



MARCELO CORRÊA MUSSEL

**IMPLEMENTAÇÃO COMPUTACIONAL E
APLICAÇÕES EM DINÂMICA
POPULACIONAL DO PRINCÍPIO DE
EXTENSÃO DE ZADEH**

LAVRAS – MG

2013

MARCELO CORRÊA MUSSEL

**IMPLEMENTAÇÃO COMPUTACIONAL E APLICAÇÕES EM
DINÂMICA POPULACIONAL DO PRINCÍPIO DE EXTENSÃO DE
ZADEH**

Dissertação apresentada à Universidade Federal de Lavras como parte das exigências do Programa de Pós-Graduação em Engenharia de Sistemas, área de concentração em Modelagem de Sistemas Biológicos, para obtenção do título de Mestre.

Orientador
Dr. Onofre Rojas Santos

LAVRAS – MG

2013

**Ficha Catalográfica Elaborada pela Coordenadoria de Produtos e
Serviços da Biblioteca Universitária da UFLA**

Mussel, Marcelo Corrêa.

Implementação computacional e aplicações em dinâmica populacional do princípio de extensão de Zadeh / Marcelo Corrêa Mussel. – Lavras : UFLA, 2013.

80 p. : il.

Dissertação (mestrado) – Universidade Federal de Lavras, 2013.

Orientador: Onofre Rojas.

Bibliografia.

1. Lógica *fuzzy*. 2. Equação de Verhulst. 3. Princípio de extensão. 4. Dinâmica populacional. 5. Inversão de funções. I. Universidade Federal de Lavras. II. Título.

CDD – 620.00113

MARCELO CORRÊA MUSSEL

**IMPLEMENTAÇÃO COMPUTACIONAL E APLICAÇÕES EM
DINÂMICA POPULACIONAL DO PRINCÍPIO DE EXTENSÃO DE
ZADEH**

Dissertação apresentada à Universidade Federal de Lavras como parte das exigências do Programa de Pós-Graduação em Engenharia de Sistemas, área de concentração em Modelagem de Sistemas Biológicos, para obtenção do título de Mestre.

APROVADA em 18 de outubro de 2013.

Dr. Onofre Rojas Santos UFLA

Dr. Danton Diego Ferreira UFLA

Dr. Marcelo Ângelo Cirillo UFLA

Dra. Thelma Sáfyadi UFLA

Dr. Onofre Rojas Santos

Orientador

LAVRAS – MG

2013

Aos meus pais, Ernani (*in memoriam*) e Minervina, a minha irmã, Vera Lúcia, a
minha esposa, Valéria.

DEDICO.

AGRADECIMENTOS

A Deus por todas as bênçãos concedidas à mim e a minha família.

Aos meus pais, Ernani Alberto Mussel e Minervina Corrêa Mussel por todo apoio, dedicação, carinho, educação e incentivo que sempre dedicaram.

À minha irmã Vera Lucia C. M. T. do Prado, pela amizade, confiança e torcida.

À minha esposa Valéria Mayworm Woll pelo carinho e apoio incondicional, estando sempre ao meu lado.

Ao meu orientador, Onofre Rojas Santos pela ajuda e ensinamentos.

À banca examinadora, que aceitou o convite que lhes foi feito.

Aos professores da UFLA: Danton Diego Ferreira, Marcelo Ângelo Cirillo, Sérgio Martins de Souza, Tadayuki Yanagi Junior e Wilson Magela Gonçalves, que lecionaram as disciplinas do programa pelos ensinamentos e conselhos.

Aos colegas do mestrado pelo apoio, companheirismo e amizade.

À Universidade Federal de Lavras (UFLA), em especial ao Programa de Pós-Graduação em Engenharia de Sistemas pela oportunidade de realização do curso.

Aos familiares e amigos, mesmo que distantes fisicamente, sempre se mostraram presentes e torcendo pelo meu sucesso.

Enfim, a todos aqueles que de alguma forma contribuíram para que atingisse meu objetivo.

Obrigado a todos!

“Uma vez que não podemos ser universais e saber tudo quanto se pode saber acerca de tudo, é preciso saber-se um pouco de tudo, pois é muito melhor saber-se alguma coisa de tudo do que saber-se tudo apenas de uma coisa.”

Blaise Pascal

RESUMO

Neste trabalho, será apresentada uma metodologia computacional que permite aplicar o princípio de extensão à equação logística, mais precisamente à equação de Verhulst. Esta metodologia é utilizada para permitir o acréscimo de incerteza (parâmetros fuzzy) a modelos matemáticos estudados pela teoria clássica. Os modelos de crescimento populacional apresentam graus de subjetividade quando precisamos estimar valores para a taxa de crescimento e/ou para capacidade suporte. Para aplicação do princípio, foi utilizada a técnica de inversão de funções, sendo que a resolução é realizada de forma analítica, para validar esta metodologia foram utilizados dados do município de Sorocaba-SP e confrontados com os resultados de ROVEDA, (2009), realizada uma projeção da população de

Lavras-MG, a partir dos dados obtidos do IBGE e um estudo comparativo sobre o crescimento de bovinos no Brasil confrontando os dados com FREITAS, et al. (2005).

Palavras-chave: Lógica *fuzzy*. Princípio de extensão. Inversão de funções. Equação de Verhulst. Dinâmica populacional.

ABSTRACT

In this work, we present a computational methodology that allows applying the extension principle to the logistic equation, more precisely the Verhulst equation. This methodology is used to allow the inclusion of uncertainty (parameters *fuzzy*) to mathematical models studied by classical theory. The population growth models depend on degree subjectivity when we need to estimate values for the rate of growth and / or carrying capacity. To apply the extension principle, we have used the technique of inverse function, and the solution is performed analytically. To apply this methodology, we used the data of Sorocaba city, and compared with the results of ROVEDA, (2009), the population projection of Lavras city, with data obtained from IBGE and a comparative study on the bovine growth in Brazil comparing with FREITAS, et al. (2005) data.

Keywords: *Fuzzy* Logic. Extension principle. Inversion function. Verhulst equation. Population dynamics.

LISTA DE GRÁFICOS

Gráfico 1	Representação da transição instantânea de temperatura	19
Gráfico 2	Representação da transição gradual de temperatura	19
Gráfico 3	Transição de temperatura com conjuntos <i>fuzzy</i>	20
Gráfico 4	Figura da imagem de um subconjunto <i>fuzzy</i> a partir do	24
Gráfico 5	Função de Pertinência triangular para o núm. "próximo" de 4	25
Gráfico 6	Função de Pertinência triangular para o núm. próximo" de 3	26
Gráfico 7	α -níveis $[A]^\alpha$ e $[A]^0 \neq U$	28
Gráfico 8	Função de pertinência triangular.	30
Gráfico 9	Função de pertinência trapezoidal.	30
Gráfico 10	Função de pertinência gaussiana.	31
Gráfico 11	função $N(t)$ com k "em torno de 100"	42
Gráfico 12	pertinência da variável k "em torno de 100"	44
Gráfico 13	pertinência de $f(k)$ em $t = 10$	44
Gráfico 14	função $N(t)$ com k "em torno de 100"	45
Gráfico 15	Função $N(t)$ com valores <i>defuzzificados</i>	45
Gráfico 16	pertinência da variável r "em torno de" 0,3.	47
Gráfico 17	pertinência de $f(r)$	47
Gráfico 18	Função $N(t)$ com r "em torno de" 0,3.	48
Gráfico 19	Função $N(t)$ com valores defuzzificados.	49
Gráfico 20	Variável população. ROVEDA, et al. (2009)	52
Gráfico 21	Δ_{ab} população. ROVEDA, et al. (2009)	52
Gráfico 22	Δ_{rel} população. ROVEDA, et al. (2009)	53
Gráfico 23	pertinência no instante $t = 10$ (1970)	56
Gráfico 24	pertinência no instante $t = 10$ (1970)	56
Gráfico 25	Conjunto <i>fuzzy</i> gerado a partir da variável r	56
Gráfico 26	Comparação com ROVEDA, et al. (2009)	57
Gráfico 27	pertinência no instante $t = 30$ (2021)	60
Gráfico 28	pertinência no instante $t = 30$ (2021)	60
Gráfico 29	Conjunto <i>fuzzy</i> gerado a partir da variável r	61
Gráfico 30	Dados <i>defuzzificados</i> comparados aos do IBGE/Censo	61
Gráfico 31	pertinência da taxa de crescimento (r) no instante $t = 10$	64
Gráfico 32	pertinência da população estimada (N) no instante $t = 10$	64
Gráfico 33	Conjunto <i>fuzzy</i> gerado a partir da variável r	65
Gráfico 34	Comparação com FREITAS, et al. (2005)	66

LISTA DE TABELAS

Tabela 1	Ilustração das operações entre subconjuntos <i>fuzzy</i>	22
Tabela 2	Dados obtidos em $t = 1$ e k “em torno de 100”	41
Tabela 3	Dados extraídos de ROVEDA, et al. (2009)	54
Tabela 4	Taxa de crescimento obtida de acordo com censo	55
Tabela 5	Comparativo entre os valores apresentados em ROVEDA, et al. (2009) e os calculados pelo Princípio de Extensão	58
Tabela 6	Evolução populacional do município de Lavras-MG (Fonte: IBGE).....	59
Tabela 7	Evolução populacional do município de Lavras-MG (IBGE x Princípio de Extensão)	62
Tabela 8	Evolução populacional do rebanho bovino brasileiro, valores em milhões	67

SUMÁRIO

1	INTRODUÇÃO	12
2	REFERENCIAL TEÓRICO	16
2.1	A equação de Verhulst	16
2.2	Lógica <i>Fuzzy</i>	16
2.2.1	Conjuntos <i>Fuzzy</i>	18
2.2.2	Operações entre Conjuntos <i>Fuzzy</i>	20
2.2.3	Princípio de Extensão	22
2.2.4	Conceito de α -nível	27
2.2.5	Números <i>Fuzzy</i>	29
2.2.6	Operações aritméticas com números <i>fuzzy</i>	32
2.2.7	Controlador <i>fuzzy</i>	34
2.2.8	Defuzzificação	36
3	METODOLOGIA	38
3.1	Recursos computacionais	38
3.2	Princípio de Extensão de Zadeh pelo método clássico	39
3.3	Aplicação do Princípio de Extensão de Zadeh através da inversão de funções – Parâmetro k	42
3.4	Aplicação do Princípio de Extensão de Zadeh através da inversão de funções – Parâmetro r	46
4	RESULTADOS E DISCUSSÕES	50
4.1	Estudo sobre a dinâmica populacional da cidade de Sorocaba- SP	50
4.2	Estudo sobre a dinâmica populacional da cidade de Lavras-MG	59
4.3	Modelagem do crescimento populacional do rebanho bovino brasileiro	63
5	CONSIDERAÇÕES FINAIS	68
6	CONCLUSÃO	69
	REFERÊNCIAS	70
	APÊNDICE A – Código da classe Verhulst	73
	APÊNDICE B – Código da classe JFTela	76
	APÊNDICE C – Imagem da interface gráfica	80

1 INTRODUÇÃO

A dinâmica de populações visa a melhor compreensão da variação do número de indivíduos, no tempo e no espaço, tratando das densidades e tamanhos e, também, dos fatores que a influenciam em tais variações. Para isso, é necessário o conhecimento das taxas em que se verificam perdas e ganhos de indivíduos e identificar os processos que regulam a variação da população.

O estudo matemático de dinâmica de populações surgiu em 1798, quando foi publicado o artigo “*An Essay on the Principle of Population as it Affects the Future Improvement of Society*” do economista e demógrafo britânico Thomas Robert Malthus, no qual afirma que a população cresce em progressão geométrica, enquanto a produção de alimentos aumenta em progressão aritmética.

Em 1838, a limitação dos recursos foi estudada por Pierre Verhulst, a pedido do governo da Bélgica que estava preocupado com o crescimento populacional. Verhulst incorporou essa limitação ao modelo de Malthus e apresentou a equação do crescimento populacional, também conhecida como equação logística (BRANNAN et al., 2008).

Em se tratando do crescimento populacional de uma dada espécie, através do modelo de Malthus, num dado intervalo de tempo, podemos verificar um crescimento exponencial, se a considerarmos sob condições ideais, sendo estas não persistentes por longos intervalos de tempo. Fatores como limitações de espaço, comida, recursos naturais, tratamento médico, condições sanitárias irão influenciar diretamente no crescimento desta população (MIZUKOSHI et al. 2007).

Inicialmente, a dinâmica de populações utiliza modelos matemáticos na forma de equações diferenciais ordinárias (EDOs) ou parciais (EDPs), para descrever a evolução temporal e temporal-espacial, respectivamente de uma ou mais populações. Assim, o que se quer conhecer, a cada instante de

tempo t , o número total de indivíduos $N(t)$ e/ou a densidade local da população $\rho(x,t)$ para modelos com informação espacial.

Uma classe importante de equações é aquela na qual a variável independente não aparece explicitamente, tais equações são ditas autônomas. Essas equações são importantes para o estudo do crescimento ou declínio populacional de uma espécie, um assunto importante em campos que vão da medicina à ecologia, passando pela economia global. Porém ao considerar fatores como: predadores, competição, sobrevivência, e assim por diante, acreditamos que a teoria *fuzzy* pode ser mais apropriada (BARROS et al., 2000).

Para manipular essas informações, que por sua natureza são incertas e imprecisas, através da matemática clássica, poderá nos conduzir a limitações significativas em uma análise, pois para descrever certos fenômenos frequentemente consideramos alguns fatores como desprezíveis ou constantes, o que não traduz a realidade.

Questionamentos a respeito de incertezas têm sido preocupações de filósofos e de pesquisadores ao longo dos tempos (BARROS, 1997). É precisamente nesse tipo de incerteza que a lógica *fuzzy* tem dado suas principais contribuições.

A teoria dos conjuntos *fuzzy* foi introduzida em 1965, pelo matemático Lotfi Asker Zadeh, com a principal intenção de dar um tratamento matemático a certos termos linguísticos subjetivos.

A lógica *fuzzy* é utilizada para automatizar atividades que compreendem situações ambíguas, tratando os problemas sob um novo paradigma.

A potencialidade está na capacidade de se obter conclusões e respostas baseadas no vago, no ambíguo. Ela aproxima a decisão computacional da decisão humana, tornando as máquinas mais capacitadas ao seu trabalho, em aplicações para automatizar atividades de natureza industrial, biológica ou química.

Estender conceitos da teoria clássica de conjuntos para a teoria de conjuntos *fuzzy* é primordial para aplicação desta técnica. O método proposto por Zadeh, também conhecido como princípio de extensão, é uma das ideias básicas que promove a extensão de conceitos matemáticos *não-fuzzy* em *fuzzy* (BARROS, 1997).

Alguns estudos recentes motivam a realização deste trabalho: Melo (2009) aplicou o princípio ao Problema do Oscilador harmônico unidimensional; Pires (2010) implementou o princípio de extensão a funções matemáticas gerais; Lima (2011) apresentou uma ferramenta computacional *fuzzy* para aplicação do princípio de extensão em funções aritméticas e Botelho (2012) uma análise rigorosa do princípio de extensão aplicado na difração de Fendas.

Face o exposto, a motivação deste trabalho está no fato do princípio de extensão de Zadeh se mostrar uma técnica flexível, de fácil compreensão, capaz de lidar com situações de imprecisão de maneira formal e consistente, e a possibilidade de obtenção de bons resultados na aplicação desta ferramenta em casos reais.

