

RUDMAR SERAFIM MATOS

**OTIMIZAÇÃO TEÓRICO-NUMÉRICA E COMPARAÇÃO DE  
DESEMPENHO DE TROCADORES DE CALOR DE TUBOS  
CIRCULARES E ELÍPTICOS NÃO ALETADOS EM  
CONVECÇÃO FORÇADA**

Dissertação apresentada ao Programa Interdisciplinar de Pós-Graduação em Engenharia da Universidade Federal do Paraná, como requisito parcial à obtenção do grau de Mestre.

Orientador: Prof. Dr. José Viriato C. Vargas

CURITIBA  
2000



## RELATÓRIO DA DEFESA DE DISSERTAÇÃO DE MESTRADO

Aos nove dias do mês de maio de 2000, no Auditório do CESEC, foi instalada pelo Prof.Dr. José Viriato Coelho Vargas, vice-coordenador do PIPE – Programa Interdisciplinar de Pós-Graduação em Engenharia, a Banca Examinadora para a sexta Dissertação de Mestrado. Área de Concentração: Energia e Ciências Térmicas. Estiveram presentes no Ato, além do vice-coordenador do Curso de Pós-Graduação, professores, alunos e visitantes.

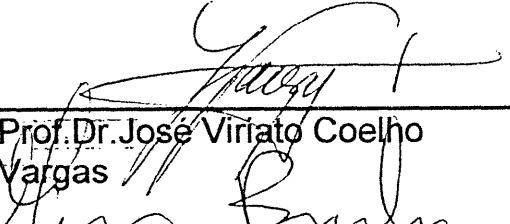
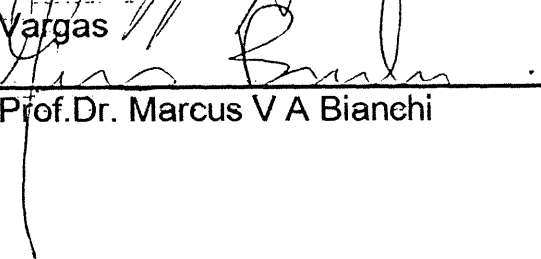
A Banca Examinadora, atendendo determinação do Colegiado do Programa Interdisciplinar de Pós-Graduação em Engenharia, ficou constituída pelos professores: **José Viriato Coelho Vargas** (Demec), **George Stanescu** (Demec), **Marcus V A Bianchi** ( Depto. Eng. Mecânica-UFRGS).

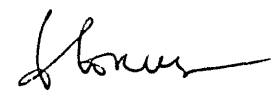
Às 09:00 horas, a banca iniciou os trabalhos, convidando o(a) Candidato **Rudmar Serafim Matos** a fazer a apresentação do tema de dissertação intitulada “**Otimização Teórico-Numérica e Comparação de Desempenho de Trocadores de Calor de Tubos Circulares e Elípticos não Aletados em Convecção Forçada**”. Encerrada a apresentação, iniciou-se a fase de arguição pelos membros participantes.

Tendo em vista a dissertação e a arguição, a banca atribuiu as seguintes notas: Prof. José Viriato Coelho Vargas, Nota 10, Prof. George Stanescu, Nota 10, Prof. Marcus V A Bianchi, Nota 10.

A média obtida 10 resulta na aprovação do candidato. (de acordo com a determinação dos Artigos 32, 33 e 34 da Resolução 74/94 de 21/10/94), e corresponde ao conceito A/B/C/D.

Curitiba, 09 de maio de 2000 .

  
\_\_\_\_\_  
Prof. Dr. José Viriato Coelho  
Vargas  
  
\_\_\_\_\_  
Prof. Dr. Marcus V A Bianchi

  
\_\_\_\_\_  
Prof. Dr. George Stanescu

Agradeço a inestimável colaboração do Professor José Viriato Coelho Vargas durante todo o Curso de mestrado, tratando-se de um verdadeiro exemplo de profissionalismo e dedicação, possibilitando a realização deste trabalho, aos professores e colegas da pós-graduação, pela amizade e sugestões apresentadas, a minha família e aos meus falecidos pais, razão maior da minha existência e formação.

## INDICE

	LISTA DE ILUSTRAÇÕES.....	v
	LISTA DE ABREVIATURAS E SIMBOLOS.....	vi
	RESUMO.....	x
	ABSTRACT.....	xi
1	INTRODUÇÃO.....	1
1.1	Considerações iniciais.....	1
1.2	Objetivos.....	2
2	REVISÃO DA LITERATURA.....	4
3	O PROBLEMA FÍSICO.....	7
3.1	Geometria do arranjo.....	7
3.2	Critério de comparação entre tubos de seção circular e elíptica.....	9
3.3	Disposição dos tubos.....	9
4	MODELO MATEMÁTICO.....	11
4.1	Equações governantes.....	11
4.2	Forma adimensional.....	12
4.3	Equações governantes adimensionais.....	14
4.4	Condições de contorno adimensionais.....	15
5	MÉTODO NUMÉRICO.....	16
5.1	Considerações iniciais.....	16
5.2	Malha utilizada.....	17
5.3	Teste de refinamento da malha .....	17
5.4	Equações de Navier-Stokes e Energia em Elementos Finitos.....	18
5.5	Algoritmo para cálculo da transferência de calor, em tubos de seção circular...	20
5.6	Algoritmo para integração da transferência de calor ao longo da superfície do tubo.....	23
5.7	Metodologia para o cálculo da transferência de calor em tubos de seção circular e elíptica.....	25

6	DISCUSSÃO DE RESULTADOS.....	28
6.1	Validação dos resultados numéricos para arranjos de tubos circulares.....	28
6.2	Otimização de arranjos tubos de seção elíptica.....	29
6.3	Comparação entre tubos de seção circular e elíptica.....	32
6.4	Distância entre centros variável na direção do escoamento.....	39
7	CONCLUSÕES.....	40
ANEXOS		
ANEXO 1	Teste do cálculo da transferência de calor para uma cavidade quadrada.....	41
ANEXO 2	Input-file para cavidade quadrada.....	48
ANEXO 3	Input-file para 4 tubos no canal elementar, $e = 1.0$ , $S/D = 1.0$ e $Re_L = 620$ .....	49
ANEXO 4	Input-file para 4 tubos no canal elementar, $e = 0.80$ , $S/2b = 1.0$ e $Re_L = 620$ .....	56
ANEXO 5	Input-file para 4 tubos no canal elementar, $e = 0.65$ , $S/2b = 1.0$ e $Re_L = 620$ .....	63
ANEXO 6	Elemento Navier-Stokes e Energia para 4 tubos no canal elementar .....	70
REFERÊNCIAS BIBLIOGRÁFICAS.....		98

## LISTA DE ILUSTRAÇÕES

Figura 1	Arranjo de tubos elípticos .....	7
Figura 2	Superfícies de troca de calor .....	9
Figura 3	Distribuição triangular eqüilateral para 4 tubos no canal elementar.....	9
Figura 4	Condições de contorno no canal elementar .....	15
Figura 5	Malha gerada através do código FEAP para 4 tubos no canal elementar.....	17
Figura 6	Coordenadas polares no círculo .....	20
Figura 7	Gráfico dos resultados numéricos e experimentais para trocadores de calor de tubos circulares com arranjo triangular eqüilateral.....	29
Figura 8	Gráfico dos resultados numéricos para trocadores de calor de tubos circulares ( $e = 1$ ) .....	30
Figura 9	Gráfico dos resultados numéricos para trocadores de calor de tubos elípticos ( $e = 0.8$ ) .....	31
Figura 10	Gráfico dos resultados numéricos para trocadores de calor de tubos elípticos ( $e = 0.65$ ) .....	31
Figura 11	Gráfico do efeito da excentricidade das elipses na transferência de calor $Re_L = 465$ .....	32
Figura 12	Gráfico do efeito da excentricidade das elipses na transferência de calor $Re_L = 620$ .....	33
Figura 13	Gráfico do efeito da excentricidade das elipses sobre a máxima condutância térmica global.....	34
Figura 14	Gráfico do efeito da excentricidade das elipses sobre o espaçamento ótimo para a máxima condutância térmica global .....	34
Figura 15	Gráfico das isothermas para $(S/2b)_{ot}$ , $e = 1.0$ e $Re_L = 620$ .....	36
Figura 16	Gráfico das isothermas para $(S/2b)_{ot}$ , $e = 0.8$ e $Re_L = 620$ .....	37
Figura 17	Gráfico das isothermas para $(S/2b)_{ot}$ , $e = 0.65$ e $Re_L = 620$ .....	38
Figura 18	Gráfico da distância entre centros ótima na direção do escoamento.....	39
Figura 19	Convecção natural em uma cavidade quadrada.....	41
Figura 20	Malha gerada pelo código FEAP para uma cavidade quadrada.....	45
Figura 21	Condições de contorno na cavidade quadrada.....	46
Figura 22	Gráfico das isothermas para uma cavidade quadrada.....	47

## LISTA DE ABREVIATURAS E SÍMBOLOS

$a$	Semi-eixo maior do tubo de secção elíptica, m
$b$	Semi-eixo menor do tubo de secção elíptica, m
$c_p$	Calor específico a pressão constante, J/(kg.K)
$\bar{c}_p$	Coeficiente referente ao calor específico a pressão constante, (FEAP)
$D$	Diâmetro do tubo circular, m
$e$	Excentricidade do tubo, $b/a$
$f$	Função que representa a equação da elipse ou do círculo
$F$	Fluxo de calor adimensional
FEAP	Finite Element Analysis Program ( Zienkiewicz e Taylor, 1989)
$g$	Gravidade, $m/s^2$
$\bar{g}$	Coeficiente referente a gravidade, (FEAP)
$\bar{g}_x$	Coeficiente referente a componente de gravidade na direção x, (FEAP)
$\bar{g}_y$	Coeficiente referente a componente de gravidade na direção y, (FEAP)
$H$	Altura do arranjo, m
$I_i$	Integral da transferência de calor ao longo da superfície do tubo
$k$	Condutividade térmica, W/(m.K)
$\bar{k}$	Coeficiente referente a condutividade térmica, (FEAP)
$k', l'$	Número de pontos para a quadratura gaussiana nas duas direções do problema bidimensional
$l$	Distância entre centros dos tubos na direção do escoamento
$\tilde{l}$	Distância adimensional entre centros dos tubos na direção do escoamento
$l_{max}$	Distância máxima entre centros dos tubos na direção do escoamento
$L$	Comprimento do arranjo, m
$\dot{m}$	Vazão mássica de ar, kg/s
$n$	Vetor normal
$N$	Número de tubos no canal elementar
$N_{cel}$	Número de células do arranjo
$Nu$	Número de Nusselt, $\frac{Q}{Q_{cond\ pura}}$

$\bar{Nu}$	Número de Nusselt médio, $\int_0^1 \left( \frac{\partial \theta}{\partial X} \right)_{X=0} dY$
$p$	Pressão, N/m <sup>2</sup>
$P$	Pressão adimensional, $\frac{p}{\rho U_\infty^2}$
$Pe_L$	Número de Peclet, $\frac{U_\infty L}{\alpha}$
$Pr$	Número de Prandtl, $\nu/\alpha$
$q$	Taxa de transferência de calor, W
$q_n''$	Taxa de transferência de calor normal a superfície
$\left  \bar{q}_{n,i}'' \right $	Taxa de transferência de calor média por unidade de área por superfície do canal elementar
$q_r$	Taxa de transferência de calor resultante
$q_x$	Taxa de transferência de calor na direção x
$q_y$	Taxa de transferência de calor na direção y
$q_i''$	Taxa de transferência de calor de cada tubo, W/m <sup>2</sup>
$\bar{q}_i''$	Taxa de transferência de calor média, W/m <sup>2</sup>
$q_{1cel}$	Taxa de transferência de calor de uma célula, W
$q_{ref}'''$	Fluxo de calor volumétrico, W/m <sup>3</sup>
$\tilde{q}_i$	Taxa de transferência de calor volumétrica adimensional por tubo, $I_i \frac{\left[ \frac{D}{L} \right]^2}{\frac{H}{D}} \frac{N_{cel}}{2}$
$\tilde{q}$	Taxa de transferência de calor volumétrica adimensional total, $\sum_{i=1}^N \tilde{q}_i$
$\tilde{q}_*$	Taxa de transferência de calor volumétrica adimensional total, $\frac{N_{cel} q_{1cel}}{k (T_w - T_\infty) LHW/D^2}$
$Q$	Taxa de geração total de calor, W



$r$	Raio do tubo circular, m
$Ra$	Número de Rayleigh
$Re_D$	Número de Reynolds baseado no diâmetro
$Re_L$	Número de Reynolds baseado no comprimento do arranjo
$S$	Distância entre tubos, m
$(S + 2b) / 2$	Distância entre centros do tubo de secção elíptica no canal elementar e perpendicular a direção do fluxo, m
$(S + 2D) / 2$	Distância entre centros do tubo de secção circular no canal elementar e perpendicular a direção do fluxo, m
$T$	Temperatura, K
$T_c$	Temperatura da parede fria, K
$T_h$	Temperatura da parede quente, K
$\bar{T}_s$	Temperatura média do fluido na saída do arranjo ( $x=3L$ ), K
$T_w$	Temperatura superficial do tubo, K
$T_\infty$	Temperatura do escoamento livre, K
$T_{max}$	Temperatura máxima na superfície dos tubos, K
$u, v$	Componentes da velocidade, m/s
$\vec{u}$	Vetor velocidade, m/s
$U, V$	Componentes da velocidade adimensionais
$U_\infty$	Velocidade do escoamento livre, m/s
$W$	Largura do arranjo, m
$x, y$	Coordenadas cartesianas, m
$X, Y$	Coordenadas cartesianas adimensionais
$x_i, y_i$	Coordenadas dos centros dos tubos

#### **Símbolos gregos:**

$\alpha$	Difusividade térmica, $m^2/s$
$\alpha_p$	Parâmetro de “upwinding”
$\bar{\beta}$	Coefficiente referente ao coeficiente de expansão térmica (FEAP)
$\theta$	Temperatura adimensional

$\bar{\theta}_s$	Temperatura média adimensional na saída do arranjo, (em 3L)
$\mu$	Viscosidade dinâmica, kg/m.s
$\bar{\mu}$	Coefficiente referente a viscosidade dinâmica, (FEAP)
$\nu$	Viscosidade cinemática, m <sup>2</sup> /s
$\rho$	Densidade, kg/m <sup>3</sup>
$\bar{\rho}$	Coefficiente referente a densidade, (FEAP)
$\gamma$	Parâmetro de penalidade
$\Delta p$	Variação de pressão
$\Delta P$	Variação de pressão adimensional
$\delta$	Ângulo do círculo em coordenadas polares
$\nabla f$	Gradiente da função f.

