliográficas e os apêndices.

Fundamentos

Neste capítulo será dada uma visão geral de:

- (i) Espaço dos intervalos: representação geométrica, aritmética, intervalos como conjuntos, topologia e conceitos básicos da análise intervalar.
- (ii) Aritmética de Exatidão Máxima.
- (iii) Probabilidade Intervalar.
- (iv) Método de Simpson Intervalar.

O software utilizado é o Maple 8.0 [Maple02, MonGed01]. As referências básicas são Alefeld e Herzberger [AleHer83], Campos [Campos97], Caprani et al. [CMN02], Kulisch e Miranker [KulMir81, KulMir86], Moore [Moore66, Moore79] e Sunaga [Sunaga58]. Os comandos no Maple para realizar os gráficos e cálculos deste capítulo encontram-se no Apêndice A. Salienta-se que a precisão numérica [Corliss05, Sterbenz74, Tucker07] adotada é a do Maple 8.0 [Maple02, MonGed01].

2.1 Espaço dos Intervalos

Esta seção introduz alguns conceitos básicos da análise intervalar.

Seja $\mathbf{IIR}$ o conjunto de todos os intervalos fechados de números reais,

$$\mathbf{IIR} = \{X = [x_1, x_2] \mid x_1, x_2 \in \mathbf{IR}, x_1 \leq x_2\},$$

x_1 , e x_2 são, respectivamente, o limite inferior e superior do intervalo X . Uma outra forma de denotar o intervalo X é $X = [x, \bar{x}]$. Associando-se a cada intervalo $[x, y] \in \mathbf{IIR}$ um ponto $(x, y) \in \mathbf{IR}^2$, obtem-se uma representação geométrica [Moore66] para $\mathbf{IIR}$, como mostra a Figura 2.1.

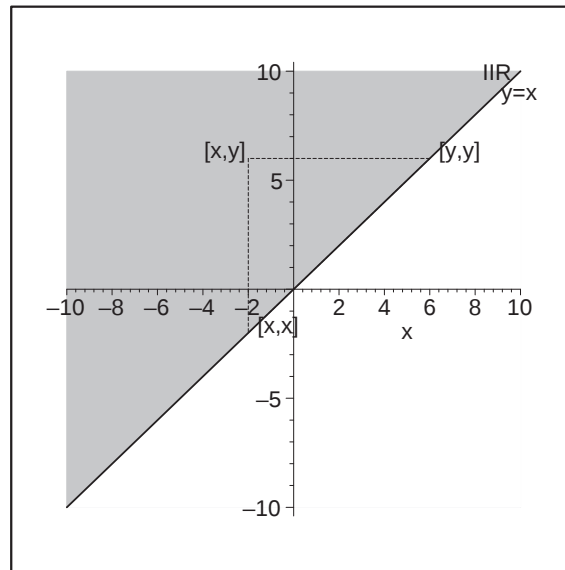


Figura 2.1 Representação geométrica de IIR.

Os pontos na diagonal $y = x$ entre (x, x) e (y, y) representam os números reais contidos no intervalo $[x, y]$ e este intervalo é representado pelo vértice (x, y) do triângulo. Diz-se que o intervalo $X = [x_1, x_2]$ é degenerado se $x_1 = x_2$. Por convenção, identifica-se um intervalo degenerado $[x, x]$ com o número real x .

Sejam $A = [a, b], B = [c, d] \in \text{IIR}$.

2.1.1 Aritmética Intervalar

Seja $*$ $\in \{+, -, \cdot, /\}$ uma operação binária sobre o conjunto dos números reais $\mathbb{R}$, então

$$A * B = \{a * b \mid a \in A, b \in B\}$$

define uma operação aritmética sobre IIR. É assumido que $0 \notin B$ no caso da divisão.

As operações aritméticas entre os intervalos A e B podem ser calculadas explicitamente como a seguir.

2.1.1.1 Adição

$$A + B = [a + c, b + d].$$

Exemplo 1. Se $A = [-3, 4.9]$ e $B = [1.5, 3.7]$, então $A + B = [-1.5, 8.6]$.

Representação geométrica: Figura 2.2 obtida através do procedimento

$$> \text{geosoma}(-3, 4.9, 1.5, 3.7);$$

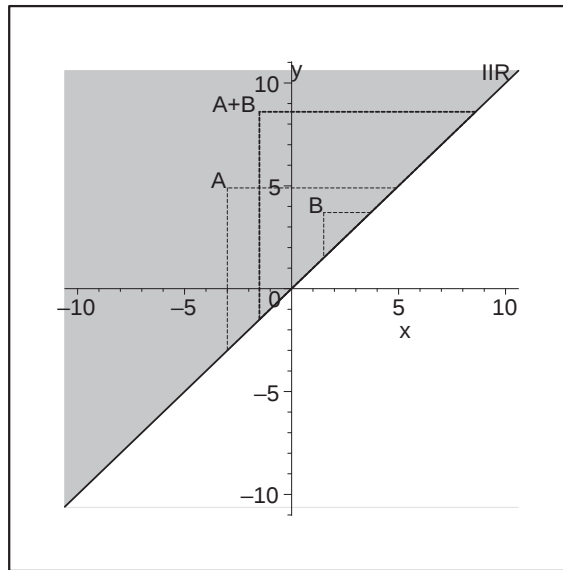


Figura 2.2 Representação geométrica da Adição.

2.1.1.2 Subtração

$$A - B = [a - d, b - c].$$

Exemplo 2. Se $A = [-1.4, 2.3]$ e $B = [-3, 6]$, então $A - B = [-7.4, 5.3]$.

Representação geométrica: Figura 2.3 obtida através do procedimento

`> geosub(-1.4, 2.3, -3, 6);`

2.1.1.3 Multiplicação

$$A \cdot B = [\min\{ac, ad, bc, bd\}, \max\{ac, ad, bc, bd\}].$$

Exemplo 3. Se $A = [-2, 3]$ e $B = [-3, 6]$, então $A \cdot B = [-12, 18]$.

Representação geométrica: Figura 2.4 obtida através do procedimento

`> geomul(-2, 3, -3, 6);`

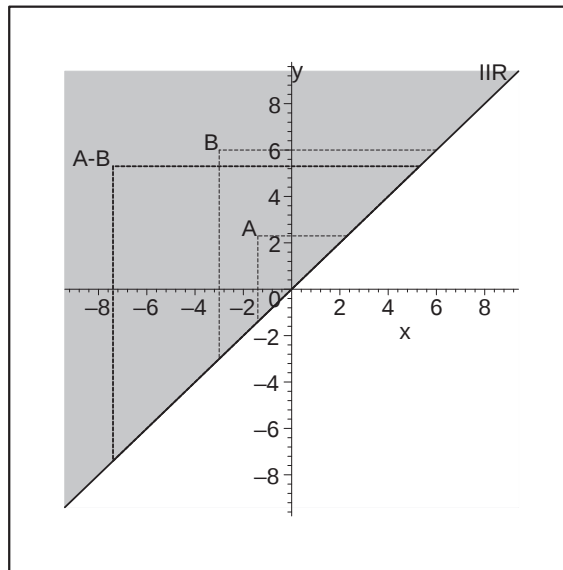


Figura 2.3 Representação geométrica da Subtração.

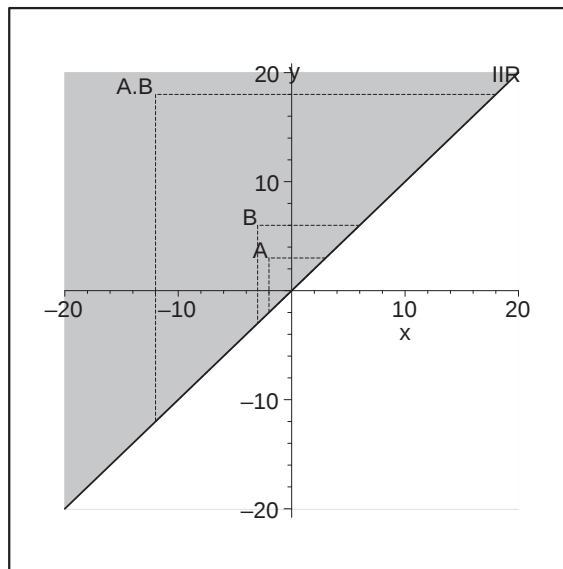


Figura 2.4 Representação geométrica da Multiplicação.

2.1.1.4 Divisão

$$A/B = [\min\{\frac{a}{c}, \frac{a}{d}, \frac{b}{c}, \frac{b}{d}\}, \max\{\frac{a}{c}, \frac{a}{d}, \frac{b}{c}, \frac{b}{d}\}].$$

desde que $0 \notin B$.

Exemplo 4. Se $A = [-4, 5]$ e $B = [1.5, 2.6]$, então $A/B = [-2.667, 3.333]$.

Representação geométrica: Figura 2.5 obtida através do procedimento

$> \text{geodiv}(-4, 5, 1.5, 2.6);$

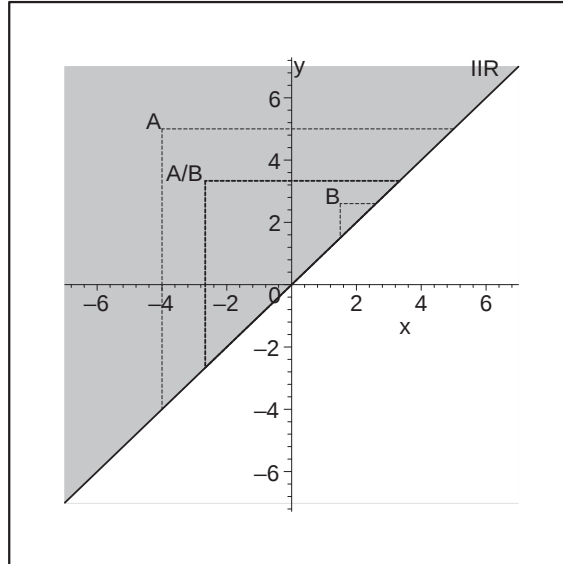


Figura 2.5 Representação geométrica da Divisão.

Exemplo 5. Se $A = [-4, 5]$ e $B = [-1.5, 2.6]$, determinar A/B .

Neste caso a divisão não está definida. E, através do procedimento

$> \text{geodiv}(-4, 5, -1.5, 2.6);$

tem-se a resposta:

"A divisão não está definida. 0 pertence à B. "

Além das operações binárias, pode-se definir as seguintes operações unárias:

2.1.1.5 Negativo de um Intervalo

O negativo do intervalo A , denotado por $-A$, é o intervalo definido por

$$-A = [-b, -a].$$

Exemplo 6. Se $A = [-\sqrt{2}, 5]$, então $-A = [-5, \sqrt{2}]$.

Representação geométrica: Figura 2.6 obtida através do procedimento

$> \text{geoposto}(\text{eval } f(-\text{sqrt}(2)), 5);$

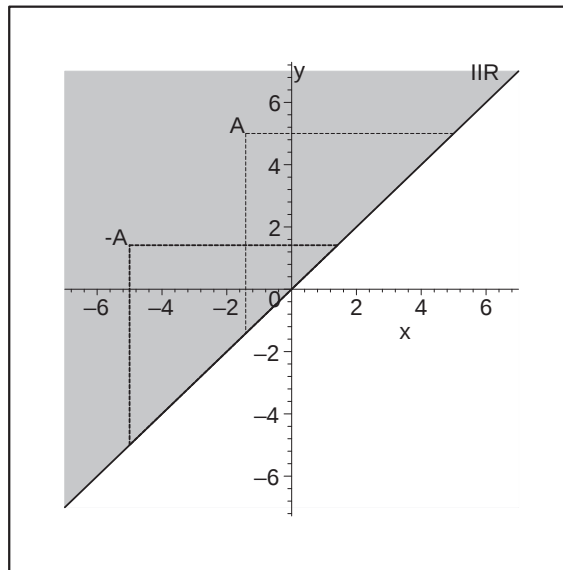


Figura 2.6 Representação geométrica do Negativo de um Intervalo.

2.1.1.6 Recíproco de um Intervalo

Se $0 \notin A$, o recíproco de A , denotado por $1/A$, é o intervalo definido por

$$1/A = [\frac{1}{b}, \frac{1}{a}].$$

Exemplo 7. Se $A = [\sqrt{2}, 5]$, então $1/A = [0.200, 0.707]$.

Representação geométrica: Figura 2.7 obtida através do procedimento

```
> geoinv(eval f(sqrt(2),5);
```

Exemplo 8. Se $A = [-1.5, 4]$, determine $1/A$.

Neste caso não existe o recíproco de A . E, através do procedimento

```
> geoinv(-1.5,4,5,5);
```

tem-se a resposta:

"Não existe o recíproco de A . 0 pertence à A ."

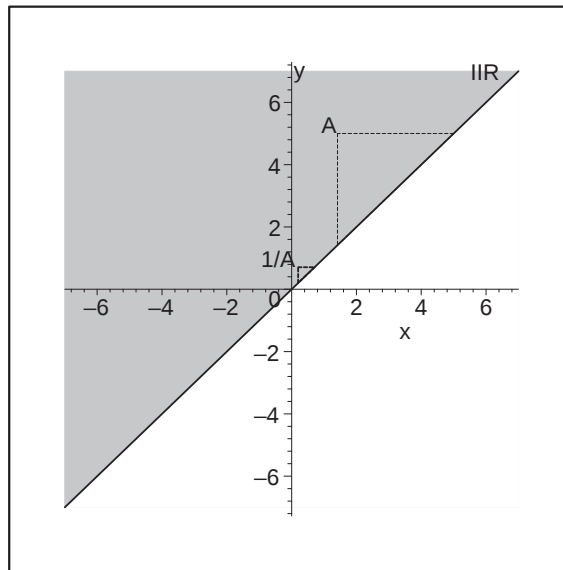


Figura 2.7 Representação geométrica do Recíproco de um Intervalo.

2.1.2 Intervalos como Conjuntos

Outras operações que podem ser definidas sobre IIR são a intersecção, a união e a união convexa de dois intervalos.

2.1.2.1 Intersecção de dois intervalos

A intersecção dos intervalos A e B , denotada por $A \cap B$, é definida por

$$A \cap B = \begin{cases} \emptyset, & \text{se } a > d \text{ ou } b < c, \\ [\max\{a, c\}, \min\{b, d\}], & \text{caso contrário.} \end{cases}$$

Exemplo 9. Sejam $A = [-2, 4]$ e $B = [2, 6]$, então $A \cap B = [2, 4]$.

Representação geométrica: Figura 2.8 obtida através do procedimento

`> geoint(-2,4,2,6);`

Exemplo 10. Sejam $A = [-2, 1]$ e $B = [2, 5]$, determine $A \cap B$.

Neste caso $A \cap B = \emptyset$ e, através do procedimento

`> geoint(-2,1,2,5);`

tem-se a resposta:

"Intersecção vazia."

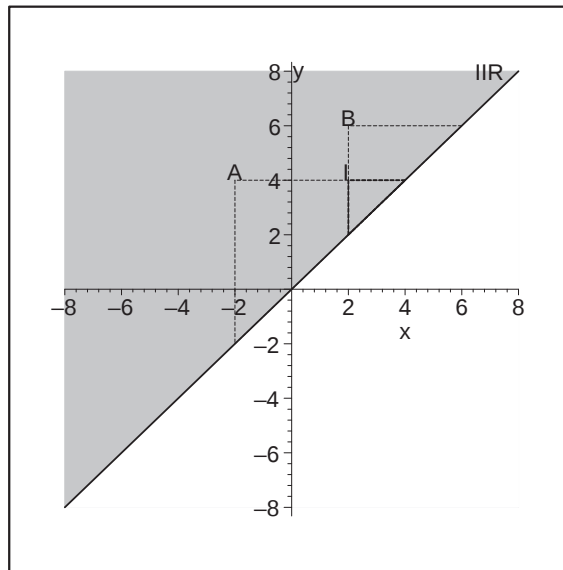


Figura 2.8 Representação geométrica da Intersecção.

2.1.2.2 União de dois intervalos

Se $A \cap B \neq \emptyset$, sua união é o intervalo definido por

$$A \cup B = [\min\{a, c\}, \max\{b, d\}].$$

Exemplo 11. Sejam $A = [-2, 4]$ e $B = [2, 6]$, então $A \cup B = [-2, 6]$.

Representação geométrica: Figura 2.9 obtida através do procedimento

$$> \text{geouni}(-2, 4, 2, 6);$$

Exemplo 12. Sejam $A = [-2, 1]$ e $B = [2, 6]$, determine $A \cup B$.

Como $A \cap B = \emptyset$ a união não está definida e, através do procedimento

$$> \text{geouni}(-2, 1, 2, 6);$$

tem-se a resposta:

"intersecção vazia, não está definida a união."

2.1.2.3 União Convexa de dois intervalos

A união convexa dos intervalos A e B , denotada por $\overline{A \cup B}$ é o intervalo

$$\overline{A \cup B} = [\min\{a, c\}, \max\{b, d\}].$$

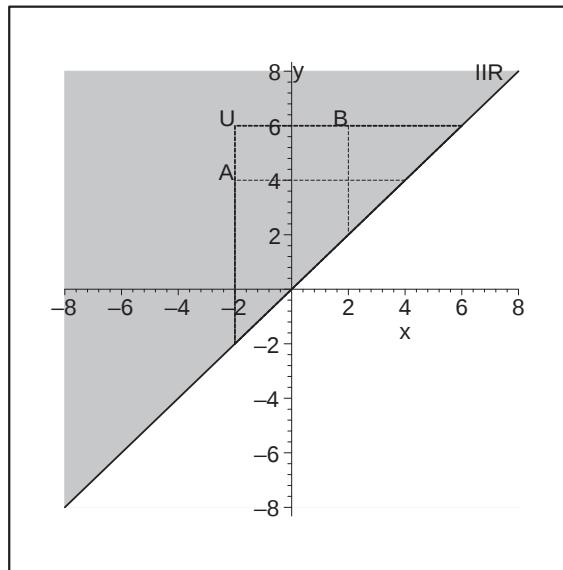


Figura 2.9 Representação geométrica da União.

Exemplo 13. Sejam $A = [-2, 1]$ e $B = [2, 6]$, então $\overline{A \cup B} = [-2, 6]$.

Representação geométrica: Figura 2.10 obtida através do procedimento

$> geounicon(-2, 1, 2, 6);$

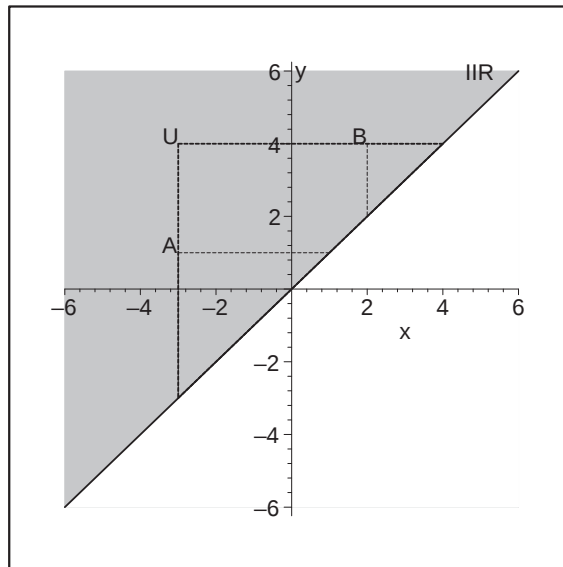


Figura 2.10 Representação geométrica da União Convexa.

2.1.3 A Topologia de IIR

Os conceitos topológicos aqui vistos serão fundamentais para o capítulo seguinte.

Sejam $A = [a, b]$ e $B = [c, d]$ em IIR, tem-se que:

o diâmetro de A , $w(A)$ é

$$w(A) = b - a;$$

o raio de A , $r(A)$ é

$$r(A) = \frac{1}{2}(b - a);$$

o ponto médio de A , $m(A)$ é

$$m(A) = \frac{1}{2}(a + b);$$

o módulo de A , $|A|$ é

$$|A| = \max\{|a|, |b|\}.$$

Definição 2.1.1. Define-se a **distância** entre os intervalos A e B , $d(A, B)$, como sendo o número real não negativo dado por

$$d(A, B) = \max\{|a - c|, |b - d|\}.$$

Esta distância se reduz à distância usual definida em $\mathbb{R}$ quando aplicada a intervalos degenerados. Prova-se que d satisfaz as seguintes propriedades: Para quaisquer A, B e $C \in \text{IIR}$, tem-se

- (i) $d(A, B) \geq 0$, e $d(A, B) = 0 \Leftrightarrow A = B$
- (ii) $d(A, B) = d(B, A)$,
- (iii) $d(A, B) \leq d(A, C) + d(C, B)$,

Exemplo 14. Sejam $A = [-1, 3]$ e $B = [2, 5]$, então

$$w(A) = 4;$$

$$r(A) = 2;$$

$$m(A) = 1;$$

$$|A| = 3;$$

$$d(A, B) = 3.$$

Teorema 1. O espaço métrico $\langle \text{IIR}, d \rangle$ com a métrica d é um espaço métrico completo.

Prova: Alefeld e Herzberger [AleHer83], página 11.

Os resultados que se seguem decorrem do fato de IIR ser um espaço métrico.

Definição 2.1.2. Define-se **bola aberta** de centro em $P \in \text{IIR}$ e raio $r > 0$, $B_r(P)$, como o conjunto dos pontos $X \in \text{IIR}$ cuja distância ao ponto A é menor do que r ,

$$B_r(P) = \{X \in \text{IIR} \mid d(X, P) < r\}.$$

Uma representação geométrica de bola aberta encontra-se na Figura 2.11.

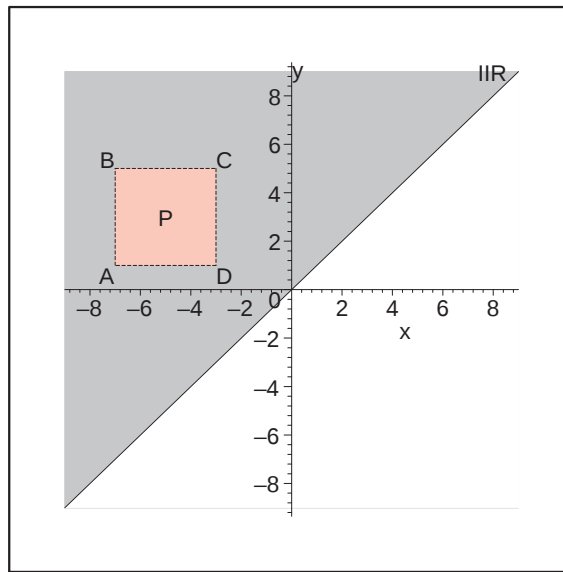


Figura 2.11 Representação geométrica da Bola Aberta de centro P e raio r .

Exemplo 15. Sejam $P = [-2, 4]$ e $r = 1.5$, então

$$B_r(P) = \{X = [x, y] \in \text{IIR} \mid \max\{|x + 2|, |y - 4|\} < 1.5\}.$$

Representação geométrica: Figura 2.12 obtida através do procedimento

$$> \text{geobaberta}(-2, 4, 1.5);$$

Exemplo 16. Sejam $P = [-1, \sqrt{3.5}]$ e $r = 4$, então

$$B_r(P) = \{X = [x, y] \in \text{IIR} \mid \max\{|x + 1|, |y - \sqrt{3.5}|\} < 4\}.$$

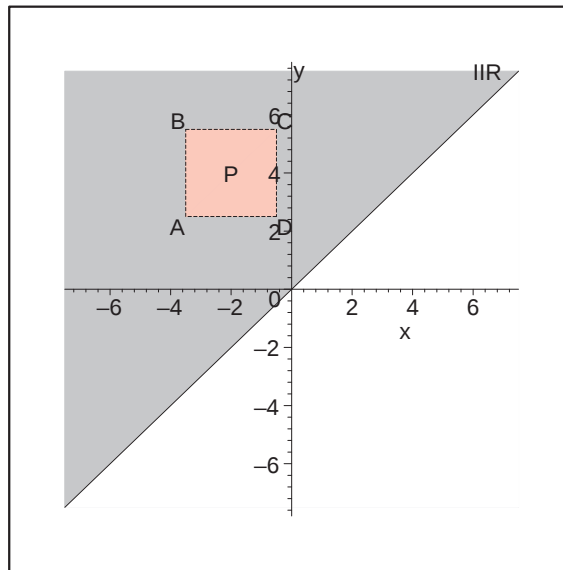


Figura 2.12 Representação geométrica da Bola Aberta de centro $P = [-2, 4]$ e raio $r = 1.5$.

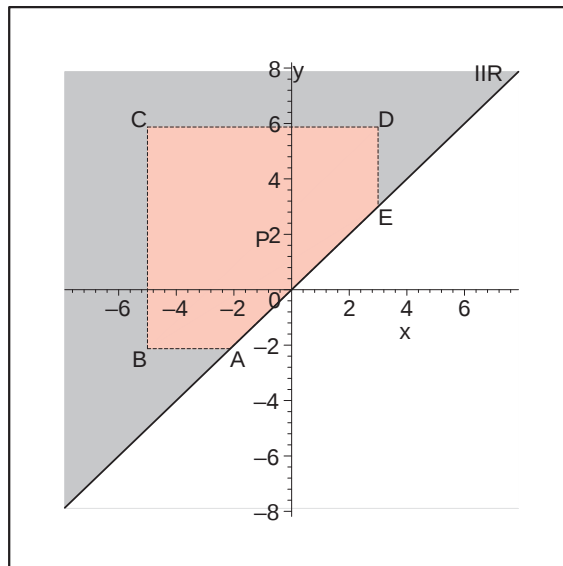


Figura 2.13 Representação geométrica da Bola Aberta de centro $P = [-1, \sqrt{3.5}]$ e raio $r = 4$.

Representação geométrica: Figura 2.13 obtida através do procedimento

$> geobaberta(-1, \sqrt{3.5}, 4);$

.

Definição 2.1.3. Define-se **bola fechada** de centro em $P \in \text{IIR}$ e raio $r > 0$, $BF_r(P)$, como o conjunto dos pontos $X \in \text{IIR}$ cuja distância ao ponto P é menor ou igual a r .

$$BF_r(P) = \{X \in \text{IIR} \mid d(X, P) \leq r\}.$$

Uma representação geométrica de bola fechada encontra-se na Figura 2.14.

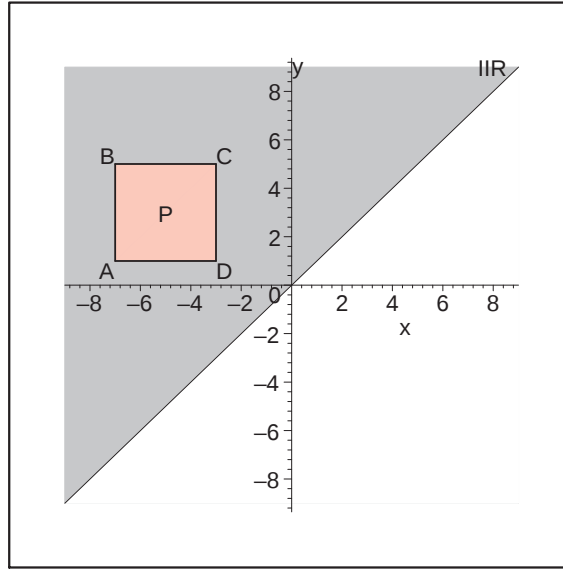


Figura 2.14 Representação geométrica da Bola Fechada de centro P e raio r .

Exemplo 17. Sejam $P = [-1, \sqrt{3.5}]$ e $r = 4$, então

$$BF_r(P) = \{X = [x, y] \in \text{IIR} \mid \max\{|x + 1|, |y - \sqrt{3.5}|\} \leq 4\}.$$

Representação geométrica: Figura 2.15 obtida através do procedimento

$$> \text{geobfechada}(-1, \sqrt{3.5}, 4);$$

Exemplo 18. Sejam $P = [-2, 7]$ e $r = 2$, então

$$BF_r(P) = \{X = [x, y] \in \text{IIR} \mid \max\{|x + 2|, |y - 7|\} \leq 2\}.$$

Representação geométrica: Figura 2.16 obtida através do procedimento

$$> \text{geobfechada}(-2, 7, 2);$$

.

Definição 2.1.4. Define-se **esfera** de centro em $P \in \text{IIR}$ e raio $r > 0$, $E_r(P)$, como o conjunto dos pontos $X \in \text{IIR}$ cuja distância ao ponto P é igual a r .

$$E_r(P) = \{X \in \text{IIR} \mid d(X, P) = r\}.$$

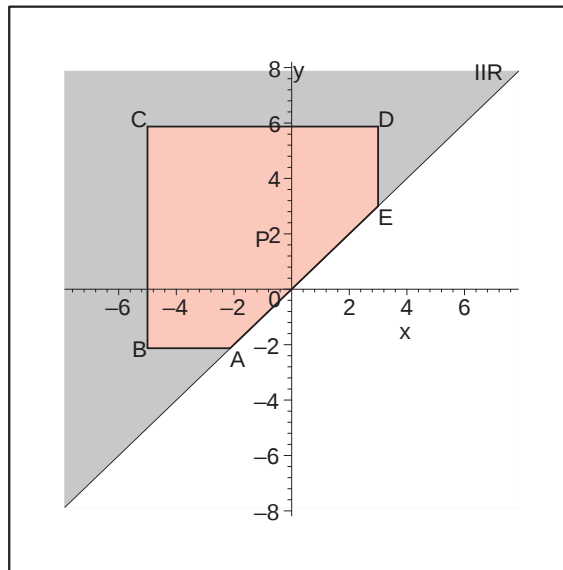


Figura 2.15 Representação geométrica da Bola Fechada de centro $P = [-1, \sqrt{3.5}]$ e raio $r = 4$.

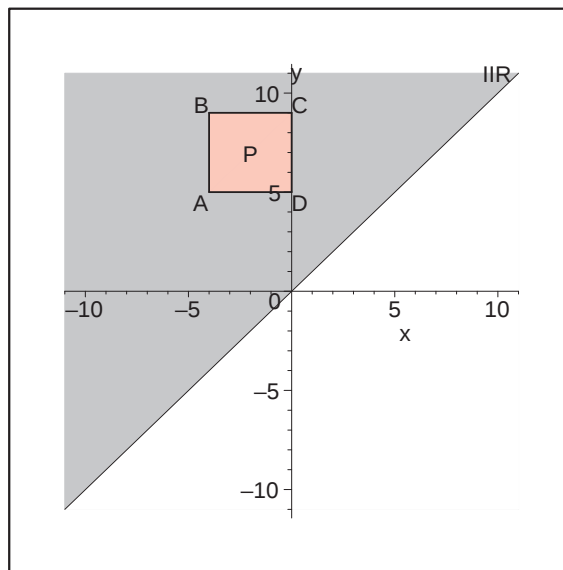


Figura 2.16 Representação geométrica da Bola Fechada de centro $P = [-2, 7]$ e raio $r = 2$.

Uma Representação geométrica de esfera encontra-se na Figura 2.17.

Exemplo 19. Sejam $P = [-1, \sqrt{3.5}]$ e $r = 4$, então

$$E_r(P) = \{X = [x, y] \in \text{IIR} \mid \max\{|x + 1|, |y - \sqrt{3.5}|\} = 4\}.$$

Representação geométrica: Figura 2.18 obtida através do procedimento

$$> \text{geoesfera}(-1, \sqrt{3.5}, 4);$$

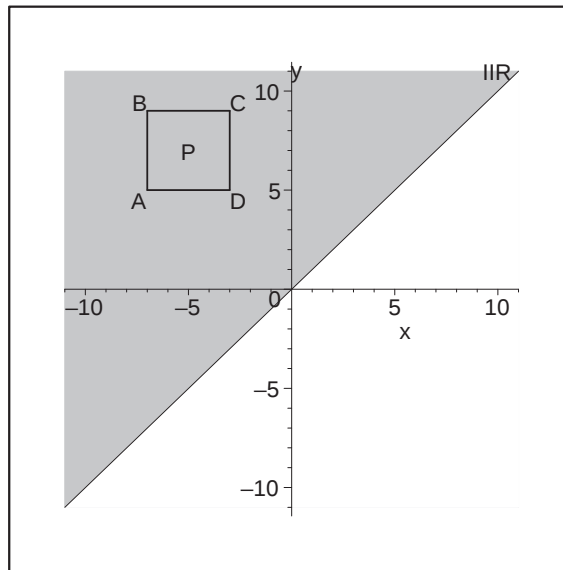


Figura 2.17 Representação geométrica da Esfera de centro P e raio r .