Sendo assim, o objetivo deste trabalho é apresentar uma metodologia computacional, que permita aplicar o Princípio de Extensão, sem utilizar operações típicas de máximos e mínimos, através da inversão de funções, conforme apresentado por (BOTELHO 2012), para analisar à Equação Logística ou Modelo de Verhulst, aplicando-a a três estudos populacionais: População da cidade de Sorocaba-SP apresentado em (ROVEDA, et al. 2009), população da cidade de Lavras-MG confrontando com os dados disponibilizados pelo IBGE/Censo e da população do rebanho de bovinos no Brasil apresentado em (FREITAS, et al. 2005).

Nesse trabalho, apresenta-se um modelo computacional do Princípio de Extensão de Zadeh através da inversão de funções, aplicado ao estudo do crescimento populacional utilizando-se o modelo de crescimento de Verhulst.

O texto está dividido da seguinte forma: a seção 2 destina-se a apresentação dos conceitos básicos: A equação de Verhulst e o modelo de crescimento populacional; Lógica *Fuzzy*, enfatizando-se as definições de: conjuntos *fuzzy*; operações entre conjuntos *fuzzy*; Princípio de Extensão; α -nível; números *fuzzy*; operações aritméticas com números *fuzzy*; controlador *fuzzy* e *defuzzificação*. Na seção 3 serão apresentados os recursos computacionais, a implementação do Princípio de Extensão pelo método clássico e a implementação do Princípio de Extensão através da inversão de funções. Na seção 4 serão apresentados e discutidos os resultados encontrados com a aplicação da metodologia proposta neste trabalho, comparados aos dados reais obtidos através do IBGE/Censo e aos apresentados em (ROVEDA, et al. 2009) e (FREITAS, et al. 2005). Nas seções 5 e 6 são apresentadas, respectivamente, as considerações finais e as conclusões obtidas na execução desse trabalho.

2 REFERENCIAL TEÓRICO

Nesta seção é apresentada uma revisão de literatura sobre a equação de Verhulst, sobre lógica *fuzzy* e o Princípio de Extensão de Zadeh.

2.1 A equação de Verhulst

Conforme descrito por GRACIAS, et al. (2010), o modelo de Verhulst, ou modelo logístico, supõe que uma população deverá crescer até um limite máximo, isto é, a população tende a se estabilizar. Essa estabilidade da população no modelo de Verhulst está relacionada com a capacidade de suporte do meio que esta população vive. A equação diferencial para este modelo é representada pela equação 2.1:

$$\frac{dN}{dt} = rN\left(1 - \frac{N}{k}\right) \quad (2.1)$$

onde r é a taxa de crescimento da população e k é o nível de saturação da população.

A solução da equação 2.1 é dada por: (ANGELINI, 2002).

$$N(t) = \frac{kN_0}{N_0 + (k - N_0)e^{-rt}} \quad (2.2)$$

onde $N(0)=N_0$ sendo que N_0 é diferente de zero e de k .

2.2 Lógica *fuzzy*

Geralmente nos expressamos, em português, através de interrogações e exclamações, mas, para comunicar fatos ou informações, usamos sentenças. Tecnicamente, uma sentença (ou proposição) é uma frase que pode ser apenas verdadeira ou falsa (GERSTING, 1995). Esse conceito é conhecido como lógica Aristotélica.

O filósofo grego Aristóteles (384 - 322 a.C.) considerado o pai da ciência da lógica, estabeleceu um conjunto de regras rígidas para que conclusões pudessem ser aceitas como logicamente válidas. O emprego da lógica de Aristóteles levou a uma linha de raciocínio lógico, baseado em premissas e conclusões. Essa lógica trata as afirmações de forma binária, classificando-as como verdadeiras ou falsas (MELO, 2009).

Há uma enorme distância entre o mundo real e nossa visão bivalente do mesmo, a começar pelo fato que o mundo real contém um número infinito de sombreamentos e graus de cinza entre as cores preta e branca. Um outro exemplo típico ocorre em diagnósticos médicos: o profissional necessita contabilizar em sua mente um número enorme de fatores diferentes, e até contraditórios, para se descrever a doença do paciente (SIMÕES et al., 2007).

A lógica *fuzzy* vem se tornando cada vez mais importante como ferramenta capaz de capturar informações vagas, ambíguas ou imprecisas, geralmente presentes no processo de comunicação humano, para transformá-las em forma numérica, permitindo ampla aplicação em sistemas computacionais.

A lógica *fuzzy* ou Lógica Difusa diferente da Lógica Clássica, que apenas permite a classificação de "Verdadeiro" ou "Falso", é capaz de atribuir valores lógicos intermediários (ZADEH, 1965).

A lógica diz respeito aos princípios formais que sustentam as leis do pensamento e reflete os fundamentos de muitas engenharias e ciência da computação. Sistemas lógicos consideram declarações com dois possíveis valores: verdadeiro ou falso. Em sistemas de muitos valores lógicos uma declaração pode ser verdadeira ou falsa ou pode assumir valores de verdade intermediários.

Em contraste, a lógica *fuzzy* vai muito além dos limites de lógicas polivalentes e admite valores de verdade que são conjuntos *fuzzy* do intervalo de unidade[0,1]. Valores de verdade podem ser considerados como uma caracterização linguística de valores de verdade numéricos. Assim, a

lógica *fuzzy* diz respeito aos princípios do raciocínio aproximado (PEDRYCZ et al., 1998).

2.2.1 Conjuntos *fuzzy*

A teoria dos conjuntos nebulosos é, substancialmente, uma extensão da teoria dos conjuntos tradicionais. Nesta última, os conjuntos são também conhecidos como conjuntos *crisp*, ou seja, têm por característica os limites claramente definidos. Neste tipo de conjunto, é estabelecida uma relação binária entre um elemento e o conjunto, existindo apenas as alternativas de *pertencer* ou *não pertencer* ao citado conjunto (CORRAR et al., 2007).

Na teoria dos conjuntos *fuzzy* esta condição foi flexibilizada, e um elemento pertence a um conjunto de acordo com um grau de pertinência. Assim, enquanto no conjunto *crisp* a pertinência é do tipo binária, "tudo" ou "nada", "sim" ou "não"; representada pelos graus 1 ou 0. No conjunto *fuzzy* a pertinência é gradual, variando de 0 (não pertence ao conjunto) a 1 (pertence integralmente ao conjunto). Por exemplo, o grau de pertinência de uma pessoa no conjunto de "pessoas altas" é mais flexível do que uma simples resposta "sim" ou "não" e pode ser um número real, tal como 0,75.

Os Conjuntos *fuzzy* foram introduzidos por Lotfi A. Zadeh em 1965 como uma extensão da noção clássica de conjunto. Na teoria clássica dos conjuntos, a associação de elementos em um conjunto é avaliada em termos binários de acordo com uma condição bivalente - um elemento ou pertence ou não pertence ao conjunto. Por outro lado, a teoria dos conjuntos *fuzzy* permite a avaliação gradual da participação de elementos em um conjunto, o que é descrito com a ajuda de uma função de pertinência valorada no intervalo unitário real $[0, 1]$. Conjuntos *fuzzy* generalizam conjuntos clássicos, visto que as funções indicadoras de conjuntos clássicos são casos especiais das funções de pertinência de conjuntos *fuzzy*, se estes só podem tomar os valores 0 ou 1.

Tomemos como exemplo a temperatura ambiente em um determinado local, indaga-se 25°C é uma temperatura morna ou quente?

Uma representação gráfica convencional para essa questão, pode ser assim definida, ver gráfico 1:

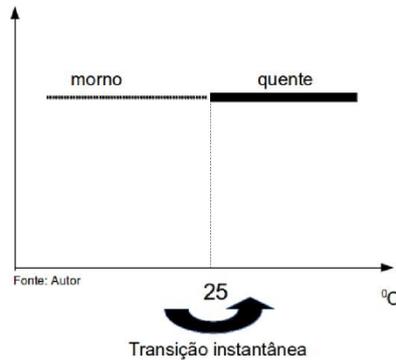


Gráfico 1 Representação da transição instantânea de temperatura.

Como se observa, a transição entre a temperatura, por exemplo, de $24,9^{\circ}\text{C}$, considerada no exemplo de um conjunto *crisp* como *morna*, para a temperatura de $25,1^{\circ}\text{C}$, considerada como *quente*, é instantânea, abrupta.

A teoria dos conjuntos *fuzzy* permite solucionar esse problema. Em primeiro lugar, devem-se definir as variáveis linguísticas. No exemplo, há apenas uma variável: **temperatura**. Em seguida, define-se o conjunto nebuloso associado à variável, que pode ser, por exemplo, **frio**, **morno** e **quente**.

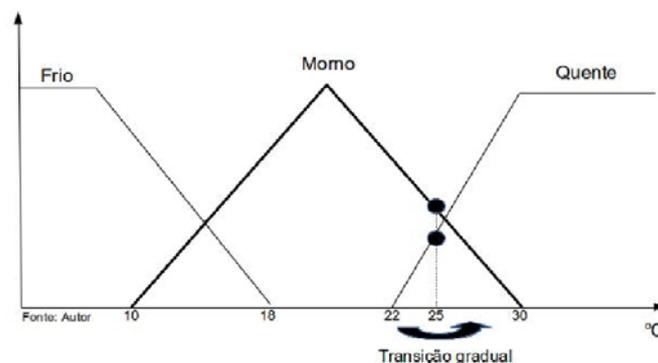


Gráfico 2 Representação da transição gradual de temperatura

Observe no gráfico 2 que para valores inferiores a 10°C a sensação é considerada **Frio**, para 20°C **Morno** e acima de 30°C **Quente**. Note que a passagem de uma sensação térmica para outra não é abrupta como no gráfico 1, portanto valores entre 20°C e 30°C habitam os dois conjuntos **Morno** e **Quente**, com diferentes graus de pertinência, conforme pode ser visto no gráfico 3.

Assim, uma temperatura de 20°C seria (0,0;1,0;0,0), ou seja, não é frio nem quente, pois o grau de pertinência nestes casos é 0,0 e é morno com grau de pertinência 1,0. Por outro lado, a temperatura de 25°C seria (0,0;0,7;0,3) caracterizando-se como não é frio - pertinência 0,0, morno com 0,7 de grau de pertinência e quente com 0,3.

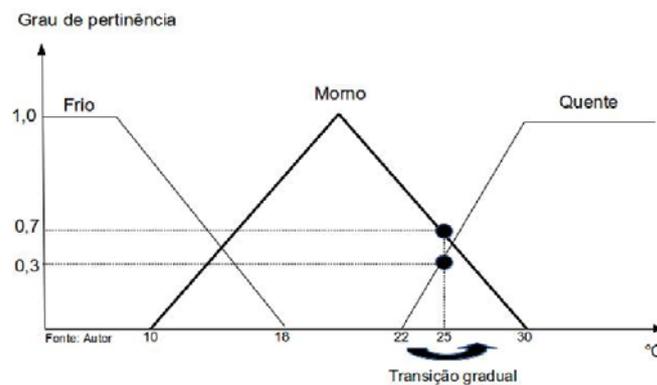


Gráfico 3 Transição de temperatura com conjuntos *fuzzy*

Com esse recurso conceitual, a transição do estado *morno* para o estado *quente* é gradual, permitindo-se afirmar que 25°C é parcialmente *morno* e parcialmente *quente*.

2.2.2 Operações entre conjuntos *fuzzy*

Existem inúmeras formas de se realizar as operações entre conjuntos *fuzzy*. Aqui, abordaremos a forma mais simples e mais frequente na

literatura, como já apresentado em (TANAKA, 1997) e (PEDRYCZ et al., 1998).

Definição 1. *Dados dois conjuntos fuzzy A e B , a união $A \cup B$, a intersecção $A \cap B$ e o complemento A' são conjuntos fuzzy com funções de pertinência dadas pelas seguintes equações, respectivamente:*

$$\mu_{A \cup B}(x) = \max\{\mu_A(x), \mu_B(x)\} = \mu_A(x) \vee \mu_B(x), \forall x \in U, \quad (2.3)$$

$$\mu_{A \cap B}(x) = \min\{\mu_A(x), \mu_B(x)\} = \mu_A(x) \wedge \mu_B(x), \forall x \in U, \quad (2.4)$$

$$\mu_{A'}(x) = 1 - \mu_A(x), \quad (2.5)$$

Exemplo 1. *(Conjuntos fuzzy dos pacientes com cefaleia e/ou dos febris). Suponha que o conjunto universo U seja composto pelos pacientes de um hospital, identificados pelos números 1,2,3,4 e 5. Sejam A e B os subconjuntos fuzzy que representam os pacientes com cefaleia e com febre, respectivamente, como apresentado na tabela 1.*

Essa tabela também ilustra as operações união, intersecção e complemento. Precisamente, os valores das colunas, exceto os da primeira, indicam os graus com que cada paciente pertence aos conjuntos fuzzy A , B , $A \cup B$, $A \cap B$, A' , $A \cup A'$, $A \cap A'$, respectivamente.

Na coluna $A \cup A'$, o valor 0,3 indica que o paciente 1 está tanto no grupo dos com cefaleia como no grupo dos sem cefaleia. Notemos que este é um fato inaceitável na teoria clássica de conjuntos na qual tem-se a lei do terceiro excluído, isto é, $A \cup A' = \emptyset$.

Dados dois subconjuntos fuzzy A e B de U , temos que A e B serão iguais se suas funções de pertinência coincidem, ou seja, se $\mu_A(x) = \mu_B(x), \forall x \in U$.

Temos, a seguir, as principais propriedades das operações com subconjuntos fuzzy: lembrando que o conjunto vazio \emptyset tem função de

pertinência $\mu_{\emptyset}(x) = 0$, enquanto que o conjunto (U) tem função de pertinência $\mu_U(x) = 1, \forall x \in U$.

Tabela 1 Ilustração das operações entre subconjuntos *fuzzy*

Paciente	Cefaleia: A	Febre: B	$A \cup B$	$A \cap B$	A'	$A \cup A'$	$A \cap A'$
1	0,7	0,6	0,7	0,6	0,3	0,3	0,7
2	1,0	1,0	1,0	1,0	0,0	0,0	1,0
3	0,4	0,2	0,4	0,2	0,6	0,4	0,6
4	0,5	0,5	0,5	0,5	0,5	0,5	0,5
5	1,0	0,2	1,0	0,2	0,0	0,0	1,0

$$A \cup B = B \cup A \text{ e } A \cap B = B \cap A, \quad (2.6)$$

$$A \cup (B \cap C) = (A \cup B) \cap C \text{ e } A \cap (B \cup C) = (A \cap B) \cup C, \quad (2.7)$$

$$A \cup A = A \text{ e } A \cap A = A, \quad (2.8)$$

$$A \cup (B \cap C) = (A \cup B) \cap (A \cup C) \\ \text{e } A \cap (B \cup C) = (A \cap B) \cup (A \cap C), \quad (2.9)$$

$$A \cap \emptyset = \emptyset \text{ e } A \cup \emptyset = A, \quad (2.10)$$

$$A \cap U = A \text{ e } A \cup U = U, \quad (2.11)$$

$$(A \cup B)' = A' \cap B' \text{ e } (A \cap B)' = A' \cup B', \quad (2.12)$$

2.2.3 Princípio de extensão

O Princípio de Extensão de Zadeh é uma das ideias básicas da teoria *fuzzy* que promove a extensão de conceitos matemáticos *não-fuzzy* em *fuzzy* (CORCOLL-SPINA, 2010). Esta ferramenta é indispensável, a nosso ver, para a estruturação matemática quando se modelam fenômenos envoltos em grande grau de incerteza (BARROS, 1997).