## RESUMO

Neste estudo, foi realizada uma análise da transferência de calor em trocadores de calor de tubos circulares e elípticos, em duas dimensões. O método de elementos finitos foi utilizado para discretizar as equações governantes do escoamento e da transferência de calor, e um elemento bidimensional, isoparamétrico, quadrilateral com funções de interpolação linear, foi implementado e agregado ao código FEAP (Zienkiewicz e Taylor, 1989). Os resultados numéricos, para uma configuração de arranjo triangular equilateral, obtidos com o novo elemento foram então validados, via comparação direta com os resultados experimentais previamente publicados para trocadores de calor de tubos circulares (Stanescu et al., 1996). A seguir, uma otimização numérica da geometria foi conduzida para maximizar a taxa total de transferência de calor entre o volume dado e o escoamento externo tanto para arranjos circulares como elípticos, para várias configurações gerais. Os resultados são apresentados para ar na faixa de  $300 \leq Re_L \leq 800$ , onde  $L$  é o comprimento varrido do volume fixado. Arranjos circulares e elípticos com a mesma área de obstrução na direção do escoamento livre foram comparados com base na transferência total de calor máxima. O efeito da excentricidade das elipses também foi investigado. Um ganho relativo na transferência de calor de até 13 % é observado no arranjo elíptico ótimo, em comparação ao arranjo circular ótimo. O ganho de transferência de calor, combinado com a redução em perda de carga de até 25 % observada em estudos anteriores (Brauer, 1964; Bordalo e Saboya, 1995) mostram que arranjos de tubos elípticos têm potencial para um desempenho consideravelmente melhor do que os convencionais circulares.

## ABSTRACT

In this study, a two-dimensional (2-D) heat transfer analysis was performed in circular and elliptic tube heat exchangers. The finite element method was used to discretize the fluid flow and heat transfer governing equations and a 2-D isoparametric, four-noded, linear element was implemented for the finite element analysis program FEAP (Zienkiewicz and Taylor, 1989). The numerical results for the equilateral triangle staggering configuration, obtained with the new element were then validated by means of direct comparison to previously published experimental results for circular tubes heat exchangers (Stanescu et al., 1996). Next, a numerical geometric optimization was conducted to maximize the total heat transfer rate between the given volume and the given external flow both for circular and elliptical arrangements, for general staggering configurations. The results are reported for air in the range  $300 \leq Re_L \leq 800$ , where  $L$  is the swept length of the fixed volume. Circular and elliptical arrangements with the same flow obstruction area were compared on the basis of maximum total heat transfer. The effect of ellipses eccentricity was also investigated. A relative heat transfer gain of up to 13 % is observed in the optimal elliptical arrangement, as compared to the optimal circular one. The heat transfer gain, combined with the relative pressure drop reduction of up to 25 % observed in previous studies (Brauer, 1964; Bordalo and Saboya, 1995), shows the elliptical arrangement has the potential for a considerably better overall performance than the conventional circular one.

# 1 INTRODUÇÃO

## 1.1 Considerações iniciais

Um novo desafio apresenta-se ao engenheiro do próximo milênio. Não é suficiente realizar um projeto que seja exequível e funcional. Além disso, o mesmo deve funcionar de forma otimizada, isto é, com máximo desempenho.

A alta competitividade do mercado nos dias atuais, requer conhecimentos cada vez mais especializados e equipamentos adequados para se obter o melhor produto pelo menor investimento. Neste sentido, muito esforço tem sido dedicado à busca de equipamentos mais eficientes, devido à preocupação com o uso ótimo da energia disponível.

A otimização dos processos de troca de energia tem sido objeto de vários estudos nos últimos tempos. Isto ocorre devido à sua importância, como por exemplo na refrigeração de equipamentos eletrônicos, que é um problema básico de transferência de calor. A evolução de tais equipamentos é acompanhada de uma redução do volume ocupado por este. Dessa forma, devemos obter um máximo de transferência de calor para um dado espaço, o que caracteriza a necessidade de estudos de otimização. Arranjos de tubos estão presentes em muitas aplicações industriais, atuando como trocadores de calor em sistemas de ar condicionado, aquecedores e radiadores.

Nos processos em que a troca de calor deve ser feita através de um fluido em escoamento forçado, deseja-se uma troca eficiente de calor, mas sem que haja, excessiva perda de carga. Uma alta perda de carga implica em alta potência de bombeamento, acarretando consumo desnecessário de energia.

O assunto deste trabalho foi motivado por esta necessidade crescente de otimização em todas as aplicações da Engenharia, visando racionalizar o uso da energia disponível e reduzir o trabalho perdido.

No caso específico deste trabalho, pretende-se através da quantificação da troca de calor em um certo volume fixo, para várias configurações de tubos, chegar, numericamente, à configuração geométrica ótima para arranjos de tubos circulares e elípticos, presentes em muitas aplicações industriais modernas.

Um grande número de problemas da engenharia é regido por equações diferenciais parciais. Entretanto, em geral, a obtenção de soluções exatas para estas equações é tarefa de grande dificuldade ou, via de regra, impossível. Esta dificuldade pode estar relacionada à própria forma da equação, à geometria, ou às condições de contorno e iniciais.

Surge assim a necessidade de resolver o problema de forma aproximada, com a utilização de métodos numéricos. Este procedimento permite obter informações mais detalhadas e compreensíveis, com tempo de projeto reduzido, além da possibilidade de simular situações impossíveis de se obter de forma experimental. Tomando como exemplo este trabalho, foram realizados inúmeros testes que experimentalmente seriam inviáveis. Esta aproximação consiste no processo de conversão das equações diferenciais parciais contínuas em um sistema de equações algébricas que representam o domínio discreto. Para a discretização foi escolhido o método de Elementos Finitos (Zienkiewicz e Taylor, 1989).

A precisão obtida na solução aproximada está diretamente relacionada à qualidade da malha (representação do contínuo). Regiões onde a solução apresente elevados gradientes devem apresentar uma maior densidade de pontos, comparadas a outras. Esta maior densidade de pontos pode ser obtida através de refinamentos e/ou concentrações de linhas da malha na região problemática, razão pela qual a malha deste problema foi mais refinada ao redor dos tubos.

A seguir, são apresentados os objetivos do presente trabalho. No capítulo 2, apresenta-se uma revisão da literatura técnica, descrevendo e discutindo estudos relacionados com o tema desta dissertação, publicados até a presente data. No capítulo 3, apresenta-se e discute-se o problema físico a ser resolvido nesta tese. No capítulo 4, listam-se as hipóteses adotadas e apresenta-se o modelo matemático para o problema físico. No capítulo 5, detalha-se e discute-se o método numérico utilizado nesta tese para a solução das equações governantes do problema, apresentando também as figuras de mérito propostas para análise de performance dos arranjos de tubos em estudo. No capítulo 6, apresentam-se os resultados numéricos obtidos nesta tese e discute-se sua validação a luz de dados experimentais, os novos resultados para a geometria de tubos elípticos e a análise comparativa entre tubos circulares e elípticos.

## **1.2 Objetivos**

Este estudo tem por objetivos:

- a) Utilizar o código aberto FEAP, "Finite Element Analysis Program", desenvolvido por Zienkiewicz e Taylor (1989), e elementos específicos para a solução numérica em duas dimensões das equações de conservação (massa, momentum e energia), para obter soluções para arranjos de tubos circulares e elípticos em convecção forçada. Primeiramente o código é validado reproduzindo-se resultados disponíveis na literatura

para a otimização de arranjos de tubos circulares em convecção forçada (Stanescu et al., 1996).

- b) Buscar a otimização de arranjos de tubos elípticos em convecção forçada, para máxima transferência de calor, sob uma restrição de volume fixo, através do código aberto FEAP. A seguir comparam-se os resultados com os resultados para arranjos de tubos circulares, reportando-se os ganhos percentuais da configuração ótima elíptica em relação à circular, do ponto de vista de transferência de calor.
- c) Apresentar os resultados, através de grupos adimensionais convenientes, portanto, de caráter geral para qualquer configuração de tubos elípticos.

## 2 REVISÃO DA LITERATURA

O desenvolvimento de técnicas de resfriamento para equipamentos eletrônicos é ilustrado pelo surgimento de resultados fundamentais que se aplicam a classes inteiras de configurações geométricas (Bar-Cohen e Kraus, 1990; Peterson e Ortega, 1990). Um exemplo claro são as normas de projeto para a seleção de espaçamento entre os componentes de um equipamento de volume fixo, tal que a condutância térmica média entre o equipamento e o refrigerante seja maximizada. Os espaçamentos ótimos já foram reportados para várias geometrias, tanto em convecção natural (Bar-Cohen e Rohsenow, 1984; Bejan, 1995; Kim et al., 1991; Anand et al., 1992), como em convecção forçada (Knight et al., 1991; Knight et al., 1992; Bejan e Morega, 1993; Hirata et al., 1970; Nakayama et al., 1988; Matsushima et al., 1992; Bejan e Sciubba, 1992).

Shepherd (1956) analisou tubos circulares com uma fileira e trocadores de calor com aletas determinando o coeficiente global de transferência de calor como uma função do número de Reynolds, assumindo aletas isotérmicas.

Brauer (1964) apresentou uma pesquisa com resultados experimentais comparando, arranjos de tubos elípticos e circulares para transferência de calor e perda de carga, baseado na observação de que a geometria do tubo elíptico tem uma configuração aerodinâmica melhor que a do tubo circular. Portanto, é razoável esperar-se uma redução na força total de arrasto e um aumento na transferência de calor, ao comparar-se arranjos de tubos elípticos com arranjos de tubos circulares, submetidos a um escoamento livre, em fluxo cruzado.

Schulemberg (1966) analisou o potencial de aplicação de tubos elípticos em trocadores de calor na indústria, mostrando experimentalmente os resultados da transferência de calor e perda de carga.

Saboya (1974) obteve experimentalmente os coeficientes de transferência de calor locais, usando a técnica de sublimação do naftaleno por uma analogia de transferência de calor e massa para trocadores de calor de uma e duas fileiras de tubos circulares aletados.

Saboya e Sparrow (1976) estenderam o estudo para trocadores de calor de tubos circulares de três fileiras. Os resultados mostraram baixos coeficientes de transferência de massa atrás dos tubos, quando comparados com a média dos arranjos.

Ximenes (1981) relatou experimentalmente resultados para coeficientes de transferência de massa em trocadores de calor de uma e duas fileiras de tubos elípticos



aletados. Na configuração para tubo elíptico foi observado que a queda do coeficiente de transferência de massa foi menos acentuada atrás dos tubos do que na configuração circular.

Rosman et al. (1984) experimentalmente determinaram os coeficientes de transmissão de calor locais e globais usando a analogia entre transferência de calor e massa, para trocadores de calor de uma e duas fileiras de tubos circulares aletados, através da determinação numérica da distribuição de temperatura da aleta e eficiência, e a temperatura do escoamento livre ao longo da aleta. Os resultados mostraram que a configuração de duas fileiras é mais eficiente que a configuração de uma fileira.

Bordalo e Saboya (1995) relataram medições da perda de carga comparando arranjos com configurações de uma, duas e três fileiras. A conclusão destes estudos baseados em evidências experimentais é que a configuração com tubos elípticos tem melhor performance do que com tubos circulares.

Stanescu et al. (1996) estudaram o espaçamento ótimo de tubos circulares em escoamento livre e convecção forçada, fluxo cruzado, e Bejan et al. (1995) apresentaram um estudo similar em convecção natural. Ambos os estudos levam em consideração a limitação de espaço disponível em qualquer projeto, através de uma restrição de volume fixo. Nestas configurações as simetrias presentes nos arranjos permitiram que os problemas fossem modelados numericamente em duas dimensões. O espaçamento ótimo entre fileiras de tubos foi determinado por maximização da transferência de calor entre os tubos circulares e o escoamento livre.

Rocha et al. (1997) estudaram trocadores de calor de tubos elípticos aletados para configurações específicas de 1 e 2 fileiras de tubos, com uma formulação simplificada, sendo que os resultados de eficiência foram comparados aos de trocadores de calor de tubos circulares. Os resultados mostraram para as configurações estudadas ganhos percentuais de até 18% para transferência de calor e de até 25% para perda de carga.

Neste trabalho, busca-se determinar o espaçamento ótimo entre fileiras de tubos para uma outra configuração básica: o arranjo de tubos elípticos com escoamento transversal em convecção forçada. Dentro deste contexto, este trabalho visa avaliar a geometria de trocadores de calor de tubos elípticos não aletados sob convecção forçada, através de simulações numéricas, utilizando o método de elementos finitos, capazes de prever o desempenho do trocador de calor, do ponto de vista de transferência de calor.

