

UMA ABORDAGEM DE APRENDIZADO ASSISTIDO POR COMPUTADOR  
UTILIZANDO PROGRAMAÇÃO HÍBRIDA SIMBÓLICO-NUMÉRICA

Marcelo Pereira Pinto

---

DISSERTAÇÃO SUBMETIDA À COORDENAÇÃO DO CURSO DE PÓS-  
GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO, COMO REQUISITO  
PARCIAL PARA A OBTENÇÃO DO GRAU DE MESTRE PELA  
UNIVERSIDADE FEDERAL DO CEARÁ.

Fortaleza - CE, 2002

## RESUMO

Adota-se o emprego simultâneo de técnicas de programação simbólica declarativa e métodos da matemática computacional de características híbridas analítico-numéricas para o desenvolvimento de ambientes informatizados que facilitam o processo de aprendizado assistido por computador. É característico nos conteúdos de diversas disciplinas pertinentes às grandes áreas das ciências exatas do conhecimento como ciência da computação, engenharias, ciências físicas, medicina e matemática a existência de uma série de estágios associados à necessidade de construção do processo de aprendizado: conceituação fenomenológica, modelagem matemática das equações governantes, solução matemática, visualização dos resultados e interpretação fenomenológica. Num único ambiente informatizado, estes estágios podem ser disponibilizados e acessados de maneira simultânea e interativa, o que é concebido segundo uma metodologia sistemática de abordagem do conhecimento. Com finalidades pedagógicas, se modularizam os diversos estágios de construção do aprendizado citados acima, que por sua vez são ramificados segundo hierarquias de aumento de complexidade de conteúdos, permitindo a construção de um processo de aprendizado mais dinâmico, personalizado e estimulando a autonomia. As aplicações educacionais são discutidas e disponibilizadas na forma de *software*, implementados via *Mathematica®*. Dois problemas de áreas distintas do conhecimento são tratados: um de crescimento populacional e outro de modelagem avançada, no caso da transferência de calor não-linear. No primeiro, concebido dentro de um projeto internacional Brasil-Alema, denominado WAVES, parte-se da necessidade de disponibilizar, em um ambiente integrado, todos os estágios do conhecimento que permitam compreender e trabalhar o assunto abordado, como eventualmente estender o modelo matemático associado. No segundo, destaca-se a existência de complexas formulações matemáticas e a necessidade de se facilitar para o aprendiz associar fórmulas e conceituação fenomenológica pertinentes. Para tratamento e solução deste problema avançado utilizam-se os formalismos de um método analítico-numérico, denominado Técnica de Transformada Integral Generalizada. Recorre-se à utilização de tabelas, gráficos e animações para favorecer o processo de aprendizado.

## ABSTRACT

It is adopted the simultaneous use of declarative symbolic programming techniques and computational mathematics methods with hybrid analytic-numerical characteristics for the development of environments that facilitate the computer aided learning process. It is familiar in the set of contents of several pertinent disciplines related with the well-known areas of the exact sciences of the knowledge as computer science, engineering, physical sciences, medicine and mathematics the existence of a amount of steps associated to the need of the learning construction process: Establishment of phenomenological concepts, mathematical modeling of the governing equations, mathematical solution, visualization of results and phenomenological interpretation. These steps can be ready for use and for interactive access in a same environment. This is designed according to a systematic methodology of knowledge approach. For pedagogical purposes, the several steps of learning construction above mentioned are arranged according sets of cells which are ramified in a tree form taking in account hierarchies of increasing complexity from the set of contents, allowing the construction of a more dynamic learning process, personalized and stimulating the autonomy. As educational applications, two problems from different areas are discussed and available as a computer program that is implemented in the software *Mathematica*®. The first one deals with population migration and the other with advanced modeling, focusing a non-linear heat transfer problem. The first one was defined in a Brazilian-German international project named WAVES, with the goal of became available all the steps of the knowledge to use in an integrated environment that allows to understand the approached subject and eventually to extend the associated mathematical model. In the second problem, it is pointed out the existence of complex mathematical formulations and the need of larger reasoning abstraction to associate formulas and phenomenological concepts. In order to solve this advanced mathematical problem it was chosen the formalisms of a analytic-numerical method, named as Generalized Integral Transform Technique. It is shown the use of tables, graphs and animation for highlighting the learning process.

## **AGRADECIMENTOS**

Ao Professor, Orientador e amigo Júlio Wilson Ribeiro;

Ao Professor, Co-orientador e amigo João Batista Furlan Duarte;

Aos colegas do Mestrado em Ciências da Computação, Laboratório do Mestrado em Ciências da Computação (MCC), Laboratório de Inteligência Artificial (LIA), Laboratório de Métodos Computacionais Híbridos (OMNI) e Departamento de Computação;

Aos financiadores da minha bolsa CNPq / Projeto Waves;

À minha eterna musa e companheira Ilana e aos amigos: Adams, Anderson, Jamile, Joselmar, Juarez, Natália, Ricardo, Cândida;

E à minha família: Sarah, Ivan, Joneide, Marcos, Mônica e Marcel.

# SUMÁRIO

1	Introdução.....	1
<b>1.1</b>	<b>Revisão Bibliográfica.....</b>	<b>2</b>
1.1.1	Aprendizado Assistido por Computador .....	2
1.1.2	Computação Híbrida.....	4
1.1.2.1	Programação Simbólica Declarativa .....	5
1.1.2.2	Métodos Analítico-Numéricos .....	6
1.1.2.2.1	A Técnica de Transformada Integral Generalizada.....	6
2	Abordagem do Aprendizado Assistido por Computador .....	8
<b>2.1</b>	<b>Tratamento Unificado do Conhecimento .....</b>	<b>9</b>
<b>2.2</b>	<b>Análise da Crescimento Populacional .....</b>	<b>11</b>
2.2.1	Projeto Waves.....	11
2.2.2	Apresentação do <i>Notebook</i> .....	12
3	Modelagem Avançada.....	18
<b>3.1</b>	<b>Tratamento Generalizado do Problema Parabólico .....</b>	<b>18</b>
<b>3.2</b>	<b>Aplicação: Transferência de Calor Não-Linear .....</b>	<b>22</b>
4	Resultados e Discussões.....	28
<b>4.1</b>	<b>Migração Populacional .....</b>	<b>28</b>
<b>4.2</b>	<b>Transferência de Calor Não-Linear.....</b>	<b>30</b>
5	Conclusões e Sugestões.....	34
	APÊNDICE A.....	35
	APÊNDICE B.....	36

## LISTA DE FIGURAS

FIGURA 2.1: VISÃO GERAL DA ABORDAGEM DO APRENDIZADO ASSISTIDO POR COMPUTADOR SEGUNDO O TRATAMENTO UNIFICADO DO CONHECIMENTO. ....	8
FIGURA 2.2: ESTÁGIOS DE CONSTRUÇÃO DO PROCESSO DE APRENDIZADO ASSISTIDO POR COMPUTADOR SEGUNDO O TRATAMENTO UNIFICADO DO CONHECIMENTO. ....	10
FIGURA 2.3: VISÃO GERAL DOS MÓDULOS DO NOTEBOOK PARA A ANÁLISE DE CRESCIMENTO POPULACIONAL.....	13
FIGURA 2.4: TRECHO DA CONCEITUAÇÃO TEXTUAL E MATEMÁTICA DO PROBLEMA DE CRESCIMENTO POPULACIONAL DE MALTHUSIAN GERADO A PARTIR DA ABERTURA DA PRIMEIRA CÉLULA DA JANELA PRINCIPAL, DISPONÍVEL NA FIGURA 2.3. ....	14
FIGURA 2.5: TRECHO DE IMPLEMENTAÇÃO DO PROBLEMA DE MALTHUSIAN GERADO A PARTIR DA ABERTURA DA CÉLULA IMPLEMENTAÇÃO DA JANELA PRINCIPAL, DISPONÍVEL NA FIGURA 2.3. ....	15
FIGURA 2.6: TRECHOS OBTIDOS A PARTIR DA ABERTURA DA TERCEIRA CÉLULA DA JANELA PRINCIPAL, DISPONÍVEL NA FIGURA 2.3. GRÁFICOS GERADOS A PARTIR (A) DA FUNÇÃO DE CRESCIMENTO POPULACIONAL COM TAXA DE MIGRAÇÃO CONSTANTE, (B) FUNÇÃO DE CRESCIMENTO POPULACIONAL COM TAXA DE MIGRAÇÃO VARIANDO COM O TEMPO. ....	16
FIGURA 2.7:GRÁFICO DE CRESCIMENTO POPULACIONAL TEMPORAL GERADO A PARTIR DA INTERPOLAÇÃO DOS DADOS COLETADOS DA POPULAÇÃO DOS ESTADOS DO CEARÁ E PIAUÍ ENTRE OS ANOS DE 1872 E 1998 (EM MILHÕES). ....	17
FIGURA 3.1 REPRESENTAÇÃO ESQUEMÁTICA DA PLACA FINA PARA O PROBLEMA UNIDIMENSIONAL DE TRANSFERÊNCIA DE CALOR NÃO-LINEAR POR CONDUÇÃO PURA. ....	23
FIGURA 4.1 GRÁFICOS DO CRESCIMENTO POPULACIONAL DURANTE 3 DIAS, COM TAXA DE CRESCIMENTO CONSTANTE $K = 0.5$ E TAXA DE MIGRAÇÃO CONSTANTE, PARA VALORES DE POPULAÇÃO INICIAL $P_0 = \{10.000, 20.000, 30.000\}$ . ....	28
FIGURA 4.2 GRÁFICOS DO CRESCIMENTO POPULACIONAL DURANTE 3 ANOS, COM TAXA DE CRESCIMENTO CONSTANTE, $K = 0.5$ E TAXA DE MIGRAÇÃO EM FUNÇÃO DO TEMPO SEGUNDO EQUAÇÃO 2.2, PARA VALORES DE POPULAÇÃO INICIAL $P_0 = 10.000$ , $A = 0,015$ E $C = 6.000$ ....	29
FIGURA 4.3 GRÁFICO DE CRESCIMENTO POPULACIONAL TEMPORAL GERADO A PARTIR DA INTERPOLAÇÃO DOS DADOS COLETADOS DA POPULAÇÃO DOS ESTADOS DO CEARÁ E PIAUÍ ENTRE OS ANOS DE 1872 E 1998 (EM MILHÕES). ....	30
FIGURA 4.4 PERFIL DE DISTRIBUIÇÃO DE TEMPERATURA, $T(X,T)$ PARA O PROBLEMA DE CONDUÇÃO DE CALOR NÃO-LINEAR ( $T = 0,1$ S). ....	32
FIGURA 4.5 PERFIL DE DISTRIBUIÇÃO DE TEMPERATURA, $T(X,T)$ PARA O PROBLEMA DE CONDUÇÃO DE CALOR NÃO-LINEAR ( $T = 0,3$ S). ....	32

FIGURA 4.6 PERFIL DE DISTRIBUIÇÃO DE TEMPERATURA, $T(X,T)$ PARA O PROBLEMA DE CONDUÇÃO DE CALOR NÃO-LINEAR ( $T = 0,01S, 0,2S, 0,4S, 0,6S$ ). .....	32
FIGURA 4.7 VISUALIZAÇÃO EM 3D DE DISTRIBUIÇÃO DE TEMPERATURA, $T(X,T)$ PARA O PROBLEMA DE CONDUÇÃO DE CALOR NÃO-LINEAR, COM O TEMPO VARIANDO DE 0,0S A 6,0S.....	33

## ÍNDICE DE TABELAS

TABELA 4.1 CURVAS DE DISTRIBUIÇÃO DE TEMPERATURA, $T(X,T)$ , PARA O PROBLEMA NÃO-LINEAR DE CONDUÇÃO DE CALOR. PARA A COORDENADA TEMPORAL, $T$ , OS VALORES ATRIBUÍDOS VARIAM DE 0,1 S A 0,9 S, COM INTERVALO DE 0,1 S. PARA A COORDENADA ESPACIAL, $X$ , VARIA DE 0,0 CM A 1,0 CM.....	31
--	----



## NOMENCLATURA

$P$	Taxa de variação populacional (2.1).
$k$	Taxa de crescimento populacional (2.1).
$r$	Taxa de migração populacional constante (2.1).
$t$	Coordenada temporal, definida na equação (2.1).
$P(t)$	Taxa de variação populacional em função do tempo (2.2).
$r(t)$	Taxa de migração variável em função do tempo (2.2).
$w(\mathbf{x})$	Coefficiente capacitivo, definido na equação definido na equação (3.1.a).
$T(\mathbf{x},t)$	Potencial, definido na equação (3.1.a).
$P(\mathbf{x},t)$	Termo fonte ou sumidouro, definido na equação (3.1.a).
$t$	Coordenada temporal, definida na equação (3.1.a).
$\mathbf{x}$	Sistema de coordenadas espaciais, definidos na equação (3.1.a).
$f(\mathbf{x})$	Distribuição inicial do potencial, definida na equação (3.1.b).
$d(\mathbf{x})$	Termo convectivo, definido na equação (3.1.d).
$K(\mathbf{x})$	Coefficiente de difusividade, definido na equação (3.1.d).
$A_i(t)$	Coefficiente da série de expansão, definido na equação (3.2).
$N_j$	Norma, definida na equação (3.6).
$\bar{T}_i(t)$	Potencial transformado, definido na equação (3.7.a).
$\bar{g}_i(t)$	Termo fonte transformado, definido na equação (3.10.b).
$\bar{f}_i$	Valor inicial do potencial transformado, definido na equação (3.13.a).

## SÍMBOLOS GREGOS

- $\phi(\mathbf{x},t)$  Termo fonte da equação da condição de contorno, definido na equação (3.1.c).
- $\alpha(\mathbf{x})$  Coeficiente da condição de contorno, definido na equação (3.1.e).
- $\beta(\mathbf{x})$  Coeficiente da condição de contorno, definido na equação (3.1.e).
- $\psi(\mu_i, \mathbf{x})$  I-ésima autofunção do problema auxiliar, definida na equação (3.2).
- $\mu_i^2$  I-ésimo autovalor de problema auxiliar, definido na equação (3.2.a).

## SUBSCRITOS

- $i, j$  Ordem das auto-quantidades ou índice de linha e coluna, respectivamente.
- $P$  Solução de regime permanente (estado estacionário).
- $H$  Relativo ao problema homogêneo.

# 1 Introdução

Em meados do século passado, com o aparecimento dos primeiros computadores nos grandes centros de pesquisa, havia um restrito número de máquinas por usuário, portanto, o acesso era muito disputado e limitado. Utilizavam-se então ambientes de programação onde as linguagens possuíam estrutura e sintaxe muito distantes das linguagens naturais e o conhecimento praticamente se encontrava disponível apenas em edições impressas em papel. Estes fatores criavam uma realidade: o homem trabalhava para a máquina. Para poder construir, implementar algoritmos e interpretar a informação processada, o programador precisava adaptar o raciocínio para desenvolver suas tarefas nas ferramentas disponíveis [01].

Atualmente, se caracteriza um processo inverso. Com a popularização do acesso ao computador e se considerando o contínuo aumento na capacidade de armazenamento de dados, velocidade de processamento e recursos das técnicas de programação, passou-se a desenvolver abordagens computacionais que melhor assistissem ao raciocínio humano, fazendo com que a máquina trabalhasse cada vez mais para o homem. Isso caracterizava uma busca por novos recursos computacionais visando dinamizar as interfaces homem-máquina, de forma que a informação pudesse ser mais rapidamente trabalhada e assimilada com o auxílio do computador.

Mesmo atualmente, o estudo de modelos matemáticos relativos a problemas oriundos das grandes áreas do conhecimento associado às ciências exatas, citando-se medicina, ciência da computação, física e engenharias, continua a ter a fase de modelagem computacional sem um uso mais racional e abrangente de modernos recursos e formalismos da ciência da computação [02-10]. Por exemplo, para solução de sistemas de equações diferenciais parciais acopladas predomina o uso de linguagens procedurais e métodos puramente numéricos [09,11,12]. Como consequência, a implementação de algoritmos e sua utilização a posteriori podem significar um grande esforço a ser despendido pelo elemento humano, chegando-se, em casos críticos, a não se atingir objetivos esperados [13-15].

Dentre os modelos tratados nessas grandes áreas, citam-se: problemas avançados em ciências de fluidos térmicas [02-05,09,13-16], crescimento tumoral [17], ondas sonoras [18], aplicação do método de Runge-Kutta em problemas de elastoplasticidade [19] e análise de circuitos eletrônicos não-lineares [20]. Nestes tipos de problemas observa-se uma necessidade de tratamento pedagógico e simultâneo de conceituação física, modelagem matemática e interpretação de resultados [21,22]. Neste contexto, didaticamente, é necessário associar, de modo interativo e facilitador, conceitos físicos a matemáticos avançados. Isto caracteriza uma crescente necessidade para se melhor viabilizar os esforços associados à implementação de algoritmos nas áreas tecnológicas e afins.

Torna-se, portanto estratégico desenvolver, simultaneamente nos aspectos físicos, matemáticos e computacionais, novos formalismos e técnicas para facilitar a solução de problemas e interpretação de resultados obtidos, através do doravante denominado Aprendizado Assistido por Computador, AAC [04,09,10,11,12,23-26]. Uma proposta que oferece novas possibilidades nesse ramo constitui a utilização conjugada de computação simbólica, técnicas de programação baseada em regras,

programação funcional e métodos não clássicos provindos da matemática, como as técnicas ditas analítico-numéricas, para melhorar o processo de raciocínio humano assistido por computador, traduzido através da computação híbrida [04,09,10, 23-26].

Neste cenário, aborda-se o uso destes recursos para se conceber uma abordagem sistemática de tratamento e análise unificada, com características interativas, que é disponibilizada num mesmo arquivo eletrônico e que simultaneamente permite: aprendizado de conceitos teóricos, manipulação analítico-numérica de fórmulas matemáticas, técnicas de solução e análise fenomenológica, concebida com auxílio de recursos gráficos e animação.

Para fins de aplicação opta-se pela escolha de dois casos: um problema real e um caso acadêmico. No primeiro, aborda-se um problema específico de modelagem de crescimento populacional para ilustração da abordagem proposta no processo de aprendizado assistido por computador. Este foi concebido a partir de um projeto internacional Brasil/Alemanha, denominado WAVES [27] e que está em andamento nos últimos seis anos. Neste, a UFC constitui o principal parceiro brasileiro e financiou uma bolsa de estudos para execução desta dissertação.

No segundo caso, se optou por um problema avançado, proveniente das áreas de ciências físicas e engenharias, abordando-se a transferência de calor em problemas não-lineares, o que envolve, além dos estágios críticos do processo de aprendizado de conceitos físicos associados, o domínio de uma técnica de solução de equações diferenciais, no caso a Técnica de Transformada Integral Generalizada, (Generalized Integral Transform Technique, GITT) [02,05,09,10]. Para tanto, como será mostrado posteriormente, serão utilizados, com maior ênfase, recursos visuais de caráter didático, como a animação gráfica, que simula para o aprendiz a dinâmica temporal do fenômeno de transferência de calor em uma barra, o que facilita a concepção física do efeito da difusão de calor.

Nos sub-capítulos seguintes é apresentada a revisão bibliográfica, abordando o AAC, a computação híbrida e os métodos analítico-numéricos.

O capítulo dois reporta-se à metodologia de abordagem do processo de aprendizagem assistida por computador, destacando-se a aplicação ao problema de crescimento populacional. No capítulo três, direciona-se esta metodologia para o problema de modelagem avançada, enfocando a transferência de calor em regime transiente. Resultados e discussão são apresentados no capítulo quatro. O capítulo consecutivo reporta-se às conclusões e sugestões.

## **1.1 Revisão Bibliográfica**

### **1.1.1 Aprendizado Assistido por Computador**

Conforme defende a literatura ocidental, os primeiros softwares educacionais a surgirem datam da década de sessenta. Nesta época, estes se restringiam a ambientes de pesquisa e caracterizavam-se basicamente na relação estímulo-resposta. As duas décadas seguintes trouxeram grandes modificações na concepção, elaboração e

utilização desses programas, com a incorporação de novos avanços alcançados principalmente nas áreas de psicologia, pedagogia e computação. Como por exemplo: estágios de desenvolvimento de Piaget, novas alternativas de ensino-aprendizagem, os conceitos de inteligência artificial [28], computação simbólica e mais recentemente a internet [29,30,31].

Essas novas realidades impuseram novos ritmos e dimensões na atividade educacional, fazendo com que as técnicas tradicionais de ensino-aprendizagem, baseadas unicamente no livro didático e no quadro-negro, cada vez mais cedessem lugar a técnicas mais atualizadas, que incluem o computador como ferramenta indispensável. Nestas novas tecnologias de aprendizado, o livro passa a ser substituído por discos ópticos (CD-ROM) e o quadro-negro, pela projeção na parede da imagem gerada pelo computador (quadro eletrônico). O computador assume a característica de um recurso pedagógico privilegiado e o software educativo o seu material didático [09,11,12]. Dentre os recentes tipos de software educativo podemos citar: simuladores, jogos, tutoriais, sistemas especialistas e outros [29].

Dentro dessa nova perspectiva, Moura [30,31] destaca a classificação do denominado ensino tecnológico profissionalizante, que difere do ensino convencional e aponta algumas características do primeiro: o uso intensivo de laboratórios experimentais, o uso do computador como importante ferramenta de auxílio na elaboração de projetos, valorização da exploração dos sentidos humanos, a utilização de métodos diferenciados de avaliação e colocação do conhecimento em prática, via estágios profissionalizantes.

Uma interessante proposta de apoio ao ensino tecnológico que se deve investir consiste no desenvolvimento de novas alternativas via AAC. Nesta ótica, destaca-se uma menção à experiência desenvolvida em universidades búlgaras, onde há mais de trinta anos, já se desenvolviam metodologias visando incorporar avanços da ciência da computação e matemática aplicada, para melhorar a arte da programação e da informática educativa. Desde cedo, foi criado um doutorado em matemática aplicada e computação, um centro de treinamento de professores de ensino médio, com abrangência nacional. Houve uma florescente produção científica voltada principalmente ao ensino e desenvolvimento científico e tecnológico [09-12,32].

Nas duas últimas décadas, o aparecimento e melhoria das linguagens de programação simbólicas, trouxeram um substancial impulso ao desenvolvimento do AAC. Adicionalmente, houve substancial contribuição de técnicas de programação funcional e programação baseada em regras [4,6-10,33,34].

Esses sistemas incorporaram uma variedade de recursos de programação com potencial aplicação na educação, tornando possível preparar documentos mais interativos, cujo impacto extrapola um ambiente de aprendizado tradicional. Assim, o processo de aprendizado pode ser construído a partir de arquivos eletrônicos, que permitem trabalhar: texto didático, inclusive utilizando hipertexto, a manipulação analítico-simbólica das equações associadas e a solução de problemas físicos.

A seguir são citadas algumas ferramentas educacionais que podem ser incorporadas na construção do aprendizado assistido por computador [04,09-11,25,26].

Recursos de hipertexto: Durante a leitura e manuseio de grande quantidade de informações, as referências podem ser facilmente cruzadas, permitindo saltos entre estas sem a necessidade de seguir uma seqüência de leitura enfadonha e única [09-11,25].

Recursos de multimídia: permitem que se armazenem informações visuais e sonoras dentro de um mesmo documento de leitura [29,31].

Recursos de computação simbólica: disponibilizam a solução analítica e numérica de um grande número de problemas, destacando-se os que exigem intensa manipulação analítica. Exemplos: cálculo integral, simplificações de expressões analíticas, operações sobre matrizes, etc. [04,15,16,23,24,35].

Recursos de computação numérica: processam a solução numérica de inúmeros problemas matemáticos que modelam fenômenos e processos do mundo real. Potenciam larga aplicação em estatística, economia, física, medicina, engenharias, matemática, entre outros [36,37,38,39].

Recursos de computação gráfica: Tornam possível uma análise qualitativa a partir de informações quantitativas. Permitem a representação visual de dados numéricos, funções e expressões matemáticas complexas, através de tabelas, gráficos bidimensionais e tridimensionais, e animações [06,07].

E finalmente a internet: Os hipertextos também podem ser utilizados como hyperlinks da atual rede mundial de computadores, o que aumenta as possibilidades em outras áreas educacionais, especialmente na educação à distância [29,30,31].

### **1.1.2 Computação Híbrida**

Na década de sessenta do século passado, durante o período da corrida espacial, visando minimizar o esforço empregado na elaboração de cálculos, dois grandes blocos econômicos investiam pesado em pesquisa. Enquanto os Estados Unidos e Europa concentravam-se no desenvolvimento de métodos denominados puramente numéricos (diferenças finitas e elementos finitos), que requisitavam um crescente esforço computacional, os países do Leste Europeu e Rússia concentravam bastante esforço em sistematizar extensas manipulações analíticas, realizando um grande avanço no desenvolvimento de métodos com características mais analíticas [04,09-12]

Ainda neste período, da necessidade de se atribuir à máquina a cansativa tarefa de manipular algebricamente extensas expressões matemáticas, nasce a computação simbólica, o que tornou ainda mais poderoso e atrativo o desenvolvimento e utilização de métodos com características mais analíticas [40,41]. Já nesse tempo foram registradas inúmeras publicações, em livros e jornais especializados do bloco oriental, apresentando métodos computacionais analítico-numéricos para aplicação ao cálculo diferencial e integral, com diversas aplicações industriais. Infelizmente quase toda esta rica literatura, escrita em russo e ucraniano, não se tornou conhecida no ocidente.

A partir da década de setenta, inúmeros cientistas do Leste Europeu migraram para o ocidente em função do declínio econômico de seus países. Com isto, pesquisadores de ambos os blocos se uniram, buscando criar novas alternativas de tratamento e solução de equações diferenciais, através de metodologias sistemáticas e o desenvolvimento da computação simbólica.

Principalmente a partir da década de oitenta, os novos avanços computacionais, tal como grande aumento da capacidade de processamento e de armazenamento de dados proporcionaram uma ampliação nos sistemas de computação simbólica [04,07,08,34,40,42]. Esses passaram a ser programas de aplicação mais geral, apresentando novos recursos de programação simbólica, numérica, gráfica e trazendo

ambientes de desenvolvimento próprios, voltados principalmente à utilização de programação declarativa.

A utilização desses ambientes de programação simbólico declarativa, tais como o *Mathematica*®, abriu uma nova tendência na criação de códigos na utilização dos recentes avanços em formalismos matemáticos com características mais analíticas, como a abordagem analítico-numérica provida pela GITT. Surge assim a computação híbrida, como uma ferramenta resultante do uso conjunto da programação simbólica declarativa e métodos analítico-numéricos [03-12,15,23-25,35].