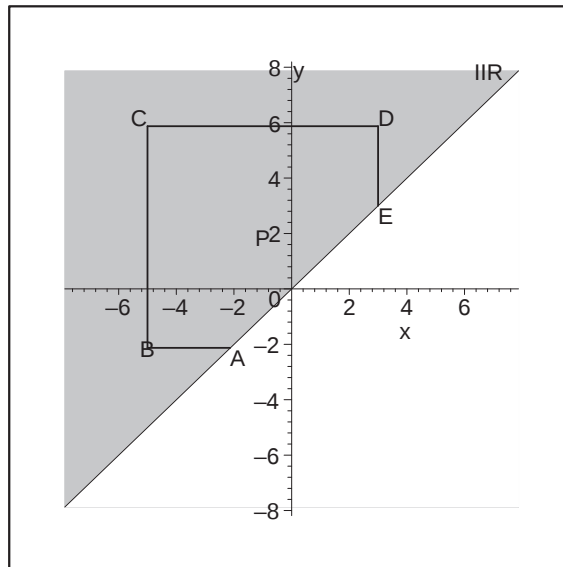


Figura 2.18 Representação geométrica da Esfera de centro $P = [-1, \sqrt{3.5}]$ e raio $r = 4$.

.

Exemplo 20. Sejam $P = [-2, 7]$ e $r = 2$, então

$$E_r(P) = \{X = [x, y] \in \text{IIR} \mid \max\{|x + 2|, |y - 7|\} = 2\}.$$

Representação geométrica: Figura 2.19 obtida através do procedimento

$$> \text{geoesfera}(-2, 7, 2);$$

.

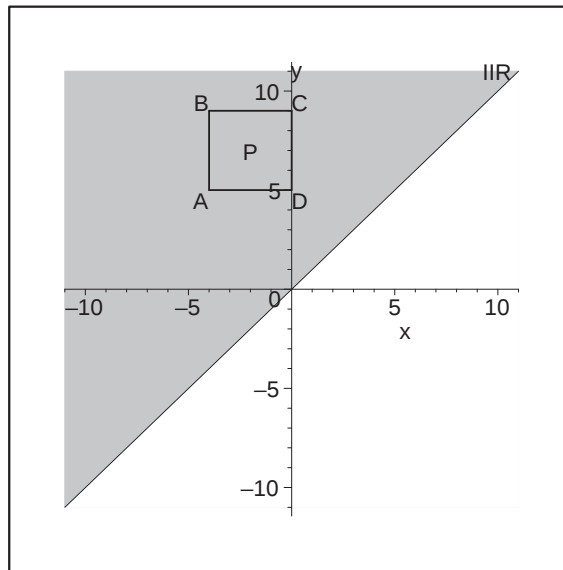


Figura 2.19 Representação geométrica da Esfera de centro $P = [-2, 7]$ e raio $r = 2$.

Definição 2.1.5. Seja $IA \subset IIR$. Um ponto $P \in IA$ é um **ponto interior** de IA se e só se P pertence a alguma bola aberta contida em IA .

Definição 2.1.6. Seja $IA \subset IIR$. IA é **aberto** se e só se cada um de seus pontos é um ponto interior.

Teorema 2. Toda bola aberta é um conjunto aberto em IIR .

Prova: Seja $B_r(P)$ uma bola aberta. Seja $T \in B_r(P)$. Tem-se que $d(P, T) < r$. Se $\varepsilon = r - d(P, T)$, então $B_\varepsilon(T) \subseteq B_r(P)$, pois para todo $X \in B_\varepsilon(T)$, $d(X, T) < \varepsilon \leq r - d(P, T)$ e

$$\begin{aligned} d(P, X) &\leq d(P, T) + d(T, X) \\ &< d(P, T) + \varepsilon \\ &= r. \end{aligned}$$

Portanto $B_r(P)$ é um conjunto aberto em IIR . ■

Teorema 3. Seja Γ o conjunto de todos os subconjuntos $IY \subset IIR$ tal que IY são uniões arbitrárias de bolas abertas, então

- (i) $IY \in \Gamma$ é um aberto.
- (ii) A intersecção de duas bolas abertas é um aberto.
- (iii) Se $IY, IZ \in \Gamma$ então $IY \cap IZ \in \Gamma$.

Prova:

(i) Seja $\mathbf{IY} = \bigcup_{\alpha} B_{r_{\alpha}}(P_{\alpha})$, onde $\alpha \in T$, T conjunto de índices. Seja $M \in \mathbf{IY}$. Como $M \in \mathbf{IY}$ existe $n \in T$ tal que $M \in B_{r_n}(P_n)$, mas $B_{r_n}(P_n)$ é aberto, logo existe uma bola aberta com centro em M tal que $B_r(M) \subset B_{r_n}(P_n)$. Como $B_{r_n}(P_n)$ é subconjunto de $\mathbf{IY}$, $B_r(M)$ também é subconjunto de $\mathbf{IY}$. Logo $\mathbf{IY}$ é aberto.

(ii) Sejam $B_{r_1}(A)$ e $B_{r_2}(B)$ bolas abertas.

1) Se $B_{r_1}(A) \cap B_{r_2}(B) = \emptyset$, o resultado é trivial.

2) Se $B_{r_1}(A) \cap B_{r_2}(B) \neq \emptyset$, seja

$$C \in B_{r_1}(A) \cap B_{r_2}(B),$$

isto é,

$$d(A, C) < r_1 \quad \text{e} \quad d(B, C) < r_2.$$

Tomando $r = \min\{r_1 - d(A, C), r_2 - d(B, C)\}$ e $B_{r/2}(C)$, tem-se

$$C \in B_{r/2}(C) \subset B_{r_1}(A) \cap B_{r_2}(B),$$

portanto C é um ponto interior de $B_{r_1}(A) \cap B_{r_2}(B)$.

Logo,

$$B_{r_1}(A) \cap B_{r_2}(B)$$

é um aberto.

(iii) Como $\mathbf{IY}, \mathbf{IZ} \in \Gamma$ são abertos, então

1) Se $\mathbf{IY} \cap \mathbf{IZ} = \emptyset$, o resultado é trivial.

2) Se $\mathbf{IY} \cap \mathbf{IZ} \neq \emptyset$. Sejam $W \in \mathbf{IY} \cap \mathbf{IZ}$, então $W \in \mathbf{IY}$ e $W \in \mathbf{IZ}$. Logo existem bolas abertas B_1 e B_2 tais que $W \in B_1 \subset \mathbf{IY}$ e $W \in B_2 \subset \mathbf{IZ}$.

Então $W \in B_1 \cap B_2 \subset \mathbf{IY} \cap \mathbf{IZ}$. Como a intersecção de bolas de duas bolas abertas quaisquer é aberta, existe uma bola aberta B_s tal que

$W \in B_s \subset B_1 \cap B_2 \subset \mathbf{IY} \cap \mathbf{IZ}$. Logo W é ponto interior de $\mathbf{IY} \cap \mathbf{IZ}$ e $\mathbf{IY} \cap \mathbf{IZ}$ é aberto.

■

2.1.4 Conceitos Básicos da Análise Intervalar

Considere a função

$$f : \mathbb{R} \rightarrow \mathbb{R}$$

definida no domínio $\mathbf{IA} \in \mathbb{IR}$. A imagem de $\mathbf{IA}$ é

$$f(\mathbf{IA}) = \{f(x) \mid x \in \mathbf{IA}\}.$$

Se f é contínua, então este conjunto é fechado e compacto. É óbvio que $f(\mathbf{IX}) \subseteq f(\mathbf{IA})$ para todo $\mathbf{IX} \subseteq \mathbf{IA}$. O menor intervalo que encapsula (contém) $f(\mathbf{IX})$ é chamado a casca (cobertura) intervalar de f e denotado por $\tilde{f}$ [CMN02].

Definição 2.1.7. Sejam $\mathbf{IX}, \mathbf{IY} \subseteq \mathbb{IR}$. A função

$$\begin{aligned} F : \mathbf{IX} &\rightarrow \mathbf{IY} \\ X &\mapsto F(X) \end{aligned}$$

é denominada **função intervalar de uma variável intervalar**.

Definição 2.1.8. Sejam M_1 e M_2 conjuntos quaisquer, $g : M_1 \rightarrow M_2$ uma função, $\mathcal{P}(M_1)$ e $\mathcal{P}(M_2)$ famílias de subconjuntos de M_1 e M_2 respectivamente. A função de conjunto $\bar{g} : \mathcal{P}(M_1) \rightarrow \mathcal{P}(M_2)$,

$$\bar{g}(X) = \{g(x) : x \in X, X \in \mathcal{P}(M_1)\},$$

é denominada **extensão unida de g** .

A função $\bar{g}$ satisfaz a propriedade de subconjuntos, isto é, $X, Y \in \mathcal{P}(M_1)$,

$$X \subseteq Y \Rightarrow \bar{g}(X) \subseteq \bar{g}(Y). \quad (2.1)$$

Definição 2.1.9. Uma função $F = F(X)$ é **inclusão monotônica** se

$$Y \subseteq X \Rightarrow F(Y) \subseteq F(X).$$

Teorema 4. Extensões unidas são inclusões monotônicas.

Prova: De fato. Por (2.1), toda extensão unida satisfaz a propriedade de subconjunto. ■

Nem toda função intervalar é inclusão monotônica. Por exemplo, seja a função intervalar F definida por

$$F(X) = m(X) + \frac{1}{2}(X - m(X)).$$

Tem-se

$$F([0, 2]) = 1 + \frac{1}{2}[-1, 1] = \left[\frac{1}{2}, \frac{3}{2}\right];$$

enquanto

$$F([0, 1]) = \frac{1}{2} + \frac{1}{2}\left[-\frac{1}{2}, \frac{1}{2}\right] = \left[\frac{1}{4}, \frac{3}{4}\right] \not\subseteq F([0, 2]).$$

Definição 2.1.10. Seja f uma função real de variável real. A função intervalar F de variável intervalar $X \in \text{IIR}$ é uma **extensão intervalar** de f se

$$F([x, x]) = f(x), x \in X.$$

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 .

Teorema 5. Se F é uma extensão intervalar inclusão monotônica de f , então

$$\bar{f}(X) \subseteq F(X).$$

Prova: Pela definição de extensão intervalar, $f(x) = F(x)$. Se F é inclusão monotônica, então o valor de $f(x)$ pertence ao intervalo $F(X)$ para todo $x \in X$. Logo, $\bar{f}(X) \subseteq F(X)$. ■

Existem várias extensões intervalares de uma função real f . Isto acontece, por exemplo, porque expressões racionais que são equivalentes na aritmética real podem não o ser na aritmética intervalar.

Exemplo 21. Seja

$$f(x) = x(1 - x) = x - x \cdot x,$$

e as extensões intervalares de f

$$F_1(X) = X(1 - X)$$

e

$$F_2(X) = X - X \cdot X.$$

Se $X = [0, 1]$ então

$$F_1(X) = [0, 1]$$

e

$$F_2(X) = [-1, 1].$$

Portanto,

$$F_1(X) \neq F_2(X).$$

Os Exemplos 22 e 23 a seguir mostram funções intervalares que são extensões intervalares inclusões monotônicas.

Exemplo 22. Seja $X = [\underline{x}, \bar{x}] \in \text{IIR}$. Para valores inteiros positivos de n , define-se a **Função Potência Intervalar** por $X^n : \text{IIR} \rightarrow \text{IIR}$

$$X^n = \begin{cases} [\underline{x}^n, \bar{x}^n], & \text{se } \underline{x} > 0 \text{ ou } n \text{ é ímpar,} \\ [\bar{x}^n, \underline{x}^n], & \text{se } \bar{x} < 0 \text{ e } n \text{ é par,} \\ [0, |X|^n], & \text{se } 0 \in X \text{ e } n \text{ é par.} \end{cases}$$

Exemplo 23. A *Função Exponencial Natural Intervalar* é definida por

$$\begin{aligned} \exp : I\mathbb{R} &\rightarrow I\mathbb{R} \\ \exp(X) &= [\exp(\underline{x}), \exp(\bar{x})] \end{aligned}$$

Seja f uma função de uma variável real contínua definida no intervalo $X = [x_1, x_2]$. Se f é monótona (crescente ou decrescente) uma extensão intervalar para f em X é dada por

$$F(X) = [f(x_1), f(x_2)],$$

se f é crescente ou

$$F(X) = [f(x_2), f(x_1)],$$

se f é decrescente.

Neste caso $\tilde{f}(X) = F(X)$, e F é uma extensão intervalar inclusão monotônica de f .

Pode-se construir extensões intervalares inclusão monotônica para funções reais definidas por várias sentenças [Moore79]. As funções intervalares definidas a seguir são exemplos de tal construção.

Exemplo 24. Suponha que f é uma função contínua definida por

$$f(x) = \begin{cases} p_1(x), & x_1 \leq x < x_2, \\ p_2(x), & x_2 \leq x < x_3, \\ p_3(x), & x_3 \leq x \leq x_4 \end{cases}$$

Sejam as extensões intervalares inclusões monotônicas P_1, P_2, P_3 para p_1, p_2, p_3 , respectivamente. Afirma-se, a função intervalar F definida por

$$F(X) = \overline{P_1(X \cap [x_1, x_2]) \cup P_2(X \cap [x_2, x_3]) \cup P_3(X \cap [x_3, x_4])}, \quad (2.2)$$

é uma extensão intervalar inclusão monotônica para f .

De fato. Tem-se que:

(a) F é inclusão monotônica.

De fato. Seja $Y \subset X$, então $(Y \cap [x_1, x_2]) \subseteq (X \cap [x_1, x_2])$ ou $(Y \cap [x_2, x_3]) \subseteq (X \cap [x_2, x_3])$ ou $(Y \cap [x_3, x_4]) \subseteq (X \cap [x_3, x_4])$. Como $P_i, i = 1, 2, 3$, é inclusão monotônica, tem-se que $P_1(Y \cap [x_1, x_2]) \subseteq P_1(X \cap [x_1, x_2])$ ou $P_2(Y \cap [x_2, x_3]) \subseteq P_2(X \cap [x_2, x_3])$ ou $P_3(Y \cap [x_3, x_4]) \subseteq P_3(X \cap [x_3, x_4])$. Logo,

$$F(Y) = \overline{P_1(Y \cap [x_1, x_2]) \cup P_2(Y \cap [x_2, x_3]) \cup P_3(Y \cap [x_3, x_4])} \subseteq F(X).$$

Portanto, F é inclusão monotônica.

(b) F é uma extensão intervalar para f .

De fato. Se $x \in X$ então, $x \in (X \cap [x_1, x_2])$ ou $x \in (X \cap [x_2, x_3])$ ou $x \in (X \cap [x_3, x_4])$. Como P_i é uma extensão intervalar para $p_i, i = 1, 2, 3$, tem-se que $P_1(x) = p_1(x)$ ou $P_2(x) = p_2(x)$ ou $P_3(x) = p_3(x)$, e como f é contínua tem-se $F(x) = f(x)$. Portanto, F é uma extensão intervalar para f .

Exemplo 25. Seja f contínua e derivável no intervalo I . Suponha que f tenha k extremos relativos, a saber: $b_1, b_2, \dots, b_k$ e que $b_i < b_{i+1}, i = 1, 2, \dots, k-1$. Determine, em $\mathbb{R}$, intervalos disjuntos $B_1, B_2, \dots, B_k$, contidos no domínio da f , $D(f)$, tais que $b_i \in B_i, i = 1, 2, \dots, k$. Após a determinação dos intervalos $B_1, B_2, \dots, B_k$, obtem-se os intervalos $C_1, C_2, \dots, C_{k+1}$, contidos em $D(f)$, onde, $C_1 = [\min(D(f)), \text{extremo inferior de } B_1]$, $C_{k+1} = [\text{extremo superior de } B_k, \max(D(f))]$ e $C_i = [\text{extremo superior de } B_{i-1}, \text{extremos inferior de } B_i], i = 2, \dots, k$ e sobre C_i a função f seja monótona. Assim o intervalo I será dado por

$$I = (I \cap C_1) \cup (I \cap B_1) \cup (I \cap C_2) \cup \dots \cup (I \cap B_k) \cup (I \cap C_{k+1}), \quad (2.3)$$

Algumas interseções em (2.3) podem ser vazias, neste caso pode-se representar I por

$$I = D_1 \cup D_2 \cup \dots \cup D_n, \quad n \leq 2k+1,$$

onde D_i é igual ou está contido em um dos intervalos B_j, C_l definidos acima.

Observação: Serão adotadas as seguintes convenções:

(a) Se $D(f) = \mathbb{R}$ toma-se $\min(D(f)) = -\infty$ e $\max(D(f)) = +\infty$.

(b) Se $D(f) = \{x \in \mathbb{R} | x \leq a\}$ toma-se $\min(D(f)) = -\infty$.

(c) Se $D(f) = \{x \in \mathbb{R} | x \geq a\}$ toma-se $\max(D(f)) = +\infty$.

Dessa forma f pode ser definida por partes, a saber:

$$f(x) = \begin{cases} p_1(x) = f|_{C_1}(x), & x \in C_1; \\ p_2(x) = f|_{B_1}(x), & x \in B_1; \\ p_3(x) = f|_{C_2}(x), & x \in C_2; \\ \vdots & \vdots \\ p_k(x) = f|_{B_k}(x) & x \in B_k; \\ p_{k+1}(x) = f|_{C_{k+1}}(x) & x \in C_{k+1}. \end{cases}$$

Em analogia como o Exemplo 24, pode-se construir uma extensão intervalar para f da seguinte forma: Seja o intervalo $X \subset D(f)$

(a) Determine a representação de X usando (2.3) para obter

$$X = D_1 \cup D_2 \cup \dots \cup D_n, \quad n \leq 2k + 1,$$

o extremo direito de D_i coincidindo com o extremo esquerdo de D_{i+1} , e cada D_i é um subconjunto de B_j ou de C_l .

(b) Se $D_i = [y_1, y_2] \subset C_l$, faça

$$F([y_1, y_2]) = [f(y_1), f(y_2)],$$

se f é crescente, ou

$$F([y_1, y_2]) = [f(y_2), f(y_1)],$$

se f é decrescente.

Se $D_i = [y_1, y_2] \subset B_j$, então, usando-se a extensão intervalar natural de f ou qualquer outra extensão intervalar inclusão monotônica, F de f , determina-se $F(D_i)$.

(c) Calcula-se a união

$$F(X) = F(D_1) \cup F(D_2) \cup \dots \cup F(D_n) \quad (2.4)$$

da esquerda para a direita. Desde que $D_1 \cap D_2$ é não vazio, também o é $F(D_1) \cap F(D_2)$ e, assim por diante.

Pelo Exemplo 24, F é uma extensão intervalar inclusão monotônica de f .

Se $\tilde{f}$ é a extensão unida de f , então

$$\tilde{f}(X) = \tilde{f}(D_1) \cup \tilde{f}(D_2) \cup \dots \cup \tilde{f}(D_n).$$

Portanto, por construção, $\tilde{f}(X) \subseteq F(X)$.

Moore [Moore66, Moore79], Moore et al. [MSY60] mostram que as funções racionais de variáveis reais têm extensões intervalares naturais, bastando para isso substituir as variáveis reais por intervalos e as operações aritméticas sobre os números reais pelas correspondentes sobre os intervalos.

(i) Seja I_d a função identidade definida por

$$I_d : \mathbb{R} \rightarrow \mathbb{R}$$

$$I_d(x) = x,$$

uma extensão intervalar para I_d é:

$$I_D : \mathbb{IR} \rightarrow \mathbb{IR}$$

$$I_D(X) = X.$$

(ii) Com respeito às operações aritméticas, $*, * \in \{+, -, \cdot, /\}$, tem-se, respectivamente, que:

$$* : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$$

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

uma extensão para $*$ é

$$* : \text{IIR} \times \text{IIR} \rightarrow \text{IIR}$$

$$*(X, Y) = X * Y$$

As funções F_{I_d} e F_* abaixo estendem um número real para um intervalo, e as operações sobre os reais para as operações sobre intervalos [Campos97] e são necessárias para a construção da probabilidade intervalar proposta por Campos [Campos97].

Definição 2.1.11. *Sejam x, y números reais, $e * \in \{+, -, \cdot, /\}$ e:*

$$F_{I_d} : \mathbb{R} \rightarrow \text{IIR}$$

$$F_{I_d}(x) = X, \quad x \in X, \quad (2.5)$$

$$F_* : \mathbb{R} \times \mathbb{R} \rightarrow \text{IIR} \times \text{IIR}$$

$$F_*(x * y) = X * Y. \quad (2.6)$$

Definição 2.1.12. *Seja F uma extensão intervalar de f sobre $A \subseteq \text{IIR}$. Diz-se que F é uma **extensão intervalar linear** se existe um $K > 0$, independente de X , tal que*

$$d(F(X), \tilde{f}(X)) \leq Kw(X), \text{ para todo } X \subseteq A.$$

Teorema 6. *Se F é uma extensão intervalar linear de uma função lipschitziana f ¹, então existe uma constante $K > 0$ tal que*

$$w(F(X)) \leq Kw(X).$$

Prova: Caprani et al [CMN02], página 25.

¹Seja $f : X \subseteq \mathbb{R} \mapsto \mathbb{R}$. Diz-se que f é lipschitziana se existe uma constante $C > 0$, tal que

$$x, y \in X \Rightarrow |f(x) - f(y)| \leq C|x - y|.$$

Teorema 7. *Seja f um função definida em $X \subseteq IIR$, e seja $X_{(1)}, \dots, X_{(p)}$ uma subdivisão de X , tal que,*

$$X_{(j)} \subseteq X \quad \text{e} \quad \bigcup_{j=1}^p X_{(j)} = X.$$

Ainda, seja F uma extensão intervalar de f em X . Então

$$\tilde{f}(X) \subseteq \overline{\bigcup_{j=1}^p F(X_{(j)})}.$$

Prova: Caprani et al [CMN02], página 33.

Em outras palavras, dada uma extensão intervalar de f e uma subdivisão do domínio X , obtem-se uma nova função intervalar,

$$F_p(X) = \overline{\bigcup_{j=1}^p F(X_{(j)})},$$

que é uma extensão intervalar de f .

Teorema 8. *Seja F uma extensão intervalar linear de f sobre $A \subseteq IIR$. Seja $K > 0$ uma constante tal que*

$$d(F(X), \tilde{f}(X)) \leq Kw(X), \forall X \subseteq A.$$

Então

$$d(F(X), \tilde{f}(X)) \leq K \max_j \{w(X_{(j)})\}.$$

Prova: Caprani et al [CMN02], página 33.

No caso da subdivisão ser equidistante, isto é, $w(X_{(j)}) = \frac{1}{p}w(X)$, o Teorema 8 diz que

$$d(F(X), \tilde{f}(X)) \leq Kw(X)/p.$$

O lado direito converge para zero para $p \rightarrow \infty$.

2.2 Aritmética de Exatidão Máxima

Computadores somente podem representar uma quantidade finita de números [BurFai03] [Goldberg91][IEEE754][RugLop96][Sterbenz74]. A representação dos reais internamente nas máquinas é realizada através dos números de ponto flutuante. Portanto, se o sistema de ponto flutuante é $F(b, l, e_1, e_2)$ onde b é a base, l a quantidade de dígitos do significando, e_1 o menor expoente e e_2 o maior expoente, então o número de elementos de F é $2(b-1)b^{l-1}(e_2 - e_1 + 1) + 1 < \infty$. Assim,

a representação dos reais, que tem infinitos elementos, é realizada por F o que ocasiona os erros de arredondamento, truncamento e inerentes.

Na verdade, o primeiro tipo de erro que ocorre ao se operar com os números reais em máquinas é o de mudança de base. Trabalha-se e pensa-se na base 10, todavia, de acordo com o padrão IEEE754 [IEEE754] a base numérica usada no processador de ponto flutuante é a binária. É sabido que

$$0.1_{(10)} = 0.0001100011 \dots_{(2)}.$$

Do ponto de vista algébrico, a adição de números de ponto flutuante nem sempre é associativa. Aliás, as operações aritméticas entre números de ponto flutuante podem nem mesmo ser um número de ponto flutuante, como é o caso da ocorrência de underflow e overflow. Considerando que o controle do erro numérico é realizado através do uso de intervalos ao invés de números reais, Kulisch [Kulisch08] e Kulisch e Miranker [KulMir81, KulMir86] propuseram que a implementação da aritmética intervalar seja realizada através da 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 de ponto flutuantes consecutivos. No caso de intervalos o suporte para a realização das operações aritméticas com exatidão máxima é usar os arredondamentos monótonos direcionados, ∇ e $\triangle$, nos extremos dos intervalos: o limite inferior é arredondado para baixo (∇) e o superior para cima ($\triangle$). Por exemplo, sejam $S(10, 5, -3, 3)$ um sistema de ponto flutuante e $X = [2.34568, 3.42036]$ um intervalo. Usando os arredondamentos monótonos direcionados tem-se, neste sistema,

$$\bar{X} = [\nabla 2.34568, \triangle 3.42036] := [2.3456, 3.4294].$$

Esta implementação da aritmética é que está implementada no IntLab [Rump06].

2.3 Uma Definição de Probabilidade Intervalar

A probabilidade intervalar proposta por Campos [Campos97] resolve problemas numéricos associados com o cálculo de probabilidades reais. Enfatiza-se ainda que a probabilidade intervalar é uma extensão da probabilidade real, mas que mantém toda a consistência semântica do cálculo probabilístico, buscando para isso recursos da análise intervalar [Acioly91, Moore66, Moore79] e da aritmética de exatidão máxima [KulMir81, KulMir86].

2.3.1 Noções Básicas de Probabilidade

A Teoria da Probabilidade é o ramo da matemática relacionado com *fenômenos aleatórios*, os quais apresentam as seguintes características: (i) podem ser repetidos sob as mesmas condições; (ii) apresentam um conjunto de possíveis resultados; (iii) quando realizado um grande número de vezes, apresentam uma regularidade. Um exemplo clássico é o lançamento de uma moeda. A teoria da probabilidade desenvolve uma estrutura matemática formal para o tratamento dos fenômenos aleatórios. Sua construção axiomática foi proposta por Kolmogorov [Burrill72, Feller68].

O conjunto dos possíveis resultados de um experimento aleatório é chamado *espaço amostral* e é denotado por Ω . Seja $\mathcal{A}$ uma família de subconjuntos de Ω . Os subconjuntos de Ω são chamados *eventos*. Um evento é um elemento A em $\mathcal{A}$. Do ponto de vista matemático $\mathcal{A}$ é apenas uma coleção especificada de subconjuntos de Ω . Serão associadas probabilidades apenas aos elementos $A \in \mathcal{A}$, isto é, aos eventos.

A questão seguinte é definir qual a estrutura da coleção $\mathcal{A}$. Inicialmente pode-se supor que $\mathcal{A}$ seja fechada com respeito a uniões e interseções finitas. Dizer que o evento A não ocorre significa dizer que ocorre seu complementar, A^c . Assim, exige-se também que a classe $\mathcal{A}$ seja fechada com respeito a complementações. Conclui-se, então que $\mathcal{A}$ deve ter uma estrutura algébrica. Porém, por razões matemáticas, $\mathcal{A}$ deve ser uma classe mais restritiva, isto é, $\mathcal{A}$ deve ser fechada com respeito a uniões enumeráveis. Exige-se, portanto, que $\mathcal{A}$ seja uma σ -álgebra.

Definição 2.3.1. Uma coleção $\mathcal{A}$ de subconjuntos de Ω é uma σ -álgebra se os seguintes axiomas são satisfeitos:

$$(A1) \quad \Omega \in \mathcal{A}.$$

$$(A2) \quad A \in \mathcal{A} \Rightarrow A^c \in \mathcal{A}.$$

$$(A3) \quad \text{Se } A_n \text{ está em } \mathcal{A}, n = 1, 2, \dots, \text{ então } \bigcup_{n=1}^{\infty} A_n \in \mathcal{A}.$$

A axiomática da teoria das probabilidades fundamenta-se nas analogias entre medida de um conjunto e probabilidade de um evento, na integral de uma função e esperança matemática de uma variável aleatória.

Definição 2.3.2. Um espaço de probabilidade é uma quádrupla $\langle \mathcal{A}, \mathbb{R}, \Omega, P \rangle$, onde $\mathcal{A}$ é uma família de conjuntos, $\mathbb{R}$ é o conjunto dos números reais, Ω é um elemento destacado de $\mathcal{A}$, o qual corresponde ao espaço amostral e $P : \mathcal{A} \rightarrow \mathbb{R}$ é uma medida sobre $\mathcal{A}$ assumindo valores no conjunto dos reais, satisfazendo os seguintes axiomas [Burrill72, Feller68]:

$$(A1) \quad P(A) \geq 0, \forall A \in \mathcal{A}.$$

(A2) $P(\Omega) = 1$.

(A3) (σ -aditividade). Se A_n é uma coleção contável de subconjuntos de $\mathcal{A}$ disjuntos dois a dois, então $P(\cup_{n=1}^{\infty} A_n) = \sum_{n=1}^{\infty} P(A_n)$.

Observa-se que o conjunto de axiomas acima é usualmente proposto, porém não é o único [Kolmogorov50, Feller68, Burrill72, Billingsley79].

Dos axiomas vistos tem-se, dentre outras, como consequências, onde todos os conjuntos são membros de $\mathcal{A}$, o seguinte:

(a) $P(\emptyset) = 0$.

(b) $P(A) \leq 1$.

(c) $P(A^c) = 1 - P(A)$.

(d) $A \subseteq B \Rightarrow P(A) \leq P(B)$.

(e) $P(A \cup B) = P(A) + P(B) - P(A \cap B)$.

(f) $A \subseteq B \Rightarrow P(B - A) = P(B) - P(A)$.

(g) Propriedade monotônica: se A_n é uma seqüência crescente ou decrescente com limite A , então $P(A_n) \rightarrow P(A)$.

(h) Desigualdade de Boole: $P(\cup_{n=1}^{\infty} A_n) \leq \sum_{n=1}^{\infty} P(A_n)$.

Observa-se que o conceito de probabilidade é suportado fundamentalmente pelas noções de conjuntos, relações e funções e o sistema de números reais. A teoria da probabilidade começa com um espaço abstrato de probabilidades, e desenvolve-se a partir desse. O conceito de variável aleatória, visto a seguir, dada a sua importância, transforma-se rapidamente em um tema central da teoria.

Definição 2.3.3. Uma variável aleatória X [Feller68, James06] em um espaço de probabilidade $\langle \mathcal{A}, \mathbb{R}, \Omega, P \rangle$ é uma função real definida sobre Ω , $X : \Omega \rightarrow \mathbb{R}$, tal que $\{\omega \in \Omega \mid X(\omega) \leq x\} \in \mathcal{A}$ para todo $x \in \mathbb{R}$.

2.3.2 Um Espaço de Probabilidade Intervalar

A probabilidade intervalar, definida por Campos [Campos97], é realizada através de uma peculiar composição de funções, envolvendo extensões intervalares de funções reais. Inicialmente a um certo evento é associada a probabilidade real p . Em seguida, a p é associado um intervalo que a contém. Da mesma forma que uma extensão canônica para um número real é um intervalo que o contenha, assim também uma extensão natural para a probabilidade real é a probabilidade intervalar. A probabilidade intervalar portanto *aproxima* a probabilidade real. A seguir é definido o espaço de probabilidade intervalar com a função probabilidade intervalar satisfazendo específicas propriedades ou axiomas.