Segundo CORCOLL-SPINA (2010), O princípio de extensão pode ser descrito da seguinte forma: (a) o grau de pertinência de um valor do contradomínio é definido diretamente pelo grau de pertinência de sua pré-imagem. (b) quando um valor do contradomínio é mapeado por vários do

domínio, o seu grau de pertinência é obtido pelo maior dos graus de pertinência dos valores da entrada, conforme pode ser observado no gráfico 4.

Definição 2. Seja $f: X \rightarrow Z$ uma função injetora e A um subconjunto fuzzy de X , enumerável (ou finito), e dado por

$$\sum_{i=1}^{\infty} \mu_A(x_i)/x_i \quad (2.13)$$

Então, o Princípio de Extensão garante que $\hat{f}(A)$, extensão da função f aplicada em A , é um subconjunto fuzzy de Z , dado por

$$\hat{f}(A) = \hat{f} \left(\sum_{i=1}^{\infty} \mu_A(x_i)/x_i \right) = \sum_{i=1}^{\infty} \mu_A(x_i)/f(x_i) \quad (2.14)$$

Definição 3. Seja a função $f: X \rightarrow Z$ e A um subconjunto fuzzy de X . A extensão de Zadeh de f é a função \hat{f} que, aplicada a A , fornece o subconjunto fuzzy $\hat{f}(A)$ de Z , cuja função de pertinência é dada por

$$\mu_{\hat{f}(A)}(z) = \begin{cases} \text{SUP}_{\{x: f(x)=z\}} \mu_A(x) & \text{se } \{x: f(x)=z\} \neq \emptyset \\ 0 & \text{se } \{x: f(x)=z\} = \emptyset \end{cases} \quad (2.15)$$

Se f for uma função bijetora, então

$$\{x: f(x)=z\} = \{f^{-1}(z)\}. \quad (2.16)$$

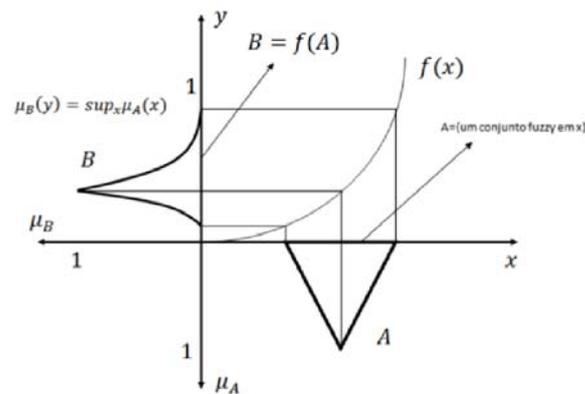


Gráfico 4 Figura da imagem de um subconjunto *fuzzy* a partir do princípio de extensão para uma função f .

Exemplo 2. Seja $f(x) = 3x + 2$, $x \geq 0$ e A um conjunto fuzzy com suporte enumerável. Deseja-se determinar um número fuzzy "em torno de 4".

Determinação da função de pertinência triangular:

$$\mu(x) \begin{cases} x - 3 & \text{se } 3 < x \leq 4, \\ 5 - x & \text{se } 4 \leq x < 5, \\ 0 & \forall \text{ outras } x. \end{cases} \quad (2.17)$$

$$\tilde{4} = 0/3 + 0,5/3,5 + 1/4 + 0,5/4,5 + 0/5.$$

Determinação da imagem de A via princípio de extensão:

$$\tilde{4} = 0/f(3) + 0,5/f(3,5) + 1/f(4) + 0,5/f(4,5) + 0/f(5).$$

$$\tilde{4} = 0/11 + 0,5/12,5 + 1/14 + 0,5/15,5 + 0/17.$$

Portanto, a imagem de A por f pode ser deduzida do conhecimento das imagens de x_i por f . O grau de pertinência de $z_i = f(x_i)$ em $\tilde{f}(A)$, é o mesmo de x_i em A , conforme gráfico 5.

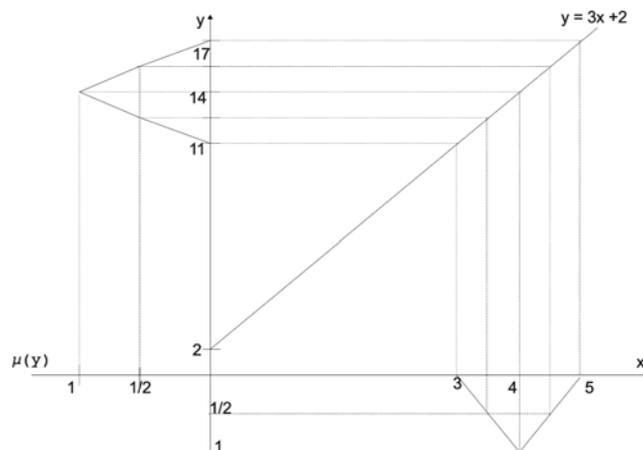


Gráfico 5 Função de Pertinência triangular para o número "próximo" de 4 em $f(x) = 3x + 2$.

Exemplo 3. Seja $f(x) = x^2 - 5x + 6$, $x > 0$, e A um conjunto fuzzy com suporte enumerável. Deseja-se determinar um número fuzzy "em torno de 3".

Determinação da função de pertinência triangular:

$$\mu(4) \begin{cases} \frac{x-1}{2} & \text{se } 1 < x \leq 3, \\ \frac{5-x}{2} & \text{se } 3 \leq x < 5, \\ 0 & \forall \text{ outros } x. \end{cases} \quad (2.18)$$

$$\tilde{3} = 0/1 + 0,25/1,5 + 0,5/2 + 0,75/2,5 + 1/3 + 0,75/3,5 + 0,5/4 + 0,25/4,5 + 0/5.$$

Determinação da imagem de A via princípio de extensão, conforme gráfico 6:

$$\tilde{3} = 0/f(1) + 0,25/f(1,5) + 0,5/f(2) + 0,75/f(2,5) + 1/f(3) + 0,75/f(3,5) + 0,5/f(4) + 0,25/f(4,5) + 0/f(5).$$

$$\tilde{3} = 0/2 + 0,25/0,75 + 0,5/0 + 0,75/-0,25 + 1/0 + 0,75/0,75 + 0,5/2 + 0,25/3,75 + 0/6.$$

Notamos, que um mesmo valor de função, possui pertinências diferentes.

Faz-se necessário então, a utilização da operação *máximo*, com intuito de obter a maior pertinência de cada valor.

$$\tilde{3} = \max(0 \wedge 0,5)/2 + \max(0,25 \wedge 0,75)/0,75 + \max(0,5 \wedge 1)/0 + 0,75/-0,25 + 0,25/3,75 + 0/6.$$

$$\tilde{3} = 0,5/2 + 0,75/0,75 + 1/0 + 0,75/-0,25 + 0,25/3,75 + 0/6.$$

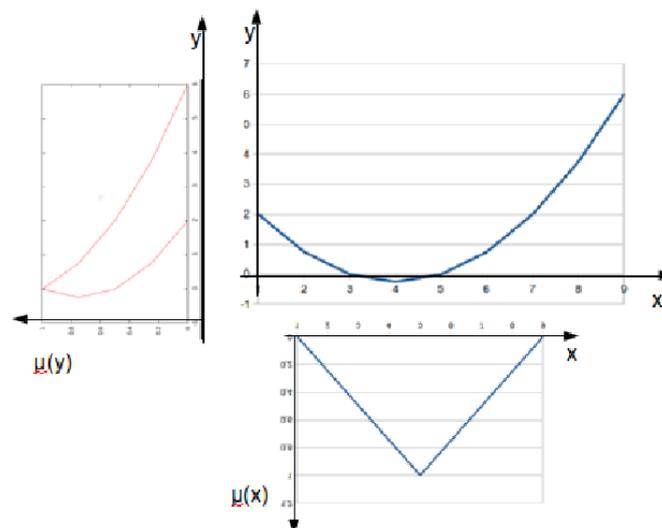


Gráfico 6 Função de Pertinência triangular para o número próximo de 3 em $f(x)=x^2-5x+6$.

2.2.4 Conceito de α -nível

A seguinte definição tem papel importante na inter-relação entre as teorias clássica de conjuntos e *fuzzy*, pois oferece uma forma de relacionar um conjunto *fuzzy* com conjuntos clássicos.

Definição 4. Um corte α em um conjunto fuzzy A é especificado por um conjunto ordinário (*crisp*) que contém todos os elementos de A , pertencentes ao universo X , que possuem grau de pertinência maior ou igual a α , ou seja:

$$A_\alpha = \{x \in X: \mu_A(x) \geq \alpha\}. \quad (2.19)$$

A partir deste conceito pode-se inferir que para $\alpha = 1$, A_α corresponde aos elementos de X que pertencem a A , com o conceito de pertinência dos conjuntos *crisp*.

Definição 5. O subconjunto clássico de U definido por $\text{supp}_F = \{x \in U: \mu_F(x) > 0\}$ é denominado suporte de F .

Notemos que o suporte de um subconjunto *crisp* coincide com o próprio conjunto.

Definição 6. Seja A um subconjunto fuzzy de U e $\alpha \in [0,1]$. O α -nível de A é o subconjunto clássico de U definido por $[A]^\alpha = \{x \in U: \mu_A(x) \geq \alpha\}$, se $0 \leq \alpha \leq 1$. O nível zero de um subconjunto fuzzy de A é definido como sendo o menor subconjunto (clássico) fechado de U que contém o conjunto suporte de A .

Exemplo 4. Seja U o conjunto de todos os alunos de uma determinada escola, e A um subconjunto fuzzy de U formado pelas meninas consideradas de estatura mediana, com função de pertinência:

$$\mu_A(x) = \begin{cases} x - 1,65, & \text{se } 1,65 \leq x \leq 1,70, \\ 1,75 - x, & \text{se } 1,70 \leq x \leq 1,75, \\ 0, & \text{se } x \in [1,65, 1,75] \end{cases} \quad (2.20)$$

Assim, chega-se a conclusão que:

$$x - 1,65 = \alpha, \text{ logo } x = \alpha + 1,65,$$

$$1,75 - x = \alpha, \text{ logo } x = 1,75 - \alpha.$$

Temos:

$$[A]^\alpha = [\alpha + 1,65, 1,75 - \alpha] \text{ para } 0 \leq \alpha \leq 1, \text{ portanto}$$

$$[A]^0 = [1,65, 1,75].$$

O gráfico 7 representa o α -níveis: $[A]^\alpha$ e $[A]^0 \in U$.

Desse modo, observa-se no gráfico 7 que o grau de pertinência de $[A]^0$ é 0 para o intervalo $[1,65, 1,75]$ e para um determinado α verifica-se o intervalo demonstrado no gráfico 7 denotado por $[A]^\alpha$.

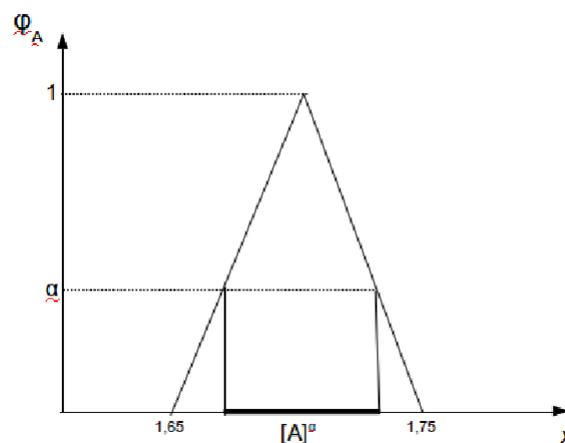


Gráfico 7 α -níveis $[A]^\alpha$ e $[A]^0 \neq U$.

Dois conjuntos *fuzzy* são considerados iguais sempre que suas funções de pertinência forem idênticas, ou seja,

$$A = B \Leftrightarrow \mu_A(x) = \mu_B(x) \text{ para } x \in X. \quad (2.21)$$

A igualdade entre conjuntos *fuzzy* pode ser determinada por meio dos α -níveis, onde a igualdade dos conjuntos é determinada quando os α -níveis coincidem para todo $\alpha \in [0,1]$.

2.2.5 Números *fuzzy*

Introduziremos o conceito de números *fuzzy*, o qual faz-se necessário para podermos quantificar predicados qualitativos e fazer contas com os mesmos.

Definição 7. Um subconjunto *fuzzy* A é chamado de número *fuzzy* quando o conjunto universo no qual μ_A está definida, é o conjunto dos números reais R e satisfazem às seguintes condições:

1. todos os α -níveis de A são não-vazios, com $0 \leq \alpha \leq 1$;
2. todos os α -níveis de A são intervalos fechados de R ;
3. $\text{supp}A = \{x \in R : \mu_A(x) > 0\}$ é limitado.

Os números *fuzzy* mais comuns são os triangulares, trapezoidais e em forma de sino.

Exemplo 5. (Triangular)

Um número *fuzzy* é dito triangular se sua função de pertinência é da forma, conforme gráfico 8.

$$\mu_A(x) = \begin{cases} 0, & \text{se } x \leq a, \\ \frac{x-a}{b-a} & \text{se } a < x \leq b, \\ \frac{b-x}{c-b} & \text{se } b \leq x < c, \\ 0, & \text{se } x \geq c. \end{cases} \quad (2.22)$$

A expressão "em torno de" pode ser modelada matematicamente pelo número *fuzzy* triangular simétrico A .

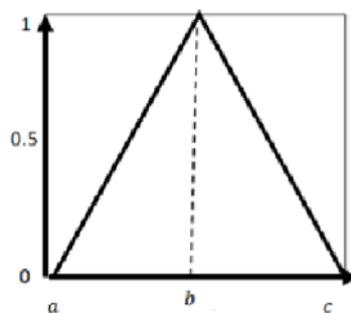


Gráfico 8 Função de pertinência triangular.

Exemplo 6. (Trapezoidal)

Um número *fuzzy* A é dito trapezoidal, ver gráfico 9, se sua função de pertinência tem a forma de um trapézio e é dada por:

$$\mu_A(x) = \begin{cases} \frac{x-a}{b-a} & \text{se } a \leq x < b, \\ 1 & \text{se } b \leq x \leq c, \\ \frac{d-x}{d-c} & \text{se } c < x \leq d, \\ 0, & \text{caso contrário.} \end{cases} \quad (2.23)$$

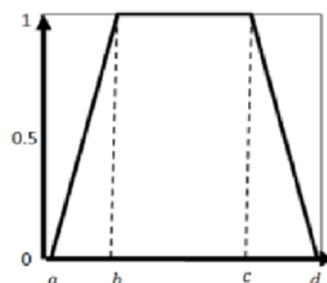


Gráfico 9 Função de pertinência trapezoidal.

Exemplo 7. (Forma de sino)

Um número fuzzy tem a forma de sino se sua função de pertinência for "suave" e simétrica em relação a um número real, conforme gráfico 10. A seguinte função de pertinência tem estas propriedades para x , m e θ dados.

$$\mu_A(x) = \{e^{-\theta(x-m)^2}; \text{ se } \theta > 0. \quad (2.24)$$

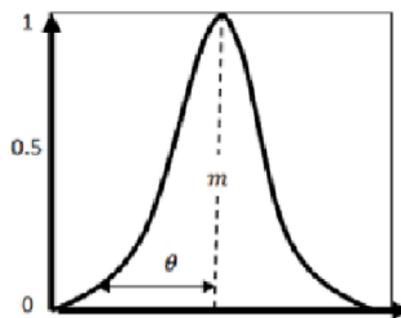


Gráfico 10 Função de pertinência gaussiana.