Esta conjunção permitiu a implementação de algoritmos matemáticos que guardam importantes informações analíticas associadas às equações diferenciais parciais e permite utilizar recursos da análise matemática para otimizar a convergência numérica dos algoritmos. As linhas de comando, representadas através de regras e funções, transferem ao computador a tarefa de manipulação analítica, o que não é executável se usando linguagens de programação procedurais, citando-se o Fortran ou C [45]. Esta última característica fica mais crítica quando se pretende tornar mais atrativa a programação avançada.

Destaca-se que o uso de linguagens de programação procedurais conjugadas a métodos ditos puramente numéricos não permite levar para os algoritmos implementados informações da matemática avançada, o que pode ser traduzido em não se proceder a um controle eficaz do processo de convergência e mesmo estabilidade numérica.

### **1.1.2.1 Programação Simbólica Declarativa**

A computação simbólica permite automatizar os processos computacionais relacionados à solução de problemas matemáticos, enfatizando a computação discreta sobre símbolos representando objetos matemáticos. Estes podem ser números, polinômios, radicais, funções trigonométricas, grupos, vetores, tensores e muitos outros [40].

Historicamente, a computação simbólica surgiu da necessidade de se atribuir à máquina a cansativa tarefa de manipular algebricamente extensas expressões matemáticas, a fim de permitir aos interessados o estudo e análise de modelos cada vez mais complexos [23]. Tipicamente, os processos a serem automatizados são operações de aritmética, álgebra e cálculo avançado.

As primeiras referências documentadas de programas manipulando símbolos por computador datam de 1953, segundo H. G. Kahrmanian [41]. A partir daí, surgem programas computacionais capazes de manipular polinômios, resolver equações e calcular derivadas de funções. Em 1966 houve as duas primeiras conferências sobre cálculo simbólico, ocorridas em Washington e Pisa. Na década de 70, surgiram programas computacionais que integravam funções analiticamente e outros para a solução simbólica de equações diferenciais e integrais [40].

Na década de oitenta surgem os primeiros grandes e complexos programas de aplicação mais geral de processamento simbólico. Eles já integravam uma série de recursos de computação numérica e gráfica ao seu poder de processamento simbólico, mas ficavam restritos ainda a computadores de grande porte, devido as suas extensas necessidades de memória. Novos avanços computacionais das duas últimas décadas,



como a crescente evolução dos microcomputadores, proporcionaram uma ampliação e popularização desses sistemas.

Os mais recentes ambientes de programação simbólica propiciam o desenvolvimento interativo de programação através de diferentes paradigmas com o uso conjunto de modernos recursos da ciência da computação para finalidades educacionais, citando-se: animações, multimídia, hipertexto, internet [06-08,22,33,42]. Há um vasto campo de aplicações em três grandes linhas, caracterizadas na pesquisa, educação e indústria [41], onde se destaca o uso predominante dos seguintes sistemas: *Maple*, *Mathematica*, *Matlab* e *Mathcad*.

O *Mathematica*® particularmente conta com um grande número de funções matemáticas já implementadas, como: diferenciação, integração, cálculo de limites, cálculo vetorial, representação em séries de Tailor, entre outras. Novas funções matemáticas podem ser definidas a partir das operações e funções já disponíveis através de um ambiente de programação.

### **1.1.2.2 Métodos Analítico-Numéricos**

Do ponto-de-vista da matemática computacional, os métodos provindos da matemática para solução de equações diferenciais podem ser definidos como puramente numéricos ou híbridos analítico-numéricos [04].

Os métodos analítico-numéricos permitem a construção e implementação de algoritmos com características fortemente analíticas, conseguindo-se maior abrangência quando se adotam ambientes de programação simbólica declarativa.

Entre as principais classes de métodos analítico-numéricos amplamente utilizados para desenvolvimento de programas aplicados nas grandes áreas das ciências exatas mencionadas anteriormente, destacam-se:

- Quadratura gaussiana [36,37,44], que permite a integração numérica.
- Transformadas Integrais [45] (Método Espectral, Método de Colocação Ortogonal e GITT), que visam à solução de inúmeras classes de problemas matemáticos, com destaque para as equações diferenciais parciais.
- Transformada de Laplace generalizada [46-49], também para solução de equações diferenciais parciais.

De particular interesse para a concepção de algoritmos matemáticos através da computação híbrida é o uso da GITT, devido suas características, que serão posteriormente discutidas e que permitem resolver vários tipos de problemas avançados de interesse das ciências físicas e engenharias.

#### ***1.1.2.2.1 A Técnica de Transformada Integral Generalizada***

Os princípios da Técnica de Transformada Integral surgiram na década de setenta, de uma generalização da teoria clássica de separação de variáveis, segundo os trabalhos de Özisik e Murray [50]. O primeiro livro sobre o assunto, publicado em 1984 por Mikailov e Özisik [02], apresenta os formalismos do Método de Transformada

Integral Clássica (CITT - Classic Integral Transform Technique) aplicados em problemas de difusão de calor e massa, que foram divididos em sete classes.

Já na década de oitenta, apareceram vários artigos na literatura especializada sobre a aplicação da CITT [51-63], agora abordando outros problemas que antes eram resolvidos apenas por métodos puramente numéricos como: diferenças finitas, elementos finitos e volumes finitos, ressaltando algumas vantagens observadas:

- Não utilização de malhas, fator que se torna crítico em problemas multidimensionais.
- Menores taxas de convergência numérica.
- Redução de tempo de processamento, pois este se concentrava apenas em uma única variável (temporal), portanto problema unidimensional.
- Soluções Benchmark.

O segundo livro sobre o método, publicado em 1993 por Cotta [03] apresentou uma extensão dos formalismos clássicos para aplicação em problemas não-lineares e propôs a utilização de novos mecanismos para acelerar a convergência numérica como: Técnicas de filtragem, ordenamento de espectro de autovalores, balanços integrais e métodos adaptativos, dando origem à GITT.

Ainda na década de noventa o método ganha um adicional suporte com a utilização da programação simbólica nos formalismos analíticos. O aparecimento dos ambientes simbólicos de programação trouxe um substancial impulso no desenvolvimento e utilização de GITT no tratamento analítico, solução numérica e visualização gráfica de sistemas de equações diferenciais parciais não lineares acopladas.

Os formalismos de GITT baseiam-se numa metodologia sistemática que parte da escolha de um problema auxiliar de autovalor, que encerra informações relativas aos operadores do problema original nas variáveis espaciais que serão eliminadas. Desenvolvem-se pares de operadores, que são aplicados ao problema original, efetuando-se a transformada integral. O sistema transformado resulta num sistema de equações diferenciais ordinárias (ODE) infinito, não-linear, que é truncado de acordo com a precisão prescrita desejada e resolvido analítica ou numericamente com controle de erro, através de bibliotecas de sub-rotinas científicas. Finalmente, os potenciais originais são construídos a partir da utilização dos operadores de inversão sobre os resultados do sistema transformado.

As aplicações atuais de GITT em ciência e tecnologia podem ser resumidamente agrupadas nos seguintes grupos [03,07,08,10-16,64]:

- a) Problemas com coeficientes variáveis nas equações.
- b) Problemas com coeficientes variáveis no contorno.
- c) Problemas com contornos variáveis.
- d) Problemas que envolvam problemas auxiliares de difícil solução.
- e) Problemas não lineares, citando-se como aplicações típicas a solução de equações de camada limite, Navier Stokes e problemas não-lineares de secagem.

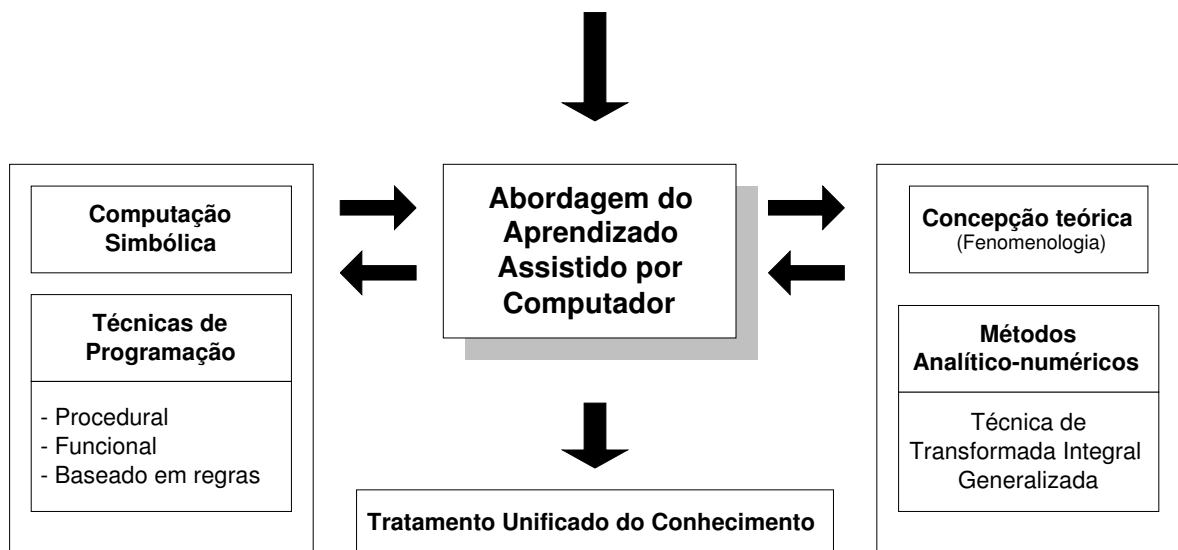
No Capítulo seguinte é discutida a metodologia de abordagem do processo de aprendizado assistido por computador, fazendo-se uma aplicação a um problema de crescimento populacional.

## 2 Abordagem do Aprendizado Assistido por Computador

A idéia básica que será tratada a seguir trata da análise e caracterização de um processo de aprendizado assistido por computador, mais especificamente falando, que possa lidar com conteúdos pertinentes a disciplinas das grandes áreas de conhecimento anteriormente caracterizadas.

Nesta direção, é necessário destacar como pré-requisito para construir esta forma de aprendizado que o aprendiz possua uma formação mínima em ciência da computação, mas entende-se que experiência pessoal anterior em lidar com processos de construção e implementação algorítmica certamente atuará como elemento facilitador nesta proposta de aprendizado.

Na Figura 2.1 exibida a seguir, ilustra-se a sistematização básica das ferramentas e formalismos utilizados para promover a abordagem do aprendizado assistido por computador, segundo uma tentativa de tratamento unificado de disciplinas das grandes áreas de conhecimento. Numa análise preliminar, para que o raciocínio humano promova o processo de aprendizado, a abordagem do conhecimento assistido por computador passa simultaneamente a se estruturar segundo dois grandes blocos, caracterizados nos campos laterais, mais a esquerda e mais a direita, da Figura 2.1.



**Figura 2.1:** Visão geral da abordagem do aprendizado assistido por computador segundo o tratamento unificado do conhecimento.

Um primeiro elemento questionado, a experiência em programação, que é apresentado no lado esquerdo superior da citada figura, caracteriza-se primordialmente pelo uso de computação simbólica e técnicas de programação declarativa. Este estágio de formação favorece o uso de técnicas de aprendizado com auxílio do computador [09-11]. Torna-se possível definir conjuntos de regras, segundo expressões e funções escritas na notação matemática tradicional, que automaticamente manipulam procedimentos analíticos e outras estruturas matemáticas avançadas [04,06-12,23-26].

Como grande ganho, tarefas são repassadas ao computador com o intuito de sistematizar o processo, diminuindo significativamente os esforços em operações repetitivas e mecânicas para refazer manualmente cálculos associados à manipulação de extensas expressões matemáticas.

Outro papel relevante na proposta em discussão é expresso pelo contato eletrônico com modelos matemáticos mais avançados, o que é realizado pelo computador via a interpretação automatizada de regras inteligentes. Um exemplo destes poderosos recursos pode ser visto nas seguintes tarefas: aplicar a regra da cadeia para obter a derivação analítica de um conjunto de funções analíticas e, em qualquer momento desejado, se introduzir facilmente no ambiente de aprendizagem novas funções e reaplicar a regra da cadeia. Este grau de liberdade promove o aprendizado autônomo.

O segundo grande bloco da Figura 2.1 é traduzido pela experiência pessoal, mostrado no seu lado direito, que expressa o grau de conhecimento anterior específico em fenomenologia física (o que mede o nível de capacidade em abstração no raciocínio), formação matemática mínima e cursos de métodos numéricos, para que se atinjam níveis de aprendizado previamente planejados.

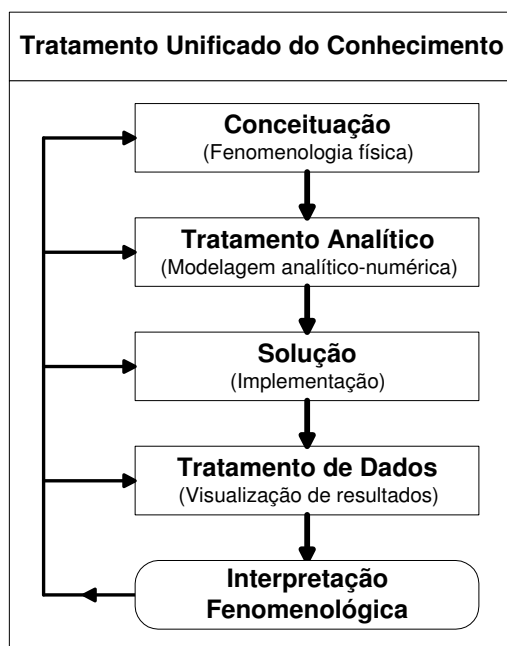
Em situação crítica, quanto maior a complexidade do conteúdo do conhecimento a ser abordado, maior pré-requisito constituirá a necessidade de uma formação humana mais específica. Esta realidade caracteriza o aparecimento da modelagem dita avançada, quando o uso marcante de métodos numéricos se torna necessário para se atingir o estado da solução numérica, o que precede os estágios finais do processo de aprendizado. Nestes tipos de problemas geralmente se lida com modelos matemáticos expressos por equações diferenciais parciais não-lineares, passando os métodos analítico-numéricos a assumirem um papel promissor na computação híbrida anteriormente discutida.

A seguir, no subitem 2.1, se discute como os estágios do processo de construção do aprendizado assistido por computador são definidos no tratamento unificado do conhecimento.

## ***2.1 Tratamento Unificado do Conhecimento***

Para a construção do aprendizado em conteúdos programáticos, relatou-se no item 1.1.1 a necessidade de simultaneamente se trabalhar: conceituação, formulação matemática e interpretação física e da aplicação dos avanços computacionais nessa realidade. No contexto da abordagem apresentada, essa característica é realizada no denominado Tratamento Unificado do Conhecimento, representado pelo bloco inferior central da Figura 2.1. Esse tratamento de construção, criado através da utilização das ferramentas e formalismos apresentados na Figura 2.1, cobre todas as fases de pesquisa e desenvolvimento de forma interativa e cíclica e visa ser geral para as grandes áreas do conhecimento científico.

A seguir, o Tratamento Unificado do Conhecimento é introduzido visualmente nos seus vários estágios e características na Figura 2.2. Esse esboço visual também pode ser visto como um detalhamento do bloco único que representava o mesmo tratamento na Figura 2.1.



**Figura 2.2: Estágios de Construção do Processo de Aprendizado Assistido por Computador Segundo o Tratamento Unificado do Conhecimento.**

Em caráter de construção temporal e hierárquico do processo de aprendizado, estruturam-se didaticamente os seguintes estágios que definem a navegação no ambiente de aprendizado: Conceituação, que expressa a necessidade de incorporar conceitos e formalismos teóricos. Tratamento Analítico envolve a concepção e manipulação passo-a-passo do modelo analítico que descreve quantitativamente o problema matemático associado. Nesta fase, eventualmente pode ocorrer a necessidade de utilização de conceitos e leis providas da matemática e do problema físico. Solução, este estágio prevê o uso de métodos providos da matemática computacional e objetivam se obter a solução do problema matemático em discussão, geralmente expresso por equações diferenciais. Tratamento de Dados, obtido a partir de arquivos de dados numéricos, gerados a partir do item anterior, e que podem ser concebidos e visualizados nas formas de tabelas, gráficos e animações. Interpretação Fenomenológica, que constitui um poderoso mecanismo de construção do aprendizado, que, de forma combinada com os demais estágios já apresentados, pode interagir com qualquer dos inúmeros estados da aprendizagem anteriormente construídos pelo aprendiz. Ressaltam-se aqui os recursos de animação, que podem reproduzir virtualmente a dinâmica associada ao fenômeno físico. Estes estágios doravante serão denominados em conjunto, por **estágios de construção do processo de aprendizado assistido por computador**.

Do ponto-de-vista do processo de construção, é possível se conceber num único arquivo eletrônico todos os estágios descritos acima, que podem ser acessadas segundo uma crescente complexidade de interpretação dos conteúdos programáticos estruturados. Os conteúdos programáticos específicos dos diversos estágios, por sua vez, são alocados em sub-células [04,07-10,25], segundo um grau de hierarquização de crescente complexidade de interpretação destes. Independentemente como são dispostas no documento, as células podem ser navegadas de modo personalizado, atingindo os ciclos definidos na Figura 2.2, até que o aprendiz vença todas as etapas da construção do aprendizado.

Durante o processo de navegação, gera-se um ciclo dinâmico, já que vários dos estágios podem ser acessados a qualquer momento e em diferentes combinações, o que formaliza um conjunto de modos de abstração do raciocínio, de modo a reforçar o aprendizado autônomo na construção do processo de aprendizagem. Em síntese, o processo de aprendizado fica mais dinâmico, ao se acessar os conteúdos de aprendizado construídos e trabalhados nas diversas formas eletrônicas mencionadas. Além do mais, é possível a criação on-line de blocos didáticos complementares a partir dos conteúdos previamente disponibilizados, enaltecendo o processo de aprendizado autônomo.

O caráter dinâmico em se acessar quaisquer dos estágios do processo do aprendizado segundo módulos discutidas acima no tratamento unificado do conhecimento, EEC, ficam reforçados pelos recursos de programação existentes no *Mathematica@*, que permitem modularizar a construção do algoritmo segundo a disposição em células e sub-células. Assim, torna-se fácil e rápido navegar e interagir nos EEC, através de um sistema de *links*, com característica de hierarquia entre as células. Como recursos adicionais, nos estados associados à interpretação fenomenológica, tabelas, gráficos e recursos de animação gráfica são disponibilizados ao longo da navegação no ambiente de aprendizagem.

Objetivando caracterizar aspectos operacionais de ambientes de ensino/aprendizado da metodologia discutida, apresenta-se na sessão a seguir um problema de crescimento populacional, que exige do aprendiz formação de terceiro grau completo. E no capítulo seguinte, um segundo ambiente informatizado de aprendizagem, voltado à matemática computacional avançada, temática associada a Cursos de Pós-Graduação em áreas de ciências exatas e engenharias.

## **2.2 Análise do Crescimento Populacional**

### **2.2.1 Projeto Waves**

O projeto WAVES (*Water Availability, Vulnerability of Ecosystems and Society in Northeastern in Brazil*) insere-se num conjunto de projetos idealizados por pesquisadores alemães, envolve vários países cooperantes e está direcionado ao estudo da climatologia global, abordando vários aspectos interdisciplinares: impactos sociais, econômicos, migração populacional, meteorologia, entre outros. Especificamente, o projeto bilateral Brasil-Alemanha, WAVES, foi concebido com o objetivo principal de estudar e buscar soluções para problemáticas na região Nordeste brasileira, partindo da modelagem integrada dos elementos relevantes na cadeia causal (clima, água, solo, políticas de preservação do meio-ambiente, vegetação, população, entre outros fatores). Busca-se, através de generalizações estabelecidas a partir do estudo destes fatores isolados e de seus correlacionamentos, gerar um pacote computacional que possa ser utilizado por especialistas para contribuir no processo de estabelecimento de políticas de planejamento regional, nacional e internacional.

Assim, os esforços específicos por área de conhecimento despendidos no projeto e implementação algorítmica dos sub-códigos associados à modelagem do projeto podem ser caracterizados como muito vastos e diversificados. Por um lado, áreas de conhecimentos com tradições muito diferentes no processo de modelagem estão sendo reunidas para responder às questões de integração colocadas pelo projeto. Por outro

lado, isto implica em um risco de que sejam gerados modelos complicados demais para serem gerenciados e compreendidos por especialistas de áreas específicas.

Para minimizar este risco, uma abordagem empregada no projeto foi ressaltar a integração entre os sub-códigos implementados a partir de subáreas específicas do conhecimento. Nesta fase, destaca-se o papel fundamental exercido pelos gerentes principais do projeto WAVES.

Especificamente este projeto de dissertação de Mestrado, constitui uma proposta de cooperação dentro do projeto bilateral WAVES/CNPq e objetiva contribuir na pesquisa no problema de crescimento populacional. Foi intitulado formalmente por “Análise do Problema de Migração e Modelagem Integrada Utilizando Computação Simbólica”. Para tanto e visando-se oferecer relevantes contribuições na configuração do modelo integrado, aplicaram-se ferramentas e metodologias, de natureza matemática e computacional, estabelecidas no Capítulo 1, nos itens 1.1.1 e 1.1.2, com especial destaque para o aprendizado assistido por computador e a computação híbrida.

Um dos objetivos específicos desse projeto constitui se estudar e propor a modelagem do fenômeno de migração populacional. O modelo matemático foi concebido sob orientação do lado Alemão, através do Prof. Maarten Krol, do PIK, Potsdam Institut for Climate Impact Reserch, que detinha o conhecimento de caráter mais global da fundamentação teórica associado às grandes áreas constituintes do projeto WAVES. Em síntese, se desenvolveu a abordagem sistemática, interativa e pedagógica do problema da migração populacional, concebida num ambiente informatizado de aprendizagem.

Constatou-se que documentos desenvolvidos anteriormente em projetos associados à modelagem integrada pelo lado alemão utilizavam preponderantemente linguagens de programação procedural, com ênfase para o Fortran. Portanto, os códigos implementados, guardavam informação de computação puramente numérica, não permitindo ao usuário um acesso mais genérico a outros tipos de informação.

Desde já se chama atenção que a metodologia de interação código-usuário pode ser perfeitamente estendida para elaboração de outros sub-códigos do projeto WAVES como também para eventualmente reforçar o processo de integração entre os diversos sub-códigos do modelo geral.

### **2.2.2 Apresentação do Notebook**

Cronologicamente o lado alemão sugeriu a leitura de literatura referente aos problemas de modelagem integrada e migração populacional [65-70]. Com finalidade acadêmica, foi selecionado o modelo disponibilizado em Gray et al. [42], quando foi concebida a proposta para a implementação de um algoritmo, visando posteriores utilizações em estudos de migração populacional nas regiões do Ceará e Piauí.

A presente proposta de implementação enfoca basicamente o tratamento unificado do crescimento populacional segundo Thomas Robert Malthus [42]. A partir da análise quantitativa do comportamento do crescimento populacional efetuado através dos resultados disponibilizados via este modelo é possível se estudar o fenômeno de migração populacional. Os módulos foram implementados a partir do conceito de *notebook* e célula. As abordagens, conceitual e matemática, são didaticamente apresentadas através de textos, fórmulas e *hyperlinks*. Em seguida, técnicas de

programação funcional e baseada em regras foram utilizadas no ambiente de programação simbólica *Mathematica*® para se proceder a um tratamento analítico-simbólico do modelo, onde finalmente rotinas disponibilizam a análise e discussão crítica dos resultados obtidos.

De acordo com Thomas Malthusian, a taxa de crescimento populacional no tempo é diretamente proporcional ao tamanho atual da população. Logo o modelo assume a equação 2.1 como a seguir:

$$\frac{\partial P}{\partial t} = kP - r \quad (2.1)$$

onde,  $\frac{\partial P}{\partial t}$  é a taxa de variação da população no tempo,  $k$  é chamada de taxa de crescimento e  $r$  é a taxa de migração constante. A seguir é mostrada na Figura 2.3 uma visão geral do *notebook*, disponibilizando um contato preliminar com o conjunto de informações.

Numa visão unificada, a figura permite observar quatro tópicos gerais modularizados em células, que representam as etapas principais (primeira hierarquia) e compõem todo o processo de aprendizado a ser trabalhado nas hierarquias subseqüentes e detalhado a seguir.

