

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
PROGRAMA DE PÓS GRADUAÇÃO EM ENGENHARIA MECÂNICA

**SOLUÇÃO ANALÍTICA PARA O CÁLCULO DA CONSTANTE DE
SOLIDIFICAÇÃO DO MODELO DE RESFRIAMENTO DE SCHWARZ**

por

Jacques Duílio Brancher

Dissertação para Obtenção do Título de
Mestre em Engenharia

Porto Alegre

1994

**SOLUÇÃO ANALÍTICA PARA O CÁLCULO DA CONSTANTE DE
SOLIDIFICAÇÃO DO DO MODELO DE RESFRIAMENTO DE SCHWARZ**

por

Jacques Duílio Brancher

Bacharel em Informática

Dissertação submetida ao Corpo Docente do Programa de Pós-Graduação em Engenharia Mecânica, PROMEC, da Escola de Engenharia da Universidade Federal do Rio Grande do Sul, como parte dos requisitos necessários para obtenção do Título de

Mestre em Engenharia

Área de Concentração: Fenômenos de Transporte

Orientador:

Prof. Dr. Marco Túllio Menna Barreto de Vilhena

Aprovada por:

Prof. Dr. Milton Antônio Zaro

Prof. Dr. Arno Müller

Prof. Dr. Horácio Vielmo

Prof. Dr. Sérgio Viçosa Möller

Coordenador do PROMEC

Porto Alegre, 28 de março de 1994

I

AGRADECIMENTOS

Aos meus pais, Duílio (In Memmorian) e Osmilda, pela vida.

À minha irmã Rosa, pela sua presença constante, e pelo apoio moral.

À Ana Cristina, minha namorada, pelo apoio nas horas difíceis, pelas alegrias nas horas de descanso, pelo companheirismo, e principalmente por acreditar.

Ao Amigo, Tutor, Mestre, Milton, pelo apoio nestes 6 anos de laboratório em que estou no Laboratório de Instrumentação, pelos ensinamentos por ele transmitidos, pelo companheirismo e pela dedicação.

Ao Amigo e Orientador Vilhena, pelos ensinamentos nas disciplinas do curso, pela proposta de tese, pelo apoio nas horas de dúvida, pelas dicas na preparação da tese, pelas correções no texto final.

Ao Rubem, por ter me ensinado os primeiros passos no Adomian.

Ao Augusto e a Cynthia pelas explicações matemáticas.

Aos funcionários da biblioteca e de periódicos da Escola de Engenharia pelo atendimento sempre atencioso.

Ao CNPq pelo apoio financeiro.

RESUMO

O presente trabalho tem por objetivo apresentar uma solução analítica para o cálculo da Constante de Solidificação do modelo de Schwarz. Neste sentido, o método da Decomposição proposto por Adomian [1] é usado para resolver a equação transcendental do modelo de Schwarz. Os resultados obtidos por este método são comparados com resultados numéricos, para ligas com calor latente de fusão na faixa de 0,4 à 2.

ABSTRACT

The purpose of this work is to present an analytical solution for the solidification constant of the Schwarz's model. To reach this goal, the decomposition method, proposed by Adomian is used to solve the transcendental equation of the Schwarz's model. The numerical results obtained by this method are compared with numerical results for alloys with latent heat of fusion ranging from 0,4 to 2.

LISTA DE SÍMBOLOS E UNIDADES

Símbolo		Unidade
α	Difusividade térmica = $k/(\rho c)$	$\frac{m^2}{s}$
ρ	Densidade	$\frac{kg}{m^3}$
η	Coordenada adimensional	
ξ	Distância entre o molde e o metal líquido.	m
a	Difusividade térmica	$\frac{m^2}{s}$
b_i	Mede a razão entre a condutividade térmica e a raiz quadrada da difusividade térmica no meio i	
c	Calor específico	$\frac{J}{kgK}$
H	Calor latente de fusão	
H^*	Calor latente de fusão adimensional	
k	Condutividade térmica	$\frac{W}{mK}$
M	Parâmetro do molde definido como $M=b_2/b_1$	
m	Razão entre b_3/b_2	
n	Razão entre as difusividades térmicas do metal sólido e líquido	
S^*	Super resfriamento adimensional	
T	Temperatura	$^{\circ}C$
t	Variável temporal	s
x	Variável de posição em coordenadas cartesianas	m

Subíndices

1 -molde

2 -metal sólido

3 -metal líquido

f - fusão

i - Interface metal-molde

o - estado inicial

v - vazão do metal líquido

Índice	pág.
1. Introdução	1
1.1. Problema de Stefan	2
1.2. Tipos de fronteira	2
1.3. Estado da Arte	5
2. O modelo de Schwarz	7
2.1. Descrição do modelo físico de Schwarz	8
2.2. Método da Semelhança	9
2.3. Solução do modelo de Schwarz pelo método da Semelhança	13
3. O Método da Decomposição	22
3.1. Introdução	22
3.2. O método da Decomposição	23
3.3. Formulação dos A_n	25
4. Solução do modelo de Schwarz	29
5. Resultados	33
6. Conclusões	48
Bibliografia	50
Apêndice I	52
I.1. FTP: File Transfer Program	52
I.1.1. Comandos de transferência	53
I.1.2. Comandos remotos	55
I.1.3. Comandos locais	57
I.2. TELNET	58
I.2.1. Modo de entrada	58

I.2.2. Modo de comandos	58
Apêndice II - O Software Reduce	60
Apêndice III - Softwares para determinação da Constante de Resfriamento	62

Índice de figuras

Figura 1- Esquema representativo do modelo de Schwarz: distribuição de temperaturas em um instante t	8
Figura 2 - Metais e respectivos H^*	33
Figura 3 - Gráfico comparativo entre o método de Newton e o método Proposto, com $S^* = 0$ e $M=1$	35
Figura 4 - Gráfico comparativo entre o método de Newton e o método Proposto, com $S^* = 0$ e $M=5$	36
Figura 5 - Gráfico comparativo entre o método de Newton e o método Proposto, com $S^* = 0$ e $M=10$	37
Figura 6 - Gráfico comparativo entre o método de Newton e o método Proposto, com $S^* = 0$ e $M=20$	38
Figura 7 - Gráfico comparativo entre o método de Newton e o método Proposto, com $S^* = 0,05$ e $M=1$	39
Figura 8 - Gráfico comparativo entre o método de Newton e o método Proposto, com $S^* = 0,05$ e $M=5$	40
Figura 9 - Gráfico comparativo entre o método de Newton e o método Proposto, com $S^* = 0,05$ e $M=10$	41
Figura 10 - Gráfico comparativo entre o método de Newton e o método Proposto, com $S^* = 0,08$ e $M=1$	42
Figura 11 - Gráfico comparativo entre o método de Newton e o método Proposto, com $S^* = 0,08$ e $M=5$	43

Figura 12 - Gráfico comparativo entre o método de Newton e o método Proposto, com $S^* = 0,08$ e $M=10$	44
Figura 13 - Gráfico comparativo entre os tempos de processamento entre o método proposto e o método de Newton para $S^* = 0$	45
Figura 14 - Gráfico comparativo entre os tempos de processamento entre o método proposto e o método de Newton para $S^* = 0,05$	46
Figura 15 - Gráfico comparativo entre os tempos de processamento entre o método proposto e o método de Newton para $S^* = 0,08$	47

1. Introdução

A solidificação é uma transformação de fase muito familiar a todos, pois muitos dos produtos produzidos pelo homem possuem em algum estágio da sua fabricação a solidificação, seja na fundição propriamente dita ou no lingotamento para posterior trabalho.

O modelo de solidificação de Schwarz, por exemplo, pode descrever o comportamento da solidificação de um lingote. Ele apresenta a distribuição das temperaturas do processo em três meios diferentes (molde, metal sólido e metal líquido), em qualquer instante de tempo.

Este modelo possui uma característica especial, que é a fronteira livre. Este tipo de problema é conhecido na literatura como problema de Stefan. Os problemas com fronteira móvel geram uma dificuldade adicional decorrente do fato dos domínios sólido e líquido variarem no tempo. São encontrados em problemas de calor, difusão, ou processos de termo-difusão acompanhados por mudança de fase.

No capítulo 1, é definido o problema de Stefan, com os tipos de condição de fronteira que ocorrem em problemas de transferência de calor. É também apresentado o problema resolvido por Lamé-Clapeyron, que foi o primeiro problema de fronteira livre resolvido. Além disto, é mostrado o do Estado da Arte na resolução de problemas de solidificação.

No capítulo 2, é apresentada a solução proposta por Schwarz [16], que é um problema de Newmann, para as distribuições de temperatura nos três meios. É

também mostrado o método da Semelhança [18], para obtenção da solução do modelo de Schwarz.

No capítulo 3, o Método da Decomposição, proposto por Adomian [2] é apresentado, que é utilizado para o cálculo de soluções analíticas para problemas não-lineares, estocásticos, e outros problemas em que não é possível efetuar simplificações nos modelos.

No capítulo 4, é mostrada a aplicação do método de Decomposição para determinação da Constante de Resfriamento, e apresentada uma formulação analítica para esta constante.

No capítulo 5, são apresentados os resultados numéricos obtidos para o método proposto, comparando-se estes resultados com valores obtidos através de métodos numéricos.

No capítulo 6, são apresentadas as conclusões do trabalho, bem como as sugestões de trabalho futuro nesta linha de pesquisa.

1.1. Problema de Stefan

Entende-se por problema de Stefan uma classe de problemas matemáticos que descrevem transferência de calor, difusão, ou mesmo processos de termo-difusão acompanhados por troca de fase no meio e absorção ou produção de calor latente. A característica principal destes problemas é a existência de uma fronteira livre entre as diferentes fases que a princípio não são conhecidas.

Assim, considere-se o problema de Stefan, constituído da transferência de calor em um corpo composto de duas fases, geralmente uma sólida e uma líquida (Exemplo: gelo e água em processos de solidificação ou fusão). As funções que

representam as temperaturas das fases satisfazem as equações de calor correspondentes. A superfície de separação que se encontra a uma temperatura constante, pode variar no tempo e é uma incógnita suplementar do problema sobre a qual existe uma condição que surge do **princípio de conservação de energia**.

O interesse e a dificuldade do problema se deve à presença de uma superfície de separação entre as fases, que se chama **fronteira livre** do problema, cuja determinação é de fundamental importância prática.

1.2. Tipos de fronteira

Os problemas de fronteira para a equação do calor ou de difusão são classificados como: fronteira fixa, móvel e livre, sendo que a fronteira livre pode ser do tipo explícita ou implícita [9].

Os problemas de fronteira fixa são aqueles nos quais o domínio durante todo o processo é fixo. Por outro lado, os problemas de "fronteira móvel" são aqueles em que o domínio é variável com o tempo, mediante uma lei de movimento conhecida inicialmente. O desconhecimento da fronteira, ou parte dela, induz à necessidade matemática de impor novas condições às funções incógnitas, as quais dependem do problema físico em estudo. Em geral, a nova condição a impor é deduzida do princípio de conservação de energia aplicado através da fronteira. Esta fronteira é uma incógnita suplementar do problema, a qual recebe o nome de "fronteira livre" do problema em questão [9].

Os problemas de fronteira livre para a equação de calor se dividem em duas classes: de tipo explícito e de tipo implícito. Nos problemas do primeiro tipo, encontra-

se explicitamente a velocidade da fronteira livre, enquanto que no último a velocidade não é encontrada explicitamente.

O primeiro problema de fronteira livre foi apresentado em 1891 por Lamé-Clapeyron [17] que resolveu o problema da solidificação de um globo líquido (a Terra) da seguinte maneira:

Supõe-se que a Terra seja uma esfera onde se verificam as seguintes hipóteses:

1. Inicialmente a Terra era líquida e composta por uma única substância, a qual se encontrava a uma temperatura de fusão T_f .
2. O resfriamento acontece da superfície para o centro, com a temperatura passando de T_f à T_0 com $T_0 < T_f$.
3. A crosta sólida formada em nossos dias não possui uma espessura considerável se comparada ao raio terrestre [17].

A partir das hipóteses acima consideradas, Lamé-Clapeyron resolveram o problema da solidificação de um material semi-infinito, representado por $x > 0$, que inicialmente se encontra na fase líquida na temperatura de fusão T_f e que a partir da fronteira $x=0$ resfria até uma temperatura T_0 , inferior à temperatura de fusão. Por considerações físicas, deduz-se que para cada tempo $t > 0$, existirá um ponto $x = s(t)$ que separará a fase sólida, representada pelo intervalo $(0, s(t))$, e a fase líquida, representada pelo intervalo $(s(t), +\infty)$, que se encontra na temperatura de fusão.

Este problema é conhecido na literatura como problema de Stefan a uma fase, porque a fase líquida se encontra à temperatura de fusão do material.

Os problemas apresentados na literatura como Stefan a duas fases são chamados desta forma quando a temperatura do material líquido estiver acima da temperatura da sua temperatura de fusão, ou abaixo, para o caso de fusão.

1.3. Estado da Arte

O grande desafio na simulação em solidificação está em resolver todas as equações que compõem o problema: calor, massa e quantidade de movimento, com convecção natural ou forçada. As últimas pesquisas conduzem para dois grupos de modelos largamente utilizados, que são o modelo em duas fases e o modelo de mistura.

O modelo de duas fases divide o domínio em duas partes, uma referente ao sólido e outra referente ao líquido. As equações são resolvidas separadamente, sendo que o acoplamento entre os dois domínios é realizado através das condições de contorno da interface sólido-líquido. O equacionamento detalhado da interface permite a não adoção da hipótese simplificativa da existência de equilíbrio termodinâmico. Este tipo de abordagem permite portanto uma descrição mais fiel da realidade [18].

Esta técnica exige uma precisa localização da interface ao longo de todo o processo, tarefa nem sempre fácil, principalmente no caso de ligas, já que a zona esponjosa apresenta-se usualmente com uma complexa geometria. A necessidade de malhas computacionais móveis, ou da imobilização do domínio através da transformação de coordenadas exige, na maioria dos casos, hipóteses simplificativas em relação à geometria da interface [18].

O modelo de misturas descreve o fenômeno levando em conta nas equações governantes, as frações de sólido e de líquido, que ocorrem em cada local

do domínio. Desta forma, todo o domínio pode ser tratado pelo mesmo conjunto de equações conservativas. São as chamadas formulações contínuas, que apresentam a vantagem de permitirem o uso de malha computacional única em todo o domínio de solução. Também é dispensada a necessidade do rastreamento da frente de mudança de fase, embora seja possível localizá-la a qualquer momento [18].

2. O Modelo de Schwarz

Neste capítulo apresenta-se o modelo proposto por Schwarz [16], que descreve o comportamento da temperatura em qualquer instante para três meios distintos: molde, metal sólido e metal líquido.

O modelo proposto por Schwarz, possui diversas hipóteses simplificativas tais como modelo unidimensional, corpos semi-infinitos, ausência de resistência térmica de contato entre a peça e o molde e ausência de resfriamento por convecção natural e radiação [18]. Sua importância no contexto da solidificação, além de histórica, é, até hoje, a de ser um modelo utilizado tanto pela sua simplicidade, como pelo fato de fornecer a temperatura em qualquer ponto e instante na peça a ser resfriada e no molde, em forma analítica.

Inicialmente, apresenta-se a descrição do modelo físico de Schwarz, a sua representação esquemática, hipóteses simplificativas e as condições de contorno.

Na seqüência é mostrado o Método de Semelhança, que é utilizado para obtenção da solução do modelo de Schwarz.

Finalmente apresentam-se os detalhes de solução do modelo de Schwarz pelo método da Semelhança.

2.1. Descrição do modelo físico de Schwarz

Schwarz [17], concebeu um modelo físico para a solidificação esquematizado na figura 1.

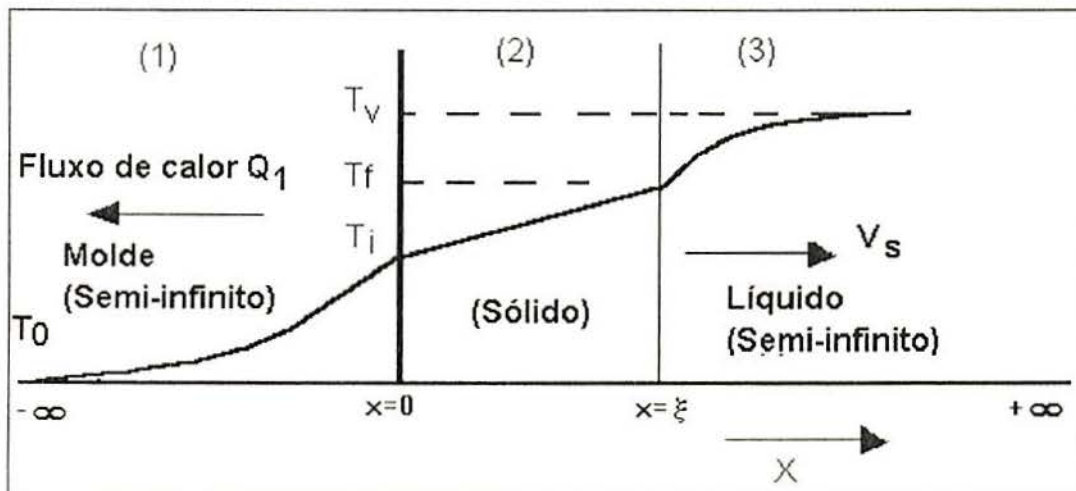


Figura 1: Esquema representativo do modelo de Schwarz: distribuição de temperaturas em um instante t .

No instante inicial todo o líquido está a uma temperatura T_v , na qual o material a ser solidificado é vazado no molde. O molde apresenta inicialmente uma temperatura T_0 . Decorrido um certo tempo, o sistema metal e molde, assume a configuração de distribuição de temperatura mostrada na figura 1, modelado por Schwarz, considerando as diversas hipóteses simplificativas [12]:

1. A interface metal-molde é plana.
2. Tanto o metal como o molde são considerados semi-infinitos na direção x ; (nas demais direções não existe contato entre o metal e o molde).

3. No instante $t=0$, a temperatura da massa líquida é igual a temperatura de vazamento T_v e no molde é igual a uma temperatura inicial T_0 .
4. Não existe nenhuma outra reação exotérmica dentro da massa líquida ou entre ela e as paredes do molde, a não ser o calor de fusão.
5. Não existe fluxo de massa.
6. As constantes térmicas dos materiais são constantes em função da temperatura.
7. O contato metal-molde é perfeito, não havendo portanto queda de temperatura na interface metal-molde.
8. No molde, para $x = -\infty$ a temperatura permanece com valor constante durante todo o processo. Esta temperatura é chamada de $T_0 = T$ ambiente.
9. No molde, para $x = 0$ e tempo $t \geq 0$, a temperatura do molde permanece constante. Esta temperatura é denominada de temperatura de interface.
10. Na interface sólido-líquido a temperatura do sólido é constante e igual a T_f , que é a temperatura de fusão do metal.
11. No metal líquido, em $x = +\infty$, a temperatura permanece constante e com valor igual a T_v , que é a temperatura de vazamento do metal líquido.

2.2. Método da Semelhança

Para que exista uma solução pelo Método da Semelhança é necessário que a equação diferencial e todos os dados (iniciais e de contorno) possam ser expressos em termos de uma só variável independente (chamada de variável de semelhança) [17].