A definição da função de pertinência é bastante subjetiva, pois uma função especificada para um mesmo conceito pode apresentar resultados variados se for definida por diferentes pessoas. Além de definir o quanto certo elemento pertence a determinado conjunto, a função de pertinência também serve para representar os limites de um conjunto *fuzzy*. Para determinar a forma de variação e os limites de cada um destes conjuntos, deve-se definir a função de pertinência adequadamente, normalmente feita através da consulta a um especialista na área estudada.

As funções de pertinência dos conjuntos *fuzzy* podem assumir diversas formas, sendo a função trapezoidal, ou um de seus casos particulares, e a função triangular, os tipos de função mais utilizadas. Usualmente uma função trapezoidal é expressa através de quatro parâmetros, que descrevem os quatro vértices do trapézio que representa a função. Na

função trapezoidal existem dois parâmetros com pertinência igual a um, ao passo que na triangular existe apenas um parâmetro com esse grau de pertinência.

A função de pertinência triangular tem sido frequentemente usada em muitas aplicações *fuzzy*. Talvez, a justificativa mais óbvia para sua utilização seja a simplicidade da função.

Temos também as operações aritméticas com números *fuzzy*. Estas estão estreitamente ligadas às operações aritméticas intervalares.

2.2.6 Operações aritméticas com números *fuzzy*

As operações aritméticas envolvendo números *fuzzy* estão estreitamente ligadas às operações aritméticas intervalares, que é um ramo da matemática desenvolvido para lidar com o cálculo de tolerância (PEDRYCZ et al., 1998). As técnicas da aritmética intervalar, consistem em uma alternativa para alcançar limites garantidos para os resultados de processos computacionais, através do controle rigoroso e automático do erro sobre o resultado.

As operações aritméticas para intervalos estendem as respectivas operações para números reais. Um número real pode ser considerado como um intervalo fechado com extremos iguais (BOAVENTURA, 2010).

Conforme descrito por BOTELHO (2012), as definições que seguem podem ser vistas como casos particulares do princípio de extensão.

Sejam A e B dois números *fuzzy* e λ um número real.

Definição 8. *Sejam dois números fuzzy A e B definidos em um mesmo conjunto universo X , representados pelas respectivas funções de pertinência $\mu_A(x)$ e $\mu_B(x)$. A operação de adição entre A e B pode ser definida em função de seus respectivos cortes- α , ou seja:*

$$A + B = [a_1^\alpha, a_2^\alpha] + [b_1^\alpha, b_2^\alpha], \quad (2.25)$$

cujos resultados da operação é dado por:

$$A + B = [a_1^\alpha + b_1^\alpha, a_2^\alpha + b_2^\alpha] = C^\alpha, \quad (2.26)$$

e função de pertinência:

$$\mu_{(A+B)}(Z) = \sup_{\min_{(x,y)} \mu_A(x), \mu_B(y)} [\mu_A(x), \mu_B(y)]. \quad (2.27)$$

Definição 9. A subtração entre dois números fuzzy A e B , definidos em um mesmo conjunto universo X , é dada por:

$$A - B = [a_1^\alpha - b_1^\alpha, a_2^\alpha - b_2^\alpha] = C^\alpha, \quad (2.28)$$

e sua função de pertinência é dada por:

$$\mu_{(A-B)}(Z) = \sup_{\min_{(x,y)} \mu_A(x), \mu_B(y)} [\mu_A(x), \mu_B(y)]. \quad (2.29)$$

Definição 10. A multiplicação de λ por A é o número fuzzy, λA , cuja função de pertinência é:

$$\mu_{\lambda A}(Z) = \begin{cases} \sup_{\min_{(x)} \lambda, \mu_A(x)} [\lambda, \mu_A(x)] & \text{se } \lambda \neq 0, \\ 0 & \text{se } \lambda = 0. \end{cases} \quad (2.30)$$

Definição 11. A multiplicação entre dois números fuzzy A e B , especificados ambos em um mesmo conjunto universo, é também definida em função de seus respectivos cortes- α , ou seja:

$$A \times B = [a_1^\alpha \times b_1^\alpha, a_2^\alpha \times b_2^\alpha] \text{ para } \alpha \in [0,1]. \quad (2.31)$$

de forma análoga à adição e subtração, obtém-se o resultado da multiplicação através da composição de seus cortes- α .

com função de pertinência:

$$\mu_{(A \times B)}(Z) = \sup_{\min_{(x,y) \in X \times Y = Z} [\mu_A(x), \mu_B(y)]}. \quad (2.32)$$

Definição 12. O procedimento de divisão entre dois números fuzzy A e B , pertencentes a um mesmo conjunto universo, é dado por:

$$A + B = [a_1^\alpha + b_1^\alpha, a_2^\alpha + b_2^\alpha] \text{ para } \alpha \in [0,1], \quad (2.33)$$

e sua função de pertinência:

$$\mu_{(A+B)}(Z) = \sup_{\min_{(x,y) \in X+Y=Z} [\mu_A(x), \mu_B(y)]}. \quad (2.34)$$

2.2.7 Controlador fuzzy

Os controladores *fuzzy* têm a característica de que, para um sistema *fuzzy* qualquer, cada entrada *fuzzy* faz-se corresponder uma saída *fuzzy*. Porém, há casos em que se a entrada for *crisp*, espera-se que a saída também seja *crisp* construída de alguma maneira específica.

A transformação de todos os conceitos vistos até agora em um mecanismo de aplicação prática foi o desafio a que se propôs (MANDANI, 1974), que projetou um modelo, denominado originariamente *Fuzzy Logic Controller*, que inspirou muitos trabalhos e vem comumente demonstrado, com ou sem adaptações, em diversas obras literárias a respeito do tema (CORRAR et al., 2007).

O módulo de *fuzzificação* é o estágio no qual as entradas do sistema são modeladas por conjuntos *fuzzy* com seus respectivos domínios. É nesse estágio que se justifica a grande importância de especialistas do fenômeno a

ser modelado. Assim, mesmo que a entrada seja crisp, essa será *fuzzificada* por meio de sua função de pertinência.

A *fuzzificação* promoverá a transformação de entradas discretas em entradas nebulosas (*fuzzy inputs*), considerando, por exemplo, os intervalos de graus Celsius que rotulam as temperaturas ambientais em fria, fresca, normal, morna e quente. Assim, por exemplo, se for imputado ao sistema a informação de que a temperatura ambiente é de 33^oC, o processo de *fuzzificação* indicará que essa é *quente* com um grau de pertinência de 0,46 e *morna* com um grau de 0,20 (CORRAR et al., 2007).

O módulo de base de regras pode ser considerado como um módulo que faz parte do núcleo do controlador *fuzzy*. Ele é composto pelas proposições *fuzzy* e cada uma destas proposições é descrita na forma linguística.

Se x_1 é A_1 e x_2 é A_2 e ... e x_n é A_n , então u_1 é B_1 e u_2 é B_2 e ... u_m é B_m .

De acordo com as informações de um especialista. É neste ponto que as variáveis e suas classificações linguísticas são catalogadas e, em seguida, modeladas por conjuntos *fuzzy*, ou ainda, funções de pertinência.

Existem vários métodos para se obter essas funções de pertinência, por exemplo: apelos intuitivos, ajustes de curvas, interpolações e até mesmo rede neurais.

As regras estabelecidas nesta etapa, podem ser de natureza condicional ou incondicional. Para as regras de natureza condicional, o formato utilizado para expressá-las utiliza a sintaxe: SE (premissa), ENTÃO (conclusão). Por exemplo: SE a temperatura é alta, ENTÃO acione o sistema de ventilação. Para as regras de natureza incondicional, não é utilizada a segunda parte condicionante, ou seja, o termo *Então*. Assim, a expressão simplesmente realiza uma asserção. No exemplo dado, a regra incondicional seria, assim, expressa: A temperatura é alta (CORRAR et al., 2007).

No módulo de inferência cada proposição *fuzzy* é traduzida *matematicamente* por meio das técnicas da lógica *fuzzy*. Esse módulo

fornecerá a saída *fuzzy* a ser adotada pelo controlador, a partir de cada entrada *fuzzy*.

Conforme descrito por CORRAR et al. (2007), os procedimentos de inferência consistem na avaliação das variáveis antecedentes pelas regras de produção estabelecidas. Para tanto, as seguintes etapas devem ser realizadas pelo controlador *fuzzy*:

- Identificar os valores correspondentes aos graus de pertinência dos termos linguísticos correspondentes às antecedentes;
- Determinar a força das conclusões de cada regra disparada, a partir de um determinado grau de pertinência dos termos linguísticos;
- Definir a saída nebulosa.

No controlador *fuzzy*, a cada entrada *fuzzy*, o módulo de inferência produz uma saída *fuzzy* que indica o controle a ser adotado. Entretanto, se a entrada for um número real, espera-se que a saída correspondente seja também um número real. Porém, isso, em geral, não ocorre em controladores *fuzzy*, pois, mesmo para uma entrada *crisp*, a saída é *fuzzy*. Dessa forma, deve-se indicar um método para *defuzzificar* a saída e obter um número real que indicará o controle a ser adotado.

2.2.8 Defuzzificação

A saída do controlador *fuzzy* é um subconjunto *fuzzy* do universo da saída. Como o processo requer um sinal *não-fuzzy* em sua entrada, deve-se fazer uma interpretação daquele conjunto *fuzzy*.

Inicialmente, o processo combina as saídas nebulosas resultantes do processo de inferência, pela função que Dubois(1980) denominou de Agregação ou Resolução de Conflitos. Segundo esse autor, a função de agregar é usada para decidir qual a ação de controle deve ser levada como resultado das diversas regras de produção ativadas (CORRAR et al., 2007).

Esta interpretação, denominada *defuzzificação*, pode ser feita através de vários métodos, mas os mais usados são:

- Método do Centro de Área, também denominado Centróide ou do Centro de Gravidade, que resulta da ponderação de todos os valores possíveis da saída nebulosa. Os graus de pertinência são utilizados como pesos para o cálculo de uma média ponderada.
- Método do Máximo, que apresenta o ponto no qual o valor de saída nebuloso tem o seu máximo. Por esse método, a decisão considerará somente o valor que apresentar o maior grau de pertinência.
- Método da Média dos Máximos, que calcula o valor médio dentre todos os pontos máximos das saídas nebulosas.

3 METODOLOGIA

Este capítulo apresenta a sequência lógica utilizada para implementar o Princípio de Extensão de Zadeh aplicado ao modelo de crescimento populacional de Verhulst, através da inversão de funções.

Para possibilitar um aprofundamento no tema lógica *fuzzy*, em especial o Princípio de Extensão de Zadeh, foi realizada uma revisão bibliográfica, verificando-se tanto os aspectos teóricos quanto aplicações práticas envolvendo essa ferramenta. Foram consultados artigos, dissertações, teses e livros, nacionais e internacionais.

3.1 Recursos computacionais

Na implementação computacional do sistema utilizou-se a linguagem Java, que é uma linguagem livre, multiplataforma, o que possibilita que o programa seja executado em qualquer sistema operacional sem necessidade de recompilação nem adaptações, confiabilidade, regularidade e facilidade de uso.

Outro fator importante na escolha desta linguagem é que através da utilização de uma IDE (Integrated Development Environment - um ambiente integrado para desenvolvimento de software), neste caso o Netbeans, que também é livre, o desenvolvimento da interface gráfica para utilização do sistema foi facilitado.

Na implementação deste trabalho, foram criadas duas classes: a primeira responsável por manipular e armazenar os dados necessários para realização dos cálculos, vide Apêndice A e a segunda responsável pela interface gráfica do programa, vide Apêndices B e C.

Para geração dos gráficos, foi utilizado o software GnuPlot, que é destinado à visualização de gráficos e superfícies, úteis em aplicações científicas nas áreas de física, matemática, estatística e engenharias. Este

aplicativo é de domínio público e tem versões para vários sistemas operacionais.

3.2 Princípio de extensão de Zadeh pelo método clássico

A partir da equação (2.2) descrita na seção 2 assumiu-se como parâmetros incertos k e r , separadamente, permitindo-se uma análise da incerteza em cada parâmetro da equação.

Convém ressaltar que a aplicação do Princípio de Extensão de Zadeh, pelo método clássico gera um conjunto de dados (números *fuzzy*) bastante extenso, do qual, deveremos descartar a maior parte destes, através de suas operações típicas de *máximos* e *mínimos*, demonstrado no exemplo 3 da seção 2, que demanda um custo computacional bastante elevado.

Mediante o exposto considerou-se as seguintes especificações escolhidas arbitrariamente:

- $N_0 = 10$ (população inicial)
- k "em torno de 100" (capacidade suporte - escolhida como fator incerto)
- $r = 0,2$ (taxa de crescimento)
- $t = [1:50]$ (intervalo de tempo observado)
- $n = 100$ (Quantidade de pontos à discretizar)

A partir destes valores determinamos o *passo*, ver equação (3.1) correspondente a variação entre cada ponto discretizado, iniciando-se no limite inferior até o limite superior, informados nos parâmetros do sistema.

$$\Delta = \frac{Lim_{sup} - Lim_{inf}}{n - 1}, \quad (3.1)$$

Devemos então tomar cada valor discretizado da variável *fuzzy* (k), junto aos demais parâmetros informados e aplicarmos no modelo definido na seção 2.1 equação (2.2) afim de obtermos o valor correspondente da função

a este ponto. Em seguida verificamos a pertinência, em cada ponto, de acordo com a equação (2.22) da seção 2.1.

Os valores obtidos para a variável *fuzzificada* k , para a função $N(t)$ e sua respectiva pertinência, são armazenados em uma estrutura de dados matriz com três colunas, (variável k , valor da função $N(t)$ e pertinência $(\mu(k)|\mu(N(t)))$, e cinco mil linhas, necessárias para armazenar a quantidade de pontos discretizados, $n=100$, para cada unidade de tempo $t= [1:50]$.

Parte dos dados obtidos através desta metodologia, estão descritos na tabela 2. Tendo em vista que o valor da função calculado corresponde à uma população qualquer num dado instante de tempo, considerou-se inserir com precisão de duas casas.

Tabela 2 Dados obtidos em $t = 1$ e k “em torno de 100”

k	$\mu(k)$	$t = [0,10]$	Vr. Função (N(t))	$\mu(N(t))$
80	0	1	11,88	0
82	0,1	1	11,89	0,1
84	0,2	1	11,90	0,2
86	0,3	1	11,91	0,3
88	0,4	1	11,91	0,4
⋮	⋮	⋮	⋮	⋮
100	1	1	11,95	1
⋮	⋮	⋮	⋮	⋮
112	0,4	1	11,98	0,4
114	0,3	1	11,98	0,3
116	0,2	1	11,98	0,2
118	0,1	1	11,99	0,1
120	0	1	11,99	0
⋮	⋮	⋮	⋮	⋮

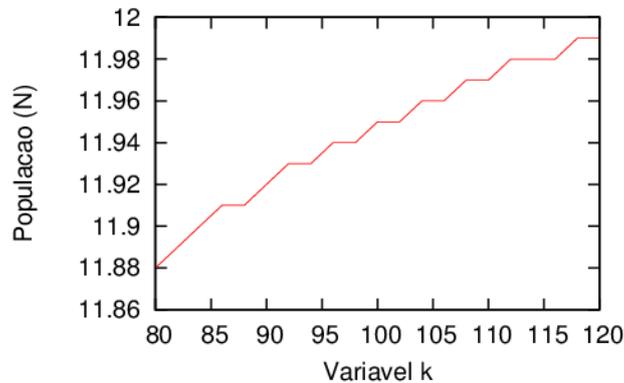


Gráfico 11 função $N(t)$ com k “em torno de 100”.