O modelo numérico deste estudo é validado qualitativamente pela comparação direta com resultados experimentais publicados para a geometria de tubos circulares (Stanescu et

al., 1996). A seguir, resultados inéditos para a geometria de tubos elípticos são obtidos e comparados com os resultados de tubos circulares.



As componentes da velocidade,  $u$ , e  $v$ , são alinhadas com os eixos,  $x$  e  $y$ , respectivamente. O domínio computacional é composto pelo canal elementar, que tem por comprimento total  $3L$ . Para isto, nas simulações, acrescentou-se o comprimento  $L$  na entrada e na saída do arranjo (não representado na Figura 1).

O arranjo geométrico dos tubos tem quatro graus de liberdade: o espaçamento vertical,  $(S+2b)/2$ , dado pela distância entre centros perpendicular à direção do fluxo no canal elementar, o número de tubos,  $N$ , montados no canal elementar de comprimento  $L$ , a excentricidade,  $e$ , e o espaçamento horizontal,  $l = \sqrt{3} (S+2b)/2$ , que representa a distância entre centros na direção do fluxo no canal elementar, para um arranjo triangular equilateral.

A velocidade do escoamento livre é um parâmetro externo e pode variar, sendo representada pelo número de Reynolds,  $UL/\nu$ , baseado no comprimento  $L$  do volume fixo.

A geometria do módulo montado no volume LHW pode ser mudada por variação de  $(S+2b)/2$ ,  $N$ ,  $e$ , e  $l$  ou uma combinação destes parâmetros.

É de interesse o arranjo geométrico, dado pela combinação de parâmetros ótima, que maximize a transferência de calor entre os tubos e o escoamento livre, para um certo espaço disponível, representado pela restrição de volume fixo, LHW.

A parte crítica de um estudo de otimização consiste na identificação dos graus de liberdade (variáveis) que permitam a maximização (ou minimização) da figura de mérito escolhida, no caso, a maximização da transferência de calor entre os tubos e o escoamento livre,  $q$ . A primeira variável identificada desta maneira foi o espaçamento entre fileiras de tubos,  $S$ . Dado um arranjo de tubos, contendo um número de canais elementares pré-especificado,  $N_{cel}$ , observa-se um espaçamento máximo entre fileiras,  $S_{max}$ , tal que o arranjo seja acondicionado no interior do espaço disponível LHW. Para justificar a escolha do parâmetro  $S$  para ser otimizado, basta que se analise dois extremos:  $S \rightarrow 0$  e  $S \rightarrow S_{max}$ . Para uma determinada condição de velocidade de entrada no arranjo,  $U_{\infty}$ , quando  $S \rightarrow 0$ , a área mínima da seção transversal de escoamento,  $A_c$ , se reduz e, portanto, o fluxo de fluido no canal elementar cai, e desta maneira  $q \rightarrow 0$ . Quando  $S \rightarrow S_{max}$ ,  $A_c$  aumenta, a velocidade na seção transversal do escoamento se reduz e, portanto,  $q$  também diminui. Este comportamento claramente indica a existência de um valor máximo para  $q$  no intervalo  $0 < S < S_{max}$  desde que  $S_{max}$  seja suficientemente grande.

### 3.2 Critério de comparação entre tubos de seção circular e elíptica

Para o estabelecimento de uma base comparativa entre tubos de seção circular e elíptica adotou-se o critério de manter-se a mesma seção dos tubos na direção do escoamento, ou seja, o diâmetro do tubo circular igual a duas vezes o semi-eixo menor do tubo elíptico conforme a Figura 2. Desta maneira, todos os arranjos sob comparação tiveram a mesma área de obstrução ao escoamento livre.

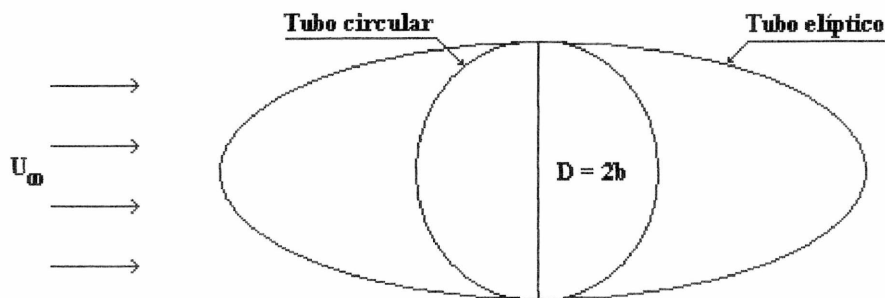


Figura 2: Superfícies de troca de calor.

### 3.3 Disposição dos tubos

A Figura 3 mostra uma distribuição triangular equilateral dos tubos no canal elementar.

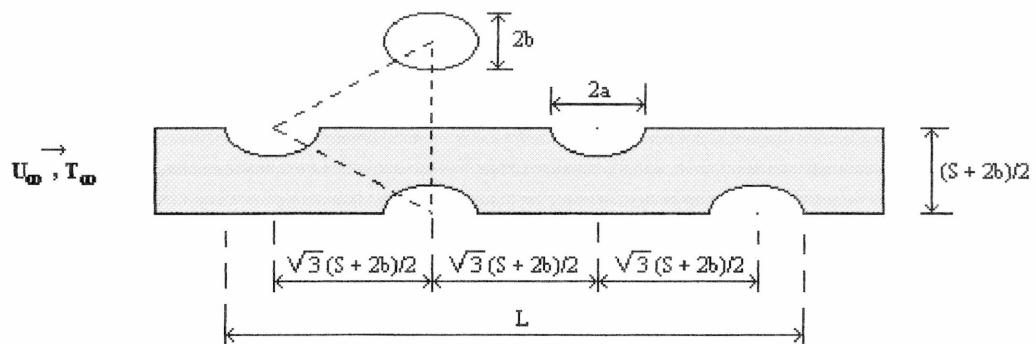


Figura 3: Distribuição triangular equilateral para 4 tubos no canal elementar.

A geometria da Figura 3 permite escrever a seguinte expressão geral:

$$\frac{L}{2b} = \frac{a}{b} + (N-1) \left( \frac{S}{2b} + 1 \right) \cos 30^\circ \quad (1)$$

A partir de  $N$  e  $L/2b$  determina-se  $S/2b$  com a equação (1), parâmetro este utilizado para a montagem das malhas e determinação do espaçamento ótimo entre fileiras de tubos.

Para o estudo da seção elíptica, a restrição de distribuição triangular equilátera não foi adotada. Portanto, variou-se somente a altura do canal elementar do arranjo  $(S+2b)/2$ .

## 4 MODELO MATEMÁTICO

O problema foi analisado partindo das seguintes hipóteses:

- Fluido newtoniano;
- Escoamento incompressível;
- 2-D (bi-dimensional);
- Regime permanente;
- Escoamento laminar;
- Propriedades constantes no fluido, e
- Dissipação viscosa desprezível.

O ar comporta-se como um fluido newtoniano aonde a tensão cisalhante é proporcional ao gradiente de velocidade e devido a pequenas variações da temperatura, consideram-se as propriedades constantes no fluido e a dissipação viscosa desprezível.

O escoamento pode ser tratado como incompressível por desenvolver-se em baixas velocidades (escoamento laminar) e pequenas variações de pressão, onde a densidade e a viscosidade também permanecem constantes.

A influência do efeito das superfícies do volume LHW é desprezível, uma vez que a altura do canal elementar  $(S + 2b)/2$  é consideravelmente menor que a altura do arranjo H, portanto, o escoamento pode ser tratado como bi-dimensional.

### 4.1 Equações governantes

Partindo-se das equações básicas de conservação de massa, momento e energia, utilizando as hipóteses apresentadas, as equações resultantes, simplificadas que governam o problema são:

#### 4.1.1 Conservação de massa

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \quad (2)$$

### 4.1.2 Conservação de momentum

- Componente na direção x:

$$u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} = -\frac{1}{\rho} \frac{\partial p}{\partial x} + \nu \left( \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) \quad (3)$$

- Componente na direção y:

$$u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} = -\frac{1}{\rho} \frac{\partial p}{\partial y} + \nu \left( \frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right) \quad (4)$$

### 4.1.3 Conservação de energia

$$u \frac{\partial T}{\partial x} + v \frac{\partial T}{\partial y} = \alpha \left( \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right) \quad (5)$$

Devido às simetrias existentes nos arranjos, o problema pode ser resolvido em duas dimensões, com o domínio reduzido a apenas uma célula unitária, mostrada na Figura 1.

## 4.2 Forma adimensional

A fim de que os resultados sejam de caráter geral e para facilitar a obtenção da solução numérica das equações, grupos adimensionais convenientes foram definidos para a representação das quantidades de interesse, i.e., temperaturas, velocidades e pressão, fazendo uso também do número de Prandtl, correspondente ao fluido, e do número de Reynolds baseado na velocidade de entrada do fluido no arranjo e no comprimento total varrido pelo escoamento,  $L$ .

Adimensionalizando as variáveis envolvidas nas equações, obtém-se:

- Velocidades:

$$U = \frac{u}{U_{\infty}} \quad (6)$$



$$V = \frac{v}{U_{\infty}} \quad (7)$$

- Posições:

$$X = \frac{x}{L} \quad (8)$$

$$Y = \frac{y}{L} \quad (9)$$

- Pressão:

$$P = \frac{p}{\rho U_{\infty}^2} \quad (10)$$

- Perda de carga:

$$\Delta P = \frac{\Delta p}{\rho U_{\infty}^2 / 2} \quad (11)$$

- Temperatura:

$$\theta = \frac{T - T_{\infty}}{T_w - T_{\infty}} \quad (12)$$

- Número de Reynolds:

$$Re_L = \frac{U_{\infty} L}{\nu} \quad (13)$$

- Número de Peclet:

$$Pe_L = \frac{U_{\infty} L}{\alpha} \quad (14)$$

- Número de Prandtl:

$$\text{Pr} = \frac{\nu}{\alpha} \quad (15)$$

#### 4.3 Equações governantes adimensionais

Substituindo os grupos adimensionais nas equações governantes, as equações governantes adimensionais resultantes são:

##### 4.3.1 Conservação de massa

$$U \frac{\partial U}{\partial X} + V \frac{\partial V}{\partial Y} = 0 \quad (16)$$

##### 4.3.2 Conservação de momentum

- Componente na direção x:

$$U \frac{\partial U}{\partial X} + V \frac{\partial U}{\partial Y} = -\frac{\partial P}{\partial X} + \frac{1}{\text{Re}_L} \left[ \frac{\partial^2 U}{\partial X^2} + \frac{\partial^2 U}{\partial Y^2} \right] \quad (17)$$

- Componente na direção y:

$$U \frac{\partial V}{\partial X} + V \frac{\partial V}{\partial Y} = -\frac{\partial P}{\partial Y} + \frac{1}{\text{Re}_L} \left[ \frac{\partial^2 V}{\partial X^2} + \frac{\partial^2 V}{\partial Y^2} \right] \quad (18)$$

##### 4.3.3 Conservação de energia

$$U \frac{\partial \theta}{\partial X} + V \frac{\partial \theta}{\partial Y} = \frac{1}{\text{Pe}_L} \left[ \frac{\partial^2 \theta}{\partial X^2} + \frac{\partial^2 \theta}{\partial Y^2} \right] \quad (19)$$

#### 4.4 Condições de contorno dimensionais

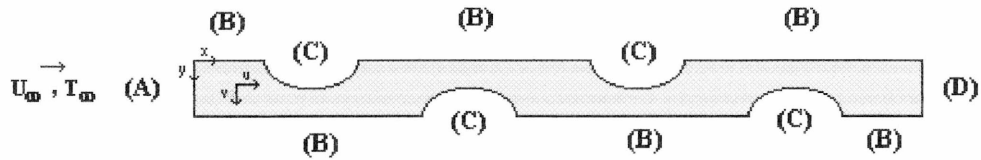
A seguir, são estabelecidas condições de contorno apropriadas ao problema, isto é, temperaturas prescritas nos tubos, nenhuma penetração nas interfaces de canal para canal, e condição de não deslizamento nas superfícies dos tubos.

$$(A) \quad U = 1; \quad V = 0; \quad \theta = 0; \quad (20)$$

$$(B) \quad \frac{\partial U}{\partial Y} = 0; \quad V = 0; \quad \frac{\partial \theta}{\partial Y} = 0; \quad (21)$$

$$(C) \quad U = V = 0; \quad \theta = 1; \quad (22)$$

$$(D) \quad \frac{\partial U}{\partial X} = \frac{\partial V}{\partial X} = \frac{\partial \theta}{\partial X} = 0; \quad (23)$$



**Figura 4: Condições de contorno no canal elementar.**

## 5 MÉTODO NUMÉRICO

### 5.1 Considerações iniciais

Para a solução numérica das Equações (16)-(19), foi aplicado o método de elementos finitos (Zienkiewicz e Taylor, 1989). Desta maneira, obtêm-se os campos de velocidades e temperaturas no domínio definido pela célula unitária da Figura 1.

O código computacional dos elementos específicos para a solução das equações de conservação, em linguagem FORTRAN, foi desenvolvido com base na forma variacional ("weak form") das Equações (16)-(19) (Reddy e Gartling, 1994), constituindo-se em elementos bidimensionais, quadrilaterais, isoparamétricos, com funções de interpolação lineares. Este código foi escrito e agregado ao código aberto do método de elementos finitos "FEAP - Finite Element Analysis Program" (Zienkiewicz e Taylor, 1989) pelo Grupo de Energia e Ciências Térmicas do PIPE, UFPR.