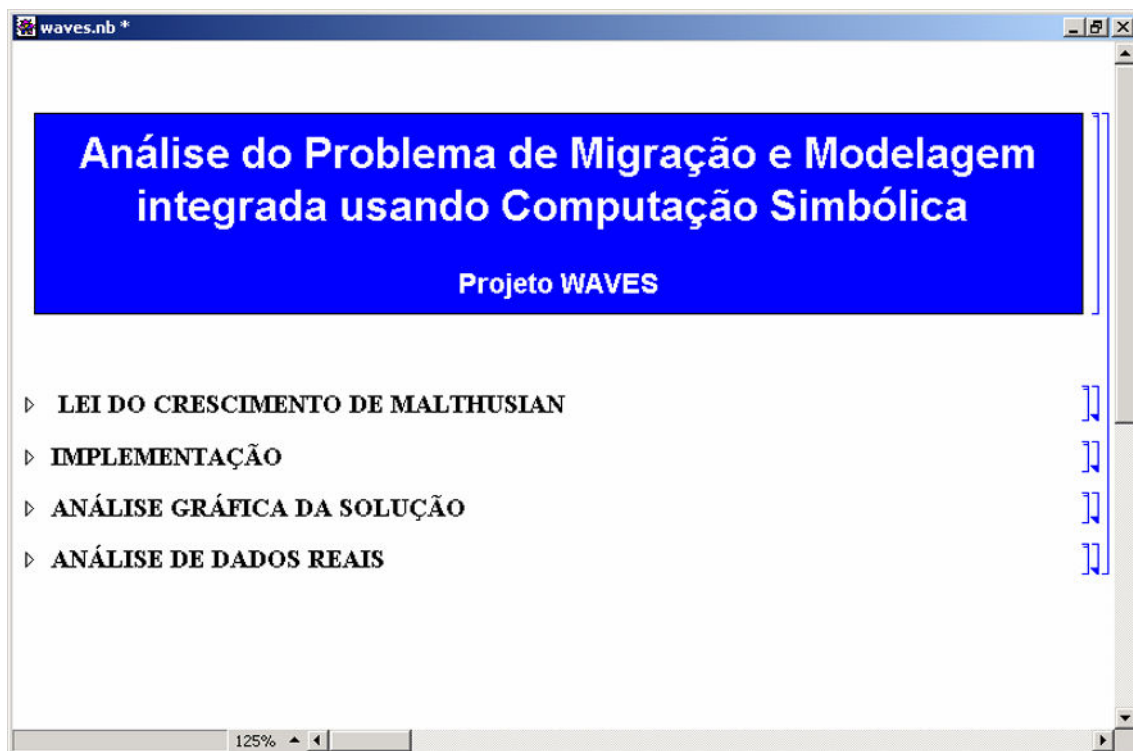
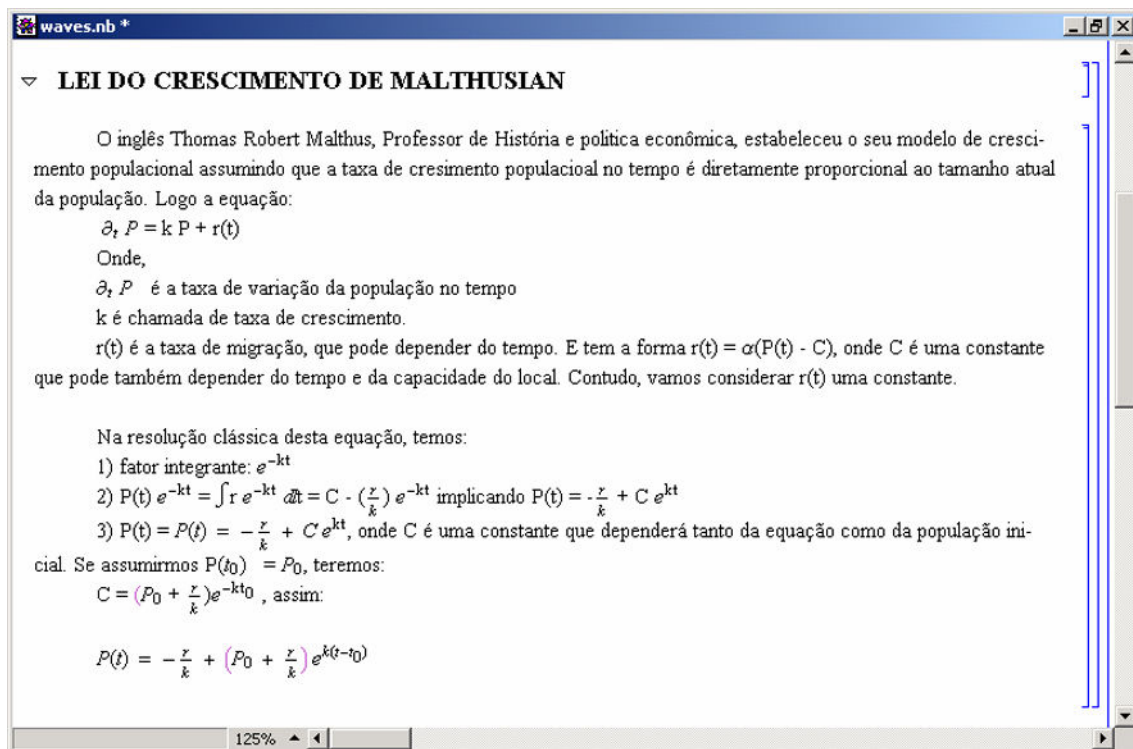


Figura 2.3: Visão geral dos módulos do notebook para a análise de crescimento populacional.

Ainda na mesma figura, indicadores à esquerda dos módulos caracterizam o acesso à primeira classe hierárquica. Os quatro módulos podem ser facilmente acessados a qualquer momento com apenas um clique de "mouse" sobre os mesmos.

A seguir, na Figura 2.4, se mostra a primeira célula aberta, que se reporta ao primeiro estágio:





**Figura 2.4:** Trecho da conceituação textual e matemática do problema de crescimento populacional de Malthusian gerado a partir da abertura da primeira célula da janela principal, disponível na Figura 2.3.

Esta célula foi concebida para se disponibilizar visualmente a conceituação textual e matemática do problema de crescimento populacional. Nesta etapa, há possibilidade de utilização de recursos de hipertexto. É possível se ter acesso adicional, se desejado, através de palavras-chave, a outros módulos do mesmo documento, outros *notebooks* ou até mesmo a outras fontes de informação via internet.

Também se observam as bordas localizadas nas laterais direitas de cada célula, que possibilitam o acesso a diferentes níveis de hierarquia, permitindo que a informação seja trabalhada variando o nível de detalhamento. Assim, cada tópico pode ser facilmente subdividido seguindo as mesmas leis de hierarquias e recursos computacionais de navegação.

Assim, utilizando os recursos descritos, o aprendiz tem o primeiro contato textual e matemático com o problema de crescimento de Malthusian, na Figura 2.4, onde são definidos, na forma de texto, os diversos parâmetros e o problema são manipulados até chegar à resolução supondo a taxa de migração constante.

A próxima célula a ser aberta na Figura 2.3, constitui a Implementação. Neste módulo, apresenta-se a solução matemática encontrada e discutida para o problema de crescimento, segundo a Figura 2.4 que expressa a taxa de variação populacional em função do tempo,  $P(t)$ , de acordo o modelo de Malthusian.

Apresenta-se também outro modelo, partindo-se de generalizações matemáticas do anterior, supondo-se agora que a taxa de migração  $r(t)$  é considerada variável, dependente do tempo, como mostrado abaixo nas equações (2.2) e (2.3),

$$\frac{\partial P}{\partial t} = kP - r(t) \tag{2.2}$$

$$r(t) = \alpha(P(t) - C) \tag{2.3}$$

Neste estágio, o programa interpreta simbolicamente as fórmulas analíticas, isto é, tem a capacidade de entender os conceitos matemáticos reportados nas diversas fórmulas. Também interpreta e aplica conceitos e propriedades matemáticas, como a derivação e exponenciação.

Depois de assimilar o desenvolvimento matemático do problema focado, torna-se agora possível acessar, alterar e processar o código: Todas variáveis e fórmulas contidas nas linhas do código estão disponíveis para modificações, que são interpretadas pelo *Mathematica*®, o qual, interativamente, fornece os resultados ou possíveis comentários inteligentes relativos a eventuais erros cometidos. A Figura 2.5 exibe a célula denominada Implementação aberta.

```

waves.nb *
IMPLEMENTAÇÃO

Implementação simbólica para r = taxa de migração(constante)

In[1]= DSolve[{P'[t] == k P[t] - r, P[0] == P0}, P[t], t]

Out[1]= {{P[t] -> (r + e^(k t) k (P0 - r/k)) / k}}

Implementação simbólica para s = taxa de migração em função do Tempo

In[2]= s[t] := alpha (P[t] - c)

In[3]= DSolve[{P'[t] == k P[t] - s[t], P[0] == P0}, P[t], t]

Out[3]= {{P[t] -> (c alpha - e^(k t) k (P0 + c/(k-alpha)) + e^(k t) alpha (P0 + c/(k-alpha))) / (-k + alpha)}}

Resolução melhorada através de recursos de manipulação algébrica:

In[4]= DSolve[{P'[t] == k P[t] - s[t], P[0] == P0}, P[t], t][[1, 1, 2]] //
FullSimplify // Apart

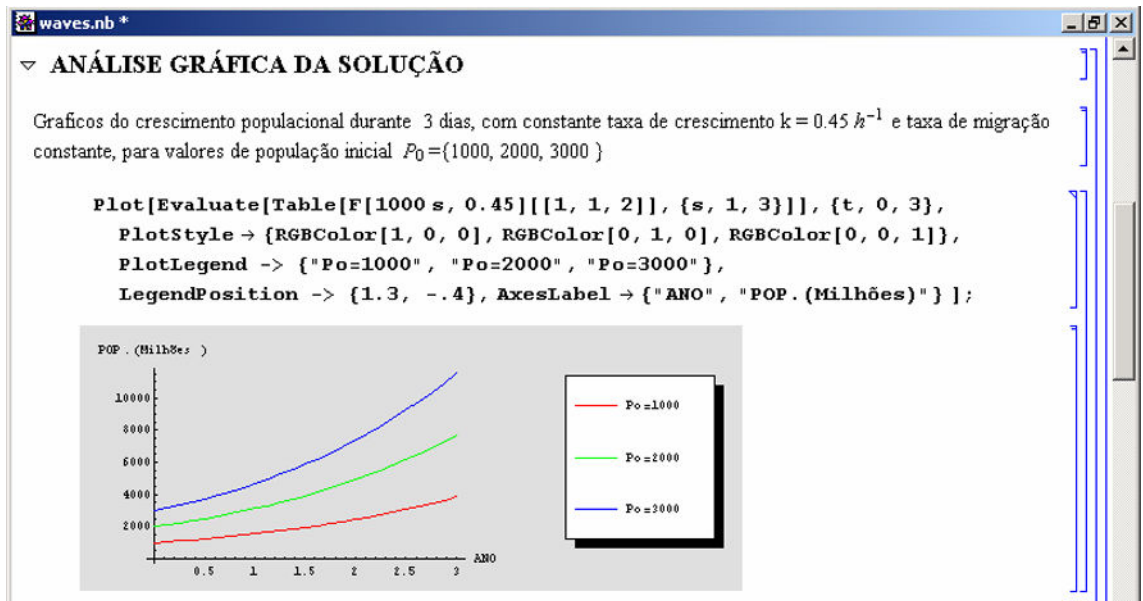
Out[4]= -c alpha / (k - alpha) + e^(k t) (k P0 + c alpha - P0 alpha) / (k - alpha)

```

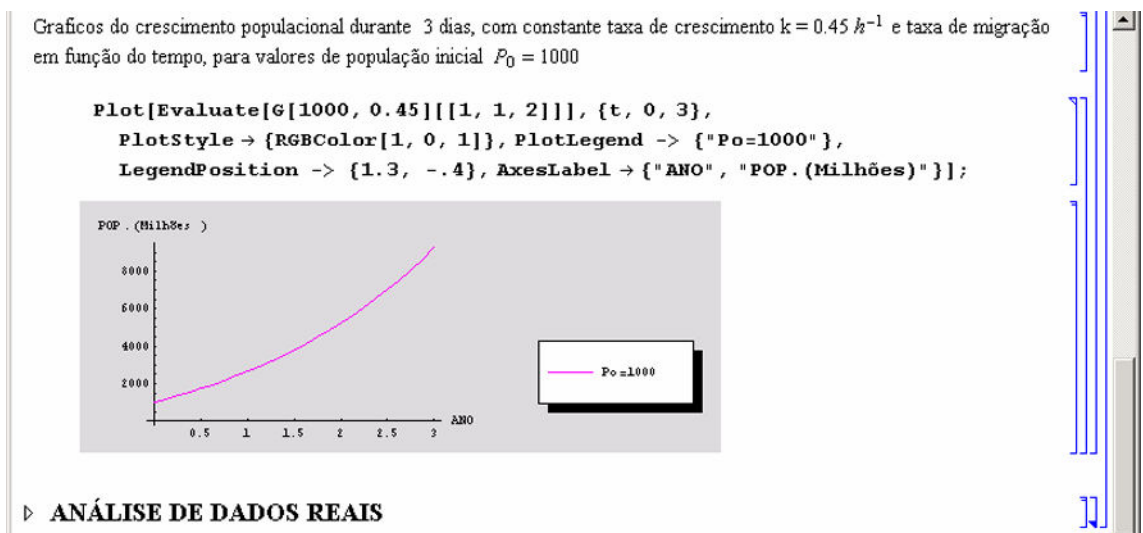
Figura 2.5: Trecho de implementação do problema de Malthusian gerado a partir da abertura da célula Implementação da janela principal, disponível na Figura 2.3.

Como não há EDP's presentes no modelo matemático, não foi necessário a utilização de métodos analítico-numéricos, como a GITT. Uma função interna do *Mathematica*®, a *DSolve*, foi habilitada para resolver a equação diferencial do problema de crescimento populacional. Depois de obtida a solução é possível expressá-la em uma forma mais simplificada, utilizando comandos de manipulação analítica, no caso *FullSimplify* e *Apart*.

Pedagogicamente, dispõe-se de um ambiente de estudo onde se podem alterar as informações originais e excursionar no mundo da modelagem matemática, para conceber variantes do problema de crescimento. Através das soluções analíticas anteriores, geram-se as funções F e G para obtenção de gráficos que fazem parte da próxima célula, mostrada nas Figuras 2.6(a) e 2.6(b).



(a)



(b)

**Figura 2.6:** Trechos obtidos a partir da abertura da terceira célula da janela principal, disponível na Figura 2.3. Gráficos gerados a partir (a) da função de crescimento populacional com taxa de migração constante, (b) função de crescimento populacional com taxa de migração variando com o tempo.

Esta etapa corresponde ao tópico Análise Gráfica das Soluções na hierarquia principal. Torna-se possível agora interpretar o comportamento das curvas de crescimento populacional e repassá-las à equipe do projeto integrado, para discussão em grupo. Essa nova fase reporta-se à visualização dos resultados, quando novamente se utilizam linhas de comando extremamente compactadas, que permitem representar em inúmeras tarefas associadas ao processo de definição e geração dos gráficos, como mostrados nas Figuras 2.6 (a) e 2.6 (b).

Alternativamente, é possível modificar as equações originais, referentes aos modelos de crescimento populacional e comodamente reprocessar o programa,

reutilizando automaticamente as novas definições das funções F e G que serão disponibilizadas, na geração dos novos gráficos.

As últimas células ilustram o uso didático de tabelas de dados e uma nova possibilidade de definição de função. Para atender a requisitos do projeto WAVES, foi concebido um quarto e último módulo na Figura 2.3, chamado Análise Gráfica de Dados Reais, apresentada na Figura 2.7, onde uma interpolação de dados reais é utilizada para gerar o terceiro e último gráfico, expresso na figura 2.7.

Analogamente é possível se modificar os dados de entrada de população e desenvolver uma nova análise do modelo real, podendo também compará-la com a modelagem trabalhada, especificamente aqui, com o modelo de *Malthusian* e sua versão analítica estendida.

Os resultados numéricos do problema acima enfocado são apresentados e discutidos no item 4.1 do Capítulo 4.

Numa proposta mais complexa, pode-se ainda ampliar os formalismos apresentados nesse capítulo para aplicação em outros modelos físico-matemático-computacionais, geralmente representados por sistemas de equações diferenciais parciais, o que é ilustrado no capítulo seguinte, quando é abordada uma aplicação em modelagem avançada, representado por um problema de transferência de calor não-linear.

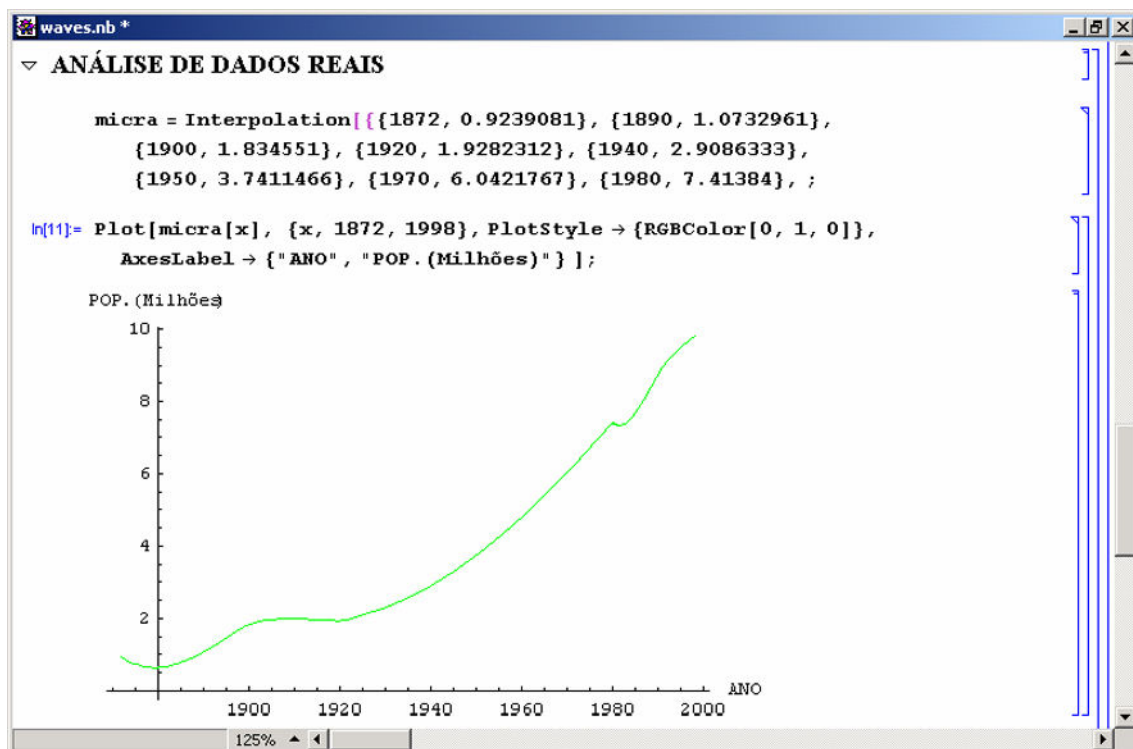


Figura 2.7:Gráfico de crescimento populacional temporal gerado a partir da interpolação dos dados coletados da população dos estados do Ceará e Piauí entre os anos de 1872 e 1998 (em milhões).

### 3 Modelagem Avançada

Uma interessante área a ser melhor explorada na informática educativa consiste no desenvolvimento de recursos e métodos para facilitar a abordagem e o processo de aprendizado de problemas avançados nos aspectos físicos, matemáticos e computacionais. Um dos maiores desafios neste campo se concentra no universo de modelos expressos por sistemas de equações diferenciais acopladas. Neste contexto, para promover o processo de aprendizado assistido por computador, pedagogicamente é necessário se associar, de modo interativo e simultâneo, diversos tipos de conceitos, o que requisita maior concentração e abstração físico-matemática, para poder correlacionar: conceituação fenomenológica, desenvolvimento de derivações analíticas associadas provenientes do cálculo avançado, e simulação assistida por computador do fenômeno físico associado.

As ferramentas de aprendizado assistido por computador, comentadas nos capítulos anteriores constituem uma promissora plataforma para facilitar o aprendizado do cálculo numérico e se resolver equações diferenciais. Visando ampliar estes recursos quando se deseja obter soluções de equações diferenciais, a presente Dissertação incorpora o método da GITT, que permite se estender à abordagem e tratamento unificados apresentados no Capítulo 2, para a construção e implementação de algoritmos para problemas pedagogicamente considerados mais difíceis.

Busca-se assim construir um algoritmo para tratamento analítico e solução numérica de um problema não-linear de transferência de calor, com o objetivo de ilustrar a abrangência e atratividade destes recursos no desenvolvimento da modelagem avançada. No próximo sub-item se apresentam os formalismos do método da GITT.

#### 3.1 Tratamento Generalizado do Problema Parabólico

Matematicamente é possível se definir uma formulação genérica que englobe inúmeros problemas provenientes das áreas de medicina, matemática computacional, engenharias e ciências físicas, visando um tratamento unificado e solução generalizada. Uma classe deste porte pode ser concebida sob a forma de um problema parabólico do tipo difusivo-convectivo [02-05,09,10,13-16], a seguir definido numa região finita  $V$ , com superfície de contorno  $S$ ,

$$w_k(\mathbf{x}) \frac{\partial T_k(\mathbf{x}, t)}{\partial t} = -L_k T_k(\mathbf{x}, t) + P_k(\mathbf{x}, t, T_1(\mathbf{x}, t), T_2(\mathbf{x}, t)), \quad \mathbf{x} \in V, \quad t > 0 \quad (3.1.a)$$

com condição inicial,

$$T(\mathbf{x}, t) = f(\mathbf{x}), \quad \mathbf{x} \in V, \quad t = 0 \quad (3.1.b)$$

e condições de contorno:

$$BT(\mathbf{x}, t) = \phi(\mathbf{x}, t), \quad \mathbf{x} \in S, \quad t > 0 \quad (3.1.c)$$

Os operadores lineares  $L$  e  $B$  do problema governante são definidos expressões

$$L \equiv -\nabla \cdot K(\mathbf{x})\nabla + d(\mathbf{x}) \quad (3.1.d)$$

$$B \equiv \alpha(\mathbf{x}) + \beta(\mathbf{x})K(\mathbf{x}) \frac{\partial}{\partial \mathbf{n}} \quad (3.1.e)$$

Nas fórmulas acima  $T(\mathbf{x},t)$  representa o potencial a ser obtido (temperatura, concentração de células cancerígenas, etc.);  $w(\mathbf{x})$ ,  $K(\mathbf{x})$  e  $d(\mathbf{x})$  expressam os coeficientes da equação (3.1.a e 3.1d) cujo termo não homogêneo  $P(\mathbf{x},t)$  assume o efeito de uma fonte ou sumidouro. Os coeficientes prescritos  $\alpha(\mathbf{x})$  e  $\beta(\mathbf{x})$  são da equação de contorno (3.1.c), cujo termo  $\phi(\mathbf{x},t)$ , se não nulo, expressa as informações de contorno não-homogêneas;  $\partial/\partial \mathbf{n}$  corresponde à derivada na direção normal e aponta para a direção exterior à superfície de contorno  $S$ . As variáveis independentes,  $\mathbf{x}$  e  $t$ , representam respectivamente as coordenadas espacial e temporal.

Como princípio básico se supõe a representação do potencial  $T(\mathbf{x},t)$  expressa através de uma expansão de autofunções, da forma,

$$T(\mathbf{x},t) = \sum_{i=1}^{\infty} A_i(t) \psi(\mu_i, \mathbf{x}) \quad (3.2)$$

As autofunções  $\psi(\mu_i, \mathbf{x})$  são obtidas a partir da adoção do seguinte problema auxiliar associado, do tipo *Sturm-Liouville*, no qual se conserva o operador  $L$ , definido no problema original, equação (3.1.a),

$$\mu_i^2 w(\mathbf{x}) \psi(\mu_i, \mathbf{x}) = L \psi(\mu_i, \mathbf{x}), \quad \mathbf{x} \in V \quad (3.3.a)$$

Possuindo as seguintes condições de contorno,

$$B \psi(\mu_i, \mathbf{x}) = 0, \quad \mathbf{x} \in S \quad (3.3.b)$$

Cujos termos guardam informações contidas nos operadores das equações (3.1.a) e (3.1.c), porém desprezando-se os termos não homogêneos  $P(\mathbf{x},t)$  e  $\phi(\mathbf{x},t)$ .

O problema auxiliar assim concebido, por ser problema de autovalor do tipo *Sturm-Liouville*, possui as seguintes propriedades:

a) os autovalores  $\mu_i$  são todos reais e positivos, sendo ordenáveis decrescentemente,

$$\mu_1 > \mu_2 > \mu_3 > \mu_4 \dots \quad (3.4.a)$$

b) as autofunções  $\psi(\mu_i, \mathbf{x})$  associadas obedecem à relação de ortogonalidade

$$\int_V w(\mathbf{x}) \psi(\mu_i, \mathbf{x}) \psi(\mu_j, \mathbf{x}) dv = 0, \quad i \neq j \quad (3.4.b)$$

Os coeficientes de expansão,  $A_i(t)$ , presentes na equação (3.2), são matematicamente obtidos, aplicando-se o operador  $\int_V w(\mathbf{x}) \psi(\mu_j, \mathbf{x}) dv$  nesta mesma equação, como mostrado a seguir:

$$\int_V w(\mathbf{x}) \psi(\mu_j, \mathbf{x}) T(\mathbf{x},t) dv = \sum_{i=1}^{\infty} A_i(t) \int_V w(\mathbf{x}) \psi(\mu_j, \mathbf{x}) \psi(\mu_i, \mathbf{x}) dv \quad (3.5.a)$$

Em seguida, aplicando-se o princípio de ortogonalidade das autofunções, ex em (3.4.b), na equação anterior, explicita-se  $A_i(t)$ ,

$$A_j(t) = \frac{1}{N_j} \int_V w(\mathbf{x}) \psi(\mu_j, \mathbf{x}) T(\mathbf{x},t) dv \quad (3.5.b)$$

sem perda de generalidade  $N_j$  constitui a integral de normalização, ou simplesmente norma e definida por,

$$N_j = \int_V w(\mathbf{x})[\psi(\mu_j, \mathbf{x})]^2 dv \quad (3.6)$$

As equações (3.2) e (3.5.b) permitem agora se definir o par de transform integral, o qual é consolidado pelas fórmulas de transformada e de inversa:

Fórmula de transformada,

$$\bar{T}_i(t) = \frac{1}{N_i^{1/2}} \int_V w(\mathbf{x})\psi(\mu_i, \mathbf{x})T(\mathbf{x}, t)dv \quad (3.7.a)$$

Fórmula de inversa,

$$T(\mathbf{x}, t) = \sum_{i=1}^{\infty} \frac{1}{N_i^{1/2}} \psi(\mu_i, \mathbf{x})\bar{T}_i(t) \quad (3.7.b)$$

Num passo seguinte se procede à transformação do problema de EDP, exp pelas equações (2.1.a-e), que elimina a dependência das coordenadas espaciais e permite se obter um sistema ordinário, ou sistema de EDO ou ainda problema de valor inicial. Os potenciais transformados,  $\bar{T}_i(t)$ , são calculados pela solução do sistema diferencial transformado conforme o procedimento apresentado a seguir,

Inicialmente se aplica o operador  $\frac{1}{N_i^{1/2}} \int_V \psi(\mu_i, \mathbf{x})dv$  sobre a equação (3.1.a), resultando,

$$\frac{d\bar{T}_i(t)}{dt} = \frac{1}{N_i^{1/2}} \int_V \psi(\mu_i, \mathbf{x})[\nabla \cdot K(\mathbf{x})\nabla T(\mathbf{x}, t) - d(\mathbf{x})T(\mathbf{x}, t) + P(\mathbf{x}, t)]dv \quad (3.8)$$

Analogamente, aplica-se o operador  $\frac{1}{N_i^{1/2}} \int_V T(\mathbf{x}, t)dv$  sobre a equação (3.3.a), resultando:

$$-\mu_i^2 \bar{T}_i(t) = \frac{1}{N_i^{1/2}} \int_V T(\mathbf{x}, t)[\nabla \cdot K(\mathbf{x})\nabla \psi(\mu_i, \mathbf{x}) - d(\mathbf{x})\psi(\mu_i, \mathbf{x})]dv \quad (3.9)$$