Isto posto, constatamos que para diferentes valores de k obtemos o mesmo valor de $N(t)$, vide gráfico11, que faz com que um mesmo valor de função tenha mais de uma pertinência.

Para solucionar este problema devemos aplicar a operação de *máximo*, conforme descrito na seção 2.2.3 exemplo 3, que requer percorrer a matriz gerada, verificando-se as duplicidades de valores para o posterior descarte dos valores repetidos, o que seria bastante dispendioso em termos de processamento computacional.

Para evitar este processamento, conseqüentemente ganhando agilidade no processo, e a necessidade de descartar os valores repetidos, será aplicado então, o Princípio de Extensão de Zadeh, através da inversão de funções.

3.3 Aplicação do princípio de extensão de Zadeh através da inversão de funções - parâmetro k

Em função das especificações dadas na seção 3.2, inicialmente procedeu-se com a determinação do valor de $N(t)$ para k em seu limite

inferior, ou seja, $k = 80$, no intervalo de tempo que se desejar analisar. Neste contexto, para $t = 1$, por exemplo, $N(t) = 11,88$ e para $t = 10$, $N(t) = 41,08$.

Determinamos ainda o valor de $N(t)$ para k em seu limite superior, ou seja, $k = 120$: Para $t = 1$, por exemplo, $N(t) = 11,99$ e para $t = 10$, $N(t) = 48,21$.

Com os valores de função estabelecidos nos limites desejados, determinamos então o Δ , vide equação (3.1), com $n=100$ (quantidade de pontos a discretizar) resultou em:

- para $t = 1 \rightarrow \Delta = 0,0011$
- para $t = 10 \rightarrow \Delta = 0,0720$

O cálculo da inversa da equação (2.2) descrita na seção 2.1, em função do parâmetro k , fazendo $N(t) = F$, foi calculada em na equação (3.2).

$$k = \frac{FN_0(1 - e^{-rt})}{N_0 - Fe^{-rt}}. \quad (3.2)$$

Neste ponto, ao invés de variarmos nosso parâmetro incerto k , na quantidade de pontos desejados, o valor da função F será discretizado, aplicando-se cada valor obtido na sua inversa afim de encontrarmos k , e após calculamos sua pertinência, conforme segue a representação dada nos gráficos 12 e 13, nos quais a função de pertinência da variável k "em torno de 100" em $t=10$, e de $f(k)$ em $t=10$, respectivamente.

Neste trabalho adotamos a função de pertinência triangular, pelo fato desta possuir um único valor para a pertinência máxima, conforme descrito na seção 2.2.5, o que nos permite aproximar o resultado *fuzzy* do valor real.

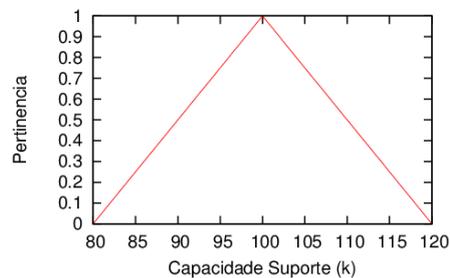


Gráfico 12 pertinência da variável k "em torno de 100".

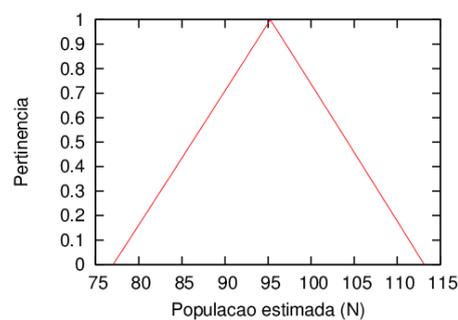


Gráfico 13 pertinência de $f(k)$ em $t = 10$.

O gráfico 14 mostra o agrupamento de todos os conjuntos *fuzzy* obtidos para a variável k , "em torno de 100", variando a cada 1 (um) ano. Este gráfico tem uma escala, onde temos os graus de pertinência representados na escala de cinza, variando no intervalo $[0,1]$. O tom mais escuro representa os valores de $f(x)$, com pertinência máxima igual a 1. Podemos observar que a partir do instante $t=15$, a incerteza aumenta e que atingimos o limite de capacidade desta população com t acima de 30. Observamos também que a medida que a população aumenta, no decorrer do tempo, aumentam-se também as incertezas.

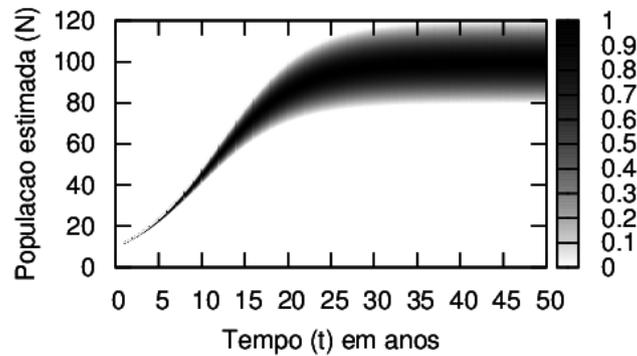


Gráfico 14 função $N(t)$ com k "em torno de 100".

Para gerar o resultado foi escolhido o método de defuzzificação Centro de Gravidade, ver seção 2.2.8, pois é mais indicado quando a variação entre os valores do conjunto de saída é pequena, pois, por exemplo em $t=1$, partindo do valor da função $N(t)$ em seu limite inferior $N(t) = 11,88$ com variação de $\Delta = 0,0011$, os valores obtidos serão muito próximos.

Abaixo vemos o gráfico 15 com o resultado da aplicação do Princípio de Extensão de Zadeh, através da Inversão de Função, onde observamos que a partir do instante $t=35$ a função atingiu seu valor limite.

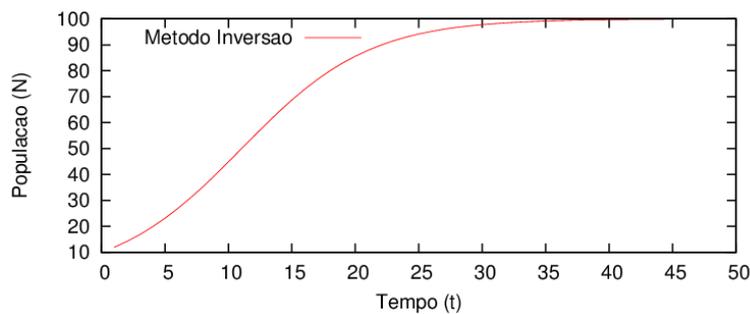


Gráfico 15 Função $N(t)$ com valores *defuzzificados*

3.4 Aplicação do princípio de extensão de Zadeh através da inversão de funções - parâmetro r

Em relação ao que foi especificado na seção 3.3, aqui tomaremos, arbitrariamente, como fator constante $k = 1000$ (capacidade suporte) e *fuzzificaremos* r "em torno de 0,3" (taxa de crescimento - considerado fator incerto).

Procedeu-se com a determinação do valor de $N(t)$ para r em seu limite inferior, ou seja, $r = 0,1$, no intervalo de tempo que se desejar analisar. Neste contexto, para $t = 1$, por exemplo, $N(t) = 11,0401$ e para $t = 10$, $N(t) = 26,0276$.

Determinamos ainda o valor de $N(t)$ para r em seu limite superior, ou seja, $r = 0,5$: Para $t = 1$, por exemplo, $N(t) = 16,3809$ e para $t = 10$, $N(t) = 599,8596$.

Com os valores de função estabelecidos nos limites desejados, determinamos Δ , vide equação (3.1).

Com $n=500$ (quantidade de pontos a discretizar) resultou em:

para $t = 1 \rightarrow \Delta = 0,010703006$,

para $t = 10 \rightarrow \Delta = 1,149963928$.

O cálculo da inversa da equação (2.2) descrita na seção 2.1, em função do parâmetro r , fazendo $N(t) = F$, foi calculada em (3.3):

$$r = \frac{[\ln(N_0) + \ln(k - F)] - [\ln(F) + \ln(k - N_0)]}{t} \quad (3.3)$$

Neste ponto, ao invés de variarmos nosso parâmetro incerto r , na quantidade de pontos desejados, o valor da função F será discretizado, aplicando-se cada valor obtido na sua inversa afim de obtermos r , e após calculamos sua pertinência, conforme segue a representação dada nos gráficos 16 e 17, nos quais a função de pertinência da variável r "em torno de 0.3" em $t=10$, e de $f(k)$ em $t=10$, respectivamente.

Neste trabalho adotamos a função de pertinência triangular, pelo fato desta possuir um único valor para a pertinência máxima, conforme descrito na seção 2.2.5, o que nos permite aproximar o resultado *fuzzy* do valor real.

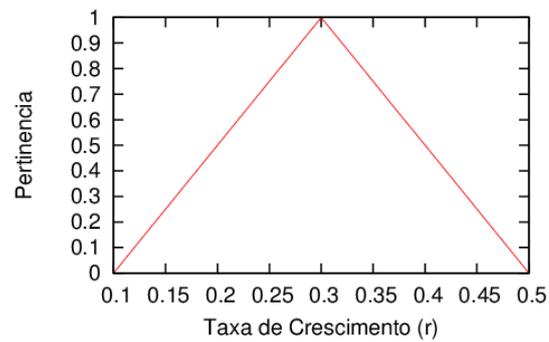


Gráfico 16 pertinência da variável r "em torno de" 0,3.

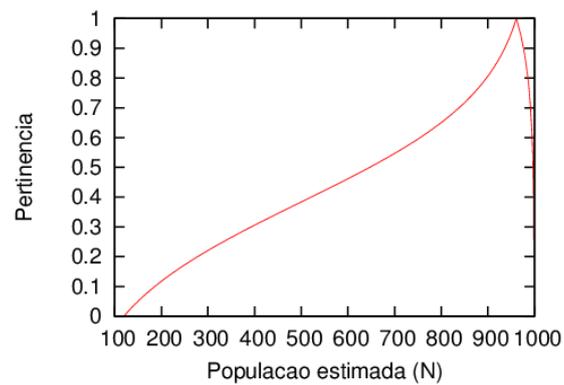


Gráfico 17 pertinência de f(r)

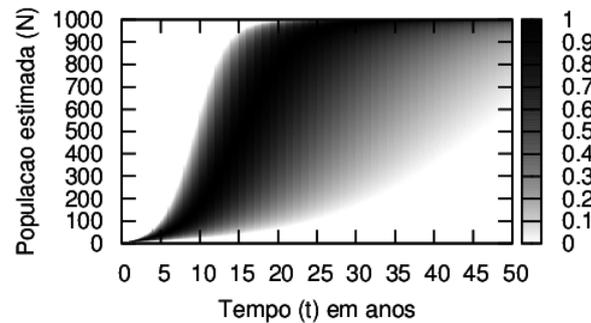


Gráfico 18 Função $N(t)$ com r "em torno de" 0,3.

O gráfico 18 mostra o agrupamento de todos os conjuntos *fuzzy* obtidos para a variável r , "em torno de 0.3", variando a cada 1 (um) ano. Este gráfico tem uma escala, onde temos os graus de pertinência representados na escala de cinza, variando no intervalo $[0,1]$. O tom mais escuro representa os valores de $f(x)$, com pertinência máxima igual a 1. Podemos observar que a partir do instante $t=10$, a incerteza aumenta e que atingimos o limite de capacidade desta população com t acima de 25. Observamos também que a medida que a população aumenta, no decorrer do tempo, aumentam-se também as incertezas.

Utilizando-se o método do Centro de Gravidade, obtemos o gráfico 19 com o resultado da aplicação do Princípio de Extensão de Zadeh, através da Inversão de Função. Observamos que nos primeiros anos, a curva de crescimento é bastante acentuada, e após o instante $t=20$, se torna menos acentuada, demonstrando uma desaceleração no crescimento da população e que ela só se aproximará do seu limite após o instante $t=50$.

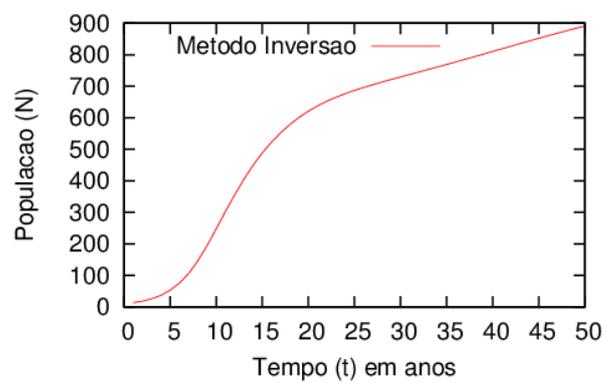


Gráfico 19 Função $N(t)$ com valores defuzzificados.

4 RESULTADOS E DISCUSSÕES

Nesta seção serão apresentadas três aplicações da metodologia proposta neste trabalho:

1. Na seção 4.1 Um comparativo ao estudo feito por ROVEDA, et al. (2009), sobre a população da cidade de Sorocaba-SP.
2. Na seção 4.2 Uma projeção para a população futura da cidade de Lavras-MG, a partir dos dados extraídos do IBGE/Censo.
3. Na seção 4.3 Um comparativo ao estudo feito por FREITAS, et al. (2005), sobre o rebanho bovino brasileiro.

4.1 Estudo sobre a dinâmica populacional da cidade de Sorocaba-SP

Em ROVEDA, et al. (2009), foi apresentado um estudo da dinâmica populacional da cidade de Sorocaba-SP, utilizando-se um sistema *p-fuzzy*, ou seja, um sistema *parcial-fuzzy* baseado em regras, que são um conjunto de regras linguísticas na forma: Se <antecedente> Então <consequente>, que descrevem a relação entre as variáveis de entrada e saída. O Sistema têm como variável de entrada "População" (p) e de saída "Variação" (Δp).

Neste estudo foram elaboradas 4 (quatro) diferentes bases de regras, identificadas como Modelo 1, Modelo 2, Modelo 3 e Modelo 4, considerando a variação absoluta $\left(\frac{\Delta p}{\Delta t}\right)$ e relativa da população (Δp) .

O Modelo 1 consiste de um Sistema *p-fuzzy* com variação absoluta; o Modelo 2 de um Sistema iterativo utilizando uma base de regras para variação relativa; o Modelo 3 também consiste de um sistema iterativo, porém com a base de regras para variação relativa modificada e o Modelo 4 utiliza a base de regras para variação absoluta com o sistema iterativo.

Denominamos de sistema parcialmente *fuzzy*, ou sistema *p-fuzzy*, ao sistema iterativo, conforme definido na equação (4.1):

$$\begin{cases} x_{k+1} = f(x_k) \\ x_0 \in R^n \end{cases} \quad (4.1)$$

onde $f(x_k)$ é quase linear, isto é, $f(x) = x + \Delta(x)$, $\Delta(x_k) \in R^n$ e $\Delta(x_k)$ é obtido por um sistema baseado em regras *fuzzy*.