Geralmente isto ocorre em problemas que apresentem as seguintes características:

1. Domínio semi-infinito.
2. Quando uma fase está inicialmente presente.
3. Quando a temperatura no instante $t=0$ é constante.
4. Quando a temperatura ou o fluxo de calor em $x=0$ é constante.

Considera-se então o seguinte problema, que será resolvido pelo Método da Semelhança, descrito pela equação (2.1):

$$\rho c \frac{\partial T}{\partial t} = k \frac{\partial^2 T}{\partial x^2} \quad (2.1)$$

onde $T = T(x, t)$ é a temperatura na posição x e no instante t

Esta equação pode ser resolvida através da seguinte transformação de variáveis [18]:

$$\eta = \frac{x}{2\sqrt{\alpha t}} \quad (2.2)$$

onde η é chamada de variável de semelhança. Utilizando-se a regra da cadeia para derivação, obtém-se as seguintes expressões para as derivadas parciais em relação a x e t que aparecem na equação (2.1), ou seja:

$$\frac{\partial T}{\partial t} = -\frac{\eta}{2t} \frac{\partial T}{\partial \eta} \quad (2.3)$$

e

$$\frac{\partial^2 T}{\partial x^2} = \left(\frac{\partial^2 T}{\partial \eta^2} \frac{1}{2\sqrt{\alpha t}} \right) \frac{1}{2\sqrt{\alpha t}} \quad (2.3a)$$

Substituindo-se (2.3) e (2.3a) na equação (2.1), resulta:

$$T'' + 2\eta T' = 0 \quad (2.4)$$

onde foi usada a seguinte notação: $T' = \frac{dT}{d\eta}$, e $T'' = \frac{d^2T}{d^2\eta}$

Considerando-se a transformação de variáveis $T' = \omega$, a equação (2.4) é então reescrita como:

$$\frac{d\omega}{d\eta} + 2\eta\omega = 0 \quad (2.5)$$

cuja solução é dada por:

$$\omega = C e^{-\eta^2} \quad (2.6)$$

Lembrando que $T' = \omega$, resulta que:

$$\frac{dT}{d\eta} = C e^{-\eta^2} \quad (2.7)$$

e, em conseqüência:

$$T(x, t) = C_1 \int_0^x e^{-\eta^2} d\eta + C_2 \quad (2.8)$$

A equação (2.8) pode ser reescrita em termos da função erro como:

$$T(x,t) = A + B \operatorname{erf}(\eta) \quad (2.8a)$$

com

$$A = C_2 \quad (2.8b)$$

e

$$B = \frac{\sqrt{\pi} C_1}{2} \quad (2.8c)$$

onde a função erro é definida como:

$$\operatorname{erf}(\varphi) = \frac{2}{\sqrt{\pi}} \int_0^{\varphi} e^{-u^2} du$$

Para a aplicação das condições de contorno associadas ao modelo de Schwarz, é necessário o conhecimento das propriedades da função erro, descritas abaixo:

1. $\operatorname{erf}(0) = 0$
2. $\operatorname{erf}(+\infty) = 1$
3. $\operatorname{erf}(-x) = -\operatorname{erf}(x)$

$$4. \frac{d(\operatorname{erf}(x))}{dx} = \frac{2}{\sqrt{\pi}} e^{-x^2} > 0, \quad \forall x \in \mathfrak{R}$$

$$5. \frac{d^2(\operatorname{erf}(x))}{dx^2} = -2x \frac{d(\operatorname{erf}(x))}{dx} < 0 \quad \forall x > 0$$

6. Representação em série de potências:

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n+1}}{(2n+1)n!}, \text{ cujo região de convergência é: } \mathfrak{R}$$

2.3. Solução do modelo de Schwarz pelo método da Semelhança

A equação que descreve o modelo proposto por Schwarz é dada por:

$$\frac{\partial T_L}{\partial t} = a_L^2 \frac{\partial^2 T_L}{\partial x^2} \quad L = 1, 2, 3, \quad (2.9)$$

cuja solução pelo método da Semelhança discutido na seção anterior (2.2) é dado por:

$$T_L(x, t) = A_L + B_L \operatorname{erf}\left(\frac{x}{2\sqrt{a_L t}}\right), \quad (2.10)$$

onde o sub-índice L descreve a região do domínio; ou seja, L=1 corresponde ao molde, L=2 ao metal sólido e L=3 ao metal líquido.

As condições de contorno associadas às hipóteses simplificativas do modelo (8) e (9) da seção 2.2, são descritas como:

$$T_1 = T_0 \quad x = -\infty \quad t \geq 0 \quad (2.11)$$

$$T_1 = T_i \quad x = 0 \quad t \geq 0 \quad (2.11a)$$

Aplicando a condição (2.11) na solução descrita pela equação (2.10), com $L=1$, resulta

$$\lim_{x \rightarrow -\infty} T_1(x, t) = A_1 - B_1 = T_0 \quad (2.12)$$

Por outro lado, aplicando (2.11a) em (2.10), com $L=1$, tem-se:

$$T_1(0, t) = A_1 = T_i \quad (2.13)$$

assim, das equações (2.12) e (2.13), encontra-se o valor da constante B_1 :

$$B_1 = T_i - T_0 \quad (2.14)$$

Substituindo-se os valores das constantes descritos em (2.13) e (2.14) na equação (2.10), com $L=1$, resulta a seguinte expressão que descreve a solução do modelo de Schwarz para a distribuição de temperatura no molde, ou seja:

$$T_1(x, t) = T_i + (T_i - T_0) \operatorname{erf}\left(\frac{x}{2\sqrt{a_1 t}}\right) \quad (2.15)$$

As condições de contorno associadas as hipóteses simplificativas (9) e (10) da seção 2.2, relativas a distribuição de temperatura no metal sólido, são as seguintes:

$$T_2 = T_i, \quad x = 0, \quad t \geq 0 \quad (2.16)$$

$$T_2 = T_f, \quad x = \zeta, \quad t \geq 0 \quad (2.16a)$$

aplicando-se a condição (2.16) na solução descrita pela equação (2.10), com $L=2$, resulta:

$$T_2(0, t) = A_2 = T_i \quad (2.17)$$

utilizando-se (2.16a) em (2.10), com $L=2$, obtém-se:

$$T_2(\zeta, t) = A_2 + B_2 \operatorname{erf}\left(\frac{\zeta}{2\sqrt{a_2 t}}\right) = T_f \quad (2.18)$$

Sendo, a temperatura T_2 na equação (2.18) constante, resulta que o argumento $\frac{\zeta}{2\sqrt{a_2 t}}$

também deva ser constante. Sendo assim, define-se o parâmetro $\varphi = \frac{\zeta}{2\sqrt{a_2 t}}$,

denominado constante de solidificação, que relaciona a espessura de material solidificado e a raiz quadrada do tempo. Assim, a equação que determina a espessura de metal solidificado em um determinado instante de tempo t é dada por:

$$\zeta = 2\varphi\sqrt{a_2 t} \quad (2.19)$$

Substituindo-se o termo $\frac{\zeta}{2\sqrt{a_2 t}}$ por φ e o valor da constante A_2 obtida em (2.17) na equação (2.18), resulta:

$$B_2 = \frac{T_f - T_i}{\operatorname{erf}(\varphi)} \quad (2.20)$$

Assim, da equação (2.10), considerando $L=2$ e as constantes descritas em (2.17) e (2.20), resulta a seguinte expressão que descreve a solução do modelo de Schwarz para a distribuição de temperatura no metal sólido, ou seja:

$$T_2(x, t) = T_i + \left(\frac{T_f - T_i}{\operatorname{erf}(\varphi)}\right) \operatorname{erf}\left(\frac{x}{2\sqrt{a_2 t}}\right) \quad (2.21)$$

As condições de contorno associadas as hipóteses simplificativas (10) e (11) da seção 2.2 para a distribuição de temperatura no metal líquido são as seguintes:

$$T_3 = T_f, \quad x = \zeta \quad t \geq 0 \quad (2.22)$$

$$T_3 = T_v, \quad x = +\infty \quad t \geq 0 \quad (2.22a)$$

da condição (2.22) aplicada em (2.10), com $L=3$, tem-se que:

$$T_3(\zeta, t) = A_3 + B_3 \operatorname{erf}\left(\frac{\zeta}{2\sqrt{a_3 t}}\right) = T_f \quad (2.23)$$

considerando ζ dado pela equação (2.19) na equação (2.23), e simplificando-se os termos em comum, obtém-se:

$$T_3(\zeta, t) = A_3 + B_3 \operatorname{erf}\left(\sqrt{\frac{a_2}{a_3}} \varphi\right) \quad (2.24)$$

Definindo-se $n = \sqrt{\frac{a_2}{a_3}}$ e substituindo-se na equação (2.24), resulta:

$$T_3(\zeta, t) = A_3 + B_3 \operatorname{erf}(n\varphi) = T_f \quad (2.25)$$

Substituindo a condição (2.22a) em (2.10), com $L=3$, obtém-se:

$$\lim_{x \rightarrow +\infty} T_3(x, t) = A_3 + B_3 = T_v \quad (2.26)$$

então, resolvendo-se o sistema de equações dado por (2.25) e (2.26), resulta que:

$$B_3 = \frac{T_v - T_f}{(1 - \operatorname{erf}(n\phi))} \quad (2.27)$$

e

$$A_3 = T_v - \left(\frac{T_v - T_f}{(1 - \operatorname{erf}(n\phi))} \right) \quad (2.28)$$

Substituindo-se os valores das constantes descritas em (2.27) e (2.28) na equação (2.10), com $L=3$, resulta a expressão, que descreve a solução do modelo de Schwarz para a distribuição de temperatura no metal líquido:

$$T_3(x,t) = T_v - \left(\frac{T_v - T_f}{(1 - \operatorname{erf}(n\phi))} \right) \left(1 - \operatorname{erf} \left(\frac{x}{2\sqrt{a_3 t}} \right) \right) \quad (2.29)$$

Para a determinação da temperatura da interface metal molde (T_i), considera-se a conservação de fluxo de calor na interface, ou seja::

$$k_1 \frac{\partial T_1}{\partial x} \Big|_{x=0} = k_2 \frac{\partial T_2}{\partial x} \Big|_{x=0} \quad (2.30)$$

Fazendo-se $x=0$ nas expressões de $T_L(x,t)$ descritas pela equação (2.10), para $L=1,2$, tem-se:

$$k_1 \frac{B_1}{\sqrt{\pi a_1 t}} = k_2 \frac{B_2}{\sqrt{\pi a_2 t}} \quad (2.31)$$

Simplificando o termo $\sqrt{\pi t}$ e definindo-se $b_i = \frac{k_i}{\sqrt{a_i}}$

$$\frac{b_2}{b_1} = \frac{B_1}{B_2} \quad (2.32)$$

Substituindo os valores de B_1 dado em (2.14) e B_2 dado em (2.20) na equação (2.32), resulta:

$$M = \frac{(T_i - T_0)}{(T_f - T_i)} \text{erf}(\varphi) \quad (2.32a)$$

sendo M definida como a razão $\frac{b_2}{b_1}$, chamada de constante do molde, a qual mede a capacidade de extração de calor do molde em relação ao metal [12]. Partindo-se da equação (2.32), encontra-se a expressão para a temperatura de interface (T_i):

$$T_i = \frac{MT_f - T_0 \text{erf}(\varphi)}{(M + \text{erf}(\varphi))} \quad (2.33)$$

considerando o valor de T_i dado em (2.33) na equação (2.20), encontra-se uma expressão alternativa para a constante B_2 , que é dada por:

$$B_2 = \frac{(T_f - T_0)}{(M + \text{erf}(\varphi))} \quad (2.34)$$

Para a determinação da equação que descreve a constante de resfriamento

φ , parte-se do seguinte balanço térmico na interface sólido-líquido:

$$k_2 \left. \frac{\partial T_2}{\partial x} \right|_{x=\xi} = k_3 \left. \frac{\partial T_3}{\partial x} \right|_{x=\xi} + \rho_2 L_{f_2} \frac{d\xi}{dt}, \quad (2.35)$$

Onde:

- i) O termo do lado esquerdo da equação (2.35) corresponde ao do calor retirado pelo sólido;
- ii) O primeiro termo do lado direito da equação (2.35) corresponde ao calor do superaquecimento;
- iii) O segundo termo do lado direito da equação (2.35) corresponde ao calor latente de solidificação.

Inicialmente, calcula-se o termo que representa o calor retirado pelo sólido, a partir da equação (2.10), com $l=2$, ou seja:

$$k_2 \left. \frac{\partial T_2}{\partial x} \right|_{x=\xi} = k_2 \frac{B_2 \sqrt{\pi}}{e^{\left(\frac{\xi^2}{4ta_2}\right)} \sqrt{ta_2} \pi} \quad (2.36)$$

Sendo:

$$\xi^2 = 4\varphi^2 ta_2 \quad (2.36a)$$

e

$$b_2 = \frac{k_2}{\sqrt{a_2}} \quad (2.36b)$$

a equação (2.36) pode ser reescrita como:

$$k_2 \frac{\partial T_2}{\partial x_2} \Big|_{x=\xi} = \frac{b_2 B_2}{e^{\phi^2} \sqrt{t\pi}} \quad (2.37)$$

Calcula-se a seguir, partindo da equação (2.10), considerando $l=3$, o calor do superaquecimento, ou seja:

$$K_3 \frac{\partial T_3}{\partial x} \Big|_{x=\xi} = K_3 \frac{B_3 \sqrt{\pi}}{e^{\left(\frac{\xi^2}{4ta_3}\right)} \sqrt{ta_3} \pi} \quad (2.38)$$

que após simplificações algébricas resulta:

$$k_3 \frac{\partial T_3}{\partial x_3} \Big|_{x=\xi} = \frac{b_3 B_3}{e^{n^2 \phi^2} \sqrt{t\pi}} \quad (2.39)$$

onde os coeficientes b_3 e n^2 foram, definidos anteriormente como as razões $\frac{k_3}{\sqrt{a_3}}$ e $\frac{a_2}{a_3}$

Finalmente, o calor latente de solidificação, é obtido da equação (2.19), ou seja:

$$\rho_2 L f_2 \frac{d\zeta}{dt} = \rho_2 L f_2 \frac{\phi a_2}{\sqrt{a_2} t} \quad (2.40)$$

observando-se que a_2 é definido por $\frac{k_2}{\rho_2 C p_2}$ e, multiplicando-se por $\sqrt{\pi}$ o numerador e

o denominador da equação (2.40), tem-se que:

$$\rho_2 L f_2 \frac{d\zeta}{dt} = \frac{L f_2 b_2 \phi \sqrt{\pi}}{C p_2 \sqrt{t\pi}} \quad (2.41)$$

Substituindo-se (2.37), (2.39) e (2.41) na equação (2.35):

$$\frac{b_2 B_2}{e^{\varphi^2} \sqrt{t\pi}} = \frac{b_3 B_3}{e^{n^2 \varphi^2} \sqrt{t\pi}} + \frac{Lf_2 b_2 \varphi \sqrt{\pi}}{Cp_2 \sqrt{t\pi}} \quad (2.42)$$

simplificando-se o termo $\sqrt{t\pi}$ e dividindo-se toda a equação (2.42) por b_2 resulta

$$B_2 e^{-\varphi^2} = \frac{B_3 b_3 e^{-n^2 \varphi^2}}{b_2} + \frac{Lf_2 \varphi \sqrt{\pi}}{Cp_2} \quad (2.43)$$

Substituindo na equação (2.43) o valor da constante B_2 , dada pela equação (2.34); o da constante B_3 , dada pela equação (2.27); o quociente $\frac{b_3}{b_2}$ por m , obtém-se:

$$\frac{m S^* e^{-n^2 \varphi^2}}{(1 - \operatorname{erf}(n\varphi))} - \frac{e^{-\varphi^2}}{(M + \operatorname{erf}(\varphi))} + H^* \varphi \sqrt{\pi} = 0 \quad (2.44)$$

que é a equação transcendental do modelo de Schwarz para a constante de solidificação, onde:

$$S^* = \frac{(T_v - T_f)}{(T_f - T_0)}; \quad (2.45)$$

é definido como o superaquecimento adimensional

$$H^* = \frac{Lf_2}{Cp_2 (T_f - T_0)} \quad (2.45a)$$

é denominado calor de fusão adimensional

3. O Método da Decomposição

3.1. Introdução

Muitos processos, sistemas, e problemas a nível industrial ou mesmo de laboratório geralmente são modelados por equações diferenciais não lineares. Na maioria dos casos porém, há necessidade de simplificações e suposições restritivas, ou então exige-se a utilização de métodos numéricos para a resolução do problema.

Cabe observar que, a utilização de métodos numéricos para a resolução de equações não lineares apresenta diversos contratempos, do tipo, estudo do método mais adequado ao problema, determinação e observância dos critérios de convergência, tempo de processamento para obtenção de um resultado aceitável, erro de computação devido ao arredondamento, entre outros.

O método da Decomposição proposto por Adomian [1], tem como objetivo fornecer uma aproximação analítica a uma ampla classe de equações não lineares (algébricas, diferenciais, integro-diferenciais, equações diferenciais parciais, etc), sem que haja necessidade de qualquer tipo de simplificação no modelo.

Nesta técnica, a solução em série, na qual cada termo é facilmente obtido, porque a não linearidade é bem descrita pelos polinômios propostos por Adomian. Cabe observar que a solução em série é truncada no cálculo numérico da solução e, geralmente, com poucos termos, resultados numéricos precisos são obtidos.

3.2. O método de Decomposição

Partindo de uma equação diferencial do tipo:

$$\mathfrak{I}u(t) = g(t) \quad (3.1)$$

onde \mathfrak{I} representa um operador diferencial ordinário não-linear envolvendo as partes lineares e não-lineares da equação, e g representa uma função qualquer de t .

O termo linear da equação é decomposto em uma soma de operadores $L+R$, onde L é facilmente inversível e R é a parte restante do operador linear. Por conveniência, L pode ser tomado como a derivada de mais alta ordem, evitando o emprego dos métodos de integração, como por exemplo, as funções de Green. Desta maneira a equação (3.1) pode ser reescrita como:

$$Lu + Ru + Nu = g \quad (3.2)$$

onde Nu representa o termo não linear da equação (3.2). Isolando no primeiro membro da equação (3.2), o termo Lu , tem-se:

$$Lu = g - Ru - Nu \quad (3.3)$$

Como o operador L é facilmente inversível, a equação (3.3) pode ser reescrita da seguinte forma:

$$u = L^{-1}Lu = L^{-1}g - L^{-1}Ru - L^{-1}Nu \quad (3.4)$$

Se L é um operador diferencial de primeira ordem então L^{-1} é um operador integral simples:

$$L^{-1}u = \int_{t_0}^t u(t) dt \quad (3.5)$$

Por outro lado, se L é um operador linear de segunda ordem, L^{-1} é um operador de integração dupla, ou seja:

$$L^{-1}Lu = u - u(t_0) - u'(t_0)(t - t_0), \quad (3.6)$$

e assim por diante

Para problemas de contorno (e problemas de valor inicial também), integrações indefinidas são usadas e as constantes são calculadas a partir das condições dadas. Considerando L um operador diferencial de segunda ordem associado à um problema de condição inicial, então obtém-se:

$$u = A + Bt + L^{-1}g - L^{-1}Ru - L^{-1}Nu \quad (3.7)$$

onde:

$$A = -(u(t_0) - t_0 u'(t_0)) \quad (3.7a)$$

e

$$B = u'(t_0) \quad (3.7b)$$

O termo não-linear Nu será expandido na série $\sum_{n=0}^{\infty} A_n$, onde os elementos A_n são polinômios determinados por Adomian. A função incógnita u também será expandida na série $\sum_{n=0}^{\infty} u_n$, com u_0 tomado como o valor $A + Bt + L^{-1}g$. Então, substituindo estas expansões em série em (3.7), resulta:

$$\sum_{n=0}^{\infty} u_n = u_0 - L^{-1}R \sum_{n=0}^{\infty} u_n - L^{-1} \sum_{n=0}^{\infty} A_n \quad (3.8)$$

Os polinômios A_n são gerados para cada não-linearidade, assim sendo [2], A_0 depende somente de u_0 , A_1 depende somente de u_0 e u_1 , A_2 depende de u_0 , u_1 , u_2 , etc. Todos os componentes u_n são calculáveis. Se as séries convergem, a n -ésima soma parcial da soma $\phi_n = \sum_{i=0}^{n-1} u_i$ será uma solução aproximada desde que:

$$\lim_{n \rightarrow \infty} \phi_n = \sum_{i=0}^{\infty} u_i = u, \text{ por definição.}$$

É importante enfatizar que os elementos A_n podem ser calculados para não-linearidades complicadas da forma $F(u, u', \dots)$ ou $f(g(u))$.