Adicionando-se, membro a membro, as equações (3.8) e (3.9), rearranjan encontra-se o seguinte sistema diferencial ordinário desacoplado:

$$\frac{d\bar{T}_i(t)}{dt} + \mu_i^2 \bar{T}_i(t) = \bar{g}_i(t), \quad t > 0, \quad i = 1, 2, \dots \quad (3.10.a)$$

onde  $\bar{g}_i(t)$  é definida por,

$$\bar{g}_i(t) = \frac{1}{N_i^{1/2}} \int_V [\psi(\mu_i, \mathbf{x})\nabla \cdot K(\mathbf{x})\nabla T(\mathbf{x}, t) - T(\mathbf{x}, t)\nabla \cdot K(\mathbf{x})\nabla \psi(\mu_i, \mathbf{x})]dv + \quad (3.10.b)$$

$$+ \frac{1}{N_i^{1/2}} \int_V \psi(\mu_i, \mathbf{x})P(\mathbf{x}, t)dv$$

A primeira integral de volume da equação (3.10.b) é então transformada em uma integral de superfície através da fórmula de *Green*, resultando:

$$\frac{1}{N_i^{1/2}} \int_V [\psi(\mu_i, \mathbf{x}) \nabla \cdot K(\mathbf{x}) \nabla T(\mathbf{x}, t) - T(\mathbf{x}, t) \nabla \cdot K(\mathbf{x}) \nabla \psi(\mu_i, \mathbf{x})] dv = \quad (3.10.c)$$

$$K(\mathbf{x}) \left[ \psi(\mu_i, \mathbf{x}) \frac{\partial T(\mathbf{x}, t)}{\partial \mathbf{n}} - T(\mathbf{x}, t) \frac{\partial \psi(\mu_i, \mathbf{x})}{\partial \mathbf{n}} \right] ds$$

Logo, de (3.10.b e c), o termo fonte transformado assume a seguinte forma,

$$\begin{aligned} \bar{g}_i(t) = & \frac{1}{N_i^{1/2}} \int_S K(\mathbf{x}) \left[ \psi(\mu_i, \mathbf{x}) \frac{\partial T(\mathbf{x}, t)}{\partial \mathbf{n}} - T(\mathbf{x}, t) \frac{\partial \psi(\mu_i, \mathbf{x})}{\partial \mathbf{n}} \right] ds + \quad (3.10.d) \\ & + \frac{1}{N_i^{1/2}} \int_V \psi(\mu_i, \mathbf{x}) P(\mathbf{x}, t) dv \end{aligned}$$

O primeiro integrando da equação (3.10.d) é algebricamente manipulado, para se eliminar o termo derivado  $\frac{\partial T(\mathbf{x}, t)}{\partial \mathbf{n}}$ , combinando-se as equações de contorno (3.1.c) e (3.3.b), de forma a se obter:

$$\begin{aligned} K(\mathbf{x}) \left[ \psi(\mu_i, \mathbf{x}) \frac{\partial T(\mathbf{x}, t)}{\partial \mathbf{n}} - T(\mathbf{x}, t) \frac{\partial \psi(\mu_i, \mathbf{x})}{\partial \mathbf{n}} \right] = \quad (3.11) \\ \phi(\mathbf{x}, t) \frac{\psi(\mu_i, \mathbf{x}) - K(\mathbf{x}) \frac{\partial \psi(\mu_i, \mathbf{x})}{\partial \mathbf{n}}}{\alpha(\mathbf{x}) + \beta(\mathbf{x})} \end{aligned}$$

Manipulando-se as equações (3.10.a), (3.10.d) e (3.11), resulta o seguinte problema transformado:

$$\frac{d\bar{T}_i(t)}{dt} + \mu_i^2 \bar{T}_i(t) = \bar{g}_i(t), \quad t > 0, \quad i = 1, 2, \dots \quad (3.12.a)$$

lembrando que  $\bar{g}_i(t)$  foi calculado anteriormente é expresso por,

$$\begin{aligned} \bar{g}_i(t) = & \frac{1}{N_i^{1/2}} \int_S \phi(\mathbf{x}, t) \frac{\psi(\mu_i, \mathbf{x}) - K(\mathbf{x}) \frac{\partial \psi(\mu_i, \mathbf{x})}{\partial \mathbf{n}}}{\alpha(\mathbf{x}) + \beta(\mathbf{x})} ds + \quad (3.12.b) \\ & + \frac{1}{N_i^{1/2}} \int_V \psi(\mu_i, \mathbf{x}) P(\mathbf{x}, t) dv \end{aligned}$$

As condições iniciais necessárias para solução do sistema de valor inicial transformado expresso pelas equações (3.12.a e b) são obtidas transformando-se a condição inicial do problema governante, que é expresso pela equação (3.1.b). Para tanto se aplica nesta última equação a definição do potencial transformado, expresso pela equação (3.7.a), resultando,

$$\bar{T}_i(0) = \frac{1}{N_i^{1/2}} \int_V w(\mathbf{x}) \psi(\mu_i, \mathbf{x}) T(\mathbf{x}, 0) dv = \frac{1}{N_i^{1/2}} \int_V w(\mathbf{x}) \psi(\mu_i, \mathbf{x}) f(\mathbf{x}) dv \quad (3.12.c)$$

É possível se expressar a solução analítica exata do sistema transformado definido pelas equações (3.12.a), (3.12.b) e (3.12.c), que após algumas manipulações, assume a seguinte forma,



$$\bar{T}_i(t) = e^{-\mu_i^2 t} \left[ \bar{f}_i + \int_0^t e^{\mu_i^2 r} \bar{g}_i(r) dr \right] \quad (3.13.a)$$

na equação anterior,

$$\bar{f}_i = \frac{1}{N_i^{1/2}} \int_V w(\mathbf{x}) \psi(\mu_i, \mathbf{x}) f(\mathbf{x}) dv \quad (3.13.b)$$

Aplicando-se a solução analítica exata do sistema transformado na fórmula de inversão, expressa pela equação (3.7.b), finalmente encontra-se a seguinte formulação explícita para o potencial T(x,t):

$$T(\mathbf{x}, t) = \sum_{i=1}^{\infty} \frac{\psi(\mu_i, \mathbf{x})}{N_i^{1/2}} e^{-\mu_i^2 t} \left[ \bar{f}_i + \int_0^t e^{\mu_i^2 r} \bar{g}_i(r) dr \right] \quad (3.14)$$

Neste estágio é importante ressaltar que para uma determinada ordem de truncamento no número de termos da série acima, à medida que cresce o valor numérico da coordenada temporal, a série (3.14) converge com menor número de termos. Isto mostra claramente que o uso de aproximações analíticas tende a diminuir expressivamente o tempo de CPU da solução em relação aos procedimentos de solução adotados via a utilização de métodos puramente numéricos, pois nestes casos não se consegue eliminar a dependência das coordenadas espaciais.

No subitem a seguir, será mostrada a aplicação desta metodologia para um problema não-linear de transferência de calor.

### 3.2 Aplicação: Transferência de Calor Não-Linear

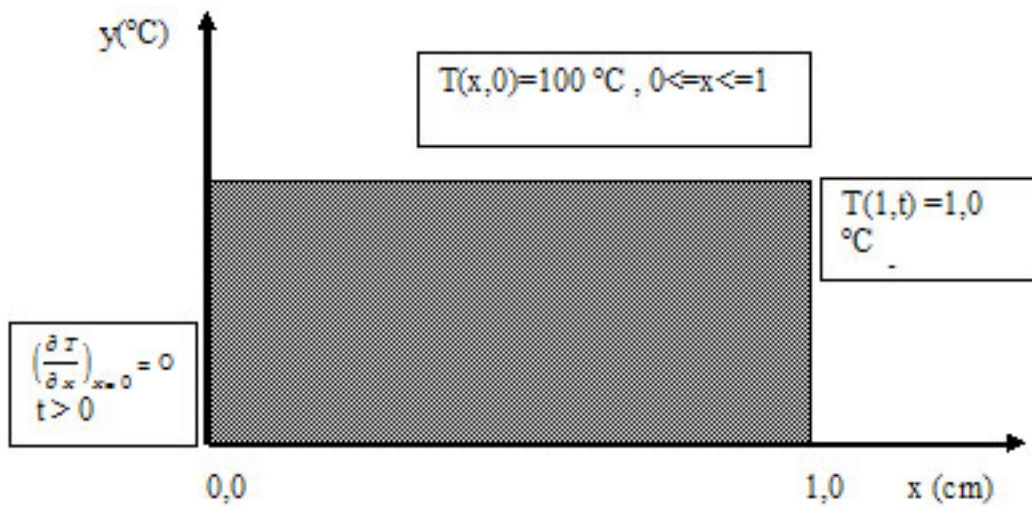
Como aplicação da metodologia de modelagem avançada através da utilização de procedimentos analítico-numéricos disponíveis nos formalismos da GITT, apresenta-se os estágios de construção e implementação de um algoritmo, através da utilização de programação simbólica declarativa, que no caso analiticamente manipula e resolve numericamente um problema de transferência de calor não-linear. Do ponto-de-vista de programação, é possível incorporar o algoritmo matemático de GITT a um *notebook*, para fins educacionais, o qual permite adicionalmente se trabalhar: conceituação física associada ao problema enfocado, tabelas, gráficos e animação.

O problema escolhido [09,10] é definido num domínio matemático que representa uma placa fina, isolada termicamente nos planos xz e yz, portanto ocorre transferência de calor apenas na direção x. Inicialmente a mesma é mantida numa temperatura conhecida, T<sub>0</sub>, a placa possui dimensão característica na direção x conhecida, e condições de contorno do primeiro e segundo tipos também conhecidas. A não-linearidade está expressa pela variação linear da capacidade térmica, ρ<sub>cp</sub>, em função da temperatura, T. A Figura 3.1 esboça o problema em discussão e é mostrada a seguir.

Através de um balanço de energia na placa, e após a manipulação dos termos componentes, a equação da condução de calor é expressa por,

$$\frac{\partial}{\partial \mathbf{x}} \left( \mathbf{k} \frac{\partial T}{\partial \mathbf{x}} \right) = \rho C_p \frac{\partial T}{\partial t}, \quad (3.15.a)$$

Para,  $t > 0$  e  $0 < x < 1$ , onde,  $T$  representa a temperatura  $T \equiv T(x,t)$ ,  $x$  e  $t$  correspondem às coordenadas espacial e temporal, respectivamente.  $K$  expressa a condutividade térmica do material,  $\rho$  é a densidade e  $c_p$  é o calor específico do material.



**Figura 3.1** Representação esquemática da placa fina para o problema unidimensional de transferência de calor não-linear por condução pura.

Condição inicial,

$$T = C1, 0 \leq x \leq 1, t = 0 \quad (3.15.b)$$

E as condições de contorno assumem a seguinte formulação,

$$\left(\frac{\partial T}{\partial x}\right)_{x=0} = 0, t > 0 \quad (3.15.c)$$

$$T = 1, x = 1, t > 0 \quad (3.15.d)$$

Ressaltando que a equação que expressa a não-linearidade é expressa por,

$$\rho c_p = k_1 + k_2 T(x, t) \quad (3.16)$$

Para solução do problema (3.15.a-d), discute-se a seguir o algoritmo matemático implementado no *Mathematica*®.

Numa primeira etapa, a equação governante, (3.15.a) é escrita da seguinte forma,

$$e1 := \partial_x (K \partial_x T[x, t]) == \rho c_p \partial_t T[x, t]$$

As condições iniciais e de contorno, respectivamente equações (3.15.b-d), são expressas por,

$$e2 := T[x, 0] == 100$$

$$e3 := (\partial_x T[x, t] /. x -> 0) == 0$$

$$e4 := T[1, t] == 1$$

No estágio seguinte o problema simbolicamente implementado é tratado utilizando-se os formalismos de GITT de problemas parabólicos. Para acelerar a taxa de convergência numérica, aplicam-se técnicas de filtragem, segundo a seguinte fórmula,

$$T[x,t]=Ts[x]+Th[x,t] \quad (3.17)$$

que separa o potencial original,  $T[x,t]$ , apresentado no problema original (3.15.a-d), que é não-homogêneo, em dois novos problemas matemáticos mais simples. Um não-homogêneo e em estado estacionário, que é definido na variável  $Ts[x]$  e outro homogêneo e transiente, expresso na variável  $Th[x,t]$ . O problema não-homogêneo requisita menor esforço computacional. O problema homogêneo concentrará maior esforço computacional, porém, certamente menor que se tentasse a solução do problema original sem a técnica de filtragem.

Para poder implementar esse poderoso filtro, é utilizada a idéia de regras locais, como demonstrado a seguir na definição do problema não homogêneo:

$$\begin{aligned} e5 &= e1 /. \{T \rightarrow (Ts[\#1] \&)\} \\ e6 &= \{e3, e4\} /. \{T \rightarrow (Ts[\#1] \&)\} \end{aligned}$$

Essas regras atuam especificamente nas linhas de código acima permitindo substituir o potencial original por  $Ts[x,t]$  na equação governante e condições de contorno, equações (3.15.a,c,d), absorvendo as não-homogeneidades. Agora **e5**, **e6** disponibilizam as seguintes atribuições:

$$\begin{aligned} K Ts''[x] &== 0 \\ \{Ts'[0] == 0, Ts[1] == 1\} \end{aligned}$$

Simbolicamente, isto significa que em qualquer linha de programação posterior, onde **e5** apareça, automaticamente a atribuição guardada em **e5** será disponibilizada. Observa-se ainda que nesta atribuição foi utilizada uma notação do *Mathematica*® para representar a equação diferencial. Para obter sua solução utiliza-se a função interna do mesmo, denominada DSOLVE.

$$\begin{aligned} eq11 &= DSolve[Flatten[\{e5, e6\}], \{Ts[x]\}, x][[1, 1]] \\ Ts[x] &\rightarrow 1 \end{aligned}$$

Observa-se que a solução para **e5** é  $Ts[x] = 1$ , o que é facilmente calculável. Agora se torna necessário obter a solução de  $Th[x,t]$  para complementar a solução geral do problema segunda a técnica de filtro aplicada. Para a definição do problema homogêneo necessita-se da definição da não-linearidade, representada aqui pela variação da capacidade térmica, expressada da seguinte forma,

$$cp := \frac{k1 + k2 (1 + Th[x, t])}{\rho}$$

Agora, analogamente utiliza-se a idéia de regras locais para aplicar  $Th[x,t]$  ao problema original, definido em **e1**, e às condições de contorno, representadas por **e3** e **e4**:

$$\begin{aligned} e7 &= e1 /. \{T \rightarrow (Th[\#1, \#2] \&)\} \\ e8 &= \{e3[[1]] == 0, e4[[1]] == 0\} /. \{T \rightarrow (Th[\#1, \#2] \&)\} \end{aligned}$$

Analogamente aos comentários feitos para o termo **e5**, **e7** armazena a seguinte equação diferencial,

$$K Th^{(2,0)}[x, t] == k1 Th^{(0,1)}[x, t] + k2 Th^{(0,1)}[x, t] + k2 Th[x, t] Th^{(0,1)}[x, t]$$

Neste estágio, os formalismos da GITT precisam ser definidos no algoritmo, pois os mesmos não são integrantes nas bibliotecas científicas do software *Mathematica*®. A nível de programação, para repassar os formalismos da GITT, são agora utilizadas técnicas de programação baseadas em regras, no caso, regras globais. Para maiores detalhes das regras globais definidas, consultar o programa completo no Apêndice B. Para fins ilustrativos, algumas destas são apresentadas a seguir,

```

Clear[op]
op[f_ + g_] := op[f] + op[g]
op[f_ == g_] := op[f] == op[g]
op[a_b_] := a op[b] /; FreeQ[a, x]
op[Th(0,1)[x, t]] := D[op[Th[x, t]], t]
op[Th[x, t]] := Tb[i][t]
op[f___] := int[ $\frac{\Psi[i][x]}{\sqrt{\text{npsi}}}$  f, {x, xi, xf}]
op[Th(2,0)[x, t]] := int[d[d[ $\frac{\Psi[i][x]}{\sqrt{\text{npsi}}}$  Th[x, t], x], x], {x, xi, xf}] - op[ $\mu[i]^2$  Th[x, t]] /. d -> D

```

Acima, o símbolo **op** representa um operador necessário para analiticamente se executar a transformação da equação diferencial parcial num sistema de equações diferenciais transformado, assim, as mesmas regras são atribuídas a **op** visando se obter a transformação comentada. Após este estágio, o passo seguinte se configura em aplicar as regras estabelecidas na equação armazenada em **e7**, como mostrado a seguir,

**e9=op[e7]**

**e10 = e9 /. {Th<sup>(0,1)</sup>[x, t] -> soma[ $\frac{\Psi[s][x]}{\sqrt{\text{npsi}}}$  D[Tb[s][t], t], {s, 1, nf}], Th[x, t] -> soma[ $\frac{\Psi[m][x]}{\sqrt{\text{npsi}}}$  Tb[m][t], {m, 1, nf}]}**

Observa-se acima que a formulação matemática armazenada no símbolo **e10** representa um sistema de equações diferenciais ordinárias e que depende apenas de uma coordenada independente, no caso, a variável temporal, **t**. O símbolo abaixo, após interpretar as regras, disponibiliza o seguinte resultado,

$$-K\mu[i]^2 Tb[i][t] == \frac{k2 \sum_{m=1}^{nf} \sum_{s=1}^{nf} \frac{(\int_0^1 \Xi[i][x] \Xi[m][x] \Xi[s][x] dx) Tb[m][t] Tb[s]'[t]}{\text{npsi}}}{\sqrt{\text{npsi}}} + k1 Tb[i]'[t] + k2 Tb[i]'[t]$$

A função  $\Psi[i][x]$  expressa o problema auxiliar associado do tipo *Sturm-Liouville*, necessário para se proceder aos cálculos analíticos caracterizados nos formalismos da técnica de transformada integral generalizada. A solução desta função é de fácil obtenção e apresenta a seguinte forma,

$$\Psi[i_][X_] = \text{Cos}[X\mu[i]]; \\ \mu[i_] = (i - .5) \pi;$$

Num estágio seguinte, visa-se obter resultados numéricos a partir dos cálculos analíticos já simbolicamente implementados. Torna-se agora necessário criar algumas regras para manipular as equações e obter resultados numéricos. Para fins ilustrativos, algumas destas são mostradas a seguir, podendo as demais ser vistas no Apêndice B.

```

Clear[int]
int[f_g_,x_]:=int[f,x]+int[g,x]
int[f_c_,{x_,xi_,xf_}]:=f int[c,{x,xi,xf}]/;FreeQ[f,x]
int[c_,{x_,xi_,xf_}]:=c(xf-xi)/;FreeQ[c,x]
int[Cos[m_ x_ ],{x_,xi_,xf_}]:=(Sin[m xf]-Sin[m xi])/m;/;FreeQ[m,x] && NumberQ[m] && m!=0.
int[Cos[0. x_ ],{x_,xi_,xf_}]:=int[1,{x,xi,xf}]
int[o_,s_] :=With[{c=Expand[o]},int[c,s]/;o!=c]

```

Nestas regras ocorre a utilização da função **int** que foi definida para resolver alguns tipos particulares de integrais, que aparecem nas etapas de solução por GITT do problema proposto. Após este estágio, se arbitra o número de termos para resolver o sistema de equações diferenciais ordinárias, definido no símbolo **nf**, inicialmente definido com o valor cinco, que caracteriza a ordem de truncamento do sistema, que terá seu valor associado à precisão numérica desejada para os potenciais originais,

**nf = 5;**

Para resolver o problema de valor inicial transformado é necessário transformar as condições iniciais do problema original, segundo os formalismos de GITT, como discutido no subitem 3.2, obtendo-se assim,

$$e11 = \text{Tb}[i][t] == \frac{99 \text{Sin}[\mu [i]]}{\sqrt{\text{npsi} \mu [i]}} / . t \rightarrow 0;$$

No estágio seguinte, visando a obtenção de valores numéricos a partir da função própria do *Mathematica*® denominada *NDSolve*, torna-se necessário se formatar algumas tabelas a partir das expressões obtidas, o que é definido simbolicamente a seguir nas expressões **e12** e **e13**:

**e12 = Table[e10 /. {soma -> Sum}, {i, 1, nf}];**

**e13=Table[e11, {i,1,nf}]**

Em seguida, **e12** e **e13** são incorporados por sua vez numa tabela única, definida por **e14** e se cria uma tabela adicional, armazenada em **e15**, que guarda os valores associados à condição inicial transformada, como mostrado a seguir:

**e14=Flatten[Append[e12,e13]];**

**e15=Flatten[Table[Tb[i][t], {i,1,nf}]]**

Segundo os valores definidos nas equações (3.16 e 3.15.a) e conforme os valores atribuídos no algoritmo apresentado no Apêndice B, as constantes **k1**, **k2** e **k**, assumem os seguintes valores,

**k1 = 1.; k2 = .3; K = 25;**

Após o cálculo numérico dos valores armazenados em e14 e e15, a função própria do *Mathematica*®, *NDSolve*, calcula numericamente o sistema transformado, o que foi atribuído no símbolo e28, para a coordenada temporal variando de zero a um,

```
e28 = Flatten[NDSolve[e14, e15, {t, 0, 1}]]
```

O símbolo **e28** armazena soluções para todos **T[i][t]**, mas é necessário atribuir para todos **T[i][t]** estas soluções, como mostrado a seguir,

```
Do[Tb[i][t_] = e28[[i, 2]], {i, 1, nf}]
```

Finalmente para se obter os potenciais desejados, sem perda de generalidade e conforme os formalismos da GITT, aplica-se a fórmula de inversa aos valores transformados homogêneos, **Tb[i][t]**, e adicionam-se os valores da solução particular do problema não-homogêneo, o que é visto a seguir,

$$Th[x_, t_] = \sum_{i=1}^{nf} \frac{\Psi[i][x]}{\sqrt{npsi}} Tb[i][t];$$

No capítulo a seguir são apresentados os resultados e discussão para os problemas de crescimento populacional e modelagem avançada.

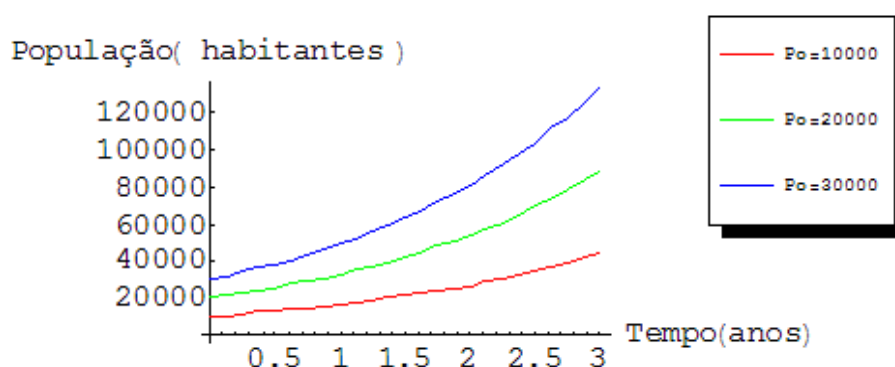
## 4 Resultados e Discussões

### 4.1 Migração Populacional

No item 2.2.2 do Capítulo 2 foram feitas a apresentação e discussão do problema de crescimento populacional, visando a construção do ambiente de trabalho e aprendizado assistidos por computador. Discute-se a seguir como o algoritmo foi estruturado e implementado num *notebook*, via o ambiente de programação simbólica *Mathematica*.

Para fins de implementação algorítmica do problema foram utilizados o *Mathematica* na versão 3.01 e um microcomputador Pentium II, 256 MB de RAM, 350 MHz e HD de 20GB.

O gráfico representado abaixo na figura 4.1 exibe a dinâmica do processo de crescimento populacional em função do tempo, segundo o primeiro modelo enfocado.



**Figura 4.1** Gráficos do crescimento populacional durante 3 dias, com taxa de crescimento constante  $k = 0.5$  e taxa de migração constante, para valores de população inicial  $P_0 = \{10.000, 20.000, 30.000\}$ .

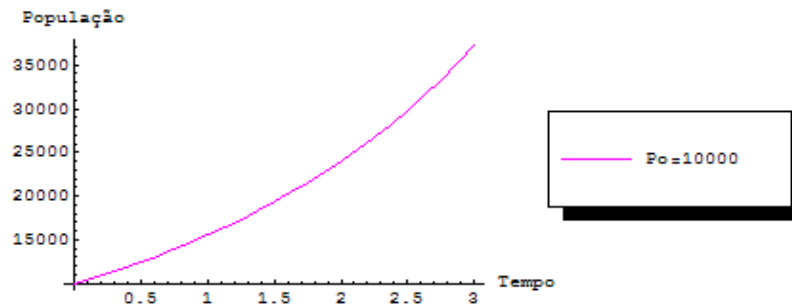
Para o primeiro modelo, onde a taxa de migração,  $r$ , é constante como mostrado na equação 2.1, os valores adotados nos parâmetros associados foram: taxa de crescimento  $k = 0.5$ , taxa de migração  $r = 100$  habitantes e valores de população inicial,  $P_0$ , de 10.000, 20.000 e 30.000 habitantes.

Nota-se que quando o valor da população inicial é maior, a velocidade de crescimento populacional se torna bem mais significativa para tempos mais longos, conforme notado pela declividade da curva de crescimento populacional em  $P_0 = 30.000$  habitantes.

A nível de aprendizado e familiaridade com a dinâmica do processo, outros valores de  $P_0$  podem ser facilmente arbitrados para se gerar novos gráficos, passo este fundamental na promoção e construção do aprendizado autônomo.

Um outro gráfico, expresso a seguir na figura 4.2, é gerado a partir do modelo de crescimento populacional com taxa de migração variando com o tempo, representado pelas Equações 2.2 e 2.3. Para tanto se segue assumindo os valores de taxa de

crescimento  $k = 0.5$ , valor de população inicial  $P_0 = 10.000$  habitantes,  $\alpha = 0,015$  e  $C = 6.000$ .



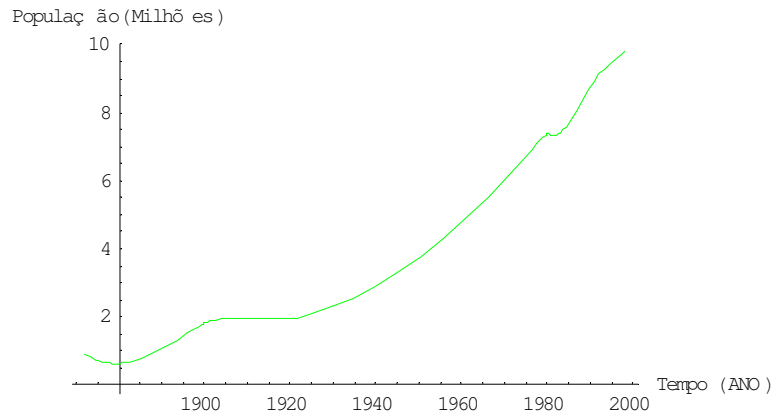
**Figura 4.2** Gráficos do crescimento populacional durante 3 anos, com taxa de crescimento constante,  $k = 0.5$  e taxa de migração em função do tempo segundo equação 2.2, para valores de população inicial  $P_0 = 10.000$ ,  $\alpha = 0,015$  e  $C = 6.000$ .