Para a variável população adotou-se o seguinte conjunto de termos linguísticos, tanto no estudo de variação absoluta quanto relativa, vide gráfico 20.

$$P = \{ \text{Baixa}(B), \text{Média Baixa}(MB), \text{Média}(M), \text{Média Alta}(MA), \text{Alta}(A), \text{Altíssima}(AL) \}.$$

A variação absoluta foi dividida da seguinte forma, vide gráfico 21.

$$\Delta P_{ab} = \{ \text{Baixa negativa}(Bn), \text{Baixa positiva}(Bp), \text{Média positiva}(Mp), \text{Alta positiva}(Ap) \}.$$

Com base então nestes conjuntos de termos linguísticos, formaram-se as bases de regras para os modelos 1 e 4, que descrevemos abaixo:

1. se a população é Baixa então a variação é Baixa positiva;
2. se a população é Média Baixa então a variação é Média positiva;
3. se a população é Média então a variação é Alta positiva;
4. se a população é Alta então a variação é Baixa positiva;
5. se a população é Altíssima então a variação é Baixa negativa.

As funções de pertinência para as variáveis População e Variação absoluta são apresentadas nos gráficos 20 e 21.

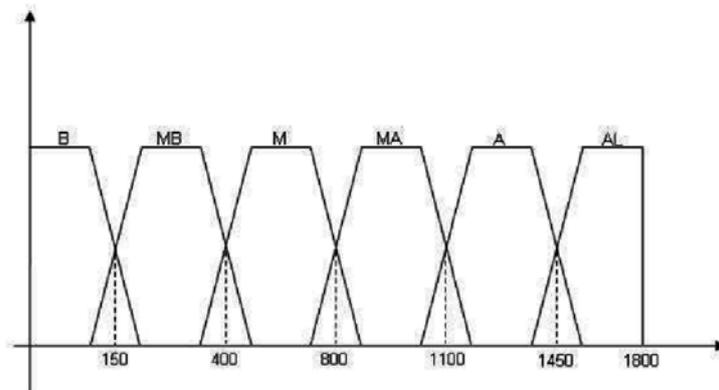


Gráfico 20 Variável população. ROVEDA, et al. (2009)

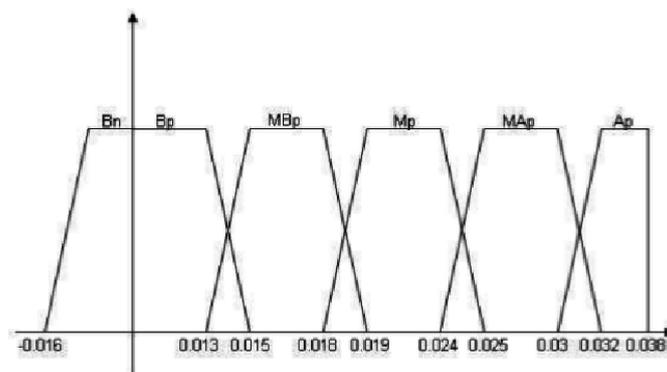


Gráfico 21 Δ_{ab} população. ROVEDA, et al. (2009).

A variação relativa foi dividida da seguinte forma, vide gráfico 22.

$$\Delta P_{rel} = \{ \text{Baixa negativa}(B_n), \text{Baixa positiva}(B_p), \text{Média Baixa positiva}(MB_p), \\ \text{Média positiva}(M_p), \text{Média Alta positiva}(MA_p), \text{Alta positiva}(A_p) \}.$$

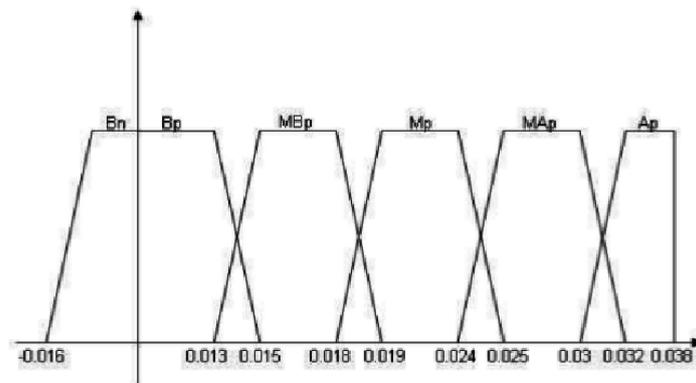


Gráfico 22 Δ_{rel} população. ROVEDA, et al. (2009).

Com base então nestes conjuntos de termos linguísticos, formaram-se as bases de regras para o modelo 2, que descrevemos abaixo:

1. se a população é Baixa então a variação é Alta positiva;
2. se a população é Média Baixa então a variação é Média Alta positiva;
3. se a população é Média então a variação é Média positiva;
4. se a população é Média Alta então a variação é Média Baixa positiva;
5. se a população é Alta então a variação é Baixa positiva;
6. se a população é Altíssima então a variação é Baixa negativa.

Para o modelo 3 foram ajustados os dois primeiros itens da base de regras:

1. se a população é Baixa então a variação é Média Alta positiva;
2. se a população é Média Baixa então a variação é Alta positiva;

Utilizando-se dos dados descritos neste trabalho, conforme tabela 3, acrescentados os cálculos de erro relativo para cada modelo em relação aos dados do censo, inicialmente fizemos uma análise da taxa de crescimento daquele município, tomando como base os dados censitários.

Tabela 3 Dados extraídos de Roveda, et al. (2009)

Ano	Censo	Verhulst	Modelo1	Modelo2	Modelo3	Modelo4
1960	136	136	136	136	136	136
1970	175	192(9,71%)	199(13,71%)	185(5,71%)	188(5,71%)	172(1,71%)
1980	269	269(0%)	274(1,86%)	243(9,67%)	266(1,12%)	225(16,36%)
1991	379	376(0,79%)	356(6,07%)	322(15,04%)	390(2,90%)	320(15,57%)
1996	431	434(0,70%)	397(7,89%)	367(14,85%)	441(2,32%)	393(8,82%)
2000	493	483(2,03%)	436(11,56%)	408(17,24%)	483(2,03%)	471(4,46%)
Erro Médio		2,65%	8,22%	12,50%	3,16%	9,38%

Para o cálculo da taxa de crescimento, utilizamos o método geométrico, conforme equação (4.2), que consiste em subtrair-se 1 da raiz enésima do quociente entre a população final (P_t) e a população no começo do período considerado (P_0), multiplicando-se o resultado por 100, sendo "n" igual ao número de anos no período.

$$r = \left[\sqrt[n]{\frac{P_t}{P_0}} - 1 \right] \times 100. \quad (4.2)$$

Tabela 4 Taxa de crescimento obtida de acordo com dados do censo.

Período	variação
1960 – 1970	2,55%
1970 – 1980	4,39%
1980 – 1991	3,13%
1991 – 1996	2,60%
1996 – 2000	3,42%

Por não possuímos informações sobre a capacidade suporte do meio, ou seja, qual seria o limite sustentável para a população da cidade de Sorocaba-SP, será *fuzzificada* a variável r que corresponde a taxa de crescimento.

Tendo em vista a variação obtida, conforme Tabela 4, adotamos então o valor de $r = 0.032$ como limite inferior, pertinência máxima em $r = 0.038$ e limite superior em $r = 0.044$. Serão discretizados $n = 100$ pontos, para um intervalo de 40 (quarenta) anos: $t=[1,40]$.

Com base no gráfico 20 adotamos a capacidade suporte $k = 1.800$ habitantes, que corresponde a variável população com valor "Altíssima".

O próximo passo, é determinarmos a valor da função, a partir da equação (2.2) descrita na seção 2.1, nos limites inferior e superior, para cada ano analisado, e assim calcularmos o Δ (passo), conforme equação (3.1) descrita na seção 3.2.

Feito isto, discretizamos o valor da função, a partir do limite inferior, em incrementos de Δ até o limite superior na quantidade de pontos desejada.

Agora, aplicamos cada valor de função discretizado, na inversa da função, conforme equação (3.3) descrita na seção 3.4, afim de obtermos o conjunto *fuzzy* da variável r - taxa de crescimento.

Em seguida, calculamos a pertinência em cada ponto, como pode ser observado nos gráficos 23 e 24.

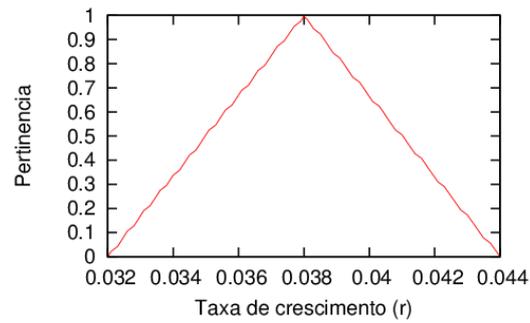


Gráfico 23 pertinência no instante $t = 10$ (1970).

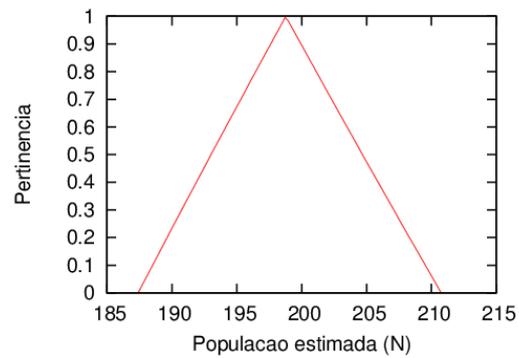


Gráfico 24 pertinência no instante $t = 10$ (1970).

No gráfico 25 visualizamos o conjunto *fuzzy* gerado, onde nota-se um aumento da incerteza a partir de $t=20$ anos.

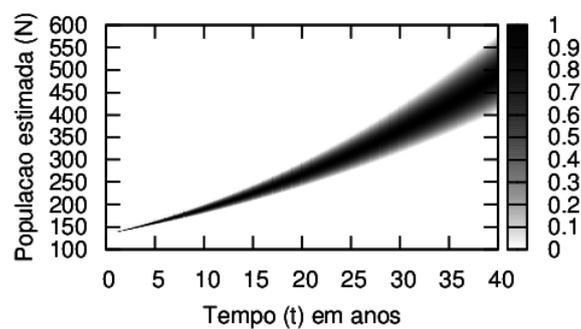


Gráfico 25 Conjunto *fuzzy* gerado a partir da variável r .

Utilizando o método centróide obtemos o gráfico 26, confrontando os valores estimados pelo Princípio de Extensão, com os de ROVEDA, et al. (2009), e determinando o erro relativo em relação aos dados censitários.

A comparação foi feita em relação ao Modelo 3, por este ter apresentado o menor erro médio relativo.

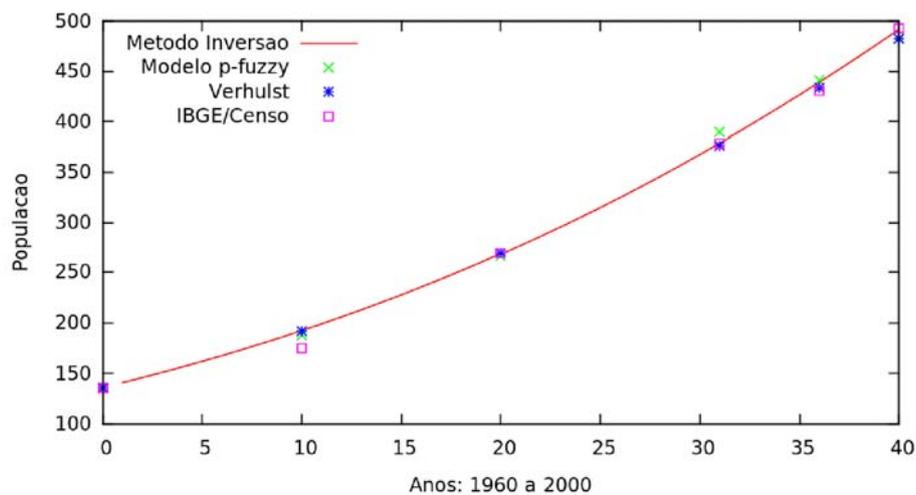


Gráfico 26 Comparação com ROVEDA, et al. (2009)

Com exceção dos valores previstos para 1970, ver gráfico 26 e tabela 5, os demais anos apresentaram um erro relativo abaixo de 2% e o erro médio foi menor que os encontrados pelo modelo de Verhulst e pelo modelo *p-fuzzy*, demonstrando que o método aplicado é satisfatório e projeta valores muito próximos dos reais.

Comparativo entre os valores apresentados em ROVEDA, et
 Tabela 5 al. (2009) e os calculados através do Princípio de Extensão
 através da inversão de funções.

Ano	Censo	Verhulst	Modelo3	P.Extensão
1960	136	136	136	136
1970	175	192(9,71%)	188(7,43%)	192(9,71%)
1980	269	269(0,00%)	266(1,12%)	268(0,37%)
1991	379	376(0,79%)	390(2,90%)	379(0,00%)
1996	431	434(0,70%)	441(2,32%)	439(1,86%)
2000	493	483(2,03%)	483(2,03)	492(0,20%)
Erro médio		2,65%	3,16%	2,43%

4.2 Estudo sobre a dinâmica populacional da cidade de Lavras-MG

A partir dos dados censitários coletados nos anos de 1991, 1996, 2000, 2007 e 2010, do município de Lavras-MG, aplicaremos o método descrito neste trabalho, com intuito de realizar uma estimativa da população futura desta cidade.

Lavras é um município brasileiro da região do Campo das Vertentes, no sul do estado de Minas Gerais. Localiza-se a uma latitude 21° 14' 30 sul e a uma longitude 44° 00' 10 oeste, estando a uma altitude de 919 metros e possuindo uma área de 564,5 km². (Fonte: IBGE).

Para determinarmos a taxa de crescimento em cada período, utilizamos o método geométrico, conforme equação (4.2), obtendo os valores descritos na tabela.

Tabela 6 Evolução populacional do município de Lavras-MG. (Fonte: IBGE)

Ano	População	Taxa de crescimento
1991	65.893	
1996	72.659	1,97% a.a
2000	78.772	2,04% a.a
2007	87.421	1,50% a.a
2010	92.200	1,77% a.a

Para estimarmos valores futuros, o grau de incerteza no parâmetro taxa de crescimento é altíssimo, tendo em vista as inúmeras variáveis que a influenciam, tais como: percentual de nascimentos, percentual de mortalidade, migração, entre outros.

Desta forma, vamos *fuzzificar* a variável r - Taxa de crescimento, com base nos valores descritos na tabela 6, "em torno" de 2,04%, com limite inferior em 1,50% e limite superior em 2,58%.

Para estimarmos a capacidade suporte (k), tomamos como base, a densidade demográfica de algumas cidades brasileiras, bem como as

características geográficas da região analisada. Assim, utilizaremos o fator ($k = 600.000$), o que corresponde a uma densidade demográfica de $1.063h/km^2$, valor este, inferior ao de municípios como Contagem-MG ($3.091h/km^2$), Ipatinga-MG ($1.453h/km^2$) e Betim-MG ($1.103h/km^2$).

Utilizando os passos descritos na seção 3.4, obtemos:

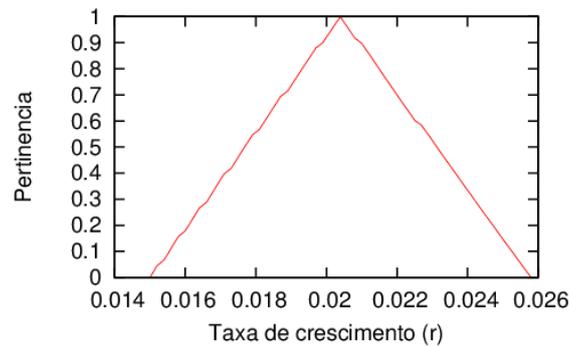


Gráfico 27 pertinência no instante $t = 30$ (2021).