Sendo assim, não há nenhuma simplificação do modelo inicialmente proposto. Então as soluções tendem a se aproximar muito da realidade do problema, muito além daquelas obtidas por métodos aproximados e métodos numéricos. Além disso, valores numéricos podem ser calculados se necessário, e é simples a verificação da convergência quando os termos forem calculados [2].

3.3. Formulação dos A_n

Duas formulações para os polinômios A_n são apresentadas a seguir, denominadas respectivamente de A_n e \hat{A}_n . Qualquer uma das duas formulações pode ser utilizada para a obtenção de soluções.

Considere uma equação cuja solução é $u(x)$, contendo um termo não linear

$$Nu \equiv f(u) = \sum_{n=0}^{\infty} A_n = \sum_{n=0}^{\infty} \hat{A}_n. \text{ Os polinômios } A_n \text{ são definidos por [1]:}$$

$$\begin{aligned}
A_0 &= f(u_0) \\
A_1 &= u_1 \left(\frac{d}{du_0} \right) f(u_0) \\
A_2 &= u_2 \left(\frac{d}{du_0} \right) f(u_0) + \left(\frac{u_1^2}{2!} \right) \left(\frac{d^2}{du_0^2} \right) f(u_0) \\
A_3 &= u_3 \left(\frac{d}{du_0} \right) f(u_0) + u_1 u_2 \left(\frac{d^2}{du_0^2} \right) f(u_0) + \left(\frac{u_1^3}{3!} \right) \frac{d^3}{du_0^3} f(u_0) \\
&\vdots \\
&\vdots
\end{aligned} \tag{3.9}$$

e assim sucessivamente.

Os polinômios \hat{A}_n são definidos como:

$$\begin{aligned}
\hat{A}_0 &= f(u_0) \\
\hat{A}_1 &= u_1 \left(\frac{d}{du_0} \right) f(u_0) + \left(\frac{u_1^2}{2!} \right) \left(\frac{d^2}{du_0^2} \right) f(u_0) + \left(\frac{u_1^3}{3!} \right) \left(\frac{d^3}{du_0^3} \right) f(u_0) + \dots \\
\hat{A}_2 &= u_2 \left(\frac{d}{du_0} \right) f(u_0) + \left(\frac{u_2^2}{2!} \right) \left(\frac{d^2}{du_0^2} \right) f(u_0) + \left(\frac{u_2^3}{3!} \right) \left(\frac{d^3}{du_0^3} \right) f(u_0) + \dots \\
&+ u_1 u_2 \left(\frac{d^2}{du_0^2} \right) f(u_0) + \left(\frac{1}{2} \right) (u_1^2 u_2 + u_2^2 u_1) \frac{d^3}{du_0^3} f(u_0) + \dots
\end{aligned} \tag{3.10}$$

e assim sucessivamente

A primeira formulação é conhecida como forma acelerada para os polinômios A_n . É chamada desta maneira, porque o seu cálculo é mais rápido que a da segunda formulação. Com relação a convergência, é mais lenta que a segunda.

Qualquer uma das duas formulações é convergente e pode ser aplicada em qualquer problema que se queira a solução analítica da equação.

Para a aplicação deste método em equações transcendentais deve-se reescrevê-la da seguinte forma:

$$u = c + f(u), \quad (3.11)$$

sendo:

(i) u a solução da equação transcendental $f(u)=0$;

(ii) c um termo constante;

(iii) $f(u)$ termo restante da equação;

O termo não linear $f(u)$ é equacionado como $\sum_{n=0}^{\infty} A_n$, onde os A_n são polinômios especiais propostos por Adomian. Então a solução da equação (3.11) pode ser escrita como:

$$u = c + \sum_{n=0}^{\infty} A_n \quad (3.12)$$

onde as expressões usadas para os polinômios A_n considerados neste trabalho (1) são expressos como:

$$\begin{aligned}A_0 &= f(u_0) \\A_1 &= A_0 \left(\frac{d}{dt} \right) f(u_0) \\A_2 &= A_1 \left(\frac{d}{dt} \right) f(u_0) + \left(\frac{A_1^2}{2!} \right) \left(\frac{d^2}{dt^2} \right) f(u_0) \\A_3 &= A_2 \left(\frac{d}{dt} \right) f(u_0) + A_1 A_2 \left(\frac{d^2}{dt^2} \right) f(u_0) + \left(\frac{A_2^3}{3!} \right) \frac{d^3}{dt^3} f(u_0) \\&\vdots\end{aligned}\tag{3.13}$$

Assim a solução da equação (3.11) pelo método da Decomposição proposto por Adomian [2], é expressa como:

$$u = A_0 + A_1 + A_2 + \dots\tag{3.14}$$

4. Solução do Modelo de Schwarz

Partindo-se da equação transcendental (2.44) para o coeficiente de resfriamento mostrada na seção 2:

$$\frac{mS^*e^{-\varphi^2}}{(1-\text{erf}(n\varphi))} - \frac{e^{-n\varphi^2}}{(M+\text{erf}(\varphi))} + H^*\varphi\sqrt{\pi} = 0 \quad (4.1)$$

supondo-se que as propriedades do metal sólido e líquido sejam iguais, então a equação (4.1) é reescrita como ($m=n=1$):

$$\frac{S^*e^{-\varphi^2}}{(1-\text{erf}(\varphi))} - \frac{e^{-\varphi^2}}{(M+\text{erf}(\varphi))} + H^*\varphi\sqrt{\pi} = 0 \quad (4.2)$$

Para a determinação de φ pelo método da decomposição, a equação deve ser colocada na forma da equação (3.11). Nesse sentido, coloca-se a exponencial na equação (4.1) em evidência resultando:

$$\frac{1}{e^{\varphi^2}} \left[\frac{MS^* + S^*\text{erf}(\varphi) - 1 + \text{erf}(\varphi)}{M - M\text{erf}(\varphi) + \text{erf}(\varphi) - \text{erf}^2(\varphi)} \right] = -H^*\sqrt{\pi}\varphi \quad (4.3)$$

Definindo-se os seguintes parâmetros:

$$P1 = MS^* - 1 \quad (4.4)$$

$$P2 = S^* + 1 \quad (4.4a)$$

$$P3 = -H^*\sqrt{\pi} \quad (4.4b)$$

$$P4 = 1 - M \quad (4.4c)$$

Então a equação (4.3) pode ser reescrita como $\varphi = c + f(\varphi)$, onde $c=0$, e

$$f(\varphi) = \frac{1}{P3e^{\varphi^2}} \left[\frac{P1 + P2\text{erf}(\varphi)}{\text{erf}(\varphi)(P4 - \text{erf}(\varphi)) + M} \right] \quad (4.5)$$

Para o cálculo dos termos A_n definidos em (3.13), é necessário que sejam computados o valor da $f(\varphi)$ e de suas derivadas. Cabe observar, que o cálculo das derivadas da $f(\varphi)$, em φ_0 , foi efetuada utilizando o software Reduce [15].

o cálculo da derivada é dado por:

$$\frac{d(f(\varphi))}{d\varphi} = \frac{2(D_1 + D_2 + D_3)}{D_4}, \quad (4.6)$$

$$D_1 = \left(((\text{erf}(\varphi) - P4)\text{erf}(\varphi) - M)\pi\varphi \right) (\text{erf}(\varphi)P2 + P1) \quad (4.6a)$$

$$D_2 = \sqrt{\pi} \left((\text{erf}(\varphi) - P4) + \text{erf}(\varphi) \right) (\text{erf}(\varphi)P2 + P1) \quad (4.6b)$$

$$D_3 = -\sqrt{\pi} \left((\text{erf}(\varphi) - P4)\text{erf}(\varphi) - M \right) P2 \quad (4.6c)$$

$$D_4 = e^{2\varphi^2} \left((\text{erf}(\varphi) - P4)\text{erf}(\varphi) - M \right)^2 P3\pi \quad (4.6d)$$

definindo-se:

$$P5 = \left((\text{erf}(\varphi) - P4) + \text{erf}(\varphi) \right) \text{erf}(\varphi) \quad (4.7)$$

Tem-se então a segunda derivada:

$$\frac{d^2(f(\varphi))}{d\varphi^2} = \frac{2 \left((E_1 + E_2 + E_3 + E_4) - 4(E_5 + E_6 + E_7 + E_8) \right)}{E_9}, \quad (4.8)$$

sendo

$$E_1 = \left(e^{\varphi^2} \left(e^{\varphi^2} (P5 - M) (2\varphi^2 + 1) \pi + 2\sqrt{\pi} P5 \varphi \right) + 4 \right) (P5 - M) \quad (4.8a)$$

$$E_2 = \left((P1 + P2 \operatorname{erf}(\varphi)) \pi \right) (P5 - M) \quad (4.8b)$$

$$E_3 = \left(2\sqrt{\pi} \left(e^{\varphi^2} \left((\operatorname{erf}(\varphi) - P4) \operatorname{erf}(\varphi) - M \right) \pi \varphi + \sqrt{\pi} P5 \right) P2 \right) (P5 - M) \quad (4.8c)$$

$$E_4 = (-2P2P5\pi)(P5 - M) \quad (4.8d)$$

$$E_5 = e^{\varphi^2} (P5 - M) \pi \varphi + \sqrt{\pi} P5 \quad (4.8e)$$

$$E_6 = P1 + P2 \operatorname{erf}(\varphi) \quad (4.8f)$$

$$E_7 = -\sqrt{\pi} (P5 - M) P2 \quad (4.8g)$$

$$E_8 = e^{\varphi^2} (P5 - M) \pi \varphi + \sqrt{\pi} P5 \quad (4.8h)$$

$$E_9 = e^{3\varphi^2} (P5 - M)^3 P3 \pi^2 \quad (4.8i)$$

Para $\varphi_0 = 0$, $f(\varphi_0)$ e suas derivadas assumem respectivamente os valores:

$$f(\varphi_0) = \frac{1 - MS^*}{MH^* \sqrt{\pi}} \quad (4.9)$$

$$\frac{d}{d\varphi} (f(\varphi_0)) = \frac{-2(M^2 S^* + 1)}{\pi H^* M^2} \quad (4.10)$$

$$\frac{d^2}{d\varphi^2} (f(\varphi_0)) = \frac{-2M^2 \pi + 2\pi M^3 S^* - 8M^3 S^* + 8}{\pi \sqrt{\pi} H^* M^3} \quad (4.11)$$

Neste trabalho, considera-se apenas os tres primeiros termos da série dada em (3.12), e para $\varphi_0 = 0$, ou sejam:

$$A_0 = \frac{(1 - MS^*)}{\sqrt{\pi}MH^*} \quad (4.12)$$

$$A_1 = \frac{-2(1 - MS^*)(M^2S^* + 1)}{\sqrt{\pi}MH^* \pi H^* M^2} \quad (4.13)$$

$$A_2 = \left(\frac{MS^* + 1}{MH^* \sqrt{\pi}} \right) \left(\frac{2(M^2S^* + 1)}{\pi H^* M^2} \right) \left(\frac{-2(M^2S^* + 1)}{\pi H^* M^2} \right) + \frac{\left(\left(\frac{MS^* + 1}{MH^* \sqrt{\pi}} \right) \left(\frac{2(M^2S^* + 1)}{\pi H^* M^2} \right) \right)^2}{2} \left(\frac{-2M^2\pi^2 + 2\pi^2M^3S^* - 8\pi M^3S^* + 8\pi}{\pi^2 \sqrt{\pi}H^*M^3} \right) \quad (4.14)$$

Resulta então a seguinte expressão para a constante de solidificação, do modelo de Schwarz pelo método da Decomposição, proposto por Adomian [2]:

$$\varphi = \sum_{i=0}^2 A_n = A_0 + A_1 + A_2 \quad (4.15)$$

Cumpra finalmente observar que para resolver a equação transcendental (2.44) é necessário somente três termos da solução em série definida em (3.12) (ver capítulo 5) para obter-se resultados precisos.

5. Resultados

A seguir são apresentadas comparações numéricas dos resultados obtidos para a constante de resfriamento obtida pelo método da decomposição com resultados do método de Newton, para o cálculo de raízes de equações transcendentais.

A linguagem utilizada para a construção dos softwares foi o Fortran. Esta linguagem foi escolhida devido a facilidade de implementação, por ser uma linguagem largamente utilizada no meio científico, e também porque o software Reduce (15), utilizado para o cálculo das derivadas possui uma saída compatível com a linguagem.

A figura 2 apresenta um gráfico com diversos valores de H^* , calor latente de solidificação adimensional definido como: $H^* = \frac{L_f}{C_p(T_f - T_0)}$. Os metais apresentados possuem as mais diversas temperaturas de fusão e valores de calor latente. Estes valores foram escolhidos para definir o intervalo onde será aplicado o método da decomposição.

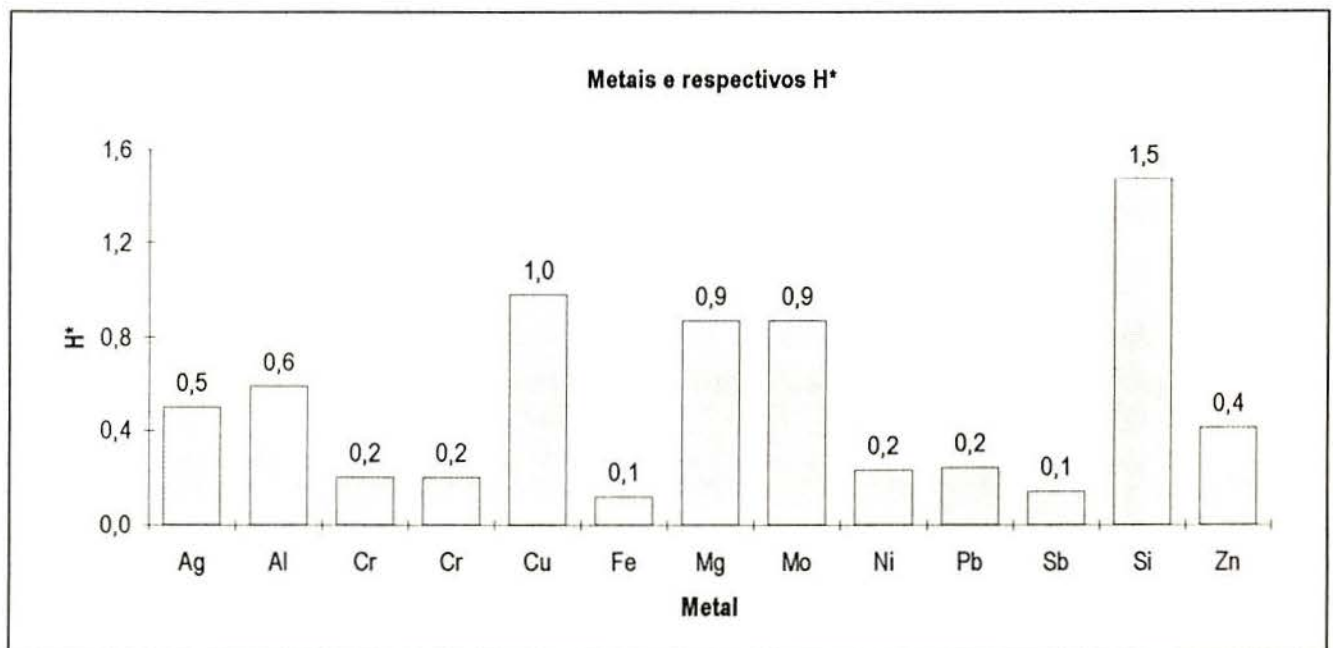


Figura 2: Metais e respectivos H^*

Para verificar a abrangência e a validade do modelo proposto, foram construídos gráficos comparativos entre a solução analítica e a solução numérica, para os seguintes valores de M: 1, 5, 10 e 20, que é a difusividade de calor entre o molde e o metal. Estes valores de M expressam os diversos moldes existentes. Para valores de $M > 20$, o modelo de Schwarz não pode ser aplicado, porque a constante de solidificação passa a ter um valor negativo, o que significa segundo Schwarz [16], que o metal não solidificará, porque o molde não extrai calor do metal.

A abcissa dos gráficos é o H^* , que é o calor latente de solidificação adimensional, que foi calculado no intervalo [0,4; 2], com passo 0,1. A ordenada dos gráficos é o φ (Constante de solidificação). O S^* que é o calor do superaquecimento adimensional, foi fixado em 3 valores: { 0; 0,05; 0,08 }.

A fórmula para se calcular o erro relativo cometido pelo método proposto é a seguinte [8]:

$$E_0 = \frac{x - x_n}{x_n} \quad (5.1)$$

onde:

Δx = Valor calculado por método numérico

x = Valor calculado pelo método da decomposição

A figura 3 apresenta um gráfico comparativo entre a solução proposta para a equação (2.45) e a solução calculada pelo método de Newton, para $S^*=0$, $M=1$ e H^* no intervalo $[0,4;2]$. Observa-se na figura 3 que o método proposto possui um erro elevado para os valores iniciais de H^* . Para $H^* = 0,4$, o erro relativo cometido é da ordem de 477 %. Observa-se também que o erro cometido pelo método proposto possui um decaimento rápido, sendo que para $H^* = 2$ o erro é da ordem de 5%.