Na figura 4.2 nota-se que a taxa de migração,  $r$ , varia com o tempo com valor positivamente crescente. Para tempos curtos, observa-se que a variação de crescimento populacional na Figura 4.2, P, mostra um crescimento temporal crescente menos significativo que o observado no mesmo período temporal na figura 4.1, onde a declividade na curva de distribuição de população é mais suave. Já para tempos longos, o efeito de crescimento populacional na Figura 4.2, P, é um pouco mais acentuado, quando comparado aos casos de taxa de crescimento populacional constante. Este efeito teoricamente já era esperado. Observe que a população inicial utilizada para efeito de comparação das figuras 4.1 e 4.2 é  $P_0 = 10.000$ .

Não obstante, agora é possível modificar os parâmetros de entrada relacionados ao valor da população inicial  $P_0$  em ambos os casos, para se observar mais dinamicamente e comparar numa ótica pessoal o comportamento das curvas de distribuição de população,  $P$ , nos diferentes casos de taxa de migração populacional,  $r$ , analisando-se, assim, o comportamento do fenômeno de crescimento populacional.

Num último passo, com o intuito de enriquecer os resultados utilizados em posteriores aplicações no estudo de migração populacional e eventuais comparações entre dados simulados e reais, inclui-se a curva de crescimento temporal da população, mostrado na Figura 4.3.





**Figura 4.3** Gráfico de crescimento populacional temporal gerado a partir da interpolação dos dados coletados da população dos estados do Ceará e Piauí entre os anos de 1872 e 1998 (em milhões).

Os dados numéricos utilizados foram obtidos a partir de curvas de interpolação segundo dados reais do IBGE, referentes aos estados do Ceará e Piauí, compreendidos do período de 1872 a 1998 [67]. No item a seguir são apresentados e discutidos resultados referentes à modelagem do problema avançado.

## 4.2 Transferência de Calor Não-Linear

Para aplicações educacionais, foi discutida na sessão 3.3, a estrutura do algoritmo matemático que soluciona analítica e numericamente o problema acima definido e se seguiu à metodologia de aprendizado discutida na sessão 2.1, razão porque se concebeu um *notebook* mais geral, que incorpora os estágios de tratamento e solução matemática do problema em discussão, e elementos anteriormente discutidos na citada sessão, que compõem o processo de aprendizado (conceituação fenomenológica, manipulação matemática, solução numérica, visualização de resultados e interpretação fenomenológica).

Para fins de implementação algorítmica do problema expresso pelo problema de condução de calor não-linear, foram utilizados o ambiente de programação simbólica *Mathematica*, versão 3.01, um microcomputador Pentium II, 256 MB de RAM, 350 MHz e HD de 20GB. Para a placa plana em análise, os valores das propriedades termofísicas adotadas e demais parâmetros são: comprimento,  $L = 1,0$  cm, temperatura inicial,  $T(x,0) = 100,0$  °C, condições de contorno  $\partial T(0,t) / \partial x = 0$  e  $T(1,t) = 1,0$  °C, condutividade térmica,  $K = 25,0$ , os coeficientes associados à capacitância térmica,  $K1 = 1,0$ ,  $K2 = 0,30$ .

Apresenta-se no Apêndice B a estrutura do *notebook* completo, enquanto que a seguir discutem-se trechos do mesmo, que se reportam com mais ênfase à manipulação e visualização do problema de condução de calor.

Para reforçar o processo de aprendizado, torna-se possível incorporar ao algoritmo recursos que permitem a geração de tabelas a partir dos resultados analíticos e numéricos obtidos no algoritmo implementado. O que pode ser observado na Tabela 4.1 a seguir onde se atribuiu à abscissa,  $x$ , os valores variando de 0,0 cm a 1,0 cm, com

intervalo de 0,2 cm. Já para a coordenada temporal, t, foram atribuídos valores de 0,1 s a 0,9 s, com intervalo de 0,1 s.

```
TableForm[Table[Chop[T1[x, t]], {t, 0.1, 0.9, 0.1}, {x, 0, 1, 0.2}],
TableHeadings ->
  {"t=0.1", "t=0.2", "t=0.3", "t=0.4", "t=0.5", "t=0.6",
   "t=0.7", "t=0.8", "t=0.9"},
  {"x=0", "x=0.2", "x=0.4", "x=0.6", "x=0.8", "x=1.0"}]]
```

	x=0	x=0.2	x=0.4	x=0.6	x=0.8	x=1.0
t=0.1	96.577	93.3842	82.2667	61.3468	32.6424	1.
t=0.2	77.5846	73.3465	61.3601	43.6554	22.8266	1.
t=0.3	55.0841	52.0005	43.3876	30.8595	16.2607	1.
t=0.4	33.1879	31.3648	26.2652	18.8256	10.1239	1.
t=0.5	13.4891	12.7984	10.8547	7.98835	4.59362	1.
t=0.6	1.99058	1.94061	1.79667	1.57548	1.30094	1.
t=0.7	1.01039	1.00988	1.0084	1.0061	1.00321	1.
t=0.8	1.00009	1.00009	1.00007	1.00005	1.00003	1.
t=0.9	1.	1.	1.	1.	1.	1.

**Tabela 4.1** Curvas de distribuição de temperatura,  $T(x,t)$ , para o problema não-linear de condução de calor. Para a coordenada temporal, t, os valores atribuídos variam de 0,1 s a 0,9 s, com intervalo de 0,1 s. Para a coordenada espacial, x, varia de 0,0 cm a 1,0 cm.

Os resultados obtidos podem ser prontamente analisados e caso sejam detectados erros na programação, o algoritmo pode ser corrigido passo-a-passo, até se sanar eventuais problemas. Caso seja de interesse, facilmente podem ser alteradas as varreduras nas variáveis espacial e temporal para observar o comportamento do processo de resfriamento em tempos mais longos ou curtos, o que deve incentivar o processo de aprendizado autônomo, como também é possível agora se analisar o comportamento de outros materiais, sendo suficiente fornecer suas propriedades.

Uma característica matemática importante é o estudo da convergência da solução, quando se deve variar o número de termos utilizados para truncar a série de expansão de autofunções, equação (3.17), que armazena a solução do problema.

Outra possibilidade de análise da fenomenologia foi através de gráficos, gerados via funções própria do Mathematica, como a Plot, para realçar a ilustração do fenômeno da condução de calor não-linear, criticamente analisado via o efeito do resfriamento da placa. Nesta etapa, graças aos recursos existentes na programação baseada em regras comenta-se que todas as modificações feitas doravante em qualquer passo do algoritmo já implementado será imediatamente incorporada pela função Plot. Nas Figura 4.4 e Figura 4.5 a seguir, gera-se, para fins ilustrativos, gráficos da distribuição de temperatura em diferentes amostragens de tempo.

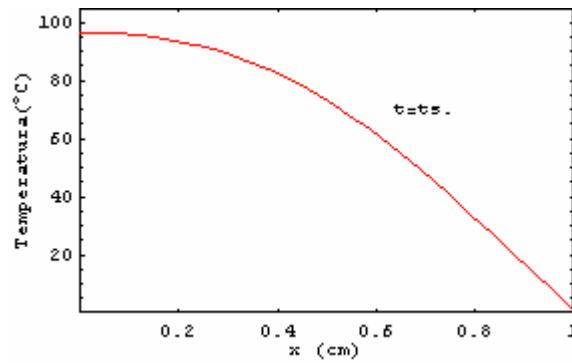


Figura 4.4 Perfil de distribuição de temperatura,  $T(x,t)$  para o problema de condução de calor não-linear ( $t = 0,1$  s).

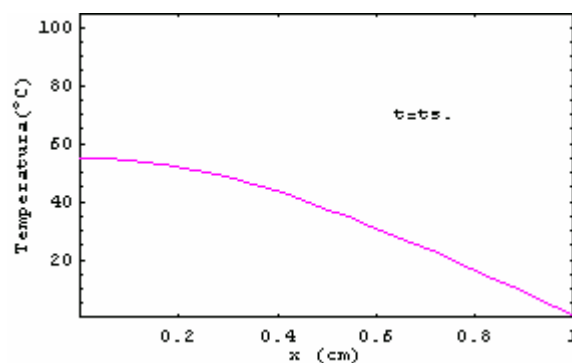


Figura 4.5 Perfil de distribuição de temperatura,  $T(x,t)$  para o problema de condução de calor não-linear ( $t = 0,3$  s).

Pode-se ainda utilizar a superposição de gráficos na mesma figura para se ter uma idéia comparativa da velocidade de resfriamento da placa para valores temporais conhecidos, no caso ilustrativo, observar a Figura 4.6, a seguir.

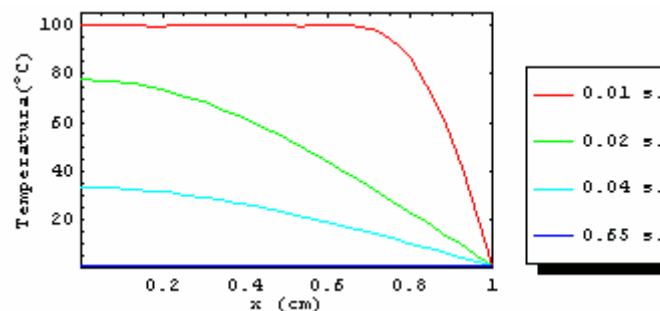
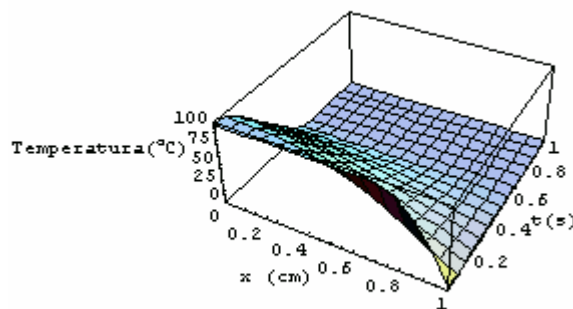


Figura 4.6 Perfil de distribuição de temperatura,  $T(x,t)$  para o problema de condução de calor não-linear ( $t = 0,01$ s, 0,2s, 0,4s, 0,6s).

Facilmente o algoritmo pode ser modificado, e a partir desses novos recursos, utilizando-se novos valores temporais para se analisar a dinâmica do processo de resfriamento da placa, permite-se desenvolver ao longo do processo de aprendizado um

sentimento físico mais apurado, como também incentivar a criatividade e o aprendizado autônomo. Neste estágio, chama-se atenção que o uso de texto congelado em muito limitaria o contato disponibilizado pelo computador em se poder interativamente criar novas situações físicas.

Outro recurso constitui a análise da informação via o uso de gráficos tridimensionais, facilmente disponibilizado através da função própria do *Mathematica*, Plot3D, o que é ilustrado em seguida na Figura 4.7 para se visualizar o processo de resfriamento da placa,



**Figura 4.7** Visualização em 3D de distribuição de temperatura,  $T(x,t)$  para o problema de condução de calor não-linear, com o tempo variando de 0,0s a 6,0s.

É possível variar a qualquer momento os parâmetros de entrada, obtendo-se o comportamento para outros materiais e em novos intervalos temporais, somando ganhos no processo de aprendizado anteriormente comentados.

Ainda para enriquecer os estágios de aprendizado assistido por computador, enumera-se o uso de recursos de animação gráfica, disponibilizado via a utilização da função própria do *Mathematica* denominada ShowAnimation, em combinação com outras funções próprias e tabelas geradas, como pode ser observado no Apêndice B, no trecho do algoritmo implementado que se reporta à animação do problema em discussão.

Caso os estágios acima enumerados causem dúvidas de serem assimilados, chama-se atenção para a importância que o *notebook* geral apresentado no Apêndice B passa a desempenhar.

Como ferramenta de uso concomitante é possível navegar interativamente e quando necessário ir em outros trechos do *notebook*, buscando reforçar a parte conceitual e descritiva dos mecanismos e leis gerais que descrevem o fenômeno da condução de calor. Este processo interativo vai progressivamente construindo o processo de aprendizado assistido por computador, conforme discutido na sessão 2.1.

Portanto, a aplicação dos formalismos da GITT quando associados às linguagens de programação simbólica proporcionam novos e promissores avanços nos aspectos de aprendizado e também ensino com ênfase para disciplinas das áreas de exatas.

## 5 Conclusões e Sugestões

O investimento em modelagem proposto e implementado contribui para o desenvolvimento de ferramentas e técnicas que progressivamente auxiliarão nos processos de aprendizado e ensino, tornando a interação homem-máquina cada vez mais abrangente.

Foram apresentadas ferramentas interativas para auxílio na construção dos estágios de aprendizado assistidos por computador, segundo eficientes recursos da programação simbólica. Estas promovem o aprendizado autônomo, segundo recursos que simultaneamente permitem: a navegação em textos de conteúdo programático (fenomenologia), a manipulação simbólica, numérica e solução matemática, visualização gráfica e interpretação de resultados. Esta cadeia de recursos exercita a criatividade e incentiva o desenvolvimento científico.

Na área específica de simulação de fenômenos físicos, cujos modelos encerram combinações de equações matemáticas, o computador torna mais dinâmico e sistemático a seqüência constituída pela necessidade de se conceituar as leis da física, gerar modelos matemáticos, manipular suas equações de forma a se obter soluções viáveis do problema e construir uma ponte de comunicação integrada que associa conceitos a interpretação dos mecanismos físicos associados disponibilizados via gráficos, tabelas e animações.

Através do conceito de *notebooks* e células e do uso de programação declarativa, construiu-se um ambiente interativo, onde os estágios do aprendizado foram didaticamente modularizados, segundo hierarquias de crescente complexidade de interpretação, disponibilizando para o usuário um contato personalizado e progressivo com cada uma das etapas do processo de aprendizado assistido por computador.

Recursos de programação baseada em regras e funcional permitiram a concepção destes documentos, onde as linhas de comando, escritas segundo notações matemáticas, podem ser alteradas e reprocessadas a qualquer momento. Um recurso muito importante é a possibilidade de modificar os valores numéricos dos parâmetros e propriedades físicas associados aos modelos matemáticos enfocados, permitindo a visão de novas realidades, não previstas no programa original, o que reforça o raciocínio criativo humano.

Os problemas de crescimento populacional e transferência de calor não-linear foram abordados e trabalhados em *notebooks* segundo as propostas de ensino assistido por computador e técnicas de programação acima discutida, visando-se ilustrar o tratamento unificado e sistemático apresentado anteriormente. Nos dois casos ressalta-se a possibilidade de aprofundamento da teorização dos problemas enfocados, repassando para os notebooks as extensões propostas.

Sugere-se aplicar esta metodologia para a confecção de *notebooks* relacionados a outras disciplinas das grandes áreas de conhecimento exatas ou para permitir a simulação de experimentos laboratoriais ou, ainda, que caracterizem disciplinas teóricas específicas acompanhadas de séries de exercícios e ilustradas via animações. Adaptar as metodologias apresentadas para conteúdos de disciplinas do segundo e primeiro graus.

# APÊNDICE A

## Package Malthus

Para facilitar a utilização dos modelos matemáticos apresentados no Capítulo 2 (eq. 2.1-3), pelos participantes do Projeto Waves, foi desenvolvido, utilizando-se o software Mathematica®, um package denominado Malthus.

O arquivo “Maltus.m”, contendo o respectivo package, permite a chamada de funções que incorporam os modelos finais desenvolvidos através de um único comando, no caso,

Maltus[ k,r, P0,t], para o modelo de crescimento populacional com taxa de migração constante (eq. 2.1)

Maltus[k,C,α,Po,t], para o modelo de crescimento populacional com taxa de migração variando também dependendo do tempo, t.

Para ser utilizado, o package precisa ser carregado dentro de um notebook através do comando:

```
<< Maltus.m
```

O arquivo “Maltus.m” precisa ser armazenado no mesmo diretório do notebook em que for chamado. Este arquivo pode ser requisitado pelo seguinte e-mail: pereira@lia.ufc.br.

Segue a seguir, a listagen desse pacote:

```
Maltus::usage = "Maltus[k,r,P0,t] use Maltus para onde,\n  k é a taxa de crescimento populacional.\n  r é a taxa de migração.\n  Po é a população inicial.\n  t é a taxa de variação do tempo em anos.\nMaltus[k,C,α,Po,t] use Maltus model to predict...onde,  k é a taxa de crescimento populacional.\n  C é a constante da taxa de migração em relação ao tempo.\n  α é a taxa de variação da migração em relação ao tempo.\n  Po é a população inicial.\n  t é a taxa de variação do tempo em anos.\n"
```

## **APÊNDICE B**

*Notebook* completo do problema de Transferência de calor não-linear

O conteúdo apresentado a seguir contém todos os estágios associados ao processo de construção do aprendizado aplicados a um problema de Transferência de Calor, que foi discutido ao longo do texto da Dissertação.

# Resolução de Um Problema Não-linear de Condução de Calor Utilizando GITT

## Conceituação

---

### ■ Introdução

A investigação matemática da condução de calor iniciou por volta de 1800, com Joseph Fourier, e ainda continua merecendo a atenção dos pesquisadores. Por exemplo, a análise da dissipação e da transferência de calor para longe das fontes, nos motores de alta velocidade, é um importante problema tecnológico. É comum nos problemas de condução de calor, existirem duas ou mais variáveis independentes, de modo que os modelos matemáticos correspondentes envolvem equações diferenciais parciais (E.D.P.'s) e não equações diferenciais ordinárias (E.D.O.'s).

### ■ Descrição do Problema

Considere o problema de condução de calor em uma placa situada ao longo do eixo  $x$  com extremos em  $x_i = 0$  e  $x_f = 1$ , condutividade térmica  $K$  constante, densidade  $\rho$ , capacidade térmica variável  $c_p$  e temperatura inicial  $T_0 = 100$  °C. Em  $x = 0$  o fluxo de calor é zero e em  $x = 1$  a temperatura é mantida constante igual a  $1$  °C.

### ■ Descrição do Algoritmo Generalizado (GITT)

O algoritmo generalizado é representado por duas equações diferenciais governantes, definidas em uma região finita  $V$ , com contorno  $S$ , composto por duas componentes  $S_1$  e  $S_2$  (isto é,  $S = S_1 \cup S_2$ ), com acoplamento no contorno  $S_2$ , cuja representação matemática se segue:

$$w_k(\mathbf{x}) \frac{\partial \theta_k(\mathbf{x}, t)}{\partial t} + L_k \theta_k = P_k(\mathbf{x}, t, \theta_1, \theta_2), \quad \mathbf{x} \in V, \quad t > 0, \quad k = 1, 2$$



possuindo as seguintes condições iniciais,

$$\theta_k(\mathbf{x}, 0) = f_k(\mathbf{x}), \quad \mathbf{x} \in V, \quad k = 1, 2$$

e as seguintes condições de contorno,

$$B_k \theta_k(\mathbf{x}, t) = \phi_k(\mathbf{x}, t, \theta_1, \theta_2), \quad \mathbf{x} \in S, \quad k = 1, 2$$

os operadores presentes na equação governante e condições de contorno são definidos como:

$$L_k \equiv -\nabla \cdot K_k(\mathbf{x}) \nabla + d_k(\mathbf{x}), \quad k = 1, 2$$

$$B_k \equiv [a_k(\mathbf{x}) + b_k K_k(\mathbf{x}) \frac{\partial}{\partial \eta}], \quad k = 1, 2$$

onde  $\theta_{\mathbf{k}}(\mathbf{x}, t)$  são os potenciais a serem obtidos, sendo  $\mathbf{x}$  e  $t$  variáveis independentes que representam as coordenadas espaciais e temporal, respectivamente. Os termos  $\mathbf{P}_{\mathbf{k}}(\mathbf{x}, t, \theta_1(\mathbf{x}, t), \theta_2(\mathbf{x}, t))$  e  $\phi_{\mathbf{k}}(\mathbf{x}, t, \theta_1(\mathbf{x}, t), \theta_2(\mathbf{x}, t))$  incorporam não-homogêneos, não-lineares e de acoplamento presentes no sistema.,  $\mathbf{a}(\mathbf{x})$  e  $\mathbf{b}(\mathbf{x})$  são coeficientes prescritos da equação de contorno e  $\partial / \partial \eta$  corresponde à derivada na direção normal e externa à superfície de contorno  $S$ .

Como primeiro passo, uma técnica de filtragem que minimiza os efeitos das não homogeneidades é utilizada, dividindo o problema em dois componentes como a seguir:

$$\theta_k(\mathbf{x}, t) = \theta_{ks}(\mathbf{x}, t) + \theta_{kh}(\mathbf{x}, t)$$

aparece uma componente independente do tempo  $\theta_{ks}$  e um componente homogêneo  $\theta_{kh}$ . O problema estacionário  $\theta_{ks}$  inclui representações características do problema geral e condições de contorno. A formulação do componente homogêneo, é obtido a partir da solução dos problemas mais fáceis seguintes:

$$L_k \theta_{kh} = P_{kh}(\mathbf{x}, t), \quad \mathbf{x} \in V, \quad k = 1, 2$$

$$B_k \theta_{kh} = \phi_{kh}(\mathbf{x}, t), \quad \mathbf{x} \in S, \quad k = 1, 2$$

O componente estacionário é facilmente resolvido enquanto que o sistema homogêneo deve ser resolvido pela técnica de transformada integral, através da solução de problemas auxiliares desacoplados ( $k=1,2$ ) de tipo de Sturm-Liouville que evitam a existência de autovalores complexos e são definidos como:

$$L_1 \Psi_i(x) = \mu_i^2 w_1(x) \Psi_i(x), x \in V, i = 1, 2, \dots$$

$$L_2 \Gamma_i(\mathbf{x}) = \lambda_i^2 w_2(\mathbf{x}) \Gamma_i(\mathbf{x}), \mathbf{x} \in V, i = 1, 2, \dots$$

com as seguintes condições de contorno:

$$B_1 \Psi_i(\mathbf{x}) = 0, \mathbf{x} \in S, i = 1, 2, \dots$$

$$B_2 \Gamma_i(\mathbf{x}) = 0, \mathbf{x} \in S, i = 1, 2, \dots$$

Pela propriedade de ortogonalidade das autofunções associadas, os pares de transformação integral é definido como a seguir:

Transformada:

$$\bar{\theta}_{1,i}(t) = \int_V w_1(\mathbf{x}) \frac{\Psi_i(\mathbf{x})}{N_i^{1/2}} \theta_{1h}(\mathbf{x}, t) dv$$

Inversa:

$$\theta_{1h}(\mathbf{x}, t) = \sum_{i=1}^{\infty} \frac{\Psi_i(\mathbf{x})}{N_i^{1/2}} \bar{\theta}_{1i}(t)$$

e,

Transformada:

$$\bar{\theta}_{2,i}(t) = \int_V w_2(\mathbf{x}) \frac{\Gamma_i(\mathbf{x})}{M_i^{1/2}} \theta_{2h}(\mathbf{x}, t) dv$$

Inversa:

$$\theta_{2h}(\mathbf{x}, t) = \sum_{i=1}^{\infty} \frac{\Gamma_i(\mathbf{x})}{M_i^{1/2}} \bar{\theta}_{2i}(t)$$

onde  $N_i$  e  $M_i$  são as integrais de normalização, ou simplesmente normas, definidas como abaixo:

$$N_i = \int_V w_1(\mathbf{x}) \Psi_i^2(\mathbf{x}) dv$$

$$M_i = \int_V w_2(\mathbf{x}) \Gamma_i^2(\mathbf{x}) dv$$

e as condições iniciais dos sistemas transformados são obtidas pela transformação das condições iniciais originais.

Finalmente trunca-se o sistema numa ordem finita  $N$ , condizente com a convergência desejada, e obtêm-se o seguinte sistema ordinário de equações diferenciais com parâmetros termofísicos constantes :

$$\frac{d\mathbf{Y}(t)}{dt} + \mathbf{A}_{2N,2N} \mathbf{Y}(t) = \mathbf{0}_{2N,1}$$

onde:

$$\mathbf{Y}(t) = \{\bar{\theta}_{11}(\tau), \bar{\theta}_{12}(\tau) \dots \bar{\theta}_{21}(\tau), \bar{\theta}_{22}(\tau) \dots \bar{\theta}_{2N}(\tau)\}^T$$

Os potenciais são computados pela soma dos resultados obtidos dos dois componentes da equação filtro.

## Modelagem

---

### ■ Equação Governante

$$\frac{\partial}{\partial x} \left( k \frac{\partial T}{\partial x} \right) = \rho c_p \frac{\partial T}{\partial t},$$

$$\rho c_p = k_1 + k_2 T(x, t)$$

### ■ Condições de Contorno

$$\left( \frac{\partial T}{\partial x} \right)_{x=0} = 0$$

$$T(1,t)=1$$

### ■ Condições Iniciais

$$T(x, 0) = 100$$

## Implementação

---

### ■ Dados Iniciais do Problema

Atribuição dos extremos no eixo x:

```
In[96]:= xi = 0;
         xf = 1;
```

Definição da equação governante:

```
In[98]:= e1 := ∂x (κ ∂x T[x, t]) == ρ cp ∂t T[x, t]
```

Das condições de contorno:

```
In[99]:= e3 := (∂x T[x, t] /. x -> 0) == 0
         e4 := T[1, t] == 1
```

E da condição inicial definida no problema:

```
In[101]:= e2 := T[x, 0] == 100
```

Utiliza-se aqui uma função filtro(Ts) que absorve as não-homogeneidades

$$T[x,t] = T_s[x] + T_h[x,t]$$

### ■ Solução Para o Problema Estacionário

Definição da equação governante do problema e condições de contorno absorvendo as não-homogeneidades:

```
In[102]:= e5 = e1 /. {T -> (Ts[#1] &) }
```

```
Out[102]= K Ts''(x) == 0
```

```
In[103]:= e6 = {e3, e4} /. {T -> (Ts[#1] &) }
```

```
Out[103]= {Ts'(0) == 0, Ts(1) == 1}
```

Solução das equações utilizando a rotina Dsolve:

```
In[104]:= eq11 = DSolve[Flatten[{e5, e6}], {Ts[x]}, x][[1, 1]]
```

```
Out[104]= Ts(x) -> 1
```

```
In[105]:= Ts[x_] = Ts[x] /. eq11
```

```
Out[105]= 1
```

## ■ Solução Para o Problema Homogêneo

Solucionada a função para o estado estacionário, passa-se agora a trabalhar com a parte homogênea(Th) da solução proposta.