Definição 2.3.4. *Seja $\mathcal{A}$ uma σ -álgebra, P a função probabilidade e a função $P_{I\mathbb{R}}$ definida por*

$$\begin{aligned} P_{I\mathbb{R}} : \mathcal{A} &\rightarrow I\mathbb{R} \\ P_{I\mathbb{R}}(A) &= (F \circ P)(A) \end{aligned} \quad (2.7)$$

onde F é como na definição 2.1.11.

Portanto, $P_{I\mathbb{R}}$ associa a cada elemento $A \in \mathcal{A}$ um intervalo X com a propriedade que dada a probabilidade do evento A , $P(A) = p \in \mathbb{R}$, tem-se que $p \in X$. Observe que o intervalo X pode ser degenerado. Como comentado anteriormente, podem existir várias extensões intervalares para uma dada função real f , porém com respeito a 2.7, ou $F = F_{I_d}$ associa um intervalo X tal que $p \in X$, ou $F = F_*$ corresponde a extensões de operações aritméticas sobre os números reais.

Lema 1. *Seja Ω o espaço amostral, $\mathcal{A}$ uma σ -álgebra de subconjuntos de Ω e a função probabilidade P . Seja ε um número real tal que $\varepsilon > 0$. A função $P_{I\mathbb{R}}$ satisfaz as seguintes propriedades:*

- (a) *Dada $P(A) = p$, $P_{I\mathbb{R}}(A) = X$, $p \in X$.*
- (b) *$P_{I\mathbb{R}}(\Omega) = 1_\varepsilon$, onde $1_\varepsilon = [1 - \varepsilon, 1 + \varepsilon]$, $\varepsilon \in \mathbb{R}$, $\varepsilon > 0$.*
- (c) *$P_{I\mathbb{R}}(\emptyset) = 0_\varepsilon$, onde $0_\varepsilon = [-\varepsilon, +\varepsilon]$, $\varepsilon \in \mathbb{R}$, $\varepsilon > 0$.*
- (d) *Se $A \cap B = \emptyset$, $A \neq \emptyset$ e $B \neq \emptyset$, então $P_{I\mathbb{R}}(A \cup B) = P_{I\mathbb{R}}(A) + P_{I\mathbb{R}}(B)$.*
- (e) *Dada a sequência $A_1, A_2, \dots$, com $A_i \neq \emptyset$, $i = 1, 2, \dots$. Se $A_i \cap A_j = \emptyset$, $\forall i \neq j$, então*

$$P_{I\mathbb{R}}(\cup_{n=1}^{\infty} A_n) = \sum_{n=1}^{\infty} P_{I\mathbb{R}}(A_n).$$

Prova: Campos [Campos97].

Definição 2.3.5. Um espaço de probabilidade intervalar é uma sêxtupla $\langle \mathcal{A}, \mathbb{R}, \mathbb{IIR}, \Omega, P, P_{\mathbb{IIR}} \rangle$, onde $\mathcal{A}$ é uma família de conjuntos, Ω é um elemento destacável de $\mathcal{A}$, $\mathbb{R}$ é o conjunto dos números reais aqui destacado, $\mathbb{IIR}$ é o espaço dos intervalos, P uma função probabilidade e $P_{\mathbb{IIR}} : \mathcal{A} \rightarrow \mathbb{IIR}$ é uma função satisfazendo as seguintes propriedades:

$$(A1) \quad \forall A \in \mathcal{A} \text{ se } P(A) \in \mathbb{R}, \text{ então } P(A) \in P_{\mathbb{IIR}}(A)$$

$$(A2) \quad 1 \in P_{\mathbb{IIR}}(\Omega)$$

$$(A3) \quad 0 \in P_{\mathbb{IIR}}(\emptyset)$$

$$(A4) \quad \text{Seja } A_1, A_2, \dots, \text{ com } A_i \neq \emptyset, i = 1, 2, \dots, \text{ uma sequência de elementos de } \mathcal{A}. \text{ Se } A_i \cap A_j = \emptyset \text{ para todo } i \neq j \text{ então, } P_{\mathbb{IIR}}(\cup_{n=1}^{\infty} A_n) = \sum_{n=1}^{\infty} P_{\mathbb{IIR}}(A_n)$$

A função $P_{\mathbb{IIR}}$ satisfazendo as propriedades A1, A2, A3 e A4 acima é denominada uma *probabilidade intervalar* ou uma *extensão intervalar de P* ou ainda uma *validação para P*.

Por analogia com a probabilidade real, as propriedades A1, A2, A3 e A4 também são chamadas axiomas. A propriedade A4 é denominada σ -aditividade.

Observa-se que a menos de $P_{\mathbb{IIR}}(\emptyset)$, quaisquer outras probabilidades intervalares, podem sem perda de generalidade ser consideradas como tendo extremos inferiores números reais maiores ou iguais a zero.

2.4 O Método de Newton Intervalar

O Método de Newton real [BurFai03, RugLop96] é um método iterativo (MPF - Métodos de Ponto Fixo) para resolver equações.

Seja f uma função contínua em $[a, b]$, intervalo que contém raiz, ε , da equação

$$f(x) = 0. \tag{2.8}$$

O MPF consiste em transformar (2.8) em uma equação equivalente

$$\varphi(x) = x \tag{2.9}$$

e a partir de uma aproximação inicial x_0 gerar a sequência $\{x_k\}$ de aproximações para ε pela relação

$$x_{k+1} = \varphi(x_k), \tag{2.10}$$

pois a função φ é tal que $f(\varepsilon) = 0$ se e somente se $\varphi(\varepsilon) = \varepsilon$. Assim o problema de encontrar um zero de f transforma-se no problema de encontrar um ponto fixo de φ . Uma função φ que satisfaz a condição acima é chamada de *função de iteração* para a Equação (2.8). O teorema a seguir dá condições suficientes que φ deve satisfazer para que se tenha convergência garantida.

Teorema 9. *Seja ε uma raiz da Equação (2.8), isolada num intervalo I centrado em ε . Seja φ uma função de iteração para a Equação (2.8). Se*

(i) φ e φ' forem contínuas em I ;

(ii) $|\varphi'(x)| \leq m < 1, \forall x \in I$;

(iii) $x_0 \in I$;

então a sequência x_k gerada por $x_{k+1} = \varphi(x_k)$, $k = 0, 1, 2, \dots$ converge para ε .

Prova: A demonstração deste teorema é feita em duas partes:

(a) prova-se que se $x_0 \in I$, então $x_k \in I, \forall k$.

De fato. Como ε é ponto fixo de φ e, para qualquer k , $x_{k+1} = \varphi(x_k)$, tem-se

$x_{k+1} - \varepsilon = \varphi(x_k) - \varphi(\varepsilon)$. Sabe-se que φ é contínua e diferenciável em I , então, pelo Teorema do Valor Médio,

$$x_{k+1} - \varepsilon = \varphi(x_k) - \varphi(\varepsilon) = \varphi'(b_k)(x_k - \varepsilon),$$

com b_k entre x_k e ε . Assim,

$$x_{k+1} - \varepsilon = \varphi'(b_k)(x_k - \varepsilon).$$

Logo, para todo k

$$|x_{k+1} - \varepsilon| = |\varphi'(b_k)| |x_k - \varepsilon| \leq m |x_k - \varepsilon|, \quad (2.11)$$

por (ii). Portanto,

$$|x_{k+1} - \varepsilon| < |x_k - \varepsilon|,$$

uma vez que $m < 1$. Ou seja, a distância entre x_{k+1} e ε é estritamente menor que a distância entre x_k e ε , e como I é centrado em ε , tem-se que se $x_k \in I$, então $x_{k+1} \in I$. Por hipótese, $x_0 \in I$, então $x_k \in I, \forall k$.

(b) prova-se que $\lim_{k \rightarrow \infty} |x_k - \varepsilon| = 0$.

De fato. De (2.11), pode-se provar as seguintes desigualdades:

$$|x_k - \varepsilon| \leq m |x_{k-1} - \varepsilon|, \quad |x_{k-1} - \varepsilon| \leq m |x_{k-2} - \varepsilon|, \dots$$

e fazendo-se majorações sucessivas, obtem-se

$$|x_k - \varepsilon| \leq m^k |x_0 - \varepsilon|.$$

Como $m < 1$,

$$\lim_{k \rightarrow \infty} |x_k - \varepsilon| = \lim_{k \rightarrow \infty} m^k |x_0 - \varepsilon| = 0.$$

Portanto,

$$\lim_{k \rightarrow \infty} |x_k - \varepsilon| = 0.$$

■

O que o Método de Newton faz, na tentativa de garantir e acelerar a convergência do MPF, é escolher para função de iteração a função φ tal que $\varphi'(\varepsilon) = 0$. Então, dada f , a função de iteração

$$\varphi(x) = x - \frac{f(x)}{f'(x)},$$

desde que $f'(x) \neq 0$, será tal que $\varphi'(\varepsilon) = 0$.

Assim, escolhido x_0 , a sequência $\{x_k\}$ será determinada por

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}, k = 0, 1, 2, \dots \quad (2.12)$$

O Método de Newton intervalar [Moore66] constrói uma sequência convergente de intervalo, cujo limite é um intervalo contendo a raiz.

Suponha f contínua e derivável em $X_0 = [a, b]$, intervalo contendo a raiz ε . Seja F' uma extensão intervalar inclusão monotônica da derivada f' da função f . O operador Newtoniano Intervalar é definido por

$$N(X) = m(X) - \frac{f(m(X))}{F'(X)} \quad (2.13)$$

e a sequência de intervalos $X_1, X_2, \dots$, é construída por

$$X_{n+1} = N(X_n) \cap X_n.$$

Teorema 10. *Uma condição necessária para $N(X)$ ser definida por (2.13) é que X contenha um único zero de f e que este zero seja uma raiz simples.*

Prova: Se f' tem um zero em $X = [a, b]$ então $\tilde{f}'(X) \subset F'(X)$ é um intervalo contendo zero, portanto $N(X)$ não está definida. Como consequência, se f tem duas raízes distintas em X , então f' deve ter um zero entre elas e $N(X)$ não está definida, além disso, f' é zero nas raízes múltiplas de f . ■

Teorema 11. *Se $N(X)$ é definida por (2.13) para um intervalo X contendo uma raiz simples de f , $f(x) = 0$, $x \in X$, então $x \in N(X)$.*

Prova: Se $m(X) = x$ então $N(X) = x$, por (2.13). Caso contrário, suponha $m(X) < x$ e considere o intervalo $I = [m(X), x]$. f é contínua em I , pelo Teorema do Valor Médio [Rudin71], existe $y \in I \subset X$ tal que

$$f(m(X)) = f'(y)(m(X) - x).$$

Portanto,

$$x = m(X) - \frac{f(m(X))}{f'(y)}$$

e, desde que $f'(y) \in F'(X)$, conclui-se, de (2.13) que $x \in N(X)$. De forma análoga se analisa o caso em que $m(X) > x$. ■

Pelo Teorema (11), se $N(X)$ está definida, tem-se duas situações possíveis:

- (a) $N(X) \cap X = \emptyset$, se X não contém um zero de f , ou
- (b) $N(X) \cap X$ é um intervalo que contém um zero de f , se X o contém.

2.5 O Método de Simpson Intervalar para Integração Numérica

Dada a função contínua $f : \mathbb{R} \rightarrow \mathbb{R}$ sobre o intervalo $A = [a, b]$ o objetivo é calcular

$$S = \int_a^b f(x) dx. \quad (2.14)$$

Como f é contínua, f é integrável a Riemann. Se G é uma primitiva de f em $[a, b]$ tem-se, pelo Teorema Fundamental do Cálculo Integral, que

$$S = G(b) - G(a).$$

$G(a)$ ou $G(b)$ podem não ter uma representação exata no computador, o que implica que seu valor tem de ser aproximado por um número de máquina. Se não é possível encontrar uma expressão analítica para uma primitiva de f , métodos numéricos são essenciais para encontrar uma solução numérica do problema. A idéia básica da integração numérica é a substituição da função f por um polinômio que a aproxime razoavelmente no intervalo $[a, b]$, assim o problema se resume a resolver uma integral de polinômio. Há na literatura vários métodos de integração numérica, como os métodos conhecidos como fórmulas de Newton-Cotes, quadratura Gaussiana, entre outros. Nas fórmulas de Newton-Cotes a idéia de polinômio que aproxime f razoavelmente é que este polinômio interpole f em pontos de $[a, b]$. As fórmulas dada pela interpolação de f por polinômio de grau 1, 2 ou n podem ser aplicadas ou no intervalo $[a, b]$, constituindo regras simples; ou em subdivisões $[x_i, x_{i+1}]$

do intervalo $[a, b]$, formando regras compostas. Considere-se a partição do intervalo $[a, b]$ em subintervalos, de comprimento h , $A_i = [a_i, a_{i+1}]$, $i = 0, 1, \dots, n-1$. Assim $w(A_i) = h = (b-a)/n$.

As *fórmulas fechadas de Newton-Cotes* são fórmulas de integração do tipo

$$a_0 = a, a_n = b$$

e

$$\int_a^b f(x)dx = \int_{a_0}^{a_n} f(x)dx \approx C_0 f(a_0) + C_1 f(a_1) + \dots + C_n f(a_n) = \sum_{i=0}^n C_i f(a_i)$$

sendo os coeficientes C_i determinados de acordo com o grau do polinômio aproximador. O método de Simpson intervalar, Caprani et al. [CMN02], é uma extensão intervalar do método de Simpson real apresentado em [BurFai03, RugLop96]. O método de Simpson intervalar é fundamentado na propriedade aditiva da integral definida e no teorema do valor médio para integrais [Rudin71]. Supondo que f é três vezes continuamente derivável em $A = [a, b]$, o método encontra um intervalo que encapsula a integral definida, isto é, um intervalo que contém o valor da integral definida, como será visto a seguir.

2.5.1 O Método de Simpson Real

O método de integração de Simpson é uma fórmula fechada de Newton-Cotes que aproxima f por um polinômio de grau 2 (pode-se usar o polinômio interpolador de Lagrange) em $A = [a, b]$. Para se interpolar f por um polinômio de grau 2 precisa-se de três pontos para a construção da fórmula da regra simples. Sejam $a_0 = a$, $d_m = m(A)$ e $a_2 = b$ os três pontos do intervalo A . Seja p o polinômio de Lagrange que interpola f nos pontos a_0 , d_m e a_2 :

$$p(x) = \frac{(x-d_m)(x-a_2)}{(a_0-d_m)(a_0-a_2)} f(a_0) + \frac{(x-a_0)(x-a_2)}{(d_m-a_0)(d_m-a_2)} f(d_m) + \frac{(x-a_0)(x-d_m)}{(a_2-a_0)(a_2-d_m)} f(a_2).$$

Assim,

$$S = \int_a^b f(x)dx \cong \int_a^b p(x)dx.$$

Resolvendo a integral definida do polinômio p , tem-se a Regra Simples de Simpson:

$$S = \int_a^b f(x)dx \cong \frac{h}{6} [f(a) + 4f(m(A)) + f(b)],$$

com $h = b - a$.

Teorema 12. *Seja f três vezes continuamente diferenciável em $[a, b]$, então o erro de truncamento da Regra Simples de Simpson é dado por:*

$$E_s = -\frac{h^5}{2880}f^{(4)}(\varepsilon), \varepsilon \in (a, b).$$

Portanto,

$$S = \int_a^b f(x)dx = \frac{h}{6}[f(a) + 4f(m(A)) + f(b)] - \frac{h^5}{2880}f^{(4)}(\varepsilon).$$

Dada uma partição $P = \{a_0, a_1, \dots, a_n\}$ de $A = [a, b]$ tal que $a_i - a_{i-1} = (b - a)/n$, então pela propriedade aditiva da integral tem-se:

$$S = \sum_{i=1}^n S_i = \sum_{i=1}^n \int_{a_{i-1}}^{a_i} f(x)dx. \quad (2.15)$$

A Regra Composta de Simpson consiste em aplicar a Regra Simples de Simpson a cada subintervalo $A_i = [a_{i-1}, a_i], i = 1, \dots, n$ e é dada por:

$$S = \sum_{i=1}^n \int_{a_{i-1}}^{a_i} f(x)dx = \sum_{i=1}^n S_i \cong \sum_{i=1}^n \frac{w(A_i)}{6}(f(a_{i-1}) + 4f(m(A_i)) + f(a_i)) \quad (2.16)$$

Supondo-se f três vezes continuamente diferenciável, tem-se que

$$S = \sum_{i=1}^n S_i = \sum_{i=1}^n \left\{ \frac{w(A_i)}{6}(f(a_{i-1}) + 4f(m(A_i)) + f(a_i)) - \frac{(w(A_i))^5}{2880}f^{(4)}(\varepsilon_i) \right\}, \quad (2.17)$$

onde $\varepsilon_i \in A_i$ e $m(A_i)$ é o ponto médio de A_i .

2.5.2 O Método de Simpson Intervalar

O método de Simpson intervalar descrito em Caprani et al. [CMN02] admite conhecidas as extensões intervalares inclusões monotônicas F e G de f e $f^{(4)}$, respectivamente. Como $f(x) \in F(X)$, para todo $x \in (X \in \mathbb{IR})$ então, por (2.15),

$$IS = \sum_{i=1}^n \int_{a_{i-1}}^{a_i} f(x)dx = \sum_{i=1}^n w(A_i)F(A_i). \quad (2.18)$$

Por (2.17) tem-se

$$IS_i = \int_{a_{i-1}}^{a_i} f(x)dx = \frac{w(A_i)}{6}(F(a_{i-1}) + 4F(m(A_i)) + F(a_i)) - \frac{(w(A_i))^5}{2880}G(A_i),$$

onde $F(a_{i-1}), F(m(A_i))$ e $F(a_i)$ são intervalos degenerados de números reais. Portanto,

$$S \subseteq IS = \sum_{i=1}^n IS_i. \quad (2.19)$$

Assumindo que G é uma extensão intervalar linear de $f^{(4)}$, tem-se

$$w(IS_i) = \frac{1}{2880} w(A_i)^5 w(G(A_i)) \leq \frac{1}{2880} w(A_i)^5 K_G w(A_i) = K_2 w(A_i)^6.$$

Se $w(A_i) = w(A)/n$ tem-se que

$$w(IS) = \sum_i w(IS_i) \leq K_3/n^5,$$

pode-se esperar que dobrando-se o valor de n reduz-se o erro de truncamento por um fator de 32.

O intervalo IS depende das extensões intervalares usadas e do valor de n .

Probabilidade Condicional Intervalar. Independência

Este capítulo é uma das contribuições desta tese. Neste, apresentam-se definições para calcular probabilidades condicionais intervalares e independência intervalar de eventos. Inicialmente prova-se uma relação de inclusão envolvendo estas probabilidades; prova-se ainda que a probabilidade condicional intervalar satisfaz um conjunto de propriedades similares às propriedades da probabilidade condicional dos reais. Os exemplos numéricos foram resolvidos usando-se o INTLAB [KulMir81, Rump06]. Como a probabilidade incondicional intervalar, a probabilidade condicional intervalar proposta mostrou ser uma alternativa teórica e numericamente correta para a probabilidade condicional real.

A probabilidade condicional intervalar apresentada nesta tese é fundamentada na probabilidade intervalar definida em Campos [Campos97].

3.1 Probabilidade Condicional

Nesta seção será proposta uma definição para uma probabilidade condicional intervalar. Para tanto, seguindo o que é feito no caso da probabilidade real, inicialmente será dada uma notação para a probabilidade condicional intervalar aqui proposta.

Sejam A e B eventos em um espaço amostral Ω , com $P(B) > 0$. Seja $P(A|B)$ a probabilidade condicional de A dado B . Denota-se por

$$P_{\text{IR}}(A|B)$$

à probabilidade condicional intervalar de A dado B .

A probabilidade intervalar proposta por Campos [Campos97] mostra que valores não representáveis de probabilidades pertencem a intervalos que, se necessário, podem ter amplitude tão pequena quanto possível (os intervalos de exatidão máxima). Este fato será usado na demonstração do Lema 2.

Exemplo 26. *Sejam A e B eventos tais que $P(A|B) = \frac{10}{19} = 0.5263$, então usando a probabilidade intervalar proposta por Campos [Campos97], a aritmética de exatidão máxima [KulMir81] e o*

INTLAB [Rump06], tem-se que

$$P_{\text{IRR}}(A | B) = [0.5263, 0.5264].$$

Lema 2. Sejam $P_{\text{IRR}}(A \cap B)$, $P_{\text{IRR}}(B)$ as probabilidades intervalares dos eventos $A \cap B$, B respectivamente, então

$$P_{\text{IRR}}(A | B) \subseteq \frac{P_{\text{IRR}}(A \cap B)}{P_{\text{IRR}}(B)}, \quad 0 \notin P_{\text{IRR}}(B).$$

Prova:

Sejam $P(A \cap B) = x$, $P(B) = z$, $P(A | B) = y$ e os intervalos de exatidão máxima $P_{\text{IRR}}(A \cap B) = [\nabla x, \Delta x]$, $P_{\text{IRR}}(B) = [\nabla z, \Delta z]$, $P_{\text{IRR}}(A | B) = [\nabla y, \Delta y]$. Da definição de probabilidade condicional real tem-se que

$$y = \frac{x}{z}.$$

Da aritmética intervalar tem-se que

$$y \in \frac{P_{\text{IRR}}(A \cap B)}{P_{\text{IRR}}(B)} = \left[\frac{\nabla x}{\Delta z}, \frac{\Delta x}{\nabla z} \right].$$

Sabe-se que:

o arredondamento monótono direcionado para baixo de y , (∇y) , é o maior número de máquina que é menor ou igual a y

e

o arredondamento monótono direcionado para cima de y , (Δy) , é o menor número de máquina maior ou igual a y .

Como

$$\nabla \left(\frac{\nabla x}{\Delta z} \right) \leq y \leq \Delta \left(\frac{\Delta x}{\nabla z} \right),$$

tem-se que

$$\nabla \left(\frac{\nabla x}{\Delta z} \right) \leq \nabla y$$

e

$$\Delta y \leq \Delta \left(\frac{\Delta x}{\nabla z} \right).$$

Consequentemente

$$[\nabla y, \Delta y] \subseteq \left[\nabla \left(\frac{\nabla x}{\Delta z} \right), \Delta \left(\frac{\Delta x}{\nabla z} \right) \right].$$

Portanto,

$$P_{\text{IRR}}(A | B) \subseteq \frac{P_{\text{IRR}}(A \cap B)}{P_{\text{IRR}}(B)}.$$

■

Exemplo 27. Suponha-se que um escritório possua 100 máquinas de calcular. Algumas dessas máquinas são elétricas (E), enquanto outras são manuais (M); e algumas são novas (N), enquanto outras são usadas (U). A Tabela 3.1 dá o número de máquinas de cada categoria. Uma pessoa entra no escritório, pega uma máquina ao acaso, e descobre que é nova. Qual será a probabilidade de que seja elétrica?

Tabela 3.1 Número de Máquinas.

	E	M
N	40	30
U	20	10

Solução: Deseja-se calcular $P(E|N)$. Tem-se

$$P(E \cap N) = \frac{4}{10}, P(N) = \frac{7}{10} \text{ e } P(E|N) = \frac{P(E \cap N)}{P(N)} = \frac{4}{7}.$$

Calculando-se as probabilidades intervalares tem-se

$$P_{\text{IRR}}(E \cap N) = [\nabla \frac{4}{10}, \triangle \frac{4}{10}] = [0.3999, 0.4001],$$

$$P_{\text{IRR}}(N) = [\nabla \frac{7}{10}, \triangle \frac{7}{10}] = [0.6999, 0.7001],$$

$$\frac{P_{\text{IRR}}(E \cap N)}{P_{\text{IRR}}(N)} = [0.3999, 0.4001] / [0.6999, 0.7001] = [0.5714, 0.5715],$$

e, considerando-se o espaço amostral reduzido,

$$P_{\text{IRR}}(E|N) = [\nabla \frac{4}{7}, \triangle \frac{4}{7}] = [0.5714, 0.5715].$$

Portanto,

$$P_{\text{IRR}}(E|N) \subseteq \frac{P_{\text{IRR}}(E \cap N)}{P_{\text{IRR}}(N)}$$

e

$$P(E|N) \in P_{\text{IRR}}(E|N).$$

Exemplo 28. Dois dados d_1 e d_2 são lançados. Sejam os eventos:

$$A = \{\text{O dado } d_1 \text{ apresenta resultado } 2\},$$

$$B = \{\text{A soma dos pontos nos dois dados é } 6\}.$$

Se a soma é 6, qual a probabilidade do dado d_1 apresentar resultado 2?

Da probabilidade real tem-se que

$$P(A \cap B) = \frac{1}{36}, P(B) = \frac{5}{36}, \text{ e } P(A|B) = \frac{P(A \cap B)}{P(B)} = \frac{1}{5}.$$

Calculando-se as probabilidades intervalares tem-se que

$$P_{\text{IR}}(A \cap B) = [0.0277, 0.0278],$$

$$P_{\text{IR}}(B) = [0.1388, 0.1389],$$

$$\frac{P_{\text{IR}}(A \cap B)}{P_{\text{IR}}(B)} = [0.1999, 0.2001],$$

$$P_{\text{IR}}(A|B) = [0.1999, 0.2001].$$

Portanto,

$$P_{\text{IR}}(A|B) \subseteq \frac{P_{\text{IR}}(A \cap B)}{P_{\text{IR}}(B)}$$

e

$$P(A|B) \in P_{\text{IR}}(A|B).$$

Exemplo 29. De um baralho de 52 cartas, uma é extraída e observa-se que seu número está entre 4 e 10 (4 e 10 inclusive). Qual a probabilidade de que o número da carta seja 6?

Solução: Sejam os eventos

$$A = \{\text{o número da carta está entre 4 e 10 (4 e 10 inclusive)}\},$$

e

$$B = \{\text{o número da carta é 6}\},$$

Da probabilidade real tem-se que

$$P(A \cap B) = \frac{1}{13}, P(A) = \frac{7}{13}, \text{ e } P(B|A) = \frac{P(A \cap B)}{P(A)} = \frac{1}{7}.$$

Calculando-se as probabilidades intervalares tem-se que

$$P_{\text{IR}}(A \cap B) = [0.0769, 0.0770],$$

$$P_{\text{IR}}(A) = [0.5384, 0.5385],$$

$$\frac{P_{\text{IR}}(A \cap B)}{P_{\text{IR}}(A)} = [0.1428, 0.1429],$$

$$P_{\text{IR}}(B|A) = [0.1428, 0.1429].$$

Portanto,

$$P_{\text{IR}}(B|A) \subseteq \frac{P_{\text{IR}}(A \cap B)}{P_{\text{IR}}(A)}$$

e

$$P(B|A) \in P_{\text{IR}}(B|A).$$

Lema 3. *Seja B um eventos em um espaço amostral Ω com $0 \notin P_{\text{IR}}(B)$. Então a função*

$$p(X, B) = \frac{P_{\text{IR}}(X \cap B)}{P_{\text{IR}}(B)}$$

possui as seguintes propriedades:

- (i) *Para qualquer evento A . $P(A|B) \in p(A, B)$*
- (ii) *$1 \in p(\Omega, B)$;*
- (iii) *$0 \in p(\emptyset, B)$;*
- (iv) *Se $A_1 \cap A_2 = \emptyset$ e $A_i \neq \emptyset, i = 1, 2$, então $p((A_1 \cup A_2), B) = p(A_1, B) + p(A_2, B)$;*
- (v) *Dada a sequência de eventos $A_1, A_2, \dots$, com $A_i \cap A_j = \emptyset, \forall i \neq j$ e $A_i \neq \emptyset, \forall i$, então*

$$p((A_1 \cup A_2 \cup \dots), B) = p(A_1, B) + p(A_2, B) + \dots.$$

Prova:

- (i) *Seja $A \in \mathcal{A}$. Sejam $P(A \cap B) = x, P(B) = z$ probabilidades reais e $P_{\text{IR}}(A \cap B) = X, P_{\text{IR}}(B) = Z$ as probabilidades intervalares correspondentes. Da definição de probabilidade condicional real tem-se que*

$$y = P(A|B) = \frac{x}{z},$$

da aritmética intervalar e da definição de probabilidade intervalar tem-se que

$$y \in \frac{P_{\text{IR}}(X \cap B)}{P_{\text{IR}}(B)}.$$

Portanto, $P(A|B) = y \in p(A, B)$.

- (ii) *Sejam $P(B) = z$ e $P_{\text{IR}}(B) = Z$. Então,*

$$p(\Omega, B) = \frac{P_{\text{IR}}(\omega \cap B)}{P_{\text{IR}}(B)} = \frac{P_{\text{IR}}(B)}{P_{\text{IR}}(B)} = \frac{Z}{Z}.$$

Logo $1 \in p(\Omega, B)$.

(iii) Seja $P(B) = z$ e $P_{\text{IRR}}(B) = Z$. Então,

$$p(\emptyset, B) = \frac{P_{\text{IRR}}(\emptyset \cap B)}{P_{\text{IRR}}(B)} = \frac{P_{\text{IRR}}(\emptyset)}{P_{\text{IRR}}(B)} = \frac{0_{\varepsilon}}{Z}.$$

Logo $0 \in p(\emptyset, B)$.

(iv) Dadas as probabilidades intervalares $P_{\text{IRR}}(A_1 \cap B)$, $P_{\text{IRR}}(A_2 \cap B)$ e $P_{\text{IRR}}(B)$. Então,

$$\begin{aligned} p((A_1 \cup A_2), B) &= \frac{P_{\text{IRR}}((A_1 \cup A_2) \cap B)}{P_{\text{IRR}}(B)} \\ &= \frac{P_{\text{IRR}}((A_1 \cap B) \cup (A_2 \cap B))}{P_{\text{IRR}}(B)} \\ &= \frac{P_{\text{IRR}}(A_1 \cap B) + P_{\text{IRR}}(A_2 \cap B)}{P_{\text{IRR}}(B)} \\ &= \frac{P_{\text{IRR}}(A_1 \cap B)}{P_{\text{IRR}}(B)} + \frac{P_{\text{IRR}}(A_2 \cap B)}{P_{\text{IRR}}(B)} \\ &= p(A_1, B) + p(A_2, B) \end{aligned}$$

Logo $p((A_1 \cup A_2), B) = p(A_1, B) + p(A_2, B)$.

(v) Dadas as probabilidades intervalares $P_{\text{IRR}}(A_i \cap B)$, $\forall i$, e $P_{\text{IRR}}(B)$. Então,

$$\begin{aligned} p((A_1 \cup A_2 \cup \dots), B) &= \frac{P_{\text{IRR}}((A_1 \cup A_2 \cup \dots) \cap B)}{P_{\text{IRR}}(B)} \\ &= \frac{P_{\text{IRR}}((A_1 \cap B) \cup (A_2 \cap B) \cup \dots)}{P_{\text{IRR}}(B)} \\ &= \frac{P_{\text{IRR}}(A_1 \cap B)}{P_{\text{IRR}}(B)} + \frac{P_{\text{IRR}}(A_2 \cap B)}{P_{\text{IRR}}(B)} + \dots \\ &= p(A_1, B) + p(A_2, B) + \dots \end{aligned}$$

Logo $p((A_1 \cup A_2 \cup \dots), B) = p(A_1, B) + p(A_2, B) + \dots$.

Prova-se assim que a função p acima satisfaz os axiomas do espaço de probabilidade intervalar propostos por Campos [Campos97]. De acordo com os resultados teóricos anteriores e os exemplos ilustrativos propõe-se, nesta tese, uma definição abaixo como sendo a de probabilidade condicional intervalar, isto é,

$$p(X, B) := P_{\text{IRR}}(A | B).$$

Definição 3.1.1. Sejam A e B eventos. Se $0 \notin P_{\text{IRR}}(B)$ a **probabilidade condicional intervalar** é definida por

$$P_{\text{IRR}}(A|B) = \frac{P_{\text{IRR}}(A \cap B)}{P_{\text{IRR}}(B)}.$$

3.2 Independência

Esta seção mostra uma propriedade de eventos independentes no contexto intervalar.

