

EXEMPLAR CORRESPONDE A REDAÇÃO FINAL
A TESE DEFENDIDA POR Carlos Eduardo Jeronymo da
COMISSÃO JULGADORA EM 07/10/91 E APROVADA PELA
Luiz Felipe Moura
ORIENTADOR

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA MECÂNICA

TESE DE MESTRADO

DESENVOLVIMENTO DE UM PROGRAMA NUMÉRICO APLICADO A PROBLEMAS
DE CONVECCÃO-DIFUSÃO USANDO O MÉTODO DOS ELEMENTOS FINITOS.
BASEADO EM VOLUMES DE CONTROLE.

Autor: Carlos Eduardo Jeronymo ¹⁹⁷⁴

Orientador: Prof. Dr. Luiz Felipe M. Moura

Aprovado por:

Luiz Felipe Moura
Prof. Dr. Luiz Felipe M. Moura, Presidente

Carlos Altemani
Prof. Dr. Carlos Alberto C. Altemani

Milton Mori
Prof. Dr. Milton Mori

Campinas, 7 de outubro de 1991

UNICAMP
BIBLIOTECA CENTRAL

10/10/91

AGRADECIMENTOS

Agradeço, em especial, a Margarete Marques por toda sua ajuda, encorajamento e amor que soube transmitir nos momentos mais difíceis.

A Kimihiro Ohno, estudante de engenharia mecânica da Universidade de Gifu (Japão), pelo empréstimo de um microcomputador IBM, que tornou possível a finalização dessa tese.

A Hilton Furquin pela elaboração dos desenhos.

A Rubens Stuginski Junior por suas sugestões e auxílio em muitas fases da elaboração desse trabalho.

Ao meu orientador Luiz Felipe M. Moura pelo seu auxílio e orientação.

A Deus por ser a razão de tudo existir.

RESUMO

Este trabalho descreve o desenvolvimento de um programa chamado FEMEY, no qual o método dos elementos finitos, baseado em volumes de controle, é utilizado na análise numérica de problemas envolvendo convecção-difusão. A formulação empregada é essencialmente conservativa, diferindo dos métodos convencionais para elementos finitos, geralmente baseados no método dos resíduos ponderados ou em princípios variacionais. O método é aplicado a uma série de problemas apresentando bons resultados.

ABSTRACT

This work presents a control volume finite element program for application in the analysis of convection-diffusion transport problems. The method has a formulation strictly conservative that is a contrast to the conventional weighted residual or variational approaches to the finite element method formulation. The method was applied to some test problems, giving good results.

ÍNDICE

	página
CAPÍTULO 1	
INTRODUÇÃO	1
CAPÍTULO 2	
0 MÉTODO DOS ELEMENTOS FINITOS BASEADO EM VOLUMES DE CONTROLE	10
2.1- Conceitos de Elementos Finitos.....	10
2.2- Conceitos de Vetor Normal e Integração em Coordenadas Locais	15
2.3- Definição de Volume de Controle	17
2.4- Discretização da Equação da Conservação de uma Grandeza Escalar para um Volume de Controle	19
2.4.1- Discretização do termo convectivo	23
2.4.2- Discretização do termo difusivo	24
2.4.3- Discretização do termo fonte	25
2.5- Operadores para os Pontos de Integração	26
2.5.1- Motivação para os pontos de integração	26
2.5.2- O operador difusivo	28
2.5.3- Termo fonte	30
2.5.4- O operador convectivo	31
2.5.5- Interação pontos de integração-equações de volume de controle	33
2.6- Implementação das Condições de Fronteira	35
2.6.1- Determinação dos fluxos nas fronteiras	35

2.6.2- Fluxo convectivo	36
2.6.3- Fluxo difusivo	37

CAPÍTULO 3

IMPLEMENTAÇÃO NUMÉRICA DO PROGRAMA FEMEY	40
3.1- Rotinas que Compõe o Programa FEMEY	40
3.1.1- Rotina MDIFF	42
3.1.2- Rotina MCONV	43
3.1.2.1- Rotina IPCONV	43
3.1.2.2- Rotina IPDIFF	47
3.1.2.3- Rotina MINV	47
3.1.2.4- Rotina PRODUT	47
3.1.3- Rotina FACES	48
3.1.4- Rotina EXPANDE	49
3.1.5- Rotina LIMITES	49
3.1.6- Rotina GERAC	50
3.1.7- Rotina SISTEMA	51
3.2- Descrição dos cartões de entrada	51

CAPÍTULO 4

RESULTADOS	59
4.1- Condução de Calor Unidimensional numa Aleta Triangular ...	59
4.2- Condução de Calor Bidimensional num Cilindro Sujeito à Convecção e Fluxo Imposto	62
4.3- Condução de Calor num Retângulo Sujeito à Geração Interna de Calor	64

4.4- Condução de Calor num Cilindro Oco Sujeito à Geração Interna e Calor e Condutividade Térmica dependente da Temperatura .	68
4.5- Condução Radial de Calor em um Cilindro Girante	71
4.6- Convecção em Degrau	74
4.7- Convecção-Difusão em um Fluxo Reverso de 180°	80
4.8- Convecção-difusão Unidimensional com Geração não Uniforme de calor	85
Conclusões e Sugestões	89
Referências	92
Apêndice A - O Coeficiente α	99
Apêndice B -Resultados Numéricos para o Problema 4.2 e tempos de CPU pelo número total de nós	103
Apêndice C -Programa FEMEY- Arquivos de entrada	105
Apêndice D -Listagem do programa	110

ÍNDICE DE FIGURAS

CAPÍTULO 2

2.1- Discretização de um domínio	11
2.2- Quadrilátero isoparamétrico de quatro nós	12
2.3- Vetor normal	15
2.4- Definição do vetor \vec{r}	16
2.5- Definição de volume de controle	18
2.6- Definição de subvolume de controle	18
2.7- Definição de pontos de integração	19
2.8- Volumes de controle interno e de fronteira	21
2.9- Integração dos volumes de controle	22
2.10- Fator de ponderação para os pontos de integração	28
2.11- Cálculo do operador difusivo	29
2.12- Comprimento de escala para o operador Laplaciano	30
2.13 (a)- Cálculo do ponto de intercessão	32
2.13 (b)- Cálculo do ponto de intercessão	32
2.13 (c)- Cálculo do ponto de intercessão	33
2.14- Volume de controle de fronteira	36
2.15- Fluxo difusivo numa fronteira	37

CAPÍTULO 3

3.1- Fluxograma do programa principal	41
3.2- Cálculo de fluxos em volumes de controle adjacentes	42
3.3- Rotina MCONV	44
3.4- Rotina IPCONV	45
3.5- Definição das retas de intercessão.....	46

3.6-	Segmentos de reta com possibilidades de serem interceptados pelo vetor velocidade com origem em i_{p1}	46
3.7-	Determinação dos segmentos de fronteira	48
3.8-	Volumes de controle situados em faces de canto	50
3.9-	Quadrantes de integração para o termo fonte	51
3.10-	Convenção utilizada para a numeração dos nós	54

CAPÍTULO 4

4.1-	Discretização da aleta triangular (malha 1×10)	59
4.2-	Comparação dos resultados para a aleta triangular (malha 1×5)	61
4.3-	Condução bidimensional num disco	62
4.4-	Linhas isotérmicas no interior do disco	63
4.5-	Histograma do erro para a condução de calor em um disco ...	64
4.6-	Condução num quadrado sujeito à geração interna	65
4.7-	Isotérmicas para um quadrado sujeito à geração interna de calor	66
4.8-	Histograma do erro para a condução num quadrado sujeito à geração interna de calor	67
4.9-	Cilindro oco com condutividade dependente da temperatura ..	68
4.10-	Comparação dos resultados obtidos para um cilindro oco (malha 1×20 elementos)	70
4.11-	Cilindro girante	72
4.12-	Resultados cilindro-girante	74
4.13-	Convecção em degrau	75
4.14.a-	Convecção em degrau ($\gamma_c = 0$)	77
4.14.b-	Convecção em degrau ($\gamma_c = 3$)	78

4.14.c- Convecção em degrau ($\gamma_c = 8$).....	78
4.14.d- Convecção em degrau ($\gamma_c = 10$).....	79
4.15.a- Comparação entre malha regular e malha irregular ($\gamma_c = 0$) método implementado	79
4.15.b- Comparação entre malha regular e malha irregular ($\gamma_c = 10$) método implementado	80
4.16- Fluxo reverso	81
4.17- Peclet = 10 (malha 20x10)	83
4.18- Peclet = 100 (malha 20x10)	83
4.19- Peclet = 500 (malha 20x10)	84
4.20- Peclet infinito (malha 20x10).....	84
4.21.a- Isotérmicas para fluxo reverso (Peclet = 10)	85
4.21.b- Isotérmicas para fluxo reverso (Peclet infinito)	85
4.22- Erro médio versus Peclet para convecção-difusão unidimensional com geração não uniforme de calor	88
4.23- Distribuição de temperaturas para convecção-difusão unidimensional com geração não uniforme de calor (Peclet=100)	88

Apêndice A

A1- Problema 4.8- Erro médio versus Peclet ($\alpha=10$)	101
A2- Problema 4.8- Distribuição de temperatura ($\alpha=10$)	101

ÍNDICE DE TABELAS

CAPÍTULO 4

4.1- Comparação dos resultados (aletas).....	61
4.2- Resultados para um quadrado sujeito a geração interna de calor	67
4.3- Comparação dos resultados para um cilindro oco.....	69
4.4- Número de iterações e tempo de CPU para a condução em um cilindro oco	71
4.5- Tempo de CPU por fases para condução num cilindro girante..	73
4.6- Tempo de CPU para a convecção em degrau	77
4.7- Variação da solução em função de α para o ponto (0.5,0.0)..	82

APÊNDICE A

A1-Tabela A1 - Ordem de convergência para Peclet	102
--	-----

APÊNDICE B

B1- Resultados Numéricos para o problema 4.2	103
B2- Tempos de CPU pelo número total de nós	104

SÍMBOLOS E ABREVIACÕES

- α - Coeficiente de difusão (m^2/s)
- C_p - Calor específico à pressão constante ($J/s-Kg-K$)
- $\Delta s, \delta$ - Comprimento (m)
- ϵ, η - Coordenadas locais (adimensional)
- ϕ, Φ - Variáveis escalares (temperatura [K])
- h - coeficiente de transferência de calor ($W/s-m^2-K$)
- λ - Coeficiente de difusão (m^2/s)
- N_j - Funções de interpolação
- $N_{,x}$ - Derivada de N em relação a x
- $N_{,y}$ - Derivada de N em relação a y
- $N_{,\epsilon}$ - Derivada de N em relação a ϵ
- $N_{,\eta}$ - Derivada de N em relação a η
- \dot{m} - Fluxo mássico (kg/s)
- Pe - número de Peclet ($\rho c_p \Gamma$)
- Q - Fluxo de calor (J/s)
- $Q_{i,j}$ - Fluxo de calor do subvolume de controle i para o subvolume de controle j
- ψ - Direção de uma linha de corrente
- ρ - Densidade (kg/m^3)
- s - Geração interna de calor (W/m^3)
- Γ - Condutividade (W/mK)
- Γ_x - Condutividade na direção x (W/mK)
- Γ_y - Condutividade na direção y (W/mK)
- T - temperatura (K)
- u - Componente do vetor velocidade na direção x

v - Componente do vetor velocidade na direção y

V - Velocidade resultante

x, y - Coordenadas globais (m)

CAPÍTULO 1

INTRODUÇÃO

A partir das décadas de 40 e 50, com o advento dos computadores eletrônicos, um maior esforço passou a ser dispendido no desenvolvimento de novas técnicas numéricas, para aplicações em diversos campos da ciência. No campo da termo-fluidodinâmica o desenvolvimento de novas técnicas numéricas, proporcionaram rápidos avanços tecnológicos, com a resolução de uma série de problemas, até então, impossíveis de serem tratados analiticamente. Dessa forma tornou-se possível o tratamento de fenômenos não lineares como turbulência, ondas de choque, convecção, etc. No entanto, mesmo com os grandes avanços nas técnicas numéricas, uma série enorme de problemas continuaram sem uma solução satisfatória.¹ Um exemplo disso, são as equações de Navier-Stokes, que mesmo em sua forma mais simples, continuam desafiando os pesquisadores na procura de um método mais apropriado e preciso para sua resolução.

Segundo Taylor,² as primeiras técnicas numéricas mais largamente utilizadas foram o método das características³ e o método das diferenças finitas.⁴ Esses métodos alcançaram um notável desenvolvimento, com refinamentos altamente sofisticados e, em circunstâncias particulares, podem ser considerados os melhores enfoques. No entanto, outros métodos vêm, pouco a pouco, ganhando uma maior popularidade. Um desses métodos é o método dos

elementos finitos.

O método dos elementos finitos é um procedimento aproximado para a resolução de equações diferenciais.⁵⁻⁸ Nesse método um meio contínuo é subdividido em pequenas subregiões de formas convenientes como triângulos, quadriláteros, etc. Uma solução aproximada é, então, formulada para cada uma dessas subregiões. Cada subregião assim gerada, é chamada elemento finito. A escolha de pontos convenientes, chamados nós, dentro desses elementos permite que as variáveis na equação diferencial sejam escritas como uma combinação linear de seus valores nodais e de suas derivadas, calculadas também nos pontos nodais.

Em problemas onde não exista um funcional, as equações diferenciais que governam o problema poderão ser transformadas em equações para elementos finitos por meio da aplicação do método dos resíduos ponderados ou do método dos volumes de controle (no qual este trabalho é baseado).

Independente do enfoque utilizado o processo de discretização de um domínio por elementos finitos envolverá, tipicamente, os seguintes passos:⁷

- (1)- Discretização do meio contínuo em elementos finitos;
- (2)- Constituição das matrizes dos elementos locais;
- (3)- Constituição da matriz global;
- (4)- Implementação das condições de contorno;
- (5)- Resolução do sistema global de equações discretizadas, por um método apropriado.

O uso do método dos elementos finitos, apesar de ser ainda bastante associado ao cálculo estrutural onde a existência

de um funcional (por exemplo o princípio dos trabalhos virtuais), facilita sua implementação, vem ganhando, como já foi mencionado, uma maior popularidade em problemas envolvendo transferência de calor ou escoamento de fluido. Em geral, nestes tipos de problemas não existe um funcional, tornando-se necessário a aplicação do método dos resíduos ponderados. Esse enfoque acarreta formulações essencialmente matemáticas, de pouco sentido físico. Só mais recentemente, com o desenvolvimento do método dos volumes de controle para elementos finitos, uma formulação conservativa e de maior sentido físico tornou-se disponível. Apesar de em muitas situações o método das diferenças finitas possuir certas vantagens sobre o método dos elementos finitos (principalmente quando se comparam requisitos de espaço de memória ou tempo de CPU), existem muitas situações onde a melhor solução é, sem dúvida, o emprego do método dos elementos finitos. Além de problemas com fronteiras irregulares, uma grande potencialidade do método encontra-se em situações onde o cálculo de tensões térmicas sejam importantes, por exemplo em estruturas aeronáuticas ou aeroespaciais. Nesses casos a mesma malha utilizada no cálculo estrutural pode ser utilizada nos cálculos térmicos, sem um trabalho adicional de geração de novas malhas.

Gradualmente os fatores que bloqueavam o uso do método dos elementos finitos em aplicações na área de transferência de calor e de mecânica dos fluidos tem sido superados. Há poucos anos atrás, a disponibilidade de CPU e tempo de processamento eram fatores, de certa forma, críticos no uso de elementos finitos. Contudo, a moderna tecnologia tem gerado novos computadores e supercomputadores com capacidade de armazenamento e velocidade de

processamento extremamente superiores às de 10 anos atrás. Hoje uma estação de trabalho, disponível em nossos laboratórios, pode ter sozinha uma capacidade de processamento superior à de um grande centro computacional de 10 anos atrás.

Como é bastante conhecido, os problemas que envolvem convecção-difusão, dado a sua característica de não linearidade, exigem técnicas especiais de tratamento para a sua solução numérica. Estas técnicas especiais incluem esquemas numéricos como upwind, exponencial, híbrido, etc, que são bastante desenvolvidos no método das diferenças finitas. Gradualmente, esquemas como esses têm sido adaptados ao método dos elementos finitos⁸⁻¹², sendo que o atual estado da arte permite soluções igualmente precisas em muitas situações. Naturalmente muito trabalho e pesquisa ainda serão necessários para expandir e aperfeiçoar o método.

Segundo Chung,⁴ o método dos elementos finitos tem suas origens nos anos 50, quando surge como uma ferramenta para a análise de tensões em grandes sistemas estruturais, como em estruturas aeronáuticas. O primeiro trabalho nesse assunto foi publicado por Turner et al.¹³ Esses autores foram seguidos por Clough¹⁴ e Argyris,¹⁵ entre outros. Aplicações do método dos elementos finitos a problemas não estruturais, como escoamentos e eletromagnetismo, foram primeiramente reportados por Zienkiewicz e Cheung.¹⁶ Oden¹⁷ contribuiu com aplicações para uma grande classe de problemas em mecânica não linear. O relacionamento da análise por elementos finitos aos clássicos conceitos variacionais de Rayleigh-Ritz¹⁸ ou ao método dos resíduos ponderados,¹⁹ estabeleceram uma importante ramificação da teoria de aproximação.

No campo de aplicações em transferência de calor e mecânica dos fluidos, Baliga e Patankar,²⁰ introduziram um novo enfoque, estritamente conservativo, ao apresentarem o método dos elementos finitos baseado em volumes de controle. Scheneider e Raw²¹⁻²² contribuíram com alguns refinamentos ao método dos volumes de controle.

Hoje várias teorias do comportamento de fluidos são disponíveis e englobam virtualmente qualquer tipo de fenômeno de interesse prático. Contudo, ainda há um número surpreendente de problemas não resolvidos em fluidodinâmica, devido às dificuldades encontradas na maioria dos métodos analíticos e numéricos convencionais. Num problema de convecção-difusão a dificuldade encontra-se em obter-se a distribuição de uma grandeza escalar na presença de um escoamento. Esta grandeza é transportada com o fluxo, enquanto seus gradientes causam o transporte por difusão.²⁰

Problemas envolvendo análise numérica dos fenômenos de convecção-difusão começaram a receber maior atenção dos pesquisadores a partir dos anos 60.²³⁻²⁴ As primeiras investigações nessa área empregavam o método das diferenças finitas baseado num esquema de diferenças centrais, e eram limitadas a problemas envolvendo baixos números de Peclet. Para números de Peclet altos, a formulação por diferenças centrais produzia resultados que exibiam oscilações espúricas e não realísticas fisicamente. Foi, então, descoberto que esse problema poderia ser evitado usando-se malhas computacionais muito finas. Contudo esse artifício era extremamente dispendioso. Esforços subsequentes para resolver esse problema resultaram no esquema *upwind*.²⁵⁻²⁶

Os esquemas *upwind* não são sujeitos a soluções oscilatórias em número de Peclet altos, mas são significativamente menos precisos que esquemas por diferenças centrais para pequenos números de Peclet. Num esforço para juntar as vantagens do *upwind* e diferenças centrais, Spalding²⁷ desenvolveu o esquema híbrido. Desde então, diversos refinamentos do esquema híbrido foram propostos.^{28,29} No entanto, Raithby,³⁰ aponta que esse esquema possui alguns defeitos sérios. Esse esquema funciona bem apenas em problemas em regime permanente ou aproximadamente permanente, onde o escoamento seja fortemente alinhado com as linhas da malha, ou onde não haja grandes gradientes. Em problemas que não satisfaçam essas condições, as hipóteses localmente unidimensionais usadas em esquemas híbridos poderão levar a uma falsa difusão.^{30,31} Quando essa falsa difusão é comparável ou maior que a difusão verdadeira, a solução resultante é imprecisa. Além disso, esquemas híbridos podem produzir erros significativos quando a equação diferencial possui um termo fonte não constante.^{30,32}

A falsa difusão que acompanha o esquema híbrido pode ser substancialmente reduzida por esquemas desenvolvidos para escoamentos multidimensionais.²⁸ Um exemplo de tal esquema é o *skew upwind* proposto por Raithby.³² Tal esquema mostra uma expressiva redução na falsa difusão mas não é completamente livre das dificuldades associadas aos termos fonte e problemas transientes. Os fatos acima relatados dão uma idéia das dificuldades encontradas na aproximação numérica de problemas de convecção-difusão e mostram as dificuldades associadas ao método das diferenças finitas.

Segundo Baliga e Patankar,²⁰ os métodos convencionais de

elementos finitos baseados no método de Galerkin, sofrem de dificuldades similares àquelas que afligem o esquema de diferenças centrais.³³ Para superar essas dificuldades, vários tipos de formulações *upwind* têm sido propostas. Christie et al.¹² formulou um esquema de elementos finitos *upwind* unidimensional baseado em funções ponderadas *skewed*. Este método foi estendido para problemas bidimensionais por Heinrich et al.³⁴ e Huyakorn.¹⁰ Heinrich e Zienkiewicz¹¹ estenderam este enfoque a elementos quadráticos. Um esquema de elementos finitos baseado numa regra especial de quadratura aplicada ao termo convectivo foi proposto por Hughes e Brooks.³⁵ Todos esses esquemas sofrem de falsa difusão similar àquela apresentada por esquemas híbridos de diferenças finitas; principalmente porque são generalizações diretas de idéias unidimensionais. Baliga e Patankar^{20,36-38} foram os primeiros a desenvolverem um método dos elementos finitos baseado em volumes de controle, usando elementos triangulares e uma função exponencial para a interpolação dos pontos. Este método recebeu uma série de melhorias por Hookey e Baliga,^{39,40} e Prakash.⁴¹⁻⁴³ Schneider^{9,21,22,44} apresentou também um método baseado em elementos finitos por volumes de controle, similar ao esquema *upwind*, mas com características próprias e com índices muito baixos de falsa difusão.

Dos vários métodos hoje existentes, o método dos volumes de controle parece ser bastante promissor em aplicações envolvendo convecção-difusão. Ele tem a capacidade de manipular domínios irregulares, possui a propriedade conservativa e a sua formulação é de fácil interpretação física. Neste enfoque as equações algébricas que representam o contínuo são geradas a partir de

equilíbrios de grandezas apropriadas como massa, energia, quantidade de movimento, etc.⁴⁴ Dessa forma para cada volume de controle haverá um conjunto de equações algébricas definindo a conservação de cada grandeza envolvida. A solução desse conjunto de equações algébricas deverá satisfazer exatamente as leis da conservação. Isto é um contraste com outros métodos, como por exemplo, o método de Galerkin, onde são usadas aproximações matemáticas dos termos da equação diferencial para gerar o sistema de equações algébricas

As vantagens do método dos volumes de controle são várias. A primeira resulta do esquema ser estritamente conservativo. Assim, quando uma quantidade especificada de uma dada variável é transportada para fora de um determinado volume de controle, a mesma quantidade é transportada para dentro do volume de controle adjacente ao primeiro. Dessa forma não há acréscimo ou decréscimo na grandeza transportada. Inexatidões que decorrem de malhas grosseiras, não são portanto resultado da não conservação da grandeza, mas devidas apenas a erros de aproximação.

A segunda vantagem vem do fato da discretização ser baseada num balanço de fluxos, seja difusivo, convectivo, ou outros. Dessa forma os termos resultantes no conjunto de equações algébricas têm um significado físico. Assim, um termo no sistema de equações algébricas, poderá representar a conservação de uma massa ou a conservação de uma grandeza escalar. Isso facilita a compreensão do efeito físico de certos processos, em oposto com os métodos variacionais ou dos resíduos ponderados que geram formulações essencialmente matemáticas.

Finalmente, a implementação do método dos volumes de

controle é direta e pouco custosa computacionalmente. Os cálculos de fluxo nas superfícies vizinhas de dois volumes de controle necessitam ser executados apenas uma vez, já que estes diferirão apenas no sinal. Isto acarreta uma redução no custo computacional, assim como uma maior simplicidade do algoritmo.

Nesse trabalho será implementado um programa computacional para aplicações envolvendo convecção-difusão. Nesta implementação será utilizado o método dos elementos finitos baseado em volumes de controle.²¹⁻²² O programa foi desenvolvido em linguagem Fortran-77, utilizando elementos quadrilaterais lineares (4 nós) do tipo isoparamétrico.

Nos próximos capítulos serão descritos maiores detalhes a respeito da implementação e aplicação do método a problemas de convecção-difusão.

CAPÍTULO 2

O MÉTODO DOS ELEMENTOS FINITOS BASEADO EM VOLUMES DE CONTROLE

Neste capítulo serão apresentados os conceitos básicos do método dos elementos finitos que permitirão a sua aplicação a problemas de convecção-difusão, segundo um enfoque estritamente conservativo. Neste enfoque, baseado no método dos volumes de controle, as equações algébricas resultantes da discretização do contínuo, provêm da elaboração de balanços de massa, energia, ou quantidade de movimento, em volumes de controle de forma definida e distribuídos ao longo do domínio. O método assegura a conservação integral de qualquer grandeza envolvida no problema, independente do número de nós. Assim mesmo numa malha pouco refinada, a conservação integral das grandezas será satisfeita, mesmo que a solução não seja muito precisa. Naturalmente, a precisão da solução encontrada, para um determinado problema, dependerá do número de nós com o qual o domínio foi discretizado.

Afim de que se possa compreender melhor o método, alguns conceitos serão estabelecidos.

2.1 Conceitos de Elementos Finitos

Como já foi dito anteriormente, um elemento finito é uma subregião de um meio contínuo que foi discretizado. Ele possui um tamanho finito e usualmente uma geometria mais simples que a do contínuo. A figura 2.1, exemplifica a discretização de um meio

contínuo em pequenas subregiões, cujo conjunto de equações algébricas para cada elemento, organizadas e agrupadas de forma global, representarão o domínio como um todo.

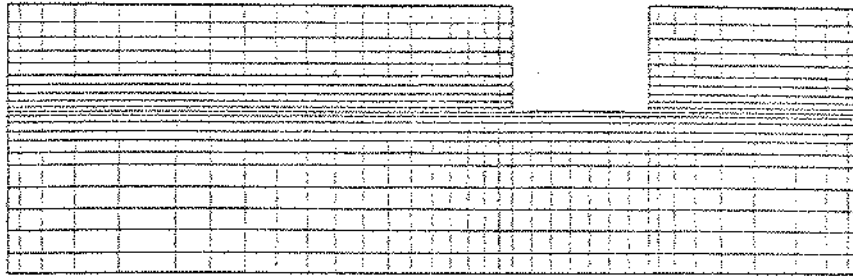


Figura 2.1 - Discretização de um domínio

Através do uso dessas subregiões (elementos finitos), poderemos converter um problema de infinitos graus de liberdade em um outro, com um número finito de graus de liberdade, o que simplificará o seu processo de solução. Todas as operações necessárias para a representação do comportamento físico do domínio serão executadas localmente em cada elemento.

Neste trabalho serão adotados elementos isoparamétricos de 4 nós (fig. 2.2), onde as funções que interpolam a geometria são as mesmas funções que interpolam as incógnitas. Os elementos isoparamétricos têm a vantagem de permitirem a definição de fronteiras irregulares, já que estes elementos podem se deformar. No entanto outros tipos de elementos poderiam ser usados. Informações mais detalhadas sobre diferentes tipos de elementos podem ser obtidas nas referências [6-8]. Naturalmente certas modificações deveriam ser executadas no programa, para a

utilização, por exemplo, de um elemento triangular. Como veremos nas próximas seções, ao invés de quatro subvolumes de controle, teríamos três. Ao invés de quatro pontos de integração seriam necessários apenas três, e assim por diante. Esses conceitos ficarão mais claros a medida que os conceitos teóricos aqui empregados sejam desenvolvidos.

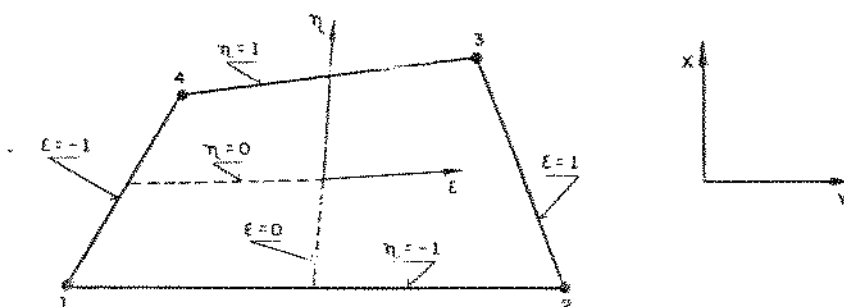


Figura 2.2 - Quadrilátero isoparamétrico de 4 nós.

No elemento mostrado na figura 2.2 é utilizado um sistema de coordenadas locais ϵ, η onde $-1 \leq \eta \leq 1$ e $-1 \leq \epsilon \leq 1$.

As coordenadas x, y ou o valor de uma variável qualquer, dentro do elemento, estarão relacionadas com o sistema de coordenadas locais pelas seguintes relações:

$$x = \sum_{i=1}^4 N_i x_i \quad (2.1.a)$$

$$y = \sum_{i=1}^4 N_i y_i \quad (2.1.b)$$

$$\phi = \sum_{i=1}^4 N_i \phi_i \quad (2.1.c)$$

onde N_i - são as funções de interpolação
 x_i, y_i, ϕ_i - valores nodais (nós de 1 a 4)

ϕ - uma variável genérica.

As funções de interpolação são dadas por:

$$\begin{aligned}N_1 &= 1/4 (1 - \epsilon) (1 - \eta) \\N_2 &= 1/4 (1 + \epsilon) (1 - \eta) \\N_3 &= 1/4 (1 + \epsilon) (1 + \eta) \\N_4 &= 1/4 (1 - \epsilon) (1 + \eta)\end{aligned}\tag{2.2}$$

As equações (2.1.a) e (2.1.b) fornecem as coordenadas globais em função das coordenadas locais do quadrilátero. A determinação das derivadas $\partial N/\partial \eta$ e $\partial N/\partial \epsilon$ é imediata, e por isso não é fornecida. O mesmo não ocorre para $\partial N/\partial x$ e $\partial N/\partial y$. Como as equações (2.1) são bilineares, as coordenadas ϵ e η não poderão ser expressas em termos das coordenadas globais x e y . Para encontrarmos os valores das derivadas $\partial N/\partial x$ e $\partial N/\partial y$ precisaremos usar a regra da cadeia. Seja:

$$\begin{aligned}\frac{\partial N}{\partial \epsilon} &= \frac{\partial N}{\partial x} \frac{\partial x}{\partial \epsilon} + \frac{\partial N}{\partial y} \frac{\partial y}{\partial \epsilon} \\ \frac{\partial N}{\partial \eta} &= \frac{\partial N}{\partial x} \frac{\partial x}{\partial \eta} + \frac{\partial N}{\partial y} \frac{\partial y}{\partial \eta}\end{aligned}\tag{2.3}$$

Em forma matricial:

$$\begin{bmatrix} \frac{\partial N}{\partial \epsilon} \\ \frac{\partial N}{\partial \eta} \end{bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial \epsilon} & \frac{\partial y}{\partial \epsilon} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix} \begin{bmatrix} \frac{\partial N}{\partial x} \\ \frac{\partial N}{\partial y} \end{bmatrix}$$

ou

$$\begin{bmatrix} N, \epsilon \\ N, \eta \end{bmatrix} = \begin{bmatrix} J \end{bmatrix} \begin{bmatrix} N, x \\ N, y \end{bmatrix}\tag{2.4}$$

onde $\begin{bmatrix} J \end{bmatrix}$ é a matriz Jacobiano, a qual contém as derivadas das

coordenadas globais com respeito às coordenadas locais. Estas derivadas são dadas por:

$$\begin{aligned}
 J_{11} = x, \epsilon &= N_{1,\epsilon} x_1 + N_{2,\epsilon} x_2 + N_{3,\epsilon} x_3 + N_{4,\epsilon} x_4 = \sum_{i=1}^4 N_{i,\epsilon} x_i \\
 J_{12} = y, \epsilon &= N_{1,\epsilon} y_1 + N_{2,\epsilon} y_2 + N_{3,\epsilon} y_3 + N_{4,\epsilon} y_4 = \sum_{i=1}^4 N_{i,\epsilon} y_i \\
 J_{21} = x, \eta &= N_{1,\eta} x_1 + N_{2,\eta} x_2 + N_{3,\eta} x_3 + N_{4,\eta} x_4 = \sum_{i=1}^4 N_{i,\eta} x_i \\
 J_{22} = y, \eta &= N_{1,\eta} y_1 + N_{2,\eta} y_2 + N_{3,\eta} y_3 + N_{4,\eta} y_4 = \sum_{i=1}^4 N_{i,\eta} y_i
 \end{aligned} \tag{2.5}$$

As expressões acima podem ser expressas mais facilmente na forma matricial:

$$[J] = 1/4 \begin{bmatrix} -(1-\eta) & (1-\eta) & (1+\eta) & -(1+\eta) \\ -(1-\epsilon) & -(1+\epsilon) & (1+\epsilon) & (1-\epsilon) \end{bmatrix} \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \\ x_4 & y_4 \end{bmatrix} \tag{2.6}$$

Multiplicando 2.4 por $[J^{-1}]$ (inverso do Jacobiano) teremos finalmente:

$$\begin{bmatrix} N, x \\ N, y \end{bmatrix} = [J^{-1}] \begin{bmatrix} N, \epsilon \\ N, \eta \end{bmatrix} \tag{2.7}$$

onde $\begin{bmatrix} J^{-1} \end{bmatrix}$ é dado pela expressão :

$$\begin{bmatrix} J^{-1} \end{bmatrix} = \frac{1}{|J|} \begin{bmatrix} J_{22} & -J_{12} \\ -J_{21} & J_{11} \end{bmatrix} \quad (2.8)$$

sendo $|J|$ o determinante de $\begin{bmatrix} J \end{bmatrix}$ definido por:

$$|J| = J_{11}J_{22} - J_{12}J_{21} \quad (2.9)$$

As equações acima definem todos os processos necessários para se calcular as derivadas globais de N em função de x e y .

2.2 Conceitos de Vetor Normal e Integração em Coordenadas Locais

As integrais a serem executadas ao longo das fronteiras de um volume de controle exigem o conceito de vetor normal. Seja o segmento de reta mostrado na figura 2.3, unindo (x_1, y_1) e (x_2, y_2) , a representação de uma fronteira de um volume de controle.