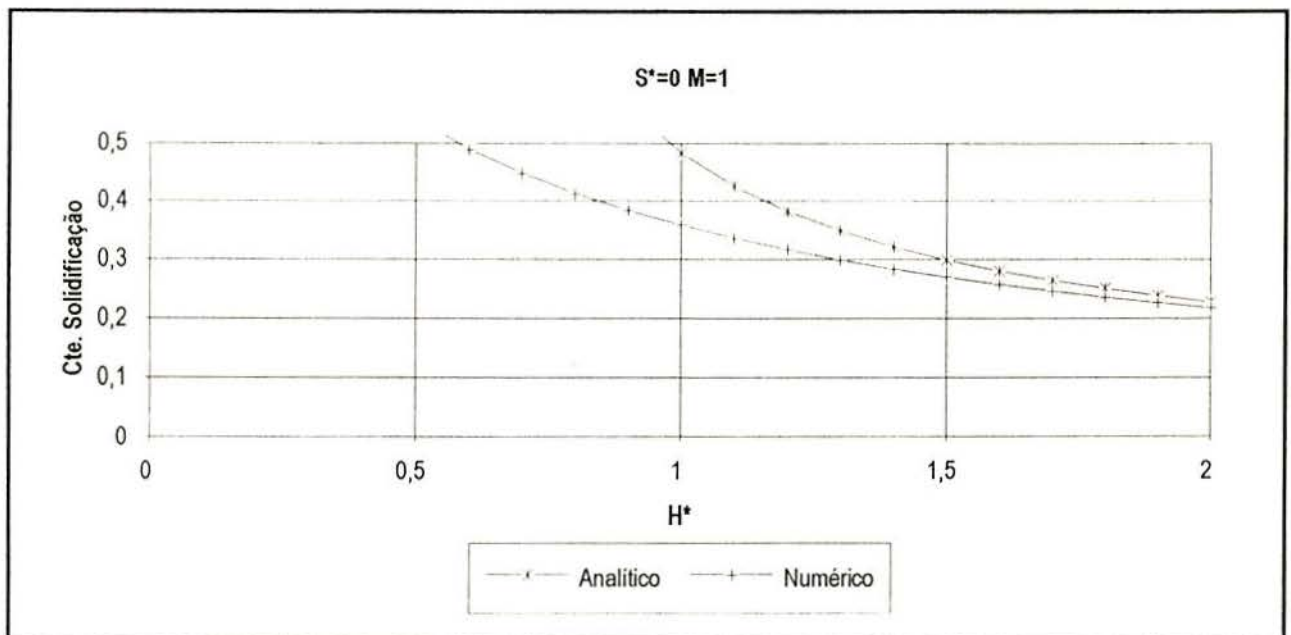


Figura 3: Gráfico comparativo entre o método de Newton e o método Proposto, com $S^*=0$ e $M=1$.

A figura 4 apresenta um gráfico comparativo entre a solução proposta para a equação (2.45) e a solução calculada pelo método de Newton, para $S^*=0$, $M=5$ e H^* no intervalo $[0,4;2]$. Observa-se na figura 4, que o erro máximo cometido pelo método Proposto é da ordem de 0,5 %, e ocorre para $H^* = 0,4$. Com o crescimento de H^* , o erro cai drasticamente, sendo que para $H^* = 1$ o erro é de 0,03 %, o que pode ser considerado desprezível.

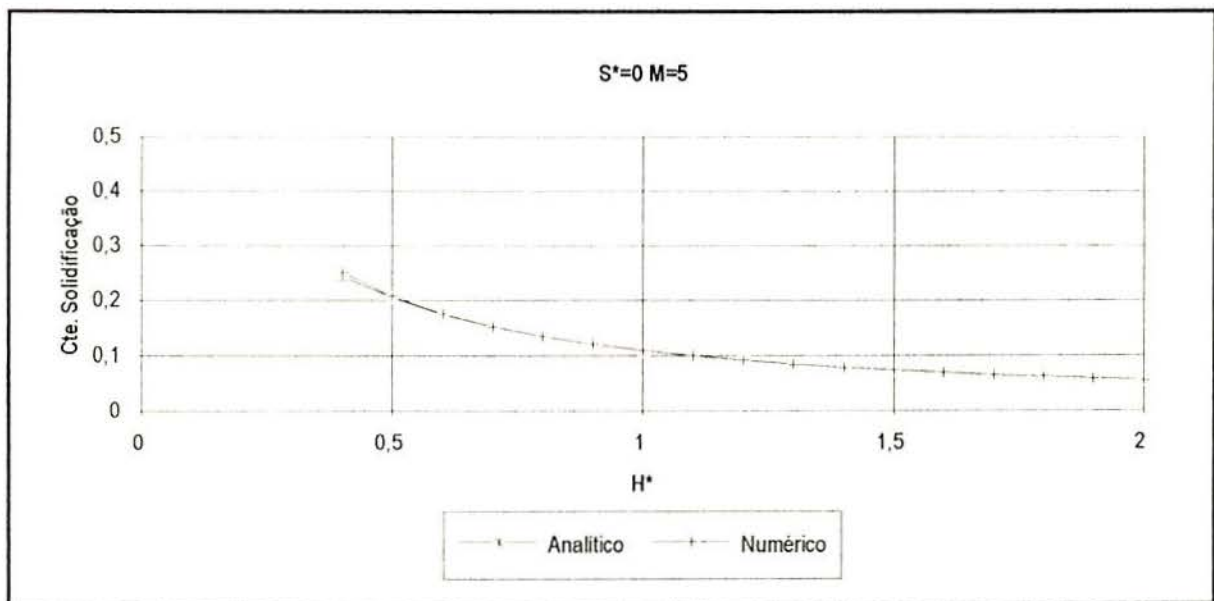


Figura 4: Gráfico comparativo entre o método de Newton e o método Proposto, com $S^*=0$ e $M=5$.

A figura 5 apresenta um gráfico comparativo entre a solução proposta para a equação (2.45) e a solução calculada pelo método de Newton, para $S^*=0$, $M=10$ e H^* no intervalo $[0,4;2]$. Observa-se na figura 5, que o erro máximo cometido pelo método Proposto se situa em torno de 0,22 %, e ocorre para $H^* = 0,4$.

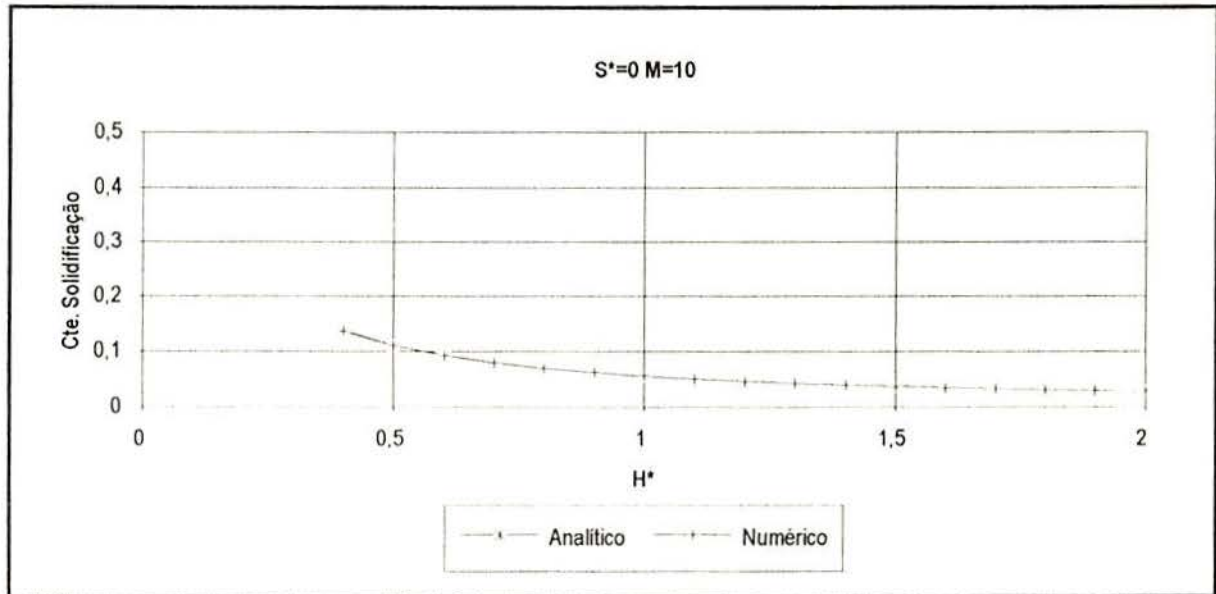


Figura 5: Gráfico comparativo entre o método de Newton e o método Proposto, com $S^*=0$ e $M=10$.

A figura 6 apresenta um gráfico comparativo entre a solução proposta para a equação (2.45) e a solução calculada pelo método de Newton, para $S^*=0$, $M=20$ e H^* no intervalo $[0,4;2]$. Observa-se na figura 6, que o erro máximo cometido pelo método se situa novamente no ponto $H^* = 0,4$, e é da ordem de 0,02 %.

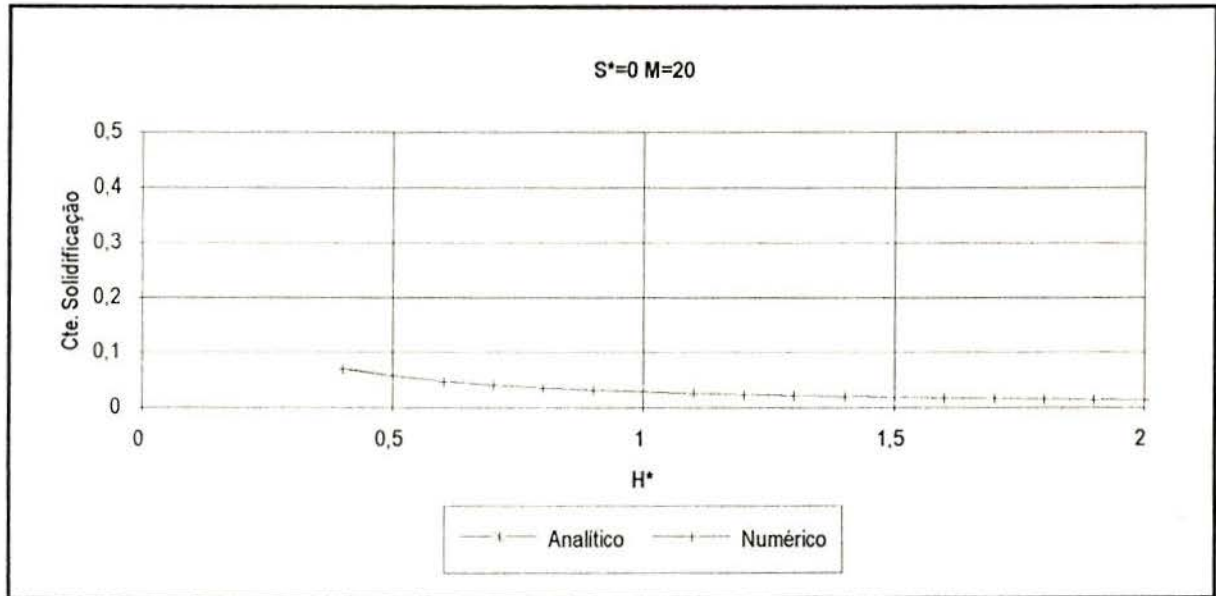


Figura 6: Gráfico comparativo entre o método de Newton e o método Proposto, com $S^*=0$ e $M=20$.

A figura 7 apresenta um gráfico comparativo entre a solução proposta para a equação (2.45) e a solução calculada pelo método de Newton, para $S^*=0$, $M=10$ e H^* no intervalo $[0,4;2]$. Na figura 7, observa-se que o erro cometido pelo método Proposto é da ordem de 500 %, para $H^*=0,4$. O comportamento da curva proposta é idêntico ao da figura 3, sendo que o erro possui um decaimento rápido, e para $H^*=1,5$ o erro se situa em 12 %.

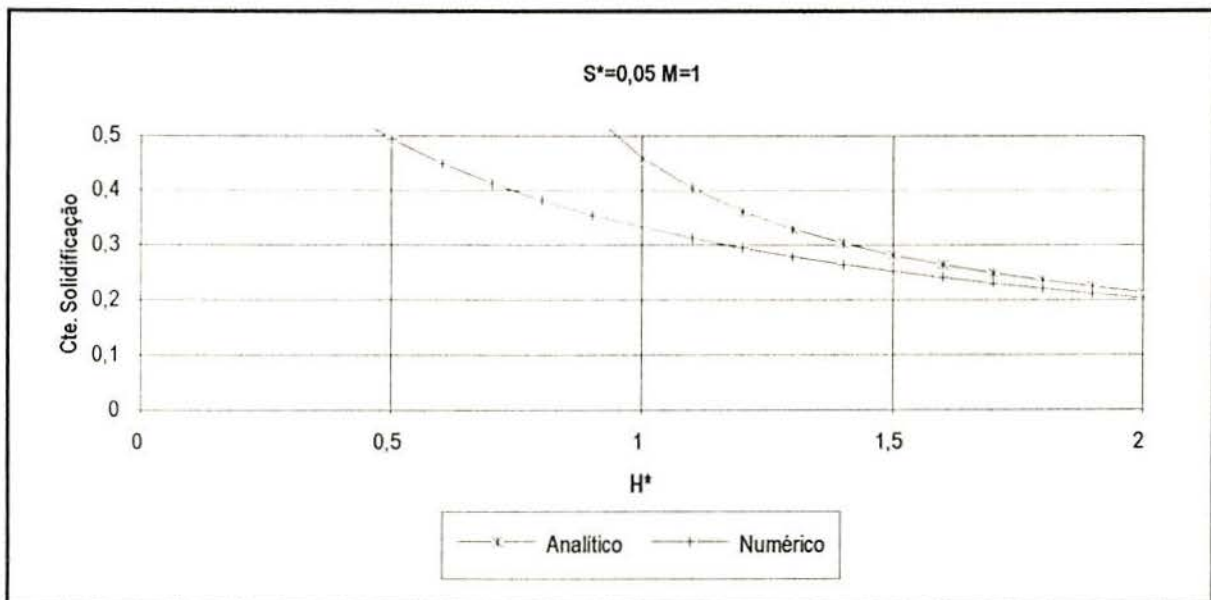


Figura 7: Gráfico comparativo entre o método de Newton e o método Proposto, com $S^*=0,05$ e $M=1$.

A figura 8 apresenta um gráfico comparativo entre a solução proposta para a equação (2.45) e a solução calculada pelo método de Newton, para $S^*=0,05$, $M=10$ e H^* no intervalo $[0,4;2]$. Na figura 8, observa-se que o erro máximo cometido pelo modelo proposto é da ordem de 3%, e ocorre para $H^*=0,4$.

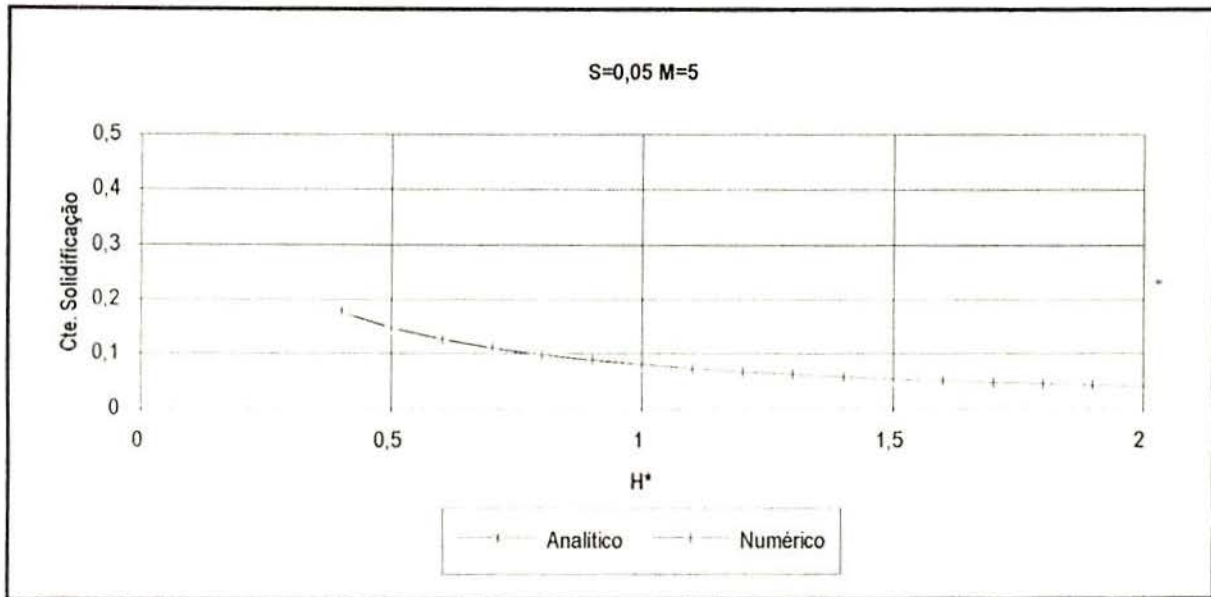


Figura 8: Gráfico comparativo entre o método de Newton e o método Proposto, com $S^*=0,05$ e $M=5$.

A figura 9 apresenta um gráfico comparativo entre a solução proposta para a equação (2.45) e a solução calculada pelo método de Newton, para $S^*=0,05$, $M=10$ e H^* no intervalo $[0,4;2]$. Na figura 9, observa-se que o maior erro cometido pelo método proposto é da ordem de 0,25 %, e ocorre para $H^*=0,4$. De um modo geral, para $M>2$, o modelo proposto possui erros relativos extremamente baixos.

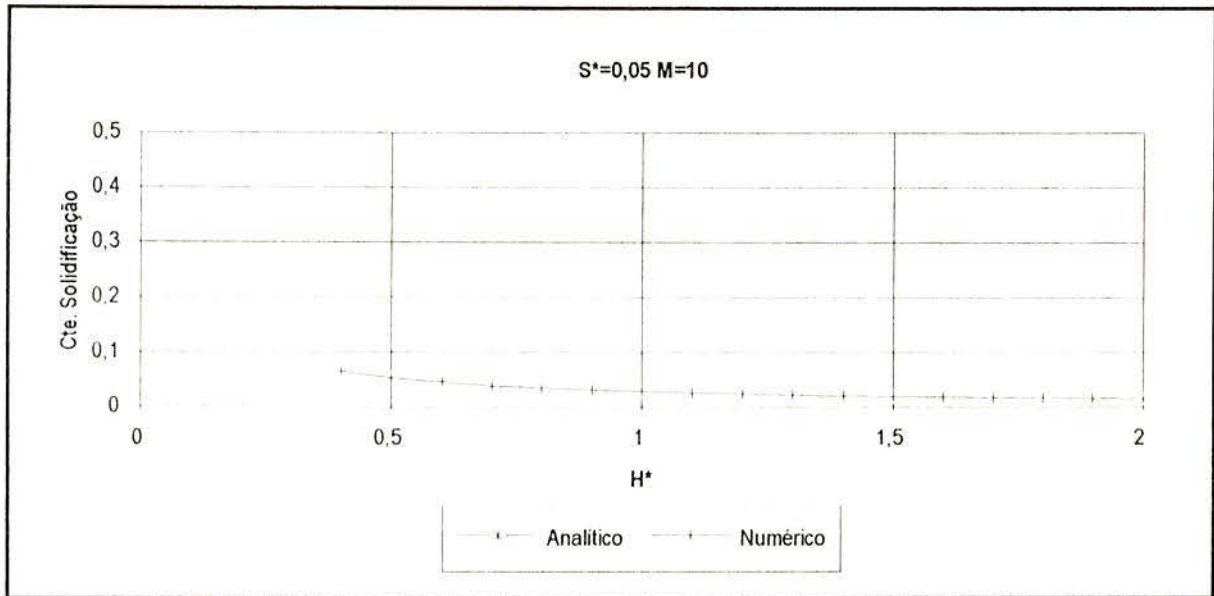


Figura 9: Gráfico comparativo entre o método de Newton e o método Proposto, com $S^*=0,05$ e $M=10$.

A figura 10 apresenta um gráfico comparativo entre a solução proposta para a equação (2.45) e a solução calculada pelo método de Newton, para $S^*=0,08$, $M=1$ e H^* no intervalo $[0,4;2]$. Na figura 10, observa-se que o erro cometido é da ordem de 480 %, para $H^*=0,4$. Na medida em que o H^* cresce, o erro diminui, sendo que para $H^*=2$, o erro é da ordem de 5 %.