Para o problema homogêneo, a variação da propriedade de capacidade térmica( $c_p \rho$ ) é definida:

$$\text{In[106]:= } c_p := \frac{k_1 + k_2 (T_s[x] + Th[x, t])}{\rho}$$

Possibilitando a definição da equação governante, juntamente com suas condições de contorno:

```
In[107]:= e7=e1/.{T->(Th[#1,#2]&)}

```

```
Out[107]= K Th^(2,0)(x, t) == (k1 + k2 (Th(x, t) + 1)) Th^(0,1)(x, t)
```

```
In[108]:= e8={e3[[1]]==0, e4[[1]]==0}/. {T->(Th[#1,#2]&)}

```

```
Out[108]= {Th^(1,0)(0, t) == 0, Th(1, t) == 0}
```

```
In[109]:= e7=ExpandAll[e7]

```

```
Out[109]= K Th^(2,0)(x, t) == k1 Th^(0,1)(x, t) + k2 Th^(0,1)(x, t) + k2 Th(x, t) Th^(0,1)(x, t)
```

## Operador OP

Conjunto de regras globais atribuídas ao símbolo **op**, responsáveis pelas transformações das equações previstas nos formalismos do método de transformada integral.

```
In[110]:= Clear[op]

```

```
op[f_ + g_] := op[f] + op[g]
```

```
op[f_ == g_] := op[f] == op[g]
```

```
op[a_ b_] := a op[b] /; FreeQ[a, x]
```

```
op[Th^(0,1)[x, t]] := D[op[Th[x, t]], t]
```

```
op[Th[x, t]] := Tb[i][t]
```

```
op[f___] := int[ $\frac{\Psi[i][x]}{npsi^{\frac{1}{2}}}$  f, {x, xi, xf}]
```

```
op[Th^(2,0)[x, t]] :=
```

```
int[d[d[ $\frac{\Psi[i][x]}{npsi^{\frac{1}{2}}}$  Th[x, t], x], {x, xi, xf}] - op[ $\mu[i]^2$  Th[x, t]] /.

```

```
d -> D
```

```

In[118]:= Unprotect[Times];
Clear[d]
d[f_ + g_, x_] := d[f, x] + d[g, x]
d[a_b_, x_] := a d[b, x] /; FreeQ[a, x]
d[d[a_, x_Symbol], x_Symbol] := d[a, x, x]
Protect[Times];

In[124]:= Clear[int]
int[a_ == b_, x_List] := int[a, x] == int[b, x]
int[o_Plus a_, s_List] := int[# a, s] & /@ o
int[a_b_, {x_, xi_, xf_}] := a int[b, {x, xi, xf}] /; FreeQ[a, x]
int[
  d[d[a_[x_, t_], x_] b_[x_, t_] e-:1 c_[i_][x_], x_], {x_, x0_, xf_}] :=
  (d[a[xf, t], x] b[xf, t] e c[i][xf]) - (d[a[x0, t], x] b[x0, t] e c[i][x0])
  int[d[a_[x_, t_] b_[i_][x_], x_], {x_, xi_, xf_}] :=
  d[a[xf, t] b[i][xf], x] - d[a[xi, t] b[i][xi], x]
int[0, {x_, xi_, xf_}] := 0
int[v_soma[a_b_, {j_, ji_, jf_}], {x_, xi_, xf_}] :=
  soma[a int[v b, {x, xi, xf}], {j, ji, jf}] /; FreeQ[a, x]
int[v_soma[a_b_, s_List], {x_, xi_, xf_}] :=
  soma[a int[v b, {x, xi, xf}], s] /; FreeQ[a, x]
  MakeBoxes[int[a_, var_], fmt_] := MakeBoxes[Integrate[a, var], fmt]

In[134]:= (* declaracoes para soma*)
Clear[soma]
soma[o_Plus, s_List] := soma[#, s] & /@ o
soma[a_soma[b_, s_], c_List] := soma[a b, c, s]
soma[(a_ + b_) c_, s_] := soma[a c, s] + soma[b c, s]
MakeBoxes[soma[a_, in_], fmt_] := MakeBoxes[Sum[a, in], fmt]

```

O operador é aplicado sobre a equação **e7** (equação governante) gerando um sistema de equações diferenciais ordinárias dependentes apenas do tempo como resultado, que é guardado na equação **e9**.

```
In[139]:= e9=op[e7]
```

$$\text{Out[139]} = -K \mu(i)^2 \text{Tb}(i)(t) = \frac{k2 \int_0^1 \text{Th}(x, t) \Psi(i)(x) \text{Th}^{(0,1)}(x, t) dx}{\sqrt{\text{npsi}}} + k1 \text{Tb}(i)'(t) + k2 \text{Tb}(i)'(t)$$

Devido o termo não-linear " $\text{Th}[x, t] \text{Th}^{(0,1)}[x, t]$ " não ser coberto pelas regras gerais do método criados em "op", é necessário a criação de uma regra local, como a seguir, para seu tratamento na obtenção do sistema transformado:

$$\text{In[140]} = \text{e10} = \text{e9} /. \left\{ \text{Th}^{(0,1)}[x, t] \rightarrow \text{soma} \left[ \frac{\Psi[s][x]}{\sqrt{\text{npsi}}} \text{D}[\text{Tb}[s][t], t], \{s, 1, \text{nf}\} \right], \right. \\ \left. \text{Th}[x, t] \rightarrow \text{soma} \left[ \frac{\Psi[m][x]}{\sqrt{\text{npsi}}} \text{Tb}[m][t], \{m, 1, \text{nf}\} \right] \right\}$$

$$\text{Out}[140]= -K \mu(i)^2 \text{Tb}(i)(t) == \frac{k2 \sum_{m=1}^{\text{nf}} \sum_{s=1}^{\text{nf}} \frac{\left(\int_0^1 \Psi(i)(x) \Psi(m)(x) \Psi(s)(x) dx\right) \text{Tb}(m)(t) \text{Tb}(s)'(t)}{\text{npsi}}}{\sqrt{\text{npsi}}} + k1 \text{Tb}(i)'(t) + k2 \text{Tb}(i)'(t)$$

```
In[141]:= e10 = e10[[2]] == e10[[1]];
```

Seguindo os passos dos formalismos, definimos um problema de Sturm-Liouville, que tem por solução:

```
In[142]:= Clear[Ψ, μ, Γ, σ]
Ψ[i_][x_] = Cos[x μ[i]];
μ[i_] = (i - .5) π;
```

Com a condição inicial(e2) :

```
In[145]:= e2
Out[145]= T(x, 0) == 100
```

E como  $T_s[x]=1$ , temos que  $T_h[x,0]=99$ . Assim sendo:

```
In[146]:= npsi = .5;
e11 = Tb[i][t] == \frac{99 \text{Sin}[\mu[i]]}{\sqrt{\text{npsi}} \mu[i]} /. t \to 0
Out[147]= Tb(i)(0) == \frac{44.5657 \text{sin}((i - 0.5) \pi)}{i - 0.5}
```

## ■ Demais Passos

Agora podemos armazenar as formulações analíticas em tabelas e resolver o problema transformado numericamente usando a função interna do Mathematica, `NDSolve`.

Regras globais são definidas a seguir para serem usadas na construção das tabelas.

```
In[148]:= Unprotect[Sin, Cos];
Sin/:Sin[x_]*Cos[y_] := Sin[x+y]/2 + Sin[x-y]/2
Sin/:Sin[x_]*Sin[y_] := Cos[x-y]/2 - Cos[x+y]/2
Cos/:Cos[x_]*Cos[y_] := Cos[x+y]/2 + Cos[x-y]/2
Sin/:Sin[x_]^(n_Integer)?Positive := Expand[(1/2 - Cos[2*x]/2) * Sin[x]^(n-2)] /
;n>1
Cos/:Cos[x_]^(n_Integer)?Positive := Expand[(1/2 + Cos[2*x]/2) * Cos[x]^(n-2)] /
;n>1
Protect[Sin, Cos];
```

```
In[155]:= Clear[int]
int[f_+g_,x_]:=int[f,x]+int[g,x]
int[f_ c_,{x_,xi_,xf_}]:=f int[c,{x,xi,xf}]/;FreeQ[f,x]
int[c_,{x_,xi_,xf_}]:=c(xf-xi)/;FreeQ[c,x]
int[Cos[m_ x_] ,{x_,xi_,xf_}]:= (Sin[m xf]-Sin[m xi])/m;/;FreeQ[m,x] &&
NumberQ[m] && m!=0.
int[Cos[0. x_] ,{x_,xi_,xf_}]:=int[1,{x,xi,xf}]
int[o_,s_] :=With[{c=Expand[o]},int[c,s]/;o!=c]
```

Definir o número de equações que irão compor o sistema.

```
In[162]:= nf = 5;
```

Construir algumas tabelas como segue:

```
In[163]:= e13=Table[e11,{i,1,nf}]
```

```
Out[163]= {Tb(1)(0) == 89.1313, Tb(2)(0) == -29.7104,
           Tb(3)(0) == 17.8263, Tb(4)(0) == -12.733, Tb(5)(0) == 9.90348}
```

```
In[164]:= e12 = Table[e10 /. {soma -> Sum}, {i, 1, nf}];
```

```
In[165]:= e14=Flatten[Append[e12,e13]];
```

```
In[166]:= e15=Flatten[Table[Tb[i][t],{i,1,nf}]]
```

```
Out[166]= {Tb(1)(t), Tb(2)(t), Tb(3)(t), Tb(4)(t), Tb(5)(t)}
```

Definir os valores das constantes k1, k2 e K :

```
In[167]:= k1 = 1.; k2 = .3; K = 25;
```

Por fim, usa-se a função interna do Mathematica, NDSolve, juntamente com as tabelas criadas :

```
In[168]:= e28 = Flatten[NDSolve[e14, e15, {t, 0, 1}]]
```

```
Out[168]= {Tb(1)(t) -> InterpolatingFunction[{{0. 1.}, <>][t],
           Tb(2)(t) -> InterpolatingFunction[{{0. 1.}, <>][t], Tb(3)(t) -> InterpolatingFunction[{{0. 1.}, <>][t],
           Tb(4)(t) -> InterpolatingFunction[{{0. 1.}, <>][t], Tb(5)(t) -> InterpolatingFunction[{{0. 1.}, <>][t]}
```

Permitindo assim, a definição funcional das transformadas:

```
In[169]:= Do[Tb[i][t_] = e28[[i, 2]], {i, 1, nf}]
```

$$\text{In[170]:= Th}[\mathbf{x}_-, t_-] = \sum_{i=1}^{nf} \frac{\Psi[i][\mathbf{x}]}{\sqrt{n\text{psi}}} \text{Tb}[i][t];$$

E a definição da função final (T= Ts+Th):

```
In[171]:= T1[x_, t_] = Ts[x] + Th[x, t];
```



## Visualização Gráfica dos Resultados

---

### ■ Tabelas

```
In[175]:= TableForm[Table[Chop[T1[x, t]], {t, 0.1, 0.9, 0.1}, {x, 0, 1, 0.2}],
  TableHeadings -> {"t=0.1", "t=0.2", "t=0.3",
    "t=0.4", "t=0.5", "t=0.6", "t=0.7", "t=0.8", "t=0.9"},
  {"x=0", "x=0.2", "x=0.4", "x=0.6", "x=0.8", "x=1.0"}]
```

Out[175]//TableForm=

	x=0	x=0.2	x=0.4	x=0.6	x=0.8	x=1.0
t=0.1	96.577	93.3842	82.2667	61.3468	32.6424	1.
t=0.2	77.5846	73.3465	61.3601	43.6554	22.8266	1.
t=0.3	55.0841	52.0005	43.3876	30.8595	16.2607	1.
t=0.4	33.1879	31.3648	26.2652	18.8256	10.1239	1.
t=0.5	13.4891	12.7984	10.8547	7.98835	4.59362	1.
t=0.6	1.99058	1.94061	1.79667	1.57548	1.30094	1.
t=0.7	1.01039	1.00988	1.0084	1.0061	1.00321	1.
t=0.8	1.00009	1.00009	1.00007	1.00005	1.00003	1.
t=0.9	1.	1.	1.	1.	1.	1.

### ■ Gráficos

```
In[90]:= << Graphics`Graphics`
```

```
In[91]:= Clear[opts]
```

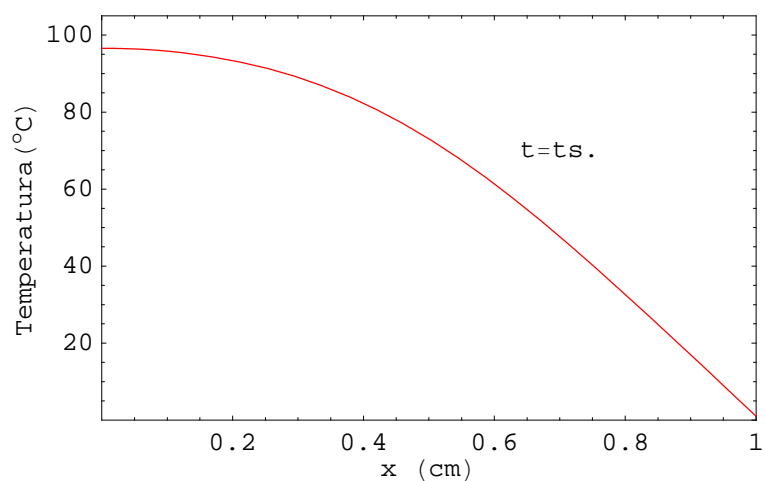
```
opts=Sequence[Prolog->{Text["t="<>ToString[t]<>"s.",{0.7,70}]},Frame->
  True,FrameLabel->{"x (cm)", "Temperatura(°C)"},PlotRange->
  {{0,1},{0,105}}];
```

```
In[93]:= Clear[t]
```

```
t=0.1
```

```
Out[94]= 0.1
```

```
In[95]:= Plot[T1[x, t], {x, 0, 1}, PlotStyle -> {RGBColor[1, 0, 0]}, Evaluate[opts] ]
```

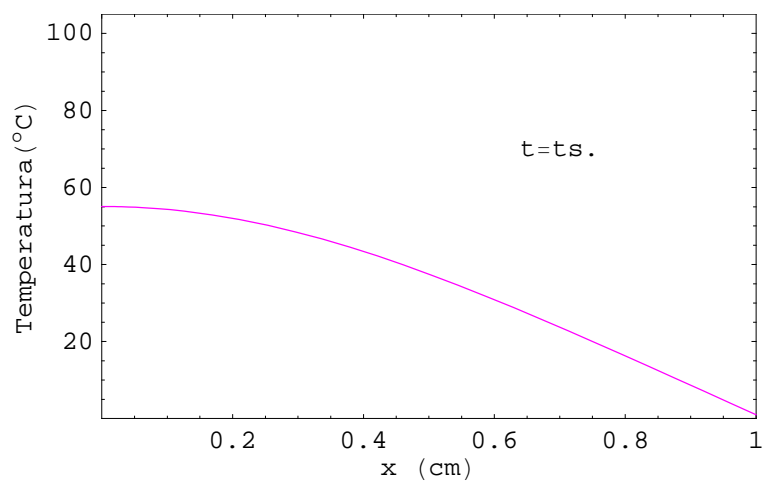


```
Out[95]= - Graphics -
```

```
In[96]:= t=0.3
```

```
Out[96]= 0.3
```

```
In[97]:= Plot[T1[x, t], {x, 0, 1}, PlotStyle -> {RGBColor[1, 0, 1]}, Evaluate[opts] ]
```

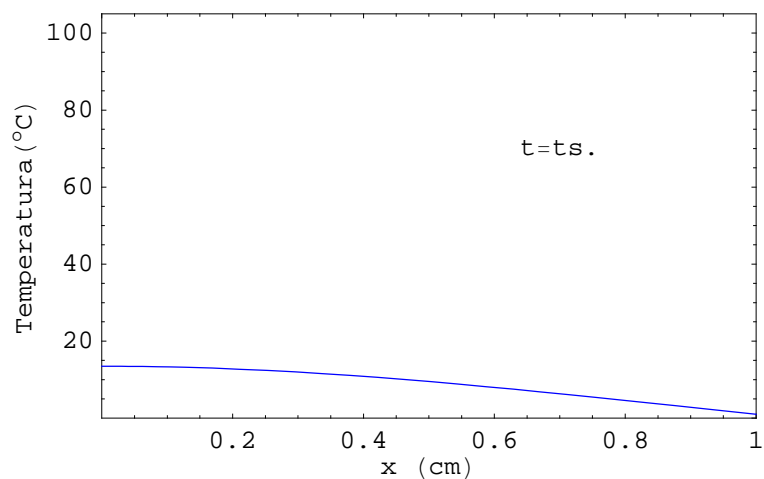


```
Out[97]= - Graphics -
```

```
In[98]:= t=0.5
```

```
Out[98]= 0.5
```

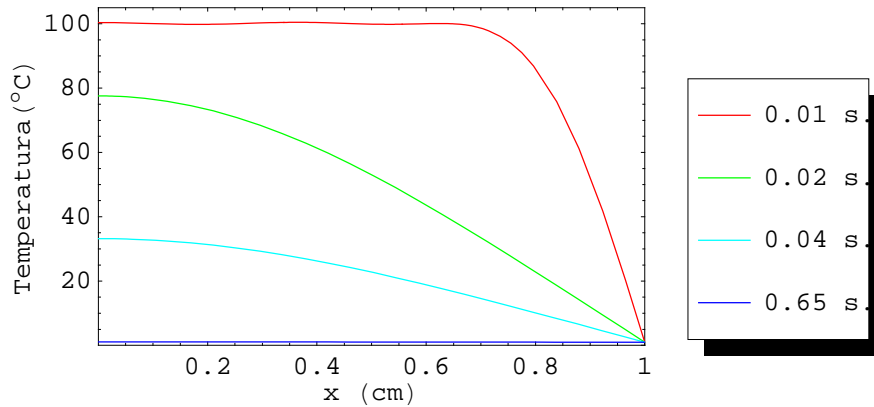
```
In[99]:= Plot[T1[x, t], {x, 0, 1}, PlotStyle -> {RGBColor[0, 0, 1]}, Evaluate[opts] ]
```



```
Out[99]= - Graphics -
```

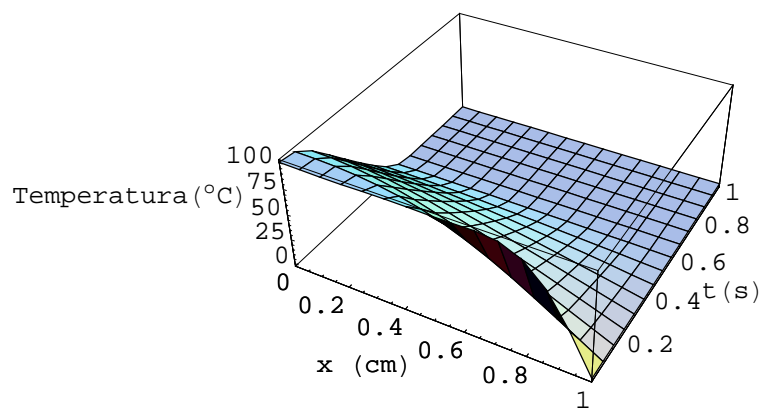
```
In[100]:= << Graphics`Legend`
```

```
In[101]:= Plot[{T1[x, 0.01], T1[x, 0.2], T1[x, 0.4], T1[x, 0.65]},
  {x, 0, 1}, PlotStyle -> {RGBColor[1, 0, 0],
  RGBColor[0, 1, 0], RGBColor[0, 1, 1], RGBColor[0, 0, 1]},
  PlotLegend -> {"0.01 s.", "0.02 s.", "0.04 s.", "0.65 s."},
  LegendPosition -> {1.1, -.4}, Frame -> True,
  FrameLabel -> {"x (cm)", "Temperatura (°C)"}, PlotRange -> {{0, 1}, {0, 105}}]
```



```
Out[101]= - Graphics -
```

```
In[102]:= Plot3D[T1[x,t],{x,0,1},{t,0.01,1},AxesLabel→
{"x (cm)","t(s)","Temperatura(°C)"}]
```



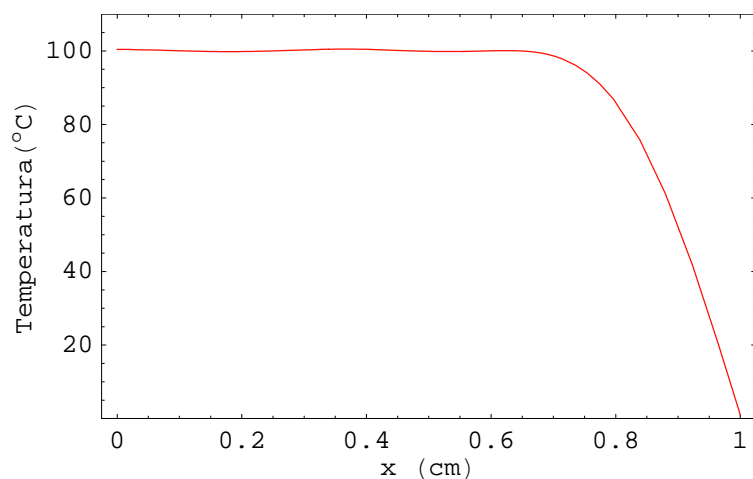
Out[102]= - SurfaceGraphics -

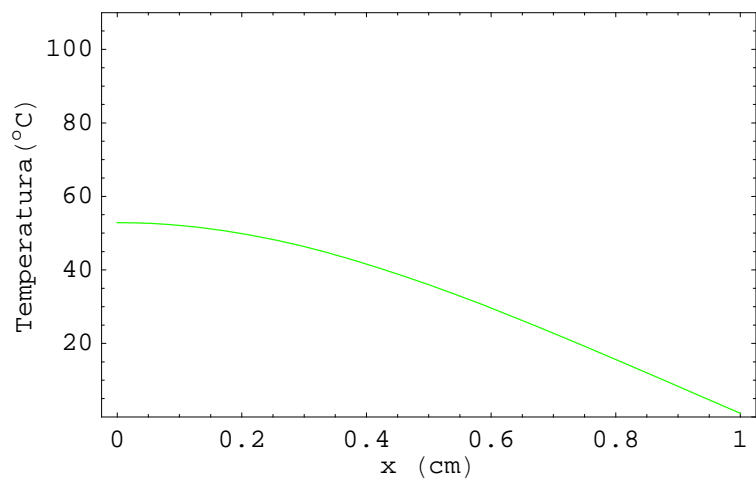
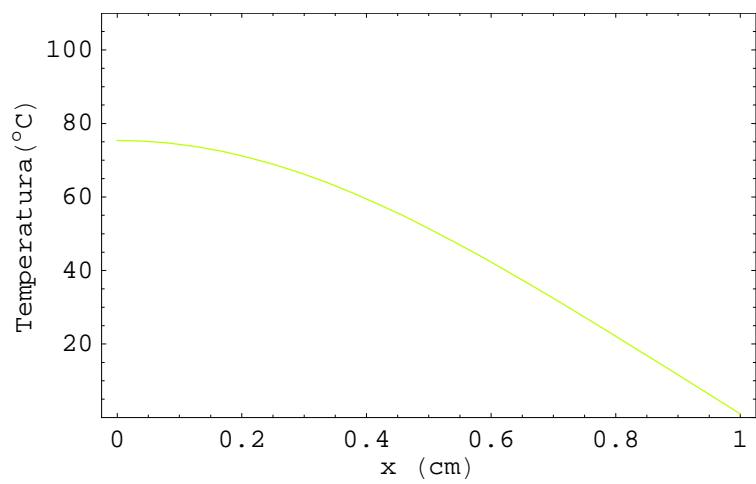
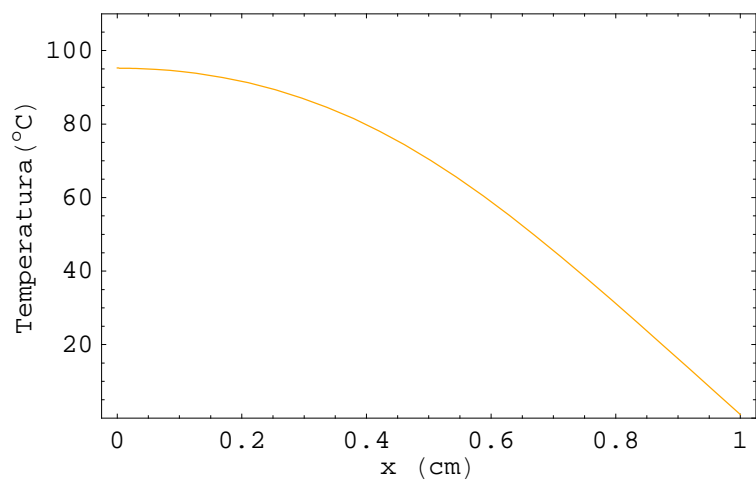
## ■ Animações