Lema 4. Se A e B são eventos independentes, então

$$P_{\text{IRR}}(A \cap B) \subseteq P_{\text{IRR}}(A) \cdot P_{\text{IRR}}(B).$$

Prova:

Pelo Lema 3,

$$P(A) = x \Rightarrow P_{\text{IRR}}(A) = X = [\nabla x, \Delta x];$$

$$P(B) = y \Rightarrow P_{\text{IRR}}(B) = Y = [\nabla y, \Delta y];$$

$$P(A \cap B) = z \Rightarrow P_{\text{IRR}}(A \cap B) = Z = [\nabla z, \Delta z].$$

Então

$$P_{\text{IRR}}(A) \cdot P_{\text{IRR}}(B) = [\nabla(\nabla x \nabla y), \Delta(\Delta x \Delta y)],$$

desde que $\nabla x \geq 0$ e $\nabla y \geq 0$.

Sejam $x \in X$, $y \in Y$ e $z \in Z$. Pelo princípio da inclusão da aritmética intervalar, $xy \in X \cdot Y$ e $z \in X \cdot Y$, tem-se

$$\nabla x \nabla y \leq \Delta x \Delta y.$$

Como $z \in Z = [\nabla z, \Delta z]$ então

$$\nabla x \nabla y \leq \nabla z \leq z \leq \Delta z \leq \Delta x \Delta y.$$

Logo,

$$[\nabla z, \Delta z] \subseteq [\nabla x \nabla y, \Delta x \Delta y],$$

e portanto

$$P_{\text{IRR}}(A \cap B) \subseteq P_{\text{IRR}}(A) \cdot P_{\text{IRR}}(B).$$

■

Exemplo 30. Um circuito eletrônico consiste de um transistor e de um condensador. O transistor é selecionado entre 100, 10 dos quais são defeituosos, e o condensador é selecionado a partir de um lote de 300, 15 dos quais são defeituosos. Seja A o evento "o transistor do circuito é defeituoso", $P(A) = 0,10$ e B o evento "o condensador do circuito é defeituoso", $P(B) = 0,05$. Determine $P(A \cap B)$.

Solução: Tem-se que $P(A \cap B) = \frac{10 \cdot 15}{100 \cdot 300} = 0,005 = (0,10) \cdot (0,05) = P(A) \cdot P(B)$. Logo os eventos A e B são independentes. Calculando-se as probabilidades intervalares tem-se

$$P_{\text{IR}}(A) = [0,0999, 0,1001],$$

$$P_{\text{IR}}(B) = [0,0499, 0,0501],$$

$$P_{\text{IR}}(A \cap B) = [0,0049, 0,0051],$$

$$P_{\text{IR}}(A) \cdot P_{\text{IR}}(B) = [0,0049, 0,0051].$$

Logo

$$P_{\text{IR}}(A \cap B) \subseteq P_{\text{IR}}(A) \cdot P_{\text{IR}}(B)$$

Exemplo 31. Um baralho de 52 cartas é subdividido em 4 naipes: copas, espadas, ouros e paus. Retirando-se duas cartas com reposição, qual a probabilidade de ser a primeira de ouros e a segunda de copas?

Solução: Sejam E o evento "sair a primeira carta de ouros" e F o evento "sair a segunda carta de copas". Tem-se que

$$P(E) = \frac{1}{4} = 0,2500 \quad \text{e} \quad P(F) = \frac{1}{4} = 0,2500.$$

Como as retiradas são feitas com reposição, os eventos E e F são independentes. Logo

$$P(E \cap F) = \frac{1}{16} = 0,0625 = P(E) \cdot P(F).$$

Calculando-se as probabilidades intervalares tem-se

$$P_{\text{IR}}(E) = [0,2499, 0,2501],$$

$$P_{\text{IR}}(F) = [0,2499, 0,2501],$$

$$P_{\text{IR}}(E \cap F) = [0,0624, 0,0626],$$

$$P_{\text{IR}}(E) \cdot P_{\text{IR}}(F) = [0,0624, 0,0626].$$

Logo

$$P_{\text{IR}}(E \cap F) \subseteq P_{\text{IR}}(E) \cdot P_{\text{IR}}(F)$$

Intervalos Encapsuladores para Probabilidades de Variáveis Aleatórias Contínuas

O objetivo deste capítulo é usar o método de Simpson intervalar, definido por Caprani et al. [CMN02], para calcular probabilidades intervalares ou intervalos encapsuladores para as variáveis aleatórias contínuas. Após o desenvolvimento das expressões teóricas de como calcular os intervalos encapsuladores para as variáveis aleatórias estudadas, os algoritmos para obtenção dos intervalos foram implementados no Matlab [MatLab04], usando o IntLab [Rump06] e encontram-se no Apêndice A.

Para as variáveis Exponencial e Normal os intervalos encapsuladores ou as probabilidades validadas foram obtidos com o método de Simpson intervalar. Para a Uniforme o intervalo encapsulador foi definido a partir da função densidade, pois a derivada de ordem quatro necessária no método de Simpson Intervalar é nula.

Apresenta-se também uma versão intervalar da tabela da Normal baseada na tabela real encontrada em [Meyer83].

4.1 Cálculo de Probabilidades de Variáveis Aleatórias Contínuas

A distribuição de probabilidade de uma variável aleatória contínua, ver [Burrill72, Feller68, James06, Meyer83], é caracterizada por sua função densidade de probabilidade, a qual satisfaz às propriedades:

- (i) $f(x) \geq 0$,
- (ii) $\int_a^b f(x)dx = P(a \leq x \leq b)$, $a < b$,
- (iii) $\int_{-\infty}^{\infty} f(x)dx = 1$.

O item (ii) indica que a probabilidade da variável aleatória assumir valor em um intervalo é dada pela integral da função nesse intervalo. Entretanto, o cálculo dessa probabilidade implica em resolver dois tipos de problemas: (i) encontrar primitivas na forma analítica, o que é possível no

caso da Uniforme e Exponencial, mas não da Normal e (ii) o valor da probabilidade, em geral, é real, não necessariamente representável em computadores. Lembrando que apenas um subconjunto finito dos racionais [Forsythe77, Goldberg91] pode ser representado e processado internamente nas máquinas.

4.1.1 Intervalo Encapsulador para a Probabilidade de uma Variável Aleatória com Distribuição Exponencial

Uma variável aleatória contínua com função densidade dada por

$$f(x) = \begin{cases} \alpha e^{-\alpha x}, & 0 \leq x < \infty, \\ 0, & x < 0, \end{cases}$$

com $\alpha \in \mathbb{R}_+^*$, é uma variável aleatória Exponencial.

Seja $A = [a, b] \subset \mathbb{R}$. Para calcular $P(a < x \leq b)$, tem-se as seguintes *situações possíveis*:

(i) Se $b \leq 0$, então

$$P(a < x \leq b) = 0.$$

(ii) Se $0 \in (a, b)$, então

$$P(a < x \leq b) = P(0 \leq x \leq b) = \int_0^b \alpha e^{-\alpha x} dx.$$

(iii) Se $a \geq 0$, então

$$P(a \leq x \leq b) = \int_a^b \alpha e^{-\alpha x} dx.$$

A função f tem primitiva na forma analítica, então, pelo Teorema Fundamental do Cálculo Integral, pode-se calcular a integral definida de f diretamente desta primitiva. O número e e suas potências, de expoentes não nulos, são irracionais, sendo seus valores aproximados, portanto sujeitos a erros de arredondamento.

O método de Simpson intervalar é uma alternativa para encontrar um intervalo que aproxime o valor desta integral, no sentido de se obter um intervalo com amplitude tão pequena quanto possível que a contenha. Para aplicá-lo é necessário determinar extensões intervalares para a densidade f e sua derivada de ordem quatro,

$$f^{(4)}(x) = \begin{cases} \alpha^5 e^{-\alpha x}, & 0 < x < \infty, \\ 0, & x < 0, \end{cases}$$

Como a probabilidade da variável aleatória X assumir valores negativos é 0 pode-se restringir, na aplicação do método de Simpson intervalar, o domínio de f e de $f^{(4)}$ ao conjunto $\mathbb{R}_+$. Sejam

$f|_{\mathbb{R}_+}$ e $f^{(4)}|_{\mathbb{R}_+}$ as restrições de f e $f^{(4)}$, isto é

$$f|_{\mathbb{R}_+}(x) = \alpha e^{-\alpha x}, \quad 0 \leq x < \infty,$$

e

$$f^{(4)}|_{\mathbb{R}_+}(x) = \alpha^5 e^{-\alpha x}, \quad 0 < x < \infty,$$

são decrescentes e as funções intervalares

$$F_E(X) = \alpha[e^{-\alpha \bar{x}}, e^{-\alpha \underline{x}}], \quad \underline{x} > 0$$

e

$$G_E(X) = \alpha^5[e^{-\alpha \bar{x}}, e^{-\alpha \underline{x}}], \quad \underline{x} > 0,$$

são extensões intervalares naturais para $f|_{\mathbb{R}_+}$ e $f^{(4)}|_{\mathbb{R}_+}$, respectivamente.

Observe que:

$$F_E(X) = f|_{\mathbb{R}_+}(X) = \tilde{f}|_{\mathbb{R}_+}(X) = \bar{f}|_{\mathbb{R}_+}(X)$$

e

$$G_E(X) = f^{(4)}|_{\mathbb{R}_+}(X) = \tilde{f}^{(4)}|_{\mathbb{R}_+}(X) = \bar{f}^{(4)}|_{\mathbb{R}_+}(X) \quad (4.1)$$

Teorema 13. G_E é uma extensão intervalar monotônica linear de $f^{(4)}|_{\mathbb{R}_+}$.

Prova: De fato, por 4.1,

$$d(G_E(Y), \tilde{f}^{(4)}|_{\mathbb{R}_+}(Y)) = 0 \leq Kw(Y),$$

$\forall K > 0, \forall Y \subseteq X$. ■

Os intervalos encapsuladores para o cálculo de probabilidades, usando o método de Simpson intervalar, foram implementados no IntLab, como arquivo M de funções, sendo sua chamada dada por

$$dexp(a, b, c, p). \quad (4.2)$$

Os parâmetros do procedimento $dexp$ são:

a = limite inferior do intervalo A ,

b = limite superior do intervalo A ,

c = o parâmetro α da função densidade f ,

p = número de subintervalos da partição de A .

Assume-se que $F_E(a_{i-1})$, $F_E(m(A_i))$ e $F_E(a_i)$ são intervalos degenerados de números reais. Os a_i correspondem aos pontos da partição do intervalo A , e $m(A_i)$ é o ponto médio do intervalo A_i .

O algoritmo a seguir sumariza o procedimento de obtenção de probabilidades quando a distribuição é a Exponencial.

Algoritmo Exponencial**(E1)** Obtenção das extensões intervalares $F_E(X)$ e $G_E(X)$;**(E2)** Instanciação de: a, b, c, p ;**(E3)** Realização do procedimento: $dexp(a, b, c, p)$.

Na implementação do algoritmo foram analisadas todas as situações possíveis observadas no início desta seção, bem como o tempo gasto para processá-lo.

Os exemplos a seguir ilustram aplicações do método aqui proposto. As soluções foram calculadas no Matlab [MatLab04], usando a biblioteca IntLab [Rump06] e a Função 4.2, nos formatos `short` (precisão simples) e `long` (precisão dupla).

Exemplo 32. *O número de defeitos de um tecido segue uma lei de Poisson com média de um defeito a cada 500m. Qual a probabilidade que o intervalo entre dois defeitos consecutivos*

(a) *seja no mínimo 1250m;*

(b) *esteja entre 1000 e 1250m;*

(c) *seja menor do que 1000m.*

Solução. Se, X , o número de defeitos é Poisson, então a distância, D , entre dois defeitos consecutivos é Exponencial com parâmetro $\alpha = \frac{1}{500} = 0.002$. Portanto,

(a) (i) Probabilidade real: $P(D \geq 1250)$

Precisão simples: $0.0821 = p_1$,

Precisão dupla: $0.08208499862390 = p_2$.

(ii) **Probabilidade encapsulada:** $P_{\text{IIR}}(D) = \text{dexp}(1250, \text{inf}, 1/500, 100)$.

Precisão simples: $[0.0820, 0.0821] = P_1$,

Precisão dupla: $[0.08208499862389, 0.08208499862390] = P_2$.

Tempo de Processamento: 0.0780s.

(b) (i) Probabilidade real: $P(1000 < D < 1250)$

Precisão simples: $0.0532 = p_3$,

Precisão dupla: $0.05325028461271 = p_4$.

(ii) **Probabilidade encapsulada:** $P_{\text{IIR}}(D) := \text{dexp}(1000, 1250, 1/500, 100)$.

Precisão simples: $[0.0532, 0.0533] = P_3$,

Precisão dupla: $[0.05325028461271, 0.05325028461272] = P_4$.

Tempo de Processamento: 2.500s.

(c) (i) Probabilidade real: $P(D < 1000)$

Precisão simples: $0.8647 = p_5$,

Precisão dupla: $0.86466471676339 = p_6$.

(ii) Probabilidade encapsulada: $P_{\text{IIR}}(D) := \text{dexp}(-\text{inf}, 1000, 1/500, 100)$.

Precisão simples: $[0.8646, 0.8647] = P_5$,

Precisão dupla: $[0.86466471676338, 0.86466471676339] = P_6$.

Tempo de Processamento: 0.0620s.

Como pode ser visto, $p_1 \in P_1$, $p_2 \in P_2$, $p_3 \in P_3$, $p_5 \in P_5$ e $p_6 \in P_6$.

4.1.2 Distribuição Normal

Uma variável aleatória X tem uma distribuição Normal se sua função densidade de probabilidade for

$$f_1(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \quad -\infty \leq x \leq \infty, \quad (4.3)$$

onde os parâmetros μ e σ devem satisfazer às condições $-\infty < \mu < \infty$, $\sigma > 0$. Prova-se [Meyer83] que μ e σ são, respectivamente, a média e o desvio-padrão de X . A notação usual é $X \sim N(\mu, \sigma^2)$. Se X tiver distribuição $N(\mu, \sigma^2)$, $\frac{X-\mu}{\sigma}$ terá distribuição $N(0, 1)$. Se $A = [a, b] \subset \mathbb{R}$ e $X \sim N(0, 1)$, então

$$P(a \leq X \leq b) = \frac{1}{\sqrt{2\pi}} \int_a^b e^{-\frac{x^2}{2}} dx.$$

A função $f(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$ não tem primitiva na forma analítica portanto, não é possível aplicar o Teorema Fundamental do Cálculo Integral. Contudo, métodos de integração numérica podem ser empregados para calcular integrais da forma acima e gerar tabelas de probabilidades.

O método proposto neste trabalho consiste em utilizar na Equação 2.19 extensões intervalares F e G para f e $f^{(4)}$, respectivamente, que, quando computados, produzam intervalos encapsuladores para a probabilidade procurada.

A derivada de f de quarta ordem é dada por

$$f^{(4)}(x) = \frac{e^{-\frac{x^2}{2}}}{\sqrt{2\pi}} (x^4 - 6x^2 + 3).$$

Observa-se que as funções f e $f^{(4)}$ não são globalmente monótonas. Assim, uma dificuldade inicial é verificar quais os intervalos da reta onde f e $f^{(4)}$ são (localmente) monótonas.

Construindo extensões intervalares inclusões monotônicas para f e $f^{(4)}$

Estas construções são fundamentadas nos Exemplos 24 e 25.

- (a) Determinação de uma extensão intervalar inclusão monotônica para f .

Sabe-se que 0 é o único extremo relativo (que é também absoluto) de f , e que à esquerda de 0, f é crescente e, à direita, f é decrescente. Pelo Exemplo 25 a função F_N definida por

$$F_N(X) = \frac{1}{\sqrt{2\pi}} \begin{cases} \left[\frac{-\bar{x}^2}{e^{\frac{x^2}{2}}}, \frac{-x^2}{e^{\frac{\bar{x}^2}{2}}} \right], & \text{se } \underline{x} > 0, \\ \left[\frac{-x^2}{e^{\frac{x^2}{2}}}, \frac{-\bar{x}^2}{e^{\frac{\bar{x}^2}{2}}} \right], & \text{se } \bar{x} < 0, \\ \left[e^{-\max\{\frac{x^2}{2}, \frac{\bar{x}^2}{2}\}}, 1 \right], & \text{se } 0 \in X, \end{cases}$$

é uma extensão intervalar inclusão monotônica para f .

- (b) Determinação de uma extensão intervalar inclusão monotônica para $f^{(4)}$.

Sabe-se que os extremos relativos de $f^{(4)}$ são: $b_1 = -\sqrt{5 + \sqrt{10}}$, $b_5 = \sqrt{5 + \sqrt{10}}$, $b_3 = 0$ (os máximos relativos), e $b_2 = -\sqrt{5 - \sqrt{10}}$, $b_4 = \sqrt{5 - \sqrt{10}}$ (os mínimos relativos). Tem-se ainda que $\min(D(f^{(4)})) = -\infty$ e $\max(D(f^{(4)})) = \infty$. Sejam $B_1, B_2, \dots, B_5$ intervalos disjuntos, tais que $b_i \in B_i$ e $B_i = [b_i - \delta_i, b_i + \delta_i]$. Os C_i são dados por: $C_1 = [-\infty, b_1 - \delta_1]$, $C_6 = [b_5 + \delta_5, +\infty]$ e $C_i = [b_{i-1} + \delta_{i-1}, b_i - \delta_i]$, $i = 2, \dots, 5$, onde δ_i são reais positivos.

Desta forma, o domínio de $f^{(4)}$ pode ser escrito por:

$$D(f^{(4)}) = C_1 \cup B_1 \cup C_2 \cup B_2 \cup C_3 \cup B_3 \cup C_4 \cup B_4 \cup C_5 \cup B_5 \cup C_6.$$

Dado $X \subset D(f^{(4)})$, encontra-se, como no Exemplo 25 os D_i tais que,

$$X = D_1 \cup D_2 \cup \dots \cup D_n, \quad n \leq 11.$$

Se $D_i = [\underline{d}, \bar{d}] \subset C_l$ faça

$$G([\underline{d}, \bar{d}]) = \frac{1}{\sqrt{2\pi}} [\exp(-((\underline{d})^2)/2))((\underline{d})^4 - 6(\underline{d})^2 + 3), \exp(-((\bar{d})^2)/2))((\bar{d})^4 - 6(\bar{d})^2 + 3)],$$

se $f^{(4)}$ é crescente em D_i , ou

$$G([\underline{d}, \bar{d}]) = \frac{1}{\sqrt{2\pi}} [\exp(-((\bar{d})^2)/2))((\bar{d})^4 - 6(\bar{d})^2 + 3), \exp(-((\underline{d})^2)/2))((\underline{d})^4 - 6(\underline{d})^2 + 3)],$$

se $f^{(4)}$ é decrescente em D_i .

Observe que em ambos os casos,

$$\tilde{f}^{(4)}(D_i) = \bar{f}^{(4)}(D_i) = G(D_i).$$

Se $D_i = [\underline{d}, \bar{d}] \subset B_j$ faça

$$G([\underline{d}, \bar{d}]) = \frac{1}{\sqrt{2\pi}} \exp(-((D_i)^2)/2)((D_i)^4 - 6(D_i)^2 + 3),$$

que é uma extensão intervalar natural inclusão monotônica de $f^{(4)}$. Tem-se, portanto, a seguinte função intervalar

$$G(X) = G(D_1) \cup G(D_2) \cup \dots \cup G(D_n), \quad n \leq 11$$

que é uma extensão intervalar inclusão monotônica para a função $f^{(4)}$, pelo Exemplo 25.

As expressões acima serão utilizadas para calcular as probabilidades intervalares que aparecem na Tabela $N_{\text{IIR}}(0, 1)$, a seguir. Esta é uma tabela de probabilidades encapsuladas para uma variável $X \sim N(0, 1)$. Nesta, os dígitos menores que aparecem na última posição significam os limites inferior e superior, respectivamente, dos intervalos. Por exemplo, 0.9937^8_7 é o intervalo $[0.9937, 0.9938]$, o qual significa que

$$P_{\text{IIR}}(X \leq 2.5) = [0.9937, 0.9938].$$

A probabilidade real [Meyer83] é $P(X \leq 2.5) = 0.9938$. Esta tabela foi construída com o Algoritmo Normal1 fazendo a e b variarem em $\mathbb{R}$ com $a < b$. Os B_i , $i = 1, 2, \dots, 5$, foram calculados usando-se uma versão do Método de Newton intervalar, de tal forma que $w(B_i) < 10^{-10}$.

Os intervalos encapsuladores foram implementados como arquivo M de funções, sendo sua chamada dada por

$$\text{normal1}(a, b, p), \tag{4.4}$$

Os parâmetros do procedimento $\text{normal1}(a, b, p)$ são:

a = limite inferior do intervalo A ,

b = limite superior do intervalo A ,

p = número de subintervalos da partição de A .

Assim como na Exponencial, assume-se que $F(a_{i-1})$, $F(m(A_i))$ e $F(a_i)$ são intervalos degenerados de números reais.

O algoritmo abaixo contém os passos para a computação de probabilidades intervalares para a distribuição Normal.

Algoritmo Normal

- (N1) Obtenção das extensões intervalares $F_N(X)$ e $G(X)$;
- (N2) Instanciação de: a , b e p ;
- (N3) Realização do procedimento: $normal1(a,b,p)$.

Tabela $N_{\text{IIR}}(0, 1)$ Probabilidades Encapsuladas para a Função de Distribuição Acumulada da $N(0, 1)$

z	0	1	2	3	4	5	6	7	8	9
-3.0	0.001 ₃ ⁴	0.001 ₃ ⁴	0.001 ₂ ³	0.001 ₂ ³	0.001 ₁ ²	0.001 ₁ ²	0.001 ₁ ²	0.001 ₀ ¹	0.001 ₀ ¹	0.001 ₀ ¹
-2.9	0.001 ₈ ⁹	0.001 ₈ ⁹	0.001 ₇ ⁸	0.001 ₈ ⁸	0.001 ₆ ⁷	0.001 ₅ ⁶	0.001 ₅ ⁶	0.001 ₄ ⁵	0.001 ₄ ⁵	0.001 ₃ ⁴
-2.8	0.002 ₅ ⁶	0.002 ₄ ⁵	0.002 ₄ ⁵	0.002 ₃ ⁴	0.002 ₂ ³	0.002 ₁ ²	0.002 ₁ ²	0.002 ₀ ¹	0.002 ₁₉ ²⁰	0.002 ₁₉ ²⁰
-2.7	0.003 ₄ ⁵	0.003 ₃ ⁴	0.003 ₂ ³	0.003 ₁ ²	0.003 ₀ ¹	0.003 ₂₉ ³⁰	0.002 ₈ ⁹	0.002 ₈ ⁹	0.002 ₇ ⁸	0.002 ₆ ⁷
-2.6	0.004 ₆ ⁷	0.004 ₅ ⁶	0.004 ₃ ⁴	0.004 ₂ ³	0.004 ₁ ²	0.004 ₀ ¹	0.004 ₃₉ ⁴⁰	0.003 ₇ ⁸	0.003 ₆ ⁷	0.003 ₅ ⁶
-2.5	0.006 ₂ ³	0.006 ₀ ¹	0.005 ₈ ⁹	0.005 ₇ ⁸	0.005 ₅ ⁶	0.005 ₃ ⁴	0.005 ₂ ³	0.005 ₀ ¹	0.005 ₄₉ ⁵⁰	0.004 ₇ ⁸
-2.4	0.008 ₁ ²	0.008 ₇₉ ⁸⁰	0.007 ₇ ⁸	0.007 ₆ ⁵	0.007 ₃ ⁴	0.007 ₁ ²	0.007 ₆₉ ⁷⁰	0.006 ₇ ⁸	0.006 ₅ ⁶	0.006 ₃ ⁴
-2.3	0.010 ₇ ⁸	0.010 ₅ ⁶	0.010 ₁ ²	0.010 ₉₉ ¹⁰⁰	0.009 ₆ ⁷	0.009 ₃ ⁴	0.009 ₁ ²	0.008 ₈ ⁹	0.008 ₆ ⁷	0.008 ₄ ⁵
-2.2	0.014 ₃₉ ⁴⁰	0.013 ₅ ⁶	0.013 ₂ ³	0.012 ₈ ⁹	0.012 ₅ ⁶	0.012 ₂ ³	0.012 ₁₉ ²⁰	0.011 ₆ ⁷	0.011 ₃ ⁴	0.011 ₀ ¹
-2.1	0.017 ₈ ⁹	0.017 ₄ ⁵	0.017 ₀ ¹	0.016 ₅ ⁶	0.016 ₁ ²	0.015 ₇ ⁸	0.015 ₃ ⁴	0.015 ₀ ¹	0.014 ₆ ⁷	0.014 ₂ ³
-2.0	0.022 ₇ ⁸	0.022 ₂ ³	0.021 ₆ ⁷	0.021 ₁ ²	0.020 ₆ ⁷	0.020 ₁ ²	0.019 ₆ ⁷	0.019 ₂ ³	0.018 ₇ ⁸	0.018 ₃ ⁴
-1.9	0.028 ₇ ⁸	0.028 ₀ ¹	0.027 ₄ ⁵	0.026 ₈ ⁹	0.026 ₁ ²	0.025 ₅ ⁶	0.025 ₄₉ ⁵⁰	0.024 ₅ ⁶	0.023 ₈ ⁹	0.023 ₂ ³
-1.8	0.036 ₅₉ ⁶⁰	0.035 ₁ ²	0.034 ₃ ⁴	0.033 ₆ ⁷	0.032 ₈ ⁹	0.032 ₁ ²	0.031 ₄ ⁵	0.030 ₇ ⁸	0.030 ₀ ¹	0.029 ₃ ⁴
-1.7	0.044 ₅ ⁶	0.043 ₆ ⁷	0.042 ₇ ⁸	0.041 ₈ ⁹	0.041 ₀₉ ¹⁰	0.040 ₀ ¹	0.039 ₂ ³	0.038 ₃ ⁴	0.037 ₅ ⁶	0.036 ₇ ⁸
-1.6	0.054 ₇ ⁸	0.053 ₇ ⁸	0.052 ₆ ⁷	0.051 ₅ ⁶	0.050 ₅ ⁶	0.049 ₄ ⁵	0.048 ₄ ⁵	0.047 ₄ ⁵	0.046 ₄ ⁵	0.045 ₅ ⁶
-1.5	0.066 ₈ ⁹	0.065 ₅ ⁶	0.064 ₂ ³	0.063 ₀ ¹	0.061 ₇ ⁸	0.060 ₅ ⁶	0.059 ₃ ⁴	0.058 ₂ ³	0.057 ₀ ¹	0.056 ₉ ¹⁰
-1.4	0.080 ₇ ⁸	0.079 ₂ ³	0.077 ₈ ⁹	0.076 ₃ ⁴	0.075 ₄₉ ⁵⁰	0.073 ₅ ⁶	0.072 ₁ ²	0.070 ₈ ⁹	0.069 ₄ ⁵	0.068 ₁ ²
-1.3	0.096 ₈ ⁹	0.095 ₀ ¹	0.093 ₄ ⁵	0.091 ₇ ⁸	0.090 ₁ ²	0.088 ₅ ⁶	0.087 ₆₉ ⁷⁰	0.085 ₃ ⁴	0.083 ₇ ⁸	0.082 ₂ ³
-1.2	0.115 ₀ ¹	0.113 ₁ ²	0.111 ₂ ³	0.109 ₃ ⁴	0.107 ₄ ⁵	0.105 ₆ ⁷	0.103 ₈ ⁹	0.102 ₀ ¹	0.100 ₂ ³	0.098 ₅ ⁶
-1.1	0.135 ₇ ⁸	0.133 ₅ ⁶	0.131 ₃ ⁴	0.129 ₂ ³	0.127 ₁ ²	0.125 ₀ ¹	0.123 ₀ ¹	0.121 ₀ ¹	0.119 ₀ ¹	0.117 ₀ ¹
-1.0	0.158 ₆ ⁷	0.156 ₃ ⁴	0.153 ₈ ⁹	0.151 ₆ ⁵	0.149 ₁ ²	0.146 ₈ ⁹	0.144 ₅ ⁶	0.142 ₃ ⁴	0.140 ₀ ¹	0.137 ₈ ⁹
-0.9	0.184 ₀ ¹	0.181 ₄ ⁵	0.178 ₇ ⁸	0.176 ₁ ²	0.173 ₆ ⁷	0.171 ₀ ¹	0.168 ₅ ⁶	0.166 ₀ ¹	0.163 ₅ ⁶	0.161 ₀ ¹
-0.8	0.211 ₈ ⁹	0.209 ₈₉ ⁹⁰	0.206 ₁ ²	0.203 ₂ ³	0.200 ₄ ⁵	0.197 ₆ ⁷	0.194 ₈ ⁹	0.192 ₁ ²	0.189 ₄ ⁵	0.186 ₇ ⁸
-0.7	0.242 ₁₉ ²⁰	0.238 ₈ ⁹	0.235 ₇ ⁸	0.232 ₆ ⁷	0.229 ₆ ⁷	0.226 ₆ ⁷	0.223 ₆ ⁷	0.220 ₆ ⁷	0.217 ₆ ⁷	0.214 ₇ ⁸
-0.6	0.274 ₂ ³	0.271 ₀₉ ¹⁰	0.267 ₆ ⁷	0.264 ₃ ⁴	0.261 ₀ ¹	0.257 ₈ ⁹	0.254 ₆ ⁷	0.251 ₄ ⁵	0.248 ₂ ³	0.245 ₀ ¹
-0.5	0.308 ₅ ⁶	0.305 ₀ ¹	0.301 ₅ ⁶	0.298 ₀ ¹	0.294 ₅ ⁶	0.291 ₁ ²	0.287 ₇ ⁸	0.284 ₃ ⁴	0.281 ₀₉ ¹⁰	0.277 ₅ ⁶
-0.4	0.344 ₅ ⁶	0.341 ₀₉ ¹⁰	0.337 ₂ ³	0.333 ₅ ⁶	0.330 ₂₉₉ ³⁰⁰	0.326 ₃ ⁴	0.322 ₇ ⁸	0.319 ₁ ²	0.315 ₆ ⁷	0.312 ₀ ¹
-0.3	0.382 ₀ ¹	0.378 ₂ ³	0.374 ₄ ⁵	0.370 ₆ ⁷	0.366 ₆₉ ⁷⁰	0.363 ₁ ²	0.359 ₄ ⁵	0.355 ₆ ⁷	0.352 ₁₉ ²⁰	0.348 ₂ ³
-0.2	0.420 ₇ ⁸	0.416 ₈ ⁹	0.413 ₂₉ ³⁰	0.409 ₀ ¹	0.405 ₁ ²	0.401 ₂ ³	0.397 ₄ ⁵	0.393 ₅ ⁶	0.389 ₇ ⁸	0.386 ₉ ¹⁰
-0.1	0.460 ₁ ²	0.456 ₂ ³	0.452 ₂ ³	0.448 ₂ ³	0.444 ₃ ⁴	0.440 ₃ ⁴	0.436 ₄ ⁵	0.432 ₅ ⁶	0.428 ₅ ⁶	0.424 ₆ ⁷
-0.0	0.500 ₀ ⁰	0.496 ₀ ¹	0.492 ₀ ¹	0.488 ₀ ¹	0.484 ₀ ¹	0.480 ₀ ¹	0.476 ₀ ¹	0.472 ₀ ¹	0.468 ₁ ²	0.464 ₁ ²