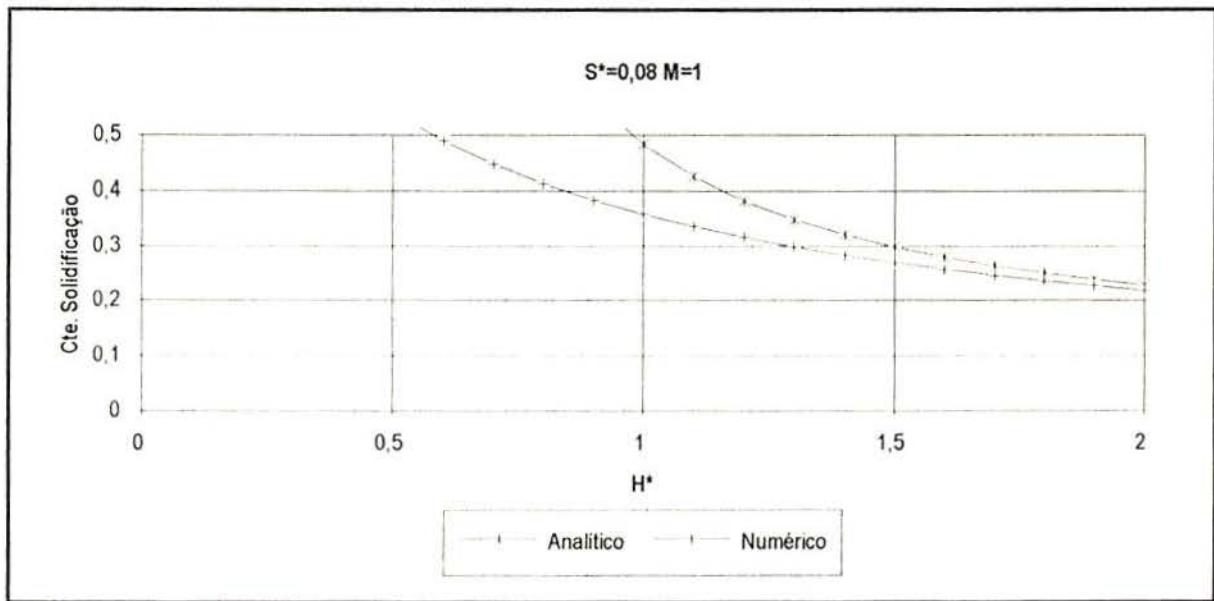


Figura 10: Gráfico comparativo entre o método de Newton e o método Proposto, com $S^*=0,08$ e $M=1$.

A figura 11 apresenta um gráfico comparativo entre a solução proposta para a equação (2.45) e a solução calculada pelo método de Newton, para $S^*=0,08$, $M=5$ e H^* no intervalo $[0,4;2]$. Na figura 11, observa-se que o erro máximo cometido pelo método proposto é da ordem de 3 %, e ocorre para $H^*=0,4$.

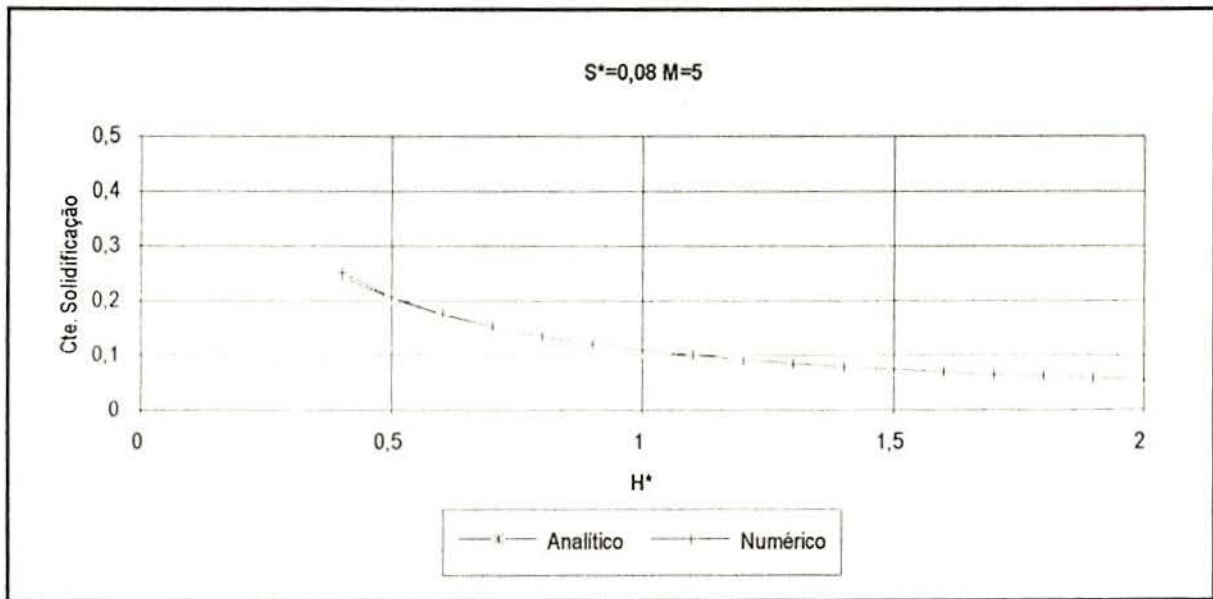


Figura 11: Gráfico comparativo entre o método de Newton e o método Proposto, com $S^*=0,08$ e $M=5$.

A figura 12 apresenta um gráfico comparativo entre a solução proposta para a equação (2.45) e a solução calculada pelo método de Newton, para $S^*=0,08$, $M=1$ e H^* no intervalo $[0,4;2]$. Na figura 12, observa-se que o erro máximo cometido pelo método Proposto é da ordem de 0,22 % e ocorreu para $H^*=0,4$.

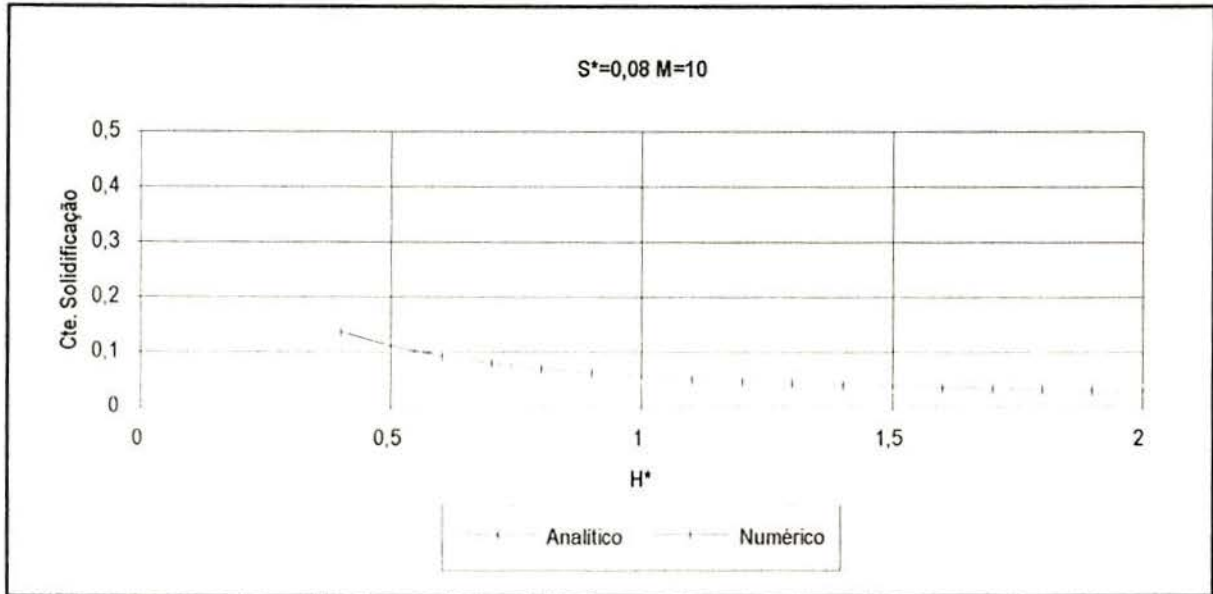


Figura 12: Gráfico comparativo entre o método de Newton e o método Proposto, com $S^*=0,08$ e $M=10$.

Para se verificar a velocidade de processamento da solução proposta, comparada com o método de Newton, foi construído um software utilizando-se a linguagem QuickBasic 7.0. A escolha da mesma se deve, devido ao fato de possuir funções embutidas, onde é possível se obter o tempo de processamento. Os valores testados foram os seguintes: $S^*=\{0; 0,05; 0,08\}$, $H^*=[0,4;2]$, $M=\{1, 5, 10, 20\}$. Os exemplos foram executados em um IBM-PC 286, com coprocessador matemático.

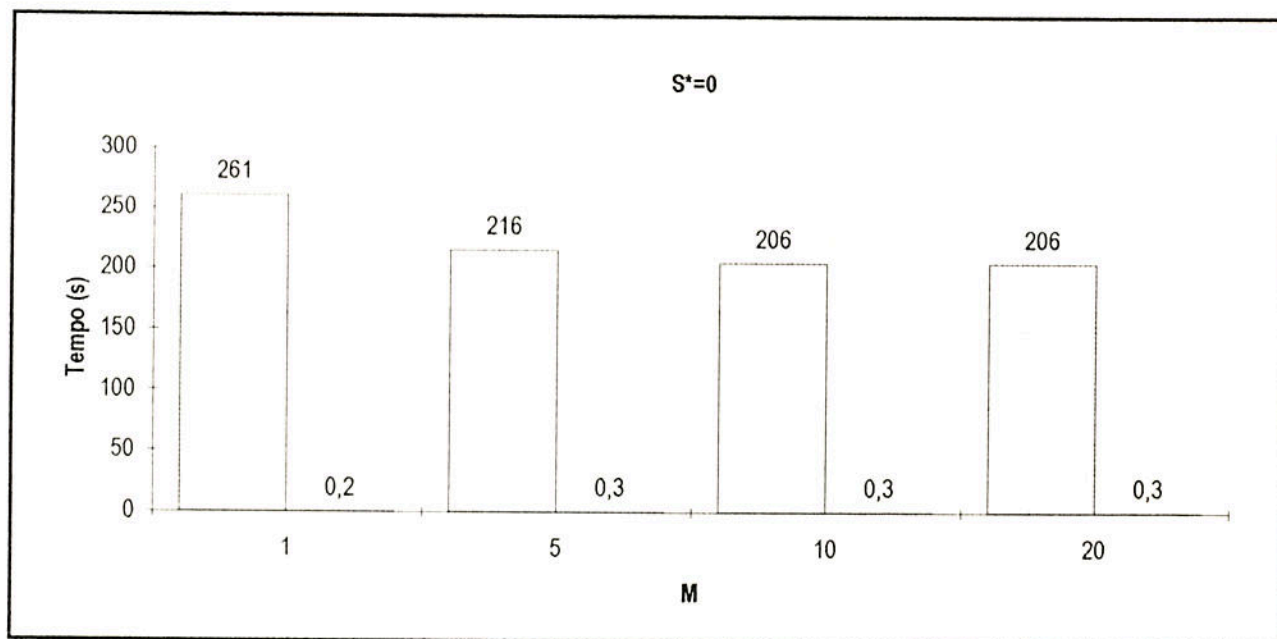


Figura 13: Gráfico comparativo entre os tempos de processamento entre o método proposto e o método de Newton para $S^*=0$.

Observa-se na figura 13 uma diferença elevada entre a solução proposta e a solução numérica utilizando-se o método de Newton. Esta diferença é devida ao fato de o método de Newton necessitar de diversas iterações (5 em média), para encontrar a solução. O método proposto, por outro lado, não necessita de interações, e o cálculo da constante de solidificação é feito de maneira direta.

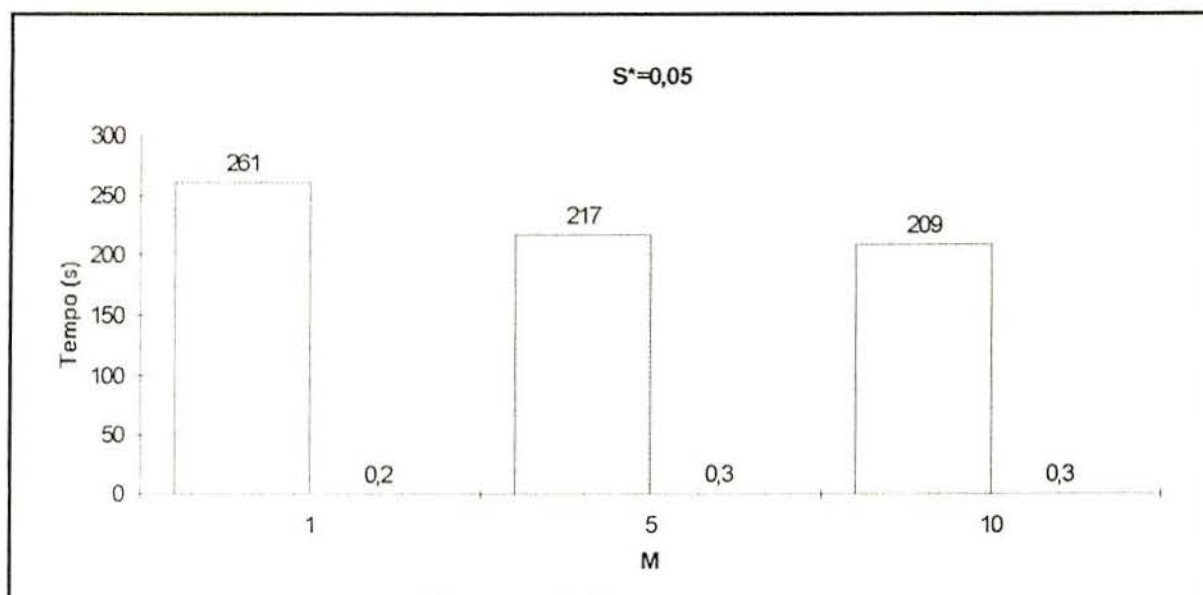


Figura 14: Gráfico comparativo entre os tempos de processamento entre o método proposto e o método de Newton para $S^*=0,05$.

Observa-se na figura 14, um comportamento semelhante ao da figura 13, onde o método proposto é muito mais rápido que o método de Newton.

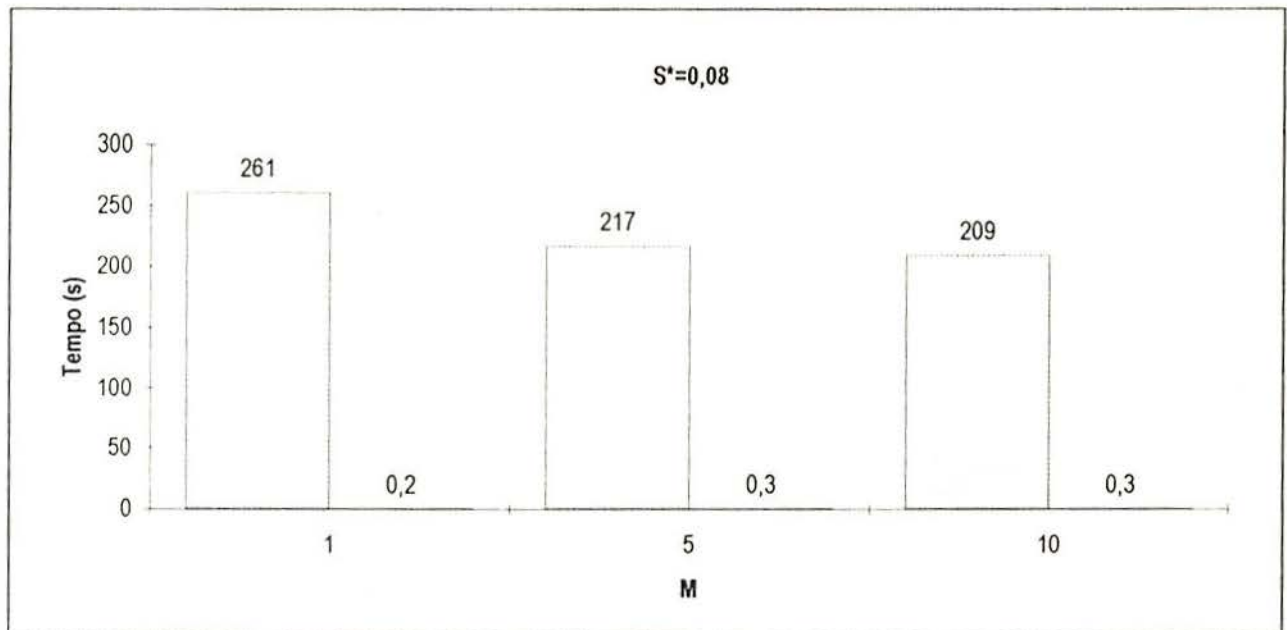


Figura 15: Gráfico comparativo entre os tempos de processamento entre o método proposto e o método de Newton para $S^*=0,08$.

Observa-se na figura 15, um comportamento semelhante ao das figuras 13 e 14, onde o método proposto é muito mais rápido que o método de Newton.

CAPÍTULO 6

Conclusões

Na literatura são encontrados diversos exemplos de utilização do método na localização de soluções analíticas de equações diferenciais totais, parciais, estocásticas, etc. A aplicação deste método para a resolução de equações transcendentais é nova, sendo que na literatura consultada não foram encontradas referências sobre o assunto.

Analisando-se os gráficos apresentados no capítulo 5, verifica-se que o modelo Proposto possui um erro máximo da ordem de 3 % para valores de M maiores que 5, independente da faixa de calor latente de solidificação adimensional (H^*), e calor do superaquecimento adimensional (S^*).

Para valores de $M < 5$, existe uma restrição na aplicação do método Proposto. Esta restrição se impõe na faixa de valores de H^* que podem ser calculados. Para $M < 5$, o H^* mínimo para cálculo, e que apresenta um valor com um erro aproximado de 15 % é de 1,5. Para valores superiores a 1,5, o erro cometido pelo método proposto decresce rapidamente e para $H^* = 2$, o erro se situa em torno de 5 %.

É importante salientar, que a solução proposta e o método de Newton tiveram um comportamento semelhante para todos os valores de S^* calculados. Partindo-se desta afirmação pode-se afirmar que para valores de S^* superiores a este valor, o modelo possui o mesmo comportamento.

O modelo proposto não pode ser aplicado para $M=0$. Isto ocorre pelo fato de que o M aparece no denominador de todas as expressões que compõe a solução proposta.

Além da “convergência numérica” da solução pelo método Proposto, foi também feito uma análise comparativa de tempo de computação entre o método Proposto e o método de Newton. Para o caso de $S^*=0,08$ e $M=20$, o método proposto é em torno de 600 vezes mais rápido que o método de Newton. Para todos os valores calculados, o comportamento observado entre as velocidades do método Proposto e da solução analítica foi semelhante. Isto implica que o método proposto, independente dos dados de entrada, possui uma velocidade de processamento muito superior aos métodos numéricos.

Finalmente ressalta-se que a solução do modelo de Schwarz tornou-se completamente analítica, uma vez que a constante de resfriamento também é obtida em forma analítica.