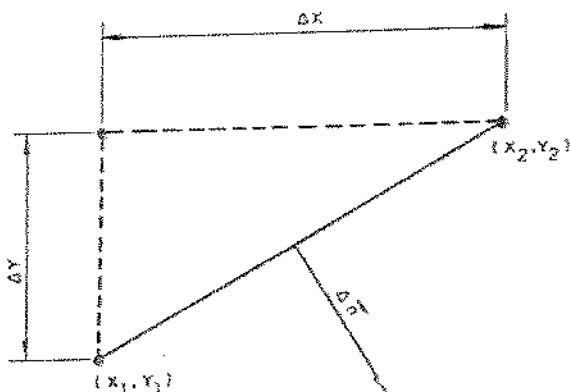


Figura 2.3 - Vetor Normal

Sendo $\Delta \vec{n}$ o vetor normal a este segmento teremos:

$$\Delta \vec{n} = \Delta y \vec{i} - \Delta x \vec{j} \quad (2.10)$$

onde:

$$\Delta x = x_2 - x_1$$

$$\Delta y = y_2 - y_1$$

Esse conceito será bastante usado nas integrações ao longo das fronteiras dos volumes de controle.

Em casos onde existam geração interna (ou termo fonte) será necessário calcular integrais de área da forma:

$$I = \iint s \, dx \, dy$$

Essa integral é mais facilmente avaliada no sistema de coordenadas locais ϵ, η . Isto poderá ser feito expressando g em função de ϵ e η e mudando-se os limites de integração para -1 a 1. Deve-se também, transformar a área infinitesimal para coordenadas locais. Para isto, definamos um vetor genérico \vec{r} , no interior do elemento (fig. 2.4), como sendo:

$$\vec{r} = x \vec{i} + y \vec{j}$$

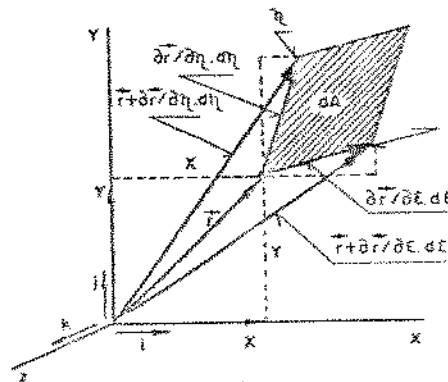


Figura 2.4 - Definição do vetor \vec{r}

Seu diferencial será dado por:

$$d\vec{r} = \frac{\partial \vec{r}}{\partial \epsilon} d\epsilon + \frac{\partial \vec{r}}{\partial \eta} d\eta = d\vec{\epsilon} + d\vec{\eta} \quad (2.11)$$

Um elemento infinitesimal de área será, então, definido no sistema de coordenadas locais pelo produto misto:

$$dA = (d\vec{\epsilon} \times d\vec{\eta}) \cdot \vec{k}$$

ou

$$dA = \left[\frac{\partial \vec{r}}{\partial \epsilon} d\epsilon \times \frac{\partial \vec{r}}{\partial \eta} d\eta \right] \cdot \vec{k} \quad (2.12)$$

Sendo:

$$\frac{\partial \vec{r}}{\partial \epsilon} = \frac{\partial x}{\partial \epsilon} \vec{i} + \frac{\partial y}{\partial \epsilon} \vec{j} \quad (2.13)$$

$$\frac{\partial \vec{r}}{\partial \eta} = \frac{\partial x}{\partial \eta} \vec{i} + \frac{\partial y}{\partial \eta} \vec{j}$$

Substituindo (2.13) em (2.12) obtemos:

$$dA = \left[\frac{\partial x}{\partial \epsilon} \frac{\partial y}{\partial \eta} - \frac{\partial y}{\partial \epsilon} \frac{\partial x}{\partial \eta} \right] d\epsilon d\eta$$

ou

$$dA = |J| d\epsilon d\eta \quad (2.14)$$

onde $|J|$ é o determinante da matriz jacobiano definido na equação (2.6).

2.3- Definição de Volume de Controle

Na discretização pelo método dos volumes de controle, para cada nó utilizado na discretização haverá um volume de controle correspondente, conforme está indicado na figura 2.6.

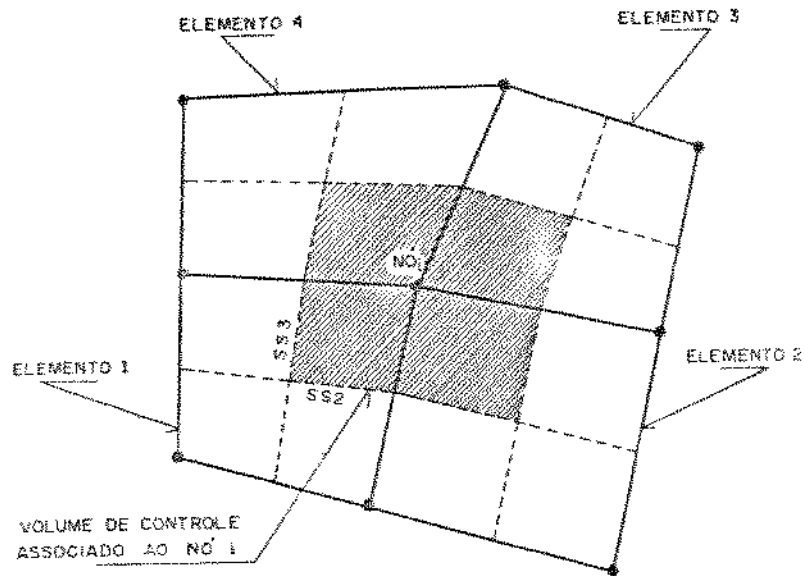


Figura 2.5 - Definição de volume de controle

Consideremos um elemento finito quadrilateral de 4 nós, conforme é mostrado na figura 2.6.

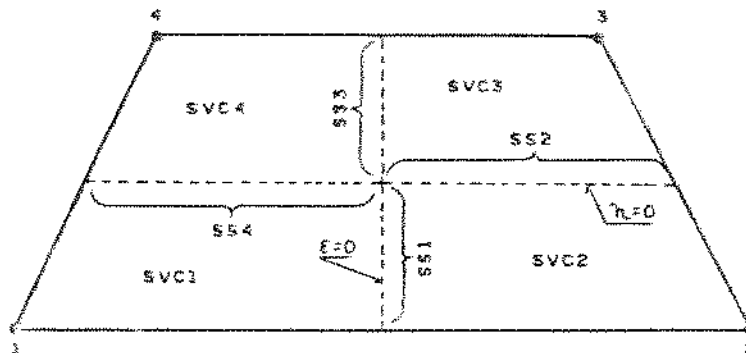


Figura 2.6 - Definição de subvolume de controle

Cada elemento será subdividido em 4 subvolumes de controle (SVCi), sendo que para cada nó, haverá um volume de controle correspondente. Como mostrado na figura 2.6, as fronteiras de cada subvolume de controle são tomadas como os

eixos $\eta = 0$, $\xi = 0$, e as fronteiras do elemento.

As superfícies de fronteira de cada subvolume de controle (segmentos SSi), internas ao elemento, serão utilizadas para o cálculo da conservação dos fluxos. No volume de controle da figura 2.5, onde 4 elementos possuem um nó em comum, haverá então, 8 segmentos de fronteira SSi, definindo o volume de controle total. O fluxo que atravessa os segmentos de fronteira SSi será integrado e o valor obtido nessa integração concentrado nos pontos médios de cada segmento. Esses pontos médios são chamados pontos de integração (ipi) e são mostrados na figura 2.7.

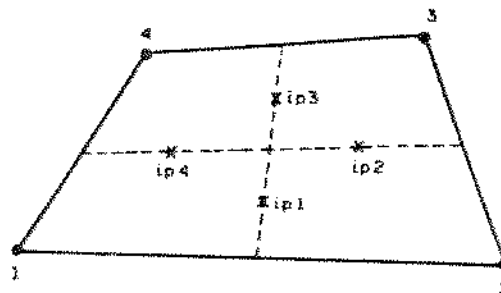


Figura 2.7 - Definição de pontos de integração

O relacionamento entre os valores das variáveis nos pontos de integração e nos pontos nodais é feito através da discretização de uma equação diferencial parcial. Maiores detalhes serão fornecidos posteriormente.

2.4- Discretização da Equação da Conservação de uma Grandeza Escalar para um Volume de Controle

A equação que descreve o fenômeno da convecção-difusão

de uma grandeza escalar ϕ , para um domínio bidimensional, pode ser expressa na seguinte forma:

$$u \frac{\partial \phi}{\partial x} + v \frac{\partial \phi}{\partial y} = \lambda \frac{\partial^2 \phi}{\partial x^2} + \lambda \frac{\partial^2 \phi}{\partial y^2} + s \quad (2.15)$$

onde:

λ - coeficiente de difusão, considerado constante

s - termo fonte

u, v - velocidade nas direções x e y respectivamente

ϕ - é uma grandeza escalar.

A grandeza escalar ϕ pode ser temperatura (ou entalpia), concentração de uma espécie química, parâmetros de turbulência, etc

O método dos volumes de controle se baseia na conservação de grandezas como, por exemplo, a energia em cada um dos volumes de controle que compõem um domínio. Dessa forma as equações algébricas resultantes da discretização de um domínio representarão um equilíbrio entre os termos difusivos, convectivos e fonte. A solução do conjunto de equações resultantes deverá satisfazer exatamente a conservação da grandeza envolvida (ou o balanço dos fluxos), embora um erro no valor das variáveis nodais ainda deva existir devido as aproximações empregadas.

A equação (2.15) é uma equação genérica para fenômenos de convecção-difusão. Nessa equação λ pode ser associado à difusividade [m^2/s] e s a um termo fonte [dimensão de ϕ/s]. Para efeito de discretização trabalharemos com a equação da conservação da energia. Assim para que s seja a geração interna de calor por unidade de volume [W/m^3] e para que λ seja a condutividade térmica [W/mK], o termo convectivo precisa ser

multiplicado por ρc_p [kg/m^3] [J/kg K]. Multiplicado-se então, o termo convectivo da equação (2.15) por ρc_p , teremos a equação da energia:

$$\rho c_p u \frac{\partial \phi}{\partial x} + \rho c_p v \frac{\partial \phi}{\partial y} = \Gamma_x \frac{\partial^2 \phi}{\partial x^2} + \Gamma_y \frac{\partial^2 \phi}{\partial y^2} + s \quad (2.16)$$

onde:

ϕ é a temperatura [K]

Γ_i é a condutividade térmica [W/mK]

s a geração interna de calor por unidade de volume
[W/m³]

Consideremos, agora, um nó genérico da figura 2.8. Este nó poderá pertencer tanto a um volume de controle interno como a um volume de controle de fronteira.

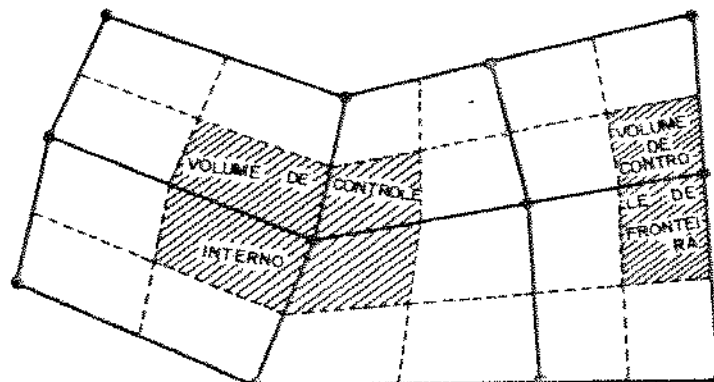


Figura 2.8 - Volumes de controle interno e de fronteira

Inicialmente, trataremos apenas de volumes de controles associados a nós internos (os volumes de controle de fronteira serão tratados separadamente). Partindo então, da equação (2.16), e integrando esta equação ao longo dos segmentos SSI associados ao

subvolume de controle do nó i (fig. 2.6) teremos:

$$\int_{SVC} (\rho c_p u \phi \vec{i} + \rho c_p v \phi \vec{j}) \cdot d\vec{n} - \left[\int_{SVC} \Gamma_x \frac{\partial \phi}{\partial x} \vec{i} + \Gamma_y \frac{\partial \phi}{\partial y} \vec{j} \right] \cdot d\vec{n} - \int_{SVC} s \, da = 0 \quad (2.17)$$

Estes termos representam o fluxo convectivo, o difusivo e a contribuição do termo fonte no interior de um elemento. O vetor $d\vec{n}$ representa o vetor normal à linha de integração ($d\vec{n} = dy\vec{i} - dx\vec{j}$), onde o sinal de dx e dy dependerão do sentido de rotação escolhido. Para o subvolume de controle SVC1 mostrado na figura 2.9, a equação (2.17) assumirá a forma:

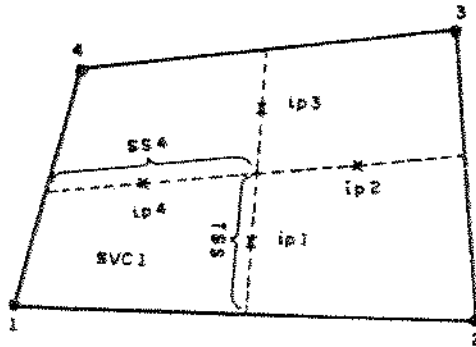


Figura 2.9 - Integração dos volumes de controle.

$$\int_{SS1 \& SS4} (\rho c_p u \phi \vec{i} + \rho c_p v \phi \vec{j}) \cdot d\vec{n} - \left[\int_{SS1 \& SS4} \Gamma_x \frac{\partial \phi}{\partial x} \vec{i} + \Gamma_y \frac{\partial \phi}{\partial y} \vec{j} \right] \cdot d\vec{n} - \int_{SVC1} s \, da = 0 \quad (2.18)$$

Separando-se, então, a equação (2.18) em seus termos convectivo, difusivo e fonte, discretizaremos cada um desses

termos isoladamente. O sentido de cálculo do vetor $d\vec{n}$ deverá ser sempre o mesmo. Nesse trabalho é adotado o sentido anti-horário. As discretizações serão desenvolvidas para o subvolume de controle 1, sendo os cálculos similares para os demais subvolumes de controle. Pode-se ver da figura (2.9), que no subvolume de controle 1, apenas os segmentos SS1 e SS4 contribuirão para o cálculo dos fluxos (como já foi mencionado, as fronteiras externas serão tratadas separadamente). Outra observação é quanto a representação das variáveis nos pontos nodais e nos pontos de integração. Sempre que uma variável estiver relacionada aos pontos nodais sua representação será feita por letras maiúsculas. Caso se refira aos pontos de integração sua representação será feita por letras minúsculas. Passaremos agora a discretização dos termos da equação (2.18).

2.4.1 - Discretização do termo convectivo

Discretizando o termo convectivo da equação 2.18 no segmento SS1:

$$\int_{SS1} \rho c_p \left[u \phi \vec{i} + v \phi \vec{j} \right] \cdot d\vec{n} \cong \rho c_p (u_1 \phi_1 \Delta y_1 - v_1 \phi_1 \Delta x_1)$$

analogamente para SS4 :

$$\int_{SS4} \rho c_p \left[u \phi \vec{i} + v \phi \vec{j} \right] \cdot d\vec{n} \cong \rho c_p (u_4 \phi_4 \Delta y_4 - v_4 \phi_4 \Delta x_4)$$

cuja soma será representada na forma:

$$\int_{SS1 \& SS4} (\rho c_p \vec{V} \phi) \cdot d\vec{n} = \sum_{j=1}^4 a_{i,j}^c \phi_j \quad (2.19)$$

onde para o subvolume de controle 1 a_{12} e a_{13} são nulos.

Na equação (2.19) o subíndice i refere-se ao subvolume de controle. Assim $i = 1$, refere-se ao subvolume 1, $i = 2$ ao subvolume 2, e assim por diante. O subíndice j refere-se ao ponto de integração (ip_1, ip_2, ip_3, ip_4). As letras minúsculas (a e ϕ) indicam que os valores estão relacionados aos pontos de integração. O superescrito g indica que o coeficiente provém de um termo convectivo. Deve-se esclarecer que os valores das variáveis nos pontos de integração são desconhecidos e que será necessário o desenvolvimento de novas equações que relacionem esses valores aos valores nodais. A forma como isto é feito será apresentado na seção 2.5.

2.4.2- Discretização do termo difusivo

Os termos de difusão são calculados, também, nos dois segmentos SS1 e SS4.

Para SS1:

$$-\int_{ss1} \left(\Gamma_x \frac{\partial \phi}{\partial x} \hat{i} + \Gamma_y \frac{\partial \phi}{\partial y} \hat{j} \right) \cdot d\vec{n} \cong - \Gamma_x \frac{\partial \phi}{\partial x} \Big|_{ip1}^{\Delta y_1} + \Gamma_y \frac{\partial \phi}{\partial y} \Big|_{ip1}^{\Delta x_1} \quad (2.20)$$

onde os valores para $\partial \phi / \partial x$ e $\partial \phi / \partial y$ são calculados nos pontos de integração (para $ip_1 \rightarrow \epsilon = 0, \eta = -1/2$)

Substituindo (2.1) em (2.20) temos:

$$\begin{aligned}
-\int_{s=1} \left(\Gamma_x \frac{\partial \phi}{\partial x} \hat{i} + \Gamma_y \frac{\partial \phi}{\partial y} \hat{j} \right) \cdot d\vec{n} &\cong -\int_{s=1} \left(\Gamma_x \sum_{j=1}^4 \frac{\partial N_j}{\partial x} \phi_j \hat{i} + \right. \\
&\left. \Gamma_y \sum_{j=1}^4 \frac{\partial N_j}{\partial y} \phi_j \hat{j} \right) \cdot d\vec{n} \cong -\Gamma_x \sum_{j=1}^4 \frac{\partial N_j}{\partial x} \bar{\phi}_j \Delta y_1 + \Gamma_y \sum_{j=1}^4 \frac{\partial N_j}{\partial y} \bar{\phi}_j \Delta x_1 \quad (2.21)
\end{aligned}$$

Na expressão acima a letra maiúscula $\bar{\phi}$ indica que o valor do ponto em questão é calculado por interpolação dos pontos nodais. Para o segmento SS4, os cálculos são similares. Dessa forma:

$$-\int_{s=1} \left(\Gamma_x \frac{\partial \phi}{\partial x} \hat{i} + \Gamma_y \frac{\partial \phi}{\partial y} \hat{j} \right) \cdot d\vec{n} = \sum_{j=1}^4 A_{i,j}^d \bar{\phi}_j \quad (2.22)$$

onde o superescrito d indica que o coeficiente provém de um termo de difusão, i ao subvolume de controle, j ao ponto nodal.

2.4.3 - Discretização do termo fonte

O termo fonte é calculado simplesmente integrando-se o valor de \underline{g} na região do subvolume de controle. Assim:

$$-\int_{svci} \underline{g} \, dA = -Bi^* \quad (2.23)$$

onde o superescrito \underline{g} indica que o termo está relacionado ao termo fonte.

Baseando-se nas discretizações vistas acima, podemos

então, representar a equação da conservação para um subvolume de controle i na seguinte forma:

$$\sum_{j=1}^4 A_{i,j}^d \bar{\phi}_j + \sum_{j=1}^4 a_{i,j}^c \phi_j = B_i \quad (2.24)$$

Considerando-se agora todos os quatro subvolumes de controle, cada qual com equações similares à equação (2.24), podemos reuni-las na forma de uma equação global da forma:

$$\begin{bmatrix} A^d \end{bmatrix} \begin{Bmatrix} \bar{\phi} \end{Bmatrix} + \begin{bmatrix} a^c \end{bmatrix} \begin{Bmatrix} \phi \end{Bmatrix} = \begin{Bmatrix} B \end{Bmatrix} \quad (2.25)$$

Nessa equação cada linha representará um subvolume de controle, associado a um valor nodal. Como foi mencionado anteriormente, os termos correspondentes à matriz $[a]$ são ainda desconhecidos e precisarão ser calculados. Em problemas de difusão o cálculo da matriz $[a]$ não será necessário.

2.5- Operadores para os Pontos de Integração

Nos itens anteriores desenvolvemos as discretizações para os termos convectivos em função dos chamados pontos de integração. Nessa seção serão desenvolvidas as equações algébricas necessárias para a determinação das variáveis nos pontos de integração.

2.5.1- Motivação dos pontos de integração

Nos enfoques mais convencionais para elementos finitos,

as interpolações para as variáveis nos pontos de integração, envolvem apenas o uso de séries de Taylor, ou de interpolações dos pontos nodais. Esses enfoques levam, contudo, aos conhecidos problemas de soluções oscilatórias em fluxos altamente convectivos. O maior inconveniente de tais enfoques é que esses tipos de interpolações não conseguem refletir as leis físicas que governam o problema. Este fato é ilustrado por Patankar²⁸, que propõe vários esquemas para contornar essa dificuldade. Neste trabalho é proposto um novo esquema,⁴⁴ cuja função é gerar equações adicionais que reflitam o comportamento dos termos convectivos nos pontos de integração.

Seja a equação diferencial parcial abaixo:

$$u \frac{\partial \phi}{\partial x} + v \frac{\partial \phi}{\partial y} - \alpha \nabla^2 \phi = s \quad (2.26)$$

Cada um dos termos dessa equação será aproximado algebricamente nos 4 pontos de integração. Como as aproximações num dado ponto de integração podem envolver os valores das variáveis em outros pontos de integração, em adição aos valores nodais, será necessário a inversão de uma matriz 4 x 4 para determinar-se os valores dos pontos de integração, em função dos valores nodais. Os valores dos pontos de integração resultantes dessas operações, serão então substituídos nas respectivas equações de volume de controle, gerando um sistema de equações em função apenas dos valores nodais. Em geral o valor de α deverá ser tomado como sendo igual ao valor do coeficiente de difusão. Em situações onde as linhas de corrente do escoamento não cruzem as fronteiras no mesmo ponto que o prolongamento do vetor \vec{V} (figura

2.10), α poderá ser tomado como um fator de ponderação entre os valores convectivo e fonte com o valor do termo difusivo. Em geral, haverá uma grande faixa de valores de α , para os quais a solução será convergente. Quanto mais difusivo for o problema, menor será a influência do valor de α . No apêndice A são fornecidos maiores detalhes a respeito da utilização do coeficiente α . Descreveremos a seguir, a maneira como os diversos operadores da equação (2.26) serão discretizados.

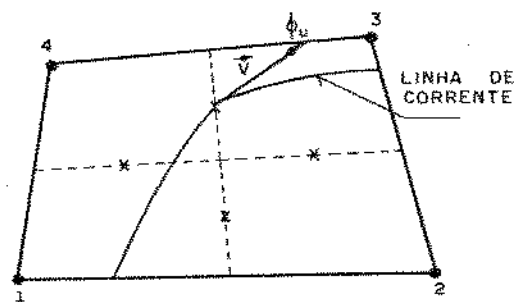


Figura 2.10 - Fator de ponderação para os pontos de integração

2.5.2- O operador difusivo

O operador difusivo é representado diferencialmente pelo Laplaciano. Devemos então, achar uma expressão que aproxime o operador:

$$L(\phi) = \nabla^2 \phi$$

Consideremos na figura (2.11) um segmento de reta unindo um nó i a um nó $i+1$. No ponto médio desse segmento o valor da segunda derivada em relação a x será dada pela expressão:

$$\frac{\partial^2 \phi}{\partial x^2} = \frac{\phi_i - 2\phi + \phi_{i+1}}{\Delta x^2 / 4} \quad (2.27)$$

Consideremos agora a figura 2.12. Usando a expressão (2.27), para o ponto ip_1 , nos segmentos indicados pelas figuras 2.12.a e 2.12.b, teremos:

$$\left. \frac{\partial \phi^2}{\partial x^2} \right|_{ip_1} = \frac{\frac{3}{8} \bar{\phi}_2 + \frac{1}{8} \bar{\phi}_3 - \phi_1 + \frac{1}{8} \bar{\phi}_4 + \frac{3}{8} \bar{\phi}_1}{\Delta x^2 / 2} = \frac{\sum_{j=1}^4 N_j \bar{\phi}_j - \phi_1}{L_{dx}^2} \quad (2.28.a)$$

e para a derivada segunda em relação a y (Fig. 2.12.b):

$$\left. \frac{\partial \phi^2}{\partial y^2} \right|_{ip_1} = \frac{\frac{3}{8} \bar{\phi}_2 + \frac{1}{8} \bar{\phi}_3 - \phi_1 + \frac{1}{8} \bar{\phi}_4 + \frac{3}{8} \bar{\phi}_1}{\Delta y^2 / 8} = \frac{\sum_{j=1}^4 N_j \bar{\phi}_j - \phi_1}{L_{dy}^2} \quad (2.28.b)$$

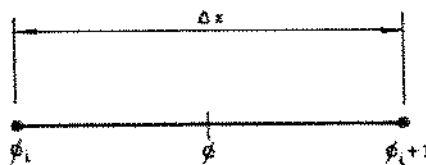
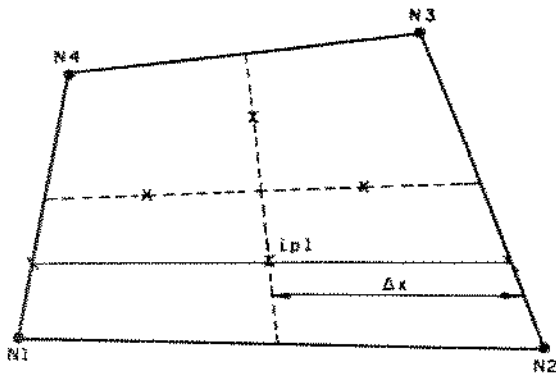
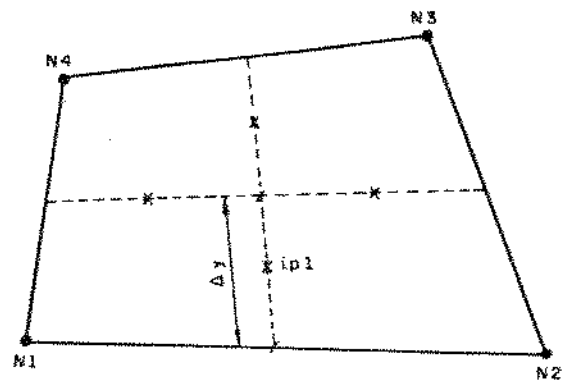


Figura 2.11 - Cálculo do operador difusivo



(a)



(b)

Figura 2.12 - Comprimento de escala para o operador Laplaciano

De forma mais compacta poderemos, então, representar o Laplaciano pela seguinte expressão:

$$\nabla^2 \phi \Big|_{ip1} = \frac{\sum_{j=1}^4 N_j \Phi_j - \phi_i}{L_d^2} = \sum_{j=1}^4 c_{i,j}^d \phi_j + \sum_{j=1}^4 C_{i,j}^d \Phi_j \quad (2.29)$$

onde L_d é um fator de escala para a difusão, que no caso de $ip1$ e $ip3$ valerá $\Delta x^2/2$ (direção x) ou $\Delta y^2/8$ (direção y). Para os pontos $ip2$ e $ip4$, L_d valerá $\Delta x^2/8$ (direção x) e $\Delta y^2/2$ (direção y).

2.5.3 - Termo fonte

O termo fonte é modelado substituindo-se o valor da fonte no correspondente ponto de integração. Sua representação será:

$$s \Big|_{ip1} = d_i \quad (2.30)$$

2.5.4 - O operador convectivo

O operador convectivo, aplicado a uma variável genérica ϕ tem a forma:

$$C(\phi) = u \frac{\partial \phi}{\partial x} + v \frac{\partial \phi}{\partial y}$$

onde \underline{u} e \underline{v} representam o campo de velocidades.

Na direção de uma linha de corrente:

$$C(\phi) = V \frac{\partial \phi}{\partial \psi} \quad (2.31)$$

onde \underline{V} é a velocidade do fluido $\left[V = (u^2 + v^2)^{1/2} \right]$ e ψ é a direção da linha de corrente cujo diferencial satisfaz:

$$d\psi = \frac{u}{V} dx + \frac{v}{V} dy$$

O operador convecção de equação (2.31), poderá, então, ser aproximado no ponto $i p_1$ por:

$$V \frac{\partial \phi}{\partial \psi} \Big|_{i p_1} = V \frac{\phi_u - \phi_i}{L_c} \quad (2.32)$$

onde ϕ_u refere-se ao ponto onde o vetor \vec{V} intercepta as fronteiras do elemento, e L_c é a distância entre $i p_1$ e o ponto de intercessão.

A figura 2.13 representa diferentes possibilidades para o cálculo de ϕ_u .

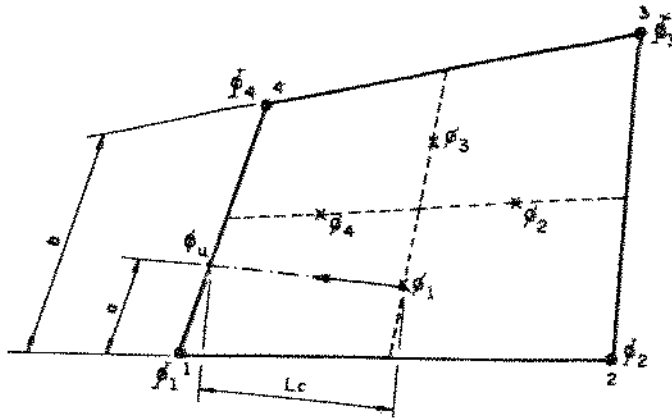


Figura 2.13(a) Cálculo do ponto de intercessão

Na figura (2.13.a) o valor de ϕ_u é interpolado entre ϕ_1 e ϕ_4 e expresso da seguinte forma:

$$\phi_u = \frac{a}{b} \phi_4 + \left[1 - \frac{a}{b} \right] \phi_1 \quad (2.33. a)$$

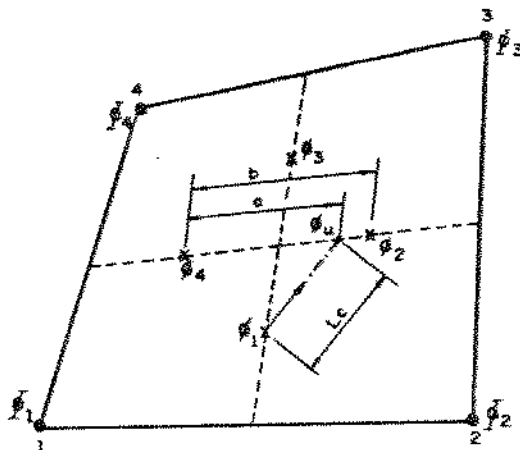


Figura 2.13.(b) Cálculo do ponto de intercessão

Nas situações da figura 2.13.(b)

$$\phi_u = \frac{a}{b} \phi_2 + \left(1 - \frac{a}{b} \right) \phi_4 \quad (2.33.b)$$

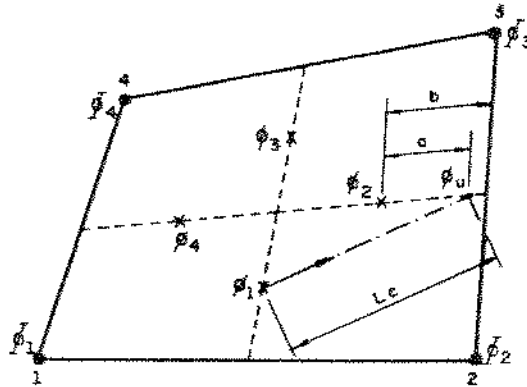


Figura 2.13.(c) Cálculo do ponto de intercessão

Na figura (2.13.c) o cálculo de ϕ_u será:

$$\phi_u = \left(1 - \frac{a}{b} \right) \phi_2 + \left(\frac{\phi_2 + \phi_4}{2} \right) \left(\frac{a}{b} \right) \quad (2.33.c)$$

Os valores de ϕ_u serão calculados de forma similar para outras situações de interesse que possam ocorrer.

Esse operador será representado pela seguinte expressão:

$$V \frac{\partial \phi}{\partial \psi} \Big|_{i p_i} = \sum_{j=1}^4 C_{i,j}^c \phi_j + \sum_{j=1}^4 c_{i,j}^c \phi_j \quad (2.34)$$

2.5.5- Interação pontos de integração-equações de volume de controle

As discretizações geradas para os pontos de integração

serão agora substituídas nas equações de volume de controle, a fim de gerar-se as equações globais do sistema. A discretização da equação (2.26) pode ser resumida na forma:

$$u \left. \frac{\partial \phi}{\partial x} \right|_{i p i} + v \left. \frac{\partial \phi}{\partial y} \right|_{i p i} = \sum_{j=1}^4 c_{i,j}^c \phi_j + \sum_{j=1}^4 C_{i,j}^c \bar{\phi}_j \quad (2.35. a)$$

$$\alpha \nabla^2 \phi \Big|_{i p i} = \sum_{j=1}^4 C_{i,j}^d \bar{\phi}_j + \sum_{j=1}^4 c_{i,j}^d \phi_j \quad (2.35. b)$$

$$s \Big|_{i p i} = d_i^* \quad (2.35. c)$$

Substituindo (2.35) em (2.26) e rearranjando os operadores:

$$\sum_{j=1}^4 (c_{i,j}^c - c_{i,j}^d) \phi_j + \sum_{j=1}^4 (C_{i,j}^c - C_{i,j}^d) \bar{\phi}_j = d_i^*$$

Na forma matricial, para todos os subvolumes de controle:

$$\begin{bmatrix} c^c & -c^d \end{bmatrix} \langle \phi \rangle + \begin{bmatrix} C^c & -C^d \end{bmatrix} \langle \bar{\phi} \rangle = \{ d_i^* \} \quad (2.36)$$

Da equação (2.36) pode-se explicitar $\langle \phi \rangle$ nos pontos de integração em função dos valores nodais:

$$\langle \phi \rangle = \begin{bmatrix} c^c & -c^d \end{bmatrix}^{-1} \left\{ \begin{bmatrix} C^d & -C^c \end{bmatrix} \langle \bar{\phi} \rangle + \{ d_i^* \} \right\} \quad (2.37)$$

o qual pode ser escrito mais compactamente como:

$$\langle \phi \rangle = [CC] \langle \bar{\phi} \rangle + \langle RCC \rangle \quad (2.38)$$

onde CC representa o produto da matriz inversa pela matriz dos coeficientes das equações dos pontos nodais, e RCC representa o produto da matriz inversa pela matriz do termo fonte.

Substituindo a equação das variáveis nos pontos de integração (2.38) na equação de volume de controle (2.25) teremos:

$$\left[[A^d] + [a^o] [CC] \right] \langle \bar{\phi} \rangle = \{ B^e \} - [a^o] \{ RCC \} \quad (2.39)$$

A equação (2.39) define, assim, a equação global do sistema.

2.6 - Implementação das Condições de Fronteira

Em situações que envolvam convecção, poderão haver fluxos atravessando as fronteiras, sem que haja uma variável nodal imposta. Nesses casos será necessário calcular os fluxos que atravessam as fronteiras externas, já que as equações geradas anteriormente tratavam apenas de nós internos. O procedimento adotado é descrito a seguir.