Tabela $N_{\text{IIR}}(0, 1)$ Probabilidades Encapsuladas para a Função de Distribuição Acumulada da $N(0, 1)$

z	0	1	2	3	4	5	6	7	8	9
0.0	0.500 ₀ ⁰	0.50 ₃₉ ⁴⁰	0.50 ₇₉ ⁸⁰	0.51 ₁₉ ²⁰	0.51 ₅₉ ⁶⁰	0.5 ₁₉₉ ²⁰⁰	0.52 ₃₉ ⁴⁰	0.52 ₇₉ ⁸⁰	0.531 ₈ ⁹	0.53 ₅₈ ⁵⁹
0.1	0.539 ₈ ⁹	0.543 ₇ ⁸	0.547 ₇ ⁸	0.551 ₇ ⁸	0.555 ₆ ⁷	0.559 ₆ ⁷	0.563 ₅ ⁶	0.567 ₄ ⁵	0.571 ₄ ⁵	0.575 ₃ ⁴
0.2	0.579 ₂ ³	0.583 ₁ ²	0.587 ₀ ¹	0.59 ₀₉ ¹⁰	0.594 ₈ ⁹	0.598 ₇ ⁸	0.602 ₅ ⁶	0.606 ₄ ⁵	0.610 ₂ ³	0.614 ₀ ¹
0.3	0.61 ₇₉ ⁸⁰	0.621 ₇ ⁸	0.625 ₅ ⁶	0.629 ₄ ³	0.633 ₀ ¹	0.636 ₈ ⁹	0.640 ₅ ⁶	0.644 ₃ ⁴	0.648 ₀ ¹	0.651 ₇ ⁸
0.4	0.655 ₄ ⁵	0.659 ₀ ¹	0.662 ₇ ⁸	0.666 ₄ ⁵	0.670 ₀ ¹	0.673 ₆ ⁷	0.677 ₂ ³	0.680 ₉ ⁸	0.684 ₃ ⁴	0.68 ₇₉ ⁸⁰
0.5	0.691 ₄ ⁵	0.69 ₄₉ ⁵⁰	0.69 ₈₄ ⁸⁵	0.70 ₁₉ ²⁰	0.705 ₄ ⁵	0.708 ₈ ⁹	0.712 ₂ ³	0.715 ₆ ⁷	0.719 ₀ ¹	0.722 ₄ ⁵
0.6	0.725 ₇ ⁸	0.729 ₀ ¹	0.732 ₃ ⁴	0.735 ₆ ⁷	0.73 ₈₉ ⁹⁰	0.742 ₁ ²	0.745 ₃ ⁴	0.748 ₅ ⁶	0.751 ₇ ⁸	0.75 ₄₉ ⁵⁰
0.7	0.758 ₀ ¹	0.761 ₁ ²	0.764 ₂ ³	0.767 ₃ ⁴	0.770 ₃ ⁴	0.773 ₃ ⁴	0.776 ₃ ⁴	0.779 ₃ ⁴	0.782 ₃ ⁴	0.785 ₂ ³
0.8	0.788 ₁ ²	0.791 ₀ ¹	0.793 ₈ ⁹	0.796 ₇ ⁸	0.799 ₅ ⁶	0.802 ₃ ⁴	0.805 ₁ ²	0.807 ₈ ⁹	0.810 ₅ ⁶	0.813 ₂ ³
0.9	0.81 ₅₉ ⁶⁰	0.818 ₅ ⁶	0.821 ₂ ³	0.823 ₉ ⁸	0.826 ₃ ⁴	0.82 ₈₉ ⁹⁰	0.831 ₄ ⁵	0.83 ₃₉ ⁴⁰	0.836 ₄ ⁵	0.83 ₈₉ ⁹⁰
1.0	0.841 ₃ ⁴	0.843 ₇ ⁸	0.846 ₁ ²	0.848 ₄ ⁵	0.850 ₈ ⁹	0.853 ₁ ²	0.855 ₄ ⁵	0.857 ₆ ⁷	0.8 ₅₉₉ ⁶⁰⁰	0.862 ₁ ²
1.1	0.864 ₃ ⁴	0.866 ₅ ⁶	0.868 ₇ ⁸	0.870 ₈ ⁷	0.872 ₈ ⁹	0.87 ₄₉ ⁵⁰	0.87 ₆₉ ⁷⁰	0.87 ₈₉ ⁹⁰	0.88 ₀₉ ¹⁰	0.88 ₂₉ ³⁰
1.2	0.88 ₄₉ ⁵⁰	0.886 ₈ ⁹	0.888 ₇ ⁸	0.890 ₆ ⁷	0.892 ₅ ⁶	0.894 ₃ ⁴	0.896 ₁ ²	0.89 ₇₉ ⁸⁰	0.899 ₇ ⁸	0.901 ₄ ⁵
1.3	0.903 ₁ ²	0.90 ₄₉ ⁵⁰	0.906 ₅ ⁶	0.908 ₂ ³	0.909 ₈ ⁹	0.911 ₄ ⁵	0.913 ₀ ¹	0.914 ₆ ⁷	0.916 ₂ ³	0.917 ₇ ⁸
1.4	0.919 ₂ ³	0.920 ₇ ⁸	0.922 ₁ ²	0.923 ₆ ⁷	0.925 ₀ ¹	0.926 ₄ ⁵	0.927 ₈ ⁹	0.929 ₂ ³	0.930 ₅ ⁶	0.931 ₈ ⁹
1.5	0.933 ₁ ²	0.934 ₄ ⁵	0.935 ₇ ⁸	0.93 ₆₉ ⁷⁰	0.938 ₂ ³	0.939 ₄ ⁵	0.940 ₆ ⁷	0.941 ₇ ⁸	0.94 ₂₉ ³⁰	0.944 ₀ ¹
1.6	0.945 ₂ ³	0.946 ₃ ⁴	0.947 ₃ ⁴	0.948 ₅ ⁶	0.949 ₅ ⁶	0.950 ₅ ⁶	0.951 ₅ ⁶	0.952 ₅ ⁶	0.953 ₅ ⁶	0.954 ₅ ⁶
1.7	0.955 ₄ ⁵	0.956 ₃ ⁴	0.957 ₂ ³	0.958 ₁ ²	0.959 ₀ ¹	0.9 ₅₉₉ ⁶⁰⁰	0.960 ₇ ⁸	0.961 ₆ ⁷	0.962 ₄ ⁵	0.963 ₂ ³
1.8	0.964 ₀ ¹	0.964 ₈ ⁹	0.965 ₆ ⁷	0.966 ₃ ⁴	0.967 ₁ ²	0.967 ₈ ⁹	0.968 ₅ ⁶	0.969 ₂ ³	0.9 ₆₉₉ ⁷⁰⁰	0.970 ₆ ⁷
1.9	0.971 ₂ ³	0.97 ₁₉ ²⁰	0.972 ₅ ⁶	0.973 ₁ ²	[0.973 ₈ ⁹	0.974 ₄ ⁵	0.975 ₀ ¹	0.975 ₅ ⁶	0.976 ₁ ²	0.976 ₇ ⁸
2.0	0.977 ₂ ³	0.977 ₇ ⁸	0.978 ₃ ⁴	0.978 ₈ ⁹	0.979 ₃ ⁴	0.979 ₈ ⁹	0.980 ₃ ⁴	0.980 ₇ ⁸	0.981 ₂ ³	0.981 ₆ ⁷
2.1	0.982 ₁ ²	0.982 ₅ ⁶	0.98 ₂₉ ³⁰	0.983 ₄ ⁵	0.983 ₈ ⁹	0.984 ₂ ³	0.984 ₆ ⁷	0.98 ₄₉ ⁵⁰	0.985 ₃ ⁴	0.985 ₇ ⁸
2.2	0.986 ₀ ¹	0.986 ₄ ⁵	0.986 ₇ ⁸	0.987 ₁ ²	0.987 ₄ ⁵	0.987 ₇ ⁸	0.988 ₀ ¹	0.988 ₃ ⁴	0.988 ₆ ⁷	0.98 ₈₉ ⁹⁰
2.3	0.989 ₂ ³	0.989 ₅ ⁶	0.989 ₈ ⁹	0.990 ₀ ¹	0.990 ₃ ⁴	0.990 ₆ ⁷	0.990 ₈ ⁹	0.991 ₁ ²	0.991 ₃ ⁴	0.991 ₅ ⁶
2.4	0.991 ₈ ⁹	0.992 ₀ ¹	0.992 ₂ ³	0.992 ₄ ⁵	0.992 ₆ ⁷	0.992 ₈ ⁹	0.993 ₀ ¹	0.993 ₂ ³	0.993 ₄ ⁵	0.993 ₆ ⁷
2.5	0.993 ₇ ⁸	0.99 ₃₉ ⁴⁰	0.994 ₁ ²	0.994 ₂ ³	0.994 ₄ ⁵	0.994 ₆ ⁷	0.994 ₇ ⁸	0.99 ₄₉ ⁵⁰	0.995 ₀ ¹	0.995 ₂ ³
2.6	0.995 ₃ ⁴	0.995 ₄ ⁵	0.995 ₆ ⁷	0.995 ₇ ⁸	0.995 ₈ ⁹	0.99 ₅₉ ⁶⁰	0.996 ₀ ¹	0.996 ₂ ³	0.996 ₃ ⁴	0.996 ₄ ⁵
2.7	0.996 ₅ ⁶	0.996 ₆ ⁷	0.996 ₇ ⁸	0.996 ₈ ⁹	0.99 ₆₉ ⁷⁰	0.997 ₀ ¹	0.997 ₁ ²	0.997 ₂ ³	0.997 ₃ ⁴	0.997 ₄ ⁵
2.8	0.997 ₄ ⁵	0.997 ₅ ⁶	0.997 ₅ ⁶	0.997 ₆ ⁷	0.997 ₇ ⁸	0.997 ₈ ⁹	0.997 ₈ ⁹	0.99 ₇₉ ⁸⁰	0.998 ₀ ¹	0.998 ₀ ¹
2.9	0.998 ₁ ²	0.998 ₁ ²	0.998 ₂ ³	0.998 ₃ ⁴	0.998 ₃ ⁴	0.998 ₄ ⁵	0.998 ₄ ⁵	0.998 ₅ ⁶	0.998 ₅ ⁶	0.998 ₆ ⁷
3.0	0.998 ₆ ⁷	0.998 ₆ ⁷	0.998 ₇ ⁸	0.998 ₇ ⁸	0.998 ₈ ⁹	0.998 ₈ ⁹	0.998 ₈ ⁹	0.99 ₈₉ ⁹⁰	0.99 ₈₉ ⁹⁰	0.99 ₈₉ ⁹⁰

Os exemplos a seguir são aplicações do método. Os resultados para a Normal foram calculados no Matlab [MatLab04], usando a biblioteca IntLab [Rump06] e a Função 4.4, nos formatos `short` (precisão simples) e `long` (precisão dupla). No caso das probabilidades intervalares, os cálculos são realizados sem e com o uso da Tabela $N_{IIR}(0, 1)$, proposta nesta tese.

Exemplo 33. *As probabilidades abaixo são calculadas usando a Tabela da distribuição $N(0, 1)$ em [Meyer83] e a Tabela $N_{IIR}(0, 1)$ deste trabalho.*

(i) $P(X \leq 0) = 0.5000$;

$$P_{IIR}(X \leq 0) = [0.5000, 0.5000].$$

Tempo de Processamento: 0.0940s.

(ii) $P(X \leq 1.23) = 0.8907$;

$$P_{IIR}(X \leq 1.23) = [0.8906, 0.8907].$$

Tempo de Processamento: 17.2030s.

(iii) $P(X \leq -0.12) = 0.4522$;

$$P_{IIR}(X \leq -0.12) = [0.4522, 0.4523].$$

Tempo de Processamentos: 17.1880s.

(iv) $P(0.16 \leq X \leq 2.43) = 0.9925 - 0.5636 = 0.4289$;

$$P_{IIR}(0.16 \leq X \leq 2.43) = [0.9924, 0.9925] - [0.5635, 0.5636] = [0.4288, 0.4290].$$

Tempo de Processamentos: 17.2500s.

(v) $P(-2.15 \leq X \leq 1) = 0.8413 - 0.0158 = 0.8255$;

$$P_{IIR}(-2.15 \leq X \leq 1) = [0.8413, 0.8414] - [0.0157, 0.0158] = [0.8255, 0.8257].$$

Tempo de Processamentos: 17.2970s.

(vi) $P(X \geq 0) = 1 - P(X < 0) = 0.5000$;

$$P_{IIR}(X \geq 0) = [1, 1] - [0.5000, 0.5000] = [0.5000, 0.5000].$$

Tempo de Processamentos: 0.1100s.

(vii) $P(X \geq 1) = 1 - P(X < 1) = 1 - 0.8413 = 0.1587$;

$$P_{IIR}(X \geq 1) = [1, 1] - P_{IIR}(X < 1) = [1, 1] - [0.8413, 0.8414] = [0.1586, 0.1587].$$

Tempo de Processamentos: 17.2500s.

(viii) $P(X \geq -1.3) = 1 - P(X < -1.3) = 1 - 0.0968 = 0.9032$;

$$P_{IIR}(X \geq -1.3) = [1, 1] - P_{IIR}(X < -1.3) = [1, 1] - [0.0968, 0.0969] = [0.9031, 0.9032].$$

Tempo de Processamentos: 17.3590s

Exemplo 34. A duração de um certo tipo de pneu, em quilômetros rodados, é uma variável normal com duração média 60000km e desvio-padrão de 10 000km.

- (a) Qual é a probabilidade de um pneu durar entre 63 500km e 70 000km?
- (b) Qual é a probabilidade de um pneu durar entre 50 000km e 70 000km?
- (c) Qual é a probabilidade de um pneu durar exatamente 65 555km?
- (d) Qual é a probabilidade de um pneu escolhido ao acaso durar mais de 75000km?

Solução.

- (a) Seja $B = \{63500 < X < 70000\}$. Logo, $P(63500 < X < 70000) = P(B)$.

$$\begin{aligned} P(63500 < X < 70000) &= P(0.35 < Z < 1) \\ &= F_Z(1) - F_Z(0.35) - P(Z = 1) \\ &= F_Z(1) - F_Z(0.35), \end{aligned} \quad (4.5)$$

Em (4.5), F_Z é a função de distribuição acumulada de uma $N(0, 1)$. Portanto,

- (i) Probabilidade real

Precisão simples: $P(B) = 0.2045 = p_1$,

Precisão dupla: $P(B) = 0.20451409489292 = p_2$.

- (ii) **Probabilidade encapsulada = normal1(0.35, 1.00, 50)**

Precisão simples: $P_{\text{IIR}}(B) = [0.2045, 0.2046] = P_1$,

Precisão dupla: $P_{\text{IIR}}(B) = [0.20451409489283, 0.20451409489302] = P_2$.

Tempo de processamento: 16.485s.

Os cálculos seguintes são realizados, em precisão simples, utilizando as **probabilidades intervalares da Tabela $N_{\text{IIR}}(0, 1)$** e a subtração intervalar.

$$P_{\text{IIR}}(Z \leq 1) = [0.8413, 0.8414],$$

$$P_{\text{IIR}}(Z \leq 0.35) = [0.6368, 0.6369],$$

logo,

$$P_{\text{IIR}}(0.35 \leq Z \leq 1) = [0.8413, 0.8414] - [0.6368, 0.6369] = [0.2044, 0.2046] = P_3.$$

Tem-se que:

$$w(P_3) = 0.0002 > w(P_1) = 0.0001,$$

$$P_1 \subset P_3,$$

$$p_1 \in P_1, p_2 \in P_2, p_1 \in P_3.$$

b) Seja $C = \{50000 < X < 70000\}$. Logo, $P(50000 < X < 70000) = P(C)$.

(i) Probabilidade real

Precisão simples: $P(C) = 0.6827 = p_1$,

Precisão dupla: $P(C) = 0.68268949213709 = p_2$.

(ii) **Probabilidade encapsulada = normal1(-1.00, 1.00, 50)**

Precisão simples: $P_{\text{IIR}}(C) = [0.6826, 0.6827] = P_1$,

Precisão dupla: $P_{\text{IIR}}(C) = [0.68268949207779, 0.68268949219732] = P_2$.

Tempo de Processamento: 16.969s

(iii) Cálculo da **probabilidade encapsulada com valores da Tabela $N_{\text{IIR}}(0, 1)$** .

$$P_{\text{IIR}}(Z < 1) = [0.8413, 0.8414],$$

$$P_{\text{IIR}}(Z < -1) = [0.1586, 0.1587],$$

logo,

$$\begin{aligned} P_{\text{IIR}}(-1 < Z < 1) &= [0.8413, 0.8414] - [0.1586, 0.1587] \\ &= [0.6826, 0.6828] = P_3, \end{aligned}$$

com $w(P_3) = 0.0002$. Assim, $w(P_3) > w(P_1)$.

c) Seja $D = \{X = 65555\}$. Portanto, $P(X = 65555) = P(D)$.

(i) Probabilidade real

Precisão simples: $P(D) = 0.0000 = p_1$,

Precisão dupla: $P(D) = 0.0000000000000000 = p_2$.

(ii) **Probabilidade encapsulada = normal1(0.5555, 0.5555, 50)**.

Precisão simples: $P_{\text{IIR}}(D) = [0.0000, 0.0000] = P_1$,

Precisão dupla: $P_{\text{IIR}}(D) = [0.0000000000000000, 0.0000000000000000] = P_2$.

Tempo de Processamento: 16.2350s.

(iii) Cálculo da **probabilidade encapsulada com valores da Tabela $N_{\text{IIR}}(0, 1)$** .

$$\begin{aligned} P_{\text{IIR}}(X = 65555) &= P_{\text{IIR}}(Z = 0.5555) \\ &= F_{\text{IIR}}(0.56) - F_{\text{IIR}}(0.56) \\ &= [0.7088, 0.7089] - [0.7088, 0.7089] \\ &= [-0.0001, 0.0001] = P_3 \end{aligned}$$

Neste caso, diferentemente da probabilidade real, o intervalo encapsulador, P_3 não é zero e sim contém o zero, o que decorre da definição da subtração intervalar.

d) Seja $E = \{X > 75000\}$. Logo, $P(X > 75000) = P(E)$.

(i) Probabilidade real

Precisão simples: $P(E) = 0.0668 = p_1$,

Precisão dupla: $P(E) = 0.06680720126886 = p_2$.

(ii) **Probabilidade encapsulada = 0.5-normal1(0, 1.5, 50).**

Precisão simples: $P_{\text{IRR}}(E) = [0.0668, 0.0669] = P_1$,

Precisão dupla: $P_{\text{IRR}}(E) = [0.06680720126054, 0.06680720127720] = P_2$.

Tempo de Processamento: 17.3910s.

(iii) Cálculo da **probabilidade encapsulada com valores da Tabela $N_{\text{IRR}}(0, 1)$.**

$$\begin{aligned} P_{\text{IRR}}(X > 75000) &= P_{\text{IRR}}(Z > 1.5) \\ &= [1, 1] - P_{\text{IRR}}(Z \leq 1.5) \\ &= [1, 1] - [0.9331, 0.9332] \\ &= [0.0668, 0.0669] = P_3 \end{aligned}$$

Portanto, $p_1 \in P_2, p_1 \in P_3, p_2 \in P_2, P_1 = P_3$

Exemplo 35. Suponha que X tenha a distribuição $N(2, 0.16)$. Calcule as seguintes probabilidades:

(a) $P(X \geq 2.3)$;

(b) $P(1.8 \leq X \leq 2.1)$.

Solução.

a) $P(X \geq 2.3) = P(Z \geq 0.75)$

(i) Probabilidade real

Precisão simples: $0.2266 = p_1$,

Precisão dupla: $0.22662735237687 = p_2$.

(ii) **Probabilidade encapsulada = 0.5 - normal1(0, 0.75, 50)**

Precisão simples: $[0.2266, 0.2267] = P_1$,

Precisão dupla: $[0.22662735237686, 0.22662735237688] = P_2$.

Tempo de Processamento: 16.8430s.

(iii) Cálculo da **probabilidade encapsulada com valores da Tabela $N_{\text{IIR}}(0, 1)$** .

$$\begin{aligned} P_{\text{IIR}}(X \geq 2.3) &= P_{\text{IIR}}(Z \geq 0.75) \\ &= [1, 1] - P_{\text{IIR}}(Z < 0.75) \\ &= [1, 1] - [0.7733, 0.7734] \\ &= [0.2266, 0.2267] = P_3. \end{aligned}$$

Tem-se que: $p_1 \in P_1, p_1 \in P_3, p_2 \in P_2$.

(b) $P(1.8 \leq X \leq 2.1) = P(-0.5 \leq Z \leq 0.25)$.

(i) Probabilidade real

Precisão simples: $0.2902 = p_1$,

Precisão dupla: $0.29016878695694 = p_2$.

(ii) **Probabilidade encapsulada = normal1(-0.5, 0.25, 50)**

Precisão simples: $[0.2901, 0.2902] = P_1$,

Precisão dupla: $[0.29016878695693, 0.29016878695695] = P_2$.

Tempo de Processamento: 16.8430s.

(iii) Cálculo da **probabilidade encapsulada com valores da Tabela $N_{\text{IIR}}(0, 1)$** .

$$\begin{aligned} P_{\text{IIR}}(1.8 \leq X \leq 2.1) &= P_{\text{IIR}}(-0.5 \leq Z \leq 0.25) \\ &= [0.5987, 0.5988] - [0.3085, 0.3086] \\ &= [0.2901, 0.2903] = P_3. \end{aligned}$$

Tem-se que: $p_1 \in P_1, p_1 \in P_3, p_2 \in P_2$.

4.1.3 Distribuição Uniforme

A distribuição Uniforme possui densidade com primitiva na forma analítica, entretanto valores da probabilidade podem não ser representáveis em computadores.

A densidade de uma Uniforme X no intervalo $A = [a, b]$ é

$$f(x) = \begin{cases} \frac{1}{b-a}, & x \in [a, b], \\ 0, & x \notin [a, b]. \end{cases}$$

Seja $B = [c, d] \subset \mathbb{R}$. Para calcular $P(c < X \leq d)$ tem-se as seguintes situações possíveis:

(i) Se $d \leq a$,

$$P(c < X \leq d) = 0.$$

(ii) Se $c < a \leq d \leq b$,

$$P(c < X \leq d) = P(a \leq X \leq d) = \int_a^d \frac{1}{b-a} dx = \frac{d-a}{b-a}.$$

(iii) Se $a < c \leq b \leq d$,

$$P(c < X \leq d) = P(c \leq X \leq b) = \int_c^b \frac{1}{b-a} dx = \frac{b-c}{b-a}.$$

(iv) Se $a \leq c < d \leq b$,

$$P(c < X \leq d) = P(c \leq X \leq d) = \int_c^d \frac{1}{b-a} dx = \frac{d-c}{b-a}.$$

(v) Se $c \leq a < b \leq d$,

$$P(c < X \leq d) = P(a \leq X \leq b) = \int_a^b \frac{1}{b-a} dx = \frac{b-a}{b-a} = 1.$$

(vi) Se $c \geq b$,

$$P(c < X \leq d) = 0.$$

Analisando as probabilidades acima, observa-se que os cálculos anteriores poderiam ter sido realizados através da seguinte definição:

Definição 4.1.1.

$$P(c < X \leq d) = \begin{cases} \frac{w([c, d] \cap [a, b])}{b-a}, & [c, d] \cap [a, b] \neq \emptyset, \\ 0, & [c, d] \cap [a, b] = \emptyset. \end{cases}$$

onde w é o diâmetro do intervalo.

A densidade da Uniforme tem primitiva na forma analítica, portanto, pelo Teorema Fundamental do Cálculo Integral, pode-se calcular qualquer integral definida, por exemplo

$$P(c < X \leq d) = \int_c^d f(x) dx,$$

diretamente desta primitiva.

Exemplo 36. Um ponto é escolhido ao acaso no segmento de reta $[0, 3]$. Qual é a probabilidade de que o ponto escolhido esteja entre 1 e 2?

Solução. Fazendo X representar a coordenada do ponto escolhido, tem-se que

$$f(x) = \begin{cases} \frac{1}{3}, & x \in [0, 3], \\ 0, & x \notin [0, 3]. \end{cases}$$

Portanto,

$$P(1 \leq X \leq 2) = \int_1^2 \frac{1}{3} dx = \frac{1}{3}.$$

Entretanto, a existência da primitiva na forma analítica não impediu a ocorrência de problemas numéricos relacionados com o cálculo das integrais, como visto no exemplo acima, os quais justificam a busca para um intervalo encapsulador para a probabilidade real. Lembrando que

$$\frac{1}{3} = 0.3333 \dots_{10} = 0.0101 \dots_2$$

o que não tem representação interna exata.

A definição a seguir, a função UNIF, é uma proposta de como calcular um intervalo encapsulador para probabilidades da Uniforme no intervalo $[a, b]$. Enfatiza-se que UNIF é uma extensão intervalar inclusão monotônica.

Definição 4.1.2. *Encapsulando Probabilidades para a Uniforme.*

$$UNIF(c, d) = \left[\frac{w([c, d] \cap [a, b])}{b - a}, \frac{\overline{w}([c, d] \cap [a, b])}{b - a} \right], [c, d] \cap [a, b] \neq \emptyset.$$

Se $[c, d] \cap [a, b] = \emptyset$, então $UNIF(c, d) = 0_\epsilon$.

UNIF, definida com suporte na Matemática Intervalar [Moore66, Moore79, MKC09] e na Aritmética de Exatidão Máxima [Campos97, Kulisch08, KulMir81, KulMir86], dá o intervalo de menor amplitude para encapsular probabilidades para a distribuição Uniforme.

O intervalo *UNIF* foi implementado no Matlab [MatLab04], como arquivo M de funções usando a biblioteca IntLab [Rump01, Rump06]. Sua chamada é

$$dunif(a, b, c, d) \tag{4.6}$$

cujos parâmetros são:

a = limite inferior do intervalo A ,

b = limite superior do intervalo A ,

c = limite inferior do intervalo $[c, d]$,

d = limite superior do intervalo $[c, d]$.

O algoritmo a seguir sumariza o procedimento para obtenção de probabilidades encapsuladas para a Uniforme.

Algoritmo Uniforme

(U1) Instanciação de: a, b, c, d ;

(U2) Realização do procedimento: $dunif(a, b, c, d)$.

O exemplo a seguir resolve o problema anterior sobre a Uniforme, encapsulando probabilidades nos formatos `short` e `long`.

Exemplo 37. (i) Probabilidade real: $P(1 \leq X \leq 2)$,

Precisão simples: 0.3333,

Precisão dupla: 0.33333333333333.

(ii) Probabilidade encapsulada = $dunif(0, 3, 1, 2)$

Precisão simples: [0.3333, 0.3334],

Precisão dupla: [0.33333333333333, 0.33333333333334].

Tempo de Processamento: 0.0460s.

Ainda neste exemplo, qual é a probabilidade de que o ponto escolhido esteja entre 1/2 e 4/7?

(i) Probabilidade real: $P(1/2 \leq X \leq 4/7)$,

Precisão simples: 0.0238,

Precisão dupla: 0.02380952380952.

(ii) Probabilidade encapsulada: $dunif(0, 3, 1/2, 4/7)$

Precisão simples: [0.0238, 0.0239],

Precisão dupla: [0.0238095238095, 0.02380952380953].

Tempo de Processamento: 0.0630s.

Conclusões, Contribuições e Trabalhos Futuros

O objetivo principal desta tese foi propor um método para calcular intervalos encapsuladores ou probabilidades autovalidáveis ou probabilidades encapsuladas ou ainda probabilidades intervalares para as variáveis Exponencial, Normal Reduzida e Uniforme. No caso da Exponencial e Normal Reduzida, o método proposto usou Simpson Intervalar de Caprani et al [CMN02]. A Uniforme, devido ao fato de ter derivada de ordem quatro nula, teve uma forma diferente de encapsular probabilidades. Portanto, nesta tese, resultados da Matemática Intervalar e da Aritmética de Exatidão Máxima foram aplicadas em um problema particular, como um estudo de caso, que foi o cálculo de probabilidades intervalares para a Exponencial, Normal Reduzida e Uniforme. Uma tabela de probabilidades encapsuladas para $N_{\text{IIR}}(0, 1)$ foi computada. A metodologia aqui proposta foi implementada no IntLab [Rump06] e resultados numéricos ilustraram os teóricos. Adicionalmente intervalos autovalidáveis foram computados para probabilidade condicional.

As principais contribuições desta tese são:

- (i) Campos [Campos97, Campos00] definiu uma probabilidade intervalar e aplicou os resultados obtidos no cálculo de probabilidades para variáveis aleatórias discretas. Entretanto, em seus trabalhos, não foi feita qualquer alusão ao cálculo de probabilidades condicionais ou independência com o enfoque intervalar. Nesta tese é proposta uma definição de probabilidade condicional intervalar e uma propriedade da probabilidade intervalar de eventos independentes. Exemplos numéricos ilustram os resultados teóricos propostos.
- (ii) Implementação, no Maple 8.0 [Maple02], das representações geométricas das operações aritméticas, intersecção, união, união convexa e dos elementos topológicos no espaço dos intervalos. As figuras 1 a 10 são comuns em textos sobre Matemática Intervalar. Não é o caso para as de números 11 a 19, que correspondem à representação de resultados topológicos básicos, com intervalos. Estes resultados encontram-se no Apêndice A.
- (iii) Construção de extensões intervalares inclusões monotônicas para as funções densidades Exponencial, Normal Reduzida e Uniforme; criação e implementação de bibliotecas com procedimentos para calcular probabilidades intervalares para as distribuições Exponencial, Normal Reduzida e Uniforme. Os cálculos intervalares foram realizados no MatLab [MatLab04], u-

sando o IntLab 5.3 [Rump06]. Todos os procedimentos para obter estes resultados encontram-se no Apêndice B.

- (iv) A Tabela $N_{\text{IIR}}(0, 1)$ é similar a encontrada em Meyer [Meyer83] só que a mesma apresenta probabilidades intervalares ou intervalos encapsuladores para uma $N(0, 1)$. Se os cálculos forem realizados com intervalos e for necessário um valor real, basta, com os resultados da Tabela $N_{\text{IIR}}(0, 1)$, tomar o ponto médio do intervalo.

Como trabalhos futuros citam-se:

- (i) Propor, se possível, versões intervalares para o chamado Teoremas da Multiplicação de Probabilidades e da Probabilidade Total.
- (ii) Estudar o conceito de independência no caso de mais do que dois eventos. No caso real, a operação aritmética multiplicação é usada. Esta operação no espaço dos intervalos aumenta o intervalo resposta.
- (iii) Calcular probabilidades intervalares, ou intervalos encapsuladores, para outras variáveis aleatórias contínuas.
- (iv) Além da Normal, outras variáveis aleatórias, de grande aplicabilidade na Estatística, têm tabelas. Construir tabelas com probabilidades encapsuladas para estas.
- (v) O Método de Simpson intervalar não considera o caso onde o integrando é uma extensão intervalar. Analisar quais as vantagens e desvantagens de se considerar este caso.
- (vi) Seja $(\Omega, \mathcal{A})$ um espaço mensurável e $(\mathbb{R}, \mathcal{B})$ a reta com $\mathcal{B}(\mathbb{R})$ sendo a σ -álgebra de Borel. $X(w)$ é uma função $\mathcal{A}$ -mensurável ou uma variável aleatória se $\{w \mid X(w) \in B\} \in \mathcal{A}, \forall B \in \mathcal{B}(\mathbb{R})$. Ou, equivalentemente, se $X^{-1}(B) \equiv \{w \in \Omega \mid X(w) \in B\}$ é um conjunto mensurável em Ω . Fazendo uma analogia com este resultado, aparentemente poderia se ter o seguinte: Seja $(\Omega, \mathcal{A})$ um espaço mensurável e $(\text{IIR}, \mathcal{B}(\text{IIR}))$ o espaço dos intervalos com $\mathcal{B}(\text{IIR})$ sendo a σ -álgebra de Borel gerada pelos abertos de IIR. Como $\langle \text{IIR}, d \rangle$ é um espaço métrico, os abertos podem ser gerados pela métrica. $X_{\text{IIR}}(w)$ é uma função $\mathcal{A}$ -mensurável ou uma variável aleatória intervalar se $\{w \mid X_{\text{IIR}}(w) \in IB\} \in \mathcal{A}, \forall IB \in \mathcal{B}(\text{IIR})$. Ou, $X_{\text{IIR}}^{-1}(IB) = \{w \in \Omega \mid X_{\text{IIR}}(w) \in IB\}$ é um conjunto mensurável em Ω . Analisar este fato e mapear resultados da Teoria da Medida e Integração considerando IIR.