Sugestões para próximos trabalhos:

1. Aumentar o número de termos da série
2. Encontrar uma outra formulação para se ter uma solução analítica para $M=0$.
3. Utilizar a formulação \hat{A} , para $M=0$

Bibliografia

- [1] Adomian G., "Nonlinear Stochastic Operator Equations", Academic Press, Orlando, FL., 1986.
- [2] Adomian G., "A Review of the Decomposition Method in Applied Mathematics", Journal of Mathematical Analysis and Applications 135, 501-544 (1988).
- [3] Adomian G., Elrod M., and Rach R. "A new approach to boundary value equations and application to a generalization of Airy's Equation", J. Math. Anal. Appl. 140 (1989), 554-568
- [4] Adomian G., Rach R., "On Nonzero Initial Conditions in Stochastic Differential Equations", Journal of Mathematical Analysis and Applications 102, 363-364 (1984).
- [5] Adomian G., Rach R., Elrod M., "On the Solution of Partial Differential Equations with Specified Boundary Conditions. Journal of Mathematical Analysis and Applications 140, 569-581 (1989)
- [6] Carslaw and Jaeger, "Conduction of Heat in Solids", Oxford at the Clarendon Press
- [7] Cherruault Yves, "Convergence of Adomian's Method"
- [8] Claudio, Dalcidio Moraes, Santos José Abel Royo dos "Microcomputadores e Minicalculadoras, Seu Uso em Ciências e Engenharia"
- [9] Cuadernos del INSTITUTO DE MATEMATICA "BEPPO LEVI". Seminario sobre el problema de Stefan y sus aplicaciones. Rosário, 1983.
- [10] Hornbeck, R. W., "Numerical Methods", Quantum Publishers, New York, (1975)

- [11] Kurz, W. Fisher, D.J. , 1986. "Fundamentals of Solidification" Trans Tech Publication, Switzerland.
- [12] Müller Arno, ."Curso de Metalurgia Física - Solidificação", apostila, DEMET/UFRGS.
- [13] Müller Arno, "Solidificacion de Metales Bajo Presion", Tese de doutorado - Buenos Aires - 1974
- [14] Ozisik, M.N., 1980. "Heat Conduction", John Wiley & Sons, New York
- [15] Portugal Renato, Reduce para PC-compatível, Universidade Federal Fluminense, Niterói - RJ.
- [16] Schwarz Carl, Z. Angen. Math und Mech", 1933, vol. 13, p.202 e "Arch. Eisenlутtenwesen", 1931, vol. 5, p.139.
- [17] Tarzia Domingo, Transferencia de calor y materia con cambio de +fase. IX ESCOLA DE MATEMÁTICA APLICADA, Rio de Janeiro, 1993
- [18] Vielmo Horácio Antonio, Simulação numérica da transferência de calor e massa na solidificação de ligas binárias. Tese de Doutorado. Florianópolis, 1993.

Apêndice I - Softwares para conexão com o VAX

Neste apêndice, serão descritos, de forma sucinta o software FTP (File Transport ...) e o TELNET, o primeiro utilizado para a transferência de arquivos do sistema operacional DOS para o UNIX e o segundo para a conexão com o VAX do Instituto de Física da UFRGS.

O texto aqui apresentado foi extraído das notas do CESUP - Centro de Super computação da UFRGS, elaboradas pelo Bacharel em Informática Roland Teodorowitch.

O endereço Internet do Vax da física, utilizado na dissertação é: 143.54.2.1

I.1 - FTP: File Transfer Program

Através de Ftp (File Transfer Program) é possível transferir arquivos em modo texto ou binário entre a máquina local e qualquer outra máquina remota conectada à rede. A máquina local é chamada de cliente e a máquina remota é chamada de servidora. No caso deste trabalho, tem-se que a cliente é um IBM-PC compatível e a máquina servidora é o VAX do Instituto de Física da UFRGS.

Este programa é útil quando se tem arquivos em uma estação de trabalho ou microcomputador e que devem ser compilados em uma máquina que não possui console, ou que se queira utilizar um computador que não se possui acesso direto. O procedimento é simples. Transferem-se os arquivos que se quer, e após conecta-se com a máquina.

A sintaxe básica de FTP é:

```
ftp [máquina]
```

Se o argumento máquina é fornecido, ftp estabelece uma conexão com o computador especificado. O argumento máquina pode ser o rótulo, o nome completo ou o endereço internet da máquina. Observe os exemplos abaixo:

```
$ ftp gauss (Rótulo da máquina remota)
```

\$ ftp if1.ufrgs.br (Nome completo da máquina remota)

\$ ftp 143.54.2.1

Se o argumento máquina é omitido, ftp entra no modo de comando indicado pelo sinal de pronto (ftp>). Neste caso é necessário utilizar o comando open máquina para estabelecer a conexão.

Uma vez que a conexão tenha sido estabelecida, o usuário deverá fornecer o login e a password da máquina remota. A partir daí ftp fornece o sinal de pronto (ftp>) na máquina remota e espera os comandos do usuário.

Os comandos de ftp podem ser divididos em 3 tipos básicos: Transferência, remotos e locais. Para obter uma lista de comandos use, a partir do sinal de pronto (ftp>), o comando 'help' ou '?', e '? comando'.

1.1.1. Comandos de transferência

Estes comandos são responsáveis tanto pela abertura e encerramento da conexão, quanto pela transferência e controle das opções de transferência.

append *arquivo-local* [*arquivo remoto*]

Acrescenta *arquivo-local* ao final de *arquivo-remoto*. Caso *arquivo-remoto* não seja especificado, é feita uma cópia de *arquivo-local* na máquina remota.

ascii

Define o modo de transferência dos dados da rede para ASCII. Este é o modo padrão.

bell

Toca um sinal sonoro após cada transferência de arquivos ter sido completada.

binary

Define o modo de transferência dos dados da rede para binário. Este modo de transferência deve ser usado sempre que se queira transferir arquivos binários, tais como imagens ou arquivos comprimidos.

bye ou quit

Termina a seção de ftp com a máquina servidora e abandona ftp.

close ou disconnect

Termina a seção de ftp com a máquina servidora, retornando ao modo de comando.

get ou recv *arquivo-remoto [arquivo-local]*

Transfere o *arquivo-remoto* especificado para a máquina local. Se o *arquivo-local* não é especificado, ele é gravado com o mesmo nome do *arquivo-remoto*, no diretório corrente da máquina local.

mget *arquivos-remotos*

Transfere os arquivos-remotos, especificados através de caracteres de máscara, para a máquina local. Para obter transferência automática, sem confirmação, ftp deverá estar com modo interativo desligado. Isto é obtido através do comando prompt.

mput *arquivos-locais*

Transfere os arquivos-locais, especificados através de caracteres de máscara, da máquina local para a máquina remota. Para obter transferência automática, sem confirmação, ftp deverá estar no modo interativo. Isto é obtido através do comando prompt.

open *máquina*

Estabelece uma conexão com a máquina especificada, permitindo ao usuário recuperar ou salvar arquivos. O formato do argumento máquina é o mesmo que pode ser especificado na linha de comando de ftp.

prompt

Altera o modo de funcionamento interativo de ftp. Com modo interativo desligado, qualquer comando como **mget**, **mput**, ou **mdelete** será executado sobre todos os arquivos.

put ou send *arquivo-local [arquivo-remoto]*

Transfere o *arquivo-local* especificado para a máquina remota. Se o *arquivo-remoto* não é especificado, ele é gravado com o mesmo nome do *arquivo-local* no diretório corrente da máquina-remota.

runique

Se ativo, evita sobreescrita de arquivos na máquina local. O arquivo é copiado e é acrescentada uma extensão '.1' ao arquivo local. Se este já existir, a extensão utilizada é '.2'. e assim sucessivamente.

status

Mostra o estado atual de ftp

sunique

Se ativo, evita sobreescrita de arquivos na máquina remota. O arquivo é copiado e é acrescentada uma extensão '.1' ao arquivo remoto. Se este já existir, a extensão utilizada é '.2'. e assim sucessivamente.

type [*nome-tipo*]

Define o modo de transferência de arquivos. Os modos válidos são: *ascii* para arquivos texto, *binary* ou *image* para arquivos binários, e *texnet* para comunicação com máquinas TEXNET.

user

Identifica um usuário de ftp para a máquina remota. É útil no caso de uma conexão ter sido aberta com a senha errada.

I.1.2. Comandos remotos

Nesta classe de comandos estão reunidos todos os comandos que trabalham diretamente sobre a máquina remota, permitindo gerenciar diretórios e arquivos na própria máquina remota.

cd [*diretório-remoto*]

Troca o diretório de trabalho na máquina remota para *diretório-remoto*

cdup

troca o diretório de trabalho na máquina remota para o diretório um nível superior.

delete *arquivo-remoto*

Remove o arquivo de nome *arquivo-remoto* na máquina-remota

dir [*diretório-remoto*] [*arquivo-local*]

Imprime uma listagem completa de arquivos do diretório especificado, *diretório-remoto*, ou do diretório corrente, caso nenhum diretório tenha sido especificado.

O conteúdo desta lista pode, opcionalmente ser enviado para um arquivo local. Caso nenhum arquivo seja especificado, o resultado é exibido no vídeo.

ls [*diretório-remoto*] [*arquivo-local*]

Imprime uma lista com o conteúdo do diretório especificado na máquina remota. Caso não seja especificado o *diretório-remoto*, é mostrado o conteúdo do diretório corrente da máquina-remota. O *arquivo-local* é o nome do arquivo para o qual a lista de arquivos é mandada, caso não seja especificado, ou seja especificado o sinal -, a lista de arquivos é enviada para o vídeo.

mdelete [*arquivos-remotos*]

Lista o conteúdo de vários diretórios na máquina remota. A saída pode, como no comando *dir*, ser enviada para um arquivo local.

mkdir *nome-diretório*

Cria um diretório na máquina remota

mls *arquivos-remotos* *arquivo-local*

Imprime uma listagem de múltiplos diretórios, tendo como destino um arquivo local.

pwd

Imprime o nome do diretório corrente na máquina remota.

rename [*inicial*] [*final*]

Renomeia o arquivo de nome inicial para o nome final na máquina remota.

rmdir *nome-diretório*

Remove o diretório especificado na máquina remota.

I.1.3. Comandos Locais

!*[comando] [argumentos]*

Executa o comando especificado com os respectivos argumentos na máquina local, sem abandonar ftp.

lcd *[diretório]*

Troca o diretório de trabalho na máquina local. Se nenhum diretório é especificado, o diretório base do usuário é usado.

I.2 - TELNET

Através deste programa é possível executar comandos em um outro computador também conectado à rede, especificando as características da conexão. A sintaxe básica de TELNET é:

telnet [*máquina*]

telnet pode funcionar de dois modos diferentes: modo de entrada e modo de comandos.

I.2.1. Modo de Entrada

Se invocado com o argumento máquina, telnet executa um comando open com este argumento, passando para o modo de entrada. No modo de entrada, o texto digitado é enviado para a máquina remota e interpretado como comandos do sistema operacional na máquina remota.

O argumento máquina pode ser o rótulo, o nome completo ou o endereço Internet da máquina. Veja os exemplos abaixo:

\$ telnet gauss	(rótulo da máquina remota)
\$ telnet gauss.cesup.ufrgs.br	(nome completo da máquina remota)
\$ telnet 143.54.21.5	(endereço Internet da máquina remota)

I.2.2. Modo de Comandos

Caso telnet seja invocado sem o argumento máquina, ele entra no modo de comandos, indicado pelo sinal de pronto (telnet>)

```
kelvin% telnet  
telnet>
```

Neste modo, o usuário pode entrar com qualquer um dos comandos de telnet, que podem ser abreviados, ou seja, digitados incompletos.

Para acessar o modo de comandos a partir do modo de entrada, basta teclar o caracter de escape de telnet, que é exibido no início da sessão (o padrão é

"^]", que corresponde às teclas <ctrl>-<]>). Automaticamente telnet passa a executar o modo de comandos.

Nem todos os comandos estão disponíveis em todas as implementações de telnet. Os comandos básicos são:

? ou help [*comando*]

Fornece ajuda sobre o telnet. Sem argumentos, um sumário dos comandos é mostrado. Se o comando é especificado, telnet imprime a informação específica disponível para este comando.

open [*máquina*]

Abre uma conexão até a máquina especificada. A especificação da máquina pode ser tanto o rótulo da máquina, quanto o seu endereço internet, especificado em notação de pontos.

close

Termina uma conexão de telnet e retorna ao modo de comando.

Quit

Termina qualquer conexão de telnet aberta, retornando ao sinal de pronto da máquina local.

escape [*caracter de escape*]

Define o caracter de escape para o telnet. Caracteres de controle podem ser especificados como ""(<CTRL> seguido simplesmente por uma letra; por exemplo, <ctrl>-x é ""x".

status

Mostra o estado atual de telnet. Estas informações incluem a máquina destino e caracter de escape.

APÊNDICE II - O Software REDUCE

O programa REDUCE pode ser usado como uma calculadora algébrica, no sentido de permitir cálculos isolados como a diferenciação ou manipulação de expressões algébricas, ou pode ser usado como uma linguagem de programação. Aqui será dada ênfase ao uso do REDUCE como calculadora algébrica e rapidamente serão apresentados alguns comandos de programação.

O REDUCE é uma linguagem interativa, isto é, cada comando é processado após se pressionar a tecla <<ENTER>>, e a resposta é dada antes do segundo comando ser introduzido. Este modo de interação é útil para correção de programas em REDUCE.

O REDUCE usa os símbolos usuais **, /, *, -, + para exponenciação, divisão, multiplicação, subtração e adição respectivamente para dar um valor numérico ou algébrico a uma variável. O comando de atribuição é usado da seguinte forma :=, por exemplo:

```
1: z:=a+1;
```

```
Z := A + 1
```

O ; no final do comando é um terminador. Todos os comandos em REDUCE devem terminar com ; ou com \$. O terminador "\$" difere do ";" pois ele não exhibe na tela o resultado da operação apesar do comando ser processado internamente da mesma forma que o terminador ";". Pode-se realizar operações sem guardar em nenhuma variável, por exemplo:

```
2: z**5;
```

$$A^5 + 5 * A^4 + 10 * A^3 + 10 * A^2 + 5 * A + 1$$

outros exemplos:

```
3: a1:=m+n;
```

$$A1 := M + N$$

```
4: a2:=m-n;
```

$$A2 := M - N$$

$$5: a1*a2;$$

$$M^2 - N^2$$

O Reduce contém um conjunto de funções internas que são: SIN, COS, TAN, COT, ASIN, ACOS, ATAN, SQRT (\sqrt{x}), EXP (e^x), LOG, SINH, COSH, TANH, ASINH, ACOSH, ERF, DILOG, EXPINT.

As três últimas funções citadas são: Função erro, di logaritmo e integral de $\frac{e^x}{x}$. Essas funções já têm implícitas algumas regras de simplificação, por exemplo:

$$1: \cos(-x)$$

$$\text{COS}(X)$$

$$2: \log(\exp(x));$$

$$X$$

$$3: \sin(5*PI);$$

$$0$$

As funções são um caso particular de operadores internos que tem uma grande utilidade no cálculo algébrico, por exemplo: DF para diferenciação, INT para integração, SOLVE para resolução de equações algébricas, etc. Exemplos de diferenciação:

$$1: f:=(x-y)**2;$$

$$F := X^2 - 2 * X * Y + Y^2$$

$$2: df(f,x);$$

$$2 * X - 2 * Y$$

$$3: df(f,x,2);$$

$$2$$

$$4: df(f,x,y);$$

É importante salientar que foram calculadas as seguintes derivadas: $\frac{\partial f}{\partial x}$, $\frac{\partial^2 f}{\partial x^2}$ e $\frac{\partial^2 f}{\partial y \partial x}$. Pode-se tomar as derivadas parciais de qualquer ordem de uma função de várias variáveis.

5: clear f;

6: df(f,x);

0

O comando CLEAR faz com que F volte a ser um indeterminado e portanto, não dependa mais de X. É bastante útil um comando que faça F ser uma função qualquer de X, esse comando é:

7: depend f,x;

8: df(f,x);

DF(F,X)

Agora a derivada não é mais nula, mas sim um indeterminado, que pode ser usado em expressões. A f pode depender de várias variáveis, por exemplo, da forma:

9: depend f,x,y,z;

Neste caso, $F=F(x,y,z)$. Para cancelar a dependência utiliza-se o comando NODEPEND.

Outro operador útil em cálculos algébricos é o SUB (SUBstitution). Por exemplo:

1: f:=x*y;

F := X*Y

2: SUB(X*R*COS(TETA),Y*R*SIN(TETA),F);

COS(TETA)*SIN(TETA)*R²

3: SUB(TETA=PI/4,WS);

R²/2

Observações:

1. Dentro do SUB usa-se "=" ao invés de ":=".

2. A expressão que se quer substituir em uma ou mais variáveis deve vir no final do parênteses do SUB.

Um outro operador bastante usado é o SOLVE, que resolve equações algébricas. Exemplo:

```
1: solve(log(asin(x+3))**2-8,x);  
   SOLN(1,1) := SIN(1/E(2*SQRT(2)))+3  
   SOLN(2,1) := SIN(1/E(2*SQRT(2)))+3
```

2

As soluções são guardadas na matriz SOLN e o número dado no final é o número de soluções. Note a existência de constantes internas e reais arbitrárias. É possível usar o SOLVE também para resolver sistemas de equações. Por exemplo, para o sistema:

$$\begin{cases} X + AY = 4 \\ 2X - Y = 1 \end{cases}$$

Use a seguinte notação:

```
4: SOLVE(LST(X+A*Y=4,2*X-Y=1),LST(X,Y));  
SOLN(1,1) := ( A + 4)/(2*A + 1)  
SOLN(1,2) := 7/(2*A + 1)
```

1

A solução para X é SOLN(1,1) e para Y é SOLN(1,2).

Além dos operadores internos podemos definir novos operadores através do comando OPERATOR. Por exemplo:

```
1: operator F,H1;
```

Podemos agora especificar quem são estes operadores:

```
2: let F(x,y)=x**2-y**2;
```

O comando LET define regras de substituição ou atribuição. Da forma como ele foi usado acima, não podemos calcular diretamente F(1,3) por exemplo, a menos que usemos o comando SUB. Para definir F como uma função, dê o comando:

```
3: for all x,y let F(x,y)=x**2 + y**2;
```

4: F(1,3);

(8)

5: F(sin a, 3*b);

SIN²(A) - 9*B²

Pode-se dar algumas propriedades aos operadores como linearidade, simetria, antissimetria e não comutatividade através dos comandos LINEAR, SYMMETRIC, ANTISYMMETRIC e NONCOM, respectivamente. Por exemplo:

6: linear H1;

7: H1(A*X+B,x);

H1(x,X)*A + H1(1,X)*B

8: DEPEND A,X;

9: INPUT 18;

H1(A*X,X) + H(1,X)*B

Note que temos que especificar qual é a variável na qual o operador é linear. Em 7: Vemos que H1 é linear na variável X de forma que os indeterminados A e B saem do operador. Em 9: A é uma função de X de forma que sofre influência do operador. Para os operadores U e V não comutarem proceda da seguinte forma:

10: OPERADOR U,V;

11: NONCOM U,V;

12: U()*V()-V()*U();

U()*V() - V()*U()

Note que o argumento dos operadores pode ter várias variáveis ou nenhuma variável.