Para possibilitar o cálculo do fluxo de calor entre os tubos e o escoamento do fluido foi necessário implementar um código computacional específico para realizar o pós-processamento das temperaturas obtidas na solução do problema no domínio computacional representado pelo canal elementar da Figura 1.

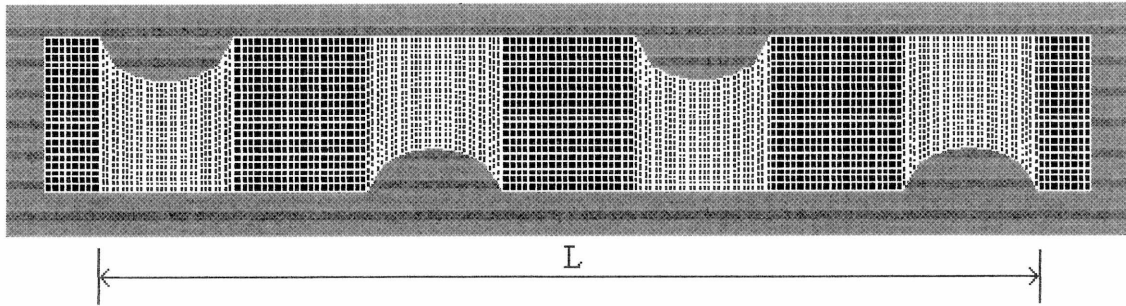
Para validação do cálculo do fluxo de calor foi realizado um teste do cálculo do fluxo de calor em uma cavidade quadrada utilizando o código desenvolvido nesta tese, conforme demonstrado no Anexo 1, comparando os valores obtidos com a solução numérica e a solução obtida por outros métodos publicados na literatura técnica.

Além disso o código foi validado, via comparação direta dos resultados numéricos para arranjos de tubos circulares com os resultados experimentais de Stanescu et al. (1996). O problema de obtenção dos campos de temperaturas e velocidades em arranjos de tubos elípticos foi resolvido numericamente para várias configurações. Foi computada a densidade volumétrica de transferência de calor de cada arranjo. Desta maneira, realizou-se a otimização numérica do espaçamento entre os tubos dos arranjos para máxima transferência de calor, para uma determinada restrição de espaço (condição de volume fixo).

O cálculo da densidade volumétrica de transferência de calor ou condutância térmica média de cada arranjo foi determinado, dividindo a taxa de geração total de calor nos tubos,  $Q$ , pelo volume total disponível para instalação do arranjo, LHW (condição de volume fixo).

## 5.2 Malha utilizada

Para efeito de verificação da teoria apresentada após o estudo de diversas configurações de malhas, chegou-se mediante testes de refinamento a uma malha com 5180 elementos, 4 nós por elemento, totalizando 5460 nós, sendo 260 nós na direção x e 21 nós na direção y, com quatro tubos no canal elementar.



**Figura 5: Malha gerada através do código FEAP para 4 tubos no canal elementar.**

Para permitir a estabilização do escoamento a montante e a jusante do arranjo foi acrescentado um comprimento  $L$  a esquerda e a direita da malha da Figura 5. Através de testes numéricos, constatou-se que para este problema não houve variação significativa no cálculo do fluxo total de calor utilizando-se um comprimento de  $L/2-2a$  ao invés de  $L$ . No entanto, para obtenção dos resultados apresentados nesta tese, a malha teve o comprimento total  $3L$ .

## 5.3 Teste de refinamento da malha

Para todos os arranjos testados, foi realizado um teste de refinamento de malha com 5380 elementos, 4 nós por elemento, totalizando 5670 nós, sendo 270 nós na direção x e 21 nós na direção y, com quatro tubos no canal elementar, cujo erro entre a malha anterior (5460 nós e 5180 elementos) e a malha posterior é dado por:

$$\text{erro} = \left| \tilde{q}_{\text{malha anterior}} - \tilde{q}_{\text{malha posterior}} \right| / \left| \tilde{q}_{\text{malha anterior}} \right| \quad (24)$$

onde  $\tilde{q}$  representa a densidade volumétrica de transferência de calor do arranjo adimensional, a ser definida na Equação (58).

O erro calculado pela Equação (24) situou-se sempre abaixo de 1 % em todos os casos testados. Portanto, adotou-se a malha menos refinada para a obtenção de todos os resultados deste trabalho.

#### 5.4 Equações de Navier-Stokes e Energia em Elementos Finitos

As equações implementadas no código FEAP (Finite Element Analysis Program), através do elemento Navier-Stokes e Energia encontram-se na forma generalizada:

$$\nabla \cdot \bar{\mathbf{u}} = 0 \quad (25)$$

$$\bar{\rho} \left( \frac{\partial \bar{\mathbf{u}}}{\partial t} + (\bar{\mathbf{u}} \cdot \nabla) \bar{\mathbf{u}} \right) = -\nabla p + \bar{\mu} \nabla^2 \bar{\mathbf{u}} + \bar{\rho} \bar{g} \bar{\beta} (T - T_{\infty}) \quad (26)$$

$$\bar{\rho} \bar{c}_p \bar{\mathbf{u}} \cdot \nabla T = \bar{k} \nabla^2 T \quad (27)$$

onde a pressão foi eliminada da Equação (25), fazendo uso de um modelo de penalidade para elementos finitos, aproximando a equação de conservação de massa para (Hughes et al., 1979):

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = -\frac{p}{\gamma} \quad (28)$$

onde  $\gamma$  - fator de penalidade, deve ser assumido grande o suficiente para que a equação de conservação de massa seja satisfeita aproximadamente.

A implementação do método de elementos finitos para solução das equações (25) – (27) requer primeiramente a obtenção da forma variacional (fraca) do problema. A seguir, a forma variacional resultante é discretizada, inicialmente por um método de Galerkin (i.e., análogo a diferenças centrais no método de diferenças finitas). No entanto, devido às características do escoamento no arranjo da Figura 1, fisicamente a solução depende preferencialmente do que ocorre nos pontos anteriores a um determinado ponto do domínio, considerando a direção do escoamento forçado. O método de Galerkin não captura este aspecto físico em suas equações discretas (análogo de diferenças centrais, no método de diferenças finitas). Portanto, a forma discreta utilizada neste trabalho foi a proposta por

Hughes (1978), que consiste num esquema “upwinding”, onde é possível adequar a forma discreta às características do escoamento.

Após o desenvolvimento da forma discreta, obtém-se a forma matricial para o problema bidimensional em regime permanente:

$$\begin{bmatrix} C(u) & 0 \\ 0 & C(u) \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \end{Bmatrix} + \begin{bmatrix} 2K_{11} + K_{22} & K_{21} \\ K_{12} & K_{11} + 2K_{22} \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \end{Bmatrix} + \begin{bmatrix} \hat{K}_{11} & \hat{K}_{12} \\ \hat{K}_{21} & \hat{K}_{22} \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \end{Bmatrix} = \begin{Bmatrix} F_1 \\ F_2 \end{Bmatrix} \quad (29)$$

$$D(u)T + LT = G \quad (30)$$

onde  $C(u)$  é a matriz de capacidade que contém os termos advectivos das equações de momentum, que depende de forma não linear da solução  $u$  (vetor biparticionado em  $u_1$  – direção  $x$  e  $u_2$  – direção  $y$ , cada um deles com um número de componentes igual ao número de incógnitas da malha);  $K_{11}$ ,  $K_{12}$ ,  $K_{21}$ , e  $K_{22}$  são as matrizes rigidez de coeficientes constantes (propriedades constantes) que contém os termos viscosos das equações de momentum;  $\hat{K}_{11}$ ,  $\hat{K}_{12}$ ,  $\hat{K}_{21}$ , e  $\hat{K}_{22}$  são as matrizes de penalidade, que contém os termos devido à eliminação da pressão das equações de momentum com o uso da equação (28) (para obtenção dos elementos destas matrizes adotou-se um procedimento de integração reduzida, pois estes termos resultam do campo de pressão, a fim de evitar o fenômeno de “locking”);  $D(u)$  é a matriz de capacidade que contém os termos advectivos da equação da energia;  $L$  é a matriz de difusão de coeficientes constantes, que contém os termos difusivos da equação da energia;  $F_1$  e  $F_2$  são os vetores força da equação de momentum que contém as forças de campo e condições de contorno de velocidade, e  $G$  o vetor força da equação da energia que contém os termos de geração de calor e condições de contorno de temperatura.

Deixa-se de apresentar os detalhes matemáticos dos componentes dessas matrizes por razões de brevidade, uma vez que o assunto desta tese não é o método de elementos finitos propriamente dito. No entanto, para maiores detalhes da formulação utilizada para codificar o elemento deste trabalho, deve-se consultar o trabalho de Reddy e Gartling (1994).

O sistema de equações não-lineares construído com as equações (29) e (30) foi resolvido pelo método de Newton-Raphson, para a obtenção das velocidades e temperaturas, para a malha utilizada (Reddy e Gartling, 1994).

Visando a correspondência entre as equações governantes adimensionais e as equações implementadas no código FEAP, torna-se necessário a correlação que se segue para o arquivo de entrada de dados:

Para os números de Reynolds e Peclet baseados no comprimento  $L$  temos a seguinte relação de equivalência:

$$Re_L = Re_D \cdot L/D; \quad Pe_L = Pe_D \cdot L/D \quad (31)$$

Exemplificando, para  $L/D = 6.2$ ,  $Re_D = 100$ ,  $Pr = 0.72$  temos:

$$\bar{\mu} = 0.0161; \gamma = 1.e6; \bar{\varrho} = 1; l' = 2; k' = 2$$

$$\alpha_p = 0; \bar{g}_x = 0; \bar{g}_y = 0; \bar{\beta} = 0; \bar{c}_p = 1; \bar{k} = 0.00224$$

Note que:

$$\bar{\mu} = \frac{1}{Re_L}; \quad Pe_L = Re_L \cdot Pr; \quad \bar{k} = \frac{1}{Pe_L} \text{ e } \alpha_p - \text{parâmetro de "upwinding" (Hughes, 1978).}$$

### 5.5 Algoritmo para o cálculo da transferência de calor, em tubos de seção circular

Inicialmente procedemos o cálculo da transferência de calor para cada nó das superfícies que envolvem os tubos.

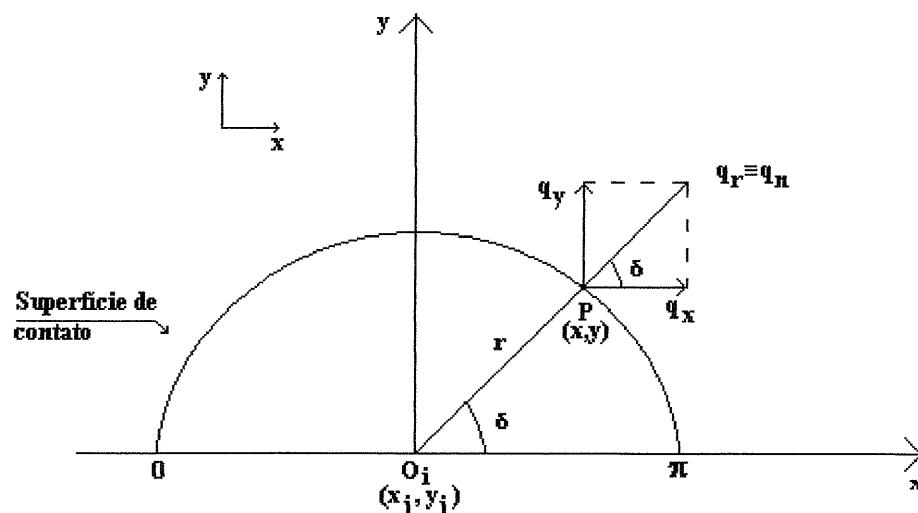


Figura 6: Coordenadas polares no círculo.



A partir da Álgebra Vetorial, calcula-se o módulo do vetor  $q_n$  como se segue:

$$|q_n| = q_x \cdot n + q_y \cdot n \quad (32)$$

onde  $n$  – vetor normal à superfície.

Considerando a função  $f$  uma superfície genérica, o vetor normal  $n$  é dado por:

$$n = \frac{\nabla f}{|\nabla f|} \quad (33)$$

A função  $f$  para o círculo é dada por:

$$f = (x - x_i)^2 + (y - y_i)^2 - r^2 = 0 \quad (34)$$

Temos que o gradiente de  $f$  é dado por:

$$\nabla f = \left( \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right) \quad (35)$$

$$\nabla f = (2(x - x_i), 2(y - y_i)) \quad (36)$$

E o módulo do gradiente de  $f$  é dado por:

$$|\nabla f| = \sqrt{\left( \frac{\partial f}{\partial x} \right)^2 + \left( \frac{\partial f}{\partial y} \right)^2} \quad (37)$$

Substituindo as derivadas:

$$|\nabla f| = 2\sqrt{(x - x_i)^2 + (y - y_i)^2} \quad (38)$$

Das coordenadas retangulares:

$$x - x_i = r \cos \delta \quad (39)$$

$$y - y_i = r \sin \delta \quad (40)$$

Substituindo, as equações (39) e (40) na equação (38):

$$|\nabla f| = 2\sqrt{r^2 \cos^2 \delta + r^2 \sin^2 \delta} \quad (41)$$

Simplificando:

$$|\nabla f| = 2r \quad (42)$$

Substituindo, as equações (42) e (36) na equação (33) e simplificando:

$$n = (\cos \delta, \sin \delta) \quad (43)$$

Substituindo, (42) em (30) temos a equação resultante para o cálculo da transferência de calor normal à superfície de troca de calor:

$$|q_n| = |q_x| \cos \delta + |q_y| \sin \delta \quad (44)$$

O ângulo  $\delta$  foi obtido através da relação, a seguir, para coordenadas polares:

$$\delta = \arctg \left[ \frac{y_i - y}{x_i - x} \right] \quad (45)$$

O que permite a obtenção do ângulo  $\delta$  para qualquer posição  $(x,y)$  na superfície circular.