2.6.1 - Determinação dos fluxos nas fronteiras

Nas fronteiras o fluxo será composto por uma parte convectiva e outra difusiva. O fluxo atravessando uma fronteira

poderá, portanto, ser representado por uma expressão da forma:

$$Q_f = Q_f^c + Q_f^d \quad (2.40)$$

onde Q representa o fluxo de uma variável escalar ϕ e o subíndice f indica que este fluxo está relacionado com as fronteiras.

A figura 2.14 ilustra um volume de controle, situado numa face de fronteira. Usando-se a equação (2.40), o balanço de energia para um volume de controle situado numa fronteira será da forma:

$$(\text{fluxos internos}) + Q_f = 0 \quad (2.41)$$

onde fluxos internos refere-se aos fluxos calculados nos pontos $p1a$, $p1b$, $p2a$, $p2b$ (nestes pontos os fluxos são calculados por interpolação nodal).

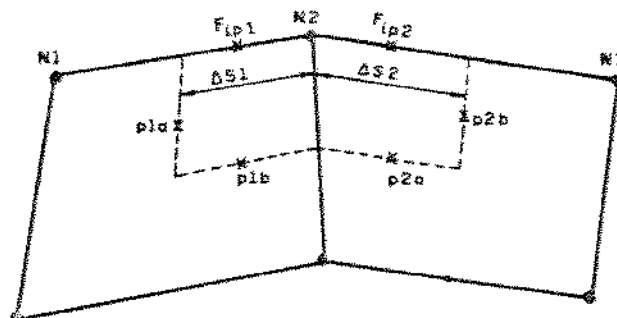


Figura 2.14 - Volume de controle de fronteira

2.6.2- Fluxo convectivo

A expressão para o fluxo convectivo será da forma:

$$Q_f^c = \int_{\text{ACP}} \rho \phi \vec{V} \cdot d\vec{n} \quad (2.42)$$

onde $d\vec{n}$ é o vetor normal à fronteira e \vec{V} , o vetor velocidade.
 Aproximando-se a expressão (2.42) para o ponto F_{ip1} :

$$Q_f^c \cong \dot{m}_{F_{ip1}} c_p \phi_{F_{ip1}} \quad (2.43)$$

onde \dot{m} é a massa que atravessa a fronteira e $\phi_{F_{ip1}}$ dado pela expressão:

$$\phi_{fip1} = a \Phi_1 + b \Phi_2 \quad (2.44)$$

onde:

$a = 0.75$; $b = 0.25$ que equivale a tomarmos F_{ip1} como sendo situado no ponto médio do segmento ΔS_1 (figura 2.14). Para o ponto F_{ip2} os cálculos são similares.

2.6.3 -Fluxo difusivo

O fluxo difusivo corresponderá às condições de temperatura imposta, fluxo de calor imposto ou convecção imposta.

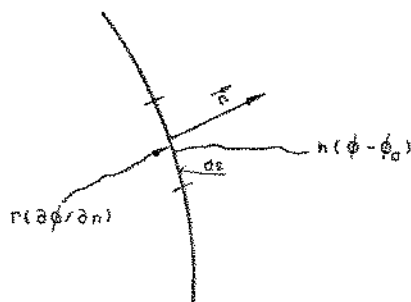


Figura 2.15 - Fluxo difusivo numa fronteira

No interior imediatamente após a fronteira a

transferência de calor será por condução (fig 2.15):

$$Q_r = -\Gamma \frac{\partial \phi}{\partial n} \quad (2.45)$$

Se a superfície externa tem convecção imposta:

$$Q = h (\phi - \phi_a) \quad (2.46)$$

onde h é o coeficiente de transferência de calor e Γ a condutividade térmica e ϕ_a é a temperatura do meio externo.

Igualando as equações (2.45) e (2.46):

$$-\Gamma \frac{\partial \phi}{\partial n} = h (\phi - \phi_a)$$

$$\Gamma \frac{\partial \phi}{\partial n} + h \phi = h \phi_a$$

ou

$$\Gamma \frac{\partial \phi}{\partial n} + h \phi = C \quad (2.47)$$

A partir da equação (2.47) podemos definir tanto o fluxo convectivo como o fluxo imposto apenas alterando-se os valores de h e C . Assim:

Para fluxo imposto:

$$h = 0, \quad C = q \quad (2.48.a)$$

Para parede adiabática:

$$h = 0, \quad C = 0 \Rightarrow \Gamma \frac{\partial \phi}{\partial n} = 0 \quad (2.48.b)$$

Para fronteira convectiva:

$$h = h, \quad C = h \phi_a \quad (2.48.c)$$

Temperatura imposta:

$$\phi = \phi_i \quad (2.48.d)$$

A implementação das condições de fronteira é feita no programa FEMEV, da seguinte forma:

$$Q_f^d = \int_f \Gamma \frac{\partial \phi}{\partial n} ds = \int_f (-h\phi + C) ds \quad (2.49)$$

$$Q_f^d = \Gamma \sum_{j=1}^4 \left(- \int (h N_j ds) \phi_i \right) + \int_f C ds \quad (2.50)$$

ou

$$Q_f^d = \Gamma \sum_{j=1}^4 \left(- (h N_j \Delta S_i) \phi_i \right) + C \Delta S_i \quad (2.51)$$

onde N_j são as funções de interpolação, ΔS_i é indicado na figura 2.14, e C e h assumem valores que dependem do tipo de fluxo imposto na fronteira.

A seção 2.10 encerra o desenvolvimento teórico do método dos elementos finitos, baseado em volumes de controle. Passaremos agora a implementação computacional do método.

CAPÍTULO 3

IMPLEMENTAÇÃO NÚMERICA DO PROGRAMA FEMEVA

Nesse capítulo será descrito o processo de implementação numérica da teoria exposta nos capítulos anteriores. Para isso descreveremos as rotinas que compõem o programa FEMEVA e quando necessário, forneceremos maiores informações a respeito da implementação da rotina em si.

3.1- Rotinas que Compõem o Programa FEMEVA

A fim de implementar a teoria desenvolvida no capítulo anterior foi desenvolvido o programa FEMEVA. O programa foi desenvolvido em linguagem Fortran-77, numa estação de trabalho Sparc-Sun. O programa não se encontra otimizado em relação ao tempo de CPU. Na verdade em muitas fases da implementação optou-se por um maior tempo de CPU em função de uma melhor interface com o usuário.

O fluxograma da figura 3.1 mostra as rotinas que compõem o programa principal. Descreveremos a seguir, a função de cada uma dessas rotinas.

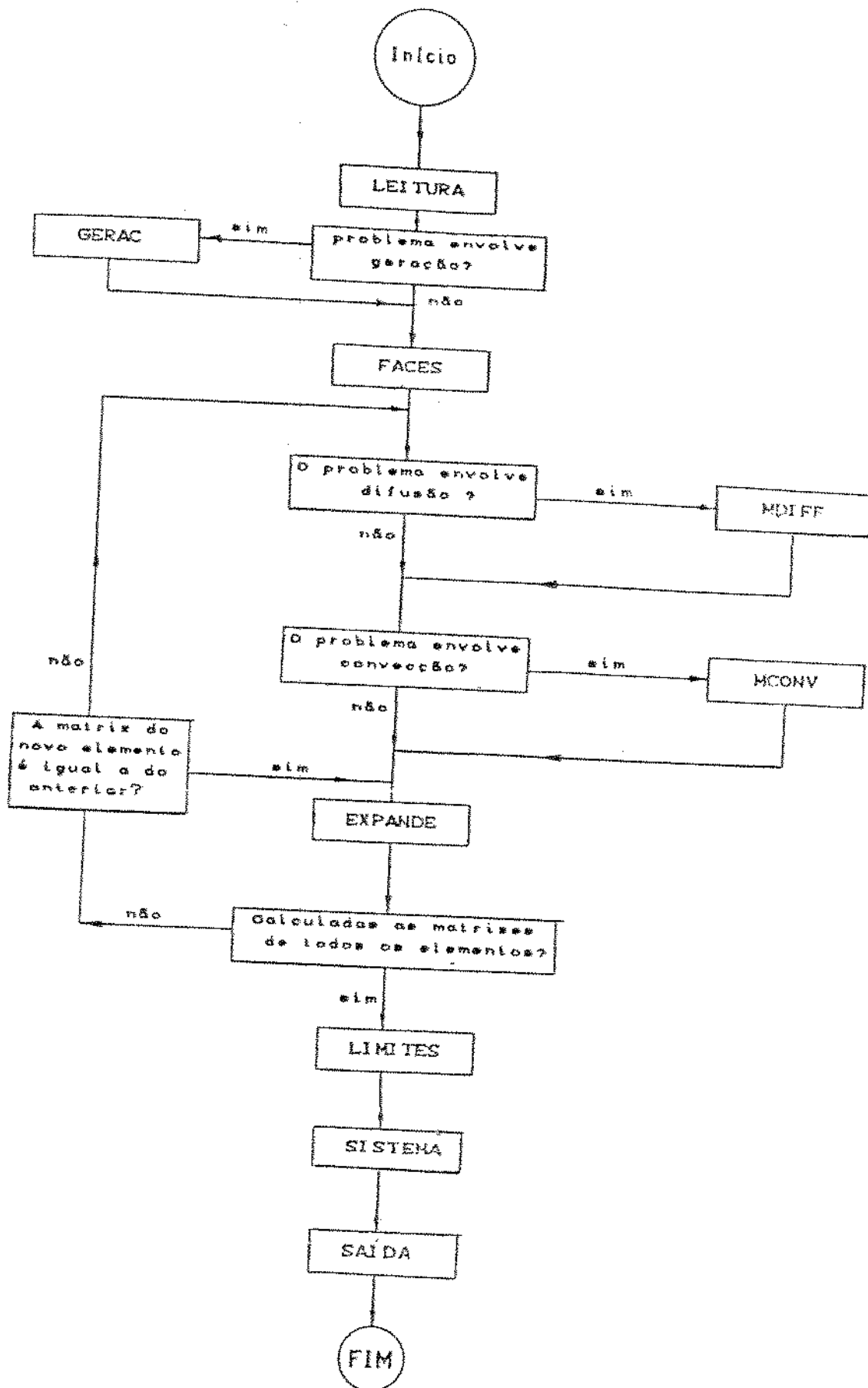


Figura 3.1- Fluxograma do programa principal

3.1.1- MDIFF

A rotina MDIFF realiza os cálculos relacionados aos termos de difusão, implementando a equação (2.21). Ela é composta da rotina JACOB4. A rotina JACOB4 calcula os valores das derivadas $\partial N_j / \partial x$ e $\partial N_j / \partial y$. As fases da rotina MDIFF são as seguintes:

- a) calcula os parâmetros geométricos necessários para a definição dos segmentos SSI
- b) monta a equação para o volume de controle (equação (2.21))

A implementação da equação (2.21) é imediata. Inicialmente deve ser definido um sentido de rotação, no caso o anti-horário. São feitos, então, os cálculos dos comprimentos Δx e Δy dos segmentos SSI. Aplicando a equação (2.21) são calculados os fluxos atravessando os segmentos SSI. Deve-se notar que o fluxo calculado, por exemplo, para o segmento SSI do subvolume de controle 1 (figura 3.2) terá o mesmo módulo, mas sinal contrário do fluxo que atravessa o segmento SSI do subvolume de controle 2. Dessa forma ele não necessita ser calculado novamente. Basta trocar o sinal do fluxo já calculado para o segmento SSI do subvolume de controle 1.

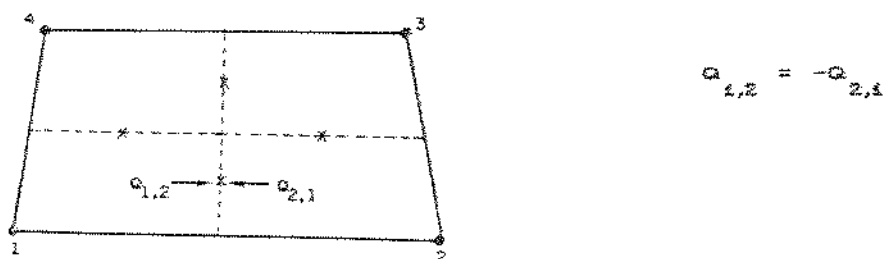


Figura 3.2 - Cálculo de fluxos em volumes de controle adjacentes

3.1.2- MCONV

Esta rotina realiza os cálculos relacionados aos termos de convecção e controla a montagem dos termos convectivos. Esta rotina tem as seguintes fases:

a) calcula os parâmetros geométricos necessários para a definição dos segmentos SS_i e dos pontos de integração.

b) calcula as matrizes dos termos difusivo, convectivo e de geração para os pontos de integração.

c) relaciona os pontos de integração aos valores nodais (equação (2.38))

d) calcula as matrizes para os volumes de controle (equação (2.39))

O diagrama de blocos da subrotina MCONV é fornecido na figura 3.3. A seguir descreve-se as rotinas que compõe MCONV.

3.1.2.1- Rotina IPCONV

Esta rotina realiza os cálculos relacionados aos pontos de integração para o operador convectivo (equação (2.32)). Um diagrama de blocos da rotina IPCONV é fornecido na figura 3.4. Como visto no capítulo anterior cada elemento é subdividido em quatro subvolumes de controle e quatro pontos de integração. Para cada ponto de integração será necessário calcular o ponto onde o vetor velocidade intercepta a fronteira. O procedimento para o cálculo do ponto de intercessão é descrito a seguir .

Considere a figura 3.5. Nessa figura são mostradas todos os possíveis segmentos onde poderão se localizar os pontos de intercessão. Esses segmentos são representados pelas 10 retas mostradas nessa figura. Para cada ponto de integração ip_i haverá

seis segmentos com possibilidades de serem interceptados pelo vetor velocidade que tem origem nesse ponto ip1. A figura 3.6 mostra os 6 possíveis segmentos para o ponto ip1. O procedimento para determinar o ponto de intercessão é descrito a seguir.

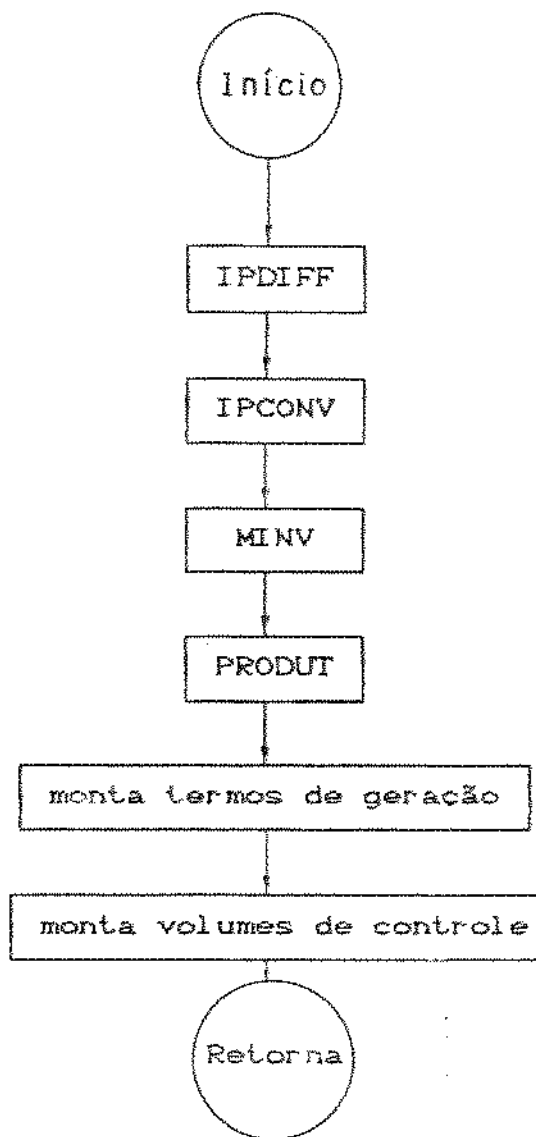


Figura 3.3 - Rotina MCONV

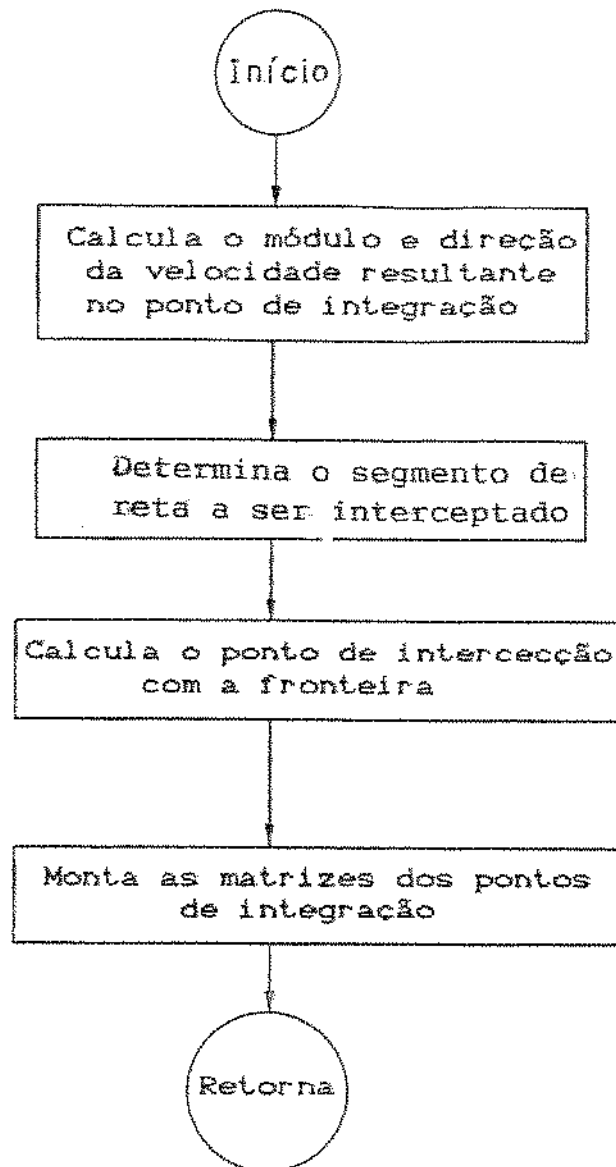


Figura 3.4- Rotina IPCONV

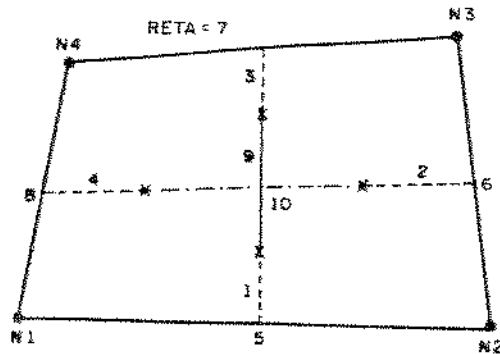


Figura 3.5- Definição das retas de intercessão

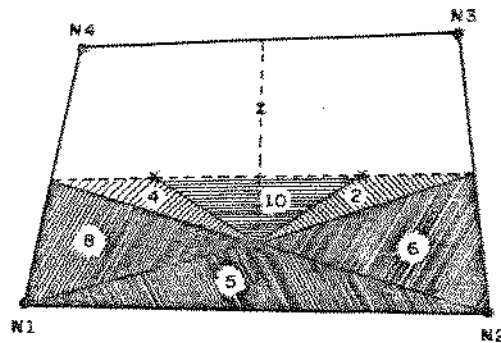


Figura 3.6- Segmentos de retas com possibilidades de serem interceptados pelo vetor velocidade com origem em ip_1

Inicialmente é calculado o ângulo que o vetor velocidade partindo de ip_1 faz em relação ao eixo x . Então é verificado qual segmento será interceptado por esse vetor velocidade. A figura 3.6 mostra todos os segmentos de reta com possibilidades de serem interceptados pelo vetor velocidade partindo de ip_1 . Para um vetor velocidade partindo, por exemplo, de ip_2 outros segmentos terão possibilidades de serem interceptados. No caso de ip_2 , os possíveis segmentos serão os segmentos das retas 6, 7, 3, 9, 1, 5.

Para o cálculo da intercessão, o vetor velocidade e a reta a ser interceptada serão representados por sua equações

paramétricas. Seja uma reta genérica Γ formada pela união de dois pontos (x_1, y_1) e (x_2, y_2) . A representação paramétrica dessa reta será da forma:

$$\begin{aligned} x &= x_1 + (x_2 - x_1)t \\ y &= y_1 + (y_2 - y_1)t \end{aligned} \quad (3.1)$$

Para o vetor velocidade a representação será similar:

$$\begin{aligned} x &= x_3 + (u - x_3)t \\ y &= y_3 + (v - y_3)t \end{aligned} \quad (3.2)$$

onde u e v referem-se ao campo de velocidades

(x_3, y_3) é a coordenada do ponto de integração ip_i

Resolvendo-se o sistema formado pelas equações (3.1) e (3.2) obtém-se os valores dos pontos de interseção com as fronteiras. A partir desses pontos são calculadas as relações da equação (2.33), e calculadas as matrizes dos pontos de integração.

3.1.2.2- Rotina IPDIFF

A rotina IPDIFF faz a montagem dos termos difusivos para os pontos de integração (eq. (2.28)). Ela é composta da rotina JACOB4. A rotina JACOB4 é quem realiza os cálculos das funções de interpolação para os pontos de integração.

3.1.2.3- Rotina MINV

Realiza a inversão de uma matriz 4x4. A matriz a ser invertida corresponde ao termo $[c^c - c^d]$ da equação (2.36).

3.1.2.4- Rotina PRODUT

Realiza o produto da matriz inversa pelas matrizes

coeficiente dos termos ϕ e d_i (eq. 2.37).

3.1.3- Rotina FACES

Nessa rotina são determinados os nós que compõe as fronteiras exteriores do domínio. Essas fronteiras são utilizadas em casos onde haja fluxos convectivos atravessando as fronteiras, sem que haja condições impostas, ou em problemas de condução onde haja convecção ou fluxo de calor impostos nas fronteiras externas. Nas situações de condução a única função do cálculo da subrotina FACES é verificar se o fluxo está realmente sendo imposto numa fronteira externa e avisar o usuário de um possível erro na entrada de dados. Essa rotina também determina quais volumes de controle estão situados em cantos do domínio. Esses volumes são utilizados, posteriormente, pela rotina LIMITES.

Para a determinação das faces exteriores é adotado o seguinte esquema:

a) Inicialmente é verificado quais são os nós que pertencem a tres ou menos elementos. Os segmentos de reta unindo estes nós serão possíveis fronteiras do domínio

b) É verificado, então, se entre os nós escolhidos no item (a) não existem nós que correspondem a segmentos internos (figura 3.7)

c) Os valores dos nós, assim determinados, são armazenados em disco para futura utilização.

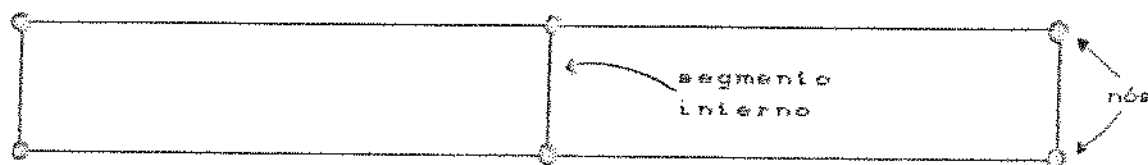


Figura 3.7 - Determinação dos segmentos de fronteira

Deve-se notar que essa rotina significa uma facilidade para o usuário mas não é indispensável. Em casos onde a principal exigência fosse um menor tempo de CPU, esta rotina seria dispensável. Bastaria modificar-se o programa e entrar diretamente com os segmentos de fronteira.

3.1.4- Rotina EXPANDE

Essa rotina é responsável pela expansão da matriz local (4x4), para a matriz global. Como já explicado, a cada linha e coluna da matriz local corresponderá um subvolume de controle e uma variável nodal respectivamente. Bastará, então, substituir o valor de cada coeficiente da matriz local na respectiva posição da matriz global, correspondente a esse subvolume de controle e variável nodal.

3.1.5- Rotina LIMITES

Essa rotina faz a montagem das condições de fronteira. Como já visto no capítulo anterior as condições de fronteira poderão ser de dois tipos: convectivas ou difusivas. Em casos de fronteiras convectivas a rotina FACES determinará quais segmentos compõe as fronteiras externas. Nesses pontos nodais será calculado o fluxo mássico que atravessa as fronteiras. Será então constituída a equação (2.43). Em todos os cálculos são utilizados apenas interpolações nodais. Um tratamento especial é dedicado aos volumes de controle situados em nós de cantos sem que haja uma condição imposta neste nó (figura 3.8). Neste tipo de volume de controle apenas dois segmentos SSI contribuirão para o cálculo dos fluxos internos, enquanto que nos outros volumes de controle

haverá quatro segmentos SSi. Ao elaborar-se as equações de volume de controle para as faces N1N2 e N1N5 deve-se ter em mente que as contribuições dessas faces necessitarão serem calculadas apenas uma vez. Para isso a rotina FACES detecta quais nós são nós de canto, de modo que a correta contribuição seja calculada nesses volumes de controle.

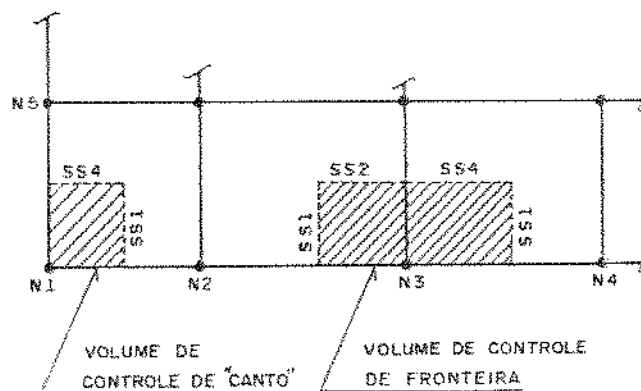


Figura 3.8 - Volumes de controle situados em faces de canto

Os termos difusivos são montados segundo a equação (2.46). Os valores de \underline{C} e \underline{h} dessa equação são determinados em função do tipo de condição imposta. Como pode ser visto na equação (2.46), os termos correspondentes a convecção são constituídos diretamente na matriz dos coeficientes de ϕ , evitando-se assim, uma solução iterativa para o sistema.

3.1.6- Rotina GERAC

A rotina GERAC tem por função executar os cálculos do termo fonte da equação (2.23). Para a integração do termo fonte é utilizada uma quadratura de Gauss de ordem 4. Como cada elemento é composto de quatro subvolumes de controle a integração é feita por quadrantes separadamente (figura 3.9).

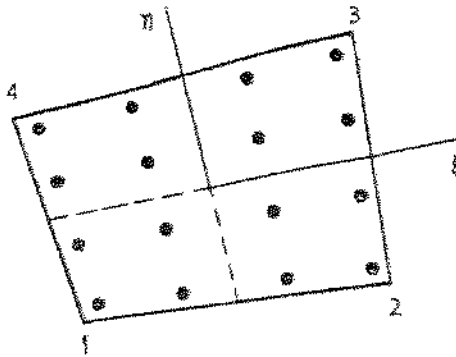


Figura 3.9 - Quadrantes de integração para o termo fonte

3.1.7- Rotina SISTEMA

Essa rotina realiza a resolução do sistema de equações algébricas. É utilizado o método de Gauss com pivotamento parcial. O sistema de equações está limitado a cerca de 900 equações (workstation SPARC-SUN), já que não foi implementado nenhum processo especial de alocação de memória como um esquema Skyline ou método frontal. A razão disso é não ser o objetivo desse trabalho gerar-se um programa de finalidades comerciais, capaz de manipular uma grande quantidade de nós, mas sim testar-se a precisão do método em si. Acreditamos que sistemas com 900 graus de liberdade sejam suficientes para isso. Em um microcomputador do tipo PC esse limite seria de cerca de 150 nós. Deve-se frisar que esses limites poderão ser aumentados caso se utilize um esquema apropriado que armazene a matriz na sua forma de banda.

3.2- Descrição do arquivo de entrada de dados

O programa foi elaborado de modo a ter sua entrada de dados feita através da leitura de um arquivo contendo informações sobre a geometria, conectividade entre os elementos, condições de

contorno, tipo de solução, etc. Os dados de entrada deverão estar contidos num arquivo chamado FEMEVENT. A saída dos resultados será feita no arquivo FEMEVSAL. O arquivo de dados será composto por caracteres numéricos e alfanuméricos. As oito primeiras colunas de uma linha de dados são reservadas para os caracteres alfanuméricos, necessariamente maiúsculos. Estes caracteres alfanuméricos são palavras que definirão o tipo de solução, as condições de fronteira, a conectividade dos elementos, etc.

O formato dos dados de entrada deverá respeitar a seguinte disposição:

1-8	9-80

Cada linha será composta por oitenta colunas. As oito primeiras colunas são reservadas para os caracteres alfanuméricos. As colunas restantes serão destinadas aos dados numéricos que poderão estar em formato livre. Essas linhas de entrada de dados serão chamadas de cartões. A seguir faremos uma descrição de todos os cartões de entrada do programa.

Obs: Nos dados de entrada abaixo foi adotado o Sistema Internacional. No entanto, pode-se utilizar qualquer sistema de unidades, desde que se mantenha a coerência nas unidades.

SOL	n1	prec	niter	fad
-----	----	------	-------	-----

n1: número inteiro (1 -> PROBLEMA CONVECCÃO-DIFUSÃO
(2 -> CONDUÇÃO

prec: O valor de prec define o valor da diferença nodalmáxima entre duas iterações , em casos de condutividade variável com a temperatura

niter: número maximo du iterações. Em problemas que não envolvam condutividade variável com a temperatura esse valor pode ser estabelecido como 1.

n2: número inteid elementos com dimensões geométricas iguais. A utilização desse recurso permite que se economize tempo de processamento, pois assim evita-se o cálculo da matriz local em elementos semelhantes

A estrutura do arquivo deve ser a seguinte:

```
+elem1
-elem2
-elem3
+elem4
-elem5
```

Nessa estrutura sempre que o elemento for positivo será calculada a matriz local. Se a esse elemento, seguir-se um elemento negativo será utilizada a matriz do elemento anterior. Assim na estrutura acima é calculada a matriz local para elem1 (elemento positivo). Para os elementos elem2, elem3 é utilizada a mesma

matriz (sinal negativo). Já para elem4 é calculada uma nova matriz que será a mesma utilizada no elemento elem5. Todos elementos que compõe a malha necessitam aparecer nesse arquivo mesmo que suas matrizes não sejam repetidas, do contrário não terão suas matrizes calculadas. Esse cartão deve ser o primeiro cartão do arquivo de dados.

CQUAD4	elem	no1	no2	no3	no4
--------	------	-----	-----	-----	-----

elem: número do elemento

no1,no2,no3,no4: nós que compõe o elemento.

Este cartão define a conectividade de um elemento. A figura 3.6 mostra a convenção adotada. A numeração dos nós deve começar no canto esquerdo inferior e seguir o sentido antihorário. Na figura também é mostrada a convenção adotada para a numeração das faces de um elemento.

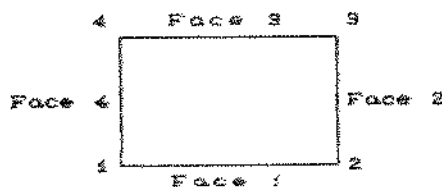


Figura 3.10 - Convenção utilizada para a numeração dos nós

GRID	no	x	y
------	----	---	---

no: número do nó

x, y: coordenadas x, y do nó (m)

CFCONV	elem	face	h	tg
--------	------	------	---	----

elem: número do elemento

face: número da face sujeita a convecção

h: coeficiente de convecção ($W/m^2 K$)

tg: temperatura do ambiente (K)

Esse cartão define convecção imposta. A convenção adotada na numeração das faces é a mesma fornecida no cartão COUAD4

CFQ	elem	face	q
-----	------	------	---

elem: número do elemento

face: número da face sujeita a convecção

q: fluxo de calor imposto (W/m^2)

Este cartão especifica um fluxo de calor imposto em faces de fronteira. A convenção adotada na numeração das faces é a mesma fornecida no cartão COUAD4

CVX	no	velocx
-----	----	--------

no: número do nó
 velocx: velocidade imposta na direção x (m/s)

CVY	no	velocy
-----	----	--------

no: número do nó
 velocy: velocidade imposta na direção y (m/s)

CTEMP	no	temp
-------	----	------

no: número do nó
 temp: temperatura imposta neste nó (K)

		?
--	--	---

no: número do nó
 temp: temperatura inicial neste nó (K)

Este dado de entrada só terá significado em problemas onde a condutividade dependa da temperatura. Nestas situações quanto mais próxima for a solução inicial da solução do problema mais rápida será a convergência.

GERACAO	elem	q
---------	------	---

elem: número do nó

q: calor gerado por unidade de volume (W/m^3)

Este cartão especifica um elemento com geração internade calor

PROP	cond	beta
------	------	------

Os valores de cond e beta definem a variação da condutividade com a temperatura segundo o seguinte expressão:

$$\Gamma = \text{cond} * (1 + \text{beta} * T)$$

cond : condutividade ($W/m.K$)

T: temperatura (K)

beta: coeficiente de condutividade térmica ($1/K$)

CFCORR	α
--------	----------

Este cartão modifica o tipo de ponderação utilizada para a interpolação dos termos convectivos. Este recurso é útil em problemas onde as linhas de corrente não sejam paralelas no interior de um elemento. Caso a discretização seja suficiente, haverá uma faixa de valores de α para os quais a solução será praticamente a mesma, independente do valor de α . Em casos de linhas de corrente paralelas, este cartão será desnecessário.

sendo $\underline{\alpha}$, automaticamente assumido como sendo igual ao coeficiente de difusão do problema. Quanto mais difusivo for o problema menor influência terá o valor de $\underline{\alpha}$. No apêndice A podem ser encontrados maiores detalhes a respeito do coeficiente $\underline{\alpha}$.

Nota: Este cartão deve ser colocado necessariamente após o cartão PROP

CDENS	dens	cp
-------	------	----

dens: densidade (kg/ m³)
 cp: calor específico a pressão constante
 (J/s) (1/Kg.K)

Em problemas já adimensionalizados, o valor de cp deve ser estabelecido como 1.

CFIP	n1
------	----

n1: número inteiro (0 - interpolação linear (default)
 (1 - interpolação por pontos de
 integração

Esse cartão determina o tipo de interpolação usada nos volumes de fronteira. Para interpolações lineares não é necessário definir-se esse cartão.

*

Cartão usado para comentários no arquivo de entrada. Os comentários devem seguir após o asterisco.