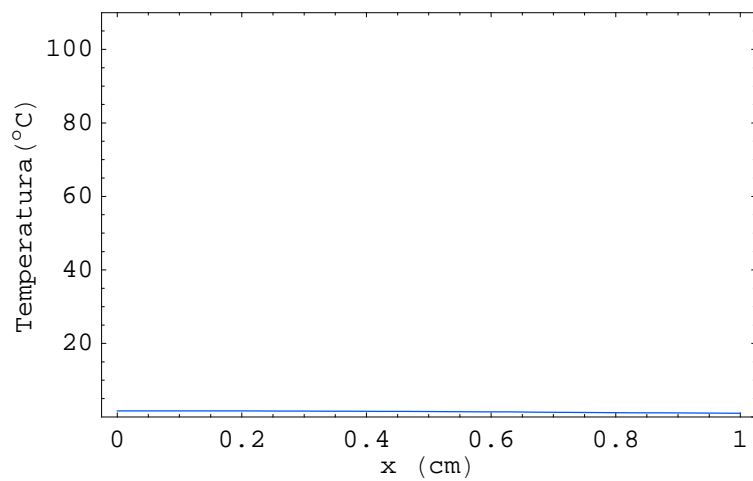
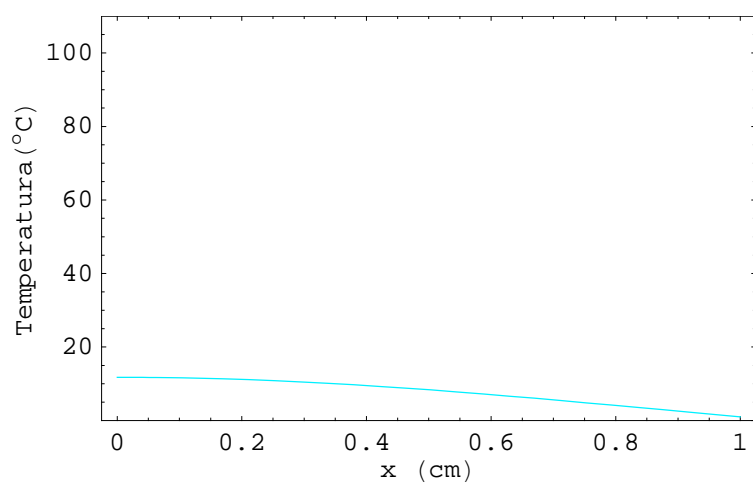
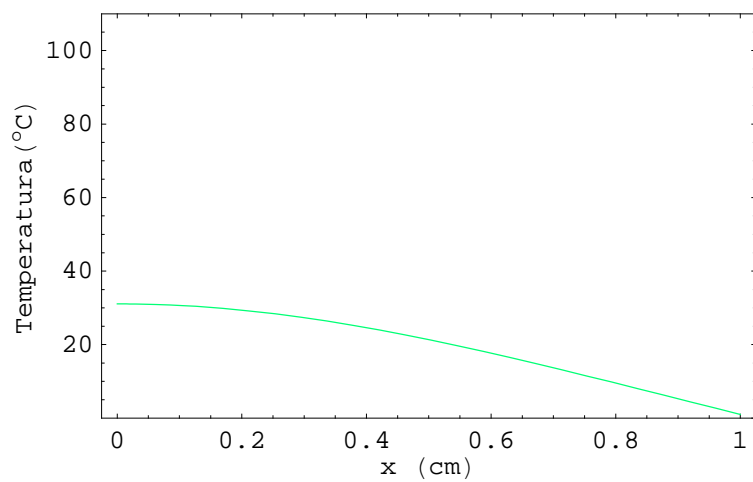
```
In[103]:= << Graphics`Animation`
```

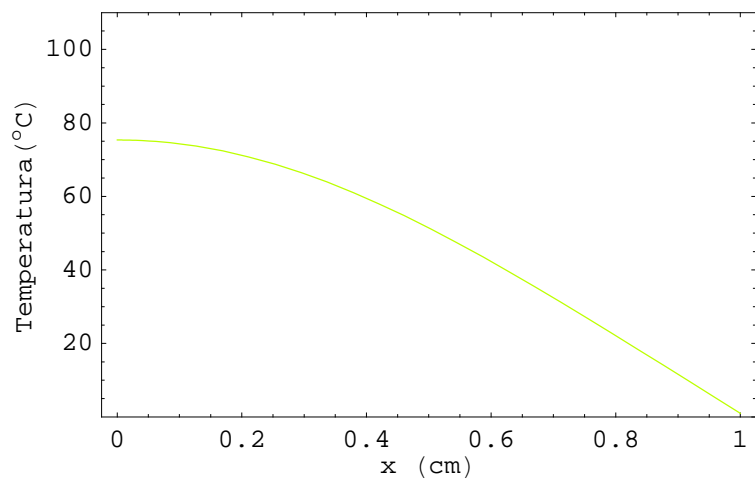
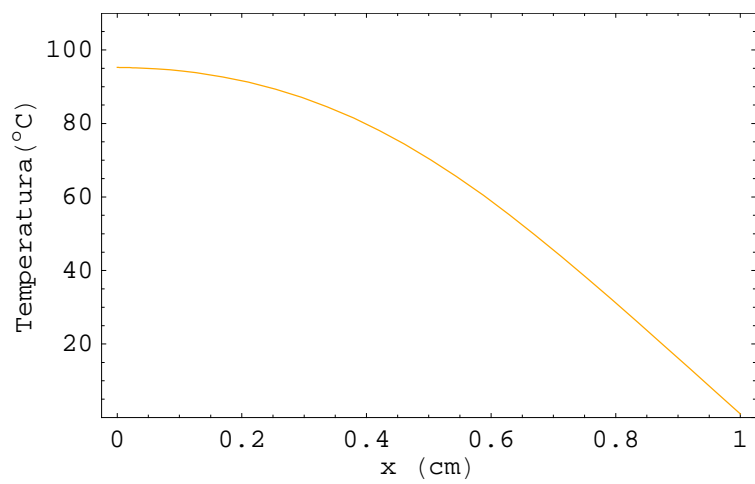
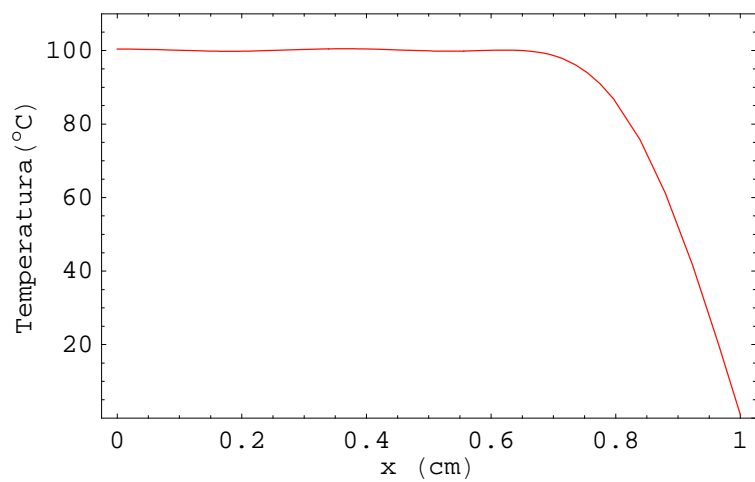
```
In[104]:= Δx=0.1;
```

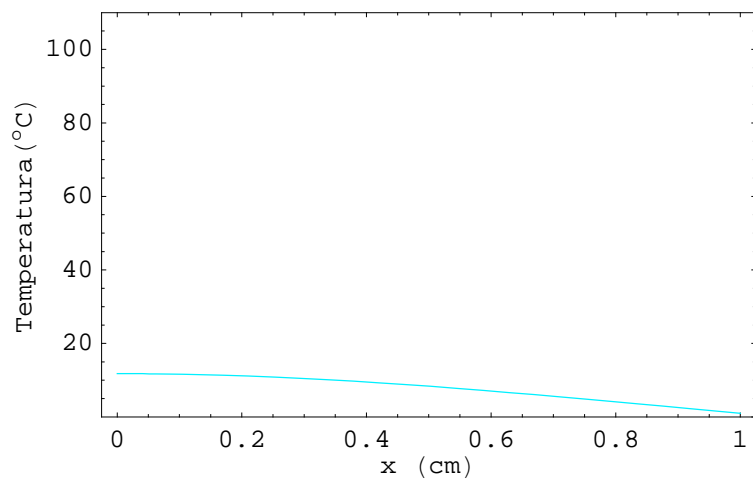
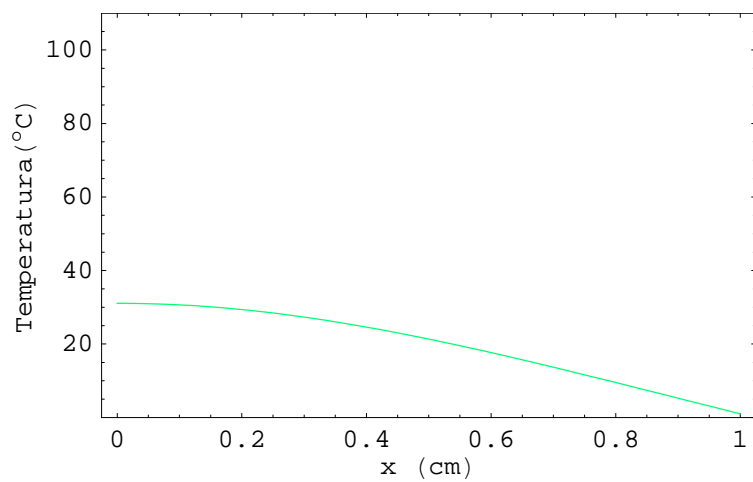
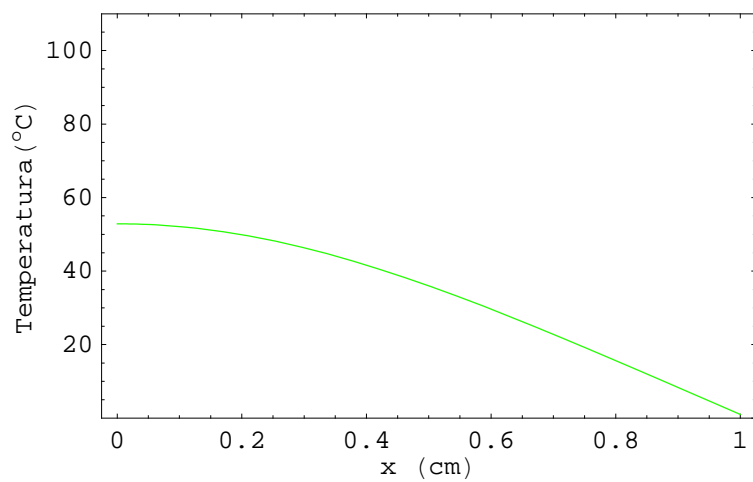
```
In[105]:= ShowAnimation[Table[Plot[T1[x,t],{x,0,1},
Frame→True,FrameLabel→{"x (cm)","Temperatura(°C)"},
PlotStyle→Hue[t],PlotRange→{0,110}],{t,0.01,0.7,0.10}]];
```



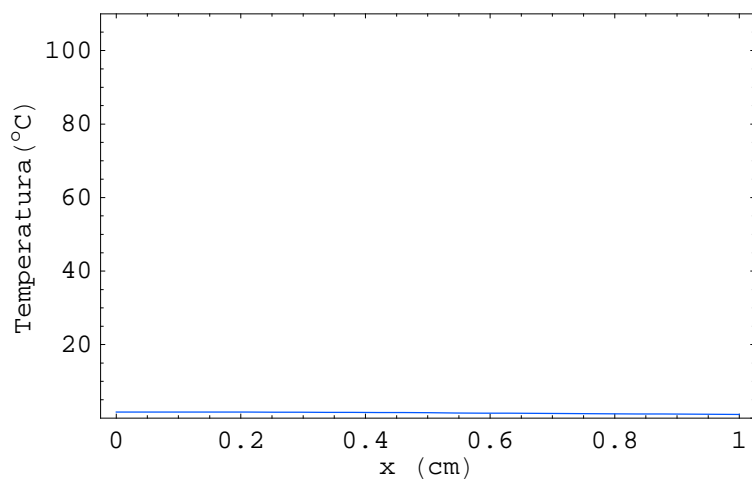




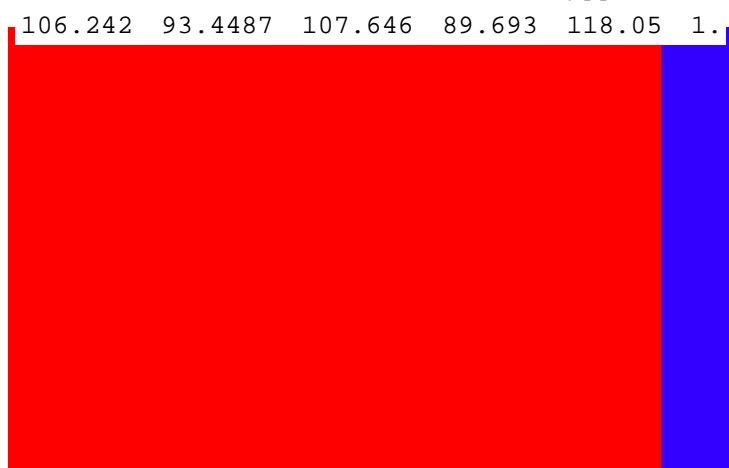




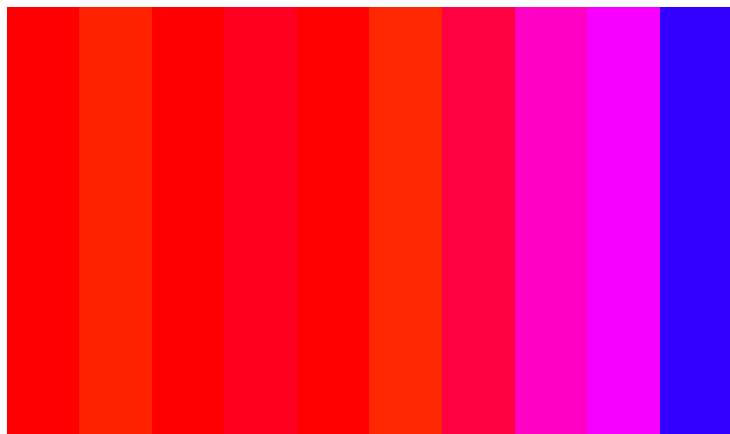




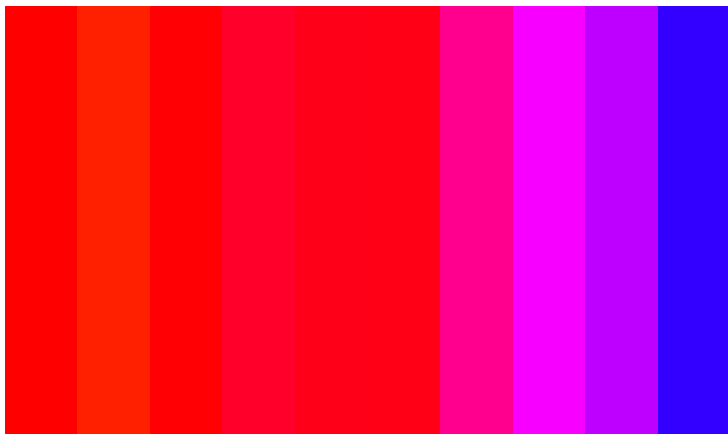
```
In[106]:= Table[Show[Graphics[
  {{RasterArray[Table[Hue[If[i == 10, 0.7, 0.7 +  $\frac{3 * T1[i / 10, j / 10]}{10 * T1[i / 10, 0]}$ ]]],
    {a, 1, 5}, {i, 1, 10}]]],
  Text[TableForm[Table[ToString[T1[x, j / 10]], {x, 0, 1, 0.2}],
    TableDirections -> {Row, Column}],
    {5, 5}, Background -> Automatic]]], {j, 0, 7, 0.2}];
```



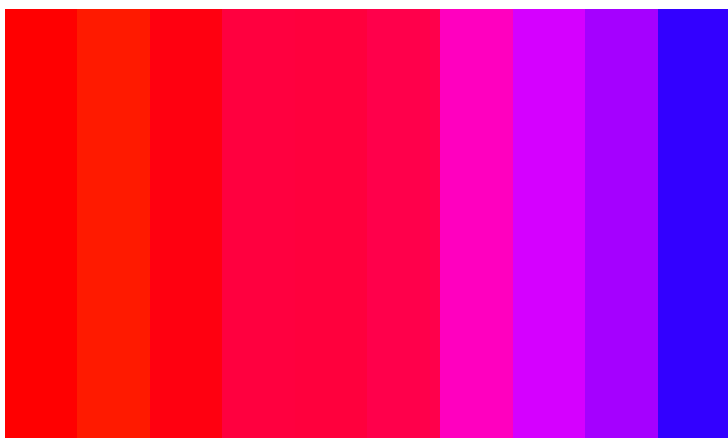
99.8531 100.376 99.7727 97.3161 67.8377 1.



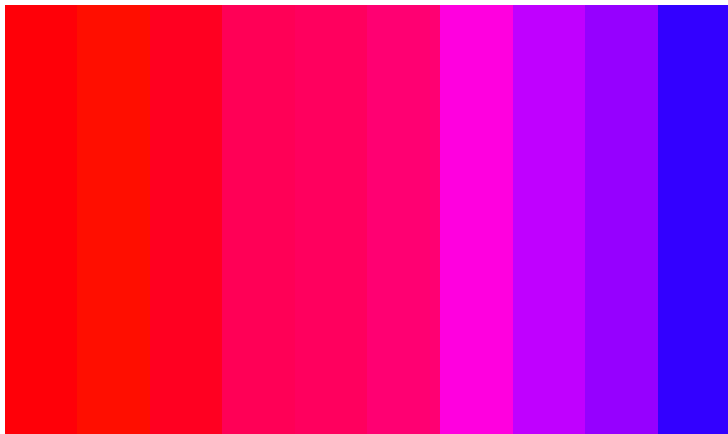
100.049 99.9639 97.7109 85.202 50.2615 1.



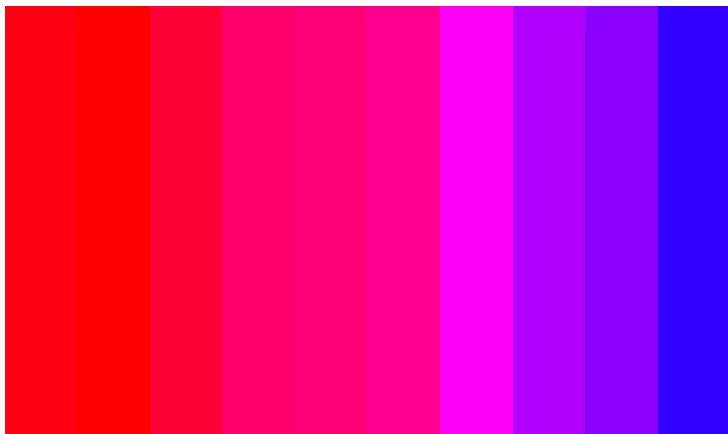
99.7726 98.7589 92.899 74.9287 41.6261 1.



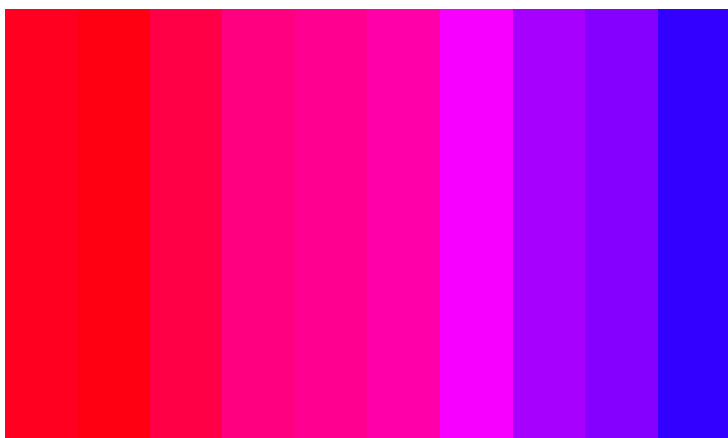
98.6583 96.4716 87.4579 67.2296 36.3158 1.



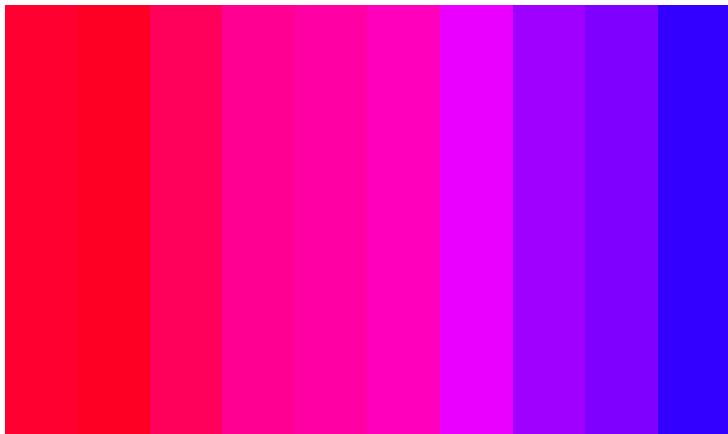
96.577 93.3842 82.2667 61.3468 32.6424 1.



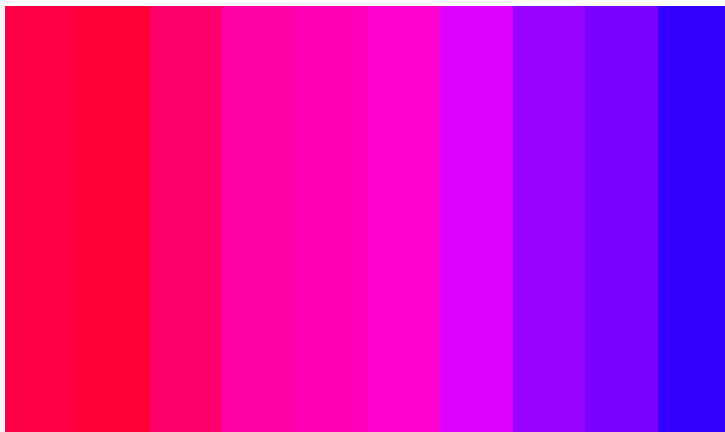
93.6621 89.7894 77.5045 56.6782 29.8984 1.



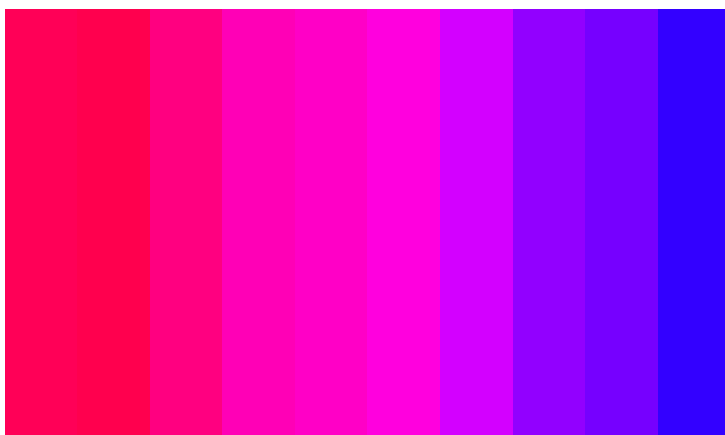
90.1299 85.8851 73.1237 52.8118 27.7207 1.



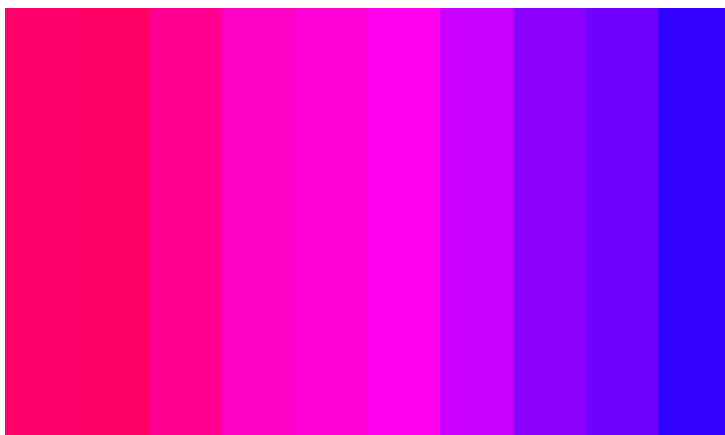
86.1791 81.7959 69.0279 49.4677 25.8966 1.

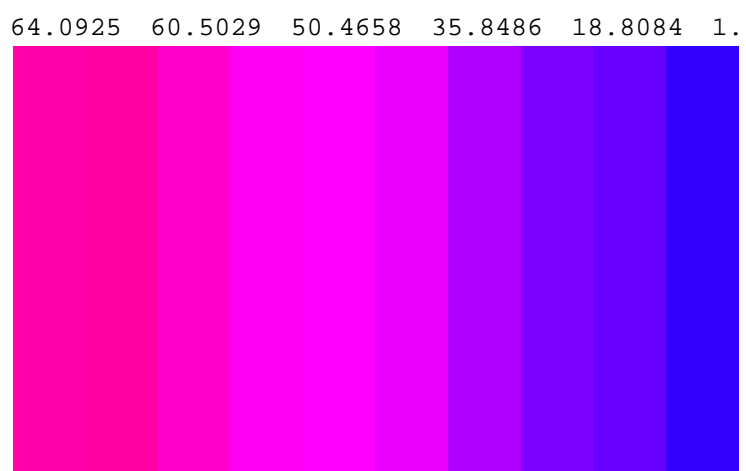
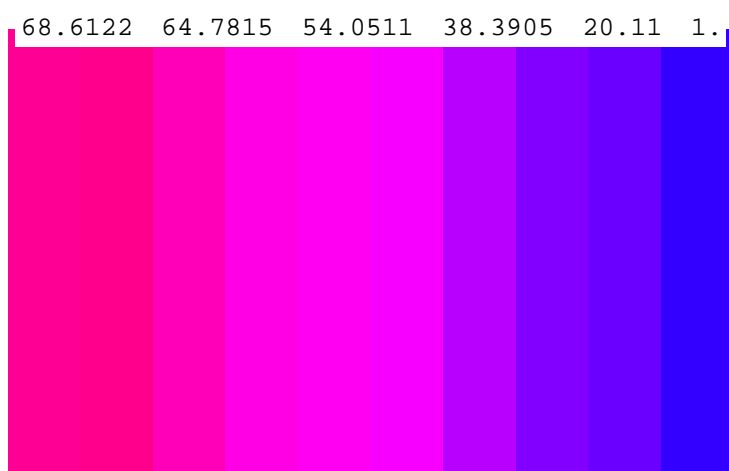
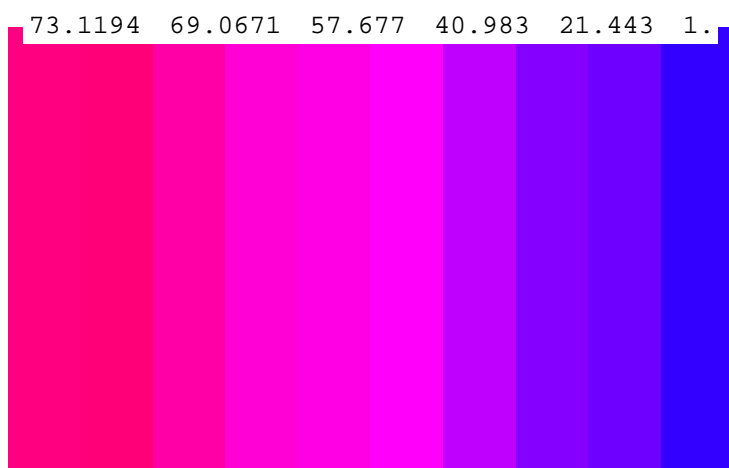


81.9617 77.5999 65.1288 46.457 24.2935 1.