O REDUCE possui o cálculo matricial implementado internamente,. Para declarar A1 uma matriz 2x1, B1 2x2 e C1 como uma matriz de tamanho indeterminado, dê o comando:

1: Matrix A1(2,1),B1(2,2),C1;

As componentes das matrizes com tamanho especificado como é o caso de A1 e B1, adquirem o valor zero no momento da declaração. Para dar valores aos elementos de uma matriz não nula, use o comando MAT, da forma:

```
2: B1:=MAT((2*A,1),(5,3));
```

```
B1(1,1) := 2*A
```

```
B1(1,2) := 1
```

```
B1(2,1) := 5
```

```
B1(2,2) := 3
```

Cada linha tem que estar entre parênteses e elas devem ser separadas por vírgulas. Podemos também dar valores componente a componente, o que é mais prático no caso de uma matriz com elementos grandes, ou com muitas componentes (Isso pode ser feito em um arquivo editado pelo editor de textos e depois carregado para o REDUCE através do comando IN"arquivo";

Podemos achar a inversa de B1 da forma B1**(-1) ou:

```
3: 1/B1;
```

```
MAT(1,1) := 3/(6*A - 5)
```

```
MAT(1,2) := (-1)/(6*A - 5)
```

```
MAT(2,1) := (-5)/(6*A - 5)
```

```
MAT(2,2) := (2*A)/(6*A - 5)
```

o determinante:

```
4: DET B1;
```

```
6*A - 5
```

o traço:

```
5: TRACE B1;
```

```
2*A + 3
```

A matriz transposta:

```
6: TP B1;'
```

```
MAT(1,1) := 2*A
```

MAT(1,2) := 5

MAT(2,1) := 1

MAT(2,2) := 3

Como um exemplo note que o sistema:

$$\begin{cases} 2ax + y = 1 \\ 5x - 3y = \pi \end{cases}$$

Pode ser resolvido da seguinte forma:

7: C1:=1/B1*(mat((1),(Pi)));

C1(1,1) := (- PI + 3)/(6*A - 5)

C1(2,1) := (2*PI*A - 5)/(6*A - 5)

A solução pode ser conferida com o comando SOLVE. Observe, que teríamos errado se tivéssemos escrito A1:=MAT((1),(PI))/B1, pois as matrizes B1 e MAT((1),(PI)) não cometam, e estaria errada a forma C1:=1/B1*MAT(1,PI) pois significaria o produto de uma matriz 2x2 por uma 1x2.

Um ARRAY pode ser pensado como a representação de um tensor. Pode ser um vetor caso tenha apenas um índice ou um tensor de ordem mais alta e de qualquer dimensão. Exemplos:

1: array v1(10), v2(5,9), tens(2,3,4);

Ao se definir um ARRAY os seus elementos adquirem um valor zero, por exemplo:

2: TENS(1,0,2);

0

Para especificar um ARRAY não nulo, temos que dar valores componente a componente. Existem comandos que facilitam essa tarefa caso o ARRAY tenha uma regra de construção. Por exemplo, um ARRAY vetorial cujo valor de cada componente é o valor de seu índice, pode ser especificado da forma:

3: FOR K:=0:10 DO WRITE V1(K):=K

V1(0) := 0

V1(1) := 1

V1(2) := 2

...

Pode-se ao mesmo tempo dar valores às componentes e exibir na tela, por exemplo:

5: FOR J=0:5 DO FOR L=0:2 DO WRITE V2(J,L):=L+J;

V2(0,0) := 0

V2(0,1) := 1

V2(0,2) := 2

....

As declarações ARRAY, MATRIX e OPERATOR tem alguns pontos em comum e tem diferenças que devem ser levadas em conta na escolha de uma aplicação particular. Aqui segue um resumo comparativo:

- O menor valor de um índice de um ARRAY é 0, de uma matriz é 1 e de um operador é irrestrito podendo ser numérico ou algébrico.

- É necessário especificar o tamanho do ARRAY no momento de sua declaração, enquanto que para matrizes não é necessário até que se refira às suas componentes. Para operadores não se declara tamanho.

- Os Arrays podem ter vários índices, as matrizes dois e os operadores um número variável.

- Somente as matrizes possuem os recursos do cálculo matricial já implementados internamente.

- Somente os operadores admitem que seus elementos sejam usados como indeterminados em uma expressão. Se em uma expressão aparecem elementos de um ARRAY ou de uma matriz, os seus valores são imediatamente substituídos. Lembre que ao se declarar uma matriz ou um array, as suas componentes adquirem valor nulo.

O Reduce permite que se guarde o contexto dos cálculos que estão sendo efetuados a qualquer momento. Da mesma forma que foram armazenados, é possível recuperar os dados.

O formato de arquivo que o Reduce aceita é o ASCII.

Para armazenar qualquer expressão em disco é necessário que se executem os seguintes passos:

- 1: out "*** Nome arquivo ";
- 2: Expressão:=Expressão;
- 3: shut "*** Nome arquivo ";

Da mesma forma em que foram armazenadas as expressões é possível recupera-las:

- 1: in "*** Nome arquivo ";

O Reduce permite que os resultados dos cálculos sejam colocados em notação do fortran através da chave fort.

- 1: b:=4;
- 2: on fort;
- 3: b;
- b = 4
- 4: off fort;

Existem uma série de comandos em REDUCE para usa-lo como uma linguagem de programação. É possível gerar loops, agrupar uma seqüência de comandos, criar comandos em especificações e parâmetros, etc.

O primeiro comando a ser apresentado é o FOR. Sua notação é a seguinte:

FOR índice= valor inicial STEP incremento UNTIL valor final

DO << comando 1;

comando 2;

... >>

SUM expressão algébrica;

PRODUCT expressão algébrica;

OBS: Se o valor do incremento é 1, a parte "STEP incremento UNTIL "pode ser substituída por dois pontos.

O índice usado no comando FOR é um índice mudo e não tem nenhuma interferência com uma variável de mesmo nome usada em outro lugar fora do FOR.

Os comandos FOR ... SUM e FOR ... PRODUCT retornam expressões algébricas como resultado enquanto que o FOR ... DO retorna o valor zero.

Exemplo:

1: ON FACTOR;

2: PROD:= FOR k=0:4 product(x+k);

prod:= (X + 4)*(X + 3)*(X + 2)*(X + 1)*X

Outros comandos:

Comando para criação de operadores com parâmetros:

PROCEDURE operador(expressões, parâmetros);

Comando condicional:

IF condição THEN comando1 ELSE comando2

Comandos para gerar loops:

WHILE condição DO comando

REPEAT comando UNTIL condição

Comando de bloco:

BEGIN;

seqüência de comandos;

RETURN resultado;

END;

Comando GO TO para usar dentro de bloco:

GO TO identificador;

Identificador: comando;

Nos comandos acima, onde aparece a palavra condição devemos usar uma "expressão booleana". Esse tipo de expressão retorna o valor verdadeiro ou falso dependendo se a condição é satisfeita ou não.

Exemplos de expressões booleanas são:

$k > 5$

$x = -1$

A tabela abaixo apresenta as principais chaves utilizadas pelo Reduce:

Chave	Padrão	Função
ALLFAC	ON	Fatora termos simples comuns nas expressões
BIGFLOAT	OFF	Permite o uso de precisão real arbitrária, em coeficientes de polinômios. Para ser usada em conjunção com a chave NUMVAL.
DIV	OFF	Cancela fatores simples em expressões com denominadores e gera termos com potência negativa.
EXP	ON	Expande as expressões em fatores com coeficientes inteiros
FACTOR	OFF	Fatora as expressões em fatores com coeficientes inteiros
FLOAT	OFF	Permite o uso números decimais com a precisão especificada pelo comando PRECISION N; onde N é um inteiro.
FORT	OFF	Faz com que as saídas das expressões sejam na notação do FORTRAN
GCD	OFF	Cancela expressões comuns ao numerador e denominador em expressões racionais
LIST	OFF	Escreve expressões colocando cada termo em uma linha
MCD	ON	Gera denominadores comuns nas expressões
NAT	ON	Quando desligada, gera expressões numa forma compatível com a entrada de comandos do REDUCE, inibindo a saída natural que é com expoentes levantados
NERO	OFF	Inibe a impressão de resultados iguais a zero
NUMVAL	OFF	Faz com que os argumentos numéricos das expressões sejam calculados e exibidos como números decimais quando a chave BIGFLOAT está ligada
RAISE	ON	Faz com que as letras minusculas sejam convertidas em maiusculas. Estão excluídos os caracteres precedidos por ! e os usados no comando WRITE " "

RAT	OFF	Faz com que os denominadores das expressões sejam colocados embaixo de cada termo. Essa chave deve ser usada em conjunção com o comando FACTOR
RATIONAL	OFF	Permite o uso de coeficientes racionais em polinômios
REDUCED	OFF	Fatora termos dentro de expressões com potências racionais.
RESUBS	ON	Quando desligada, inibe a recursividade do comando LET
TIME	OFF	Exibe o tempo de CPU dos comandos do REDUCE

Na tabela abaixo são apresentados os principais operadores disponíveis em Reduce. Em todos os operadores a *variável* pode ser substituída por um KERNEL. Um KERNEL é uma forma funcional. Exemplos: x , $\cos(a*b)$, $\exp(\sin(a+b))$.

Nome	Função
ABS(numero)	Retorna o valor absoluto do número
COEFF(polínomio, variável, armazenador)	Parte o polinômio em seus coeficientes com relação a variável e os armazena no armazenador que pode ser um ARRAY ou um identificador.
DEG(polínomio, variável)	Retorna o grau do polinômio
DEN(expressão racional, expressão)	Retorna o denominador da expressão
DET(matriz)	Retorna o determinante da matriz
DF(expressão, variável, número, var, num, ...)	Calcula a derivada parcial de ordem especificada pelo número da expressão com relação à variável
FACTORIZE(polínomio, armazenador)	Retorna o determinante da matriz
GCD(polínomio1, polínomio 2)	Retorna o maior divisor comum entre o polinômio 1 e o polinômio 2
INT(expressão, variável)	Retorna a integral da expressão com relação a variável
LCOF(polínomio, variável)	Retorna o coeficiente do termo de maior grau do polinômio
LTERM(polínomio, variável)	Retorna o termo de maior grau do polinômio
MAX(num1, num2, ...)	Retorna o maior número
MIN(num1, num2, ...)	Retorna o menor número
NUM(expressão racional)	Retorna o numerador da expressão
PART(expressão, número)	Retorna a parte da expressão especificada pelo número
PRECISION(número)	Fixa a precisão dos números reais. Para ser usado em conjunção com as chaves FLOAT e BIGFLOAT

REMAINDER(polinômio1, polinômio2)	Retorna o resto da divisão entre o polinômio 1 e o polinômio 2
SOLVE(expressão, variável)	Resolve a equação expressão com relação à variável
SUB(variável1, variável2, ..., expressão)	Substitui as variáveis variavel1, variavel2, na expressão
TP(matriz)	Calcula a transposta da matriz
TRACE(matriz)	Calcula o traço da matriz

A tabela abaixo apresenta alguns comandos e declarações disponíveis em reduce.

ANTISYMMETRIC op1, op2, ...	Declara os operadores op1, op2 como antisimétricos em seus argumentos
ARRAY var1(num1, num2, ...), var2(num1, num2)	Declara as variáveis var1, var2, ... como arrays de tamanho especificado pelos números num1, num2.
BYE	Interrompe a execução do reduce sem guardar os resultados dos cálculos feitos durante a seção.
CLEAR exp1, exp2, ...	Faz com que as expressões exp1, exp2, ... voltem a ser indeterminados.
COMMENT texto;	Introduz comentários até encontrar o terminador ; ou \$
CONT	Serve para continuar o processamento de um arquivo de lote a partir do ponto em que tenha sido interrompido com o comando PAUSE
DEFINE exp1, exp2, ...	As expressões exp1, exp2, ... devem ser equações. Esse comando define o lado esquerdo da equação como sendo o lado direito.
DEPEND var1, var2, ...	Faz com que a variável var1 dependa dos kernels var2, var3, etc.
DISPLAY numero	Mostra na tela os últimos comandos
ED número	Editado para correção o comando especificado pelo número
END	Usado para terminar um arquivo ou um bloco. Serve também para transferir o sistema para o LISP
FACTOR exp1, exp2	Declara as expressões exp1, ... como fatores na saída das expressões subsequentes
FOR	Comando para gerar loops
FORALL var1, var2, ... comando	Declara as variáveis var1, etc, como arbitrarias na substituição de comando.
GO TO var1	Transfere o cálculo para o comando precedido por var1. Esse comando deve ser usado dentro do bloco
IF	Usado para definir comandos condicionais

INTEGER var1, var2, ...	Declara as variáveis var1, var2, ... como variáveis inteiras locais em comandos de blocos
IN var1, var2, ...	Processa os arquivos externos var1, var2, etc, que devem conter comandos de REDUCE
LET exp1, exp2, ...	As expressões exp1, exp2, ... devem ser equações. Esse comando faz com que o lado esquerdo da equação seja substituído pelo lado direito toda vez que ele aparecer em alguma expressão.
Linear var1, var2, ...	Declara os operadores var1, var2, como sendo lineares em seus argumentos
MATRIX var1(num1, num2, ...)	Declara as variáveis como matrizes de tamanho num1 x num2
NODEPEND var1, var2	Remove a dependência da variável var1 com relação a var2, var3, etc.
NONCOM id1, id2, ...	Declara os operadores id1, id2, ..., como não comutativas entre si com relação a multiplicação
OFF var1, var2, ...	Desliga as chaves var1, var2, ..., etc
ON var1, var2, ...	Liga as chaves var1, var2, ... , etc
OPERATOR var1, var2, ...	Declara var1, var2, ... como operadores
OUT "var1"	Declara var1 como um arquivo de saída
PAUSE	Comando interativo para ser usado em arquivo de lote. Esse comando interrompe o processamento de um arquivo de lote
PROCEDURE	Dá nome a comandos para uso repetido em cálculos
QUIT	Interrompe a execução de uma seção do REDUCE
REAL var1, var2, ...	Declara as variáveis var1, var2, ... como variáveis reais locais em comandos de bloco
REMFAC exp1, exp2,...	Remove a declaração de que exp1, exp2, etc, sejam fatores em expressões
RETURN exp1	Para ser usado em comandos compostos, retorna o valor de exp1
SAVEAS ex1	Guarda em exp1 a última expressão calculada. Este comando é equivalente a: exp1:=ws\$
SCALAR var1, var2, ...	Declara as variáveis var1, ... como variáveis escalares locais em comandos de bloco
SHOWTIME	Fornece o intervalo de tempo desde a última vez que esse comando foi dado ou desde o início da seção
SHUT var1, var2,	Fecha os arquivos de nomes var1, var2, ... , etc, abertos com o comando out "var1,..."

SYMMETRIC id1, id2, ..	Declara os operadores id1, id2, ..., etc, como simétricos em seus argumentos
WRITE exp1, exp2, ...	Retorna o valor das expressões exp1, exp2, etc. Se alguma dessas expressões estiver entre aspas, retorna exatamente o que está escrito entre aspas

Apendice III - Softwares para determinação da Constante de Resfriamento

Os softwares apresentados nesta seção foram utilizados para a determinação dos valores da Constante de Resfriamento ϕ com os diversos métodos numéricos utilizados. É apresentado também o software utilizado para o cálculo do ϕ via método da Decomposição.

Todos os softwares foram desenvolvidos utilizando-se a linguagem Fortran. Além disto, a entrada de dados e a função erro implementada é a mesma para todos. Sendo assim, a descrição da entrada de dados e da função erro é apresentada apenas uma vez. Nos demais itens, é apresentado apenas o algoritmo em fortran para o cálculo, com seus critérios de parada.

As primeiras linhas do programa são de comentários a respeito das modificações feitas no software durante o seu desenvolvimento.

Procura seqüencial

```
C -----
C Data da criacao .....: 14/10/1993
C Data da ultima alteracao .....: 19/01/1994
C Programmer .....: Jacques D.B.
C Compilador .....: Fortran 5.0
C -----
C Software para localizacao do valor de phi correto
C -----
C 11/10 - Composicao do arquivo de entrada (a ser lido pelo programa):
C +-----+
C + Variavel + Significado fisico      + Unidade  +
C +-----+
C + Tf      + Temperatura de fusao do metal + G. Celsius +
C + To      + Temperatura ambiente      + G. Celsius +
C + Tv      + Temperatura de vazao      + G. Celsius +
C + K1      +          +          +
C + a1      +          +          +
C + k2      +          +          +
C + a2      +          +          +
C + lf2     +          +          +
C + cp2     +          +          +
C -----
C 20/10 - Foi retirada a rotina de impressao dos dados na tela, e criada
C uma implementacao onde o software lê um arquivo de dados, con-
```

C tendo diversos modelos de teste e rastreia todos e armazena em
C um outro arquivo de dados, que será lido por um outro software
C que calculará os resultados e um terceiro software gerará uma
C tabela com todos os dados processados.
C O nome do arquivo é tabela.dat

C-----
C 26/10 - Nesta data foi colocado a formatação para a leitura dos dados em
C arquivo, para facilitar o entendimento por parte do programador
C a respeito dos dados que estão sendo processados.