### 5.6 Algoritmo para integração da transferência de calor ao longo da superfície do tubo

Da transferência de calor, aplicando-se a lei de Fourier na interface fluido /sólido (condução pura) temos que  $q_n''$  é dado por:

$$q_n'' = -k \frac{\partial T}{\partial n}, \quad q_x'' = -k \frac{\partial T}{\partial x} \quad \text{e} \quad q_y'' = -k \frac{\partial T}{\partial y} \quad (46)$$

Combinando as equações (44) e (46), obtém-se:

$$|q_n''| = \left| -k \frac{\partial T}{\partial x} \right| \cos \delta + \left| -k \frac{\partial T}{\partial y} \right| \sin \delta \quad (47)$$

Usando as equações (8), (9) e (12), define-se:

$$F = \frac{|q_n''| L}{k(T_w - T_\infty)} = \left| \frac{\partial \theta}{\partial X} \right| \cos \delta + \left| \frac{\partial \theta}{\partial Y} \right| \sin \delta \quad (48)$$

Fazendo a média ao longo da superfície de troca de calor e integrando de 0 a  $\pi$ :

$$\frac{|\bar{q}_n''| L}{k(T_w - T_\infty)} = \frac{1}{\pi} \int_0^\pi F \, d\delta \quad (49)$$

onde a integral I ao longo da superfície é representada por:

$$I = \frac{|\bar{q}_n''| L \pi}{k(T_w - T_\infty)} = \int_0^\pi F \, d\delta \quad (50)$$

A seguir, a taxa de geração de calor por unidade de volume total do arranjo, é dada por:

$$\frac{Q}{W L H} = \frac{N_{cel} q_{1cel}}{W L H} \quad (51)$$

Simplificando a equação (51):

$$Q = N_{cel} q_{1cel} \quad (52)$$

A transferência de calor média por unidade de área por superfície do canal elementar é dada por:

$$\left| \bar{q}_{n,i}'' \right| = \frac{(q_{1cel})_i}{\frac{\pi D}{2} W} \quad (53)$$

onde  $i = 1, \dots, N$  e  $N$  representa o número de superfícies em contato com o canal elementar.

A Função Objetivo representada pela transferência de calor volumétrica adimensional por canal elementar é definida como:

$$\tilde{q}_i = \frac{Q_i}{k (T_w - T_\infty) H W L / D^2} \quad (54)$$

Substituindo, a equação (52) na equação (54):

$$\tilde{q}_i = \frac{(N_{cel} q_{1cel})_i}{k (T_w - T_\infty) H W L / D^2} \quad (55)$$

Substituindo, a equação (53) na equação (55):

$$\tilde{q}_i = \frac{N_{cel} \left| \bar{q}_{n,i}'' \right| \frac{\pi D}{2} W}{k (T_w - T_\infty) H W L / D^2} \quad (56)$$

Simplificando e multiplicando a equação (56) por  $L/L$ :

$$\tilde{q}_i = \frac{|\tilde{q}_{n,i}| L \pi}{k (T_w - T_\infty)} \frac{\left[\frac{D}{L}\right]^2}{\frac{H}{D}} \frac{N_{cel}}{2} \quad (57)$$

Substituindo  $I_i$  da equação (50) na equação (57):

$$\tilde{q}_i = I_i \frac{\left[\frac{D}{L}\right]^2}{\frac{H}{D}} \frac{N_{cel}}{2} \quad (58)$$

onde:

$$\tilde{q} = \sum_{i=1}^N \tilde{q}_i \quad (59)$$

A integral  $I_i$  foi implementada no código FEAP, que passou a computar diretamente com as equações (58) e (59) o valor de  $\tilde{q}$  para cada geometria de arranjo estudada.

O cálculo de  $\tilde{q}$  foi realizado neste trabalho apenas para tubos circulares, a fim de permitir a validação de resultados numéricos por comparação direta com resultados experimentais (Stanescu et al., 1996).

### 5.7 Metodologia para o cálculo da transferência de calor em tubos de seção circular e elíptica

No presente trabalho, foi definida uma nova figura de mérito para o cálculo da transferência de calor utilizando a temperatura no final do canal elementar com o objetivo de obter maior precisão dos resultados numéricos.

Parte-se das equações (51) e (52) que representam a taxa de geração de calor por unidade de volume total do arranjo e utiliza-se como referência o fluxo de calor volumétrico a seguir:

$$q_{ref}''' = k \frac{(T_w - T_{\infty})}{(2b)^2} \left[ \frac{W}{m^3} \right] \quad (60)$$

A Função Objetivo representada pela transferência de calor volumétrica adimensional é dada por:

$$\tilde{q}_* = \frac{Q/LHW}{q_{ref}'''} \quad (61)$$

Substituindo, a equação (51) na equação (61), obtêm-se:

$$\tilde{q}_* = \frac{N_{cel} q_{1cel}}{k (T_w - T_{\infty}) LHW / (2b)^2} \quad (62)$$

A transferência de calor para uma célula é obtida por um balanço global de energia tomando uma célula como volume de controle:

$$q_{1cel} = \dot{m} c_p (\bar{T}_s - T_{\infty}) \quad (63)$$

onde  $\bar{T}_s$  – temperatura média do fluido na seção de saída da célula unitária.

A vazão mássica de ar é calculada na seção de entrada da célula unitária como:

$$\dot{m} = \rho U_{\infty} \left( \frac{S + 2b}{2} \right) W \quad (64)$$

Substituindo, as equações (63) e (64) na equação (62):

$$\tilde{q}_* = \frac{N_{cel} \rho U_{\infty} \left( \frac{S + 2b}{2} \right) W c_p (\bar{T}_s - T_{\infty})}{k (T_w - T_{\infty}) LHW / (2b)^2} \quad (65)$$

Simplificando e substituindo a equação (12) em (65):

## 6 DISCUSSÃO DE RESULTADOS

### 6.1 Validação dos resultados numéricos para arranjos de tubos circulares

Nesta seção, busca-se obter resultados numéricos para um arranjo de tubos de dimensões idênticas às utilizadas no experimento de Stanescu et al. (1996) dadas por:

$$L = 39.2 \text{ mm}, H = 35.2 \text{ mm}, W = 134 \text{ mm}, D = 6.35 \text{ mm} \text{ e a relação } L/D = 6.2.$$

Estabelecemos para o nosso estudo comparativo um arranjo com doze tubos no volume fixo LHW de referência, e quatro tubos no canal elementar. Desta maneira, é possível investigar qual o espaçamento ótimo entre fileiras de tubos, em um arranjo de geometria conhecida ou especificada.

Consideram-se estes dados como a base do arranjo utilizado na experimentação numérica, dimensões estas que adimensionalizadas segundo o comprimento L do arranjo resultam em:

$$\tilde{L} = 1.0$$

$$\tilde{H} = 0.9678$$

$$\tilde{W} = 3.4$$

$$\tilde{D} = 0.1613$$

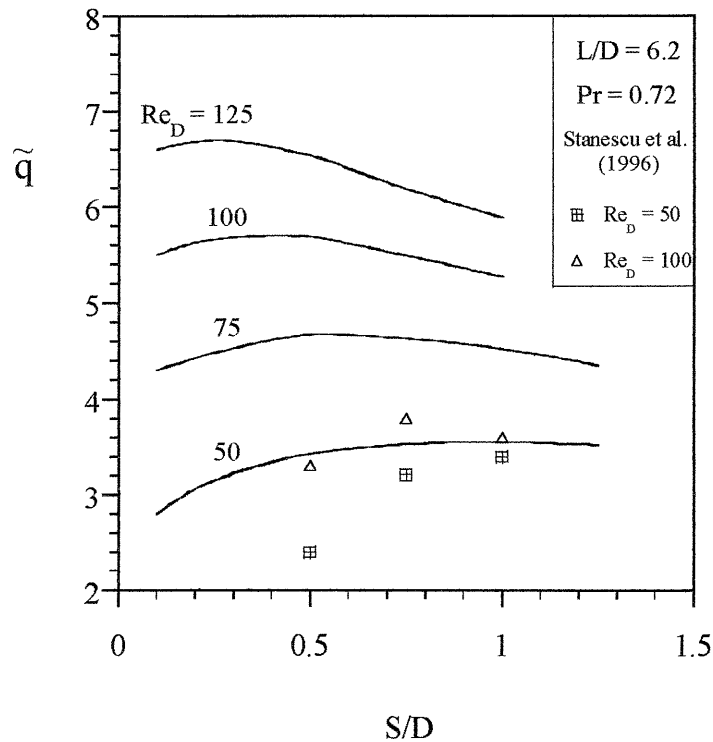
$$L/D = 6.2$$

$$H/D = 6.0$$

Mantendo a restrição de triângulo equilátero para o arranjo, utilizando a equação (1), onde  $a = b$ ,  $D = 2b$ ,  $N = 4$ ,  $N_{\text{cel}} = 6$  e  $L/D = 6.2$ , obtém-se  $S/D = 1.0$ .

Partiu-se inicialmente da relação  $S/D = 1.0$  máxima para o arranjo. Este espaçamento entre tubos foi então reduzido para um mesmo número de tubos no arranjo (12 tubos) até que fosse encontrado o espaçamento ótimo dado pela máxima transferência de calor.

Através do código FEAP foram obtidos os fluxos de calor, segundo a equação (59) para vários valores de  $S/D$  e  $Re_L = 310, 465, 620$  e  $775$ , cujos valores de  $(S/D)_{\text{ot}}$  encontram-se entre 0.25 e 1.0, de acordo com o gráfico da Figura 7.



**Figura 7: Gráfico dos resultados numéricos e experimentais para trocadores de calor de tubos circulares com arranjo triangular equilateral.**

A Figura 7 mostra que os resultados numéricos obtidos neste trabalho apresentam concordância qualitativa com os resultados experimentais de Stanescu et al. (1996), para  $Re_D = 50$  e 100 principalmente, no que se refere à localização do espaçamento ótimo  $(S/D)_{ot}$ .

A concordância é marcante, observando também que o arranjo experimental era pequeno e, portanto, não era um grande banco de cilindros. Nos experimentos, utilizou-se fluxo de calor uniforme nos tubos, e na simulação numérica os arranjos são infinitamente mais largos e com cilindros isotérmicos.

Realizada a validação dos resultados numéricos para tubos circulares, parte-se a seguir para a otimização dos arranjos de tubos elípticos.

## 6.2 Otimização de arranjos de tubos de seção elíptica

De acordo com o critério de equivalência entre tubos de seção circular e elíptica e a disposição dos tubos (variando somente  $(S+2b)/2$ ), tratados nos itens 3.2 e 3.3 deste trabalho,



parte-se dos seguintes valores adimensionalizados para a otimização dos arranjos de tubos de seção elíptica:

$$\tilde{L}=1.0$$

$$L/2b=6.2$$

$$S/2b=1.0$$

Para as excentricidades :

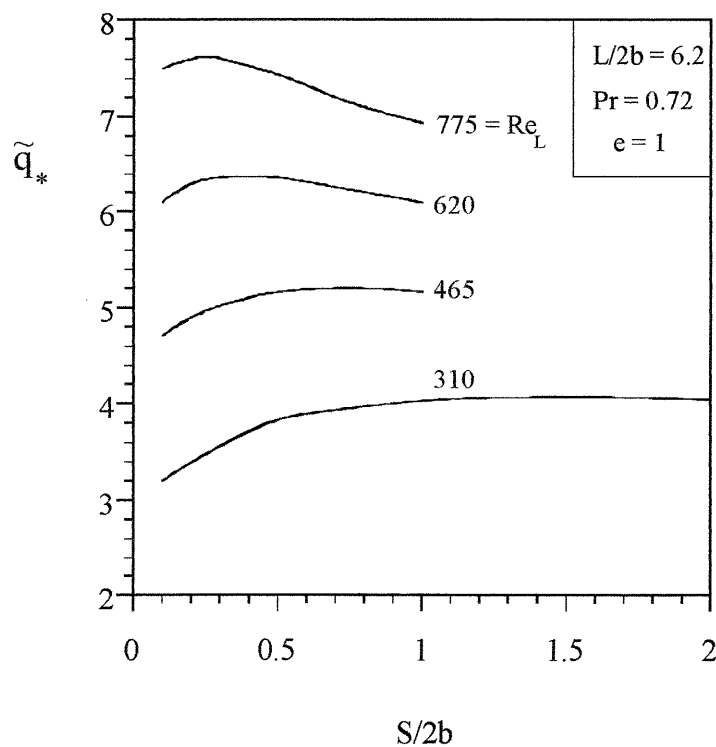
$e = 1.0$ ,  $\tilde{a} = 0.08065$  e  $\tilde{b} = 0.08065$  seção circular;

$e = 0.80$ ,  $\tilde{a} = 0.1241$  e  $\tilde{b} = 0.08065$  seção elíptica;

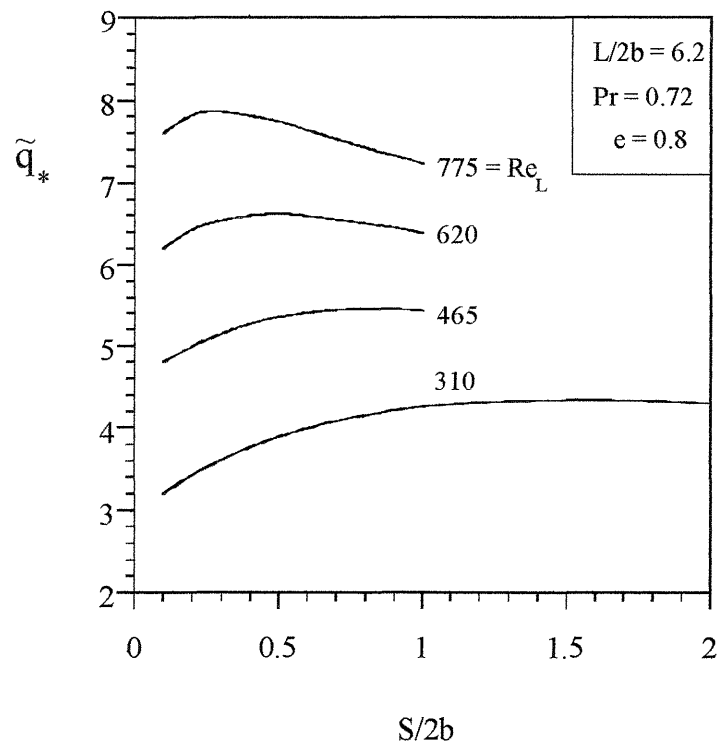
$e = 0.65$ ,  $\tilde{a} = 0.101$  e  $\tilde{b} = 0.08065$  seção elíptica;

obtiveram através do código FEAP os fluxos de calor segundo a nova figura de mérito da equação (67), tratada no item 5.8 para  $Re_L = 310, 465, 620$  e  $775$ , cujos valores ótimos para  $S/2b$  encontram-se entre 0.25 e 1.5 de acordo com os gráficos das Figuras 8, 9 e 10 respectivamente.