Referências Bibliográficas

- [Acioly91] B. M. Acioly (1991), *Fundamentação Computacional da Matemática Intervalar*. PhD thesis, Porto Alegre, Departamento de Informática Teórica/UFRGS.
- [AleHer83] G. Alefeld, G. and J. Herzberger (1983), *Introduction to Interval Computations*. Academic Press, New York.
- [Billingsley79] P. Billingsley (1979), *Probability and Measure*. John Wiley Sons, Inc., New York.
- [BurFai03] R. L. Burden, J. D. Faires (2003), *Análise Numérica*. Pioneira Thompson Learning, São Paulo.
- [Burrill72] C. W. Burrill (1972), *Measure, Integration and Probability*. McGraw-Hill, New York.
- [Campos97] M. A. Campos (1997), *Uma Extensão Intervalar para a Probabilidade Real*. Tese, Pós-Graduação em Ciência da Computação, Centro de Informática/UFPE, Recife, Pe.
- [Campos00] M. A. Campos (2000), *The Interval Probability: Applications to Discrete Random Variables*. TEMA, Vol. 1, No. 2, pp. 333-344.
- [CDCAD02] M. A. Campos, G. P. Dimuro, A. C. R. Costa, J. F. F. Araújo, A. M. Dias (2002), *Probabilidade Intervalar e Cadeia de Markov Intervalares no Maple*. TEMA, Vol. 3, No.2, pp. 53-62.
- [CMN02] O. Caprani, K. Madsen, H. B. Nielsen (2002), *Introduction to Interval Analysis*. IMM - Informatics and Mathematical Modelling, pp. 82. Technical University of Denmark, DTU.
- [CMR81] O. Caprani, K. Madsen, L. B. Rall (1981), *Integration of Interval Functions*. SIAM, J. Math. Anal. Vol. 12, No. 3, pp. 321-341.
- [Corliss05] G. F. Corliss (2005), *Tutorial on Validated Scientific Computing Using Interval Analysis: Interval Arithmetic, Algorithms and Tools*. Second Scandinavian Workshop on Interval Methods and their Applications. Technical University of Denmark.
- [Feller68] W. Feller (1968), *An Introduction to Probability and Its Applications*. John Wiley & Sons, New York.

- [Forsythe77] G. E. Forsythe (1977), *Pitfalls in Computation, or a Why a Math Book Isn't Enough*. Amer. Math. Monthly, pp. 931-955.
- [Goldberg91] D. Goldberg (1991), *What Every Computer Scientist Should Know About Floating-Point Arithmetic*. ACM Computing Surveys, 23(1), pp. 5-48.
- [IEEE754] ANSI/IEEE STD 754 (1987), *IEEE Standard for Binary Floating - Point Arithmetic*. ACM SIGPLAN, 22, pp. 5-48.
- [James06] B. R. James (2006), *Probabilidade: um curso em nível intermediário*. 3.ed. Rio de Janeiro: IMPA, 299 p. (Projeto Euclides)
- [Kearfott95] R. B. Kearfott and V. Kreinovich (1995), *Applications of Interval Computations: An Introduction*. Disponível em < [http : //www.cs.utep.edu/vladik/1995/tr95 – 32.pdf](http://www.cs.utep.edu/vladik/1995/tr95-32.pdf) >. Acessado em 2007
- [Kolmogorov50] A. N. Kolmogorov (1950), *Foundations of the Theory of Probability*. Chelsea, New York.
- [Kulisch08] U. W. Kulisch (2008), *Complete Interval Arithmetic and its Implementation on the Computer*. Disponível em < [http : //www.math.kit.edu/iwrmm/seite/preprints/media/preprint%20nr.%2008.03.pdf](http://www.math.kit.edu/iwrmm/seite/preprints/media/preprint%20nr.%2008.03.pdf) >. Acessado em 2009.
- [KulMir81] U. W. Kulisch and W. L. Miranker (1981), *Computer Arithmetic in Theory and Practice*. Academic Press. New York.
- [KulMir86] U. W. Kulisch and W. L. Miranker (1986), *The Arithmetic of the Digital Computer: A New Approach*. SIAM Review, 28(1) pp. 1-40, March.
- [Maple02] *MAPLE 8.0*(2002). Waterloo Maple Inc.
- [MatLab04] *MATLAB 7.0* (2004). The MathWorks Inc.
- [Meyer83] P. L. Meyer (1983), *Probabilidade Aplicações à Estatística*. LTC Editora, Rio de Janeiro.
- [MonGed01] M. B. Monagan, K. O. Geddes (2001), *Maple 7 Programming Guide*. Waterloo Maple Inc. Canadá.
- [MKC09] R. E. Moore, R. B. Kearfott, M. J. Cloud (2009), *Introduction to Interval Analysis*. SIAM. Philadelphia.

- [Moore66] R. E. Moore (1966), *Interval Analysis*. Prentice Hall, Englewood Cliffs, NJ.
- [Moore79] R. E. Moore (1979), *Methods and Applications of Interval Analysis*. SIAM, Stud. Appl. Math. 2, Philadelphia.
- [MSY60] R. E. Moore, W. Strother, and C. T. Yang (1960), *Interval Integrals*. Disponível em < [http : //interval.louisiana.edu/Moores_early_papers/Moore_integrals.pdf](http://interval.louisiana.edu/Moores_early_papers/Moore_integrals.pdf) >. Acessado em 2008.
- [Rall82] L. B. Rall (1982), *Integration of Interval Functions II. The Finite Case*. SIAM, J. Math. Anal. Vol. 13, No. 4, pp. 690-697.
- [Rudin71] W. Rudin (1971), *Princípios de Análise Matemática*. Livro Técnico S.A., Rio de Janeiro.
- [RugLop96] M. A. G. Ruggiero, V. L. R. Lopes (1996), *Cálculo Numérico Aspectos Teóricos e Computacionais*. MAKRON books, São Paulo.
- [Rump01] S. M. Rump (2001), *Rigorous and Portable Standard Functions* BIT 41(3), pp. 540-562.
- [Rump06] S. M. Rump (2006), *INTLAB 5.3 - INTerval LABoratory*. Disponível em < www.ti3.harburg.de/rump/intlab >. Acessado em 2007.
- [SDBCRC06] A. V. Santos, G. P. Dimuro, L. V. Barboza, A. C. R. Costa, R. H. S. Reiser, M. A. Campos (2006), *Probabilidades Intervalares em Modelos Ocultos de Markov*. TEMA, Vol. 7, pp. 361-370.
- [Sterbenz74] Pat H. Sterbenz (1974), *Floating-Point Computation*. Prentice-Hall, Inc.
- [Sunaga58] T. Sunaga (1958), *Theory of An Interval Algebra and its Application to Numerical Analysis*. RAAG Memoirs, (2), pp. 29-46.
- [Tucker07] W. Tucker (2007), *Auto-validating Numerical Methods*. Disponível em < www.math.uu.se/~warwick/vt07/FMB/avnm1.pdf >. Acessado em 2007.
- [WanKen94] M. C. Wang, W. J. Kennedy (1994), *Self-Validating Computations of Probabilities for Selected Central and Noncentral Univariate Probability Functions*. JASA, V. 89, N. 427, pp. 878-887.

Implementação Intervalares no Maple

Este apêndice apresenta implementações no Maple 8.0 [Maple02] das representações geométrica das operações aritméticas, intersecção, união, união convexa e dos elementos topológicos.

Os programas abaixo foram implementados no Maple 8.0 [Maple02] e geram gráficos representativos das operações de Adição, Subtração, Multiplicação, Divisão, Negativo, Recíproco, Intersecção, União, União Convexa, Bola Aberta, Bola Fechada e Esfera em IIR.

A.1 Representação geométrica das operações aritméticas entre intervalos

Os programas `geosoma`, `geosub`, `geomul`, `geodiv` que correspondem respectivamente as operações entre os intervalos $A = [a, b]$ e $B = [c, d]$ de adição, subtração, multiplicação, divisão necessitam de quatro parâmetros de entrada, os extremos a , b , c , d , nesta ordem, de A e B . Por exemplo, a representação gráfica da adição entre A e B é gerada por `geosoma(a, b, c, d)`.

Os programas `geoposto` e `geoinv` que correspondem ao negativo e ao recíproco, respectivamente do intervalo A precisam de dois parâmetros de entrada, os extremos a , b , nesta ordem, de A . Por exemplo, a representação do negativo de A é gerada por `geoposto(a, b)`.

`geosoma.msw`

```
> #O procedimento geosoma determina um esboço gráfico da Adição
#entre os intervalos A=[a, b] e B=[c, d].
> #Representação Geométrica da Adição de Intervalos.
>
>with(plots):
>
> geosoma:=proc(a,b,c,d)
>   plotsetup(ps,plotoutput='intsoma.ps',plotoptions='colour=cmymk,width=7.5cm,
height=7.5cm');
>   h:=max(abs(a), abs(b), abs(c),abs(d),abs(a+c),abs(b+d)):
>   w:=plot(y=x,x=-(h+2)..h+2);   t4:=textplot([0.25,h+2,"y"]):
>   if (a<=b and c<=d)
```

```

> then
>   poly1:=[ [a,b], [a,a], [b,b] ];
>   poly3:=[ [c,d], [c,c], [d,d] ];
>   poly2:=[ [a+c,a+c], [b+d,b+d], [a+c,b+d] ];
>   z:=polygonplot (poly1,axes=normal,thickness=0,color=grey,linestyle=3);
>   z1:=polygonplot (poly3, axes=normal,thickness=0,color=grey,linestyle=3);
>   z2:=polygonplot (poly2, axes=normal,thickness=3,color=grey,linestyle=3);
>   m:=inequal ({x<=y},x=-(h+2)..h+2,y=-(h+2)..h+2, thickness=3
optionsfeasible=(color=grey), optionsexcluded=(color=white));
>   t:=textplot ([a,b,"A"],align={ABOVE,LEFT});
>   t2:=textplot ([c,d,"B"],align={ABOVE,LEFT});
>   t1:=textplot ([h+1,h+2,"IIR"]);
>   t3:=textplot ([a+c,b+d,"A+B"],align={ABOVE,LEFT});
>   return display ([w,m,t,z1,z2,z,t1,t2,t3,t4]);
>   printf (" A=[%.3f,%.3f];\n B=[%.3f,%.3f];\n A+B=[%.3f,%.3f];\n"
>   ,a,b,c,d,a+c,b+d);
> else
>   print ("Extremo inferior maior que o extremo superior, tente novamente.");
> end if;
> end:

```

geosub.msw

```

> #Representação Geométrica da Subtração entre Intervalos
> #O procedimento geosub determina um esboço gráfico para a Subtração
> #entre os intervalos A=[a, b] e B=[c, d].
>
> with(plots):
>
> geosub:=proc(a,b,c,d)
>   plotsetup(ps,plotoutput='intsub.ps',plotoptions='colour=cmyk,width=7.5cm,
>     height=7.5cm');   h:=max(abs(a), abs(b), abs(c),abs(d),abs(a-d),abs(b-c)):
>   w:=plot(y=x,x=-(h+2)..h+2);   t4:=textplot([0.25,h+2,"y"]):
>   if (a<=b and c<=d)
>     then
>       m:=inequal({x<=y},x=-(h+2)..h+2,y=-(h+2)..h+2, thickness=3,
>         optionsfeasible=(color=grey), optionsexcluded=(color=white));
>       poly1:=[a,b],[a,a],[b,b];
>       poly2:=[c,d],[c,c],[d,d];
>       poly3:=[a-d,a-d],[b-c,b-c],[a-d,b-c];
>       z:=polygonplot(poly1,axes=normal,thickness=0,color=grey,linestyle=3);
>       z1:=polygonplot(poly2,axes=normal,thickness=0,color=grey,linestyle=3);
>       z2:=polygonplot(poly3,axes=normal,thickness=3,color=grey,linestyle=3);
>       t:=textplot([a,b,"A"],align={ABOVE,LEFT});
>       t2:=textplot([c,d,"B"],align={ABOVE,LEFT});
>       t1:=textplot([h+1,h+2,"IIR"]);
>       t3:=textplot([a-d,b-c,"A-B"],align={ABOVE,LEFT});
>       return display([w,m,t,z1,z2,z,t1,t2,t3,t4]),printf(" A=[%.3f,%.3f];\n
>         B=[%.3f,%.3f]; \n A-B=[%.3f,%.3f];\n" ,a,b,c,d,a-d,b-c);
>     else
>       print("Extremo inferior maior que o extremo superior, tente novamente.");
>     end if;
>   end:

```

geomult.msw

```

> #O procedimento geomul determina um esboço gráfico para a Multiplicação
> #entre os intervalos A=[a, b] e B=[c, d].
>
> #Representação Geométrica da Multiplicação de Intervalos
>
> with(plots):
>
> geomul:=proc(a,b,c,d)
>   plotsetup(ps,plotoutput='intmult.ps',plotoptions='colour=cmymk,width=7.5cm,
>     height=7.5cm');
>   h:=max(abs(a), abs(b), abs(c),abs(d),abs(a*c),abs(a*d),abs(b*c),abs(b*d)):
>   w:=plot(y=x,x=-(h+2)..h+2); t4:=textplot([0.25,h+2,"y"]);
>   if (a<=b and c<=d)
>   then
>     m:=inequal({x<=y},x=-(h+2)..h+2,y=-(h+2)..h+2, thickness=3,
>       optionsfeasible=(color=grey),optionsexcluded=(color=white));
>     poly1:=[[a,b],[a,a],[b,b]];
>     poly2:=[[c,d],[c,c],[d,d]];
>     poly3:=[[min(a*c,a*d,b*c,b*d),max(a*c,a*d,b*c,b*d)],[min(a*c,a*d,b*c,
>       b*d),min(a*c,a*d,b*c,b*d)],[max(a*c,a*d,b*c,b*d),max(a*c,a*d,b*c,b*d)]];
>     z:=polygonplot(poly1,axes=normal,thickness=0,color=grey,linestyle=3);
>     z2:=polygonplot(poly2,axes=normal,thickness=0,color=grey,linestyle=3);
>     z3:=polygonplot(poly3,axes=normal,thickness=2,color=grey,linestyle=3);
>     t:=textplot([a,b,"A"],align={ABOVE,LEFT});
>     t1:=textplot([h+1,h+2,"IIR"]);
>     t2:=textplot([c,d,"B"],align={ABOVE,LEFT});
>     t3:=textplot([min(a*c,a*d,b*c,b*d),max(a*c,a*d,b*c,b*d),"A.B"],
>       align={ABOVE, LEFT});
>     return display([w,m,t,z3,z2,z,t1,t2,t3,t4]);
>     printf(" A=[%.3f,%.3f];\n B=[%.3f,%.3f];\n C=[%.3f,%.3f];\n" ,a,b,c,d,
>       min(a*c,a*d,b*c,b*d),max(a*c,a*d,b*c,b*d));
>   else
>     print("Extremo inferior maior que o extremo superior, tente novamente.");
>   end if;
> end:

```

geodiv.msw

```

> #O procedimento geodiv determina um esboço gráfico para a Divisão
> #entre os intervalos A=[a, b] e B=[c, d].
>
> #Representação Geométrica da Divisão de Intervalos
> with(plots):
>
> geodiv:=proc(a,b,c,d)
>   plotsetup(ps,plotoutput='intdiv.ps',plotoptions='colour=cmymk,width=7.5cm,
>     height=7.5cm');
>   h:=max(abs(a), abs(b), abs(c),abs(d),abs(a/c),abs(a/d),abs(b/c),abs(b/d)):
>   w:=plot(y=x,x=-(h+2)..h+2); t4:=textplot([0.25,h+2,"y"]);
>   if (a<=b and c<=d)
>   then
>     if (c>0 or d<0)
>     then
>       m:=inequal({x<=y},x=-(h+2)..h+2,y=-(h+2)..h+2,thickness=3,
>         optionsfeasible=(color=grey),optionsexcluded=(color=white));
>       poly1:=[[a,b],[a,a],[b,b]];
>       poly2:[[c,d],[c,c],[d,d]];
>       poly3:[[min(a/c,a/d,b/c,b/d),max(a/c,a/d,b/c,b/d)],[min(a/c,a/d,b/c,b/d),
>         min(a/c,a/d,b/c,b/d)],[max(a/c,a/d,b/c,b/d),max(a/c,a/d,b/c,b/d)]];
>       z:=polygonplot(poly1,axes=normal,thickness=0,color=grey,linestyle=3);
>       z2:=polygonplot(poly2,axes=normal,thickness=0,color=grey,linestyle=3);
>       z3:=polygonplot(poly3,axes=normal,thickness=3,color=grey,linestyle=3);
>       t:=textplot([a,b,"A"],align={ABOVE,LEFT});
>       t3:=textplot([h+1,h+2,"IIR"]);
>       t1:=textplot([c,d,"B"],align={ABOVE,LEFT});
>       t2:=textplot([min(a/c,a/d,b/c,b/d),max(a/c,a/d,b/c,b/d),"A/B"],
>         align={ABOVE, LEFT});
>       return display([w,m,t,z3,z2,z,t1,t2,t3,t4]);
>       printf(" A=[%.3f,%.3f];\n B=[%.3f,%.3f];\n C=[%.3f,%.3f];\n" ,a,b,c,d,
>         min(a/c,a/d,b/c,b/d),max(a/c,a/d,b/c,b/d));
>     else
>       " A divisão não está definida. 0 pertence à B ";
>     end if;
>   else
>     print("Extremo inferior maior que o extremo superior, tente novamente.");
>   end if;
> end:

```

geoposto.msw

```

> #O procedimento geoposto determina um esboço gráfico para o Negativo do
> #intervalo A=[a,b].
>
> #Representação Geométrica de Negativo de um Intervalo.
>
> with(plots):
>
> geoposto:=proc(a,b)
>   plotsetup(ps,plotoutput='intopost.ps',plotoptions='colour=cmyk,width=7.5cm,
>     height=7.5cm');
>   h:=max(abs(a), abs(b)): w:=plot(y=x,x=-(h+2)..h+2);
>   if (a<=b)
>     then
>       poly:=[[a,b],[a,a],[b,b]];
>       poly1:[[-b,-a],[-a,-a],[-b,-b]];
>       z:=polygonplot(poly,axes=normal,color=grey,thickness=0,linestyle=3);
>       z1:=polygonplot(poly1,axes=normal,color=grey,thickness=3,linestyle=3);
>       m:=inequal({x<=y},x=-(h+2)..h+2,y=-(h+2)..h+2,thickness=3,
>         optionsfeasible=(color=grey),optionsexcluded=(color=white));
>       t:=textplot([a,b,"A"],align={ABOVE,LEFT});
>       t1:=textplot([h+1,h+2,"IIR"]);
>       t2:=textplot([-b,-a,"-A"],align={ABOVE,LEFT});
>       t3:=textplot([0.25,h+2,"y"]);
>       return display([w,z,z1,m,t,t1,t2,t3]),printf("  A=[%.3f,%.3f];\n
>         -A=[%.3f,%.3f];\n",a,b,-b,-a);
>     else
>       print("Extremo inferior maior que o extremo superior, tente novamente.");
>     end if;
> end:

```

geoinv.msw

```

> #O procedimento geoinv determina um esboço gráfico para o Recíproco
> #do intervalo A=[a,b].
> #Representação Geométrica do Recíproco de um Intervalo.
>
> with(plots):
>
> geoinv:=proc(a,b)
>   plotsetup(ps,plotoutput='intinv.ps',plotoptions='colour=cmyk,width=7.5cm,
>     height=7.5cm');   h:=max(abs(a),abs(b), abs(1/a), abs(1/b)):
>   w:=plot(y=x,x=-(h+2)..h+2);   t3:=textplot([0.25,h+2,"y"]):
>   if (a<=b)
>     then
>       if (a>0 or b<0)
>         then
>           poly:=[a,b],[a,a],[b,b];
>           poly1:=[1/b,1/a],[1/a,1/a],[1/b,1/b];
>           z:=polygonplot(poly,axes=normal,color=grey,thickness=0,linestyle=3);
>           z1:=polygonplot(poly1,axes=normal,color=grey,thickness=3,linestyle=3);
>           m:=inequal({x<=y},x=-(h+2)..h+2,y=-(h+2)..h+2,thickness=3,
>             optionsfeasible=(color=grey),optionsexcluded=(color=white));
>           t:=textplot([a,b,"A"],align={ABOVE,LEFT});
>           t1:=textplot([h+1,h+2,"IIR"]);
>           t2:=textplot([1/b,1/a,"1/A"],align={ABOVE,LEFT});
>           return display([w,z,z1,m,t,t1,t2,t3]);
>           printf("  A=[%.3f,%.3f];\n 1/A=[%.3f,%.3f];\n",a,b,1/b,1/a);
>         else
>           " Não existe o recíproco de A. 0 pertence à A.";
>         end if;
>       else
>         print("Extremo inferior maior que o extremo superior, tente novamente.");
>       end if;
>     end;

```

A.2 Representação geométrica das operações entre intervalos como conjuntos

Os programas `geoint`, `geouni` e `geounicon` que correspondem respectivamente as operações entre os intervalos $A = [a, b]$ e $B = [c, d]$ de intersecção, união e união convexa necessitam de quatro parâmetros de entrada, os extremos a, b, c, d , nesta ordem, de A e B .

`geoint.msw`

```
> #O procedimento geoint determina o esboço gráfico da Intersecção
# entre os intervalos A=[a,b] e B=[c,d].
> #Representação da Intersecção.
>
> with(plots):
>
> geoint:=proc(a,b,c,d,n1,n2)
>   plotsetup(ps,plotoutput='intint.ps',plotoptions='colour=cmymk,width=7.5cm,
height=7.5cm'); h:=max(abs(a),abs(c),abs(b),abs(d)):
>   w:=plot(y=x,x=-(h+2)..h+2);
>   t3:=textplot([0.25,h+2,"y"]):
>   if (a<=b and c<=d)
>   then
>     if max(a,c) <= min(b,d)
>     then
>       poly1:=[a,b],[a,a],[b,b];
>       poly2:=[max(a,c),max(a,c)], [min(b,d),min(b,d)], [max(a,c),min(b,d)];
>       poly3:=[c,d],[c,c],[d,d];
>       z:=polygonplot(poly1,axes=normal,color=grey,thickness=0,linestyle=3);
>       z1:=polygonplot(poly2, axes=normal,color=grey,thickness=3,linestyle=3);
>       z2:=polygonplot(poly3, axes=normal,color=grey,thickness=0, linestyle=3);
>       m:=inequal({x<=y},x=-(h+2)..h+2,y=-(h+2)..h+2,thickness=3,
optionsfeasible=(color=grey),optionsexcluded=(color=white) );
>       t:=textplot([a,b,'A'],[c,d,'B'],align=ABOVE);
>       t1:=textplot([max(a,c),min(b,d),'I'],align={ABOVE,LEFT});
>       t2:=textplot([h+1,h+2,"IIR"]);
>       return display([w,z,z1,z2,m,t,t1,t3]),printf(" A=[%.3f,%.3f];
\n B=[%.3f,%.3f]; \n I=[%.3f,%.3f];\n",a,b,c,d,max(a,c),min(b,d));
>     else
>       "Intersecção vazia.";
>     end if
>   else
```

```
>     print("Extremo inferior maior que o extremo superior, tente novamente.");  
>     end if;  
> end:
```

geouni.msw

```

> #O procedimento geouni determina o esboço gráfico da União entre os
> intervalos A=[a,b] e B=[c,d].
>
> #Representação Geométrica da União de Intervalos.
>
> with(plots):
>
> geouni:=proc(a,b,c,d)
>   plotsetup(ps,plotoutput='intuni.ps',plotoptions='colour=cmymk,width=7.5cm,
>     height=7.5cm'); h:=max(abs(a),abs(c),abs(b),abs(d)):
>   w:=plot(y=x,x=-(h+2)..h+2); t0:=textplot([0.25,h+2,"y"]);
>   if (a<=b and c<=d)
>     then
>       if max(a,c) <= min(b,d)
>         then
>           poly1:=[[a,b],[a,a],[b,b]];
>           poly3:=[[c,d],[c,c],[d,d]];
>           poly2:=[[min(a,c),min(a,c)],[max(b,d),max(b,d)],[min(a,c),max(b,d)]];
>           z:=polygonplot(poly1,axes=normal,thickness=0,color=grey,linestyle=3);
>           z1:=polygonplot(poly3, axes=normal,thickness=0,color=grey, linestyle=3);
>           z2:=polygonplot(poly2, axes=normal,thickness=3,color=grey, linestyle=3);
>           m:=inequal({x<=y},x=-(h+2)..h+2,y=-(h+2)..h+2,thickness=3,
>             optionsfeasible=(color=grey),optionsexcluded=(color=white));
>           t:=textplot([a,b,'A'],[c,d,'B'],align={ABOVE,LEFT});
>           t1:=textplot([min(a,c),max(b,d),'U'],align={ABOVE,LEFT});
>           t2:=textplot([h+1,h+2,"IIR"]);
>           return display([w,z,z1,z2,m,t,t1,t2,t0]),printf(" A=[%.3f,%.3f];
>             \n B=[%.3f,%.3f]; \n U=[%.3f,%.3f];\n",a,b,c,d,min(a,c),max(b,d));
>         else
>           "intersecção vazia, não é definida a união.";
>         end if
>       else
>         print("Extremo inferior maior que o extremo superior, tente novamente.");
>       end if;
> end:

```

geounicon.msw

```

> #O procedimento geounicon determina o esboço gráfico da União Convexa entre
  # os intervalos A=[a,b] e B=[c,d].
>
> #Representação Geométrica da Uniao Convexa
>
> with(plots):
>
> geounicon:=proc(a,b,c,d)
>   plotsetup(ps,plotoutput='intuniconv.ps',plotoptions='colour=cmyk,
     width=7.5cm,height=7.5cm'); h:=max(abs(a),abs(c),abs(b),abs(d)):
>   w:=plot(y=x,x=-(h+2)..h+2):
>   t3:=textplot([0.25, h+2,"y"]);
>   if (a<=b and c<=d)
>     then
>       poly1:=[[a,b],[a,a],[b,b]];
>       poly3:=[[c,d],[c,c],[d,d]];
>       poly2:[[min(a,c),min(a,c)],[max(b,d),max(b,d)],[min(a,c),max(b,d)]];
>       z:=polygonplot(poly1,axes=normal,thickness=0,color=grey, linestyle=3);
>       z1:=polygonplot(poly3, axes=normal,thickness=0,color=grey, linestyle=3);
>       z2:=polygonplot(poly2, axes=normal,thickness=3,color=grey, linestyle=3);
>       m:=inequal({x<=y},x=-(h+2)..h+2,y=-(h+2)..h+2,thickness=3,
         optionsfeasible=(color=grey), optionsexcluded=(color=white));
>       t:=textplot([a,b,'A'],[c,d,'B'],align={ABOVE,LEFT});
>       t1:=textplot([min(a,c),max(b,d),'U'],align={ABOVE,LEFT});
>       t2:=textplot([h+1,h+2,"IIR"]);
>       return display([w,z,z1,z2,m,t,t1,t2,t3]),printf(" A=[%.3f,%.3f];
         \n B=[%.3f,%.3f]; \n U=[%.3f,%.3f];\n",a,b,c,d,min(a,c),max(b,d));
>     else
>       print("Extremo inferior maior que o extremo superior,tente novamente.");
>     end if;
> end:

```

A.3 Representação geométrica de conceitos topológicos

Os programas `geobaberta`, `geobfechada` e `geoesfera` que correspondem aos elementos topológicos bola aberta, bola fechada e esfera de centro A e raio $r > 0$, respectivamente, necessitam de três parâmetros de entrada, os extremos a , b , nesta ordem, de A e o raio r . Por exemplo, a representação da esfera de centro A e raio r é gerada por `geoesfera(a, b, r)`.