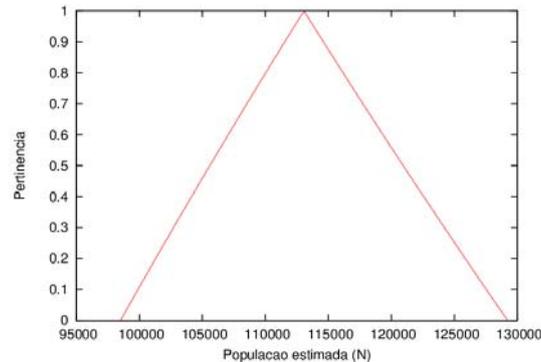


Gráfico 28 pertinência no instante $t = 30$ (2021).

Assim como descrito na seção 3.3 adotamos a função de pertinência triangular, pelo fato desta possuir um único valor para a pertinência máxima, o que nos permite aproximar o resultado *fuzzy* do valor real.

No gráfico 29 visualizamos o agrupamento de todos os números *fuzzy* gerados. Observamos que com o decorrer do tempo, mais precisamente a partir de $t = 30$ (2021), há um aumento significativo da incerteza.

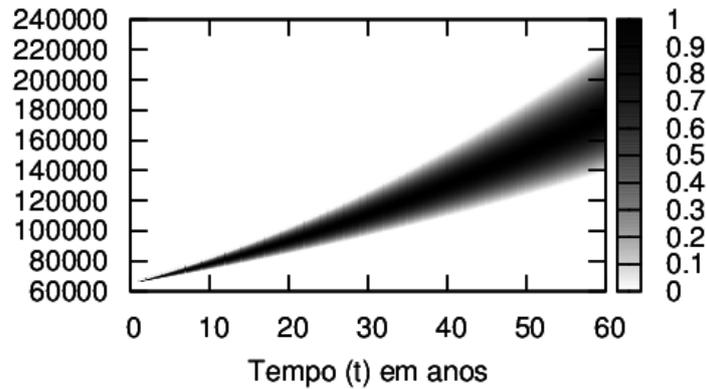


Gráfico 29 Conjunto *fuzzy* gerado a partir da variável *r*.

Utilizando o método centroide obtemos o gráfico 30, e confrontamos os dados censitários com os valores estimados pelo Princípio de Extensão através da inversão de função, e observamos que o erro relativo é pequeno, sendo 1,63% o maior encontrado, e 0,74% o erro médio relativo, indicando que o método aplicado é eficiente para projetar populações futuras.

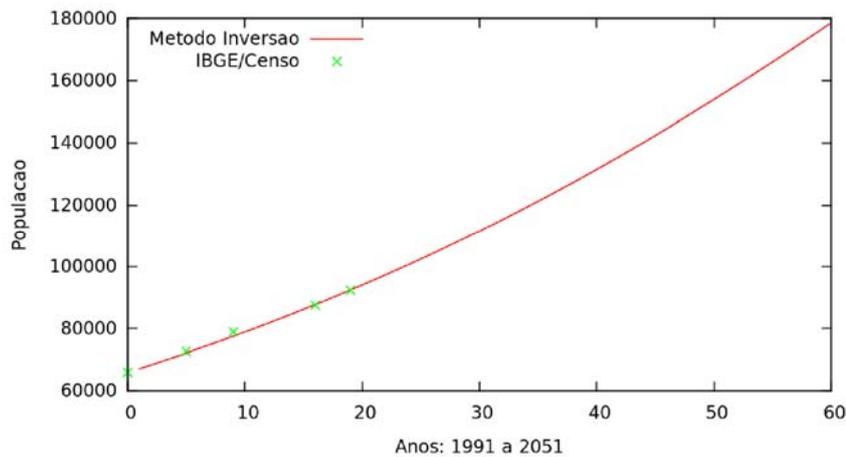


Gráfico 30 Dados *defuzzificados* comparados aos do IBGE/Censo

Tabela 7 Evolução populacional do município de Lavras-MG (IBGE x Princípio de extensão).

Ano	População(Censo)	População estimada	Erro relativo
1991	65.893	65.893	
1996	72.659	72.127	0,73% (-)
2000	78.772	77.488	1,63% (-)
2007	87.421	87.711	0,33% (+)
2010	92.200	92.433	0,25% (+)
	Erro médio relativo		0,74%
2020		109.717	
2030		129.444	
2040		151.615	
2051		178.660	

4.3 Modelagem do crescimento populacional do rebanho bovino brasileiro

Nesta seção, aplicaremos a metodologia proposta comparando os resultados obtidos com os de (FREITAS, et al. 2005).

Em FREITAS, et al. (2005), foi realizado um estudo para estimar-se o crescimento da população de bovinos, utilizando-se o modelo de Richards, ajustado pela técnica de verossimilhança profile. O crescimento do rebanho bovino foi irregular de 1983 a 2000, crescente de 1983 a 1991, apresentou oscilações de 1991 a 1995 e foi crescente daí por diante, o que significa que ainda não há dados suficientes para se estimar a capacidade de suporte dessa população. Optou-se, então, por estipular um valor alto para este nível (foi utilizado $K = 400$ milhões, porém qualquer outro valor suficientemente grande, como por exemplo, 500 ou 600 milhões poderia ser usado). A estimativa da taxa de crescimento global anual do rebanho bovino brasileiro é de 18,23%, de modo que, com qualquer taxa de abate inferior a esse valor, a população continuará crescente. Com probabilidade superior a 0,7, estima-se que o rebanho bovino brasileiro atingirá 200 milhões de animais em 11 anos, com taxa de abate anual de 16 a 17%, e, em 20 anos, com taxa de abate de aproximadamente 18%.

A partir destes dados, aplicaremos o Princípio de Extensão através da inversão de funções, *fuzzificando* a variável r (taxa de crescimento), em torno de 1,23%, que representa a diferença entre a taxa de crescimento e a taxa de abate, com limite inferior estabelecido em 0,23% e limite superior em 2,23%. Faremos ainda, a variável $k = 300$ milhões, afim de obter um melhor ajuste da função de crescimento.

Utilizando os passos descritos na seção 3.4, obtemos:

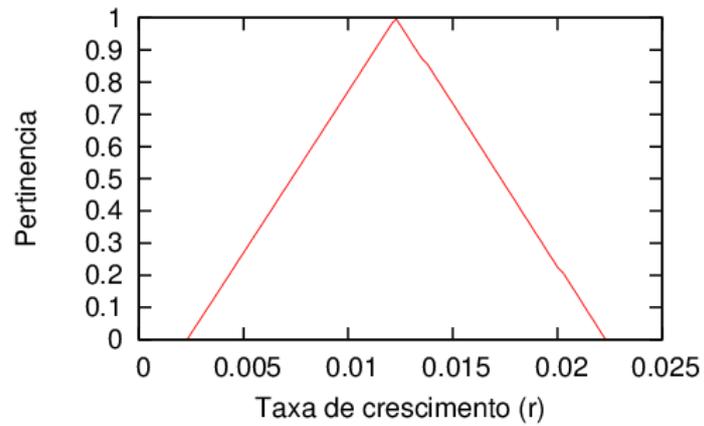


Gráfico 31 pertinência da taxa de crescimento (r) no instante $t = 10$

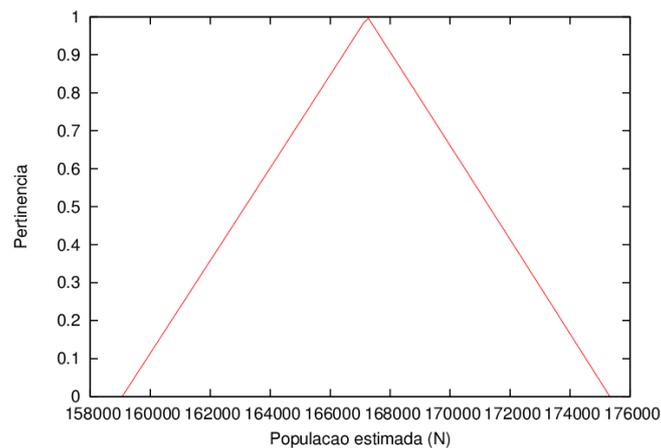


Gráfico 32 pertinência da população estimada (N) no instante $t = 10$.

Assim como descrito na seção adotamos a função de pertinência triangular, pelo fato desta possuir um único valor para a pertinência máxima, o que nos permite aproximar o resultado *fuzzy* do valor real.

No gráfico 33 visualizamos o agrupamento de todos os números *fuzzy* gerados. Observamos que com o decorrer do tempo, mais precisamente a partir de $t = 4$ (2005), nota-se que o grau de incerteza aumenta consideravelmente em função do tempo.

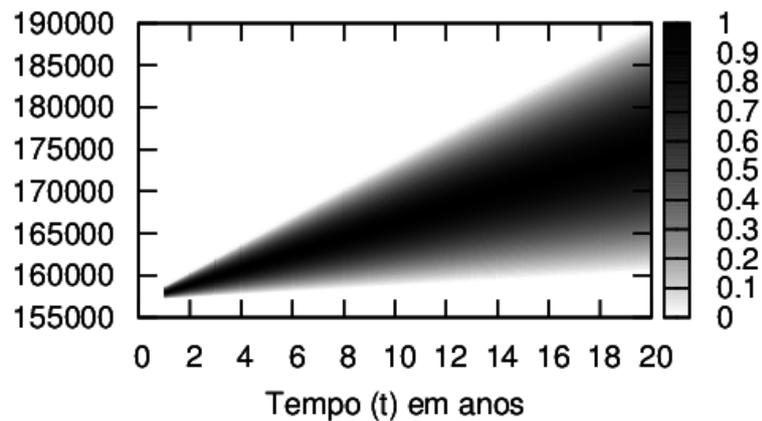


Gráfico 33 Conjunto *fuzzy* gerado a partir da variável *r*.

Utilizando o método centroide obtemos o gráfico 34, e confrontamos os dados de FREITAS, et al. (2005) com os valores estimados pelo Princípio de Extensão através da inversão de funções, e observamos que o erro relativo é pequeno, sendo 2,41% o maior encontrado, e 1,07% o erro médio, conforme tabela 8.

Dado a quantidade de fatores incertos, tais como: Taxa de crescimento, que depende de fatores internos (conforto térmico, metabolismo do animal, ração e outros) e também fatores externos (forma de criação: confinamento ou pasto), Taxa de abate, Fatores comerciais, que podem influenciar um aumento ou diminuição na taxa de abate e o erro relativo encontrado, concluímos que o método aplicado pode ser utilizado em estudos desta natureza.

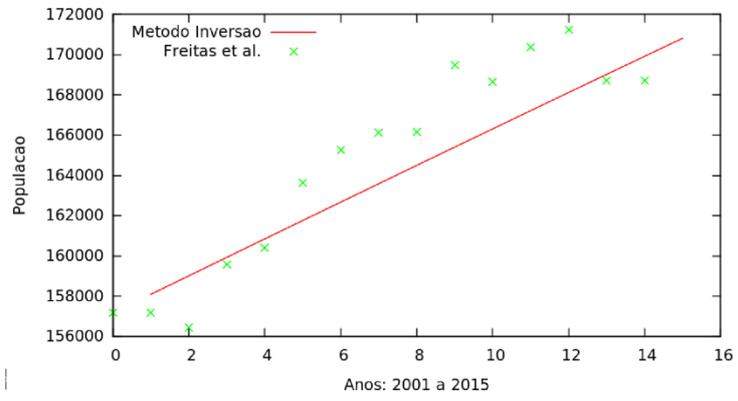


Gráfico 34 Comparação com FREITAS, et al. (2005).

Tabela 8 Evolução populacional do rebanho bovino brasileiro. (em milhões)

Ano	População(FREITAS, et al., 2005)	Pop. estimada	Erro relativo
2001	157.170	157.170	
2002	157.190	158.090	0,57% (+)
2003	156.430	159.009	0,26% (+)
2004	159.580	159.927	0,22% (+)
2005	160.390	160.844	0,28% (+)
2006	163.620	161.760	1,14% (-)
2007	165.280	162.674	1,58% (-)
2008	166.120	163.587	1,52% (-)
2009	166.150	164.498	0,99% (-)
2010	169.490	165.408	2,41% (-)
2011	168.660	166.315	1,39% (-)
2012	170.370	167.221	1,85% (-)
2013	171.240	168.124	1,82% (-)
2014	168.700	169.025	0,19% (+)
2015	168.720	169.923	0,71% (+)
Erro médio relativo			1,07%

5 CONSIDERAÇÕES FINAIS

Neste trabalho, foi apresentada uma proposta de modelo computacional para implementar o Princípio de Extensão de Zadeh, através da inversão de funções, aplicado à equação de Verhulst que é utilizada para estudo de dinâmica populacional.

Os resultados apresentados nesse trabalho correspondem a comparações feitas entre o que foi obtido pela metodologia aplicada aos apresentados em outros trabalhos sobre crescimento populacional.

Como principais resultados alcançados com a utilização do Princípio de Extensão de Zadeh, através da inversão de funções, destacam-se:

1. O modelo consegue reproduzir a dinâmica populacional de uma espécie qualquer no caso de inobservância de fatalidades, como as provocadas por guerras e epidemias, ou fatores externos.;
2. Diminui-se o custo computacional em relação à técnica clássica do Princípio de Extensão, por eliminar a etapa de descarte de valores repetidos através da operação *máximo*.
3. A possibilidade da escolha de qual variável será *fuzzificada* garante flexibilidade ao modelo.

Os resultados apresentados até o momento descrevem algumas configurações simples do modelo, que ainda pode ser customizado de diversas maneiras. Para fazer com que o modelo ganhe mais flexibilidade e outras possibilidades de análise, pretende-se possibilitar a escolha de outras funções de pertinência tais como a trapezoidal e a gaussiana.

6 CONCLUSÃO

Para estudos de dinâmica populacional, é possível implementar o Princípio de Extensão de Zadeh, de forma analítica, através da inversão de funções para projeção de populações futuras. Ao utilizar a técnica de inversão de funções diminui-se o custo computacional, se comparada à metodologia clássica, inferindo um ganho de performance. No Estudo da população de Sorocaba, os resultados obtidos, comparados aos dados censitários, aos obtidos pela aplicação do modelo determinístico de Verhulst e aos obtidos através de um sistema *p-fuzzy* são satisfatórios, uma vez que o erro médio relativo é menor que o encontrado nos demais métodos. Em relação aos estudos feitos sobre a população de Lavras e sobre o crescimento populacional de bovinos, os resultados também são considerados satisfatórios, uma vez que o erro médio ficou em torno de 1%. Como trabalho futuro, pretende-se melhorar a ferramenta, possibilitando a utilização do Princípio de Extensão de Zadeh, através de outras funções de pertinência, e a outros modelos determinísticos.

REFERÊNCIAS

ANGELINI, R. **Avaliação da capacidade-suporte da represa do Broa para a colocação de tanques-rede.** Revista Saúde e Ambiente / Health and Environment Journal 3, p. 42-48, 2002.

BARROS, L. C. **Sobre sistemas dinâmicos *fuzzy* - teoria e aplicações.** 1997. 107p. Tese de Doutorado em Estatística e Computação Científica, Instituto de Matemática da Universidade Estadual de Campinas, Campinas, 1997.