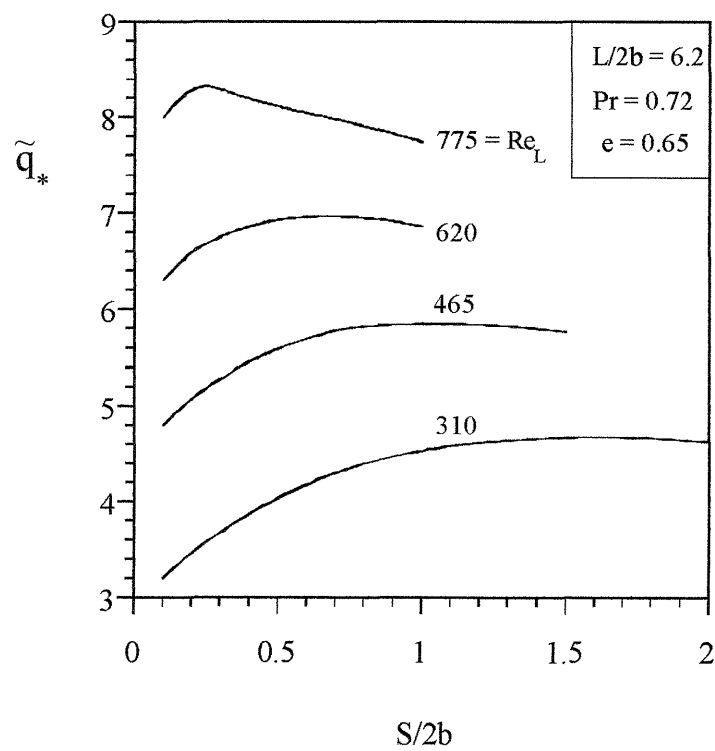
A influência da variação de  $Re_L$  é também investigada nas Figuras 8, 9 e 10. Quando  $Re_L$  aumenta  $\tilde{q}_*$  aumenta. O máximo é menos pronunciado para valores baixos de  $Re_L$ .



**Figura 8:** Gráfico dos resultados numéricos para trocadores de calor de tubos circulares ( $e=1$ ).



**Figura 9: Gráfico dos resultados numéricos para trocadores de calor de tubos elípticos ( $e=0.8$ ).**



**Figura 10: Gráfico dos resultados numéricos para trocadores de calor de tubos elípticos ( $e=0.65$ ).**

### 6.3 Comparação entre tubos de seção circular e elíptica

As figuras 11 e 12 mostram o efeito da excentricidade das elipses sobre  $\tilde{q}_*$ , para  $Re_L = 465$  e 620, respectivamente. Quando a excentricidade decresce,  $\tilde{q}_*$  aumenta, portanto a geometria elíptica tem uma melhor taxa de transferência de calor total entre os tubos e o escoamento livre.

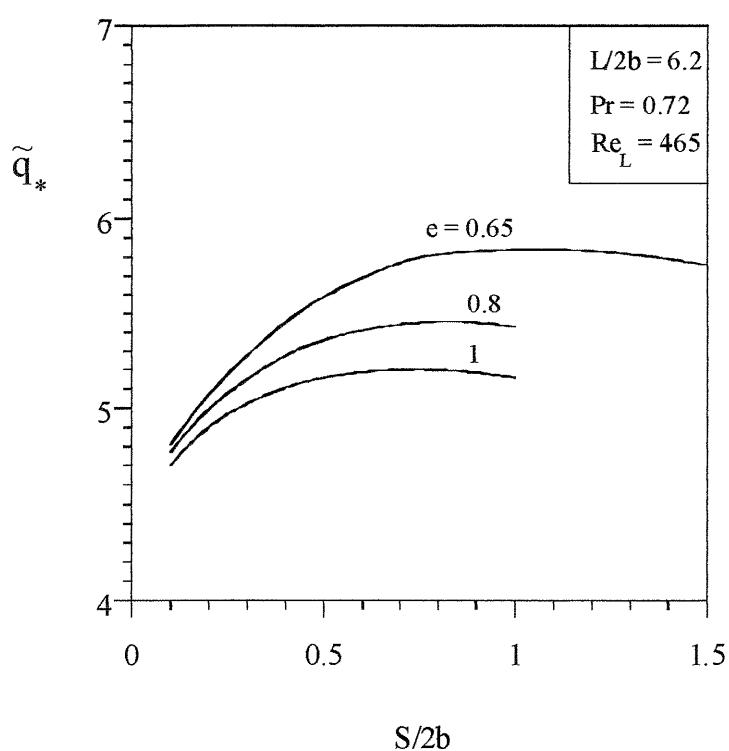
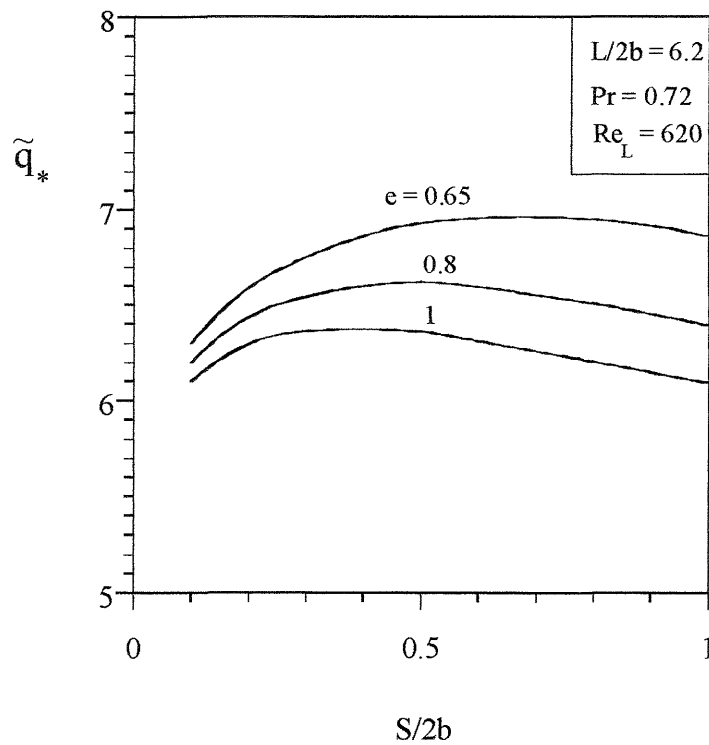


Figura 11: Gráfico do efeito da excentricidade das elipses na transferência de calor ( $Re_L=465$ ).



**Figura 12:** Gráfico do efeito da excentricidade das elipses na transferência de calor ( $Re_L=620$ ).

Os resultados mostrados nas Figuras 8 a 12 são resumidos nas Figuras 13 e 14. O efeito da excentricidade das elipses sobre  $\tilde{q}_{*,\max}$  é representado na Figura 13, onde a medida que  $Re_L$  e  $\tilde{q}_{*,\max}$  aumentam, a excentricidade decresce, isto é, quanto mais alongadas as elipses maior é a transferência de calor total. Em uma análise quantitativa, é importante realçar que foi observado um ganho máximo de 13 % na transferência de calor em comparação com o arranjo circular tradicional, através da simulação numérica, observado para o arranjo elíptico com  $e = 0.65$ .

A Figura 14 mostra que o espaçamento ótimo decresce a medida que a velocidade do escoamento livre aumenta (ou  $Re_L$ ).

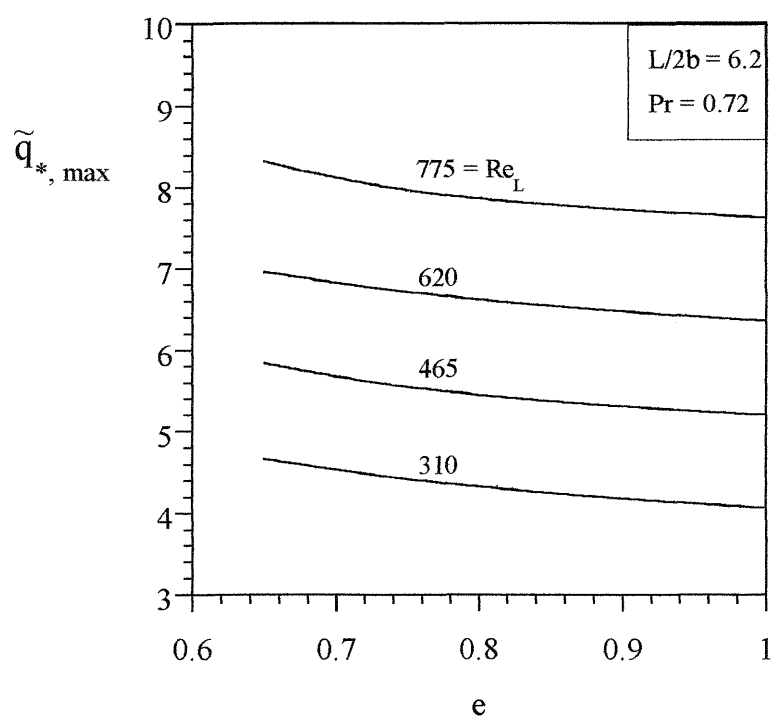


Figura 13: Gráfico do efeito da excentricidade das elipses sobre a máxima condutância térmica global.

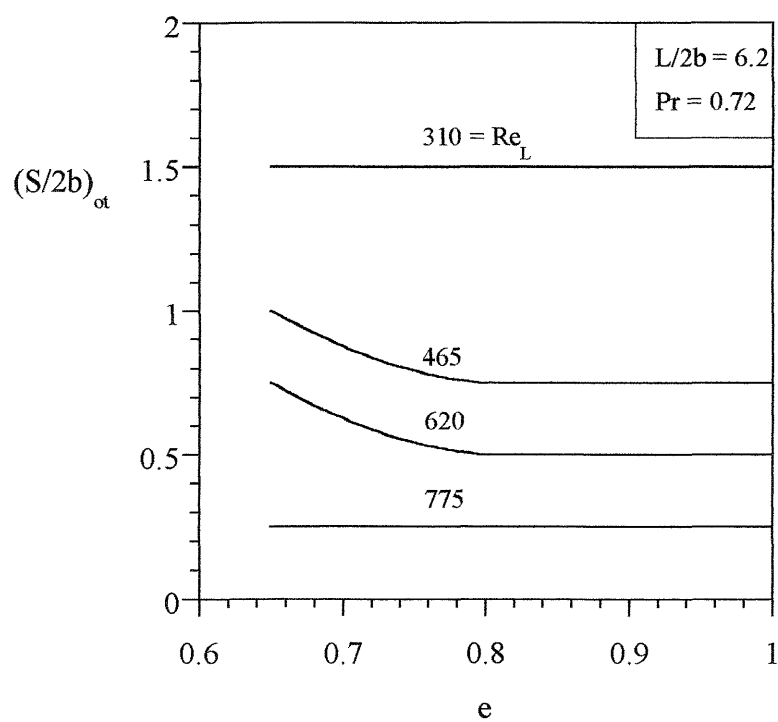


Figura 14: Gráfico do efeito da excentricidade das elipses sobre o espaçamento ótimo para a máxima condutância térmica global.

O valor de  $S/D = 0.1$  foi o valor mínimo possível com o que foram obtidas soluções numéricas. A explicação para tal limitação prende-se ao fato de que para espaçamentos pequenos, ( $S \rightarrow 0$ ), o modelo utilizado não representa o comportamento esperado do ponto de vista físico ( $\tilde{q}_{*, \max} \rightarrow 0$ ), pelo fato de ao se aproximar-se os tubos a velocidade na seção crítica (mínima) aumenta e o escoamento, eventualmente, passa a ser compressível, portanto, não representado pelo modelamento apresentado.