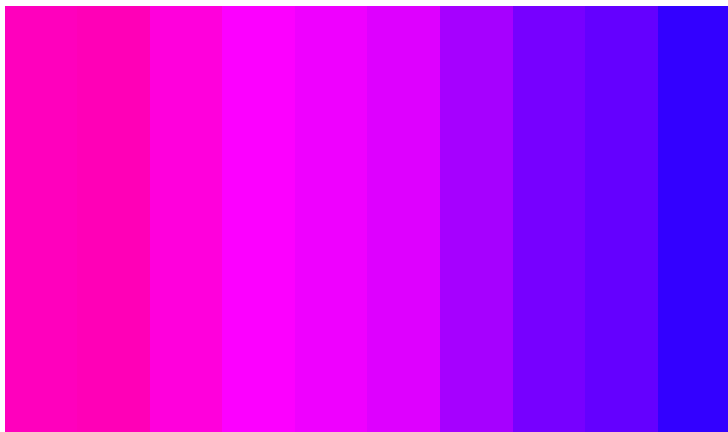


77.5846 73.3465 61.3601 43.6554 22.8266 1.

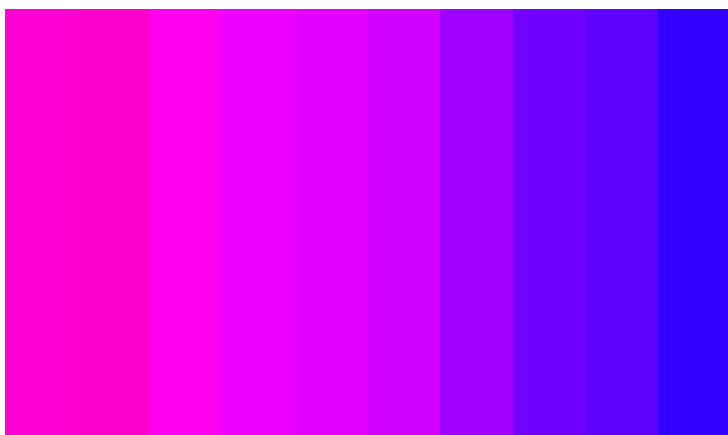




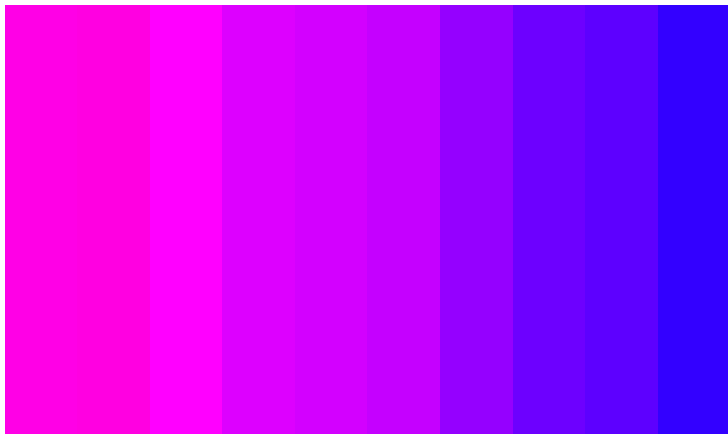
59.5789 56.2403 46.9124 33.341 17.5271 1.



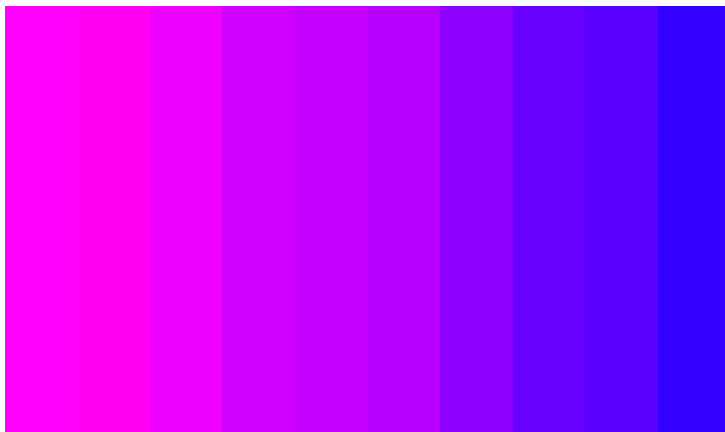
55.0841 52.0005 43.3876 30.8595 16.2607 1.



50.617 47.7893 39.8912 28.4009 15.0067 1.



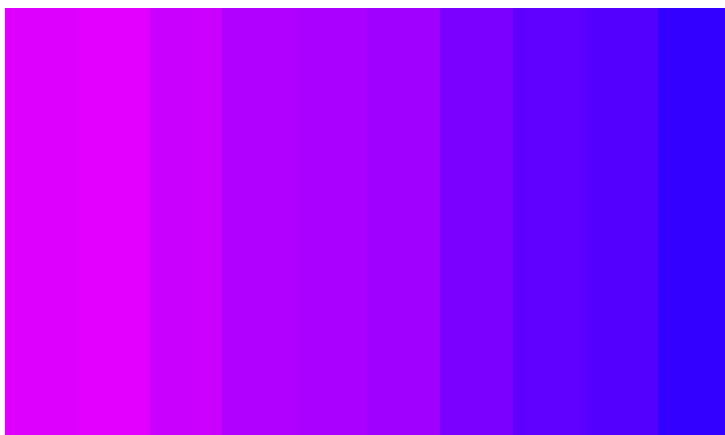
46.1854 43.6128 36.4256 25.9652 13.7646 1.



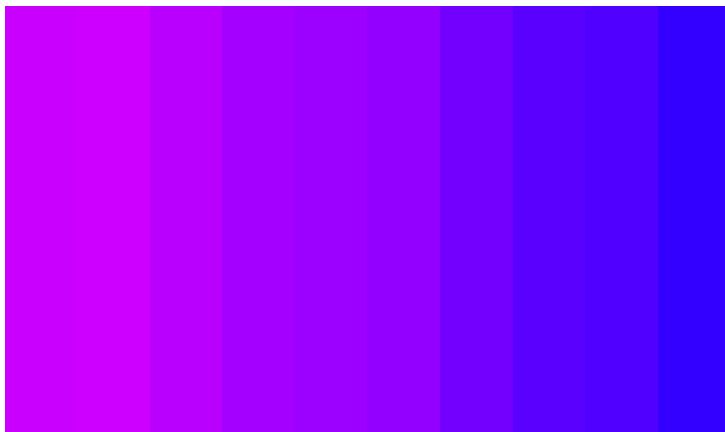
41.797 39.4775 32.9951 23.5546 12.5354 1.



37.4609 35.3914 29.6055 21.1729 11.3209 1.



33.1879 31.3648 26.2652 18.8256 10.1239 1.



28.9921 27.4109 22.9849 16.5202 8.94811 1.

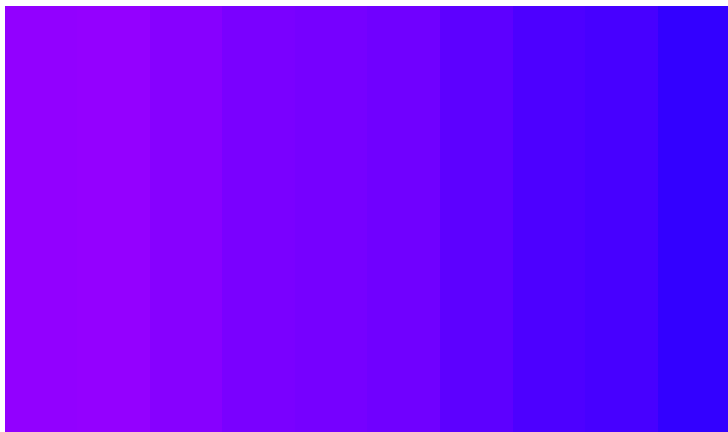


24.8926 23.5475 19.7792 14.2667 7.7986 1.

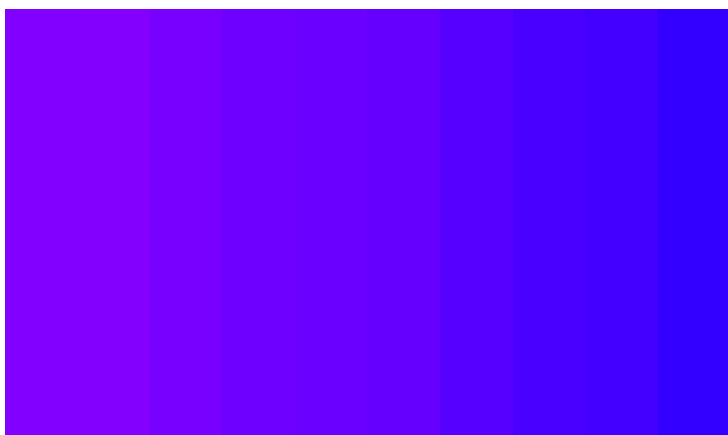




20.9154 19.799 16.6681 12.0792 6.68243 1.



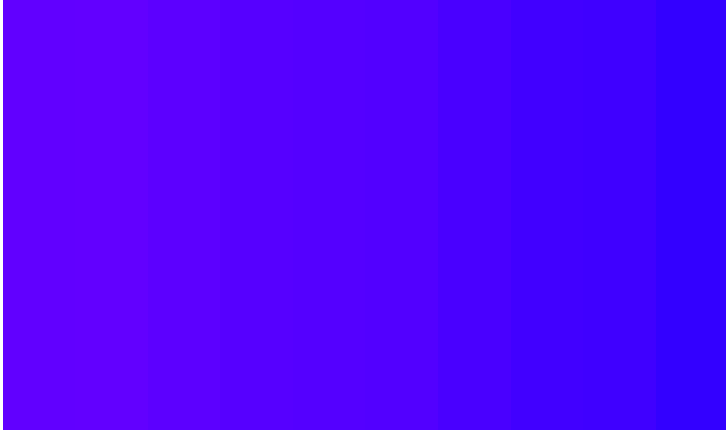
17.0969 16.1997 13.6799 9.97714 5.60943 1.



13.4891 12.7984 10.8547 7.98835 4.59362 1.



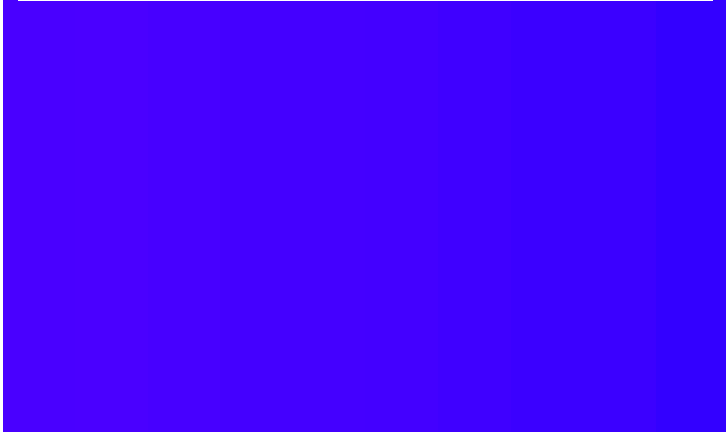
10.1673 9.66572 8.25032 6.15304 3.65526 1.




7.23723 6.90098 5.94863 4.52797 2.82297 1.




4.833 4.63046 4.05387 3.18599 2.1337 1.




3.07793 2.97072 2.66365 2.1965 1.6232 1.



1.99058 1.94061 1.79667 1.57548 1.30094 1.



1.42707 1.40586 1.34453 1.24962 1.1309 1.



1.17352 1.16498 1.14021 1.10175 1.05343 1.




1.06853 1.06517 1.05542 1.04024 1.02115 1.



1.02674 1.02543 1.02163 1.01571 1.00826 1.



1.01039 1.00988 1.0084 1.0061 1.00321 1.



## REVISÃO BIBLIOGRÁFICA

- [01] Mikhailov, M. D., Notas e discussões pessoais. 2001.
- [02] Mikhailov, M. D. and Özisik, M. N., Unified Analysis and Solutions of Heat and Mass Diffusion, Wiley-Interscience & Sons, New York, 1984.
- [03] Cotta, R.M., Integral Transform in Computational Heat and Fluid Flow, CRC Press Boca Raton, 1993.
- [04] Cotta, R.M, Mikhailov, M. D., Heat Conduction - Lumped Analysis, Integral Transforms, Symbolic Computation, John Wiley & Sons, New York, 1997.
- [05] Cotta, R.M., The Integral Transform Method in Thermal and Fluid Science and Engineering, Begell House, UK, 1998.
- [06] Glynn, J. and Gray, T., “The Beginner’s Guide to Mathematica Version 3”, Cambridge, 1st Edition. UK, (1997).
- [07] Wolfran, S., The Mathematica Book. Cambridge University Press. UK, 1998.
- [08] Gray, J. W., Mastering Mathematica. Academic Press, 2nd Edition. USA, 1998.
- [09] Pinto, M. P., Cunha, R. O., Duarte, J. B. F., Ribeiro, J. W., “Symbolic Computation Approach in Education and Heat Transfer”. XX CILAMCE. Arquivo Eletrônico (Cd-Rom). SP, SP, Nov. 1999.
- [10] Pinto, M. P., Duarte, J. B. F. and Ribeiro, J. W., “Computer-Aided Learning about Non-Linear Heat Transfer”, Int. J. of Num. Methods in Heat and Fluid Flow, (Submitted), 2000.
- [11] Ribeiro, J. W., Mikhailov, M. D., Andrade, F. E. and Duarte, J. B. F., “Ensino Assistido por Computador Aplicado à Matemática e Física Utilizando Computação Simbólica”. II INFOEDUCAR. Arquivo Eletrônico (Notebook). Fortaleza, Ce, junho, 1997.
- [12] Ribeiro, J. W., Duarte, J. B. F. and Andrade F. E. “Computação Híbrida Aplicada à Modelagem Avançada e Educação”, EINE 99 - II Escola de Informática da SBC., pp. 72-90, Edição NE / SBC, Brasil, 1999.
- [13] Ribeiro, J.W., Cotta, R.M. and Mikhailov, M.D., “Integral Transform Solution of Luikov’s Equations for Heat and Mass Transfer in Capillary Porous Media”, Int. J. Heat Mass Transfer, **36**, pp. 4467-4475, 1993.

- [14] Ribeiro, J.W. and Cotta, R.M., “On the Solution of Non-Linear Drying Problems in Capillary Porous Media Through Integral Transformation of Luikov Equations”, *Int. J. Num. Meth. in Eng.*, **38**, pp. 1001-1020, 1995.
- [15] Duarte, J.B.F. and Ribeiro, J.W., “Integral Transform Solution of Three-Dimensional Luikov Problem”, *Proc. Eurotherm53-Advanced Modelling*, 167-174, Mons, Belgium, 1997.
- [16] Santos, C. A. C, Quaresma, J. N. N. and Lima, J. A., “Convective Heat Transfer in Ducts”, *E-Papers Serviços Editoriais*, pp. 348, RJ, Brasil, 2001.
- [17] Uddin, R. and Saeed, I. M., “Structure and Growth of Tumors: The Effect of Cartesian and Spherical Geometries”. *Biotransport 98: Int. Symp. On Heat and Mass Transfer in Biological and Medical Engineering.*, Ann. N.Y. Acad. Sci. Vol. 858. Kusadasi Turkey, 1998.
- [18] Picaut, J., “Numerical modeling of urban sound fields by a diffusion process”. *J. of Applied Acoustics*. Ed. Elsevier. V.63, n.9, pp.965-991, 2002.
- [19] Buttner, J. and Simeon, B., “Runge-Kutta methods in elastoplasticity”. *Applied Num. Mathematics*, Ed. Elsevier Science B.V., V. 41, n.4 pp. 443-458, 2002.
- [20] Herdem, S. and Köksal, M., “A fast algorithm to compute the steady-state solution of nonlinear circuits by piecewise linearization”, *Computer and Electrical Engineering*, Ed. Elsevier Science Ltd., V. 28 n. 2, pp. 91-101, 2002.
- [21] Macdonald, J., Heap, N. and Mason, R., “Have I learnt it?” Evaluating skills for resource-based study using electronic resources”, *British J. of Educational Technology*, Ed. Blackwell Publishers., V. 32, N. 4 , 2001.
- [22] Scanlon, E., “Modelling Changes in Understanding: Case Studies in Physical Reasoning”. *Computer & Education*, Ed. Elsevier Science Ltd., V. 37 n.3-4, pp. 387-388, 2001.
- [23] Andrade, F. E., “Soluções de Equações Diferenciais Acopladas Pela Técnica de Transformada Integral e Computação Simbólica”. *Dissertação de Mestrado MCC/UFC*, pp. 82. Fortaleza, Ce, 1996.
- [24] Duarte, J. B. F., “Abordagem Analítico-Numérica em Meios Multidimensionais Capilares Porosos”. *Programa de Doutorado em Física / UFC*, Fortaleza, 1998.
- [25] Ramalho, J. V. A., Ribeiro, J. W., Duarte, J. B. F. and Martins, A. T., “Métodos Computacionais Híbridos no Ensino de Campos Elétricos”. *CILAMCE 2000*, edição em CD ROM. 2000.
- [26] Bleicher, L., Silva, M. M., Ribeiro, J. W, and Mesquita, M. G., “Análise e simulação de ondas sonoras assistidas por computador”. *Rev. Bras. de Ensino de Física*, pp.05, Brasil, (Submetido), 2002.
- [27] Ribeiro, J. W., et al., “Analysis of Migration Problem and Integrated Modelling Using Symbolic Computation”. *Projeto de Cooperação Internacional Brasil (CNPq)/Alemanha (PIK) WAVES*, Vigência, Mar. 98 a Mar00. 2000.
- [28] Adey, R. A., Rzevski, G., Nolan, P., “Applications of Artificial Intelligence in Eng. XIII”. *Proceedings of AIENG XIII, Comp. Mech. Publications*, pp.180. UK, 1998.

- [29] Santos, G. L., Maia, J. E. B. and Vidal, E. M. “Informática Educativa”. Ed. SEDUC-Ce., Fortaleza, Ce, 2001.
- [30] Moura, C. O. and Oliveira, A. M., “Videoconferência em Educação à Distância”, Ed. CEFET-CE, 1988.
- [31] Serra, B. A., “Uma Solução de Distribuição para Aplicações em Tempo Real no Contexto do Ensino Tecnológico à Distância”. Dissertação de Mestrado em Ciência da Computação/UFC. pp. 166. Fortaleza, Brasil, 2001.
- [32] Galvão, R. K. H., Yoneyama, T. and Nascimento, C. L. JR., “Um Ambiente para Ensino de Fundamentos de Automática e Treinamento Assistido de Operadores Humanos”, VIII SBIE. S. J. dos Campos, SP. 1997.
- [33] Hughes, J. , “Why Functional Programming Matters”, Computer Journal, 32(2), 1989.
- [34] Glynn, J. and Gray, T., “The Beginner’s Guide to Mathematica Version 3”, Cambridge, 1st Edition. UK, 1997.
- [35] Duarte, J. B. F. and Ribeiro, J. W., “Generalised Integral Transform Solutions of 3-D Luikov Problems In: Nowak A. J. et al., Advanced Computational Methods in Heat Transfer”. V. Comp. Mech. Publ. pp. 439-448, Ashurst, UK, (1998).
- [36] Demidovich, B. P. and Maron, I. A., “Computational Mathematics”. Mir Publishers, pp. 688, Moscow, URSS, 1987.
- [37] Ruggerio, M. A. G. and Lopes, V. L. E., “Cálculo Numérico - Aspectos Teóricos e Computacionais”, Ed. Makron Books, pp. 295, 1988.
- [38] Nolan, J., “Analytic Differentiation on a Digital Computer”, S.M. Thesis, MIT, Massachusetts, 1953.
- [39] Kahrimanian, H. G., “Analytic Differentiation by a Digital Computer”, M.Sc. Thesis, Temple Univ., Philadelphia, 1953.
- [40] Freire, D.G., “Interação entre Computação Simbólica e Numérica: Aplicações à Engenharia Mecânica”, X COBEM, Rio de Janeiro, Brasil, Dezembro 1989.
- [41] Juozapavièius, A., “Symbolic Computation: Systems and Applications”, Nonlinear Analysis: Modelling and Control No 3, Vilnius, Lithuania, 1998.
- [42] Gray, A., Mezzino, M. and Pinsky, M.A., “Introduction to Ordinary Differential Equations with Mathematica”, Springer-Verlag New York, USA. (1997).
- [43] Nyhoff, L. and Leestma, S. “FORTRAN 77 and Numerical Methods for Engineers and Scientists”. Prentice Hall, pp. 764, London UK, 1995.
- [44] Milne, E., “Numerical Calculus”, 1949.
- [45] Debnath L., “Integral Transforms and their Applications”. CRC Press., Boca Raton, USA, 2000.
- [46] Vilhena, M. T. and Barichello, L. B., “A Closed-Form Solution to the One-Dimensional Linear and Non-Linear Radiative Heat Transfer Problem”, Hybrid Methods in Engineering, Begell House, inc. Publishers, v.1, n.1, N. Y. USA, 1999.



- [47] Vilhena, M. T., “Combining Integral and Laplace Transform Techniques”, The Integral transform Method in Thermal and Fluids Science and Engineering, Begell House, inc. Publishers, pp. 335-352, N. Y., USA, 1998.
- [48] Zabadal, J. R. S., Marco T. M. B. de Vilhena, M. T. M. B. and Ribeiro, J. W., “Simulation of Two Dimensional Viscous Flows Using Lie Algebra”. Int. J. for Numerical Methods in Engineering, Swansea, UK, (submitted), 2001.
- [49] Monticelli, C. O, Poffal, C., A., Lima, D. R. and Gomes, M. G., “Transformadas Integrais em Domínios Não Limitados e Discretos”, Egatea, v. 25, n.6, Porto Alegre, Brasil, 1997.
- [50] Özisik, M.N. and Murray, R.L., “On the Solution of Linear Diffusion Problems with Variable Boundary Conditions”, J. Heat Transfer, v.96, pp.48-51, 1974.
- [51] Neto, F. S., Cotta, R.M. e Ribeiro, J.W, “Alternative Analytical Approach to Diffusion Problems with General Coupling at the Boundaries”, Anais do X Congr. Bras. Eng. Mecânica, pp.173-176, Rio de Janeiro, Dezembro, 1989.
- [52] Cotta, R.M., “Steady-State Diffusion with Space-Dependent Boundary Condition Parameters”, Anais do 1<sup>o</sup> Encontro Nacional de Ciências Térmicas-ENCIT 86, pp.163-166, Rio de Janeiro, 1986.
- [53] Santos, C.A C., Cotta, R.M. and Özisik, M.N., “Laminar Forced Convection Inside Externally Finned Tubes”, Anais do 2<sup>o</sup> Encontro Nacional de Ciências Térmicas-ENCIT 88, pp.87-90, Rio de Janeiro, 1988.
- [54] Aparecido, J.B., Cotta, R.M. and Özisik, M.N., “Analytical Solutions to Two-Dimensional Diffusion Problems in Irregular Geometries”, J. Franklin Inst., v.326, pp.421-434, 1989.
- [55] Aparecido, J.B., and Cotta, R.M., “Fully Developed Laminar Flow in Trapezoidal Ducts”, Anais do IX Congr. Bras. de Eng. Mecânica - IX COBEM, v.1, pp.25-28, 1987.
- [56] Cotta, R.M. and Özisik, M.N., “Diffusion Problems with General Time-Dependent Coefficients”, Rev. Bras. de Ciências Mecânicas, v.9, n.4, pp.269-292, 1987.
- [57] Cotta, R.M., “Diffusion in Media with Prescribed Moving Boundaries: - Application to Metals Oxidation at High Temperatures”, Anais do II Congresso Latino-Americano de Transferência de Calor e Massa, v.1, pp.502-513, São Paulo, 1986.
- [58] Leite, S.Q.B., Özisik, M.N. and Verghese, K., “On the Solution of Linear Diffusion Problems in Media with Moving Boundaries” , Nucl. Sci. & Eng., v.76, pp.345-350, 1980.
- [59] Cotta , R.M. and Özisik, M.N., “Transient Forced Convection in Laminar Channel Flow with Stepwise Variations of Wall Temperature”, Can. J. Chem. Eng., v.64, pp.734-742, 1986.
- [60] Cotta , R.M. and Özisik, M.N., “Laminar Forced Convection in Ducts with Periodic Variations of Inlet Temperature”, Int. J. Heat and Mass Transfer, v.29, pp.1495-1501, 1986.

- [61] Cotta, R.M., “On the Solution of Periodic Multidimensional Diffusion Problems”, *Int. Comm. Heat and Mass Transfer*, v.16, pp.549-579, 1989.
- [62] Kakaç, S., Li, W. and Cotta, R.M., “Theoretical and Experimental Study of Transient Laminar Forced Convection in a Duct with Timewise Variation of Inlet Temperature”, *ASME Winter Annual Meeting*”, *HTD*, v.123, pp.57-63, São Francisco, 1989.
- [63] Bayazitoglu, Y. and Özisik, M.N., “On the Solution of Graetz Type Problems with Axial Conduction”, *Int. J. Heat and Mass Transfer*, v.23, pp.1399-1402, 1980.
- [64] Guedes, R.O.C., Özisik, M.N. and Cotta, R.M., “Conjugated Periodic Turbulent Forced Convection in a Parallel Plate Channel”, *National Heat Transfer Conference*, *HTD* v.201, pp.63-70, San Diego, CA, August, 1992.
- [65] Alcamo, J., G.J.J. Kreileman, M.S. Krol and G. Zuidema, “Modeling the global society-biosphere-climate system. Part 1: model description and testing”, *WASP* 76, p 1-35, 1994.
- [66] Edmonds, J.A., M.A. Wise and C. MacCracken, “Advanced energy technologies and climate change: an analysis using the global change assessment model (GCAM)”, PNL-9798, UC402. Richland, Washington: Pacific Northwest Laboratory, 1994.
- [67] Magalhães, A.R., H.C. Filho, F.L. Garagorry, J.G. Gasques, L.C.B. Motion, M. da S.A. Neto, C.A. Nobre, E.R. Porto and O.K. de Rebouças, “The effects of climatic variations on agriculture in Northeast Brazil, in Parry”, in M.L., T.R: Carter and N.T, 1988.
- [68] Massachusetts Institute of Technology (MIT), Center for Global Change Science, and Center for Energy and Environmental Policy Research, 1993. Joint program on the science and policy of global change, brochure. Cambridge, Massachusetts.
- [69] Morgan, M.G. and H. Dowlatabadi, “Learning from integrated assessment of climate change”, Mimeo, Pittsburgh: Carnegie Mellon University, 1995.
- [70] Parson, E.A. and K. Fisher-Vanden, “Searching for integrated assessment: a preliminary investigation of methods, models and projects in the integrated assessment of global climatic change”, Consortium for International Earth Science Information Network (CIESIN), Univ. Michigan, 1995.