`geobaberta.msw`

```
> #O procedimento geobaberta determina o esboço gráfico da Bola Aberta de
  #centro [a,b] e raio r dados.
> # Bola Aberta de Centro [a,b] e Raio r
>
> with(plots):
>
> geobaberta:=proc(a,b,r)
>   plotsetup(ps,plotoutput='intaberta2.ps',plotoptions='colour=cmymk,
     width=7.5cm,height=7.5cm');
>   h:=max(abs(a-r),abs(a+r),abs(b-r),abs(b+r)):
>   w:=plot(x,x=-(h+2)..h+2); t6:=textplot([0.25,h+2,"y"]);
>   if a<=b
>     then
>       if r <= (b-a)/2
>         then
>           poly:=[[a-r,b-r],[a-r,b+r],[a+r,b+r],[a+r,b-r]];
>           z:=polygonplot(poly,axes=normal,color=pink,thickness=0, linestyle=3);
>           t1:=textplot([a-r,b-r,"A"],align={BELOW,LEFT});
>           t3:=textplot([a+r,b-r,"D"],align={BELOW,RIGHT});
>           m:=inequal({x<=y},x=-(h+2)..h+2,y=-(h+2)..h+2,thickness=2,
             optionsfeasible=(color=grey),optionsexcluded=(color=white));
>           t:=textplot([a,b,"P"]);
>           t2:=textplot([a-r,b+r,"B"], align={ABOVE,LEFT}):
>           t4:=textplot([a+r,b+r,"C"], align={ABOVE,RIGHT}):
>           t5:=textplot([h+1,h+2,"IIR"]):
>           return display([w,z,m,t,t1,t2,t3,t4,t5,t6]);
>           printf(" P=[%.3f,%.3f], r=%.3f;\n A=[%.3f,%.3f], B=[%.3f,%.3f];
>           \n C=[%.3f,%.3f],
>           D=[%.3f,%.3f].\n",a,b,r,a-r,b-r,a-r,b+r,a+r,b+r,a+r,b-r);
>         else
>           poly:=[[a-r,b-r],[a-r,b+r],[a+r,b+r],[a+r,a+r],[b-r,b-r]];
>           t0:=textplot([a-r,b-r,"B"],align={BELOW,LEFT});
```

```

> t2:=textplot([a+r,a+r,'E'],align={BELOW,RIGHT});
> t3:=textplot([a+r,b+r,'D'],align={ABOVE,RIGHT});
> t4:=textplot([a-r,b+r,'C'],align={ABOVE,LEFT});
> t1:=textplot([b-r,b-r,'A'],align={BELOW,RIGHT});
> t5:=textplot([h+1,h+2,"IIR"]):
> z:=polygonplot(poly,axes=normal,color=pink,thickness=0,linestyle=3);
> m:=inequal({x<=y},x=-(h+2)..h+2,y=-(h+2)..h+2,thickness=3
,optionsfeasible=(color=grey),optionsexcluded=(color=white));
> t:=textplot([a,b,"P"]);
> return display([w,z,m,t,t0,t1,t2,t3,t4,t5,t6]);
printf(" P=[%.3f,%.3f], r=%.3f;\n A=[%.3f,%.3f], B=[%.3f,%.3f];
\n C=[%.3f,%.3f], D=[%.3f,%.3f];\n E=[%.3f,%.3f].
\n",a,b,r,b-r,b-r,a-r,b-r,a-r, b+r,a+r,b+r,a+r,a+r);
> end if
> else
> print("Extremo inferior maior que o extremo superior, tente novamente.");
> end if;
> end:

```

geobfechada.msw

```

> #0 procedimento geobfechada determina o esboço gráfico da Bola Fechada de
> #centro [a,b] e raio r dados.
> #Bola Fechada de Centro [a,b] e Raio r.
>
> with(plots):
>
> geobfechada:=proc(a,b,r)
>   plotsetup(ps,plotoutput='intfechada2.ps',plotoptions='colour=cmymk,
>     width=7.5cm, height=7.5cm');
>   h:=max(abs(a-r),abs(a+r),abs(b-r),abs(b+r)):
>   w:=plot(y=x,x=-(h+2)..h+2); t6:=textplot([0.25,h+2,"y"]):
>   if a<=b
>     then
>       if r <= (b-a)/2
>         then
>           poly:=[a-r,b-r],[a-r,b+r],[a+r,b+r],[a+r,b-r];
>           z:=polygonplot(poly,axes=normal,color=pink,thickness=3);
>           t1:=textplot([a-r,b-r,"A"],align={BELOW,LEFT});
>           t2:=textplot([a+r,b-r,"D"],align={BELOW,RIGHT});
>           m:=inequal({x<=y},x=-(h+2)..h+2,y=-(h+2)..h+2,thickness=2,
>             optionsfeasible=(color=grey),optionsexcluded=(color=white));
>           t:=textplot([a,b,"P"]);
>           t3:=textplot([a-r,b+r,"B"], align={ABOVE,LEFT}):
>           t4:=textplot([a+r,b+r,"C"], align={ABOVE,RIGHT}):
>           t5:=textplot([h+1,h+2,"IIR"]): return
>             display([w,z,m,t,t1,t2,t3,t4,t5,t6]);
>           printf(" P=[%.3f,%.3f], r=%.3f;\n A=[%.3f,%.3f], B=[%.3f,%.3f];\n
>             C=[%.3f,%.3f], D=[%.3f,%.3f].\n",a,b,r,a-r,b-r,a-r,b+r,a+r,b+r,a+r,b-r);
>         else
>           poly:=[a-r,b-r],[a-r,b+r],[a+r,b+r],[a+r,a+r],[b-r,b-r];
>           z:=polygonplot(poly,axes=normal,color=pink,thickness=3);
>           t0:=textplot([a-r,b-r,"B"],align={BELO,LEFT});
>           t1:=textplot([b-r,b-r,"A"],align={BELOW,RIGHT});
>           t2:=textplot([a+r,a+r,"E"],align={BELOW,RIGHT});
>           t3:=textplot([a+r,b+r,"D"], align={ABOVE,RIGHT}):
>           t4:=textplot([a-r,b+r,"C"],align={ABOVE,LEFT});
>           t5:=textplot([h+1,h+2,"IIR"]):
>           m:=inequal({x<=y},x=-(h+2)..h+2,y=-(h+2)..h+2,thickness=1,
>             optionsfeasible=(color=grey),optionsexcluded=(color=white));
>           t:=textplot([a,b,"P"]);
>           return display([w,z,m,t,t0,t1,t2,t3,t4,t5,t6]);

```

```
printf("P=[%.3f,%.3f], r=%.3f;\n A=[%.3f,%.3f], B=[%.3f,%.3f];\n  
C=[%.3f,%.3f], D=[%.3f,%.3f];\n E=[%.3f,%.3f]. \n",a,b,r,b-r,b-r,  
a-r,b-r,a-r,b+r,a+r,b+r,a+r,a+r);  
>     end if  
>     else  
>         print("Extremo inferior maior que o extremo superior, tente novamente.");  
>     end if;  
> end:
```

geoesfera.msw

```

> #0 procedimento geoesfera determina o esboço gráfico da Esfera de centro
    #[a,b] e raio r dados.
> # Esfera de Centro [a,b] e Raio r
>
> with(plots):
>
> geoesfera:=proc(a,b,r)
>   plotsetup(ps,plotoutput='intesfera2.ps',plotoptions='colour=cmymk,
    width=7.5cm,height=7.5cm');
>   h:=max(abs(a-r),abs(a+r),abs(b-r),abs(b+r)):   w:=plot(y=x,x=-(h+2)..h+2,
    color=black);
    t10:=textplot([0.25,h+2,"y"]):
>   if a<=b
>     then if r <= (b-a)/2 then
>       poly:=[a-r,b-r],[a-r,b+r],[a+r,b+r],[a+r,b-r];
>       z:=polygonplot(poly,axes=normal,color=grey,thickness=3);
>       t:=textplot([a,b,"P"]);
>       t1:=textplot([a-r,b-r,"A"],align={BELOW,LEFT});
>       t3:=textplot([a+r,b-r,"D"],align={BELOW,RIGHT});
>       m:=inequal({x<=y},x=-(h+2)..h+2,y=-(h+2)..h+2,thickness=2,
        optionsfeasible=(color=grey),optionsexcluded=(color=white));
>       t2:=textplot([a-r,b+r,"B"],align={ABOVE,LEFT}):
>       t4:=textplot([a+r,b+r,"C"],align={ABOVE,RIGHT}):
>       t5:=textplot([h+1,h+2,"IIR"]):
>       return display([w,z,m,t,t1,t2,t3,t4,t5,t10]);
        printf(" P=[%.3f,%.3f], r=%.3f;\n A=[%.3f,%.3f],B=[%.3f,%.3f];
          \n C=[%.3f,%.3f], D=[%.3f,%.3f].
          \n",a,b,r,a-r,b-r,a-r,b+r,a+r,b+r,a+r,b-r);
>     else
>       t0:=textplot([a-r,b-r,"B"],align={BELOWLEFT});
>       t1:=textplot([b-r,b-r,"A"],align={BELOW,RIGHT});
>       t2:=textplot([a+r,a+r,"E"],align={BELOW,RIGHT});
>       t3:=textplot([a+r,b+r,"D"],align={ABOVE,RIGHT}):
>       t4:=textplot([a-r,b+r,"C"],align={ABOVE,LEFT}):
>       t5:=textplot([h+1,h+2,"IIR"]):
>       z:=pointplot([b-r,b-r],[a-r,b-r],[a-r,b+r],[a+r,b+r],[a+r,a+r],
        thickness=3, style=line);
>       m:=inequal({x<=y},x=-(h+2)..h+2,y=-(h+2)..h+2,thickness=2,
        optionsfeasible=(color=grey),optionsexcluded=(color=white));
>       t:=textplot([a,b,"P"]);
>       return display([w,z,m,t,t0,t1,t2,t3,t4,t5,t10]);

```

```
printf(" P=[%.3f,%.3f], r=%.3f;\n A=[%.3f,%.3f], B=[%.3f,%.3f];\n C=[%.3f,%.3f], D=[%.3f,%.3f]; \n E=[%.3f,%.3f].\n",a,b,r,b-r,b-r,a-r,b-r,a-r, b+r,a+r,b+r,a+r,a+r);\n> end if\n> else\n>     print("Extremo inferior maior que o extremo superior, tente novamente");\n> end if;\n> end:
```

Implementação dos Métodos Intervalares no MatLab/IntLab

Este apêndice apresenta, no MatLab [MatLab04], implementações dos intervalos encapsuladores para as distribuições Exponencial, Normal Padrão e Uniforme. O INTLAB - INTerval LABoratory (versão 5.3) foi desenvolvido por [Rump06] e é obtido em

< www.ti3.harburg.de/~rump/intlab >.

Todas as rotinas no IntLab são arquivos M de funções do MatLab. O IntLab usa a rotina *setround* para trocar, no processador, quando necessário, os modos de arredondamento: mais próximo, para cima e para baixo. Esta rotina foi usada para desenvolver funções de entrada (input), saída (output) e aritmética de intervalos.

B.1 Conceitos Básicos do IntLab

Comandos básicos:

- (i) » `startintlab` - inicia o IntLab.
- (ii) » `help intval` - lista completa das funções no IntLab.
- (iii) » `intvalinit('DisplayInfsup')` - mostra o intervalo usando o ínfimo e o supremo do intervalo.
- (iv) » `intvalinit('DisplayMidrad')` - mostra o intervalo usando o ponto médio e o raio do intervalo.
- (v) » `intvalinit('Display')` - mostra representação de incerteza.

Tem-se quatro possibilidades para gerar intervalos, a saber:

Sejam X um intervalo, e a, b, c , e d números reais.

- (i) » $X = \text{infsup}(a, b)$ - gera intervalo X de ínfimo a e supremo b arredondando, quando necessário, a para baixo e b para cima.

(ii) » $X = \text{midrad}(a, c)$ - gera o intervalo $X = [a - c, a + c]$ a partir de seu ponto médio e raio.

(iii) » $X = \text{intval}(a)$ - gerar um intervalo encapsulador de menor amplitude para a .

Funções Básicas

Estas funções são fundamentais para a computação intervalar. Sejam X e W intervalos, então:

(a) $\text{abs}(X)$ - valor absoluto de X .

(b) $\text{diam}(X)$ - diâmetro de X

(c) $\text{in}(\text{arg}, X)$ - retorna 1 se arg pertence (ou está contido) a X .

(d) $\text{inf}(X)$ - limite inferior de X .

(e) $\text{intersect}(X, W)$ - calcula a interseção entre X e W .

(f) $\text{hull}(X, W)$ - união convexa entre X e W .

(g) $\text{mag}(X)$ - Magnitude (maior valor absoluto) de X .

(h) $\text{mid}(X)$ - ponto médio de X .

(i) $\text{mig}(X)$ - Mignitude (menor valor absoluto).

(j) $\text{qdist}(X, W)$ - distância entre X e W .

(k) $\text{rad}(X)$ - raio do intervalo X .

(l) $\text{sup}(X)$ - limite superior de X .

As operações aritméticas intervalares $\{+, -, *, /\}$ são as definidas no Matlab.

Exemplo 38. Para os intervalos abaixo determine: disp_- , infsup , midrad , rad , in0 . Use precisão simples e dupla.

(i) Se $X = \text{midrad}(\pi, 1e-8)$, tem-se

```
>> X=midrad(pi,1e-8)
intval X =
          3.1415
>> format short
>> infsup(X)
```

```

intval X =
    [3.1415, 3.1416]
>> midrad(X)
intval X =
    <3.1416, 0.0001>
>> disp_(X)
intval X =
    3.1415
>> in0(pi,X)
ans =
    1
>> rad(X)
ans =
    1.0000e-008

>> format long
>> X=midrad(pi,1e-8)
intval X =
    3.1415927_____
>> infsup(X)
intval X =
    [3.14159264358979, 3.14159266358980]
>> midrad(X)
intval X =
    <3.14159265358979, 0.00000001000001>
>> disp_(X)
intval X =
    3.1415927_____
>> in0(pi,X)
ans =
    1
>> rad(X)
ans =
    1.000000038331450e-008.

```

(ii) Se $X = \text{intval}(0.3)$, tem-se

```

X=intval(0.3)
intval X =
    0.3000
>> infsup(X)
intval X =
    [0.2999, 0.3000]

```

```

>> midrad(X)
intval X =
    <0.3000, 0.0001>
>> rad(X)
ans =
    0
>> in0(0.3, X)
ans =
    0

>> format long
>> X=intval(0.3)
intval X =
    0.3000000000000000
>> infsup(X)
intval X =
    [0.299999999999998, 0.300000000000000]
>> midrad(X)
intval X =
    <0.300000000000000, 0.000000000000001>
>> disp_(X)
intval X =
    0.3000000000000000
>> in0(0.3, X)
ans =
    0
>> rad(X)
ans =
    0

```

(iii) Se $X = \text{intval}('0.3')$, tem-se

```

>>format short
>> X=intval('0.3')
intval X =
    0.3000
>> infsup(X)
intval X =
    [0.2999, 0.3001]
>> midrad(X)
intval X =
    <0.3000, 0.0001>
>> disp_(X)

```

```

intval X =
    0.3000
>> in0(0.3, X)
ans =
    1
>> rad(X)
ans =
    5.5511e-017

>> format long
>> X=intval('0.3')
intval X =
    0.3000000000000000
>> infsup(X)
intval X =
    [0.2999999999999998, 0.3000000000000001]
>> midrad(X)
intval X =
    <0.3000000000000000, 0.0000000000000001>
>> disp_(X)
intval X =
    0.3000000000000000
>> in0(0.3, X)
ans =
    1
>> in0('0.3', X)
ans =
    0
>> rad(X)
ans =
    5.551115123125783e-017

```

Exemplo 39. Sejam $X = [-2, 3]$ e $W = [-2.5, 2]$. Determine $X + W$, $X - W$, $X * W$, X/W , $\text{intersect}(X, W)$, $\text{hull}(X, W)$ e $\text{qdist}(X, W)$. Use precisão simples.

```

>> format short
>> X=insup(-2,3);
>> W=insup(-2.5,2);
>> intvalinit('DisplayInfSup')
==> Default display of intervals by infimum/supremum
>> X+W

```

```

intval ans =
    [-4.5000, 5.0000]
>> X-W
intval ans =
    [-4.0000, 5.5000]
>> X*W
intval ans =
    [-7.5000, 6.0000]
>> X/W
intval ans =
    [NaN, NaN]
>> intersect(X,W)
intval ans =
    [-2.0000, 2.0000]
>> hull(X,W)
intval ans =
    [-2.5000, 3.0000]
>> qdist(X,W)
ans =
    1

```

B.2 Implementações dos intervalos encapsuladores no MatLab/IntLab

Os programas abaixo foram implementados no MatLab [MatLab04], usando o IntLab 5.3 [Rump06], e geram intervalos encapsuladores para as variáveis aleatórias contínuas Exponencial, Normal Padrão e Uniforme.

B.2.1 Implementação da fórmula de Simpson intervalar

A função *rsimpson1* calcula um intervalo encapsulador para a integral definida de uma função f no intervalo X . Necessita de quatro argumentos de entrada: as imagens de f nos extremos e no ponto médio do intervalo, e o intervalo contendo o conjunto imagem da derivada de ordem 4 da função f no intervalo dado. Esta função é chamada pelas funções *dexp* e *normal1*.

```

%fomula de Simpson intervalar
%a imagem, da funcao a ser integrada, no extremo inferior do intervalo
%b imagem, da funcao a ser integrada, no extremo superior do intervalo
%m ponto medio do intervalo
%d amplitude do intervalo
%z intervalo contendo o ranger da derivada de ordem 4

```

```
function [rs]=rsimpson(a,b,m,d,z)

y=(d/6)*(a + 4*m + b)-((d^5)/2880)*z;
rs=y;
```

B.2.2 Implementação do intervalo encapsulador para a densidade Exponencial

A função *dexp* determina o intervalo encapsulador para uma variável aleatória Exponencial de parâmetro c e necessita de quatro argumentos de entrada: os limites de integração, o parâmetro c e o número de subintervalos da partição.

```
%Calculo do intervalo encapsulador para a variavel exponencial usando o metodo
% de simpson intervalar
%a, b e c sao os argumentos funcao dexp
% a=limite inferior do intervalo de integracao
% b= limite superior do intervalo de integracao
% c= parametro da distribuicao exponencial
% p=numero de subintervalos de [a,b]
% a funcao rsimpson1 e chamada

function [S]=dexp(a,b,c,p)

if (b==inf)&(a<=0)
    s=intval(1)
elseif (b==inf)&(a>0)
    setround(-1)
    s1=exp(-c*a);
    setround(1)
    s2=exp(-c*a);
    setround(0)
    s=infsup(s1,s2)
    %s=(exp(intval(-c*a)))
elseif (a<0)&(b~=inf)&(b>0)
    s=1-(exp(intval(-c*b)));
elseif (a~-=-inf)&(a<0)&(b<0)
    s=0
else (a>=0)&(b>0)&(b~=inf)
    syms z
    yy=c*exp(-c*z);%funcao
    y=diff(yy,4);%derivada de ordem 4 da funcao yy
    %e=infsup(a,b)
    h=(b-a)/p; % comprimento de cada subintervalo
```

```

n=1:p+1
x(n)=a + h*(n-1); %vetor com os extremos dos subintervalos
s=0;
for i=1:p
    t(i)=infsup(x(i),x(i+1)); %criando os subintervalos
    m(i)=mid(t(i)); % ponto medio do intervalo i
    f(i)=subs(yy,z,x(i)); %calculando o valor de yy em x(i)
    f(i+1)=subs(yy,z,x(i+1)); %calculando o valor de yy em x(i+1)
    fm(i)=subs(yy,z,m(i)); %calculando o valor de yy em m(i)
    if (c<0); %calcula uma extensao intervalar da derivada de ordem 4
        g(i)=infsup((c^5)*exp(-c*x(i)), (c^5)*exp(-c*x(i+1)));
        %ou
    else
        g(i)=infsup((c^5)*exp(-c*x(i+1)), (c^5)*exp(-c*x(i)));
    end
    intvalinit('DisplayInfSup');
    %determinando o intervalo encapsulador usando de Simpson intervalar
    %(h^2/6)*(f(x(i))+4*f(m(i)) + f(x(i+1)))-(h^5/2880)*extensao intervalar da
    %derivada de ordem 4 .
    s=s + rsimpson1(f(i),f(i+1),fm(i),h,g(i));
end
end
intvalinit('DisplayInfSup')

S=s %calculo da integral de (c*exp((-c*x) distribuicao exponencial de
%media 1/c 0 e variancia 1/c^2

r=rad(S)

```

B.2.3 Implementação do Método de Newton Intervalar

```
%calcula um intervalo contendo o extremo relativo da derivada de ordem 4
%da funcao normal padrao os extremos relativos desta funcao sao
% +/- sqrt(5+sqrt(10)), +/- sqrt(5-sqrt(10)),0
%usando uma versao intervalar do Metodo de Newton para encontrar um
%intervalo encapsulador para estes extremos relativos
%os arquivos extder4, extder5 e extder6 calculam as derivadas de ordem 4,
% 5 e 6 respectivamente da funcao normal Padrão
function [w1]=f1(a,b)

w=infsup(a,b);
z=mid(w);
f=extder5(z); %calcula a derivada de ordem 5 no ponto medio.
ff=extder5(w); %calcula derivada de ordem 5 no intervalo w.
g=extder6(w); %calcula a derivada de ordem 6 em w.
ss=infsup(g);
bb=extder4(w); %calcula o derivada de ordem 4 em w.
t=rad(g);
if in(0, ff)==0
    fprintf('nao existe zero da funcao neste intervalo')

else
    nn=mid(w)- f/g;
    w1=intersect(nn,w);
    while diam(w1)> 10^(-10)
        z1=mid(w1);
        %formula de Newton intervalar para calculo de raiz.
        nn1=mid(w1)-extder5(z1)/extder6(w1);
        w1=intersect(nn1,w1);

    end

    ww=infsup(w1);
    vv=diam(w1);
    rr=infsup(extder6(w1));
    yy=infsup(extder5(w1));
    tt=infsup(ff);
    r1=infsup(extder4(w1));
    t1=rad(r1);
    yyy=rad(yy);
    %intersect(infsup(2.878,2.979),w1);

end
%
```

B.2.4 Implementação da extensão intervalar inclusão monotônica derivada de ordem 4 da função $\exp(-x^2/2)$

A função *compox* calcula, conhecendo-se os extremos relativos da derivada de ordem 4 da função $f(x) = \frac{1}{\sqrt{2\pi}}\exp(-x^2/2)$, um intervalo contendo o conjunto imagem desta derivada num intervalo X . Esta função é chamada pela função *normal1*.

```
%Construcao da funcao G, extensao intervalar inclusao monotonica para derivada
%de ordem 4 da normal Padrão, no intervalo [a,b]
%os intervalos encapsuladores para os extremose relativos sao determinados
% usando o metodo Newton Intervalar (arquivo fl.m)
function [nn]=compox(a,b)

x=infsup(a,b);
y1=a;
y2=b;
%gera o intervalo contendo maximo relativo -sqrt(5+sqrt(10))
b1=f1(-2.9,-2.8);
%bb1=infsup(b1);
b2=f1(-1.4,-1.3);%gera o intervalo contendo minimo relativo -sqrt(5-sqrt(10))
%bb2=infsup(b2);
b3=f1(-0.1,0.2);%gera o intervalo contendo maximo relativo 0
%bb3=infsup(b3);
b4=f1(1.3,1.4);%gera o intervalo contendo minimo relativo sqrt(5-sqrt(10))
%bb4=infsup(b4);
b5=f1(2.8, 2.9);%gera o intervalo contendo maximo relativo sqrt(5+sqrt(10))
%bb5=infsup(b5);
% os ci's representam os intervalos onde a funcao e crescente
% e as di's onde a funcao e decrescente
c1=infsup(-inf, inf(b1));
%cc1=infsup(c1)
d1=infsup(sup(b1), inf(b2));
c2=infsup(sup(b2),inf(b3));
%cc2=infsup(c2)
d2=infsup(sup(b3),inf(b4));
c3=infsup(sup(b4),inf(b5));
%cc3=infsup(c3)
d3=infsup(sup(b5), inf);
%decomposicao de x como intersecao de intervalos
c=[c1 c2 c3];
d=[d1 d2 d3];
b=[b1 b2 b3 b4 b5];
sdd=rad(b(3));
cc=intval([0 0 0]);
```

```

dd=intval([0 0 0]);
for i=1:3
    c(i)=intersect(c(i),x);
    d(i)=intersect(d(i),x);
end
pp=infsup(c);
pp1=infsup(d);

for i=1:5
    b(i)=intersect(b(i),x);
    %if b(i)~=nan
    %    bb(i)=extder4(b(i));
    % end
end
pp2=infsup(b);

if ((c(1)~=nan)&(b(1)==nan))
    cc=infsup(extder4(inf(c1)),extder4(sup(c1)));
    nn=hull(cc);

elseif ((c(1)~=nan)&(b(1)~=nan)&(d(1)==nan))
    cc=infsup(extder4(inf(c1)),extder4(sup(c1)));
    bb=(extder4(b(1)));
    nn=hull(cc,bb);

elseif ((c(1)~=nan)&(b(1)~=nan)&(d(1)~=nan)&(b(2)==nan))
    cc=infsup(extder4(inf(c(1))),extder4(sup(c(1))));
    bb=(extder4(b(1)));
    dd=infsup(extder4(sup(d(1))),extder4(inf(d(1))));
    nn=hull(cc,bb,dd);

elseif ((c(1)~=nan)&(b(1)~=nan)&(d(1)~=nan)&(b(2)~=nan)&(c(2)==nan))
    cc=infsup(extder4(inf(c(1))),extder4(sup(c(1))));
    bb=(extder4(b(1)));
    dd=infsup(extder4(sup(d(1))),extder4(inf(d(1))));
    bb1=extder4(b(2));
    nn=hull(cc,bb,dd,bb1);

elseif ((c(1)~=nan)&(b(1)~=nan)&(d(1)~=nan)&(b(2)~=nan)&(c(2)~=nan)&..
    (b(3)==nan))
    cc=infsup(extder4(inf(c(1))),extder4(sup(c(1))));
    bb=(extder4(b(1)));
    dd=infsup(extder4(sup(d(1))),extder4(inf(d(1))));
    bb1=extder4(b(2));

```

```

cc1=infsup(extder4 (inf (c (2))) , extder4 (sup (c (2)))) ;
nn=hull (cc,bb,dd,bb1,cc1) ;

elseif ((c (1)~=nan) & (b (1)~=nan) & (d (1)~=nan) & (b (2)~=nan) & (c (2)~=nan) & . .
        (b (3)~=nan) & (d (2)~=nan) )
cc=infsup(extder4 (inf (c (1))) , extder4 (sup (c (1)))) ;
bb=(extder4 (b (1))) ;
dd=infsup(extder4 (sup (d (1))) , extder4 (inf (d (1)))) ;
bb1=extder4 (b (2)) ;
cc1=infsup(extder4 (inf (c (2))) , extder4 (sup (c (2)))) ;
bb2=extder4 (b (3)) ;
nn=hull (cc,bb,dd,bb1,cc1,bb2) ;

elseif ((c (1)~=nan) & (b (1)~=nan) & (d (1)~=nan) & (b (2)~=nan) & (c (2)~=nan) & . .
        (b (3)~=nan) & (d (2)~=nan) & (b (4)~=nan) )
cc=infsup(extder4 (inf (c (1))) , extder4 (sup (c (1)))) ;
bb=(extder4 (b (1))) ;
dd=infsup(extder4 (sup (d (1))) , extder4 (inf (d (1)))) ;
bb1=extder4 (b (2)) ;
cc1=infsup(extder4 (inf (c (2))) , extder4 (sup (c (2)))) ;
bb2=extder4 (b (3)) ;
dd1=infsup(extder4 (sup (d (2))) , extder4 (inf (d (2)))) ;
nn=hull (cc,bb,dd,bb1,cc1,bb2,dd1) ;

elseif ((c (1)~=nan) & (b (1)~=nan) & (d (1)~=nan) & (b (2)~=nan) & (c (2)~=nan) & . .
        (b (3)~=nan) & (d (2)~=nan) & (b (4)~=nan) & (c (3)~=nan) )
cc=infsup(extder4 (inf (c (1))) , extder4 (sup (c (1)))) ;
bb=(extder4 (b (1))) ;
dd=infsup(extder4 (sup (d (1))) , extder4 (inf (d (1)))) ;
bb1=extder4 (b (2)) ;
cc1=infsup(extder4 (inf (c (2))) , extder4 (sup (c (2)))) ;
bb2=extder4 (b (3)) ;
dd1=infsup(extder4 (sup (d (2))) , extder4 (inf (d (2)))) ;
bb3=extder4 (b (4)) ;
nn=hull (cc,bb,dd,bb1,cc1,bb2,dd1,bb3) ;

elseif ((c (1)~=nan) & (b (1)~=nan) & (d (1)~=nan) & (b (2)~=nan) & (c (2)~=nan) & . .
        (b (3)~=nan) & (d (2)~=nan) & (b (4)~=nan) & (c (3)~=nan) & (b (5)~=nan) )
cc=infsup(extder4 (inf (c (1))) , extder4 (sup (c (1)))) ;
bb=(extder4 (b (1))) ;
dd=infsup(extder4 (sup (d (1))) , extder4 (inf (d (1)))) ;
bb1=extder4 (b (2)) ;
cc1=infsup(extder4 (inf (c (2))) , extder4 (sup (c (2)))) ;
bb2=extder4 (b (3)) ;

```

```

dd1=infsup(extder4(sup(d(2))),extder4(inf(d(2))));
bb3=extder4(b(4));
cc2=infsup(extder4(inf(c(3))),extder4(sup(c(3))));
nn=hull(cc,bb,dd,bb1,cc1,bb2,dd1,bb3,cc2);

elseif ((c(1)~=nan)&(b(1)~=nan)&(d(1)~=nan)&(b(2)~=nan)&(c(2)~=nan)&..
        (b(3)~=nan)&(d(2)~=nan)&(b(4)~=nan)&(c(3)~=nan)&(b(5)~=nan)&..
        (d(3)==nan))
cc=infsup(extder4(inf(c(1))),extder4(sup(c(1))));
bb=(extder4(b(1)));
dd=infsup(extder4(sup(d(1))),extder4(inf(d(1))));
bb1=extder4(b(2));
cc1=infsup(extder4(inf(c(2))),extder4(sup(c(2))));
bb2=extder4(b(3));
dd1=infsup(extder4(sup(d(2))),extder4(inf(d(2))));
bb3=extder4(b(4));
cc2=infsup(extder4(inf(c(3))),extder4(sup(c(3))));
bb4=extder4(b(5));
nn=hull(cc,bb,dd,bb1,cc1,bb2,dd1,bb3,cc2,bb4);

elseif ((c(1)~=nan)&(b(1)~=nan)&(d(1)~=nan)&(b(2)~=nan)&(c(2)~=nan)&..
        (b(3)~=nan)&(d(2)~=nan)&(b(4)~=nan)&(c(3)~=nan)&(b(5)~=nan)&(d(3)~=nan))
cc=infsup(extder4(inf(c(1))),extder4(sup(c(1))));
bb=(extder4(b(1)));
dd=infsup(extder4(sup(d(1))),extder4(inf(d(1))));
bb1=extder4(b(2));
cc1=infsup(extder4(inf(c(2))),extder4(sup(c(2))));
bb2=extder4(b(3));
dd1=infsup(extder4(sup(d(2))),extder4(inf(d(2))));
bb3=extder4(b(4));
cc2=infsup(extder4(inf(c(3))),extder4(sup(c(3))));
bb4=extder4(b(5));
dd2=infsup(extder4(sup(d(3))),extder4(inf(d(3))));
nn=hull(cc,bb,dd,bb1,cc1,bb2,dd1,bb3,cc2,bb4,dd2);

%segunda etapa

elseif ((c(1)==nan)&(b(1)~=nan)&(d(1)==nan))
        bb=extder5(x);
        nn=hull(bb);

elseif ((c(1)==nan)&(b(1)~=nan)&(d(1)~=nan)&(b(2)==nan))
        dd=infsup(extder4(sup(d(1))),extder4(inf(d(1))));
        bb=(extder4(b(1)));

```

```

nn=hull(dd,bb);

elseif((c(1)==nan)&(b(1)~=nan)&(d(1)~=nan)&(b(2)~=nan)&(c(2)==nan))
    bb=(extder4(b(1)));
    dd=infsup(extder4(sup(d(1))),extder4(inf(d(1))));
    bb1=extder4(b(2));
    nn=hull(bb,dd,bb1);

elseif ((c(1)==nan)&(b(1)~=nan)&(d(1)~=nan)&(b(2)~=nan)&(c(2)~=nan)&..
    (b(3)==nan))
    bb=(extder4(b(1)));
    dd=infsup(extder4(sup(d(1))),extder4(inf(d(1))));
    bb1=extder4(b(2));
    cc1=infsup(extder4(inf(c(2))),extder4(sup(c(2))));
    nn=hull(bb,dd,bb1,cc1);

elseif ((c(1)~=nan)&(b(1)~=nan)&(d(1)~=nan)&(b(2)~=nan)&(c(2)~=nan)&..
    (b(3)~=nan)&(d(2)==nan))
    bb=(extder4(b(1)));
    dd=infsup(extder4(sup(d(1))),extder4(inf(d(1))));
    bb1=extder4(b(2));
    cc1=infsup(extder4(inf(c(2))),extder4(sup(c(2))));
    bb2=extder4(b(3));
    nn=hull(bb,dd,bb1,cc1,bb2);

elseif((c(1)==nan)&(b(1)~=nan)&(d(1)~=nan)&(b(2)~=nan)&(c(2)~=nan)&..
    (b(3)~=nan)&(d(2)~=nan)&(b(4)==nan))
    bb=(extder4(b(1)));
    dd=infsup(extder4(sup(d(1))),extder4(inf(d(1))));
    bb1=extder4(b(2));
    cc1=infsup(extder4(inf(c(2))),extder4(sup(c(2))));
    bb2=extder4(b(3));
    dd1=infsup(extder4(sup(d(2))),extder4(inf(d(2))));
    nn=hull(bb,dd,bb1,cc1,bb2,dd1);

elseif((c(1)==nan)&(b(1)~=nan)&(d(1)~=nan)&(b(2)~=nan)&(c(2)~=nan)&..
    (b(3)~=nan)&(d(2)~=nan)&(b(4)~=nan)&(c(3)==nan))
    bb=(extder4(b(1)));
    dd=infsup(extder4(sup(d(1))),extder4(inf(d(1))));
    bb1=extder4(b(2));
    cc1=infsup(extder4(inf(c(2))),extder4(sup(c(2))));
    bb2=extder4(b(3));
    dd1=infsup(extder4(sup(d(2))),extder4(inf(d(2))));
    bb3=extder4(b(4));