BARROS, L. C.; BASSANEZI, R. C.; TONELLI, P. A. ***Fuzzy modelling in population dynamics.*** Ecological Modelling 128, p. 27-33., 2000.

BOAVENTURA, I. A. G. **Números *Fuzzy* em Processamento de Imagens Digitais e Suas Aplicações na Detecção de Bordas.** 2010. 218p. Tese de Doutorado em Engenharia Elétrica, Escola de Engenharia de São Carlos da Universidade de São Paulo, São Carlos, 2010.

BOTELHO, T. G. **Uma nova proposta para a implementação computacional do princípio de extensão de Zadeh.** 2012. 77p. Dissertação de Mestrado em Engenharia de Sistemas, Universidade Federal de Lavras, Lavras, 2012.

MIZUKOSHI M. T; BARROS, L. C; CHALCO-CANO, Y.; ROMAN-FLORES, H.; BASSANEZI, R. C. ***Fuzzy differential equations and extension principle.*** Information Sciences 177, p. 3627-3635, 2007.

CORCOLL-SPINA, C. de O. **Lógica *Fuzzy*: reflexões que contribuem para a questão da subjetividade na construção do conhecimento matemático.** 2010. 165p. Tese de Doutorado em Educação, Faculdade de Educação da Universidade de São Paulo, São Paulo, 2010.

CORRAR, L. J.; PAULO, E.; DIAS FILHO, J. M. **Análise Multivariada.** São Paulo : Atlas, 541p., 2007.

FREITAS, A. R. de; LOIBEL, S. M. C.; ANDRADE, M. G. de; VAL, J. B. R. do. **Modelagem do Crescimento Populacional do Rebanho Bovino Brasileiro.** Revista Brasileira de Zootecnia., v.34, n.6, p.2225-2232, jul. 2005 (supl.).

GERSTING, J. L. **Fundamentos matemáticos para a ciência da computação** Rio de Janeiro : LTC, 518p., 1995.

GRACIAS, A. C.; LOURENÇO, S. R. **Aplicação de um modelo matemático na simulação da produção e importação de gás natural no Brasil até 2017.** Revista Produção Online, v.10, n.3, p. 698-717, set., 2010.

LIMA, F.P. **Ferramenta computacional fuzzy para aplicações do princípio de extensão de Zadeh em funções aritméticas.** } 2011. 93p. Dissertação de Mestrado em Engenharia de Sistemas, Universidade Federal de Lavras, Lavras, 2011.

MANDANI, E.H. **Application of Fuzzy Algorithms for Control of Simple Dynamic Plant.** IEEE (Control and Science), v.121(12), p.1585-1588, 1974.

MELO, G.J.A. **Princípio de extensão de Zadeh aplicado a funções não monótonas com dois parâmetros fuzzy.** 2009. 80p. Dissertação de Mestrado em Engenharia de Sistemas, Universidade Federal de Lavras, Lavras, 2012.

PEDRYCZ, W.; GOMIDE, F. **An introduction to fuzzy analysis and design.** New York: Massachusetts Institute of Technology, 1998.

PIRES, D.M. **Implementação computacional do princípio de extensão de Zadeh.** 2010. 59p. Dissertação de Mestrado em Engenharia de Sistemas, Universidade Federal de Lavras, 2010.

ROVEDA, J. A. F.; ROVEDA, S. R. M. M.; MARTINS, A. C. G. **Dinâmica populacional com sistemas p-fuzzy: um estudo de caso para a cidade de Sorocaba.** Biomatemática 19, p. 69-80, IMECC - Unicamp, ago., 2009.

SIMÕES, M. G.; SHAW, I. S. **Controle e modelagem *fuzzy***. São Paulo : Blucher, 186p., 2007.

TANAKA, K. **An Introduction to *fuzzy* logic for practical applications**. New York: Springer, 1997.

ZADEH, L. A. **Fuzzy sets**. Information and Control, Berkeley 8, p. 338-353, 1965.

APÊNDICE A – CÓDIGO DA CLASSE VERHULST

```

public class Verhulst {
    private float variavelFuzzy;
    private int pontosDiscretizar;
    private float limiteInferior;
    private float limiteSuperior;
    private float pertinenciaMaxima;
    private float delta;
    private double Delta;
    private double valorFuncao;
    private double vrFuncaoInf;
    private double vrFuncaoSup;
    private float popInicial;
    private float variavel;

    public float getVariavelFuzzy() { return variavelFuzzy; }
    public void setVariavelFuzzy(float variavelFuzzy) {
        this.variavelFuzzy = variavelFuzzy;
    }
    public int getPontosDiscretizar() { return pontosDiscretizar; }
    public void setPontosDiscretizar(int pontosDiscretizar) {
        this.pontosDiscretizar = pontosDiscretizar;
    }
    }
    public float getLimiteInferior() { return limiteInferior; }
    public void setLimiteInferior(float limiteInferior) {
        this.limiteInferior = limiteInferior;
    }
    }
    public float getLimiteSuperior() { return limiteSuperior; }
    public void setLimiteSuperior(float limiteSuperior) {
        this.limiteSuperior = limiteSuperior;
    }
    }
    public float getPertinenciaMaxima() { return pertinenciaMaxima; }
    public void setPertinenciaMaxima(float pertinenciaMaxima) {
        this.pertinenciaMaxima = pertinenciaMaxima;
    }
    }
    public float getdelta() { return delta; }
    public void setdelta(float delta) { this.delta = delta; }
    public double getDelta() { return Delta; }
    public void setDelta(double Delta) { this.Delta = Delta; }
    public double getValorFuncao() { return valorFuncao; }
    public void setValorFuncao(double valorFuncao) {
        this.valorFuncao = valorFuncao;
    }
    }
    public float getPopInicial() { return popInicial; }
}

```

```

public void setPopInicial(float popInicial) { this.popInicial = popInicial;}
public float getVariavel() { return variavel;}

public void setVariavel(float variavel) { this.variavel = variavel;}
public double getVrFuncaoInf() { return vrFuncaoInf;}
public void setVrFuncaoInf(double vrFuncaoInf) {
    this.vrFuncaoInf = vrFuncaoInf;
}
public double getVrFuncaoSup() { return vrFuncaoSup;}
public void setVrFuncaoSup(double vrFuncaoSup) {
    this.vrFuncaoSup = vrFuncaoSup;
}
public double CalculaValorFuncao (float n0, float k, float r, int t){
    return ((n0*k)/(n0+(k-n0)*Math.exp(-r*t)));
}
public double CalculaInversaK(double F, float n0, float r, int t){
    return (((F*n0)*(1-Math.exp(-r*t)))/(n0-F*Math.exp(-r*t)));
}
public double CalculaInversaR(double F, float n0, float k, int t){
    return (-((Math.log(n0)+Math.log(k-F)) -
        (Math.log(F)+Math.log(k-n0)))/t);
}
public double CalculaPertinencia(double ValorFuzzy,double xInf,double
xSup, double delta){
    double retorno;
    if (ValorFuzzy <= this.pertinenciaMaxima){
        retorno = (ValorFuzzy - xInf)/(delta);
        if (retorno < 0){
            retorno*=-1;
        }
    }else{
        retorno = (xSup - ValorFuzzy)/(delta);
        if (retorno < 0){
            retorno*=-1;
        }
    }
    return retorno;
}
public void SalvarArquivo(String DadosTxt)
throws HeadlessException {
    JFileChooser JanelaSalvarArquivo = new JFileChooser();
    JanelaSalvarArquivo.setFileSelectionMode
        (JFileChooser.FILES_ONLY);
    int result=JanelaSalvarArquivo.showSaveDialog(null);
    if(result == JFileChooser.APPROVE_OPTION){
        Writer writer = null;

```

```
try {
String text=DadosTxt;
File file=JanelaSalvarArquivo.getSelectedFile();
writer=new BufferedWriter(new FileWriter(file));
writer.write(text);
} catch(FileNotFoundException e){
} catch(IOException e){
} finally {
    try {
        if (writer != null){
            writer.close();
        }
    } catch(IOException e){
    }
}
}
}
}
```

APÊNDICE B – CÓDIGO DA CLASSE JFTELA

```

private void jBAplicarActionPerformed(java.awt.event.ActionEvent evt)
// Evento do Botao Aplicar Princ. de Ext.
FuncaoVerhulst novo = new FuncaoVerhulst();
novo.setLimiteInferior(Float.parseFloat(jTFLimInf.getText()));
novo.setPertinenciaMaxima(Float.parseFloat(jTFPertMax.getText()));
novo.setLimiteSuperior(Float.parseFloat(jTFLimSup.getText()));
novo.setPontosDiscretizar(Integer.parseInt(jTFPontos.getText()));
novo.setPopInicial(Float.parseFloat(jTFPopInicial.getText()));
novo.setVariavel(Float.parseFloat(jTFVariavel.getText()));
novo.setdelta(novo.getPertinenciaMaxima() -
               novo.getLimiteInferior());
NumberFormat df =NumberFormat.getCurrencyInstance(Locale.US);
((DecimalFormat)df).applyPattern("0.0000");
int Tempo=Integer.parseInt(jTFTempoFim.getText())
           - Integer.parseInt(jTFTempoIni.getText()) + 1;
int PontoDefuz = (int)(Tempo/2);
double[][] DadosFuzzy = new double[novo.getPontosDiscretizar()*
                                   Tempo][3];
double[][] DadosDefuzzificados = new double[Tempo][2];
if (jRBCapacidade.isSelected()){
float n0 = novo.getPopInicial();
float r = novo.getVariavel();
double F, xInf, xSup, delta;
int Tamanho;
Tamanho = ((Integer.parseInt(jTFTempoFim.getText()) -
                Integer.parseInt(jTFTempoIni.getText())+1) *
            novo.getPontosDiscretizar());
int Grupo = novo.getPontosDiscretizar();
xInf = novo.getLimiteInferior();
xSup = novo.getLimiteSuperior();
delta = novo.getdelta();
int Tini=Integer.parseInt(jTFTempoIni.getText());
double ValorFuncao;
int m = 0;
int n = 0;
int cont = 0;
while (m < Tamanho){
float k = novo.getLimiteInferior();
novo.setVrFuncaoInf(novo.CalculaValorFuncao
                    (n0,k,r,Tini));
k = novo.getLimiteSuperior();
novo.setVrFuncaoSup(novo.CalculaValorFuncao
                    (n0,k,r,Tini));

```

```

    novo.setDelta((novo.getVrFuncaoSup() -novo.getVrFuncaoInf()) /
                  (novo.getPontosDiscretizar()-1));
    ValorFuncao = novo.getVrFuncaoInf();
    double SomaFuncao = 0;
    double SomaPert = 0;
    for (int p=m;p<n+Grupo;p++){
        DadosFuzzy[m][1] = ValorFuncao;
        F = ValorFuncao;
        DadosFuzzy[m][0]= (novo.CalculaInversaK(F,n0,r,Tini));
        DadosFuzzy[m][2]=novo.CalculaPertinencia(DadosFuzzy[m][0],
                                                  xInf, xSup, delta);
        jTAResultado.append(Tini + " " + df.format(DadosFuzzy[m][1])+" "+
                           df.format(DadosFuzzy[m][2])+"\n");
        SomaFuncao+=(DadosFuzzy[m][1] *DadosFuzzy[m][2]);
        SomaPert+=DadosFuzzy[m][2];
        m++;
        ValorFuncao+=novo.getDelta();
    }
    DadosDefuzzificados[cont][0]=Tini;
    DadosDefuzzificados[cont][1] = (SomaFuncao/SomaPert);
    cont++;
    n = m;
    Tini++;
    jTAResultado.append("\n");
}
novo.SalvarArquivo(jTAResultado.getText());
jTAResultado.setText("");
for (int p=(novo.getPontosDiscretizar()*PontoDefuz);
     p<((novo.getPontosDiscretizar() *
        PontoDefuz)+novo.getPontosDiscretizar());
     p++){
    jTAResultado.append(df.format(DadosFuzzy[p][0]) + " " +
                       df.format(DadosFuzzy[p][1])+" "+
                       df.format(DadosFuzzy[p][2])+"\n");
}
novo.SalvarArquivo(jTAResultado.getText());
jTAResultado.setText("");
for (int p=0;p<Tempo;p++){
    jTAResultado.append(df.format(DadosDefuzzificados[p][0])
                       +" "+df.format(DadosDefuzzificados[p][1])+"\n");
}
novo.SalvarArquivo(jTAResultado.getText());
}
else if (jRBTaxa.isSelected()){
    float n0 = novo.getPopInicial();
    float k = novo.getVariavel();

```

```

double F, xInf, xSup, delta;
int Tamanho;
Tamanho=((Integer.parseInt(jTFTempoFim.getText()) -
            Integer.parseInt(jTFTempoIni.getText()+1) *
            novo.getPontosDiscretizar());
int Grupo = novo.getPontosDiscretizar();
xInf = novo.getLimiteInferior();
xSup = novo.getLimiteSuperior();
delta = novo.getdelta();
int Tini=Integer.parseInt(jTFTempoIni.getText());
double ValorFuncao;
int m = 0;
int n = 0;
int cont = 0;
while (m < Tamanho){
float r = novo.getLimiteInferior();
novo.setVrFuncaoInf(novo.CalculaValorFuncao
                    (n0,k,r,Tini));
r = novo.getLimiteSuperior();
novo.setVrFuncaoSup(novo.CalculaValorFuncao
                    (n0,k,r,Tini));
novo.setDelta((novo.getVrFuncaoSup() -
                novo.getVrFuncaoInf()) /
                (novo.getPontosDiscretizar()-1));
ValorFuncao = novo.getVrFuncaoInf();
double SomaFuncao = 0;
double SomaPert = 0;
for (int p=m;p<n+Grupo;p++){
    DadosFuzzy[m][1] = ValorFuncao;
    F = ValorFuncao;
    DadosFuzzy[m][0]=(novo.CalculaInversaR(F, n0, k, Tini));
    DadosFuzzy[m][2]=novo.CalculaPertinencia
    (DadosFuzzy[m][0],xInf, xSup, delta);
    jTAResultado.append(Tini + " " + df.format(DadosFuzzy[m][1])+" " +
        df.format(DadosFuzzy[m][2])+"\n");
    SomaFuncao += (DadosFuzzy[m][1]*DadosFuzzy[m][2]);
    SomaPert += DadosFuzzy[m][2];
    m++;
    ValorFuncao+=novo.getDelta();
}
DadosDefuzzificados[cont][0]=Tini;
DadosDefuzzificados[cont][1] = (SomaFuncao/SomaPert);
cont++;
n = m;
Tini++;
jTAResultado.append("\n");

```

```

    }
    novo.SalvarArquivo(jTAResultado.getText());
    jTAResultado.setText("");
    for (int p=(novo.getPontosDiscretizar()*PontoDefuz);
        p<((novo.getPontosDiscretizar()*PontoDefuz)+
            novo.getPontosDiscretizar());
        p++){
        jTAResultado.append(df.format(DadosFuzzy[p][0])+" "+
            df.format(DadosFuzzy[p][1])+" "+
            df.format(DadosFuzzy[p][2])+"\n");
    }
    novo.SalvarArquivo(jTAResultado.getText());
    jTAResultado.setText("");
    for (int p=0;p<Tempo;p++){
        jTAResultado.append
            (df.format(DadosDefuzzificados[p][0])+" "+
            df.format(DadosDefuzzificados[p][1])+"\n");
    }
    novo.SalvarArquivo(jTAResultado.getText());
    }
}

```

APÊNDICE C – IMAGEM DA INTERFACE GRÁFICA