CAPÍTULO 4

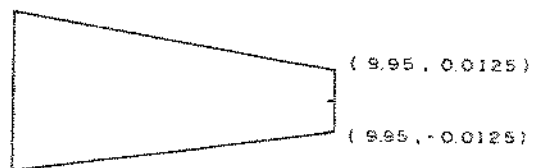
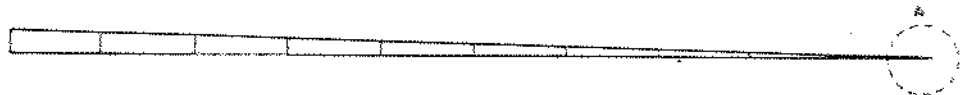
RESULTADOS

Nesse capítulo serão descritos os resultados da aplicação do programa FEMEV, em casos que envolvam condução e convecção .

4.1- CONDUÇÃO DE CALOR UNIDIMENSIONAL NUMA ALETA TRIANGULAR

Este problema tem a função de verificar o comportamento do programa em casos de condução de calor, em malhas de geometria não regular e sujeitos a transferência de calor para o meio exterior por convecção.

$$\begin{aligned}h &= 1.6 \text{ BTU/hr-ft}^2\text{-F} \\ \Gamma &= 192 \text{ BTU/hr-ft-F} \\ T_0 &= 250 \text{ F} \\ T_a &= 70.1 \text{ F} \\ L &= 10 \text{ ft} \\ \delta_1 &= 0.25 \text{ ft}\end{aligned}$$



DETALHE A

Figura 4.1 - Discretização da aleta triangular (malha 1x10)

A solução analítica desse problema é dada pela expressão⁴⁵:

$$T = \frac{T_0 \cdot J_0(2.1N\sqrt{Lx})}{J_0(2.1NL)} \left[T_0 - T_a \right] + T_a \quad (4.1)$$

onde:

J_0	função de Bessel de ordem zero
N	$\sqrt{h/(\Gamma\delta_t)}$
h	coeficiente de transferência de calor por conveccção
Γ	condutividade térmica
L	comprimento da aleta
T_a	temperatura ambiente
T_0	temperatura na base da aleta
δ_t	largura da base da aleta

O problema foi modelado em elementos finitos através de uma malha 1x10 elementos (Fig. 4.1) e através de uma malha 1x5, para comparar-se a precisão obtida em função do número de elementos. Em ambos os casos a malha é regularmente espaçada, exceto os dois últimos nós que tem por coordenadas x , o valor $x=9,95$ (detalhe na figura 4.1). Os resultados são mostrados na figura 4.2 para a malha 1x5.

A tabela 4.1 fornece os dois pontos do gráfico da figura 4.2 com as maiores percentagens de erro e os compara com os resultados obtidos usando-se uma malha 1x10 elementos.

Tabela 4.1 - Comparação dos resultados

Num. de elementos	Posição (x/L)	Teórico	Calculado	Erro(%)
05	0.5	130.07260	129.27104	0.675
10	0.5		129.95936	0.066
05	0.8	95.09164	94.44959	0.616
10	0.8		95.02923	0.087

O tempo total de CPU, na estação de trabalho SPARC-SUN, foi de 0.3 segundos para 5 elementos e de 0.8 segundos para 10 elementos. Observa-se que para casos de condução unidimensional de calor um pequeno número de elementos é suficiente para gerar resultados de boa precisão. Como esperado, aumentando-se o número de elementos, a precisão aumenta.

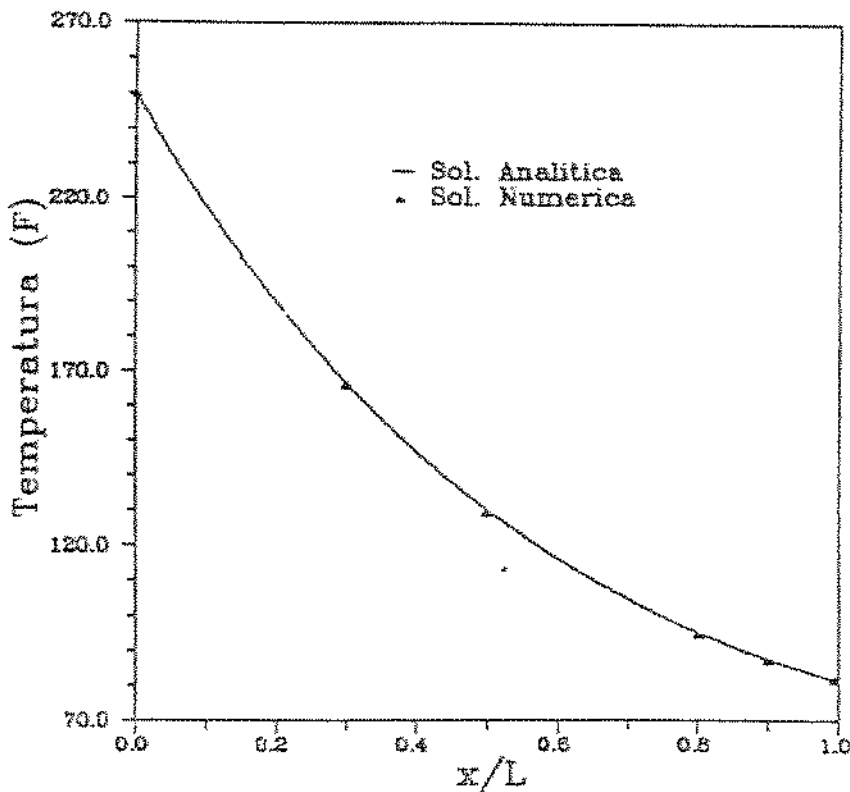
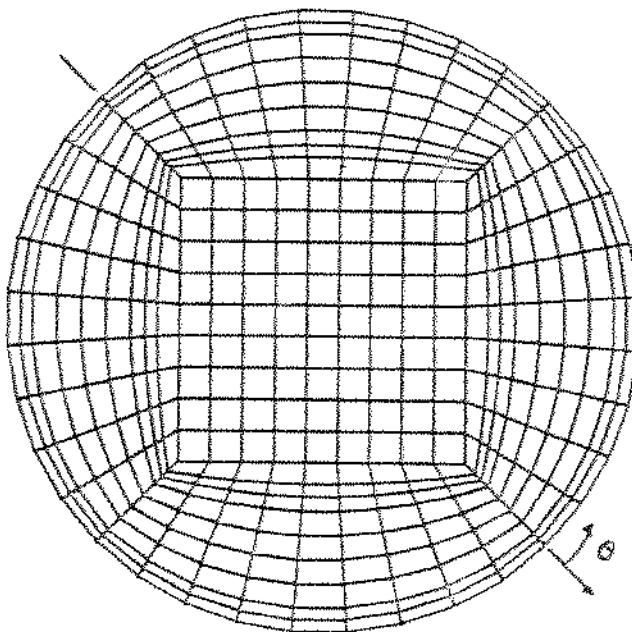


Figura 4.2 - Comparação dos resultados para a aleta triangular (malha 1x5)

4.2 - CONDUÇÃO DE CALOR BIDIMENSIONAL NUM CILINDRO
SUJEITO À CONVECÇÃO E FLUXO IMPOSTO

A figura 4.3, ilustra o problema e mostra a malha usada em sua discretização. Foram utilizados 424 nós e 375 elementos. O problema equivale a um cilindro exposto a radiação em metade de sua face e sujeito a convecção constante em seu perímetro. A solução teórica, para o caso, é fornecida pela expressão⁴⁶:

$$\frac{T(r, \theta)}{s_0/h} = \frac{1}{\pi} + \frac{(r/R)}{2(1 + \Gamma/hR)} \sin \theta - \frac{2}{\pi} \sum_{m=1}^{\infty} \frac{(r/R)^{2m}}{(4m^2 - 1)(1 + 2m\Gamma/hR)} \cos 2m\theta \quad (4.2)$$



$$s_0 = \begin{cases} 2000 \sin \theta & 0 \leq \theta \leq \pi \\ 0 & \pi \leq \theta \leq 2\pi \end{cases}$$

$$s_0 \text{ (BTU/hr-ft}^2\text{)}$$

$$\Gamma = 10 \text{ BTU/hr-ft-F}$$

$$h = 2.51 \text{ BTU/hr-ft}^2\text{F}$$

$$R = 3 \text{ ft}$$

Figura 4.3 - Condução bidimensional num disco

A figura 4.4 ilustra as linhas de isotemperatura para a solução numérica (coincidentes com a solução analítica). As linhas de isotemperatura estão traçadas apenas para o quadrado situado dentro do cilindro por não dispor-se de um programa para isotemperaturas, capaz de manipular geometrias não retangulares. O tempo de CPU total para a resolução do programa é de 127.9s. No apêndice A são fornecidos alguns valores calculados.

O histograma apresentado na figura 4.5 relaciona a frequência do erro em função de intervalos de 0.1 para ϵ . O valor de ϵ é dado pela expressão:

$$\epsilon = \frac{(\text{valor teórico} - \text{valor numérico})}{(\text{valor teórico})} * 100 \quad (4.3)$$

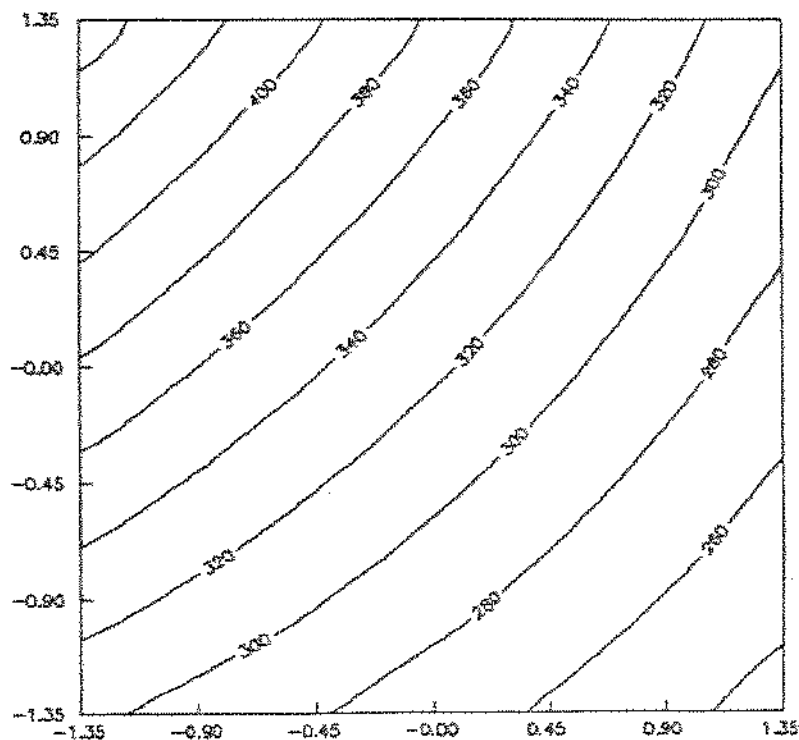


Figura 4.4 - Linhas isotérmicas no interior do disco

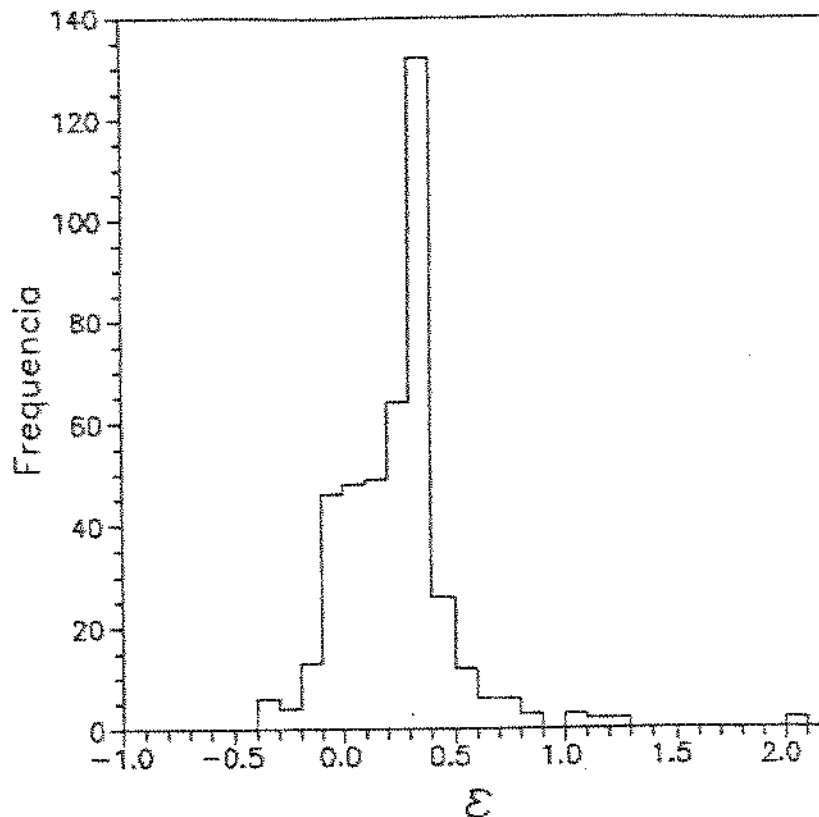


Figura 4.5 - Histograma do erro para a condução de calor em um disco

Na figura 4.5 nota-se que o histograma de erros é bastante compacto, indicando que após atingida uma determinada precisão na solução, as percentagens de erro estarão compreendidas numa faixa bem definida.

4.3- CONDUÇÃO DE CALOR NUM RETÂNGULO SUJEITO À GERAÇÃO INTERNA DE CALOR

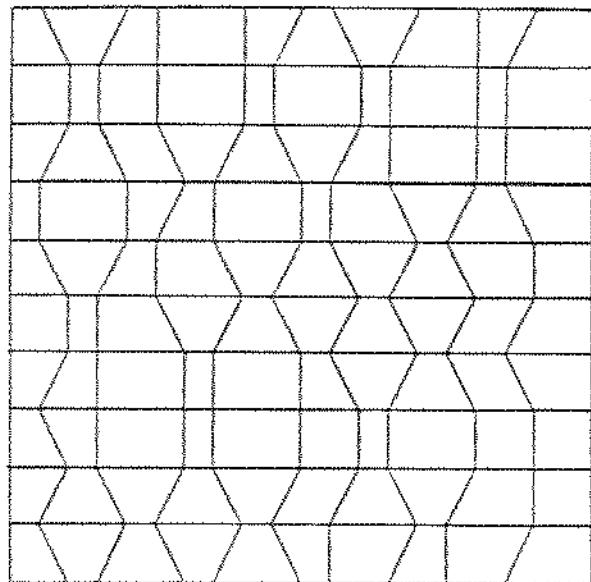
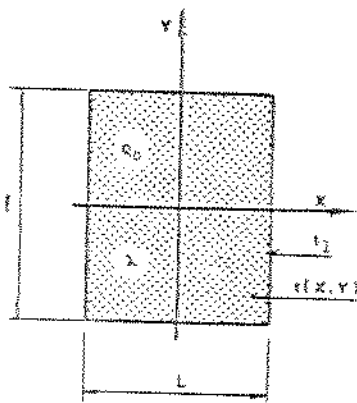
O terceiro problema a ser testado é um problema envolvendo condução bidimensional de calor com geração interna. É utilizada uma malha irregular (Fig. 4.6) como forma de verificar-se o comportamento do programa a domínios irregulares. A

solução analítica desse problema é fornecida pela seguinte expressão⁴⁵:

$$\frac{T - t_s}{s_o} = \frac{16}{\Gamma L} \sum_{i=1}^{\infty} \sum_{j=1}^{\infty} \frac{(-1)^{i+j}}{\mu_i \eta_j (\mu_i^2 + \eta_j^2)} \cos(\mu_i x) \cos(\eta_j y) \quad (4.4)$$

onde:

$$\left. \begin{aligned} \mu_i &= \frac{\pi}{L} (2i - 1) \\ \eta_j &= \frac{\pi}{L} (2j - 1) \end{aligned} \right\} i = j = 1, 2, 3, \dots$$



$$\begin{aligned} s_o &= 200 \text{ BTU/hr-ft}^3 \\ \Gamma &= 10 \text{ BTU/hr-ft-F} \\ L = l &= 20 \text{ ft} \\ t_s &= 0 \end{aligned}$$

Figura 4.5 - Condução num quadrado sujeito à geração interna

A malha utilizada na discretização é de 10×10 elementos, com 121 nós, para um quadrante do domínio. A figura 4.7 ilustra as isotemperaturas para a solução numérica (idêntica as da solução analítica) e a figura 4.8, o histograma de erro para a solução encontrada.

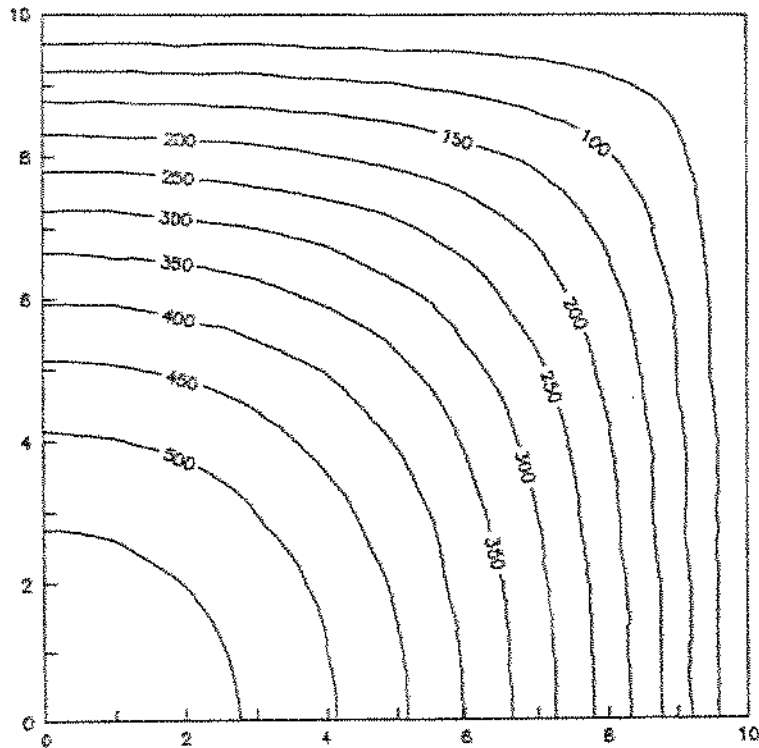


Figura 4.7 - Isotérmicas para um quadrado sujeito à geração interna de calor

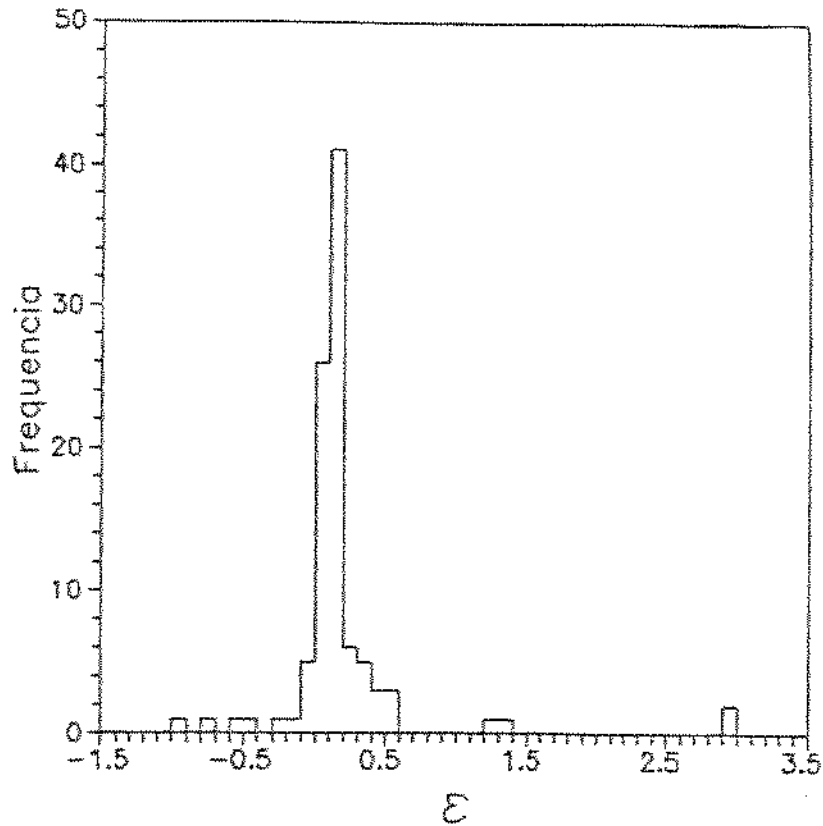


Figura 4.8 - Histograma do erro para a condução num quadrado sujeito geração interna de calor

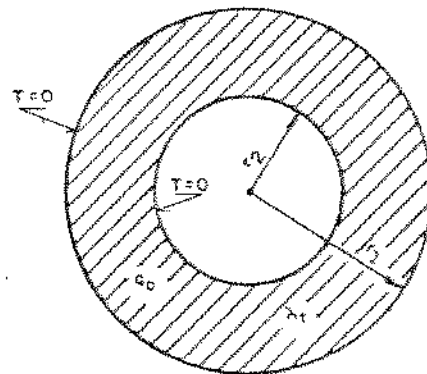
O tempo de CPU é de 6s. A tabela 4.2 ilustra alguns resultados do programa para uma coordenada x constante ($x=5$) e os compara com os resultados analíticos.

Tabela 4.2 - Resultados para um quadrado sujeito à geração interna de calor

Coordenada (y)	Solução Numérica	Solução Analítica	Erro (%)
0.0	459.244	458.679	-0.123
2.0	444.263	443.963	-0.063
5.5	341.123	340.622	-0.147
8.5	147.959	147.776	-0.130

4.4- CONDUÇÃO DE CALOR NUM CILINDRO OCO SUJEITO À
GERAÇÃO INTERNA DE CALOR E CONDUTIVIDADE TÉRMICA
DEPENDENTE DA TEMPERATURA

A figura 4.9 ilustra o problema:



$$s_0 = 1 \text{ ES BTU/hr-ft}^3$$

$$r_1 = \frac{1}{4} \text{ in}$$

$$r_1/r_2 = 4$$

$$\Gamma_0 = 10 \text{ BTU/hr-ft-F}$$

Figura 4.9 - Cilindro oco com condutividade dependente da temperatura.

A solução analítica desse problema é dado pela seguinte expressão⁴⁵:

$$T = \frac{1}{\beta_0} \left[\sqrt{1 + 2\beta_0\alpha} - 1 \right] \quad (4.5)$$

onde:

$$\alpha = \frac{s_0}{4\Gamma_0} \left[(r_1^2 - r_2^2) - \frac{(r_1^2 - r_2^2)}{\ln(r_1/r_2)} \ln(r_1/r_2) \right]$$

s_0 = geração interna

e a dependência da condutividade com a temperatura (T) expressa da seguinte forma:

$$\Gamma_1 = \Gamma_0(1 + \beta_0 T)$$

Γ_o = condutividade térmica

β_o = coeficiente de condutividade térmica

A figura 4.10 ilustra a solução analítica e a solução numérica para uma malha de 1×20 elementos. O ângulo do setor circular utilizado é de $\pi/180$ rad.

A tabela 4.3 compara os resultados da figura 4.10 para $\beta_o = -0.0075$, com os piores resultados encontrados numa discretização 1×5 elementos para um mesmo ângulo do setor circular.

Tabela 4.3 - Comparação de resultados para um cilindro oco

Num. de elementos	Posição (r/r2)	Teórico	Calculado	Erro(%)
20	2.2	68.499	68.434	0.095
5	2.2		57.620	1.280
20	2.8	60.302	60.253	0.065
5	2.8		59.782	0.862

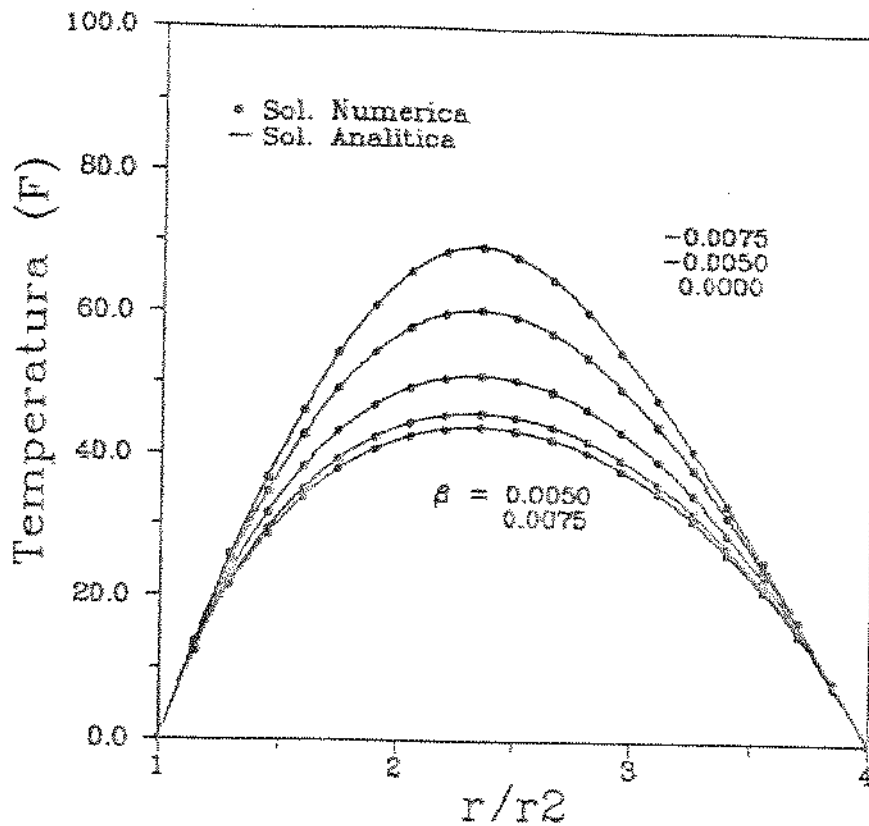


Figura 4.10 - Comparação dos resultados obtidos para um cilindro oco (malha 1x20 elementos)

Observa-se da tabela 4.3 que, mesmo para uma malha 1 x 5, a solução encontrada também possui uma boa precisão. Na tabela 4.4 é fornecido o número de iterações para a convergência a partir da solução nula. O critério de convergência adotado foi o de uma diferença máxima nos valores das temperaturas entre duas iterações de 1E-04.

Tabela 4.4 - Número de iterações e tempo de CPU para condução em um cilindro oco

elementos	β_0	iterações	tempo de CPU
5	0.0050	7	0.3
20		6	1.4
5	-0.0050	8	0.4
20		7	1.5
5	-0.0075	10	0.4
20		9	1.8
5	0.0075	7	0.3
20		7	1.6

4.5- CONDUÇÃO RADIAL DE CALOR EM UM CILINDRO GIRANTE

O problema é esquematicamente ilustrado na figura 4.12, onde r_1 é tomado como 1. No sistema de coordenadas cilíndricas (r, θ) o problema é na verdade, um problema de condução unidimensional com solução analítica. Passando-se para sistema de coordenadas cartesianas, pode-se transformar este problema de condução unidimensional num problema de convecção-difusão.

Introduzindo variáveis adimensionais da forma:

$$\phi^* = \frac{\phi - \phi_2}{\phi_1 - \phi_2}$$

$$x^* = \frac{x}{r_1};$$

$$y^* = \frac{y}{r_1}$$

$$u^* = \frac{u}{\omega r_1};$$

$$v^* = \frac{v}{\omega r_1}$$

$$\Gamma^* = 1/Pe(x^{*2} + y^{*2})$$

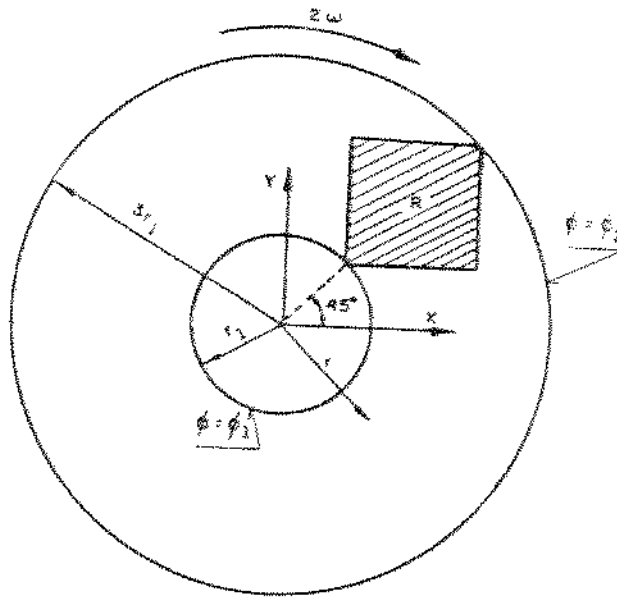


Figura 4.11- Cilindro girante

A formulação do problema será da forma:

$$u^* \frac{\partial \phi^*}{\partial x^*} + v^* \frac{\partial \phi^*}{\partial y^*} = \left[\frac{\partial^2 \Gamma^* \phi^*}{\partial x^{*2}} + \frac{\partial^2 \Gamma^* \phi^*}{\partial y^{*2}} \right] \quad (4.6)$$

Fixando-se o campo de velocidades, entre os cilindros, como sendo corresponde à uma rotação de corpo rígido, tem-se que o campo de velocidades será expresso por:

$$\begin{aligned} u^* &= 2y^* \\ v^* &= -2x^* \end{aligned}$$

As condições de fronteira são fixadas como:

$$\phi^* = \begin{cases} 1 & r = 1 \\ 0 & r = 3 \end{cases}$$

E a solução analítica dada pela expressão:²⁰

$$\phi^* = \frac{\theta - (x^{*2} + y^{*2})}{2} \quad (4.7)$$

O problema FEMEV, foi utilizado para resolver esse problema na região R da fig. 4.11. Os valores de ϕ^* nas quatro fronteiras de R foram especificados a partir da solução exata.

Uma malha regular de 10x10 elementos foi utilizada para discretizar a região R. Os resultados são mostrados na Fig.4.13, focalizando a percentagem de erro no centro da malha ($x^* = \sqrt{2}$, $y^* = \sqrt{2}$). A percentagem de erro é calculada pela expressão 4.3.

Os resultados foram comparados com esquemas para diferenças finitas²⁰ (MDF) e mostraram-se mais precisos. A melhor precisão da solução pode ser considerada como devida aos menores índices de falsa difusão, que o método aqui empregado apresenta.

Como o programa não se encontra otimizado fornecemos na tabela 4.5 os tempos de cada fase do programa, para efeito de possíveis comparações com outros métodos.

Tabela 4.5 - Tempo de CPU por fases para a condução num cilindro girante

	Tempo de CPU(s)	Perc. do tempo total(%)
Leitura dos dados	1.3	18.3
Montagem do sistema	3.7	52.1
Resolução do sistema	2.1	29.6

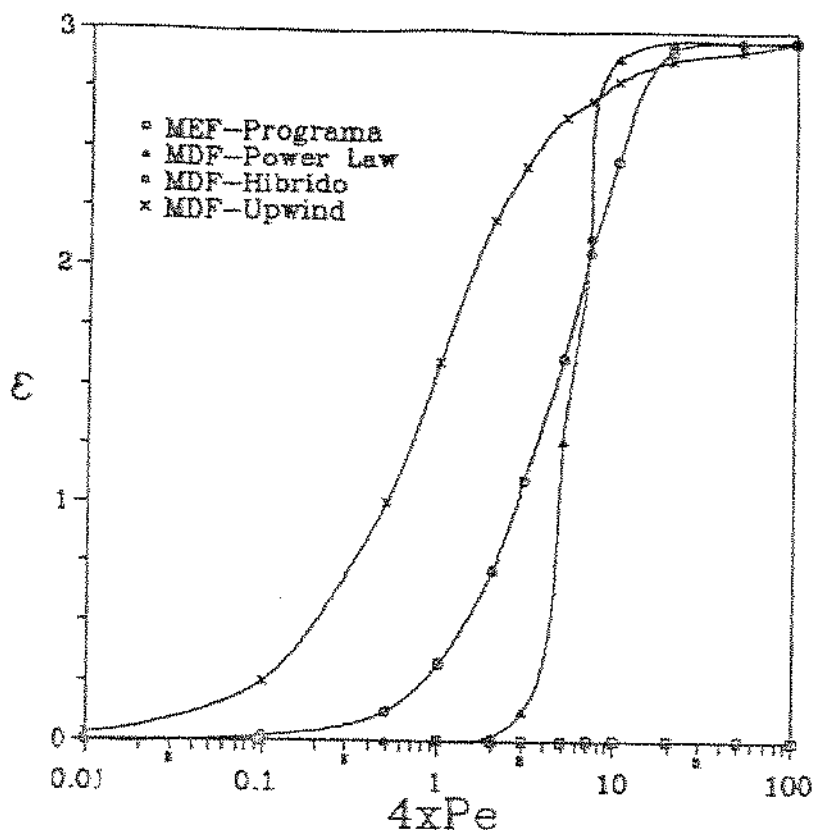


Figura 4.12- Resultados cilindro girante

4.6- CONVECCÃO EM DEGRAU

O problema proposto é ilustrado na figura 4.14. O campo de velocidades é uniforme e paralelo à linha que atravessa o domínio e passa pelo seu centro, partindo-se de y_c . Não há termo de geração e o regime é permanente. O coeficiente de difusão (Γ) é nulo. Os valores nas fronteiras são $\phi = 0$, abaixo da linha sólida; $\phi = 1$, acima da linha sólida; $\phi = 0.5$ nos pontos onde a linha sólida intercepta as fronteiras. A solução analítica do problema é imediata sendo mostrada nas figura 4.14. Este problema é um teste particularmente severo para a falsa-difusão.