```

```

nn=hull(bb,dd,bb1,cc1,bb2,dd1,bb3);

elseif ((c(1)==nan)&(b(1)~=nan)&(d(1)~=nan)&(b(2)~=nan)&(c(2)~=nan)&..
        (b(3)~=nan)&(d(2)~=nan)&(b(4)~=nan)&(c(3)~=nan)&(b(5)==nan))

        bb=(extder4(b(1)));
        dd=infsup(extder4(sup(d(1))),extder4(inf(d(1))));
        bb1=extder4(b(2));
        cc1=infsup(extder4(inf(c(2))),extder4(sup(c(2))));
        bb2=extder4(b(3));
        dd1=infsup(extder4(sup(d(2))),extder4(inf(d(2))));
        bb3=extder4(b(4));
        cc2=infsup(extder4(inf(c(3))),extder4(sup(c(3))));
        nn=hull(bb,dd,bb1,cc1,bb2,dd1,bb3,cc2);

elseif ((c(1)==nan)&(b(1)~=nan)&(d(1)~=nan)&(b(2)~=nan)&(c(2)~=nan)&..
        (b(3)~=nan)&(d(2)~=nan)&(b(4)~=nan)&(c(3)~=nan)&(b(5)~=nan)&(d(3)==nan))

        bb=(extder4(b(1)));
        dd=infsup(extder4(sup(d(1))),extder4(inf(d(1))));
        bb1=extder4(b(2));
        cc1=infsup(extder4(inf(c(2))),extder4(sup(c(2))));
        bb2=extder4(b(3));
        dd1=infsup(extder4(sup(d(2))),extder4(inf(d(2))));
        bb3=extder4(b(4));
        cc2=infsup(extder4(inf(c(3))),extder4(sup(c(3))));
        bb4=extder4(b(5));
        nn=hull(bb,dd,bb1,cc1,bb2,dd1,bb3,cc2,bb4);

elseif ((c(1)==nan)&(b(1)~=nan)&(d(1)~=nan)&(b(2)~=nan)&(c(2)~=nan)&..
        (b(3)~=nan)&(d(2)~=nan)&(b(4)~=nan)&(c(3)~=nan)&(b(5)~=nan)&(d(3)~=nan))

        bb=(extder4(b(1)));
        dd=infsup(extder4(sup(d(1))),extder4(inf(d(1))));
        bb1=extder4(b(2));
        cc1=infsup(extder4(inf(c(2))),extder4(sup(c(2))));
        bb2=extder4(b(3));
        dd1=infsup(extder4(sup(d(2))),extder4(inf(d(2))));
        bb3=extder4(b(4));
        cc2=infsup(extder4(inf(c(3))),extder4(sup(c(3))));
        bb4=extder4(b(5));
        dd2=infsup(extder4(sup(d(3))),extder4(inf(d(3))));
        nn=hull(bb,dd,bb1,cc1,bb2,dd1,bb3,cc2,bb4,dd2);

```

```

%terceira etapa

elseif ((c(1)==nan) & (b(1)==nan) & (d(1)~=nan) & (b(2)==nan))

    dd=infsup(extder4(sup(d(1))),extder4(inf(d(1))));
    nn=hull(dd);

elseif ((c(1)==nan) & (b(1)==nan) & (d(1)~=nan) & (b(2)~=nan) & (c(2)==nan))

    dd=infsup(extder4(sup(d(1))),extder4(inf(d(1))));
    bb1=extder4(b(2));
    nn=hull(dd,bb1);

elseif ((c(1)==nan) & (b(1)==nan) & (d(1)~=nan) & (b(2)~=nan) & (c(2)~=nan) & . .
        (b(3)==nan))

    dd=infsup(extder4(sup(d(1))),extder4(inf(d(1))));
    bb1=extder4(b(2));
    cc1=infsup(extder4(inf(c(2))),extder4(sup(c(2))));
    nn=hull(dd,bb1,cc1);

elseif ((c(1)~=nan) & (b(1)==nan) & (d(1)~=nan) & (b(2)~=nan) & (c(2)~=nan) & . .
        (b(3)~=nan) & (d(2)==nan))

    dd=infsup(extder4(sup(d(1))),extder4(inf(d(1))));
    bb1=extder4(b(2));
    cc1=infsup(extder4(inf(c(2))),extder4(sup(c(2))));
    bb2=extder4(b(3));
    nn=hull(bb,dd,bb1,cc1,bb2);

elseif ((c(1)==nan) & (b(1)==nan) & (d(1)~=nan) & (b(2)~=nan) & (c(2)~=nan) &
        (b(3)~=nan) & (d(2)~=nan) & (b(4)==nan))

    dd=infsup(extder4(sup(d(1))),extder4(inf(d(1))));
    bb1=extder4(b(2));
    cc1=infsup(extder4(inf(c(2))),extder4(sup(c(2))));
    bb2=extder4(b(3));
    dd1=infsup(extder4(sup(d(2))),extder4(inf(d(2))));
    nn=hull(dd,bb1,cc1,bb2,dd1);

elseif ((c(1)==nan) & (b(1)==nan) & (d(1)~=nan) & (b(2)~=nan) & (c(2)~=nan) & . .
        (b(3)~=nan) & (d(2)~=nan) & (b(4)~=nan) & (c(3)==nan))

    dd=infsup(extder4(sup(d(1))),extder4(inf(d(1))));

```

```

bb1=extder4(b(2));
cc1=infsup(extder4(inf(c(2))),extder4(sup(c(2))));
bb2=extder4(b(3));
dd1=infsup(extder4(sup(d(2))),extder4(inf(d(2))));
bb3=extder4(b(4));
nn=hull(dd,bb1,cc1,bb2,dd1,bb3);

elseif ((c(1)==nan)&(b(1)==nan)&(d(1)~=nan)&(b(2)~=nan)&(c(2)~=nan)&..
(b(3)~=nan)&(d(2)~=nan)&(b(4)~=nan)&(c(3)~=nan)&(b(5)==nan))

dd=infsup(extder4(sup(d(1))),extder4(inf(d(1))));
bb1=extder4(b(2));
cc1=infsup(extder4(inf(c(2))),extder4(sup(c(2))));
bb2=extder4(b(3));
dd1=infsup(extder4(sup(d(2))),extder4(inf(d(2))));
bb3=extder4(b(4));
cc2=infsup(extder4(inf(c(3))),extder4(sup(c(3))));
nn=hull(dd,bb1,cc1,bb2,dd1,bb3,cc2);

elseif ((c(1)==nan)&(b(1)==nan)&(d(1)~=nan)&(b(2)~=nan)&(c(2)~=nan)&
(b(3)~=nan)&(d(2)~=nan)&(b(4)~=nan)&(c(3)~=nan)&(b(5)~=nan)&(d(3)==nan))

dd=infsup(extder4(sup(d(1))),extder4(inf(d(1))));
bb1=extder4(b(2));
cc1=infsup(extder4(inf(c(2))),extder4(sup(c(2))));
bb2=extder4(b(3));
dd1=infsup(extder4(sup(d(2))),extder4(inf(d(2))));
bb3=extder4(b(4));
cc2=infsup(extder4(inf(c(3))),extder4(sup(c(3))));
bb4=extder4(b(5));
nn=hull(dd,bb1,cc1,bb2,dd1,bb3,cc2,bb4);

elseif ((c(1)==nan)&(b(1)==nan)&(d(1)~=nan)&(b(2)~=nan)&(c(2)~=nan)&..
(b(3)~=nan)&(d(2)~=nan)&(b(4)~=nan)&(c(3)~=nan)&(b(5)~=nan)&(d(3)~=nan))

dd=infsup(extder4(sup(d(1))),extder4(inf(d(1))));
bb1=extder4(b(2));
cc1=infsup(extder4(inf(c(2))),extder4(sup(c(2))));
bb2=extder4(b(3));
dd1=infsup(extder4(sup(d(2))),extder4(inf(d(2))));
bb3=extder4(b(4));
cc2=infsup(extder4(inf(c(3))),extder4(sup(c(3))));
bb4=extder4(b(5));
dd2=infsup(extder4(sup(d(3))),extder4(inf(d(3))));

```

```

nn=hull(dd,bb1,cc1,bb2,dd1,bb3,cc2,bb4,dd2);

%quarta etapa

elseif ((c(1)==nan) & (b(1)==nan) & (d(1)==nan) & (b(2)~=nan) & (c(2)==nan))
    bb1=extder5(x);
    nn=hull(dd);

elseif ((c(1)==nan) & (b(1)==nan) & (d(1)==nan) & (b(2)~=nan) & (c(2)~=nan) & . .
    (b(3)==nan))

    bb1=extder4(b(2));
    cc1=infsup(extder4(inf(c(2))),extder4(sup(c(2))));
    nn=hull(bb1,cc1);

elseif ((c(1)~=nan) & (b(1)==nan) & (d(1)==nan) & (b(2)~=nan) & (c(2)~=nan) & . .
    (b(3)~=nan) & (d(2)==nan))

    bb1=extder4(b(2));
    cc1=infsup(extder4(inf(c(2))),extder4(sup(c(2))));
    bb2=extder4(b(3));
    nn=hull(dd,bb1,cc1,bb2);

elseif ((c(1)==nan) & (b(1)==nan) & (d(1)==nan) & (b(2)~=nan) & (c(2)~=nan) &
    (b(3)~=nan) & (d(2)~=nan) & (b(4)==nan))

    bb1=extder4(b(2));
    cc1=infsup(extder4(inf(c(2))),extder4(sup(c(2))));
    bb2=extder4(b(3));
    dd1=infsup(extder4(sup(d(2))),extder4(inf(d(2))));
    nn=hull(bb1,cc1,bb2,dd1);

elseif ((c(1)==nan) & (b(1)==nan) & (d(1)==nan) & (b(2)~=nan) & (c(2)~=nan) & . .
    (b(3)~=nan) & (d(2)~=nan) & (b(4)~=nan) & (c(3)==nan))

    bb1=extder4(b(2));
    cc1=infsup(extder4(inf(c(2))),extder4(sup(c(2))));
    bb2=extder4(b(3));
    dd1=infsup(extder4(sup(d(2))),extder4(inf(d(2))));
    bb3=extder4(b(4));
    nn=hull(bb1,cc1,bb2,dd1,bb3);

elseif ((c(1)==nan) & (b(1)==nan) & (d(1)~=nan) & (b(2)~=nan) & (c(2)~=nan) & . .
    (b(3)~=nan) & (d(2)~=nan) & (b(4)~=nan) & (c(3)~=nan) & (b(5)==nan))

```

```

bb1=extder4(b(2));
cc1=infsup(extder4(inf(c(2))),extder4(sup(c(2))));
bb2=extder4(b(3));
dd1=infsup(extder4(sup(d(2))),extder4(inf(d(2))));
bb3=extder4(b(4));
cc2=infsup(extder4(inf(c(3))),extder4(sup(c(3))));
nn=hull(bb1,cc1,bb2,dd1,bb3,cc2);

elseif ((c(1)==nan)&(b(1)==nan)&(d(1)==nan)&(b(2)~=nan)&(c(2)~=nan)&..
(b(3)~=nan)&(d(2)~=nan)&(b(4)~=nan)&(c(3)~=nan)&(b(5)~=nan)&(d(3)==nan))

bb1=extder4(b(2));
cc1=infsup(extder4(inf(c(2))),extder4(sup(c(2))));
bb2=extder4(b(3));
dd1=infsup(extder4(sup(d(2))),extder4(inf(d(2))));
bb3=extder4(b(4));
cc2=infsup(extder4(inf(c(3))),extder4(sup(c(3))));
bb4=extder4(b(5));
nn=hull(bb1,cc1,bb2,dd1,bb3,cc2,bb4) ;

elseif ((c(1)==nan)&(b(1)==nan)&(d(1)==nan)&(b(2)~=nan)&(c(2)~=nan)&..
(b(3)~=nan)&(d(2)~=nan)&(b(4)~=nan)&(c(3)~=nan)&(b(5)~=nan)&(d(3)~=nan))

bb1=extder4(b(2));
cc1=infsup(extder4(inf(c(2))),extder4(sup(c(2))));
bb2=extder4(b(3));
dd1=infsup(extder4(sup(d(2))),extder4(inf(d(2))));
bb3=extder4(b(4));
cc2=infsup(extder4(inf(c(3))),extder4(sup(c(3))));
bb4=extder4(b(5));
dd2=infsup(extder4(sup(d(3))),extder4(inf(d(3))));
nn=hull(bb1,cc1,bb2,dd1,bb3,cc2,bb4,dd2);

%sexta etapa

elseif ((c(1)~=nan)&(b(1)==nan)&(d(1)==nan)&(b(2)==nan)&(c(2)~=nan)&..
(b(3)==nan))
cc1=infsup(extder4(inf(c(2))),extder4(sup(c(2))));
nn=hull(cc1);

elseif ((c(1)~=nan)&(b(1)==nan)&(d(1)==nan)&(b(2)==nan)&(c(2)~=nan)&..
(b(3)~=nan)&(d(2)==nan))

```

```

cc1=infsup(extder4 (inf (c (2))) , extder4 (sup (c (2)))) ;
bb2=extder4 (b (3)) ;
nn=hull (cc1,bb2) ;

elseif ((c (1)==nan) & (b (1)==nan) & (d (1)==nan) & (b (2)==nan) & (c (2) ~=nan) & . .
        (b (3) ~=nan) & (d (2) ~=nan) & (b (4) ==nan) )

cc1=infsup(extder4 (inf (c (2))) , extder4 (sup (c (2)))) ;
bb2=extder4 (b (3)) ;
ddl=infsup(extder4 (sup (d (2))) , extder4 (inf (d (2)))) ;
nn=hull (cc1,bb2,ddl) ;

elseif ((c (1)==nan) & (b (1)==nan) & (d (1)==nan) & (b (2)==nan) & (c (2) ~=nan) & . .
        (b (3) ~=nan) & (d (2) ~=nan) & (b (4) ~=nan) & (c (3) ==nan) )

cc1=infsup(extder4 (inf (c (2))) , extder4 (sup (c (2)))) ;
bb2=extder4 (b (3)) ;
ddl=infsup(extder4 (sup (d (2))) , extder4 (inf (d (2)))) ;
bb3=extder4 (b (4)) ;
nn=hull (cc1,bb2,ddl,bb3) ;

elseif ((c (1)==nan) & (b (1)==nan) & (d (1)==nan) & (b (2)==nan) & (c (2) ~=nan) & . .
        (b (3) ~=nan) & (d (2) ~=nan) & (b (4) ~=nan) & (c (3) ~=nan) & (b (5) ==nan) )

cc1=infsup(extder4 (inf (c (2))) , extder4 (sup (c (2)))) ;
bb2=extder4 (b (3)) ;
ddl=infsup(extder4 (sup (d (2))) , extder4 (inf (d (2)))) ;
bb3=extder4 (b (4)) ;
cc2=infsup(extder4 (inf (c (3))) , extder4 (sup (c (3)))) ;
nn=hull (cc1,bb2,ddl,bb3,cc2) ;

elseif ((c (1)==nan) & (b (1)==nan) & (d (1)==nan) & (b (2)==nan) & (c (2) ~=nan) & . .
        (b (3) ~=nan) & (d (2) ~=nan) & (b (4) ~=nan) & (c (3) ~=nan) & (b (5) ~=nan) & (d (3) ==nan) )

cc1=infsup(extder4 (inf (c (2))) , extder4 (sup (c (2)))) ;
bb2=extder4 (b (3)) ;
ddl=infsup(extder4 (sup (d (2))) , extder4 (inf (d (2)))) ;
bb3=extder4 (b (4)) ;
cc2=infsup(extder4 (inf (c (3))) , extder4 (sup (c (3)))) ;
bb4=extder4 (b (5)) ;
nn=hull (cc1,bb2,ddl,bb3,cc2,bb4) ;

elseif ((c (1)==nan) & (b (1)==nan) & (d (1)==nan) & (b (2)==nan) & (c (2) ~=nan) & . .
        (b (3) ~=nan) & (d (2) ~=nan) & (b (4) ~=nan) & (c (3) ~=nan) & (b (5) ~=nan) & (d (3) ~=nan) )

```

```

cc1=infsup(extder4 (inf (c (2))) , extder4 (sup (c (2)))) ;
bb2=extder4 (b (3)) ;
dd1=infsup(extder4 (sup (d (2))) , extder4 (inf (d (2)))) ;
bb3=extder4 (b (4)) ;
cc2=infsup(extder4 (inf (c (3))) , extder4 (sup (c (3)))) ;
bb4=extder4 (b (5)) ;
dd2=infsup(extder4 (sup (d (3))) , extder4 (inf (d (3)))) ;
nn=hull (cc1, bb2, dd1, bb3, cc2, bb4, dd2) ;

%setima etapa

elseif ((c (1)==nan) & (b (1)==nan) & (d (1)==nan) & (b (2)==nan) & (c (2)==nan) & . .
    (b (3)~=nan) & (d (2)==nan) )
    bb1=extder5 (x) ;
    nn=hull (bb1) ;

elseif ((c (1)==nan) & (b (1)==nan) & (d (1)==nan) & (b (2)==nan) & (c (2)==nan) & . .
    (b (3)~=nan) & (d (2)~=nan) & (b (4)==nan) )

    bb2=extder4 (b (3))
    dd1=infsup(extder4 (sup (d (2))) , extder4 (inf (d (2)))) ;
    nn=hull (bb2, dd1) ;

elseif ((c (1)==nan) & (b (1)==nan) & (d (1)==nan) & (b (2)==nan) & (c (2)==nan) & . .
    (b (3)~=nan) & (d (2)~=nan) & (b (4)~=nan) & (c (3)==nan) )

    bb2=extder4 (b (3)) ;
    dd1=infsup(extder4 (sup (d (2))) , extder4 (inf (d (2)))) ;
    bb3=extder4 (b (4)) ;
    nn=hull (bb2, dd1, bb3) ;

elseif ((c (1)==nan) & (b (1)==nan) & (d (1)==nan) & (b (2)==nan) & (c (2)==nan) & . .
    (b (3)~=nan) & (d (2)~=nan) & (b (4)~=nan) & (c (3)~=nan) & (b (5)==nan) )

    bb2=extder4 (b (3)) ;
    dd1=infsup(extder4 (sup (d (2))) , extder4 (inf (d (2)))) ;
    bb3=extder4 (b (4)) ;
    cc2=infsup(extder4 (inf (c (3))) , extder4 (sup (c (3)))) ;
    nn=hull (bb2, dd1, bb3, cc2) ;

elseif ((c (1)==nan) & (b (1)==nan) & (d (1)==nan) & (b (2)==nan) & (c (2)==nan) & . .
    (b (3)~=nan) & (d (2)~=nan) & (b (4)~=nan) & (c (3)~=nan) & (b (5)~=nan) & . .
    (d (3)==nan) )

```

```

bb2=extder4(b(3));
dd1=infsup(extder4(sup(d(2))),extder4(inf(d(2))));
bb3=extder4(b(4));
cc2=infsup(extder4(inf(c(3))),extder4(sup(c(3))));
bb4=extder4(b(5));
nn=hull(bb2,dd1,bb3,cc2,bb4);

elseif ((c(1)==nan)&(b(1)==nan)&(d(1)==nan)&(b(2)==nan)&(c(2)==nan)&..
        (b(3)~=nan)&(d(2)~=nan)&(b(4)~=nan)&(c(3)~=nan)&(b(5)~=nan)&..
        (d(3)~=nan))

bb2=extder4(b(3))
dd1=infsup(extder4(sup(d(2))),extder4(inf(d(2))));
bb3=extder4(b(4));
cc2=infsup(extder4(inf(c(3))),extder4(sup(c(3))));
bb4=extder4(b(5));
dd2=infsup(extder4(sup(d(3))),extder4(inf(d(3))));
nn=hull(bb2,dd1,bb3,cc2,bb4,dd2) ;

%oitava etapa

elseif ((c(1)==nan)&(b(1)==nan)&(d(1)==nan)&(b(2)==nan)&(c(2)==nan)&..
        (b(3)==nan)&(d(2)~=nan)&(b(4)==nan))
        dd1=infsup(extder4(sup(d(2))),extder4(inf(d(2))));
        nn=hull(dd1);

elseif((c(1)==nan)&(b(1)==nan)&(d(1)==nan)&(b(2)==nan)&(c(2)==nan)&..
        (b(3)==nan)&(d(2)~=nan)&(b(4)~=nan)&(c(3)==nan))

        dd1=infsup(extder4(sup(d(2))),extder4(inf(d(2))));
        bb3=extder4(b(4));
        nn=hull(dd1,bb3);

elseif ((c(1)==nan)&(b(1)==nan)&(d(1)==nan)&(b(2)==nan)&(c(2)==nan)&..
        (b(3)==nan)&(d(2)~=nan)&(b(4)~=nan)&(c(3)~=nan)&(b(5)==nan))

        dd1=infsup(extder4(sup(d(2))),extder4(inf(d(2))));
        bb3=extder4(b(4));
        cc2=infsup(extder4(inf(c(3))),extder4(sup(c(3))));
        nn=hull(dd1,bb3,cc2);

elseif ((c(1)==nan)&(b(1)==nan)&(d(1)==nan)&(b(2)==nan)&(c(2)==nan)&..
        (b(3)==nan)&(d(2)~=nan)&(b(4)~=nan)&(c(3)~=nan)&(b(5)~=nan)&..

```

```

(d(3)==nan))

dd1=infsup(extder4(sup(d(2))),extder4(inf(d(2))));
bb3=extder4(b(4));
cc2=infsup(extder4(inf(c(3))),extder4(sup(c(3))));
bb4=extder4(b(5));
nn=hull(dd1,bb3,cc2,bb4);

elseif ((c(1)==nan)&(b(1)==nan)&(d(1)==nan)&(b(2)==nan)&(c(2)==nan)&..
        (b(3)==nan)&(d(2)~=nan)&(b(4)~=nan)&(c(3)~=nan)&(b(5)~=nan)&..
        (d(3)~=nan))

dd1=infsup(extder4(sup(d(2))),extder4(inf(d(2))));
bb3=extder4(b(4));
cc2=infsup(extder4(inf(c(3))),extder4(sup(c(3))));
bb4=extder4(b(5));
dd2=infsup(extder4(sup(d(3))),extder4(inf(d(3))));
nn=hull(dd1,bb3,cc2,bb4,dd2);

%nona etapa

elseif ((c(1)==nan)&(b(1)==nan)&(d(1)==nan)&(b(2)==nan)&(c(2)==nan)&..
        (b(3)==nan)&(d(2)==nan)&(b(4)~=nan)&(c(3)==nan))

bb3=extder5(x);
nn=hull(bb3);

elseif ((c(1)==nan)&(b(1)==nan)&(d(1)==nan)&(b(2)==nan)&(c(2)==nan)&..
        (b(3)==nan)&(d(2)==nan)&(b(4)~=nan)&(c(3)~=nan)&(b(5)==nan))

bb3=extder4(b(4));
cc2=infsup(extder4(inf(c(3))),extder4(sup(c(3))));
nn=hull(bb3,cc2);

elseif ((c(1)==nan)&(b(1)==nan)&(d(1)==nan)&(b(2)==nan)&(c(2)==nan)&..
        (b(3)==nan)&(d(2)==nan)&(b(4)~=nan)&(c(3)~=nan)&(b(5)~=nan)&..
        (d(3)==nan))

bb3=extder4(b(4));
cc2=infsup(extder4(inf(c(3))),extder4(sup(c(3))));
bb4=extder4(b(5));
nn=hull(bb3,cc2,bb4);

elseif ((c(1)==nan)&(b(1)==nan)&(d(1)==nan)&(b(2)==nan)&(c(2)==nan)&..

```

```

(b(3)==nan) & (d(2)==nan) & (b(4)~=nan) & (c(3)~=nan) & (b(5)~=nan) & . .
(d(3)~=nan))

bb3=extderb(b(4));
cc2=infsup(extder4(inf(c(3))),extder4(sup(c(3))));
bb4=extder4(b(5));
dd2=infsup(extder4(sup(d(3))),extder4(inf(d(3))));
nn=hull(bb3,cc2,bb4,dd2);

%dez etapa

elseif ((c(1)==nan) & (b(1)==nan) & (d(1)==nan) & (b(2)==nan) & (c(2)==nan) & . .
(b(3)==nan) & (d(2)==nan) & (b(4)==nan) & (c(3)~=nan) & (b(5)==nan))

cc2=infsup(extder4(inf(c(3))),extder4(sup(c(3))));
nn=hull(cc2);

elseif ((c(1)==nan) & (b(1)==nan) & (d(1)==nan) & (b(2)==nan) & (c(2)==nan) & . .
(b(3)==nan) & (d(2)==nan) & (b(4)==nan) & (c(3)~=nan) & (b(5)~=nan) & . .
(d(3)==nan))

cc2=infsup(extder4(inf(c(3))),extder4(sup(c(3))));
bb4=extder4(b(5));
nn=hull(cc2,bb4);

elseif ((c(1)==nan) & (b(1)==nan) & (d(1)==nan) & (b(2)==nan) & (c(2)==nan) & . .
(b(3)==nan) & (d(2)==nan) & (b(4)==nan) & (c(3)~=nan) & (b(5)~=nan) & . .
(d(3)~=nan))

cc2=infsup(extder4(inf(c(3))),extder4(sup(c(3))));
bb4=extder4(b(5));
dd2=infsup(extder4(sup(d(3))),extder4(inf(d(3))));
nn=hull(cc2,bb4,dd2);

%onze etapa

elseif ((c(1)==nan) & (b(1)==nan) & (d(1)==nan) & (b(2)==nan) & (c(2)==nan) & . .
(b(3)==nan) & (d(2)==nan) & (b(4)==nan) & (c(3)==nan) & (b(5)~=nan) & . .
(d(3)==nan))

bb4=extder5(x);
nn=hull(bb4);

elseif ((c(1)==nan) & (b(1)==nan) & (d(1)==nan) & (b(2)==nan) & (c(2)==nan) & . .

```

```

(b(3)==nan) & (d(2)==nan) & (b(4)==nan) & (c(3)==nan) & (b(5)~=nan) & . .
(d(3)~=nan))

bb4=extder4(b(5));
dd2=infsup(extder4(sup(d(3))),extder4(inf(d(3))));
nn=hull(bb4,dd2);

%doze etapa

else
    dd2=infsup(extder4(sup(d(3))),extder4(inf(d(3))));
    nn=dd2

end
%intvalinit('DisplayInfSup')

qq=infsup(nn);
vv=(derivada(y1,y2));
vv1=infsup(derivada(y1,y2));

```

B.2.5 Implementação do intervalo encapsulador para a Normal Padrão

A função *normal1* determina o intervalo encapsulador para uma variável aleatória Normal Padrão e necessita de três argumentos de entrada: os limites de integração e o número de subintervalos da partição.

```

%Desenvolvimento do calculo intervalar por aproximacao
%Calculo do intervalo encapsulador para a variavel normal padrao
%usando o metodo de simpson intervalar
% a=limite inferior do intervalo de integracao
% b= limite superior do intervalo de integracao
% p=numero de subintervalos de [a,b]
% os arquivos compox, rsimpson sao chamados pela funcao normal1
function [S]=normal1(a,b,p)
tic
syms z
yy=exp(-z^2/2);%funcao
y=diff(exp(-z^2/2),4);%derivada de ordem 4 da funcao yy
%e=infsup(a,b)
if((a== -inf) & (b<0))

```

```

h=-b/p; % comprimento de cada subintervalo
n=1:p+1;
x(n)=b + h*(n-1); %vetor com os extremos dos subintervalos
s=0;
for i=1:p
    t(i)=infsup(x(i),x(i+1)); %criando os subintervalos
    m(i)=mid(t(i));% ponto medio do intervalo i
    f(i)=subs(yy,z,x(i)); %calculando o valor de yy em x(i)
    f(i+1)=subs(yy,z,x(i+1));%calculando o valor de yy em x(i+1)
    fm(i)=subs(yy,z,m(i)); %calculando o valor de yy em m(i)
    gd(i)=compo(x(i),x(i+1)); %calcula uma extensao intervalar da derivada
    % de ordem 4
    %intvalinit('DisplayInfSup');
    %calculando a formula de simpson
    % (h^2/6)*(f(x(i))+4*f(m(i)) + f(x(i+1)))-(h^5/2880)*extensao intervalar
    %da derivada de ordem 4
    s=s + rsimpson(f(i),f(i+1),fm(i),h,gd(i));
end
%intvalinit('DisplayInfSup');
%calculo da integral de (1/(sqrt(2*pi))*exp(-x^2/2) distribuicao
%normal de media 0 e variancia 1
S=0.5 - (1/(sqrt(2*pi)))*s;
rr=rad(S);
    elseif((a== -inf)&(b==0))
        S=0.5

        elseif((a== -inf) & (b>0) & (b~=inf))

h=b/p; % comprimento de cada subintervalo
n=1:p+1;
x(n)= h*(n-1); %vetor com os extremos dos subintervalos
s=0;
    for i=1:p
        t(i)=infsup(x(i),x(i+1)); %criando os subintervalos
        m(i)=mid(t(i));% ponto medio do intervalo i
        f(i)=subs(yy,z,x(i)); %calculando o valor de yy em x(i)
        f(i+1)=subs(yy,z,x(i+1));%calculando o valor de yy em x(i+1)
        fm(i)=subs(yy,z,m(i)); %calculando o valor de yy em m(i)
        gd(i)=compo(x(i),x(i+1)); %calcula uma extensao intervalar da derivada
        %de ordem 4
        %intvalinit('DisplayInfSup');
        %calculando a formula de simpson
        % (h^2/6)*(f(x(i))+4*f(m(i)) + f(x(i+1)))-(h^5/2880)*extensao
        %intervalar da derivada de ordem 4

```

```

s=s + rsimpson1(f(i),f(i+1),fm(i),h,gd(i));
end
%intvalinit('DisplayInfSup')
%calculo da integral de (1/(sqrt(2*pi))*exp(-x^2/2)
%distribuicao normal de media 0 e variancia 1
S=0.5 + (1/(sqrt(2*pi)))*s;
rr=rad(S);
    else((a~=-inf)&(b ~= inf))
h=(b-a)/p; % comprimento de cada subintervalo
n=1:p+1;
x(n)=a + h*(n-1); %vetor com os extremos dos subintervalos
s=0;
    for i=1:p
t(i)=infsup(x(i),x(i+1)); %criando os subintervalos
m(i)=mid(t(i));% ponto medio do intervalo i
f(i)=subs(yy,z,x(i)); %calculando o valor de yy em x(i)
f(i+1)=subs(yy,z,x(i+1));%calculando o valor de yy em x(i+1)
fm(i)=subs(yy,z,m(i)); %calculando o valor de yy em m(i)
gd(i)=compo(x(i),x(i+1)); %calcula uma extensao intervalar da derivada
%de ordem 4
%intvalinit('DisplayInfSup');
%calculando a formula de simpson
%(h^2/6)*(f(x(i))+4*f(m(i)) + f(x(i+1)))-(h^5/2880)*extensao
%intervalar da derivada de ordem 4
s=s + rsimpson1(f(i),f(i+1),fm(i),h,gd(i));
end
%intvalinit('DisplayInfSup');
%calculo da integral de (1/(sqrt(2*pi))*exp(-x^2/2) distribuicao normal
%de media 0 e variancia 1
S=(1/(sqrt(2*pi)))*s;
rr=rad(S);

end
In=S;
RR=rad(In);
toc

```

B.2.6 Implementação do intervalo encapsulador para a densidade Uniforme

A função *dunif* determina um intervalo encapsulador para uma variável aleatória Uniforme definida no intervalo $[a, b]$ e necessita de quatro argumentos de entrada: a,b e os limites de integração.

```
%Calculo do intervalo encapsulador da distribuicao uniforme
% a=limite inferior do intervalo de integracao
% b= limite superior do intervalo de integracao
% X=intervalo de variacao [c,d]
function [S]=dunif(a,b,c,d)
tic
    if intersect(infsup(a,b),infsup(c,d))==[]
        s=0;
    else
        y= diam(intersect(infsup(a,b),infsup(c,d)))
        setround (-1)
        s1= y/diam(infsup(a,b));

        setround(1)
        s2= y/diam(infsup(a,b));
        % setround(0)
        s=intval(y)/diam(infsup(a,b))
        s3=infsup(s1,s2)
        f=infsup(s3), h=infsup(s)

    end
    intvalinit('DisplayInfSup');

    %S=s %intervalo encapsulador da Uniforme

    %r=diam(s)
    toc
```