Não houve perda de generalidade dos resultados por fixar-se  $N_{\text{cel}} = 6$  no presente estudo, como é deduzido através das equações (57) e (67). O efeito da variação do número de tubos em um canal elementar,  $N$ , está ainda para ser investigado, mas pode-se notar que  $N = L/a$  representa o limite onde as elipses se tocam. De qualquer modo, não é difícil verificar que a figura de mérito dada pela equação (54) é análoga ao número de Nusselt médio para todo o arranjo,  $\tilde{q} = \overline{Nu} = \frac{\bar{h}(2b)}{k}$ , de modo que,  $\bar{h} = \frac{q(2b)}{(T_w - T_\infty)LHW}$ , onde  $\bar{h}$  representa o coeficiente de transferência de calor médio equivalente,  $W/(m^2.K)$ . Portanto, para números maiores de fileiras,  $\tilde{q}_{\max}$  (ou  $\tilde{q}_{*, \max}$ ) computado para  $N = 4$  é uma boa aproximação. Isto é explicado pelo fato de que com um número grande de fileiras, o escoamento seria completamente desenvolvido, portanto, com nenhuma mudança significativa no número de Nusselt médio para uma geometria particular, seja ela circular ou elíptica. Este comportamento foi observado experimentalmente comparando resultados de arranjos de tubos circulares de três fileiras, apresentados por Saboya e Sparrow (1976), com resultados de arranjos de tubos circulares de duas fileiras, apresentados por Rosman et al. (1984), ambos para trocadores de calor aletados. O mesmo fenômeno foi também observado numericamente em um recente estudo de Fowler et al. (1997), em placas submetidas à convecção forçada, onde conclui-se que o efeito de  $N$  em  $\tilde{q}_{\max}$  é quase inexistente para  $2 \leq N \leq 65$ .

As isotermas mostradas nas Figuras 15, 16 e 17 comprovam o ganho da transferência de calor com o aumento da temperatura no final do canal elementar na faixa de 0.7 a 0.9, com a redução da excentricidade. A medida que a excentricidade é reduzida, o fluido na saída do arranjo fica mais aquecido, portanto,  $\bar{T}_s$  se eleva, e conseqüentemente  $\tilde{q}_*$ .

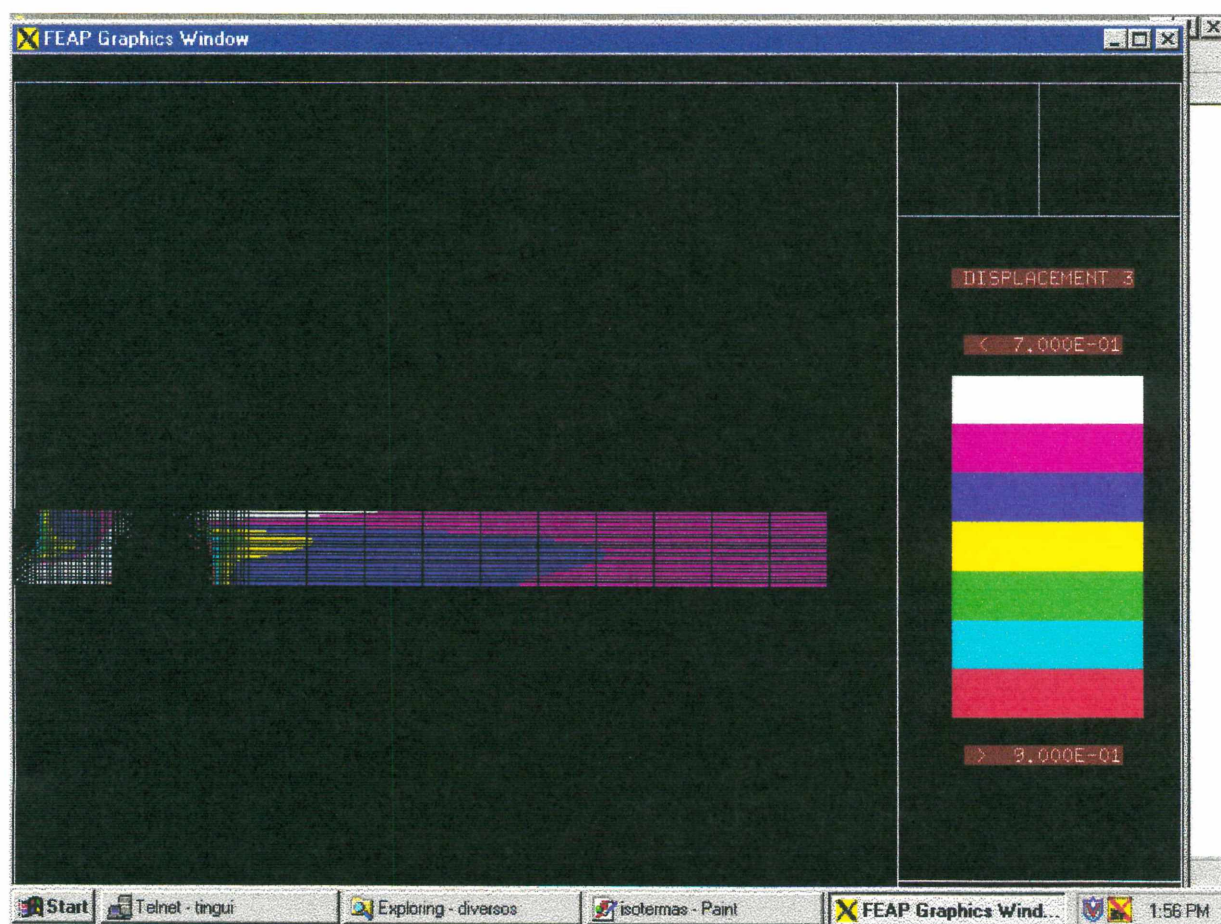


Figura 15: Gráfico das isothermas para o espaçamento ótimo  $S/2b = 0.50$ ,  $e = 1.0$  e  $Re_L = 620$ .

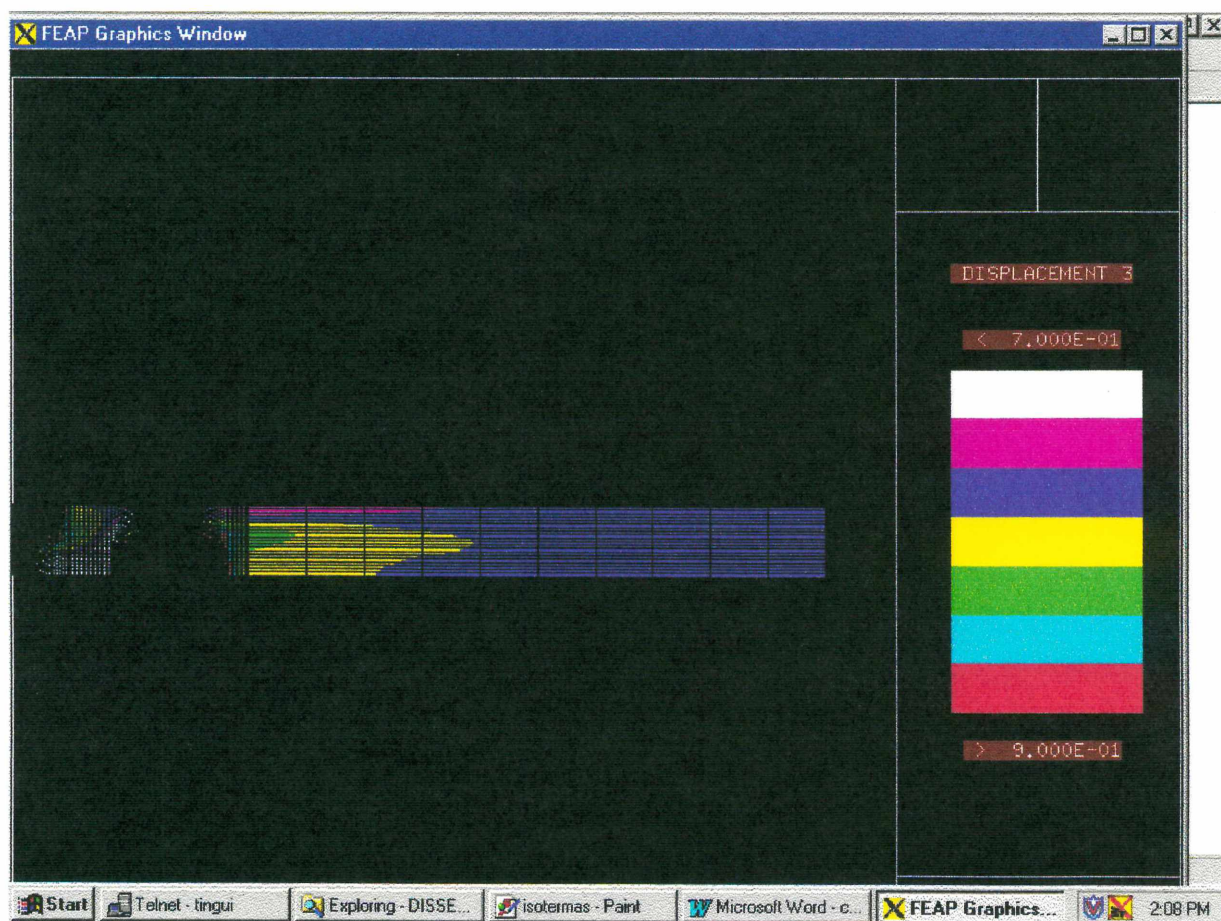


Figura 16: Gráfico das isotermas para o espaçamento ótimo  $S/2b = 0.50$ ,  $e = 0.80$  e  $Re_L = 620$ .



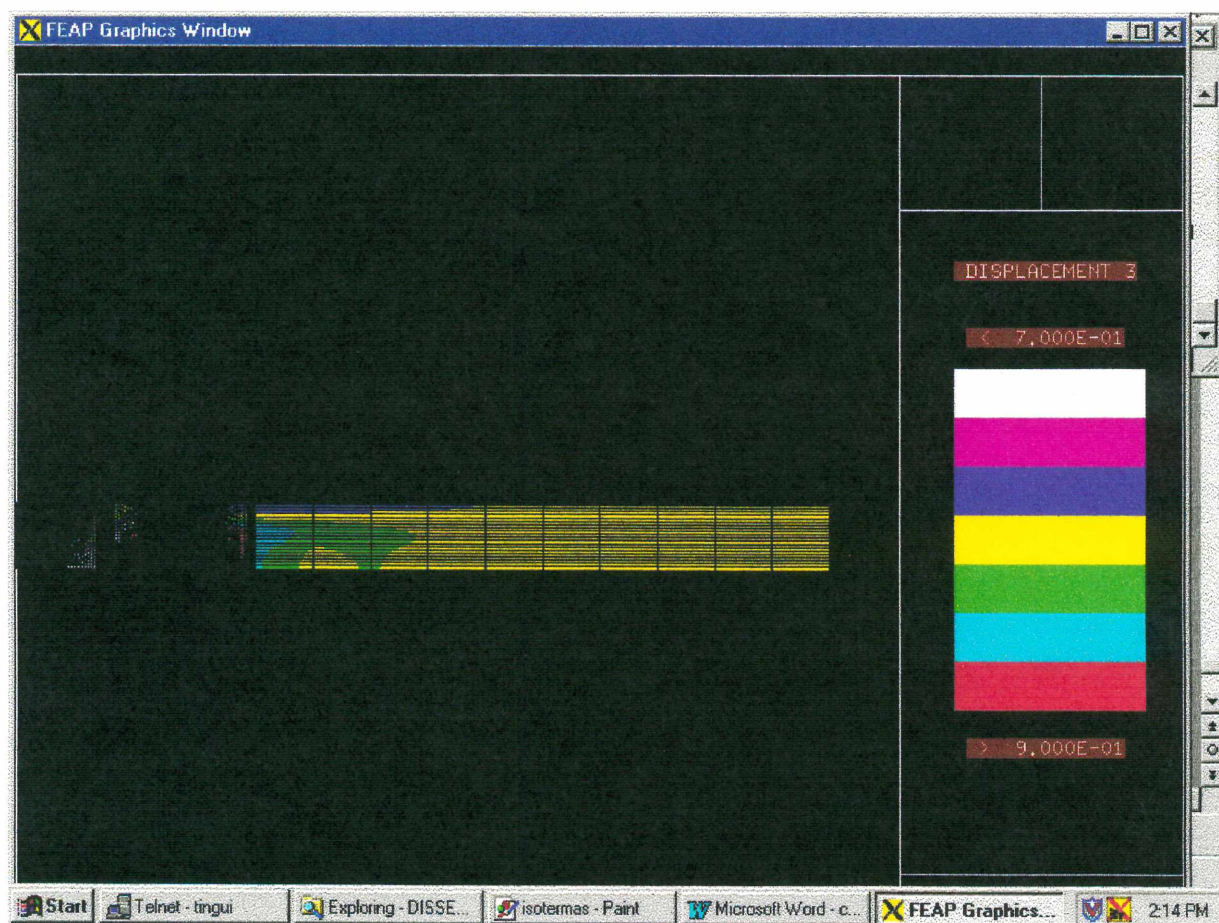


Figura 17: Gráfico das isothermas para o espaçamento ótimo  $S/2b = 0.75$ ,  $e = 0.65$  e  $Re_L = 620$ .

#### 6.4 Distância entre centros variável na direção do escoamento

Alternativamente, o arranjo de tubos pode ser otimizado fixando-se a altura do canal elementar do arranjo  $(S+2b)/2$  e variando-se a distância entre centros na direção do escoamento  $(1 = \sqrt{3} (S+2b)/2)$  da Figura 1. Foi realizado um teste partindo de um arranjo com:  $e = 1$ ,  $\tilde{L} = 1.2$ ,  $L/2b = 7.44$  e  $S/2b = 1.5$ , como dimensões do maior arranjo e  $Re_L = 620$ . O valor de  $\tilde{1}$  para a máxima transferência de calor situa-se em torno de 0.8 para este caso de acordo com a Figura 18.