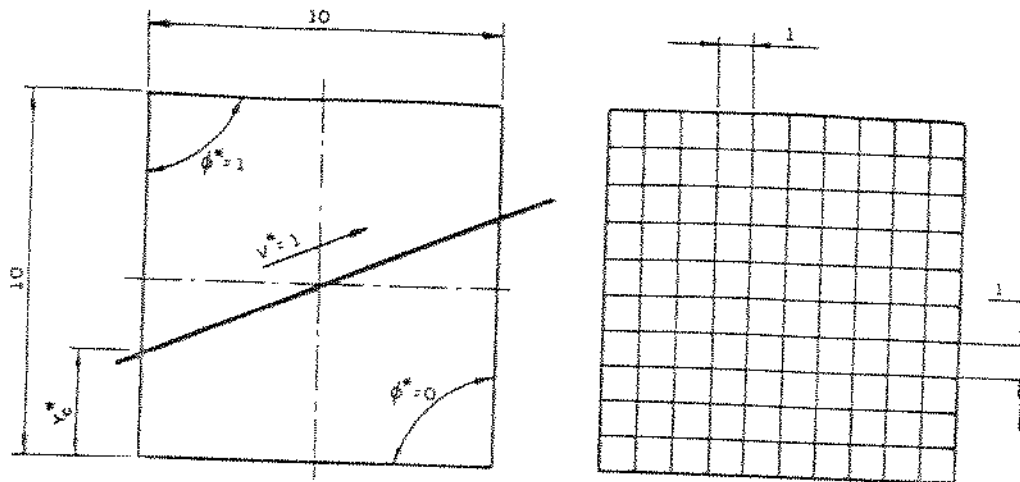


Figura 4.13- Conveccão em degrau

Uma malha regular 10×10 foi utilizada para resolver este problema. Comparações são feitas com o método de volumes de controle proposto por Baliga²⁰ e com um esquema de diferenciação por *upwind*. O programa é testado para quatro valores diferentes de y_c ($y_c = 0, 3, 8, 10$). Para $y_c = 5$ a solução é nodalmente exata e por isso não é apresentada. A figura 4.15 apresenta os resultados obtidos e os compara com os outros métodos (Nota: as curvas para os métodos de Baliga e diferenças finitas (*upwind*) foram retirados diretamente de gráficos da literatura²⁰ e podem ser ligeiramente imprecisas).

Para $y_c = 0$, vemos que tanto os resultados de Baliga quanto os do método aqui empregado dão resultados muito próximos da solução exata. O esquema *upwind* apresenta uma falsa difusão considerável.

Para $y_c=3$, o esquema *upwind* diminui a falsa-difusão, mas ainda assim, esta é maior que a do método de Baliga ou do método aqui empregado. Nota-se que o método aqui empregado e o de Baliga

apresentam índices de falsa difusão muito próximos.

Para $yc = 8$ o esquema *upwind* continua apresentando uma maior falsa difusão. O método de Baliga passa a apresentar uma maior falsa difusão que o método aqui empregado, além de uma maior oscilação.

No caso final, $yc = 10$, equivalente a uma direção simétrica em relação a $yc = 0$ (direções de escoamento de 45° e -45° , respectivamente), a falsa difusão no método aqui apresentado, volta a ter índices muito baixos, o que é o esperado para uma situação simétrica a $yc = 0$. O mesmo não ocorre no método de Baliga revelando uma dependência com a orientação do elemento. O esquema *upwind* é novamente, o que apresenta maiores índices de falsa difusão.

Na figura 4.15 é feita uma comparação entre os resultados obtidos usando-se uma malha regular e uma malha irregular. A malha irregular é a mesma utilizada na figura 4.5. A comparação é feita para $yc = 0$ e $yc = 10$ por terem solução nodalmente exata. Evidentemente a solução para a malha irregular será diferente para $yc = 0$ e $yc = 10$ (já que a malha não é simétrica). Pode-se verificar que mesmo usando-se a malha irregular os resultados são ainda superiores a solução por diferenças finitas pelo esquema *upwind*.

Na tabela 4.6 são mostrados os valores do tempo de CPU para duas situações. Na primeira (IREP = 0) todas as matrizes locais dos elementos são calculadas. Na segunda os cálculos são realizados apenas para um elemento e a matriz gerada é utilizada para todos os demais elementos (IREP = 1).

Tabela 4.6 - Tempo de CPU para convecção em degrau

	IREP = 0	IREP = 1
Tempo de CPU (s)	7.4	6.7

A economia de tempo usando-se a opção IREP=1, é de cerca de 10%. Deve-se ter cuidado em utilizar essa opção apenas em situações onde todas as matrizes sejam realmente iguais, considerando-se não só a geometria, mas também, a condutividade, e o campo de velocidades.

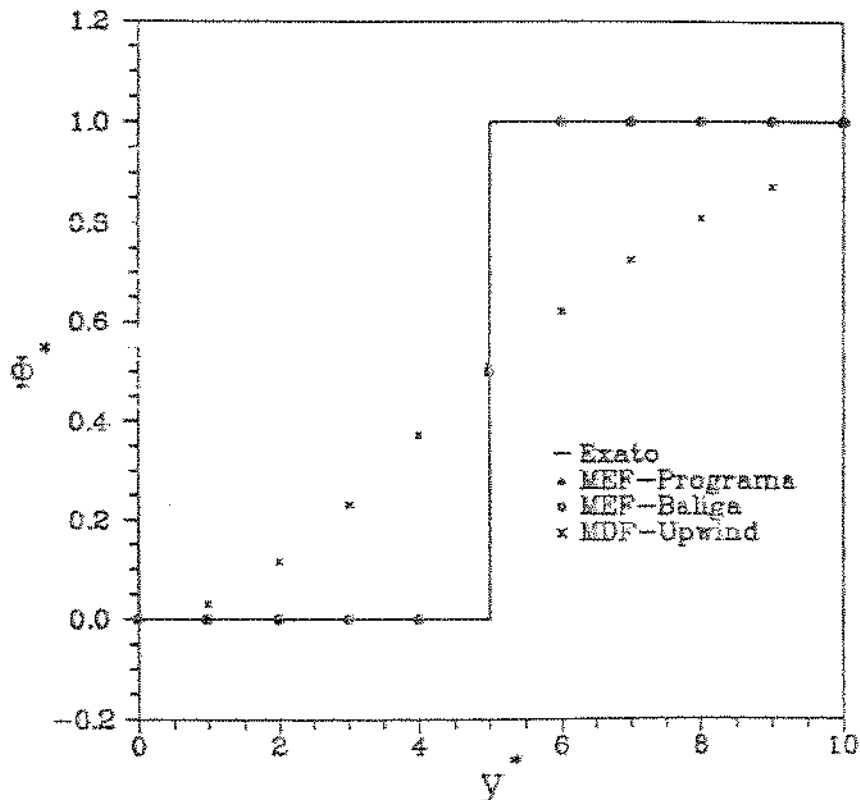


Fig. 4.14.a - Convecção em degrau ($y_c = 0$)

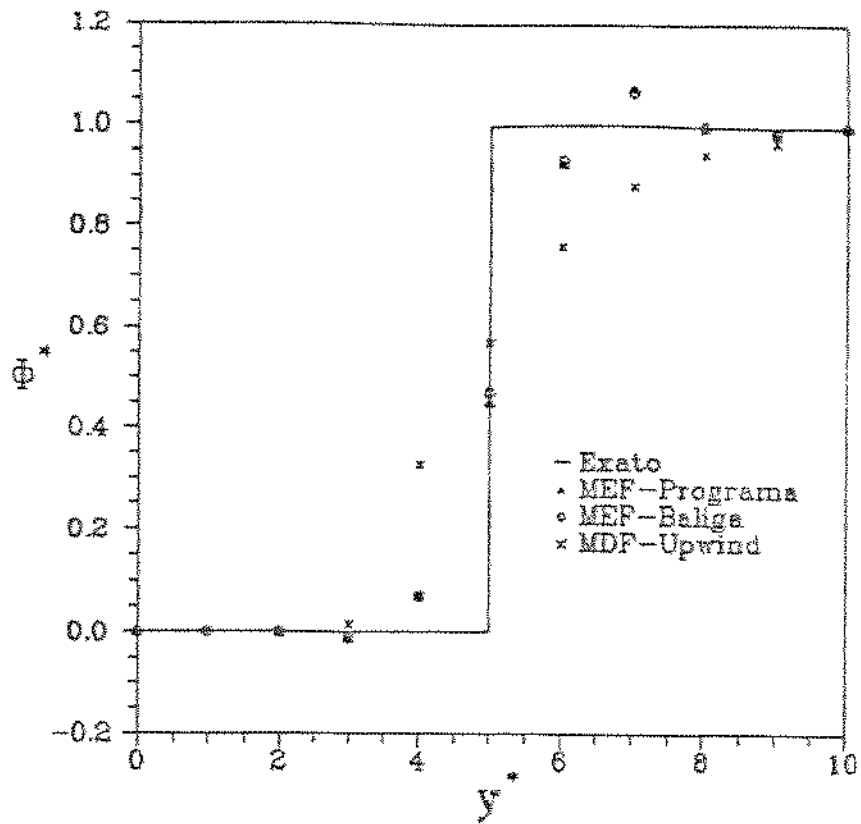


Fig. 4.14.b - Convecção em degrau ($y_c = 3$)

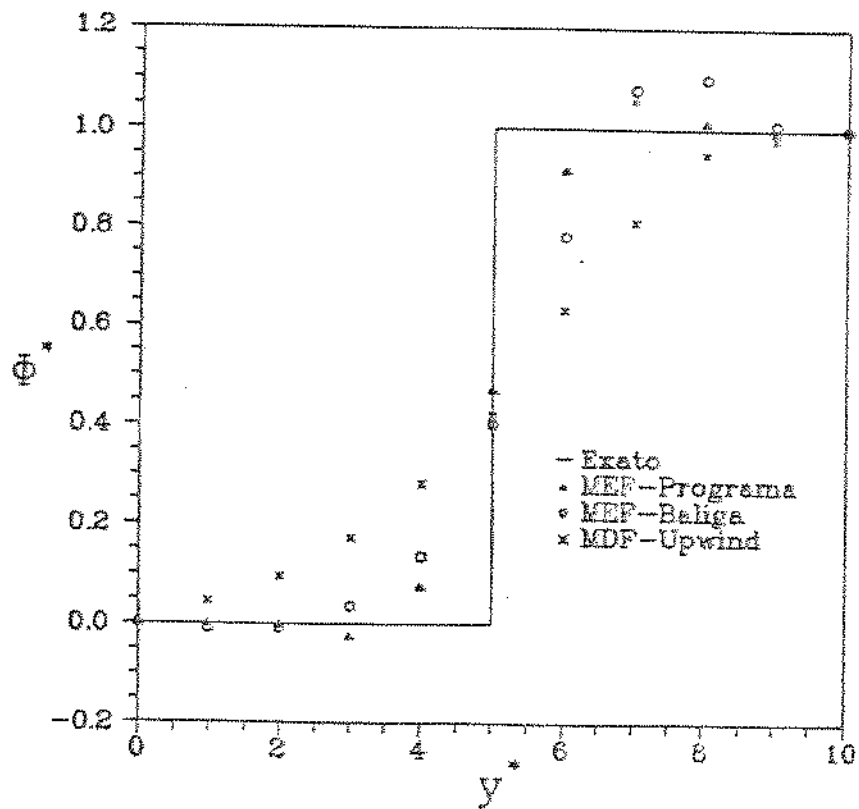


Fig. 4.14.c - Convecção em degrau ($y_c = 2$)

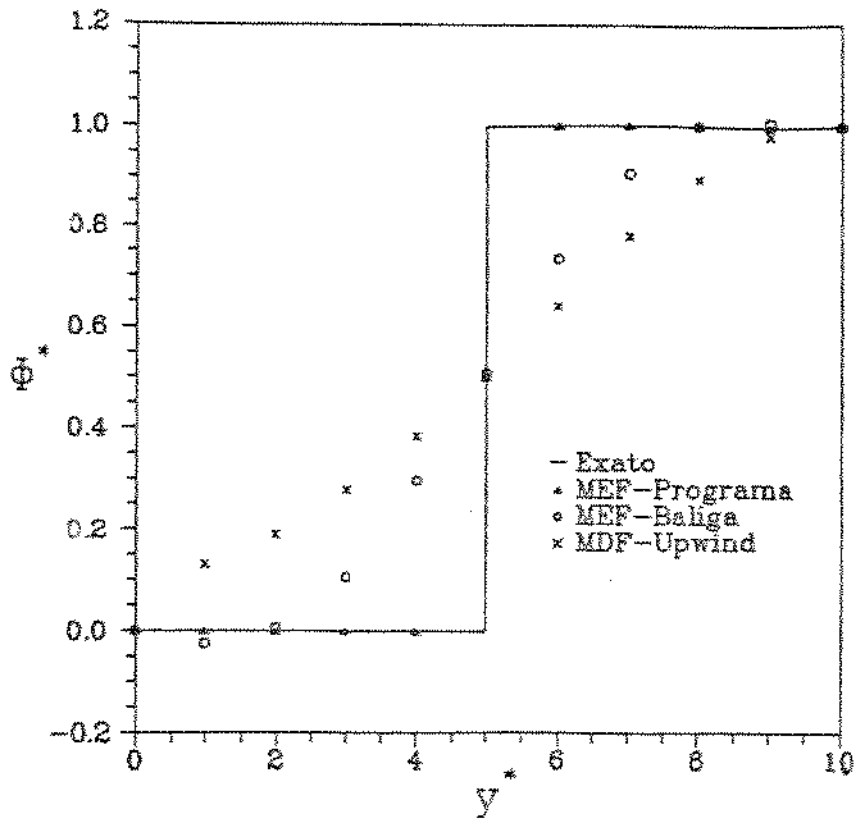


Fig. 4.14. d - Convecção em degrau ($y_c = 10$)

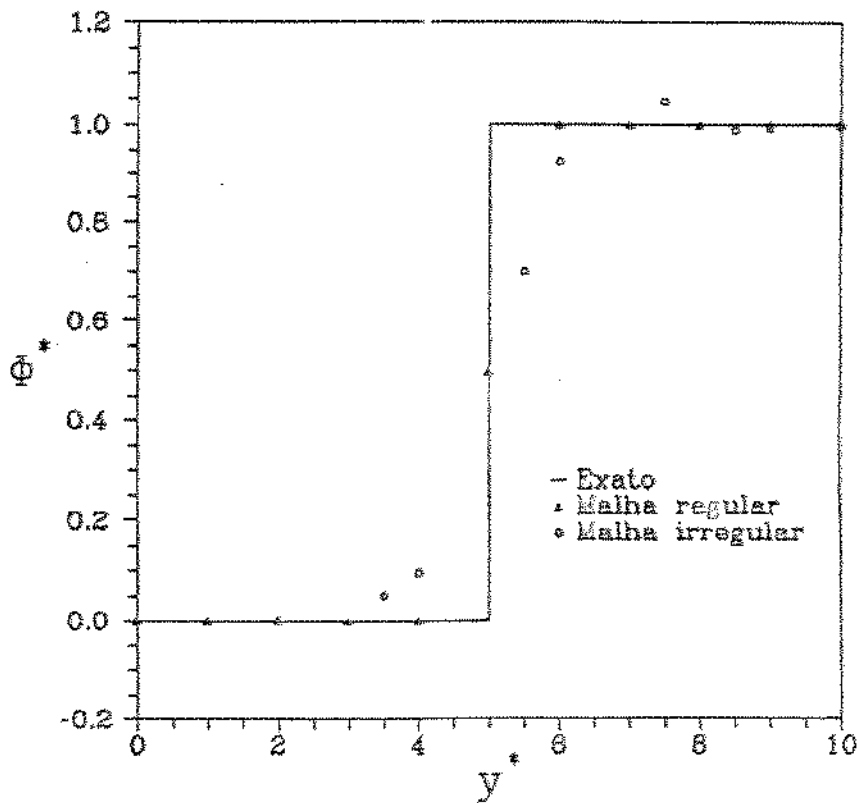


Fig. 4.15. a - Comparação entre malha regular e malha irregular ($y_c = 0$) - método implementado

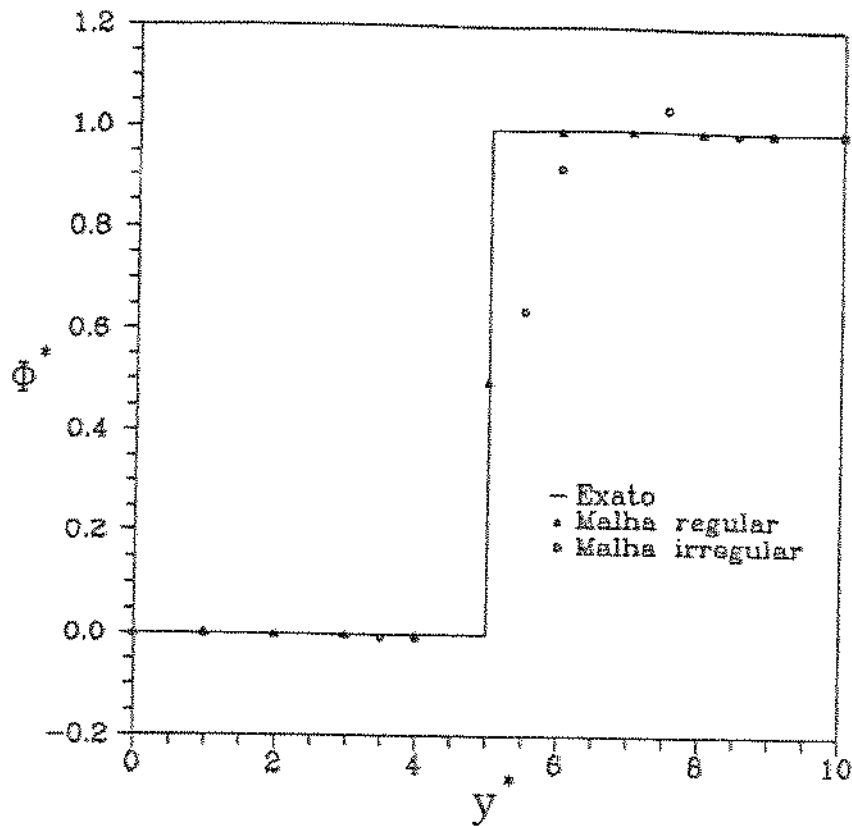


Fig. 4.15.b - Comparação entre malha regular e malha irregular ($y_c = 10$) - método implementado

4.7- CONVECÇÃO-DIFUSÃO EM UM FLUXO REVERSO DE 180°

O problema é ilustrado na figura 4.16. A equação que rege o problema é:

$$\vec{\nabla} \times \nabla \phi - \frac{1}{Pe} \nabla^2 \phi = 0 \quad (4.8)$$

onde:

$$\vec{\nabla} = u \vec{i} + v \vec{j}$$

e

$$u = 2y(1 - x^2)$$

$$v = -2x(1 - y^2)$$

com fronteiras definidas por:

$$\phi = 1 + \operatorname{tgh} [(2x + 1)\gamma] \quad y = 0, \quad -1 \leq x \leq 0$$

$$\phi = 1 - \operatorname{tgh}(\gamma) \left\{ \begin{array}{ll} x = -1 & 0 \leq y \leq 1 \\ y = 1 & -1 \leq x \leq 1 \\ x = 1 & 0 \leq y \leq 1 \end{array} \right.$$

onde:

$$\gamma = 10$$

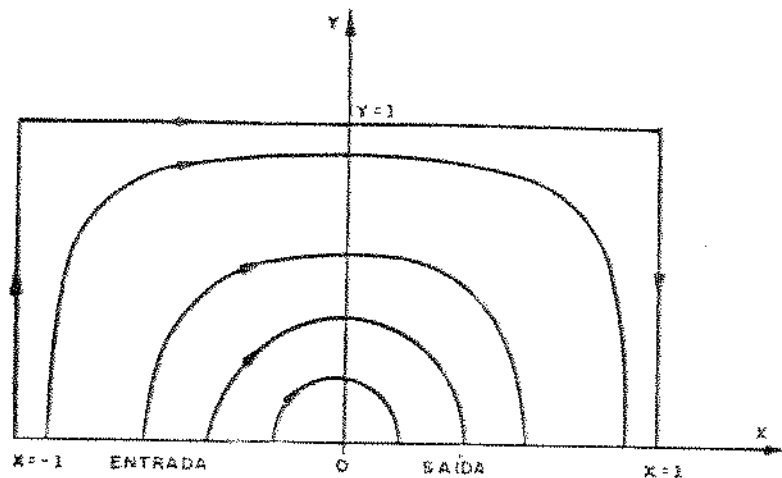


Fig. 4.16 Fluxo reverso

As figuras 4.17 a 4.20 mostram os resultados encontrados para diferentes números de Peclet.

Este problema foi testado com uma série de métodos⁴⁸, usando malhas tão finas quanto 100x50, e assim, apesar do problema só possuir solução analítica para Peclet infinito, as soluções aqui apresentadas como "exatas" certamente mostram a tendência correta da solução. Em todos os casos os resultados mostraram-se melhores que os de um esquema para diferenças finitas tipo *upwind*. Nesse problema foi utilizado um fator de ponderação ($\alpha = 10$) para

todos os números de Peclet, pois a partir desse valor de α a solução se torna praticamente invariante. A tabela 4.7 mostra a variação da solução no ponto ($x= 0.5$) e ($y= 0.0$) em função de diferentes valores de α para Peclet 500 e infinito. Para valores menores do número de Peclet, a influência de α é menos significativa, já que a influência da difusão se torna preponderante. A figura 4.22 ilustra as isotemperaturas para Peclet 10 e Peclet infinito. O tempo total de CPU para a resolução desse problema foi de 25.3 segundos.

Tabela 4.7 - Variação da solução em função de α para o ponto (0.5, 0.0)

α	Peclet			
	500		infinito	
	númeroico	"exata"	númeroico	exata
0.000	xxxxxxxx	0.947	0.000159	1.000
0.002	-0.24571		xxxxxxxx	
0.010	11.5040		-0.135270	
0.100	0.93804		1.252359	
1.000	0.93324		1.007365	
10.000	0.93301		1.000553	
1E02	0.93299		1.000007	
1E03	0.93299		1.000001	
1E04	0.93299		1.000000	

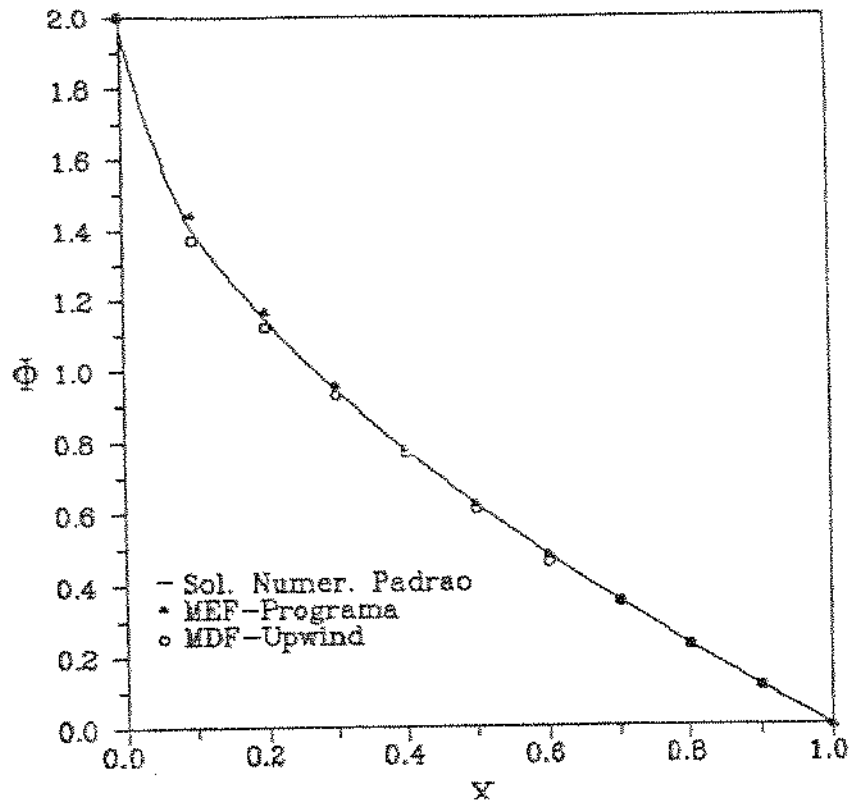


Fig 4.17 - Peclet = 10 (malha 20x10)

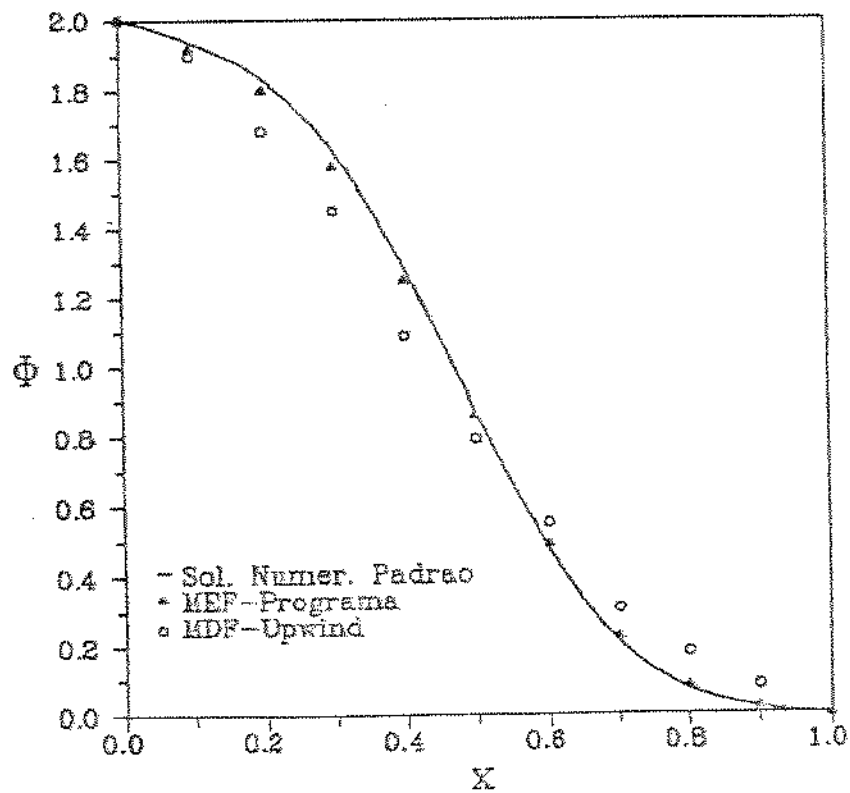


Fig. 4.18 - Peclet = 100 (malha 20x10)

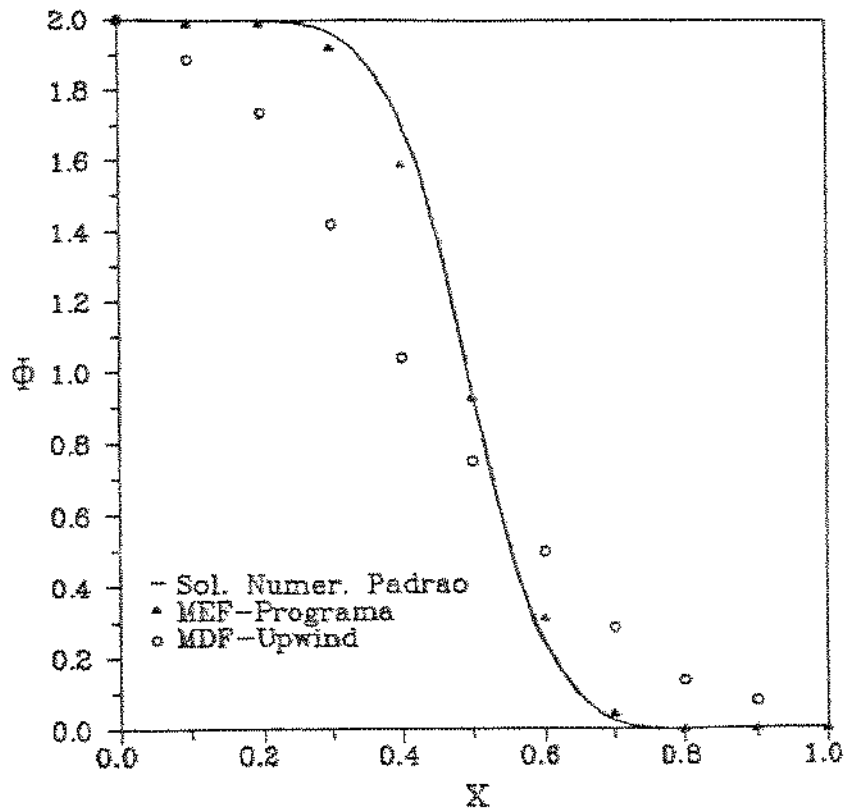


Fig. 4.19 - Peclet = 500 (malha 20x10)

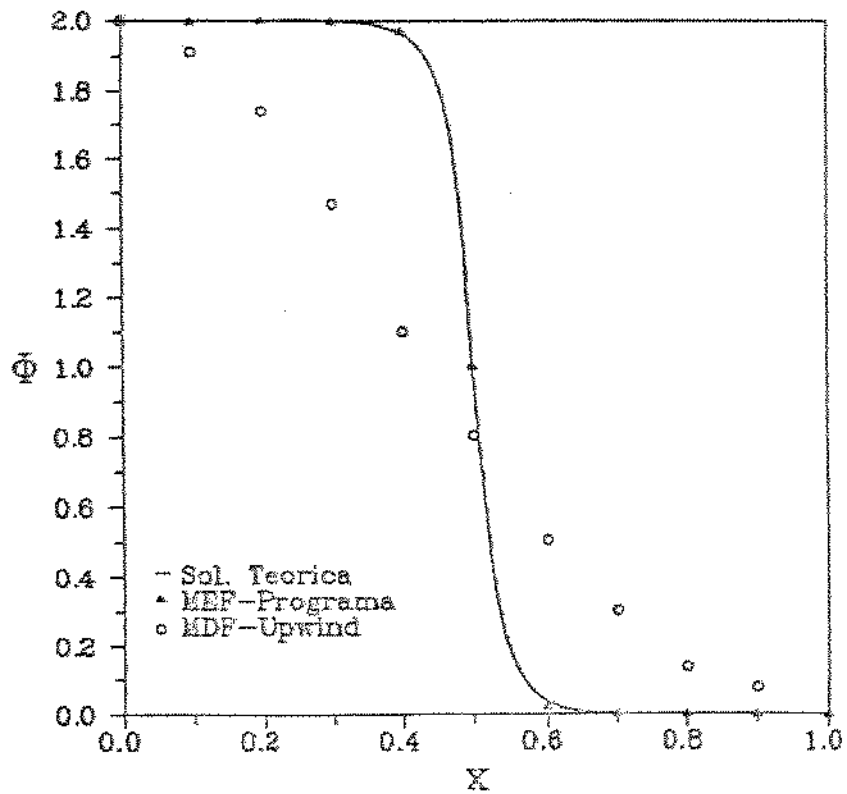


Fig. 4.20 - Peclet infinito (malha 20x10)

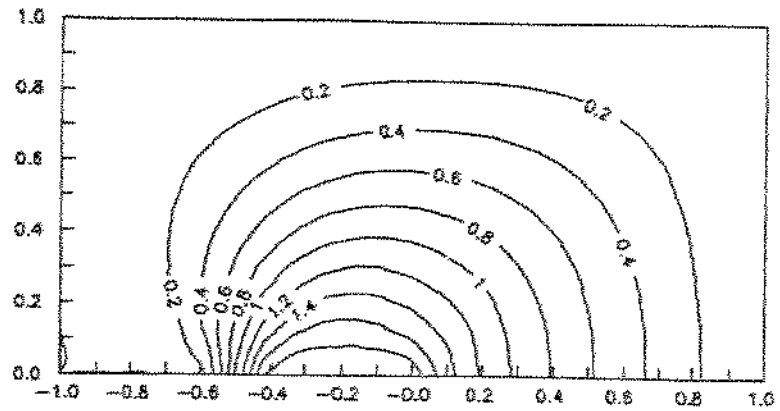


Figura 4.21.a - Isotérmicas para fluxo reverso (Peclet = 10)

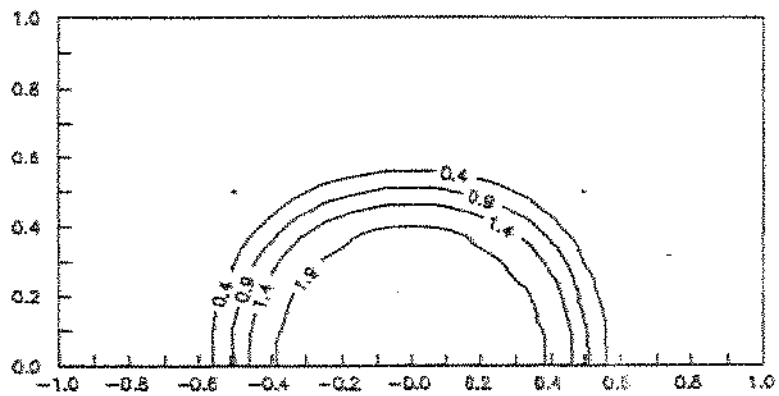


Figura 4.21.b - Isotérmicas para fluxo reverso (Peclet infinito)

4.8 - CONVECCÃO-DIFUSÃO UNIDIMENSIONAL COM GERAÇÃO NÃO UNIFORME DE CALOR

Este problema visa testar a eficiência do método no

tratamento de termos de geração não uniforme. O problema escolhido possui solução analítica⁵⁰, e é definido pela seguinte expressão em coordenadas adimensionais:

$$\frac{\partial \phi}{\partial x} = s + \frac{1}{Pe} \frac{\partial^2 \phi}{\partial x^2} \quad 0 \leq x \leq 1 \quad (4.9)$$

onde s é dado pela expressão:

$$s = 1 - \frac{2}{Pe} + 2x + \pi \cos(\pi x) + \frac{\pi^2}{Pe} \sin(\pi x) \quad (4.10)$$

com condições de contorno do tipo:

$$\phi \begin{cases} 1 + \exp(-Pe) & x = 0 \\ 4 & x = 1 \end{cases}$$

A solução exata do problema é dada por⁵⁰:

$$\phi = 1 + x + x^2 + \exp[-Pe(1-x)] + \sin(\pi x) \quad (4.11)$$

O problema foi resolvido através de uma malha de 10x1 elementos, igualmente espaçados. A largura na direção y foi tomada como 0,1, sendo a velocidade na direção y estabelecida como zero e na direção x como um. O problema foi resolvido para diferentes números de Peclet e assim elaborado um gráfico do "Erro Médio" versus Peclet (figura 4.22). O "Erro Médio" é calculado pela seguinte expressão:

$$\varepsilon_m = \frac{\sum \varepsilon}{\text{todos os pontos}} \quad \text{número de pontos} \quad (4.12)$$

onde ϵ é dado pelo módulo da expressão (4.3) para cada ponto nodal.

Na equação (4.12) são tomados apenas nós internos, já que os pontos de contorno são impostos.