C-----
C 19/01 - Nesta data, foram feitas as modificações para a geração dos números
C a serem incluídos na tese

C-----
open(unit=10,file='tabela02.dat')
open(unit=6,file='search02.dat')
open(unit=11,file='molde.dat')
read(10,86) N
86 format(l2)
read(11,77) a1,ak1
77 format(F5.3,2x,F5.3)
do 930 i=1,N
write(*,164) i,N
164 format(' Processando ',l2,' de ',l2)
read(10,99) tf,to,tv,a2,ak2,alf2,cp2
99 format(F4.0,1x,F2.0,1x,F4.0,1x,F4.2,1x,F4.2,1x,F5.2,1x,F4.3)
write(*,101) tf,to,tv,a1,ak1,a2,ak2,alf2,cp2
101 format(F6.0,1x,F5.0,1x,F5.0,1x,F5.3,1x,F5.3,1x,F5.3,1x,F5.3,
.1x,F10.5,1x,F5.3)

C-----
C Definição das variáveis que serão utilizadas no programa

C-----
pi=3.1415926535897

C-----
C m grande - Constante do solido/liquido

C-----
b1=ak1/a1**0.5
b2=ak2/a2**0.5
AM=b2/b1

C-----
C S estrela - Superaquecimento adimensional

C-----
sest=(tv-tf)/(tf-to)

C-----
C H estrela - Calor de fusão adimensional

C-----
hest=alf2/(cp2*(tf-to))

C-----
C m pequeno - Relação solido/liquido

C-----
am1=1

```

        rpi=pi**0.5
        p=am1*sest/(hest*rpi)
        q=1/(hest*rpi)
        x=-.2
        e=exp(1)
        ap=0.0001
500  fx1=am1*sest*exp(x**2)/(1-erf(x))
        fx2=-exp(-x**2)/(am+erf(x))
        fx3=hest*x*rpi
        soma=fx1+fx2+fx3
        if (abs(soma).lt.1e-3) goto 950
            x=x+ap
            if (x.gt.1.5) goto 950
            goto 500
950  write(*,414) x,soma
414  format('x .....: ',F10.8, ' F(x) ....: ',F10.8)
        write(6,18) x,soma
18   format(F15.10,5x,F15.12)
930  continue
        end
        function erf(x)
            n = 100
            spar = 0
            simpar = 0
            delta = x / n
            do 100 j = 1,n - 1,2
100   simpar = simpar + funcao(j * delta)
            do 200 j = 2,n - 2,2
200   spar = spar + funcao(j * delta)
            zero=0
            total=delta*(2*spar+4*simpar+funcao(x)+funcao(zero)) / 3
            erf = 1.12837917671955*total
            return
        end
        function funcao (x)
            funcao = exp(-x**2)
            return
        end

```

Bissecção

C Definicao das variaveis que serao utilizadas no programa

C-----
 pi=3.1415926535897

C-----
 C m grande - Constante do solido/liquido

C-----
 b1=ak1/a1**0.5
 b2=ak2/a2**0.5
 AM=b2/b1

C-----

C S estrela - Superaquecimento adimensional

```
C-----  
      sest=(tv-tf)/(tf-to)
```

```
C-----  
C H estrela - Calor de fusao adimensional
```

```
C-----  
      hest=alf2/(cp2*(tf-to))  
      h=hest
```

```
C-----  
C m pequeno - Relacao solido/liquido
```

```
C-----
```

```
      am1=1  
      rpi=pi**0.5  
      p=am1*sest/(hest*rpi)  
      q=1/(hest*rpi)  
      x=0  
      e=exp(1)  
      a=-1  
      b=1  
      conta=0  
500  x=(a+b)/2  
      conta=conta+1  
      fa=fdex(sest,am,hest,rpi,a)  
      fb=fdex(sest,am,hest,rpi,b)  
      fx=fdex(sest,am,hest,rpi,x)  
      ars=fa*fx  
      if (ars.lt.0) goto 765  
      a=x  
      goto 546  
765  b=x  
546  write(*,193) a,b,x,conta  
193  format(' a : ',F15.8,' b : ',F15.8,' x : ',F15.8,10x,F5.0)  
      if (abs(fx).lt.1e-4) goto 950  
      if (i.eq.5.or.i.eq.8.and.conta.gt.18) goto 950  
      if (conta.gt.40) goto 950  
      goto 500  
950  write(*,191) x  
      write(6,195) x,fx  
195  format(F15.10,10x,F15.7)  
191  format(' Valor de x .....: ',F15.8)  
      write(*,192) fx  
192  format(' F(x) .....: ',F15.10)  
930  continue  
      end  
      function fdex(sest,am,hest,rpi,x)  
      fx1=sest*exp(x**2)/(1-erf(x))  
      fx2=-exp(-x**2)/(am+erf(x))  
      fx3=hest*x*rpi  
      soma=fx1+fx2+fx3  
      fdex=soma
```

```
return
end
```

NEWTON

```
C-----
C Definicao das variaveis que serao utilizadas no programa
C-----
      pi=3.1415926535897
C-----
C m grande - Constante do solido/liquido
C-----
      b1=ak1/a1**0.5
      b2=ak2/a2**0.5
      AM=b2/b1
C-----
C S estrela - Superaquecimento adimensional
C-----
      sest=(tv-tf)/(tf-to)
C-----
C H estrela - Calor de fusao adimensional
C-----
      hest=alf2/(cp2*(tf-to))
      h=hest
C-----
C m pequeno - Relacao solido/liquido
C-----
      am1=1
      rpi=pi**0.5
      p=am1*sest/(hest*rpi)
      q=1/(hest*rpi)
      x=0
      e=exp(1)
      write(*,121) q,sest,am
121  format(' Valor do q .....: ',F15.10,/,
.      ' Valor do s* .....: ',F15.10,/,
.      ' Valor do M .....: ',F15.10)
500  fx1=sest*exp(x**2)/(1-erf(x))
      fx2=-exp(-x**2)/(am+erf(x))
      fx3=hest*x*rpi
      soma=fx1+fx2+fx3
      D1=(2.*S*(SQRT(PI)+E**(X**2)*ERF(X)*PI*X-E**(X**2)*PI
.      *X))/(E**(2.*X**2)*PI*(ERF(X)**2-2.*ERF(X)+1.))
      D2=(2.*(SQRT(PI)+E**(X**2)*ERF(X)*PI*X+E**(X**2)*AM*
.      PI*X))/(E**(2.*X**2)*PI*(ERF(X)**2+2.*ERF(X)*AM+AM**
.      2))
      D3=SQRT(PI)*H
      sder=d1+d2+d3
      s=x-soma/sder
      write(*,193) x,soma,sder
      x=s
```

```

193  format(' x .: ',F15.8,' Fx .: ',F15.8,' Der .: ',F15.8)
      if (abs(soma).lt.1e-4) goto 950
      goto 500
950  write(*,191) x
      write(6,195) x,soma
195  format(F15.10,10x,F15.10)
191  format(' Valor de x .....: ',F15.8)
      write(*,192) soma
192  format(' F(x) .....: ',F15.10)
930  continue
      end

```

Secante

```

C-----
C Definicao das variaveis que serao utilizadas no programa
C-----
      pi=3.1415926535897
C-----
C m grande - Constante do solido/liquido
C-----
      b1=ak1/a1**0.5
      b2=ak2/a2**0.5
      AM=b2/b1
C-----
C S estrela - Superaquecimento adimensional
C-----
      sest=(tv-tf)/(tf-to)
C-----
C H estrela - Calor de fusao adimensional
C-----
      hest=alf2/(cp2*(tf-to))
      h=hest
C-----
C m pequeno - Relacao solido/liquido
C-----
      am1=1
      rpi=pi**0.5
      p=am1*sest/(hest*rpi)
      q=1/(hest*rpi)
      x=0
      e=exp(1)
      a=.5
      b=1
      fa=fdex(sest,am,hest,rpi,a)
      fb=fdex(sest,am,hest,rpi,b)
      conta=0
500  conta=conta+1
      fb=fdex(sest,am,hest,rpi,b)
      x=b-(b-a)*fb/(fb-fa)
      fx=fdex(sest,am,hest,rpi,x)

```

```

a=b
fa=fb
b=x
fb=fx
write(*,193) x,fx
193 format(' x : ',F15.8,' fx : ',F15.8)
if (abs(fx).lt.1e-4) goto 950
if (i.eq.18.or.i.eq.21.or.i.eq.23) goto 143
goto 144
143 if (conta.eq.15) goto 950
144 if (conta.eq.50) goto 950
goto 500
950 write(*,191) x
write(6,195) x,fx
195 format(F15.10,10x,F15.7)
191 format(' Valor de x .....: ',F15.8)
write(*,192) fx
192 format(' F(x) .....: ',F15.10)
930 continue
end
function fdex(sest,am,hest,rpi,x)
fx1=sest*exp(x**2)/(1-erf(x))
fx2=-exp(-x**2)/(am+erf(x))
fx3=hest*x*rpi
soma=fx1+fx2+fx3
fdex=soma
return
end
function erf(x)
n = 100
spar = 0
simpar = 0
delta = x / n
do 100 j = 1,n - 1,2
100     simpar = simpar + funcao(j * delta)
do 200 j = 2,n - 2,2
200     spar = spar + funcao(j * delta)
zero=0
total=delta*(2*spar+4*simpar+funcao(x)+funcao(zero)) / 3
erf = 1.12837917671955*total
return
end
function funcao (x)
funcao = exp(-x**2)
return
end

```

Proposto

A listagem que segue é do software utilizado para o cálculo da constante de resfriamento pelo método proposto pelo presente trabalho. É importante salientar que as derivadas encontradas neste software foram geradas automaticamente pelo software Reduce.

```

C-----
C Data da criacao .....: 14/10/1993
C Data da ultima alteracao .....: 20/01/1994
C Programmer .....: Jacques D.B.
C Compilador .....: Fortran 5.0
C -----
C Software para calculo da constante de resfriamento de Schwarz
C utilizando-se o metodo de Adomian
C Nesta aproximacao, o x inicial eh zero e a aproximacao sera
C feita ate a terceira derivada da funcao
C O software utilizado foi o Reduce, no Vax
C O calculo da funcao erro eh feito utilizando-se o metodo de
C Simpson com 100 divisoes
C -----
C -----
C Composicao do arquivo de entrada (a ser lido pelo programa):
C +-----+
C + Variavel + Significado fisico      + Unidade  +
C +-----+
C + Tf      + Temperatura de fusao do metal + G. Celsius +
C + To      + Temperatura ambiente        + G. Celsius +
C + Tv      + Temperatura de vazao      + G. Celsius +
C + a1      +          +          +
C + k1      +          +          +
C + a2      +          +          +
C + k2      +          +          +
C + lf2     +          +          +
C + cp2     +          +          +
C -----
C 14/10 - Esta aproximacao foi criada para teste, devido ao fato de
C que a aproximacao by Rubens nao funciona para o molde cobre e por-
C que a aproximacao inicial da um valor muito alto.
C -----
C 11/11 - Neste dia, alterei, para constar todos os valores de entrada,
C para encontrar o porque que o cobre nao fecha como os outros
C casos.
C -----
C 20/01 - Neste dia alterei para que o software guardasse em arquivo
C todos os dados a serem verificados para ver porque e onde que
C o modelo nao funciona
C foi testado tambem uma verificacao de leitura de string
C -----
open(unit=10,file='tabela02.dat')

```

```

open(unit=12,file='jacq09.dat')
open(unit=11,file='molde.dat')
open(unit=1, file='parte.dat')
read(10,86) N
86 format (I2)
read(11,77) aa1,ak1
77 format(F5.3,2x,F5.3)
do 930 i=1,N
write(*,164) i,N
164 format(' Processando ',I2,' de ',I2)
read(10,99) tf,to,tv,a2,ak2,alf2,cp2
99 format(F4.0,1x,F2.0,1x,F4.0,1x,F4.2,1x,F4.2,1x,F5.2,1x,F4.3)
write(*,999) tf,to,tv,aa1,ak1,a2,ak2,alf2,cp2
999 format(' Temperatura de fusao .....: ',F10.5,/
. ' Temperatura ambiente .....: ',F10.5,/
. ' Temperatura de vazao .....: ',F10.5,/
. ' A1 .....: ',F10.5,/
. ' K1 .....: ',F10.5,/
. ' A2 .....: ',F10.5,/
. ' K2 .....: ',F10.5,/
. ' Lf2.....: ',F10.5,/
. ' Cp2.....: ',F10.5)
E = EXP(1)

```

C-----
C Definicao das variaveis que serao utilizadas no programa
C-----

```

PI = 3.1415926535897
rpi=pi**0.5

```

C-----
C m grande - Constante do solido/liquido
C-----

```

b1=ak1/aa1**0.5
b2=ak2/a2**0.5
AM = b2 / b1
write(*,10) b1,b2
10 format('B1 .....: ',F15.10,/
. ' B2 .....: ',F15.10)

```

C-----
C S estrela - Superaquecimento adimensional
C-----

```

sest=(tv-tf)/(tf-to)

```

C-----
C H estrela - Calor de fusao adimensional
FC-----

```

hest=alf2/(cp2*(tf-to))

```

C-----
C m pequeno - relacao solido/liquido
C-----

```

am1 = 1

```

```

write(*,890) am,sest,hest
890 format(/,' Valor do M .....: ',F15.10,/,
. ' S estrela .....: ',F15.10,/,
. ' H estrela .....: ',F15.10)
rp1=pi**0.5
q = hest * rpi
a1 = 1 - sest * AM
a2 = -1 - sest
x = 0
FX=(-(ERF(X)*A2+A1))/(E**(X**2)*(ERF(X)**2+ERF(X)*AM-
. ERF(X)-AM)*Q)
D1=(2.*((E**(X**2)*(ERF(X)**2+ERF(X)*AM-ERF(X)-AM)*X*
. PI+SQRT(PI)*(AM-1.)+2.*SQRT(PI)*ERF(X))*(ERF(X)*A2+
. A1)-SQRT(PI)*(ERF(X)**2+ERF(X)*AM-ERF(X)-AM)*A2))/(E
. **2.*X**2)*(ERF(X)**2+ERF(X)*AM-ERF(X)-AM)**2*Q*PI
D2=(2.*(((E**(X**2)*(2.*SQRT(PI)*((AM-1.)+2.*ERF(X))*
. X+E**(X**2)*(ERF(X)**2+ERF(X)*AM-ERF(X)-AM)*(2.*X**2
. +1.)*PI)*PI+4.*PI*(ERF(X)*A2+A1)+2.*SQRT(PI)*(E**(X
. **2)*(ERF(X)**2+ERF(X)*AM-ERF(X)-AM)*X*PI+SQRT(PI)*(
. AM-1.)+2.*SQRT(PI)*ERF(X))*A2-2.*((AM-1.)+2.*ERF(X))
. *PI*A2)*(ERF(X)**2+ERF(X)*AM-ERF(X)-AM)-4.*((E**(X**
. 2)*(ERF(X)**2+ERF(X)*AM-ERF(X)-AM)*X*PI+SQRT(PI)*(AM
. -1.)+2.*SQRT(PI)*ERF(X))*(ERF(X)*A2+A1)-SQRT(PI)*(
. ERF(X)**2+ERF(X)*AM-ERF(X)-AM)*A2)*(SQRT(PI)*((AM-1.
. )+2.*ERF(X))+E**(X**2)*(ERF(X)**2+ERF(X)*AM-ERF(X)-
. AM)*X*PI)))/(E**(3.*X**2)*(ERF(X)**2+ERF(X)*AM-ERF(X
. )-AM)**3*Q*PI**2)

ANS5=-((E**(X**2)*(2.*SQRT(PI)*((AM-1.)+2.*ERF(X))*X+
. E**(X**2)*(ERF(X)**2+ERF(X)*AM-ERF(X)-AM)*(2.*X**2+
. 1.)*PI)*PI+4.*PI*(ERF(X)*A2+A1)+2.*SQRT(PI)*(E**(X**
. 2)*(ERF(X)**2+ERF(X)*AM-ERF(X)-AM)*X*PI+SQRT(PI)*(AM
. -1.)+2.*SQRT(PI)*ERF(X))*A2-2.*((AM-1.)+2.*ERF(X))*
. PI*A2)*(2.*(SQRT(PI)*((AM-1.)+2.*ERF(X))+E**(X**2)*(
. ERF(X)**2+ERF(X)*AM-ERF(X)-AM)*X*PI)-SQRT(PI)*((AM-
. 1.)+2.*ERF(X)))-2.*(E**(X**2)*(2.*SQRT(PI)*((AM-1.)+
. 2.*ERF(X))*X+E**(X**2)*(ERF(X)**2+ERF(X)*AM-ERF(X)-AM
. )*(2.*X**2+1.)*PI)*PI+4.*PI*((E**(X**2)*(ERF(X)**2+
. ERF(X)*AM-ERF(X)-AM)*X*PI+SQRT(PI)*(AM-1.)+2.*SQRT(
. PI)*ERF(X))*(ERF(X)*A2+A1)-SQRT(PI)*(ERF(X)**2+ERF(X
. )*AM-ERF(X)-AM)*A2)

ANS4=(E**(X**2)*(E**(X**2)*(SQRT(PI)*((AM-1.)+2.*ERF(
. X))*(2.*X**2+1.))+E**(X**2)*(ERF(X)**2+ERF(X)*AM-ERF(
. X)-AM)*(2.*X**2+3.)*X*PI)*PI+E**(X**2)*(2.*SQRT(PI)*
. ((AM-1.)+2.*ERF(X))*X+E**(X**2)*(ERF(X)**2+ERF(X)*AM
. -ERF(X)-AM)*(2.*X**2+1.)*PI)*X*PI+SQRT(PI)*E**(X**2)
. *((AM-1.)+2.*ERF(X))*PI+4.*PI*X*(ERF(X)*A2+A1)*PI+
. 2.*SQRT(PI)*(E**(X**2)*(2.*SQRT(PI)*((AM-1.)+2.*ERF(X
. ))*X+E**(X**2)*(ERF(X)**2+ERF(X)*AM-ERF(X)-AM)*(2.*X
. **2+1.)*PI)*PI+4.*PI*A2-4.*SQRT(PI)*PI*A2)*(ERF(X)

```

```

**2+ERF(X)*AM-ERF(X)-AM)+ANS5
  ANS6=(ERF(X)**2+ERF(X)*AM-ERF(X)-AM)
  ANS3 = ANS4 * ANS6
  ANS7=-3.*(((E**(X**2)*(2.*SQRT(PI))*((AM-1.)+2.*ERF(X)
) *X+E**(X**2)*(ERF(X)**2+ERF(X)*AM-ERF(X)-AM))*(2.*X
**2+1.) *PI)*PI+4.*PI)*(ERF(X)*A2+A1)+2.*SQRT(PI)*(E
**(X**2)*(ERF(X)**2+ERF(X)*AM-ERF(X)-AM)*X*PI+SQRT(
PI)*(AM-1.)+2.*SQRT(PI)*ERF(X))*A2-2.*((AM-1.)+2.*
ERF(X))*PI*A2)*(ERF(X)**2+ERF(X)*AM-ERF(X)-AM)-4.*((
E**(X**2)*(ERF(X)**2+ERF(X)*AM-ERF(X)-AM)*X*PI+SQRT(
PI)*(AM-1.)+2.*SQRT(PI)*ERF(X))*(ERF(X)*A2+A1)-SQRT(
PI)*(ERF(X)**2+ERF(X)*AM-ERF(X)-AM)*A2)*(SQRT(PI))*((
AM-1.)+2.*ERF(X))+E**(X**2)*(ERF(X)**2+ERF(X)*AM-ERF
(X)-AM)*X*PI))*(SQRT(PI))*((AM-1.)+2.*ERF(X))+E**(X**
2)*(ERF(X)**2+ERF(X)*AM-ERF(X)-AM)*X*PI)
  ANS2 = ANS3 + ANS7
  ANS1 = 4! * ANS2
  D3=ANS1/(E**(4.*X**2)*(ERF(X)**2+ERF(X)*AM-ERF(X)-AM)
**4*Q*PI**3)

  a0 = fx
  ab1 = a0 * d1
  a2 = ab1 * d1 + a0 * a0 * d2 / 2
  a3 = a2 * d1 + a0 * ab1 * d2 + a0 * a0 * a0 * d3 / 6
  soma = a0 + ab1 + a2
  s0 = a0
  s1 = s0 + ab1
  s2 = s1 + a2
  s3 = s2 + a3
  write(*,826) fx,d1,d2,d3,soma
826 format(' F(x) .....: ',F15.10/,
  ' D1 .....: ',F15.10/,
  ' D2 .....: ',F15.10/,
  ' D3 .....: ',F15.10/,
  ' Resultado .....: ',F15.5)
  write(12,816) s0,s1,s2,s3
816 format(F15.8,3x,F15.8,3x,F15.8,3x,F15.8,3x,F15.8)
  write(1,911) b1,b2,am,sest,hest,fx,d1,d2,d3,soma
911 format(F12.8,2x,F12.8,2x,F12.8,2x,F12.8,2x,F12.8,2x,F12.8,
.2x,F12.8,2x,F12.8,2x,F12.8,2x,F12.8)
930 continue
  END
  FUNCTION erf (x)
  n = 100
  spar = 0
  simpar = 0
  delta = x / n
  do 100 j = 1,n - 1,2
100   simpar = simpar + funcao(j * delta)
  do 200 j = 2,n - 2,2

```



```
200    spar = spar + funcao(j * delta)
      zero = 0
      total = delta * (2 * spar + 4 * simpar + funcao(x) + funcao(zero)) / 3
      erf = 1.12837917671955 * total
      RETURN
      END
      FUNCTION funcao (x)
            funcao = exp(-x**2)
      RETURN
      END
```