A figura 4.22 compara os resultados do programa FEMEV com o esquema LOADS⁵¹ de Wong e Raithby e com um esquema exponencial para diferenças finitas.⁵⁰ Vê-se que para números de Peclet inferiores a 1 a precisão do método aqui empregado é comparável a do esquema LOADS e superior a do esquema exponencial. A partir de Peclet igual a 1 o erro médio aumenta rapidamente e a partir de Peclet 50 volta a diminuir. Na faixa de Peclet 1 a 100 a solução é superior ao esquema exponencial mas inferior ao esquema LOADS.1

A figura 4.23 compara a distribuição de temperaturas para a solução numérica e a solução exata para Peclet 100. Não notou-se oscilações espaciais na solução numérica. O tempo de CPU foi de 0.6 segundos numa estação de trabalho Sparc SUN. Num microcomputador NEC PC-9801 RS, equivalente a um computador da série 386, o tempo de CPU foi de 8 segundos. Nessa solução é usado o cartão CFIP (capítulo 3) igual a 1 afim de se utilizar interpolações por pontos nodais nos volumes de controle de fronteira.

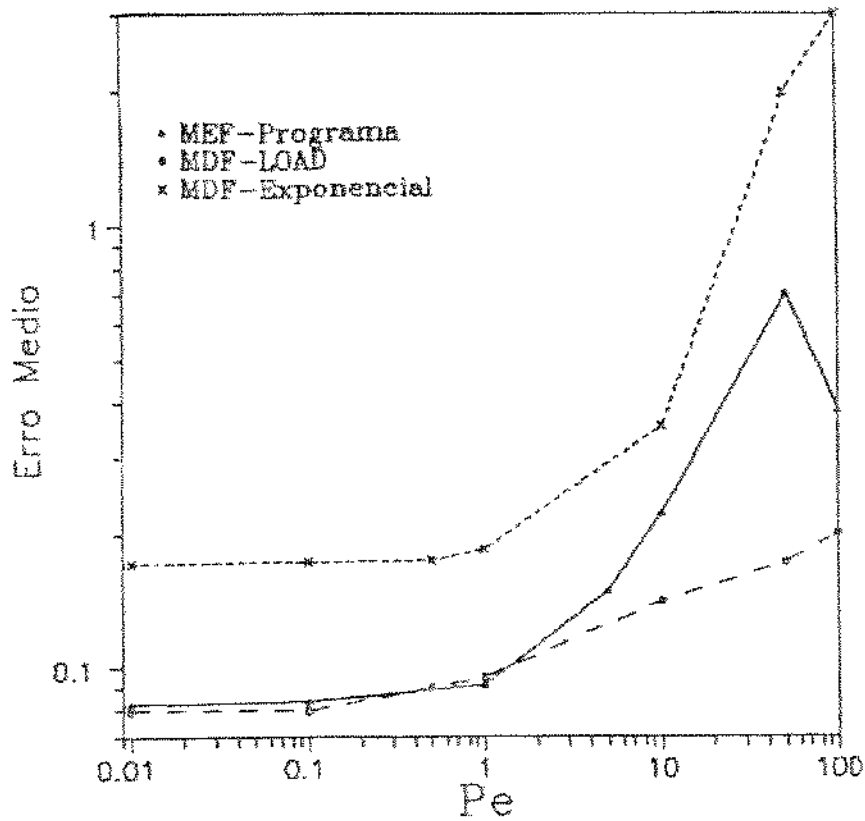


Figura 4.22- Erro médio versus Peclet para conveccão-difusão unidimensional com geração não uniforme de calor

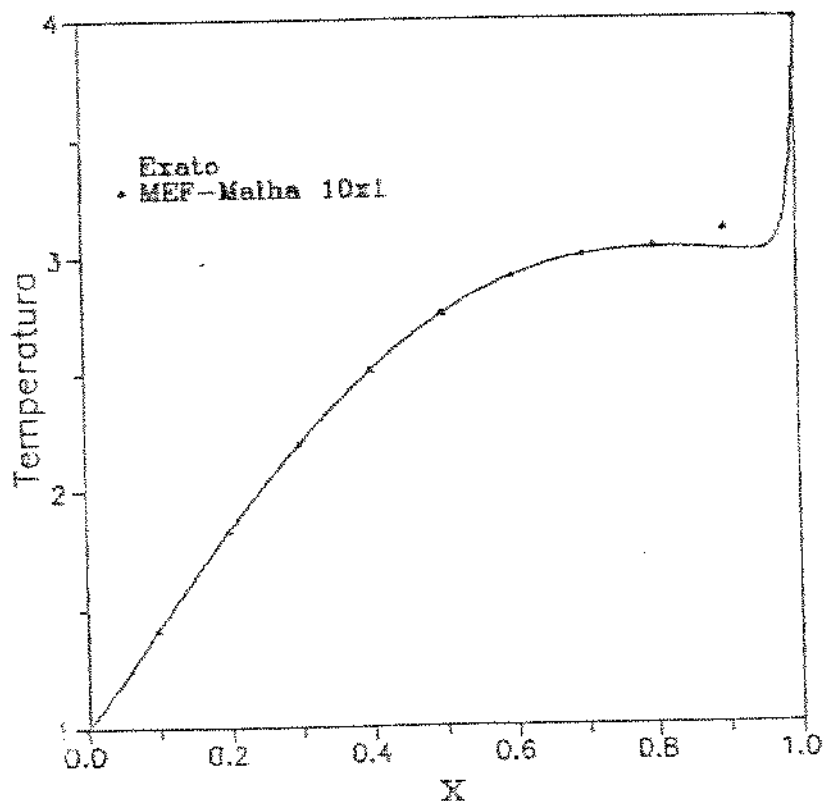


Figura 4.23- Distribuição de temperaturas para conveccão-difusão unidimensional com geração não uniforme de calor (Peclet = 100)

CONCLUSÕES E DISCUSSÕES

Nesse trabalho foi apresentado um novo método para elementos finitos, destinado a aplicações envolvendo convecção-difusão. Este novo método é baseado numa formulação estritamente conservativa e difere da maioria dos enfoques convencionais para elementos finitos, geralmente baseados no método de Galerkin. Apesar da implementação aqui desenvolvida ser limitada a campos de velocidades conhecidos, o trabalho apresenta uma grande potencialidade tanto em relação a sua utilização imediata, como em relação aos futuros desenvolvimentos. O método empregado pode ser considerado como o estado da arte em aplicações envolvendo o método dos elementos finitos, baseado em volumes de controle, em problemas de convecção-difusão. Em todos os casos aqui testados o método apresentou índices muito baixos de falsa difusão, mostrou-se relativamente insensível ao sentido de orientação da malha, e mesmo em geometrias não regulares, como no caso da convecção em degrau com malha irregular, apresentou baixos índices de falsa difusão. Sua maior limitação se encontra em casos envolvendo geração não uniforme de calor para altos números de Peclet, onde apresentou um performance inferior ao esquema LOAD.

O método parece bastante recomendável em aplicações que envolvam geometrias complexas e não ortogonais, ou em simulações numéricas de escoamentos. Na verdade o programa FEMEVI pode ser encarado como um pré-requisito básico para uma futura implementação envolvendo a resolução numérica das equações de

Navier-Stokes, onde as equações de quantidade de movimento são, basicamente, equações relacionadas a convecção e difusão dos campos de velocidade. Na medida do possível, as rotinas aqui desenvolvidas para o programa FEMEV foram tornadas o mais modulares possível, levando em conta uma futura expansão para o tratamento de problemas de escoamento

Com relação ao aperfeiçoamento e utilização do programa algumas sugestões e observações podem ser feitas:

a) A primeira seria quanto a implementação do método frontal ou skyline para um tratamento mais otimizado da alocação de memória pelo programa. Isso permitiria o tratamento de problemas com um número muito maior de nós.

b) Acrescentar a possibilidade de resolução de problemas transientes e aperfeiçoar o tratamento dos termos de geração interna de calor.

c) Acrescentar a possibilidade de resolução de escoamentos de fluidos. Essa possibilidade representaria um grande potencial, permitindo a análise de escoamentos em geometrias complexas, como por exemplo a análise de escoamentos em bombas do tipo radial.

d) Outra sugestão seria quanto a forma de desenvolvimento das rotinas do programa. Uma tendência observada em pacotes comerciais é com relação a modularidade das rotinas. Na medida do possível essa filosofia deveria orientar futuros aperfeiçoamentos que possam vir a ser feitos ao programa FEMEV. Isso facilitará a implementação de novos recursos e modificações no software segundo as necessidades dos possíveis usuários.

e) Uma grande necessidade em programas envolvendo elementos finitos são programas de pós e pré processamento de dados, além de geração automática de malhas. Infelizmente esses tipos de programa não se encontram disponíveis em nosso ambiente, apesar de softwares comerciais muito bons estarem disponíveis no mercado. A falta desses recursos pode tornar extremamente penosa ou mesmo inviável uma análise por elementos finitos em problemas complexos. A medida em que esses recursos se tornem disponíveis, é possível que um maior interesse pelo método possa ser sentido.

f) Nossa última sugestão é quanto ao processamento de erros em dados de entrada. Seria extremamente valioso a elaboração de rotinas mais completas para a análise e processamento dos dados de entrada que pudessem detectar prontamente possíveis erros, evitando assim, um processamentos inútil de dados incorretos.

Naturalmente a execução ou não de tais sugestões dependerão do grau de interesse que o método possa despertar. Acreditamos haver um grande potencial no método e que muitos aperfeiçoamentos ainda poderão ser implementados.

Finalizando, o método dos elementos finitos não deve ser encarado de forma preconceituosa ou extremamente otimista. Na verdade ele é simplesmente mais uma ferramenta numérica, que poderá gerar bons ou maus resultados dependendo da utilização em que seja empregado. E somente através de uma série de testes e aperfeiçoamentos é que poderemos predizer em quais problemas sua utilização será plenamente satisfatória.

REFERÊNCIAS

- 1- T.J. Chung, *Finite Element Analysis of Fluid Dynamics*, Mc Graw-Hill, New York, 1978.
- 2- C. Taylor and T.G. Hughes, *Finite Element Programming of the Navier-Stokes Equations*, Pineridge Press Ltd., 1981.
- 3- Henderson, F.M, *Open Channel Flow*, The Macmillan Company, New York, 1966.
- 4- D.A. Anderson, *Computational Fluid Mechanics and Heat Transfer*, Hemisphere, New York, 1984.
- 5- A.J. Baker, *Finite Element Computational Fluid Mechanics*, McGraw-Hill, New York, 1983.
- 6- W. Weaver, Jr. and P.R. Johnston, *Finite Elements for Structural Analysis*, Prentice-Hall, 1984.
- 7- J.J. Connor and C.A. Brebbia, *Finite Element Techniques for Fluid Flow*, Newnes-Butterworths, 1976.
- 8- J.E. Akin, *Application and Implementation of Finite Element Methods*, Academic Press, London, 1982.
- 9- G.E. Schneider and M.J. Raw, A Skewed Positive Influence Coefficient Upwinding Procedure for Control-Volume Base Finite-Element Convection-Diffusion Computation, *Numerical Heat Transfer*, 9, 1-26, 1985.

- 10- P.S. Huyakorn, Solution of Steady-State, Convective Transport Equation Using an Upwind Finite Element Scheme, *Appl. Math. Model.*, 1, 187-195, 1977
- 11- J.C. Heinrich and O.C. Zienkiewicz, Quadratic Finite Element Schemes for Two-dimensional Convective-Transport Problems, *Int. J. Numer. Methods Eng.*, 11, 1831-1844, 1977.
- 12- I. Christie, D.F. Griffiths, A.R. Mitchel and O.L. Zienkiewicz, Finite Element Methods for second Order Differential Equations with Significant First Derivatives, *Int. J. Numer. Methods Eng.*, 10, pp. 1389-1396, 1976.
- 13- Turner, M.J., Clough, R.W., Martin, H.C., and Topp, L. P., Stiffness and Deflection Analysis of Complex Structures, *J. Aeron. Sci.*, 23, No. 9, 805-823, 1955.
- 14- Clough, R. W. ,The Finite Element Method in Plane Stress Analysis , *Proceedings of 2nd Conf. on Electronic Computation*, American Society of Civil Engineers, Pittsburgh Penn., pp. 345-378, 1960.
- 15- Argyris, J. H., *Recent Advances in Matrix Methods of Structural Analysis*, Pergamon Press, Elmsford, New York, 1963.
- 16- Zienkiewicz, O.C., and Cheung, Y.K, Finite Elements in the Solution of Field Problems, *The Engineer*, pp. 507-510, 1965.
- 17- Oden, J.T., *Finite Elements of Nonlinear Continua*, McGraw-Hill, New York, 1972.

- 18- Rayleigh, J. W. S., Theory of Sound, 1st edn revised, Dover Publications, Inc., New York, 1945.
- 19- B. A. Finlayson, The Method of Weighted Residuals and Variational Principles, Academic Press, New York, 1972.
- 20- B. R. Baliga and S.V. Patankar, A New Finite-Element Formulation for Convection-Diffusion Problems, *Numerical Heat Transfer*, 3, 393-409, 1980.
- 21- G.E. Schneider and M.J. Raw, Control Volume Finite-Element Method for Heat Transfer and FLuid FLOW Using Collocated Variables - 1 Computational Procedure. *Numerical Heat Transfer*, 11, 363-390, 1987.
- 22- G.E. Schneider and M.J. Raw, Control Volume Finite-Element Method for Heat Transfer and FLuid FLOW Using Collocated Variables - 2 Computational Procedure. *Numerical Heat Transfer*, 11, 391-400, 1987.
- 23- K. V. Roberts and N. O. Weiss, Convective Difference Schemes, *Math. Comput.*, 20, 272-299, 1966.
- 24- O. R. Burggraf, Analytical and numerical studies of steady separated flows. *J. Fluid Mech.* 24, 113-151, 1966.
- 25- R. A. Gentry, R. E. Martin, and B. J. Daly, An Eulerian Differencing Method for Unsteady Compressible Flow Problems, *J. Comput. Phys.*, 1, 87-118, 1966.
- 26- A. K. Runchal and M. Wolfshtein, A finite difference procedure for the integration of the Navier-Stokes equations, *J. Mech.*

- Engng Sci. 11, 445-453, 1969.
- 27- D.B. Spalding, A Novel Finite Difference Formulation for Differential Expressions Involving Both First and Second Derivatives, *Int. J. Numer. Methods Eng.*, 4, 551-559, 1962.
- 28- S.V. Patankar, *Numerical Heat Transfer and Fluid Flow*, Hemisphere, Washington, D.C., 1980.
- 29- F.G. Blottner, Numerical Solution of Convection-Diffusion Equations, *Comput. Fluids*, 6, 15-24, 1978.
- 30- G.D. Raithby, A Critical Evaluation of Upstream Differencing Applied to Problems Involving Fluid Flow, *Comput. Methods Appl. Mech. Eng.*, 19, 59-98, 1979.
- 31- G. de Vahl Davis and G. D. Mallinson, An Evaluation of Upwind and Central Difference Approximations by a Study of Recirculating Flow, *Comput. Fluids*, 4, 24-43, 1976.
- 32- G.D. Raithby, Skew Upstream Differencing Schemes for Problems Involving Fluid Flow, *Comput. Methods Appl. Mech. Eng.*, 9, 153-164, 1976.
- 33- P.M. Gresho and R.L. Lee, Don't Suppress the Wiggles - They're Telling You Something!, in T.J.R. Hughes (ed.), *Proceedings of a Symposium on Finite Element Methods for Convection Dominated Flows*, pp. 37-61, ASME Winter Annu. Meet., New York, 1979.
- 34- J.C. Heinrich, P.S. Huyakorn, O.C. Zienkiewicz and A.R. Mitchell, An "Upwind" Finite Element Scheme for Two-dimensional Convective Transport Equation, *Int. J. Numer.*

Methods Eng., 11, 131-143, 1977.

- 35- T.J.R. Hughes, and A. Brooks, Finite Element Analysis of Incompressible Viscous Flows by the Penalty Function Formulation, *J. Comp. Phys.*, 30, 1-60, 1979.
- 36- B.R. Baliga and S.V. Patankar, A Control Volume Finite-Element Method for Two-Dimensional Fluid-Flow and Heat Transfer, *Numerical Heat Transfer*, 6, 245-261, 1983.
- 37- B.R. Baliga, S.V. Patankar, Elliptic Systems: Finite-Element Method II, *Handbook of Numerical Heat Transfer*, John Wiley & Sons, 1988.
- 38- B.R. Baliga, T.T. Pham and S.V. Patankar, Solution of Some Two-Dimensional Incompressible Fluid Flow and Heat Transfer Problems, Using a control Volume Finite-Element Method, *Numerical Heat Transfer*, 6, 263-282, 1983.
- 39- N.A. Hookey, and B.R. Baliga, Evaluation and Enhancements of Some Control Volume Finite-Element Methods. Part 1. Convection-Diffusion Problems. *Numerical Heat Transfer*, 14, 255-272, 1988.
- 40- N.A. Hookey and B.R. Baliga, Evaluation and Enhancements of Some Control Volume Finite-Element Methods - Part 2. Incompressible Fluid Flow Problems, *Numerical Heat Transfer*, 14, 273-293, 1988.
- 41- C. Prakash, An Improved Control Volume Finite Element Method for Heat and Mass Transfer, and for Fluid Flow Using

- Equal-Order Velocity-Pressure Interpolation, *Numerical Heat Transfer*, 9, 253-276, 1986.
- 42- C. Prakash, Examination of the Upwind (Donor-Cell) Formulation in Control Volume Finite-Element Methods for Fluid Flow and Heat Transfer, *Numerical Heat Transfer*, 11, 401-416, 1987.
- 43- C. Prakash and S.V. Patankar, A Control Volume-Based Finite-Element Method for Solving the Navier Stokes Equations Using Equal-Order Velocity-Pressure Interpolation, *Numerical Heat Transfer*, 8, 259-280, 1985.
- 44- G.E. Schneider, Elliptic Systems: Finite-Element Method I, *Handbook of Numerical Heat Transfer*, John Wiley & Sons, 1988.
- 45- P.J. Schneider, *Conduction Heat Transfer*, Addison-Wesley, 1955.
- 46- V. S. Arpaci, *Conduction Heat Transfer*, Addison-Wesley, 1966.
- 47- A.K. Runchal, Convergence and Accuracy of three Finite Difference Schemes for a Two-dimensional conduction and Convection Problem, *Int. J. Numer. Methods Eng.*, 4, 541-550, 1972.
- 48- R. M. Smith and A.G. Hutton, The Numerical Treatment of Advection: A Performance Comparison of Current Methods, *Numerical Heat Transfer*, 5, 439-461, 1982.
- 49- J.M. Marins, *Apostila de CMP-20, Cálculo Numérico*, ITA, 1987.

50- C. Prakash, Application of the Locally Analytic Differencing Scheme to Some Test Problems for The Convection-Diffusion Equation, *Numer. Heat Transfer*, 7, 165-182, 1984.

51- H. H. Wong and G. D. Raithby, Improved Finite Difference Methods Based on a Critical Evaluation of the Approximation Errors, *Numer. Heat Transfer*, 2, 139-163, 1979.

APÊNDICE A

O coeficiente α

No capítulo 2, seção 2.5, os operadores para os pontos de integração foram desenvolvidos em função de um coeficiente α , que poderia assumir um valor diferente do coeficiente de difusão em casos onde as linhas de corrente, dentro do elemento, não fossem paralelas à direção do vetor velocidade nos pontos de integração (ipi). Nessa seção explicaremos qual a influência da alteração do valor de α na solução de um problema.

Consideremos a discretização da equação 2.26 para um elemento quadrado. A discretização dessa equação para o subvolume de controle i será da forma:

$$C1 (\phi_u - \phi_i) - \frac{\alpha (N_j \bar{\phi}_j - \phi_i)}{C2} = s \quad (A1)$$

onde C1 e C2 são constantes que dependem das dimensões do elemento.

Desenvolvendo a equação A1:

$$\phi_i = \frac{C2 s + \alpha N_j \bar{\phi}_j - C3 \phi_u}{(\alpha - C3)} \quad (A2)$$

onde $C3 = C1 * C2$

Da expressão (A2) temos o seguinte comportamento:

Se $\alpha \rightarrow \infty$, $\phi \rightarrow N_j \bar{\phi}_j$ (ou seja, a interpolação para o termo convectivo se torna linear).

Para o caso de convecção pura ($\alpha = 0$), temos:

$$\phi_i = \phi_u \quad (\text{esquema upwind})$$

Dessa forma vimos que variando-se os valores de α entre 0 e ∞ , a interpolação para o termo convectivo variará entre dois limites bem definidos. Caso não haja uma convergência da solução para variações do valor de α , é possível que a discretização da malha seja insuficiente. Não há uma regra geral sobre quando variar ou não o valor de α . Em geral a melhor forma de se encontrar a solução de um problema será refinando a malha. A medida que se refine a malha a solução tanto por meio de uma interpolação linear como pelo uso dos pontos de integração deverá tender para um mesmo valor.

Nas figuras A1 e A2 são mostrados os efeitos da utilização de α igual a 10, na resolução do problema 4.8. Para esse valor de α as interpolações já são praticamente lineares. Vê-se que para Peclet igual a 100 o erro médio tornou-se superior ao do esquema exponencial e ao do esquema LOADS, o que não acontece quando é utilizado a interpolação pelos pontos de integração (figura 4.22). Na figura A2 são mostrados as distribuições de temperatura para diferentes refinamentos de malha usando-se uma interpolação linear ($\alpha=10$). Notam-se grandes oscilações espaciais na solução numérica, o que é uma característica de interpolações lineares para altos números de Peclet. A medida que o refinamento aumenta, as oscilações diminuem. Na figura 4.23 onde é utilizado a interpolação por pontos de integração, as oscilações espaciais inexistem mostrando

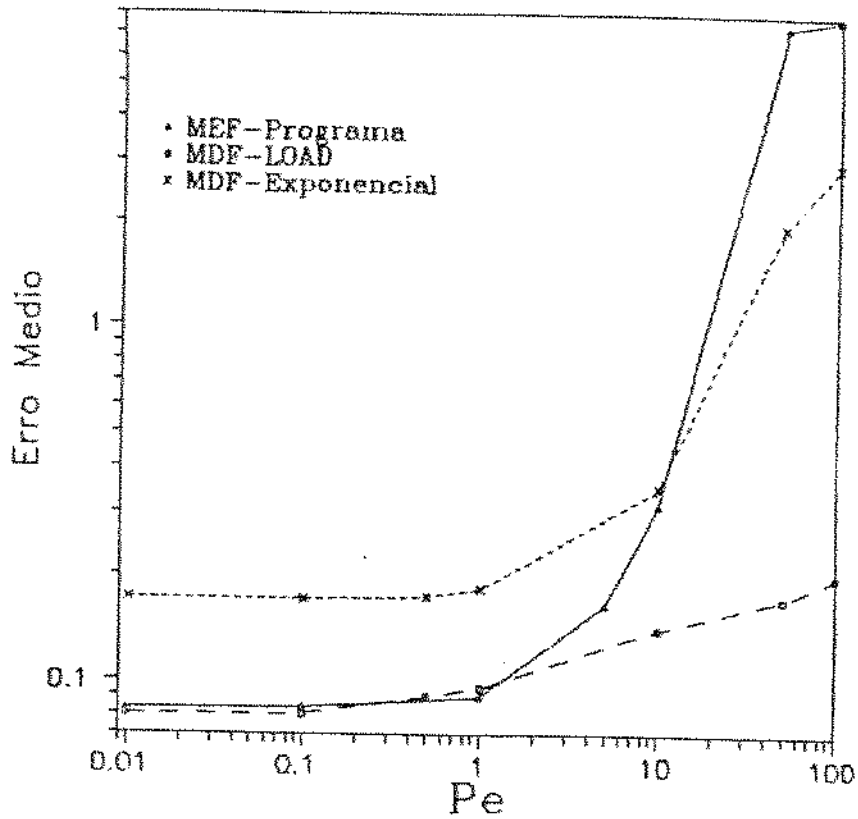


Figura A1- Problema 4.8: Erro Medio versus Peclet ($\alpha = 10$)

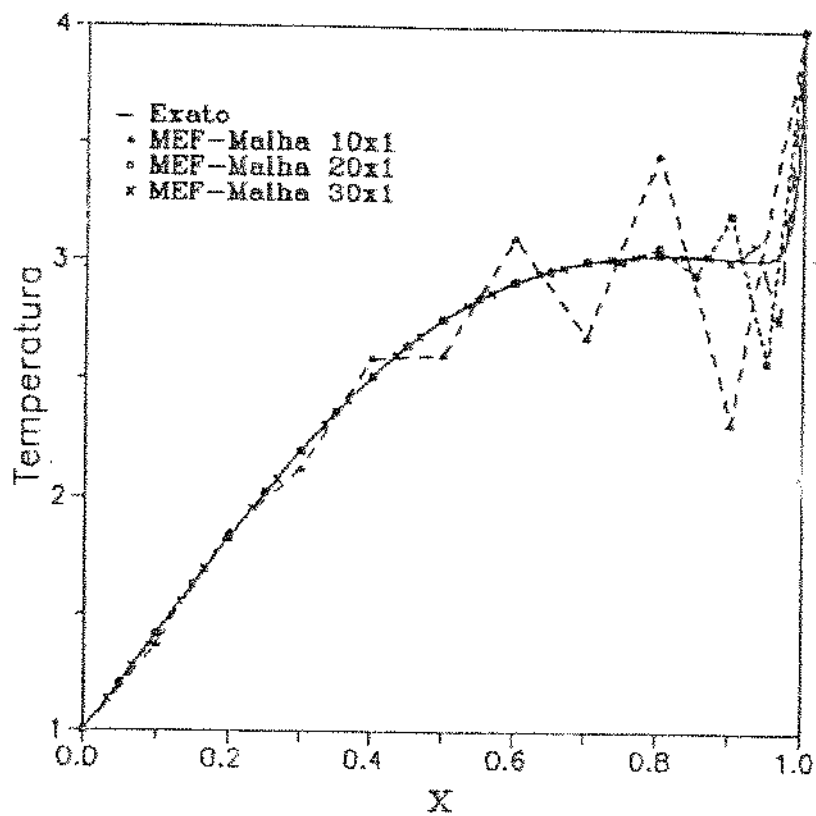


Figura A2- Problema 4.8: Distribuição de temperatura ($\alpha = 10$)

que o esquema utilizado é, realmente, capaz de refletir o comportamento físico esperado nesses pontos.

A tabela A1 mostra os tempos de CPU e a razão de convergência para os diferentes refinamentos da malha.

Tabela A1 - Ordem de Convergência para Peclet 100

número de elementos	ϵ_m	Tempo de CPU (s)
10	7.839	0.6
20	1.337	1.9
30	0.429	2.5

APÊNDICE B

RESULTADOS NÚMERICOS PARA O PROBLEMA 4.2 e TEMPOS de CPU pelo NÚMERO TOTAL de NÓS

Nesse apêndice são fornecidos alguns resultados numéricos para o problema 4.2, para $r = 3$ e os tempos de CPU pelo número total de nós.

TABELA B1 - Resultados Numéricos para o problema 4.2

Coordenadas (θ)	Teórico	Numérico	Erro(%)
0	270.1	273.7	1.1
30	589.6	596.1	1.1
70	518.8	529.2	2.0
90	538.3	549.6	2.1
120	495.7	502.1	1.3
150	389.6	393.1	0.9
220	209.4	211.3	0.9
270	196.0	198.4	1.2
300	389.6	393.5	1.0

TABELA B2- Tempos de CPU pelo número total de nós

Problema	Número de nós	tempo de CPU	Tempo de CPU/ número total de nós
4.1	12	0.3	0.025
	22	0.8	0.036
4.2	424	127.9	0.302
4.3	121	6.0	0.050
4.4	42	1.4	0.033
4.5	121	6.1	0.050
4.6	121	7.4	0.061
4.7	231	25.3	0.110
4.8	12	0.6	0.050

APÊNDICE C

PROGRAMA FEMEV-ARQUIVOS DE ENTRADA

Nesse apêndice são fornecidos os arquivos de entrada dos problemas 4.1 e 4.8 (Peclet =100), como exemplos para a elaboração de arquivos de entrada.

Programa FEMEV- Dados de entrada para o problema 4.1

(malha 5x1)

*Problema 4.2 (malha 5x1)

```
SOL      2      0.01      1  0
PROP     132          0
GRID     1      0.000000      -0.250000
GRID     2      0.000000       0.250000
GRID     3      3.000000      -0.175000
GRID     4      3.000000       0.175000
GRID     5      5.000000      -0.125000
GRID     6      5.000000       0.125000
GRID     7      8.000000      -0.050000
GRID     8      8.000000       0.050000
GRID     9      9.000000      -0.025000
GRID    10      9.000000       0.025000
GRID    11      9.950000      -0.001250
GRID    12      9.950000       0.001250
CQUAD4   1      1  3   4  2
CQUAD4   2      3  5   6  4
CQUAD4   3      5  7   8  6
CQUAD4   4      7  9  10  8
CQUAD4   5      9 11  12 10
```

CTEMP	1		250	
CTEMP	2		250	
CFCONV	1	1	1.6	70.1
CFCONV	2	1	1.6	70.1
CFCONV	3	1	1.6	70.1
CFCONV	4	1	1.6	70.1
CFCONV	5	1	1.6	70.1
CFCONV	1	3	1.6	70.1
CFCONV	2	3	1.6	70.1
CFCONV	3	3	1.6	70.1
CFCONV	4	3	1.6	70.1
CFCONV	5	3	1.6	70.1

END

Programa Femev- Datos de entrada para o problema 4.8

(Peclet = 100)

* Problema 4.8 - Peclet = 100

SOL	1	0.01	1	0
PROP	0.01	0		
*CFCORR	100			
CFIP	1			
CDENS	1	1		
GRID	1	0.0	0.0d00	
GRID	2	0.0	0.1d00	
GRID	3	0.1	0.0d00	
GRID	4	0.1	0.1d00	
GRID	5	0.2	0.0d00	
GRID	6	0.2	0.1d00	
GRID	7	0.3	0.0d00	
GRID	8	0.3	0.1d00	
GRID	9	0.4	0.0d00	
GRID	10	0.4	0.1d00	
GRID	11	0.5	0.0d00	

GRID	12	0.5	0.1d00
GRID	13	0.6	0.0d00
GRID	14	0.6	0.1d00
GRID	15	0.7	0.0d00
GRID	16	0.7	0.1d00
GRID	17	0.8	0.0d00
GRID	18	0.8	0.1d00
GRID	19	0.9	0.0d00
GRID	20	0.9	0.1d00
GRID	21	1.0	0.0d00
GRID	22	1.0	0.1d00
CQUAD4	1	1 3 4 2	
CQUAD4	2	3 5 6 4	
CQUAD4	3	5 7 8 6	
CQUAD4	4	7 9 10 8	
CQUAD4	5	9 11 12 10	
CQUAD4	6	11 13 14 12	
CQUAD4	7	13 15 16 14	
CQUAD4	8	15 17 18 16	
CQUAD4	9	17 19 20 18	
CQUAD4	10	19 21 22 20	
CVX	1	1	
CVY	1	0	
CVX	2	1	
CVY	2	0	
CVX	3	1	
CVY	3	0	
CVX	4	1	
CVY	4	0	
CVX	5	1	
CVY	5	0	
CVX	6	1	
CVY	6	0	
CVX	7	1	
CVY	7	0	
CVX	8	1	
CVY	8	0	

CVX	9	1
CVY	9	0
CVX	10	1
CVY	10	0
CVX	11	1
CVY	11	0
CVX	12	1
CVY	12	0
CVX	13	1
CVY	13	0
CVX	14	1
CVY	14	0
CVX	15	1
CVY	15	0
CVX	16	1
CVY	16	0
CVX	17	1
CVY	17	0
CVX	18	1
CVY	18	0
CVX	19	1
CVY	19	0
CVX	20	1
CVY	20	0
CVX	21	1
CVY	21	0
CVX	22	1
CVY	22	0
GERACAO	1	4.1983538976954
GERACAO	2	4.1239866170936
GERACAO	3	3.7712301110756
GERACAO	4	3.1941920379066
GERACAO	5	2.4689342980711
GERACAO	6	1.6860275657938
GERACAO	7	0.94168660034396
GERACAO	8	0.32634717291719
GERACAO	9	-7.4372484347946D-02

```
GERACAO      10  -0.20747497200356
CTEMP        1      1
CTEMP        2      1
CTEMP       21      4
CTEMP       22      4
END
```

APÊNDICE D

LISTAGEM DO PROGRAMA FEMEY

```
PROGRAM FEMEY4
IMPLICIT REAL*8 (A-H,O-Z)
IMPLICIT INTEGER*2 (I-N)
INTEGER*4 NNOS,NELEM
COMMON /JELEM/ JELEM(4,102)
COMMON /COORD/ COORD(2,123)
COMMON /RE/ REC(123,123)
COMMON /F/ F(123),FAD
COMMON /PROP/ RO,PROP,B0,RC(8,4),RFC(4),WITER,PREC
COMMON /U/ UC(123)
COMMON /V/ VC(123)
COMMON /P/ PC(123)
```

```
*****
C* ESTE PROGRAMA VISA A SOLUCAO DE PROBLEMAS DE CONVECCAO-DIFUSAO
C* APLICACAO DO METODO DOS ELEMENTOS FINITOS POR VOLUMES DE CONTROLE
C* SAO UTILIZADOS ELEMENTOS ISOPARAMETRICOS DE QUATRO NOS.
C* ESTAO IMPLEMENTADAS CONDICoes DE BOUDS DO TIPO :
C* -CONVECCAO FORCADA, TEMPERATURA IMPOSTA EM UM NO, GERACAO INTERNA
C* DE CALOR E FLUXO DE CALOR IMPOSTO NUMA FACE.
C*
```

```
*****
```

C* Nessa versao:

```
C* NUMERO MAXIMO DE NOS: 122
C* NUMERO MAXIMO DE ELEMENTOS: 102
C* PARA UTILIZAR-SE UM NUMERO MAIOR DE NOS EH NECESSARIO
C* MODIFICAR-SE AS DIMENSOES DAS MATRIZES
```

```
*****
```

C DESCRICAO DAS VARIAVEIS :

C

C COORD : CONTEM COORD. X E Y

```

C JELEM      : CONECTIVIDADE ENTRE ELEMENTOS
C RE         : MATRIZ GLOBAL
C RR ,FI     : TERMOS DE PONDERACAO PARA A INTEGRACAO DE GAUSS
C COND       : CONDUTIVIDADE
C NELEM      : NUMERO TOTAL DE ELEMENTOS
C NNOS       : NUMERO TOTAL DE NOS
C DETJ       : DETERMINANTE DO JACOBIANO
C DNX,DNY    : FUNCAO DE INTERPOLACAO PARA A DERIVADA DO ELEMENTO
C            ISOPARAMETRICO DE QUATRO NOS .

```

```

C*****

```

```

      OPEN (7,FILE ='LIXO',STATUS ='UNKNOWN' )
      OPEN (8,FILE='FEMEVENT',STATUS ='OLD')
C -> UNIT 7 ,8 : LEITURA
      OPEN (9,FILE='FEMEVSAI',STATUS ='UNKNOWN' )
      OPEN (10,STATUS ='SCRATCH' )
      OPEN (11,STATUS ='SCRATCH' )
C -> UNIT 10,11: DETECCAO DE FRONTEIRAS
      OPEN (12,STATUS ='SCRATCH' )
C -> UNIT 12:      SOURCES
      OPEN (13,STATUS ='SCRATCH' )
C -> UNIT 13:      VAR. IMPOSTAS
      OPEN (14,STATUS ='SCRATCH' )
      OPEN (15,STATUS ='SCRATCH' )
C -> UNIT 15:      GERACAO DE CALOR NUM VOLUME
      OPEN (18,FILE = 'ELEMREP',STATUS ='OLD' )
      CALL DIMVAR(NELEM,NNOS)
      STOP
      END

```

```

C*****

```

```

      SUBROUTINE ZERA
      IMPLICIT REAL*8 (A-H,O-Z)
      IMPLICIT INTEGER*2 (I-N)
      COMMON /PROP/ RO,PROP,BO,RC(8,4),RFC(4),NITER,PREC
      DO 5 I =1 ,4
      RFC(I) = 0.0D00
      DO 5 J =1 ,4
5      RCI ,J) = 0.0D00

```

RETURN

END

C*****

SUBROUTINE LE (NELEM, NNOS, ISOL, IREP, ICF)

IMPLICIT REAL*8 (A-H, O-Z)

IMPLICIT INTEGER*2 (I-N)

INTEGER*4 NELEM, NNOS

REAL*8 TF

INTEGER*2 ELEM

CHARACTER*1 CARD*8, LINHA*80

COMMON /JELEM/ JELEM(4,102)

COMMON /COORD/ COORDC(2,123)

COMMON /RE/ REC(123,123)

COMMON /F/ FC(123), FAD

COMMON /PROP/ RO, PROP, BO, RC(8,4), RFC(4), WITER, PREC

COMMON /U/ UC(123)

COMMON /V/ VC(123)

COMMON /P/ PC(123)

C ESSA PROCEDURE FAZ A LEITURA DOS DADOS DE ENTRADA

C DESCRICAO DAS VARIAVEIS :

C NO : NUMERO DO NO

C F : ARQUIVO DE DADOS ASSUMIDO COMO DADOS.DAT

C TF : TEMPERATURA DO MEIO EXTERNO

C*****

REWINDC(12)

ICF = 0

IER = 0

15 READ (8,20,END=999) CARD, LINHA

20 FORMAT (A8,A72)

REWINDC(7)

WRITEC(7,25) LINHA

25 FORMAT(CA80)

IF (CARD .EQ. 'GRID ') THEN

IER = 1

REWINDC(7)

READ (7,*,ERR=990) NO, COORDC(1,NO), COORDC(2,NO)

```

      NNOS = NNOS+1
      ELSEIF ( CARD .EQ. 'SOL      ') THEN
        IER = 2
        REWIND(7)
        READ(7,*,ERR=990) ISOL,PREC,WITER,I REP
C
C - ISOL  -> TIPO DE SOLUCAO
C - PREC  -> PRECISAO DA SOLUCAO
C - WITER -> NUMERO MAXIMO DE ITERACOES
C - IREP  -> 1 LE ARQUIVO ELEMREP ; 0 NAO LE
C
      ELSEIF ( CARD .EQ. 'CVY      ') THEN
        IER = 3
        REWIND(7)
        READ(7,*,ERR=990) NO,UU
        VCNO) = UU
      ELSEIF ( CARD .EQ. 'CVX      ') THEN
        IER = 4
        REWIND(7)
        READ(7,*,ERR=990) NO,UU
        UCNO) = UU
      ELSEIF ( CARD .EQ. 'CTI      ') THEN
        IER = 5
        REWIND(7)
        READ(7,*,ERR=990) NO,UU
        PCNO) = UU
      ELSEIF ( CARD .EQ. 'CTEMP    ') THEN
        IER = 6
        REWIND(7)
        READ (7,*,ERR=990) NO,UU
        PCNO) = UU
        WRITE (13,*) NO,UU
      ELSEIF ( CARD .EQ. 'PROP     ') THEN
        IER = 7
        REWIND(7)
        READ(7,*,ERR=990) PROP,BO
        FAD = PROP

```

```

ELSEIF ( CARD .EQ. 'CDENS ' ) THEN
    IER = 8
    REWIND(7)
    READ(7,*,ERR=990) RO,CP
    RO = RO*CP
ELSEIF ( CARD .EQ. 'CQUAD4 ' ) THEN
    IER = 9
    REWIND(7)
    READ (7,*,ERR=990)ELEM,(JELEMCI,ELEM,I=1,4)
    NELEM = NELEM +1
ELSEIF ( CARD .EQ. 'CFCONV ' ) THEN
    IER = 10
    REWIND(7)
    READ (7,*,ERR=990)ILEM,IFACE,H,TF
    TF = H*TF
    WRITE(14,*)ILEM,IFACE,H,TF
ELSEIF ( CARD .EQ. 'CFQ ' ) THEN
    IER = 11
    REWIND(7)
    READ (7,*,ERR=990)ILEM,IFACE,TF
    H = 0.0000
    WRITE(14,*)ILEM,IFACE,H,TF
ELSEIF ( CARD .EQ. 'GERACAO ' ) THEN
    IER = 12
    IS = 1
    NE = 0
    REWIND(7)
    READ (7,*,ERR=990)ELEM,VALOR
    CALL GERACCELEM,VALOR,IS,NE
    WRITE(12,*) ELEM,VALOR
ELSEIF (CARD(1:1) .EQ. '*') THEN
    CONTINUE

ELSEIF (CARD .EQ. 'CFIP ' ) THEN
    IER = 13
    REWIND(7)
    READ(7,*,ERR=990)ICF

```

C

```

C DETERMINA O TIPO DE INTERPOLACAO USADA NAS FRONTEIRAS
  ELSEIF ( CARD .EQ. 'CFCORR ' ) THEN
    IER = 14
    REWIND(7)
    READ(7,*,ERR=990) FAD
  ELSEIF ( CARD .NE. 'END ' ) THEN
    WRITE(*,555) CARD
555 FORMAT (1X,' ERRO NOS CARTOES DE ENTRADA -> CARTAO: ',A8)
    STOP
  ELSEIF ( CARD .EQ. 'END ' ) THEN
    GO TO 950
  ELSEIF ( ISOL .LE. 0 .OR. ISOL .GT. 2 ) THEN
    WRITE(*,*) 'ERRO NO CARTAO DE SOLUCAO'
    STOP
  ENDIF
  GO TO 15
999 IER = 17
950 CONTINUE
C ARMAZENA VALORES DA FONTE NO ARQUIVO 15
  DO 980 I=1,NMOS
980 WRITE(15,*) FCID
    CLOSE(8)
    RETURN
990 IF ( IER .EQ. 1 ) THEN
    WRITE(*,*) 'ERRO NOS CARTOES: GRID '
  ELSEIF ( IER .EQ. 2 ) THEN
    WRITE(*,*) 'ERRO NOS CARTOES: SOL '
  ELSEIF ( IER .EQ. 3 ) THEN
    WRITE(*,*) 'ERRO NOS CARTOES: CVY '
  ELSEIF ( IER .EQ. 4 ) THEN
    WRITE(*,*) 'ERRO NOS CARTOES: CVX '
  ELSEIF ( IER .EQ. 5 ) THEN
    WRITE(*,*) 'ERRO NOS CARTOES: CTT '
  ELSEIF ( IER .EQ. 6 ) THEN
    WRITE(*,*) 'ERRO NOS CARTOES: CTEMP '
  ELSEIF ( IER .EQ. 7 ) THEN

```



```

WRITE(*,*) 'ERRO NOS CARTOES:  PROP '
ELSEIF ( IER .EQ. 8 ) THEN
WRITE(*,*) 'ERRO NOS CARTOES:  CDENS '
ELSEIF ( IER .EQ. 9 ) THEN
WRITE(*,*) 'ERRO NOS CARTOES:  CQUAD4 '
ELSEIF ( IER .EQ. 10 ) THEN
WRITE(*,*) 'ERRO NOS CARTOES:  CFCONV '
ELSEIF ( IER .EQ. 11 ) THEN
WRITE(*,*) 'ERRO NOS CARTOES:  CFQ '
ELSEIF ( IER .EQ. 12 ) THEN
WRITE(*,*) 'ERRO NOS CARTOES:  GERACAO '
ELSEIF ( IER .EQ. 13 ) THEN
WRITE(*,*) 'ERRO NOS CARTOES:  CFIP '
ELSEIF ( IER .EQ. 14 ) THEN
WRITE(*,*) 'ERRO NOS CARTOES:  CFCORR '
ENDIF
CONTINUE
IF (IER .NE. 0) STOP
RETURN
END

```

```

DOUBLE PRECISION FUNCTION FE(I ,NETA)
IMPLICIT REAL*8 (A-H,O-Z)
IMPLICIT INTEGER*2 (I-N)
C  ESSA PROCEDURE CALCULA AS DERIV DA FUNCAO DE INTERPOLACAO
C  EM RELACAO A EPSLON
REAL*8 NETA
INTEGER*2 I
GO TO (1,2,3,4) I
1 FE = (NETA -1.0D00) /4.0D00
RETURN
2 FE = (1.0D00 -NETA) /4.0D00
RETURN
3 FE = (1.0D00+NETA) /4.0D00
RETURN
4 FE = -(1.0D00+NETA) /4.0D00
RETURN

```

END

C*****

DOUBLE PRECISION FUNCTION FNCI ,EPS)
IMPLICIT REAL*8 (A-H,O-Z)
IMPLICIT INTEGER*2 (I-N)

C ESSA FUNCTION CALCULA AS DERIV DA FUNCAO DE INTERPOLACAO
C EM RELACAO A NETA "

REAL*8 EPS
GO TO (1,2,3,4) I

1 FN = (EPS-1.0D00)/4.0D00

RETURN

2 FN = -(1.0D00+EPS)/4.0D00

RETURN

3 FN = (1.0D00+EPS)/4.0D00

RETURN

4 FN = (1.0D00-EPS)/4.0D00

5 RETURN

END

C*****

DOUBLE PRECISION FUNCTION NCI ,EPS,NETA)
IMPLICIT REAL*8 (A-H,O-Z)
IMPLICIT INTEGER*2 (I-N)

C ESSA FUNCTION CALCULA OS VALORES DA FUNCAO DE INTERPOLACAO
C CQUADRILATERO ISOPARAMETRICO DE 4 NOS "

REAL*8 EPS,NETA
GO TO (1,2,3,4) I

1 N = (1.0D00-EPS)*(1.0D00-NETA)/4.0D00

RETURN

2 N = (1.0D00+EPS)*(1.0D00-NETA)/4.0D00

RETURN

3 N = (1.0D00+EPS)*(1.0D00+NETA)/4.0D00

RETURN

4 N = (1.0D00-EPS)*(1.0D00+NETA)/4.0D00

RETURN

END

C*****

```

DOUBLE PRECISION FUNCTION DISTC (X1,Y1,X2,Y2)
IMPLICIT REAL*8 (A-H,O-Z)
IMPLICIT INTEGER*2 (I-N)
DIST = SQRT ( (X1-X2)*(X1-X2)+ ( Y1-Y2)*(Y1-Y2) )
RETURN
END

```

C*****

```

FUNCTION CONDCI,ELEM )
IMPLICIT REAL*8 (A-H,O-Z)
IMPLICIT INTEGER*2 (I-N)
INTEGER*2      ELEM
REAL*8 EPS,NETA
COMMON /JELEM/ JELEM(4,102)
COMMON /COORD/ COORDC(2,123)
COMMON /RE/ RE(123,123)
COMMON /F/ FC(123),FAD
COMMON /PROP/ R0,PROP,BO,RCB,4),RFC(4),WITER,PREC
COMMON /U/ UC(123)
COMMON /V/ VC(123)
COMMON /P/ PC(123)

```

c N DEVE SER REAL NO CASO DO CALCULO DO CILINDRO GIRANTE!!

```

IF ( I.EQ. 1 ) THEN
    EPS = 0.0D00
    NETA = -0.5D00
ELSEIF ( I .EQ. 2) THEN
    EPS = 0.5D00
    NETA = 0.0D00

ELSEIF ( I.EQ. 3) THEN
    EPS = 0.0D00
    NETA = 0.5D00
ELSEIF ( I .EQ. 4) THEN
    EPS = -0.5D00
    NETA = 0.0D00
ENDIF

X=0
Y=0

```

C OS COMANDOS ABAIXO SAO USADOS APENAS PARA DEFINIR A CONDUTIVIDADE
 C PARA CILINDRO GIRANTE

```

c          DO 10 J =1 ,4
c          IH = ABS(JELEM(J,ELEM))
c          X  = X + NC(J, EPS, NETA) * COORDC(1, IH)
c10        Y  = Y + NC(J, EPS, NETA) * COORDC(2, IH)
c          COND = 1 / (X * X + Y * Y) / PROP
c==cond variavel =====
          NO = ABS(JELEM(I, ELEM))
          COND = PROP * (1.0D00 + BO * P(NO))
RETURN
END

```

C*****

```

SUBROUTINE LIMITECISOL, NNOS, ICF)
IMPLICIT REAL*8 (A-H, O-Z)
IMPLICIT INTEGER*2 (I-N)

INTEGER*4 NNOS
INTEGER*2 IFC(2)
REAL*8 N, EPS, NETA
COMMON /JELEM/ JELEM(4,102)
COMMON /COORD/ COORDC(2,123)
COMMON /RE/ REC(123,123)
COMMON /F/ FC(123), FAD
COMMON /PROP/ RO, PROP, BO, RC(8,4), RFC(4), WITER, PREC
COMMON /U/ UC(123)
COMMON /V/ VC(123)
COMMON /P/ PC(123)

```

```

C CALCULA FLUXOS NORMAIS
REWIND(14)
20 READ(14, *, END=25) ILEM, IFAC, H, VALOR
REWIND(11)
IER = 0
21 READ(11, *, END=24) NO1, NO2, IELEM, IFACE
NO1 = ABS(NO1)
NO2 = ABS(NO2)

```

C VERIFICA SE FACE IMPOSTA EH FRONTEIRA

```

        IF (ILEM .EQ. IELEM .AND. IFAC.EQ.IFACED) THEN
            IER =1
            DX = COORDX1,N02)-COORDX1,N01)
            DY = COORDX2,N02)-COORDX2,N01)
            DS = SQRT(DX*DX+ DY*DY)/2.0D00
C -> FLUXOS ESCALARES
C
            FC(N01) = VALOR*DS + FC(N01)
            FC(N02) = VALOR*DS + FC(N02)
C
            FT = 0.75D00
            REC(N01,N01) = H*DS *FT+ REC(N01,N01)
            REC(N01,N02) = H*DS *(1-FT) + REC(N01,N02)
            FT = 0.25D00
            REC(N02,N01) = H*DS *FT+ REC(N02,N01)
            REC(N02,N02) = H*DS *(1.0D00-FT) + REC(N02,N02)
C
            ENDIF
C TERMINA IF ESCALAR -----
            GOTO 21
24      CONTINUE
            IF ( IER .EQ. 0 ) THEN
                WRITE(*,*) ' ***** AVISO: ***** '
                WRITE(*,*) 'NO: ',NO,' POSSUI FLUXO ESCALAR IMPOSTO,MAS NAO EH
+ FRONTEIRA ,OU FACE INDICADA ESTA ERRADA'
            ENDIF
            GOTO 20
25      CONTINUE
C-----
C -> MONTA CONDIÇÕES IMPOSTAS:
            IF ( ISOL .LE. 2 ) THEN
                REWIND(13)
5          READ(13,*,END=10)NO, VALOR
                PCNO) =VALOR
                DO 8 J = 1,NNOS
8          RECNO,J) = 0.0D00

```

```

RECNO,NO) = 1.00102101D00
FCNO) = VALOR* 1.00102101D00
GOTO 5
10 CONTINUE
ENDIF

C-----
C VERIFICA FRONTEIRAS
REWINDC11)
120 READC11,*,END=@$O)NO1,NO2,IELEM,IFACE
C-----
C - EH UTILIZADO A MATRIZ R (POR ECONOMIA DE MEMORIA) PARA
C ARMAZENAR OS DADOS GEOMETRICOS QUE SERAO NECESSARIOS PARA
C OS CALCULOS
C- RC1,3) ,RC2,3) ARMAZENAM O VALOR DO CGX E CGY DO ELEMENTO
RC1,3) = 0.0D00
RC2,3) = 0.0D00
DO 15 I = 1,4
NO = ABS(CJELEMC1,IELEM)
RC1,1) = COORDC1,NO)
RC1,2) = COORDC2,NO)
RC1,3) =RC1,3) + RC1,1)
15 RC2,3) =RC2,3) + RC1,2)
RC1,3) =RC1,3)/4.0D00
RC2,3) =RC2,3)/4.0D00
DO 27 I = 1,4
IU = I+1
IF (IU .EQ. 5) IU = 1
RCI+4,1) = (RC1,1) + RCIU,1))/2.0D00
27 RCI+4,2) = (RC1,2) + RCIU,2))/2.0D00
C-----
C CADA FRONTEIRA EH COMPOSTA DE DOIS SUBVOLUMES DE CONTROLE
C ASSIM, II VARIA DE 1 A 2.
DO 40 II = 1,2
CONTINUE
IF ( II .EQ. 2 ) THEN
NO = NO1

```

```
NO1 = NO2
```

```
NO2 = NO
```

```
ENDIF
```

```
C
```

```
IF ( REC(ABS(NO1),ABS(NO1)) .NE. 1.00102101D00 ) THEN  
IF ( IFACE .EQ. 1 ) THEN
```

```
C
```

```
IF ( II .EQ. 1 ) THEN
```

```
IPC1) = 4
```

```
IPC2) = 1
```

```
ELSE
```

```
IPC1) = 1
```

```
IPC2) = 2
```

```
ENDIF
```

```
ELSEIF ( IFACE .EQ. 2 ) THEN
```

```
IF ( II .EQ. 1 ) THEN
```

```
IPC1) = 1
```

```
IPC2) = 2
```

```
ELSE
```

```
IPC1) = 2
```

```
IPC2) = 3
```

```
ENDIF
```

```
ELSEIF ( IFACE .EQ. 3 ) THEN
```

```
IF ( II .EQ. 1 ) THEN
```

```
IPC1) = 2
```

```
IPC2) = 3
```

```
ELSE
```

```
IPC1) = 3
```

```
IPC2) = 4
```

```
ENDIF
```

```
ELSEIF ( IFACE .EQ. 4 ) THEN
```

```
IF ( II .EQ. 1 ) THEN
```

```
IPC1) = 3
```

```
IPC2) = 4
```

```
ELSE
```

```
IPC1) = 4
```

```
IPC2) = 1
```

ENDIF

ENDIF

C-----

C CALCULO DO FLUXO NA FRONTEIRA EXTERNA

NO = IFACE +1

IF (IFACE.EQ. 4) NO = 1

DX1 = (RCNO,1)-RC(IFACE,1))/2.0D00

DY1 = (RCNO,2)-RC(IFACE,2))/2.0D00

UM = UC(ABSC(NO1))*0.75D00 + UC(ABSC(NO2))*0.25D00

VM = VC(ABSC(NO1))*0.75D00 + VC(ABSC(NO2))*0.25D00

XMPT = RO*(UM*DY1-VM*DX1)

C

IF (ISOL.EQ. 1) THEN

C TERMO PARA A VARIÁVEL ESCALAR

REC(ABSC(NO1),ABSC(NO1)) = REC(ABSC(NO1),ABSC(NO1)) +XMPT*0.75D00

REC(ABSC(NO1),ABSC(NO2)) = REC(ABSC(NO1),ABSC(NO2)) +XMPT*0.25D00

C

ENDIF

C-----

C CALCULO DO FLUXO NO INTERIOR DO VOLUME DE CONTROLE

C SITUADO NUMA FRONTEIRA

C CADA SUB VOLUME EH COMPOSTO DE DOIS SEGMENTOS INTERNOS

C ,ASSIM JJ VARIA DE 1 A 2.

DO 55 JJ -1,2

IF (IPCJJ).EQ. 1) THEN

EPS = 0.0D00

NETA = -0.5D00

ELSEIF (IPCJJ).EQ. 2) THEN

EPS = 0.5D00

NETA = 0.0D00

ELSEIF (IPCJJ).EQ. 3) THEN

EPS = 0.0D00

NETA = 0.5D00

ELSEIF (IPCJJ).EQ. 4) THEN

EPS = -0.5D00

NETA = 0.0D00

ENDIF


```

        UM = 0.0DOO
        VM = 0.0DOO
DO 45 I = 1 ,4
        NO  = ABS(JELEMCI,IELEM)
        AUX  = NCI, EPS, NETA)
        UM   = AUX*UCNO) + UM
45      VM   = AUX*VCNO) + VM
        NO   = IPCJJ) + 4
        DX1  = RCNO,1) - RC1,3)
        DY1  = RCNO,2) - RC2,3)
C      NECESSARIO INVERTER SINAL, POIS SENTIDO EH TOMADO SEMPRE
C      COMO ANTI-HORARIO
        IF ( JJ .EQ. 2 ) THEN
            DX1 = -DX1
            DY1 = -DY1
        ENDIF
C
        IF ( NO1 .LT. 0.0DOO) THEN
            DX1 = DX1/2.0DOO
            DY1 = DY1 /2.0DOO
        ENDIF
C
        XMPT= RC*UM*DY1 -VM*DX1 )
C=====
        IF (ISOL .EQ. 1 .AND. ICF .EQ. 0 ) THEN
            DO 50 I = 1,4
                NO = ABS(JELEMCI,IELEM)
50      REC(ABSCNO1),NO) = REC(ABSCNO1),NO) + XMPT*NCI, EPS, NETA)
            ENDIF
C=====
65      CONTINUE
C-----
            ENDIF
C-> TERMINA IF RECNO1,NO1 )
40      CONTINUE
        GOTO 120
950     CONTINUE

```

C-----

RETURN

END

C *****

SUBROUTINE MINVCA,ELEM

IMPLICIT REAL*8 (A-H,O-Z)

IMPLICIT INTEGER*2 (I-N)

INTEGER*2 ELEM

REAL*8 AC(4,4),AINV(4,4)

DET = 1.0D00

DO 1 I=1,4

DO 1 J=1,4

IF (I.EQ. J) THEN

AINV(I,J) = 1.0D00

ELSE

AINV(I,J) = 0.0D00

ENDIF

1 CONTINUE

DO 9 IPASS =1,4

IMX = IPASS

DO 2 IROW = IPASS,4

IF (ABS(AC(IROW,IPASS)) .GT. ABS(AC(IMX,IPASS))) THEN

IMX = IROW

ENDIF

2 CONTINUE

IF (IMX .NE. IPASS) THEN

DO 3 ICOL =1,4

TEMP = AINV(IPASS,ICOL)

AINV(IPASS,ICOL) = AINV(IMX,ICOL)

AINV(IMX,ICOL) = TEMP

IF (ICOL .GE. IPASS) THEN

TEMP = AC(IPASS,ICOL)

AC(IPASS,ICOL) = AC(IMX,ICOL)

AC(IMX,ICOL) = TEMP

ENDIF

3 CONTINUE

ENDIF

```

PIVOT = ACIPASS,IPASS)
DET = DET*PIVOT
IF (DET .EQ. 0.0000) THEN
  DO 15 I =1 ,4
  DO 15 J=1 ,4
15  AINVCI,J) = 0.0000
  WRITE(*,*) 'ERRO NA EXECUCAO NAO HOUE INVERSA '
  WRITE(*,*) 'ELEMENTO: ',ELEM
  WRITE(*,*) 'PROSSEGUINDO EXECUCAO'
  WRITE(*,*)
  RETURN
  ENDIF
  DO 6 ICOL =1,4
    AINVCIPASS,ICOL) = AINVCIPASS,ICOL)/PIVOT
    IF (ICOL .GE. IPASS) THEN
      ACIPASS,ICOL) = ACIPASS,ICOL)/PIVOT
    ENDIF
6    CONTINUE
    DO 8 IROW = 1,4
  IF (IROW .NE. IPASS) THEN
    FACTOR = ACIROW,IPASS)
    ENDIF
    DO 7 ICOL = 1,4
  IF (IROW .NE. IPASS) THEN
    AINVCIROW,ICOL) = AINVCIROW,ICOL) -
+ FACTOR*AINVCIPASS,ICOL)
    ACIROW,ICOL) = ACIROW,ICOL) -FACTOR*ACIPASS,ICOL)
    ENDIF
7    CONTINUE
8    CONTINUE
9    CONTINUE
    DO 20 I = 1,4
    DO 20 J = 1,4
20  ACI,J) = AINVCI,J)
    RETURN
    END

```

C*****

```

SUBROUTINE GERACCELEM,Q,I,S,NE)
IMPLICIT REAL*8 (A-H,O-Z)
IMPLICIT INTEGER*2 (I-N)
REAL*8      AUX,RR(2),FX(2),FY(2)
+          ,EPS,NETA,Q,X(4),Y(4),J11,J12,J21,J22
INTEGER*2  ELEM
COMMON /JELEM/ JELEM(4,102)
COMMON /COORD/ COORDX(2,123)

COMMON /RE/ REC(123,123)
COMMON /F/ F(123),FAD
COMMON /PROP/ RO,PROP,BO,RC(8,4),RFC(4),WITER,PREC
C  ESSA SUBROTINA CALCULA OS VALORES CORRESPONTES
C  PARA GERACAO INTERNA DE UM ELEMENTO.
C  PARAMETROS PARA INTEGRACAO DE GAUSS PARA N = 4
      RR(1) = 0.3478548451D00
      RR(2) = 0.6521451549D00
      C      = 0.8511353116D00
      D      = 0.3333310435D00
      DO 40 IVOL = 1 , 4
          AUX = 0.0D00
C  DETERMINA PARAMETROS PARA INTEGRAL DE CADA VOLUME DE CONTROLE
      IF (IVOL .EQ. 1) THEN
C  INTEGRACAO ENTRE:      -1<X<0   E   -1<Y<0
          FX(1) = -C
          FX(2) = -D
          FY(1) = -C
          FY(2) = -D
          ELSEIF (IVOL .EQ. 2) THEN
C  INTEGRACAO ENTRE:      0<X<1   E   -1<Y<0
          FX(1) = C
          FX(2) = D
          FY(1) = -C
          FY(2) = -D
          ELSEIF (IVOL .EQ. 3) THEN
C  INTEGRACAO ENTRE:      0<X<1   E   0<Y<1
          FX(1) = C

```

```

FX(2) = D
FY(1) = C
FY(2) = D
      ELSEIF (IVOL .EQ. 4 ) THEN
C INTEGRACAO ENTRE:   -1<X<0   E   0<Y<1
      FX(1) = -C
      FX(2) = -D
      FY(1) = C
      FY(2) = D
      ENDIF
C -> I, J VARIAVEIS PARA INTEGR. GAUSS EM CADA VOL. DE CONTROLE
      DO 30 I = 1 , 2
      DO 30 J = 1 , 2
      EPS = FX(I)
      NETA = FY(J)
      J11 = 0.0D00
      J12 = 0.0D00
      J22 = 0.0D00
      J21 = 0.0D00
      DO 20 IA = 1,4
      XCIA) = FECIA,NETA)
      YCIA) = FNCIA,EPS)
      JE = JELEMCIA,ELEM)
      J11 = XCIA) * COORDX1,JED) + J11
      J12 = XCIA) * COORDX2,JED) + J12
      J21 = YCIA) * COORDX1,JED) + J21
20      J22 = YCIA) * COORDX2,JED) + J22
      DETJ = J11*J22 - J12*J21
      IF (DETJ .EQ. 0.0D00) THEN
      WRITE(*,*) 'DET. NULO, ERRO EM DADOS DE ENTRADA -> VERIFIQUE NOS
+ OU CONECTIVIDADE'
      WRITE(*,*) 'ELEMENTO: ',ELEM
      STOP
      ENDIF
30      AUX = Q* ABS(DETJ)* RRC(I)* RRC(J) + AUX
      JE = JELEMCIVOL,ELEM)*IS - NE
      FC(JE) = FC(JE) + AUX

```

```

40      CONTINUE
      RETURN
      END

```

```

C*****

```

```

      SUBROUTINE JACOB4 ( ELEM, EPS, NETA, DETJ, DNX, DNY)
      IMPLICIT REAL*8 (A-H, O-Z)
      IMPLICIT INTEGER*2 (I-N)
C  ESSA PROCEDURE CALCULA O JACOBIANO DE UM DADO ELEMNTO "
C  DESCRICAO DAS VARIAVEIS :
C  J11, J12, J21, J22 : ELEMENTOS DA MATRIZ DO JACOBIANO
C  X ,Y ,I, II, JJ : VARIAVEIS AUXILIARES
C
      REAL*8 DETJ, DNX(4)
+ , DNY(4), EPS, NETA, XC(4), YC(4), J11, J12, J21, J22, FE, FN
      INTEGER*2 ELEM
      COMMON /JELEM/ JELEM(4,102)
      COMMON /COORD/ COORDX(2,123)
      COMMON /RE/ REC(23,123)
      COMMON /F/ F(123), FAD
      COMMON /PROP/ RO, PROP, BO, RC(8, 4), RFC(4), WITER, PREC
      DO 15 I = 1, 4
          XC(I) = FE*CI, NETA)
15      YC(I) = FW*CI, EPS)
          J11 = 0.0D00
          J12 = 0.0D00
          J22 = 0.0D00
          J21 = 0.0D00
      DO 20 I = 1, 4
          J = ABS(JELEM(I, ELEM))
          J11 = XC(I) * COORDX(1, J) + J11
          J12 = XC(I) * COORDX(2, J) + J12
          J21 = YC(I) * COORDX(1, J) + J21
20      J22 = YC(I) * COORDX(2, J) + J22
          DETJ = J11*J22 - J12*J21
          IF (DETJ .EQ. 0.0D00) THEN
              WRITE(*,*) 'DET. NULO, ERRO EM DADOS DE ENTRADA -> VERIFIQUE NOS
+ OU CONECTIVIDADE'

```

```

WRITE(*,*) 'ELEMENTO: ', ELEM
      STOP
      ENDIF
      DO 30 I = 1 , 4
          DN $X$ (I) = (J22* $X$ (I) - J12* $Y$ (I))/DETJ
30      DN $Y$ (I) = (-J21* $X$ (I) + J11* $Y$ (I))/DETJ
      RETURN
      END
C *****
      SUBROUTINE IPCONV(ELEM,UIP,VIP,X,Y,AA,BB)
      IMPLICIT REAL*8 (A-H,O-Z)
      IMPLICIT INTEGER*2 (I-N)
      INTEGER*4 NELEM,NNOS
      INTEGER*2 ELEM
      REAL*8 UIP(4),VIP(4),EPS,NETA,N,
+ XC(13),YC(13),MO,EB(4,4),AA(4,4),A
      COMMON /JELEM/ JELEM(4,102)
      COMMON /COORD/ COORD(2,123)
      COMMON /RE/ REC(123,123)
      COMMON /F/ FC(123),FAD
      COMMON /PROP/ RO,PROP,BO,RC(8,4),RFC(4),WITER,PREC
      COMMON /U/ UC(123)
      COMMON /V/ VC(123)
      COMMON /P/ PC(123)
      IER = 0
      IRETA = 0
      PI = 3.141592653589793
      DO 1000 IP = 1,4
          IF (IP .EQ. 1) THEN
              EPS = 0.0D00
              NETA = -0.5D00
              XIP = XC(9)
              YIP = YC(9)
          ELSEIF (IP .EQ. 2) THEN
              EPS = 0.5D00
              NETA = 0.0D00
              XIP = XC(10)

```

```

        YIP = Y(10)
ELSEIF (CIP .EQ. 3) THEN
        EPS = 0.0D00
        META = 0.5D00
        XIP = X(11)
        YIP = Y(11)
ELSEIF (CIP .EQ. 4) THEN
        EPS = -0.5D00
        META = 0.0D00
        XIP = X(12)
        YIP = Y(12)
ENDIF

UIPCIP) = 0.0D00
VIPCIPI) = 0.0D00
DO 17 I = 1,4
        NO = ABS(JELEM(I),ELEM)
        A = NCI, EPS, META)
        UIPCIPI) = A*UCNO) + UIPCIPI)
17      VIPCIP) = A*V(CNO) + VIPCIP)
C-----
        VR = SQRT(UIPCIP)*UIPCIP + VIPCIP)*VIPCIP)
        IF ( VR .EQ. 0.0D00) GOTO 1000
C-----
C   CALCULO DE MO:
        AX = UIPCIPI)
        AY = VIPCIP)
        BX = 1.0D00
        BY = 0.0D00
        AMOD = SQRT(AX*AX + AY*AY)
        BMOD = SQRT(BX*BX + BY*BY)
        AXB = AX*BX + AY*BY
        MO = ACOS(AXB/AMOD/BMOD)
        IF (( UIPCIPI) .LT. 0.0D00 .AND. VIPCIP).LT.0.0D00) .OR.
+ ( VIPCIP) .LT. 0.0D00 .AND. UIPCIPI) .GE. 0.0D00)
+ MO = -MO
C
        IF (MO.LT. 0) MO = 2*PI+ MO

```


C CALCULO DOS ANGULOS DE VISAO DOS PONTOS DE INTEGRACAO:

```
IRETA = 0
DO 45 I =1,6
IF ( IP .EQ. 1 ) THEN
IF ( CI .EQ. 1 ) THEN
NO1 = 1
NO2 = 2
IRETA = 5
ELSEIF ( CI .EQ. 2 ) THEN
NO1 = 2
NO2 = 6
IRETA = 6
ELSEIF ( CI .EQ. 3 ) THEN
NO1 = 6
NO2 = 10
IRETA = 2
ELSEIF ( CI .EQ. 4 ) THEN
NO1 = 10
NO2 = 12
IRETA = 9
ELSEIF ( CI .EQ. 5 ) THEN
NO1 = 12
NO2 = 8
IRETA = -4
ELSEIF ( CI .EQ. 6 ) THEN
NO1 = 8
NO2 = 1
IRETA = -8
ENDIF
```

C---

```
ELSEIF ( IP .EQ. 2 ) THEN
IF ( CI .EQ. 1 ) THEN
NO1 = 5
NO2 = 2
IRETA = 5
ELSEIF ( CI .EQ. 2 ) THEN
NO1 = 2
```

```

        NO2 = 3
        IRETA = 6
ELSEIF (I .EQ. 3) THEN
        NO1 = 3
        NO2 = 7
        IRETA = 7
ELSEIF (I .EQ. 4) THEN
        NO1 = 7
        NO2 = 11
        IRETA = 3
ELSEIF (I .EQ. 5) THEN
        NO1 = 11
        NO2 = 9
        IRETA = -10
ELSEIF (I .EQ. 6) THEN
        NO1 = 9
        NO2 = 5
        IRETA = -1
ENDIF

```

C --

```

ELSEIF (IP .EQ. 3) THEN
        IF (CI .EQ. 1) THEN
                NO1 = 12
                NO2 = 10
                IRETA = -9
        ELSEIF (CI .EQ. 2) THEN
                NO1 = 10
                NO2 = 5
                IRETA = -2
        ELSEIF (CI .EQ. 3) THEN
                NO1 = 5
                NO2 = 3
                IRETA = 6
        ELSEIF (CI .EQ. 4) THEN
                NO1 = 3
                NO2 = 4
                IRETA = 7

```

```
ELSEIF CI .EQ. 5) THEN
```

```
    NO1 = 4
```

```
    NO2 = 8
```

```
    IRETA = 8
```

```
ELSEIF CI .EQ. 6) THEN
```

```
    NO1 = 8
```

```
    NO2 = 12
```

```
    IRETA = 4
```

```
ENDIF
```

C---

```
ELSEIF ( IP .EQ. 4 ) THEN
```

```
    IF CI .EQ. 1) THEN
```

```
        NO1 = 1
```

```
        NO2 = 5
```

```
        IRETA = 5
```

```
    ELSEIF CI .EQ. 2) THEN
```

```
        NO1 = 5
```

```
        NO2 = 9
```

```
        IRETA = 1
```

```
    ELSEIF CI .EQ. 3) THEN
```

```
        NO1 = 9
```

```
        NO2 = 11
```

```
        IRETA = 10
```

```
    ELSEIF CI .EQ. 4) THEN
```

```
        NO1 = 11
```

```
        NO2 = 7
```

```
        IRETA = -3
```

```
    ELSEIF CI .EQ. 5) THEN
```

```
        NO1 = 7
```

```
        NO2 = 4
```

```
        IRETA = 7
```

```
    ELSEIF CI .EQ. 6) THEN
```

```
        NO1 = 4
```

```
        NO2 = 1
```

```
        IRETA = -8
```

```
ENDIF
```

```
ENDIF
```

C-----

```
AX = XCNO1) - XIP
AY = YCNO1) - YIP
BX = 1.0D00
BY = 0.0D00
AMOD = SQRT(CAX*AX + AY*AY)
BMOD = SQRT(CBX*BX + BY*BY)
AXB = AX*BX + AY*BY
TETA1 = ACOSC(AXB/AMOD/BMOD)
      IF ((XCNO1) .LT. XIP .AND. YCNO1) .LT. YIP)
+ .OR. (YCNO1) .LT. YIP .AND. XCNO1) .GT. XIP))
+ TETA1 = -TETA1
```

C---

```
AX = XCNO2) - XIP
AY = YCNO2) - YIP
BX = 1.0D00
BY = 0.0D00
AMOD = SQRT(CAX**2.0D00 + AY**2.0D00)
BMOD = SQRT(CBX**2.0D00 + BY**2.0D00)
AXB = AX*BX + AY*BY
TETA2 = ACOSC(AXB/AMOD/BMOD)
      IF ((XCNO2) .LT. XIP .AND. YCNO2) .LT. YIP)
+ .OR. (YCNO2) .LT. YIP .AND. XCNO2) .GT. XIP))
+ TETA2 = -TETA2
```

C-----

```
A = MO
      IF ( TETA2 .LT. 0.0D00 ) TETA2 = 2*PI+TETA2
      IF ( TETA1 .LT. 0.0D00 ) TETA1 = 2*PI+TETA1
      A = MO -TETA1
      TETA2 = TETA2 - TETA1
      TETA1 = 0.0D00

      IF ( TETA2 .LT. 0.0D00) TETA2 = 2.0D00*PI + TETA2
      IF ( A .LT. 0.0D00) A = 2.0D00*PI + A
      IF ( A .LE. TETA2 .and. A .ge. TETA1) GOTO 50
      IRETA = 0
```

C---

45 CONTINUE

C-----

```
50 IF (IRETA .EQ. 0 ) THEN
    WRITE(*,*) 'DETECTADO ERRO NO PROCESSAMENTO - SUB IPCONV'
    WRITE(*,*) 'NAO ENCONTRADA A INTERCECCAO DA RETA '
    STOP
ENDIF
```

C---

```
IF (IRETA .LT. 0.0000 ) THEN
    NO = NO2
    NO2 = NO1
    NO1 = NO
    IRETA = -IRETA
ENDIF
```

C

```
IF ( IRETA .EQ. 5 ) THEN
    NO1 = 1
    NO2 = 2
ELSEIF ( IRETA .EQ. 6 ) THEN
    NO1 = 2
    NO2 = 3
ELSEIF ( IRETA .EQ. 7 ) THEN
    NO1 = 3
    NO2 = 4
ELSEIF ( IRETA .EQ. 8 ) THEN
    NO1 = 1
    NO2 = 4
ENDIF
X1 = X(NO1)
Y1 = Y(NO1)
X2 = X(NO2)
Y2 = Y(NO2)
XLAMB = ( XIP*VIP(CIP) - YIP*UIP(CIP) +UIP(CIP)*Y1 - VIP(CIP)*X1)
XLAMB = XLAMB/(VIP(CIP)*(X2-X1) - UIP(CIP)*(Y2-Y1))
XU = X1+XLAMB*(X2 -X1)
YU = Y1+XLAMB*(Y2 -Y1)
```

100

```
A = DIST ( XU,YU,X1,Y1)
B = DIST ( X1,Y1,X2,Y2)
XC = DIST ( XU,YU,XIP,YIP)
```

C-----

```
IF (XC .EQ. 0.0000) THEN
WRITE(*,*) 'ERRO DE PROCESSAMENTO EM IPCONV: '
WRITE(*,*) 'PONTO DE INTERSECCAO INCORRETO '
STOP
ENDIF
ROVRXC = -RO*VR/XC
AUX = (1.0000-A/B)
AACIP,IP) = ROVRXC + AACIP,IP)
IF (IRETA .EQ. 1) THEN
BBCIP,1) = AUX*ROVRXC/2.0000 + BBCIP,1)
BBCIP,2) = AUX*ROVRXC/2.0000 + BBCIP,2)
AACIP,1) = -A/B*ROVRXC + AACIP,1)
ELSEIF (IRETA .EQ. 2) THEN
BBCIP,2) = AUX*ROVRXC/2.0000 + BBCIP,2)
BBCIP,3) = AUX*ROVRXC/2.0000 + BBCIP,3)
AACIP,2) = -A/B*ROVRXC + AACIP,2)
ELSEIF (IRETA .EQ. 3) THEN
BBCIP,4) = AUX*ROVRXC/2.0000 + BBCIP,4)
BBCIP,3) = AUX*ROVRXC/2.0000 + BBCIP,3)
AACIP,3) = -A/B*ROVRXC + AACIP,3)
ELSEIF (IRETA .EQ. 4) THEN
BBCIP,4) = AUX*ROVRXC/2.0000 + BBCIP,4)
BBCIP,1) = AUX*ROVRXC/2.0000 + BBCIP,1)
AACIP,4) = -A/B*ROVRXC + AACIP,4)
ELSEIF (IRETA .EQ. 5) THEN
BBCIP,2) = A/B*ROVRXC + BBCIP,2)
BBCIP,1) = AUX*ROVRXC + BBCIP,1)
ELSEIF (IRETA .EQ. 6) THEN
BBCIP,3) = A/B*ROVRXC + BBCIP,3)
BBCIP,2) = AUX*ROVRXC + BBCIP,2)
ELSEIF (IRETA .EQ. 7) THEN
BBCIP,4) = A/B*ROVRXC + BBCIP,4)
BBCIP,3) = AUX*ROVRXC + BBCIP,3)
```

```

ELSEIF ( IRETA .EQ. 8 ) THEN
BBCIP,4) = A/B*ROVRXC + BBCIP,4)
BBCIP,1) = AUX*ROVRXC + BBCIP,1)
ELSEIF ( IRETA .EQ. 9 ) THEN
AACIP,4) = -A/B*ROVRXC + AACIP,4)
AACIP,2) = -AUX*ROVRXC + AACIP,2)
ELSEIF ( IRETA .EQ. 10 ) THEN
AACIP,3) = -A/B*ROVRXC + AACIP,3)
AACIP,1) = -AUX*ROVRXC + AACIP,1)
ENDIF
1000 CONTINUE
RETURN
END

```

C*****

```

SUBROUTINE IPDIFFC ELEM ,X,Y,AA,BB)
IMPLICIT REAL*8 (A-H,O-Z)
IMPLICIT INTEGER*2 (I-N)
INTEGER*4 NELEM,NNOS
REAL*8 EPS,NETA,X(13),Y(13),N
+ ,AAC(4,4),BBC(4,4)
INTEGER*2 ELEM
COMMON /JELE%/ JELEM(4,102)
COMMON /COORD/ COORD(2,123)
COMMON /RE/ REC(123,123)
COMMON /F/ FC(123),FAD
COMMON /PROP/ RO,PROP,BO,RC(8,4),RFC(4),WITER,PREC
DO 1000 IP = 1,4
IF ( IP .EQ. 1 ) THEN
EPS = 0.0D00
NETA = -0.5D00
ELSEIF ( IP .EQ. 2 ) THEN
EPS = 0.5D00
NETA = 0.0D00
ELSEIF ( IP .EQ. 3 ) THEN
EPS = 0.0D00
NETA = 0.5D00

```

```

ELSEIF (IP .EQ. 4) THEN
    EPS = -0.5D00
    NETA = 0.0D00
ENDIF
CALL JACOB4CELEM, EPS, NETA, DETJ, DNX, DNY)
IF (CIP .EQ. 1 .OR. IP.EQ.3) THEN
    DY = Y(5)-Y(13)
    DX = X(6)-X(13)
    XLDX = 0.5D00*DX*DX
    XLDY = 1.0D00/8.0D00*DY*DY
ELSE
    DY = Y(7)-Y(13)
    DX = X(8)-X(13)
    XLDY = 0.50D00*DY*DY
    XLDX = 1.0D00/8.0D00*DX*DX
ENDIF
AACIP, IP) = (1.0D00/XLDY +1.0D00/XLDX)*FAD
DO 10 J=1,4
10    BBCIP, J) = NCJ, EPS, NETA) * (1.0D00/XLDX +1.0D00/XLDY) *FAD
1000    CONTINUE
RETURN
END

```

C*****

```

SUBROUTINE PRODUT(AA, BB, JM)
    REAL*8 AAC(4,4), BBC(4,4), CCC(4,4)
    DO 15 K=1, JM
    DO 15 I=1, 4
        CCC(I,K) = 0.0D00
    DO 10 J=1, 4
10    CCC(I,K) = AAC(I,J)*BBC(J,K) + CCC(I,K)
15    CONTINUE
    DO 20 I=1, 4
    DO 20 J=1, JM
20    BBC(I,J) = CCC(I,J)
RETURN
END

```

C*****


```

SUBROUTINE MCONV CELEND
  IMPLICIT REAL*8 (A-H,O-Z)
  IMPLICIT INTEGER*2 (I-NO)
  INTEGER*4      NELEM,NNOS
  REAL*8  X(13),Y(13)
+   ,UIP(4),VIP(4)
+   ,AA(4,4),BB(4,4),SS(4,4)
  INTEGER*2 ELEM
  COMMON /JELEM/ JELEM(4,102)
  COMMON /COORD/ COORD(2,123)
  COMMON /RE/ RE(123,123)
  COMMON /F/ FC(123),FAD
  COMMON /PROP/ RO,PROP,BO,RC(8,4),RFC(4),WITER,PREC
  COMMON /U/ UC(123)
  COMMON /V/ VC(123)
  COMMON /P/ PC(123)
  DO 10 I =1,4
  DO 10 J=1,4
    AAK(I,J)= 0.0D00
10    BBC(I,J)= 0.0D00
    XC(13) = 0.0D00
    YC(13) = 0.0D00
  DO 15 I = 1,4
    NO = ABS(JELEM(I,ELEM))
    X(I) = COORD(1,NO)
    Y(I) = COORD(2,NO)
    XC(13) = XC(13) + X(I)
15    YC(13) = YC(13) + Y(I)
    XC(13) = XC(13)/4.0D00
    YC(13) = YC(13)/4.0D00
  DO 25 I = 1,4
    IU = I+1
    IF (IU .EQ. 5) IU = 1
    X(I+4) = (X(I) + X(IU))/2.0D00
    Y(I+4) = (Y(I) + Y(IU))/2.0D00
    X(I+8) = (XC(13) + X(I+4))/2.0D00
25    Y(I+8) = (YC(13) + Y(I+4))/2.0D00

```

C

```
CALL IPDIFF ( ELEM ,X,Y,AA,BB)
CALL IPCONVCELEM,UIP,VIP,X,Y,AA,BB)
CALL MINVCAA,ELEM)
CALL PRODUT (AA,BB,4)
DO 30 I =1,4
30   SSC1,I) = 0.0000
      REWINDX12)
74   READX12,*,END=75)ILEM,VALOR
      IF (ILEM .EQ. ELEM ) THEN
          DO 78 IP =1,4
78     SSCIP,I) = VALOR
      ENDIF
      GOTO 74
78   CONTINUE
      CALL PRODUT(AA,SS,I)
```

C MONTAGEM DOS VOLUMES DE CONTROLE

C-----
C -> VOLUMES 1 E 4 E 2
C-----

```
DX1 = X(8)-X(12)
DY1 = Y(8)-Y(12)
DX2 = X(13)-X(5)
DY2 = Y(13)-Y(5)
AUX1 = RO*(UIP(4)*DY1-VIP(4)*DX1)
AUX2 = RO*(UIP(1)*DY2-VIP(1)*DX2)
```

C

```
DO 110 II =1,4
  IF (JELEMC1,ELEM.GT.0.0000) RC1,II) = AUX1*BBC4,II)
+   +AUX2*BBC1,II)+RC1,II)
  IF(JELEMC2,ELEM.GT.0.0000) RC2,II)=-AUX2*BBC1,II)+ RC2,II)
  IF(JELEMC4,ELEM.GT.0.0000) RC4,II)=-AUX1*BBC4,II)+ RC4,II)
110  CONTINUE
  IF (JELEMC1,ELEM.GT.0.0000) RFC1) = -SSC1,I)*AUX2
+   -SSC4,I)*AUX1 + RFC1)
C
  IF (JELEMC4,ELEM.GT.0.0000) RFC4) = -SSC4,I)*(C-AUX1)+RFC4)
```

```

      IF (JELEM(2,ELEM).GT.0.OD00) R(2) = -SS(1,1)*(-AUX2)+R(2)
C
C-----
C -> VOLUMES 2 E 3
C-----
      DX1 = X(13)-X(6)
      DY1 = Y(13)-Y(6)
      AUX1 = RO*(UIP(2)*DY1-VIP(2)*DX1)
DO 120 II =1,4
      IF(JELEM(2,ELEM).GT.0.OD00) R(2,II) = AUX1*BB(2,II)+R(2,II)
      IF(JELEM(3,ELEM).GT.0.OD00) R(3,II) = -AUX1*BB(2,II)+R(3,II)
120   CONTINUE
      IF (JELEM(2,ELEM).GT.0.OD00) R(2) = -SS(2,1)*AUX1 +R(2)
      IF (JELEM(3,ELEM).GT.0.OD00) R(3) = -SS(2,1)*(-AUX1)+R(3)
C-----
C -> VOLUMES 3 E 4
C-----
      DX1 = X(13)-X(7)
      DY1 = Y(13)-Y(7)
      AUX1 = RO*(UIP(3)*DY1-VIP(3)*DX1)
C
DO 130 II =1,4
      IF(JELEM(3,ELEM).GT.0.OD00) R(3,II) = AUX1*BB(3,II)+R(3,II)
      IF(JELEM(4,ELEM).GT.0.OD00) R(4,II) = -AUX1*BB(3,II)+R(4,II)
130   CONTINUE
      IF (JELEM(3,ELEM).GT.0.OD00) R(3) = -SS(3,1)*AUX1 +R(3)
      IF (JELEM(4,ELEM).GT.0.OD00) R(4) = -SS(3,1)*(-AUX1)+R(4)
C
      RETURN
      END
C*****
      SUBROUTINE MDIFF (ELEM)
      IMPLICIT REAL*8 (A-H,O-Z)
      IMPLICIT INTEGER*2 (I-N)
      REAL*8      EPS,NETA,DNX1(4),DNY1(4),DNX2(4),DNY2(4)
      +      ,X(13),Y(13)
      INTEGER*2  ELEM

```

```

COMMON /JELEM/ JELEM(4,102)
COMMON /COORD/ COORD(2,123)
COMMON /RE/ REC(123,123)
COMMON /F/ FC(123),FAD
COMMON /PROP/ RO,PROP,BO,RCB,4),RFC(4),WITER,PREC
  I1 = 1
  I2 = 2
  I3 = 3
  I4 = 4
  X(13) = 0.0D00
  Y(13) = 0.0D00
  DO 15 I = 1,4
    NO = APSCJELEM(I,ELEM)
    XCI) = COORD(1,NO)
    YCI) = COORD(2,NO)
    X(13) = X(13) + XCI)
    Y(13) = Y(13) + YCI)
    X(13) = X(13)/4.0D00
    Y(13) = Y(13)/4.0D00
  DO 25 I = 1,4
    IU = I+1
    IF (IU .EQ. 5) IU = 1
    X(I+4) = (XCI) + X(IU))/2.0D00
    Y(I+4) = (YCI) + Y(IU))/2.0D00
    X(I+8) = (X(13) + X(I+4))/2.0D00
    Y(I+8) = (Y(13) + Y(I+4))/2.0D00
25
C ESSA PROCEDURE FAZ A MONTAGEM DA MATRIZ LOCAL DE UM ELEMENTO
C *****
C VOLUMES DE CONTROLE 1,2,4
C *****
  DX1 = X(8) - X(13)
  DY1 = Y(8) - Y(13)
  DX2 = X(13) - X(5)
  DY2 = Y(13) - Y(5)
  EPS = -0.5D00
  NETA = 0.0D00
CALL JACOB4 C ELEM, EPS, NETA, DETJ, DNX1, DNY1)

```

```

      EPS      = 0.0D00
      NETA     = -0.5D00
      CALL JACOB4 ( ELEM, EPS, NETA, DETJ, DNX2, DNY2)
      DO 10 II = 1 , 4
C     PRIM. TERMO -> Q4,1
      AUX1 = -(CONDC14, ELEM * DNX1(II) * DY1 - CONDC14, ELEM * DNY1(II) * DX1)
C     SEGUNDO TERMO -> Q2,1
      AUX2 = -(CONDC11, ELEM * DNX2(II) * DY2 - CONDC11, ELEM * DNY2(II) * DX2)
      RC1,II) = AUX1 + AUX2 + RC1,II)
C INCLUI TERMO PARA VOLUME 4
      RC4,II) = -AUX1 + RC4,II)
C INCLUI TERMO PARA VOLUME 2
      RC2,II) = -AUX2 + RC2,II)
10    CONTINUE
C
C *****
C     VOLUMES DE CONTROLE 2,3
C *****
      DX1 = X(13) - X(6)
      DY1 = Y(13) - Y(6)
      EPS      = 0.5D00
      NETA     = 0.0D00
      CALL JACOB4 ( ELEM, EPS, NETA, DETJ, DNX1, DNY1)
      DO 20 II=1,4
C     PRIMEIRO TERMO -> Q3,2
      AUX1 = -(CONDC12, ELEM * DNX1(II) * DY1 - CONDC12, ELEM * DNY1(II) * DX1)
      RC2,II) = AUX1 + RC2,II)
C INCLUI TERMO PARA VOLUME 3
      RC3,II) = -AUX1 + RC3,II)
20    CONTINUE
C *****
C     VOLUMES DE CONTROLE 3,4
C *****
      DX2 = X(13) - X(7)
      DY2 = Y(13) - Y(7)
      EPS      = 0.0D00
      NETA     = 0.5D00

```

```

C
      CALL JACOB4 ( ELEM, EPS, NETA, DETJ, DNX2, DNY2)
C   SEGUNDO TERMO  -> Q4,3
      DO 30  II=1 , 4
      AUX2 = -(CONDCI3, ELEM * DNX2(CII) * DY2 - CONDCI3, ELEM * DNY2(CII) * DX2)
      RC(3,II) = AUX2 + RC(3,II)
      RC(4,II) = -AUX2 + RC(4,II)
30    CONTINUE
      RETURN
      END

```

C*****

```

      SUBROUTINE SISTEMAC(N)
      IMPLICIT REAL*8 (A-H, O-Z)
      IMPLICIT INTEGER*2 (I-N)
      COMMON /RE/ REC(123,123)
      COMMON /F/ F(123), FAD
      COMMON /U/ UC(123)
      COMMON /V/ VC(123)
      COMMON /P/ PC(123)
      INTEGER*4      N
C   ESSA PROCEDURE CALCUA A SOLUCAO DE UM SISTEMA PELO METODO DE
C   ELIMINACAO DE GAUSS .
C
      REWIND(9)
      DO 10  I = 1, N
10     REC(I, N+1) = FCI)
C-----
      DO 70  I= 1 , N
      DO 40  K = I+1 , N
      DO 35  J=I , ( K-1)
      IF (ABS(RECK, I)) .GT. ABS( RECJ, I)) THEN
          DO 30 II = 1, N+1
          AUX = RECK, II)
          RECK, II) = RECJ, II)
          RECJ, II) = AUX
30     CONTINUE
      ENDIF

```

```

35      CONTINUE
40      CONTINUE
        D= RECI,I)
        IF (D.EQ.0.0000) THEN
WRITE(*,*)'PROGR. TERM. PARA EVITAR ERRO DE DIV. POR ZERO'
WRITE(*,*) 'LINHA: ' ,I
WRITE(9,*)'PROGR. TERM. PARA EVITAR ERRO DE DIV. POR ZERO'
WRITE(9,*) 'LINHA: ' ,I
        STOP
        ENDIF
        IF (ABS(D).LT.1.0D-5) THEN
WRITE(*,*)'PIVO MENOR QUE 1.0D-5 - NO: ',I
WRITE(9,*)' NA SOLUCAO POR GAUSS FOI ENCONTRADO UM '
WRITE(9,*)'PIVO MENOR QUE 1.0D-5 NA LINHA: ',I
WRITE(9,*)'VALOR DO PIVO: ',D
WRITE(9,*)'ISTO PODERA LEVAR A VALORES INCORRETOS NA SOLUCAO'
        ENDIF
DO 50 JJ = 1 , N+1
50      RECI,JJ) = RECI,JJ)/D
DO 65 J = I,N
        IF (J.NE.I) THEN
S = -RECI,I)
        IF ( RECI,I) .NE. 0.0000 ) THEN
DO 60 K = 1 , N+1
RECI,K) = RECI,K)*S + RECI,K)
        IF (K .EQ. I) RECI,K) = 0.0000
60      CONTINUE
        ENDIF
        ENDIF
65      CONTINUE
C END J
70      CONTINUE
C END I
C-----
        I=N
80      AUX = 0.0000
DO 85 K = N,I,-1

```

```

      IF (I .NE. 1) AUX = RECI(I)*FC(I) + AUX
85    CONTINUE
      FC(I) = (RECI(N+1) - AUX)/RECI(I)
      I = I-1
      IF (I .GE. 1) GO TO 80

```

C-----

```

      RETURN
      END

```

C*****

```

      SUBROUTINE SAIDAC(SOL, NNOS, NELEM)
      IMPLICIT REAL*8 (A-H, O-Z)
      IMPLICIT INTEGER*2 (I-N)
      INTEGER*4 NELEM, NNOS
      REAL*8 SOL(123)
      COMMON /JELEM/ JELEM(4,102)
      COMMON /COORD/ COORDX(2,123)
      COMMON /RE/ REC(123,123)
      COMMON /F/ FC(123), FAD
      COMMON /PROP/ RO, PROP, BO, RC(8,4), RFC(4), WITER, PREC
      WRITE(9,1000)
1000    FORMAT(11H RESULTADOS: )
      C
      WRITE(9,1020) NNOS
1020    FORMAT(/,21H NUMERO TOTAL DE NOS: ,I6)
      C
      WRITE(9,1010) PROP
1010    FORMAT(16H CONDUTIVIDADE: ,F9.4)
      WRITE(9,1080) BO
1080    FORMAT(6HBETA: ,F9.5)
      WRITE(9,1060) RO
      WRITE(9,1030) PREC
1030    FORMAT(10HPRECISAO: ,F10.6)
1060    FORMAT(11HDENSIDADE: ,F9.4)
      WRITE(9,1040)
1040    FORMAT(/,33H-----,/,8HSOLUCAO:
      DO 10 I =1, NNOS
      XX = COORDX(1,I)

```



```

C
10      WRITE (9,1050) I, XX,COORDX(2,I),SOL(I)
1050    FORMAT (2HT:,I4,2X,2F12.4,3X,F12.6,)
      RETURN
      END

C*****
      SUBROUTINE  FACESCNNOS,NELEM,ICF)
      IMPLICIT REAL*8 (A-H,O-Z)
      IMPLICIT INTEGER*2 (I-NO)
      INTEGER*4  NNOS,NELEM
      INTEGER*2   ELEM,ELEM1,MCANTOC(10)
      LOGICAL    ACHOU
      COMMON /JELEM/ JELEM(4,102)
      COMMON /COORD/ COORDX(2,123)
C -> ESTA SUBROUTINA DETERMINA QUIAS NOHS SAO NOHS DE FRONTEIRA
C -> ENCONTRA FRONTEIRAS:
      REWIND(10)
      NCANTO = 0
      WRITE(*,*) 'PROCURANDO FRONTEIRAS'
      DO 300 NO = 1,NNOS
      IND = 0
      DO 200 ELEM = 1,NELEM
      DO 200 I =1,4
      IF ( NO .EQ. JELEM(I,ELEM) ) IND = IND +1
200    CONTINUE
      IF ( IND .LE. 3 ) WRITE (10,*)NO
C
      IF ( IND.EQ. 1) THEN
      NCANTO = NCANTO + 1
      MCANTOC(NCANTO) =NO
      ENDIF
C
300    CONTINUE
C-----
      REWIND(11)
      DO 400 ELEM =1,NELEM
      DO 400  I=1,4

```

```

REWINDC10)
  ACHOU = .FALSE.
350   READC10,*,END = 380) NO
      IF (JELEMC1,ELEMD .EQ. NO) THEN
          ACHOU = .TRUE.
          GOTO 380
        ENDIF
      GOTO 350
390   IF (ACHOU .EQV. .TRUE. ) THEN
          WRITEC11,*) -NO
        ELSE
          WRITEC11,*) JELEMC1,ELEMD
        ENDIF
390   CONTINUE
400   CONTINUE
      REWINDC10)
      REWINDC11)
      DO 111 ELEM =1,NELEM
      READC11,*,END=114)NO1A
      DO 112 IFACE =1,4
      IF (IFACE .NE. 4 ) READC11,*,END=444) NO2B
      IF (IFACE .EQ. 1) NO = NO1A
444   IF (IFACE .EQ. 4) THEN
      NO1A = NO2B
      NO2B = NO
      ENDIF
      IF (NO1A .LT. 0.0000 .AND. NO2B .LT. 0.0000)
+     WRITEC10,*) -NO1A, -NO2B,ELEM,IFACE
          NO1A = NO2B
112   CONTINUE
111   CONTINUE
C-----
C VERIFICA SE ENTRE OS ELEMENTOS ESCOLHIDOS COMO POSSIVEIS FRONTEIRAS
C NAO HA SEGMENTOS INTERNOS
114   REWINDC11)
      REWINDC10)
113   READ (10,*,END=110)NO1A,NO2B,ELEM1,IFACE

```

```

      IND = 0
      DO 115 ELEM = 1 ,NELEM
      DO 115 I = 1,4
      IU = I+1
      IF (I .EQ. 4 ) IU = 1
      IAUX1 = ABS(JELEM(I),ELEM)
      IAUX2 = ABS(JELEM(IU),ELEM)
      IFC (IAUX1 .EQ. NO1A .AND. IAUX2 .EQ. NO2B)
+ .OR. (IAUX1.EQ.NO2B .AND. IAUX2.EQ. NO1A) )IND = IND +1
C
115  CONTINUE
      IF (IND .EQ. 1 ) THEN
      DO 117 III = 1,NCANTO
      IF ( NO1A .EQ. MCANTOC(III) ) NO1A = -NO1A
      IF ( NO2B .EQ. MCANTOC(III) ) NO2B = -NO2B
117  CONTINUE
      WRITE(11,*)NO1A,NO2B,ELEM1,IFACE
      DO 10 I = 1,4
      IAUX1 = JELEM(I,ELEM1)
      IFC ( IAUX1 .EQ. ABSC(NO1A) .OR. IAUX1 .EQ. ABSC(NO2B) ) .AND. ICF .EQ. 0)
+ JELEM(I,ELEM1) = -ABSC(IAUX1)
10  CONTINUE
      ENDIF
C
      GOTO 113
C-----
119  CONTINUE
C-----
      RETURN
      END
C*****
      SUBROUTINE EXPANDE CELEM
      IMPLICIT REAL*8 (A-H,O-Z)
      IMPLICIT INTEGER*2 (I-N)
      INTEGER*2 ELEM
      COMMON /JELEM/ JELEM(4,102)
      COMMON /COORD/ COORD(2,123)

```

```

COMMON /RE/ REC123,123)
COMMON /F/ F(123),FAD
COMMON /PROP/ RO,PROP,BO,RC(8,4),RFC4),WITER,PREC
COMMON /U/ UC123)
COMMON /V/ VC123)
COMMON /P/ PC123)
DO 10 I =1,4
NO1=ABSCJELEM(I,ELEM)
  FC(NO1) = FC(NO1) + RFC(I)
DO 10 J =1,4
NO2=ABSCJELEM(J,ELEM)
10 REC(NO1,NO2) = REC(NO1,NO2) + RCI,J)
RETURN
END

```

C*****

SUBROUTINE DIMVARCNELEM,NNOS)

IMPLICIT REAL*8 (A-H,O-Z)

IMPLICIT INTEGER*2 (I-N)

INTEGER*4 NELEM,NNOS

INTEGER*2 ELEM

COMMON /JELEM/ JELEM(4,102)

COMMON /COORD/ COORD(2,123)

COMMON /RE/ REC123,123)

COMMON /PROP/ RO,PROP,BO,RC(8,4),RFC4),WITER,PREC

COMMON /F/ F(123),FAD

COMMON /U/ UC123)

COMMON /V/ VC123)

COMMON /P/ PC123)

NELEM =0

NNOS =0

XITER = 0.000

C -> IREP= 1 INDICA QUE ELEMENTOS IGUAIS -FORNECIDOS NO ARQUIVO

C NAO PRECISARAO SEREM CALCULADOS DE NOVO.

IREP =0

CALL LE (NELEM,NNOS,ISOL,IREP,ICF)

CALL FACESCNNOS,NELEM,ICF)

10 XITER = XITER +1.000

```

REWIND(16)
  IS = ISOL
  IF (ISOL .EQ. 2) IS = 1
C-----
C VERIFICA SE EH NECESSARIO CALCULAR MATRIZES DE TODOS ELEMENTO OU NAO
  IF (IREP .EQ. 1) THEN
    REWIND(18)
15    READ (18,*,END=20) ELEM
      IF (ELEM .GT. 0) THEN
        DO 16 II =1,4
16    JELEMCII,ELEM = ABS( JELEMCII,ELEM )
      CALL ZERA
      IF (PROP.NE. 0.000) CALL MDIFFCELEM
      IF (ISOL .NE. 2) CALL MCONVCELEM
      ENDIF
      ELEM = ABSCELEM
      CALL EXPANDECELEM
      GOTO 15
20    CONTINUE
C-----
      ELSE
        DO 1000 ELEM =1,NELEM
C ->  CALCULO DO LAPLACIANO
          CALL ZERA
          IF (PROP.NE. 0.000) CALL MDIFFCELEM
C ->  SOLUCAO PARA PROBLEMAS DE CONVECCAO-DIFUSAO
          IF (ISOL .NE. 2) THEN
C
            CALL MCONVCELEM
            ENDIF
C
            CALL EXPANDECELEM
C
1000    CONTINUE
          ENDIF
C-----

```

```

CALL LIMITECISOL,NNOS,ICF)
CALL SISTEMAC(NNOS)
IEC = 0
DIFMAX = 0.0000
DO 17 I=1,NNOS
  DIF = ABS( F(I)-P(I))
  IF ( DIF .GT. PREC) IEC = 1
  IF ( DIF .GT. DIFMAX ) THEN
    NOM = I
    DIFMAX = DIF
  ENDIF
  F(I) = F(I)
17 CONTINUE
  WRITE (*,*) ' MAIOR DIFERENCA: ',DIFMAX
  WRITE (*,*) ' NO: ',NOM
C
  IF (IEC .EQ. 1 .AND. XITER .LT. WITER) THEN
    REWIND(15)
    I = 1
25 READ (15,*,END=30) F(I)
    I = I+1
    GOTO 25
30 CONTINUE
C ZERA MATRIZ PARA A PROXIMA ITERACAO
DO 32 I = 1 ,NNOS
DO 32 J = 1 ,NNOS
32 RECI,J) = 0.0000
    GOTO 10
  ENDIF
  WRITE(*,*) ' ESCRIVENDO SOL. NO ARQ.: -> FEMEVSAL '
  WRITE(9,*) ' NUMERO DE ITERACOES PARA CONV.: ',XITER
  CALL SAIDACF,NNOS,NELEM
C
  CLOSE( 7, STATUS = 'DELETE')
  CLOSE(10, STATUS = 'DELETE')
  CLOSE(11, STATUS = 'DELETE')
  CLOSE(12, STATUS = 'DELETE')

```

```
CLOSE(13, STATUS = 'DELETE')
CLOSE(14, STATUS = 'DELETE')
CLOSE(15, STATUS = 'DELETE')
```

C

```
RETURN
END
```

C*****

```
BLOCK DATA
IMPLICIT REAL*8 (A-H,O-Z)
IMPLICIT INTEGER*2 (I-N)
COMMON /JELEM/ JELEM(4,102)
COMMON /COORD/ COORD(2,123)
COMMON /RE/ REC(123,123)
COMMON /F/ FC(123),FAD
COMMON /PROP/ RO,PROP,B0,RC(8,4),RFC(4),WITER,PREC
COMMON /U/ UC(123)
COMMON /V/ VC(123)
COMMON /P/ PC(123)
DATA RE/15129*0.0D00/
DATA F/123*0.0D00/
DATA U/123*0.0D00/
DATA P/123*0.0D00/
DATA V/123*0.0D00/
DATA COORD/246*0.0D00/
END
```