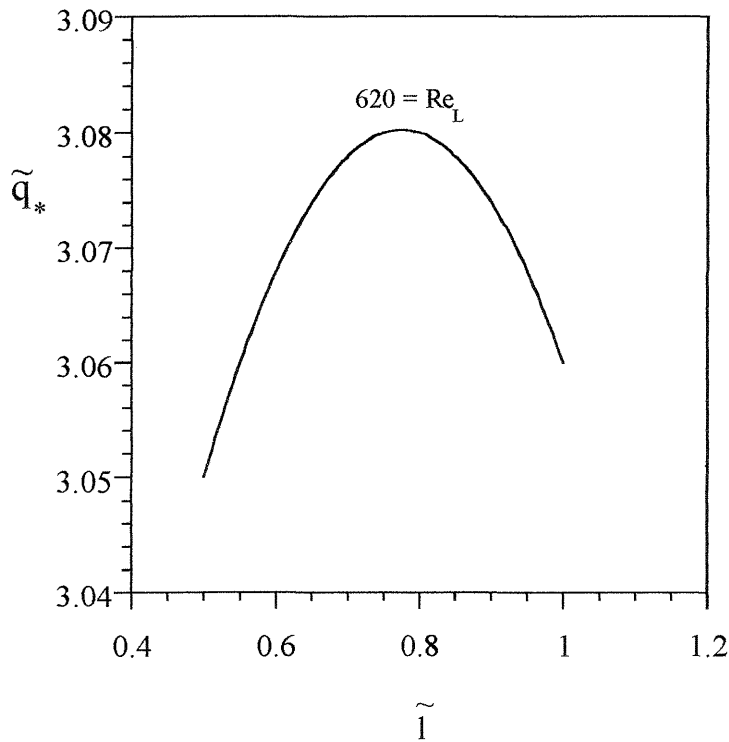


Figura 18: Gráfico da distância entre centros ótima na direção do escoamento.

A distância máxima entre centros na direção do escoamento,  $l_{\max}$  é dada por:

$$l_{\max} = (L-2b) / (N-1) \quad (68)$$

A distância entre centros na direção do escoamento adimensionalizada ( $\tilde{1}$ ) é dada por:

$$\tilde{1} = l / l_{\max} \quad (69)$$

A finalidade do estudo apresentado nesta seção foi de apenas estudar a possibilidade de estender o presente estudo em trabalhos futuros, para um estudo de otimização de arranjos de tubos elípticos com relação a mais um grau de liberdade.

## 7 CONCLUSÕES

Os resultados numéricos obtidos nesta tese demonstram que arranjos de tubos de seção circular e elíptica podem ser otimizados para máxima transferência de calor (ou máxima condutância térmica), quando a otimização é sujeita a uma determinada restrição de volume. Do ponto de vista de aplicação prática direta dos resultados aqui apresentados, cumpre ressaltar que isto dependerá da similaridade entre a configuração analisada neste trabalho, pela Figura 1, e o projeto real em estudo. No entanto, do ponto de vista fundamental os resultados obtidos permitem afirmar que sempre haverá um espaçamento ótimo entre fileiras de tubos de trocadores de calor, que é importante ser encontrado.

Do ponto de vista de transferência de calor, foi demonstrado que a configuração elíptica tem melhor desempenho que a circular. Entre os casos estudados, o ganho máximo obtido foi de 13 %, para  $e = 0.65$ , com  $Re_L = 465$ . O ganho em transferência de calor, combinado com a redução em perda de carga de até 25 % observada em estudos anteriores (Brauer, 1964; Bordalo e Saboya, 1995) mostram que arranjos de tubos elípticos têm potencial para um desempenho consideravelmente melhor do que os convencionais circulares.

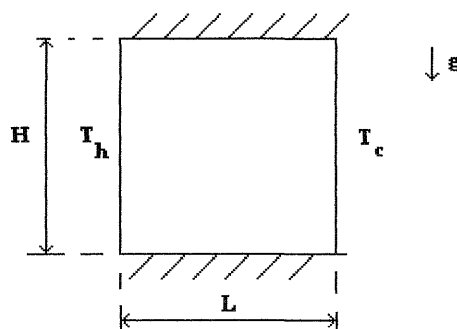
Este estudo ainda demonstrou a possibilidade de otimização dos arranjos com relação à distância entre centros dos tubos, na direção do escoamento. Este grau de liberdade adicional deve ser adicionalmente investigado em estudos posteriores.

Como uma seqüência natural para o presente estudo, a fim de quantificar o ganho total em desempenho do ponto de vista de transferência de calor e perda de carga, sugere-se a utilização de uma nova figura de mérito. O cálculo da entropia gerada no processo, a partir dos campos de temperatura e pressão obtidos neste trabalho combinaria os dois efeitos. A otimização seria, portanto, direcionada para a minimização da geração de entropia no processo.

## ANEXO 1

### Teste do cálculo da transferência de calor em uma cavidade quadrada

Visando a comparação entre a solução numérica obtida com o código FEAP e a solução exata reportada por Bejan (1995), foi efetuado o cálculo da transferência de calor para uma cavidade quadrada em convecção natural, mostrada na Figura 19.



**Figura 19: Convecção natural em uma cavidade quadrada.**

O problema foi analisado partindo das seguintes hipóteses:

- Fluido newtoniano;
- Escoamento incompressível;
- 2-D (bi-dimensional);
- Regime permanente;
- Escoamento laminar;
- Propriedades constantes no fluido, e
- Dissipação viscosa desprezível.

#### 1. Equações governantes

Partindo-se das equações básicas de conservação de massa, momento e energia e de acordo com as hipóteses pré-definidas, para fluxo constante do fluido as equações que governam o problema resultantes são:

### 1.1 Conservação de massa

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \quad (70)$$

### 1.2 Conservação de momentum

- Componente na direção x:

$$u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} = -\frac{1}{\rho} \frac{\partial p}{\partial x} + \nu \left( \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) \quad (71)$$

- Componente na direção y:

$$u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} = -\frac{1}{\rho} \frac{\partial p}{\partial y} + \nu \left( \frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right) - \rho g \quad (72)$$

### 1.3 Conservação de energia

$$u \frac{\partial T}{\partial x} + v \frac{\partial T}{\partial y} = \alpha \left( \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right) \quad (73)$$

## 2. Forma adimensional

Por se tratar de um problema envolvendo convecção natural o número de Rayleigh será utilizado como parâmetro.

Adimensionalizando as variáveis envolvidas nas equações tem-se:

- Velocidades:

$$U = u \frac{H}{\alpha} Ra^{-1/2} \quad (74)$$

$$V = v \frac{H}{\alpha} Ra^{-1/2} \quad (75)$$

- Posições:

$$X = \frac{x}{H} \quad (76)$$

$$Y = \frac{y}{H} \quad (77)$$

- Pressão:

$$P = \frac{p.H^2}{\mu \propto Ra^{1/2}} \quad (78)$$

- Temperatura:

$$\theta = \frac{T - T_c}{T_h - T_c} \quad (79)$$

- Número de Rayleigh:

$$Ra = \frac{g\beta (T_h - T_c) H^3}{\alpha \nu} \quad (80)$$

### 3. Equações governantes adimensionais

Substituindo as equações adimensionais nas equações governantes, as equações governantes adimensionais resultantes são apresentadas a seguir:

#### 3.1 Conservação de massa

$$\frac{\partial U}{\partial X} + \frac{\partial V}{\partial Y} = 0 \quad (81)$$

#### 3.2 Conservação de momentum

- Componente na direção x:

$$\frac{Ra^{1/2}}{Pr} \left[ U \frac{\partial U}{\partial X} + V \frac{\partial U}{\partial Y} \right] = -\frac{\partial P}{\partial X} + \frac{\partial^2 U}{\partial X^2} + \frac{\partial^2 U}{\partial Y^2} \quad (82)$$

- Componente na direção y:

$$\frac{Ra^{1/2}}{Pr} \left[ U \frac{\partial V}{\partial X} + V \frac{\partial V}{\partial Y} \right] = -\frac{\partial P}{\partial Y} + \frac{\partial^2 V}{\partial X^2} + \frac{\partial^2 V}{\partial Y^2} + Ra^{1/2} \theta \quad (83)$$

### 3.3 Conservação de energia

$$Ra^{1/2} \left[ U \frac{\partial \theta}{\partial X} + V \frac{\partial \theta}{\partial Y} \right] = \frac{\partial^2 \theta}{\partial X^2} + \frac{\partial^2 \theta}{\partial Y^2} \quad (84)$$

## 4. Solução experimental/integral

O número de Nusselt para o presente problema é definido como (Bejan, 1995):

$$Nu = \frac{Q}{Q_{\text{cond pura}}} \quad (85)$$

onde a transferência de calor através da cavidade é dada por:

$$Q = k \int_0^H \left( \frac{\partial T}{\partial x} \right)_{x=0} dy \quad (86)$$

Resolvendo as equações (70) – (73), sujeitas às condições de contorno da Figura 20 por um método integral (Bejan, 1995) e usando a equação (86) obtém-se:

$$Q = 0.364 (T_h - T_c) Ra_H^{1/4} \quad (87)$$

O fluxo de calor por condução pura é dado por:

$$Q_{\text{cond pura}} = kH \frac{(T_h - T_c)}{L} \quad (88)$$

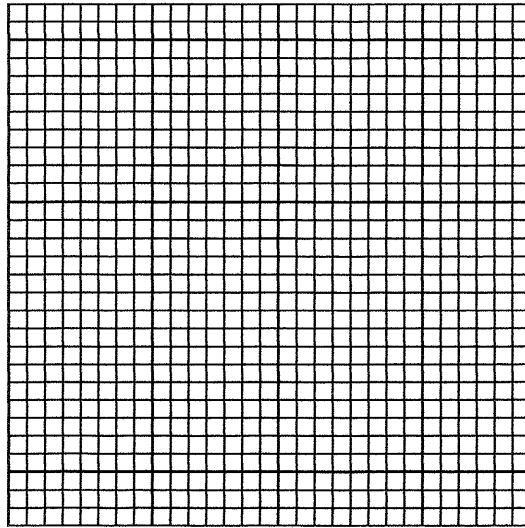
Substituindo as equações (87) e (88) na equação (85), obtém-se:

$$Q = 0.364 \frac{L}{H} Ra^{1/4} \quad (89)$$

Para  $L/H = 1$ ;  $Ra^{1/4} = 10^3$ ; obtém - se  $\overline{Nu} = 2.0469$

## 5 Solução numérica

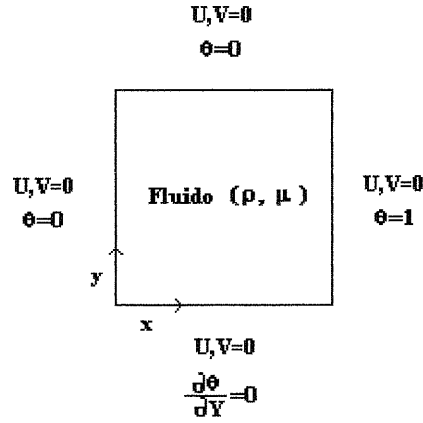
A malha utilizada para a resolução do problema é composta de 841 elementos, 4 nós por elemento, totalizando 900 nós, sendo 30 nós na direção x e 30 nós na direção y, de acordo com a Figura 20.



**Figura 20: Malha gerada pelo código FEAP para a cavidade quadrada.**

As condições de contorno adimensionais, encontram-se de acordo com a Figura 21.





**Figura 21: Condições de contorno na cavidade quadrada.**

As equações implementadas no código FEAP, através do elemento Navier-Stokes e Energia encontram-se na forma:

$$\bar{\rho} \left( \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} \right) = -\nabla p + \bar{\mu} \nabla^2 \mathbf{u} + \bar{\rho} \bar{g} \bar{\beta} (T - T_\infty) \quad (90)$$

$$\bar{\rho} \bar{c}_p \mathbf{u} \cdot \nabla T = \bar{k} \nabla^2 T \quad (91)$$

Visando a correspondência entre as equações governantes adimensionais e as equações implementadas no código FEAP, torna-se necessário a correlação que se segue para o arquivo de entrada de dados:

Exemplificando para,  $Ra = 10^4$ ,  $Pr = 0.71$  tem-se:

$$\begin{aligned} \bar{\mu} &= 1.0; \gamma = 1.e6; \bar{\rho} = 140.84507; l' = 2; k' = 2 \\ \alpha_p &= 0; \bar{g}_x = 0; \bar{g}_y = -1.0; \bar{\beta} = 0.71; \bar{c}_p = 0.71; \bar{k} = 1.0 \end{aligned}$$

onde:

$$\bar{\rho} = Ra^{1/2} \cdot Pr^{-1}; \quad \bar{c}_p = Pr = \frac{\nu}{\alpha} \quad (92)$$

$$\overline{Nu} = \int_0^1 \left( \frac{\partial \theta}{\partial X} \right)_{X=0} dY \quad (93)$$

Através da implementação da equação (93) no código FEAP, para o cálculo do fluxo de calor o resultado obtido para o número de Nusselt foi de 2.165 valor este próximo do apresentado na Figura 5.13, do livro (Bejan, 1995) para  $H/L=1$  e  $Ra=10^4$ .

A Figura 22 apresenta as isothermas geradas para as condições de contorno anteriormente estabelecidas.

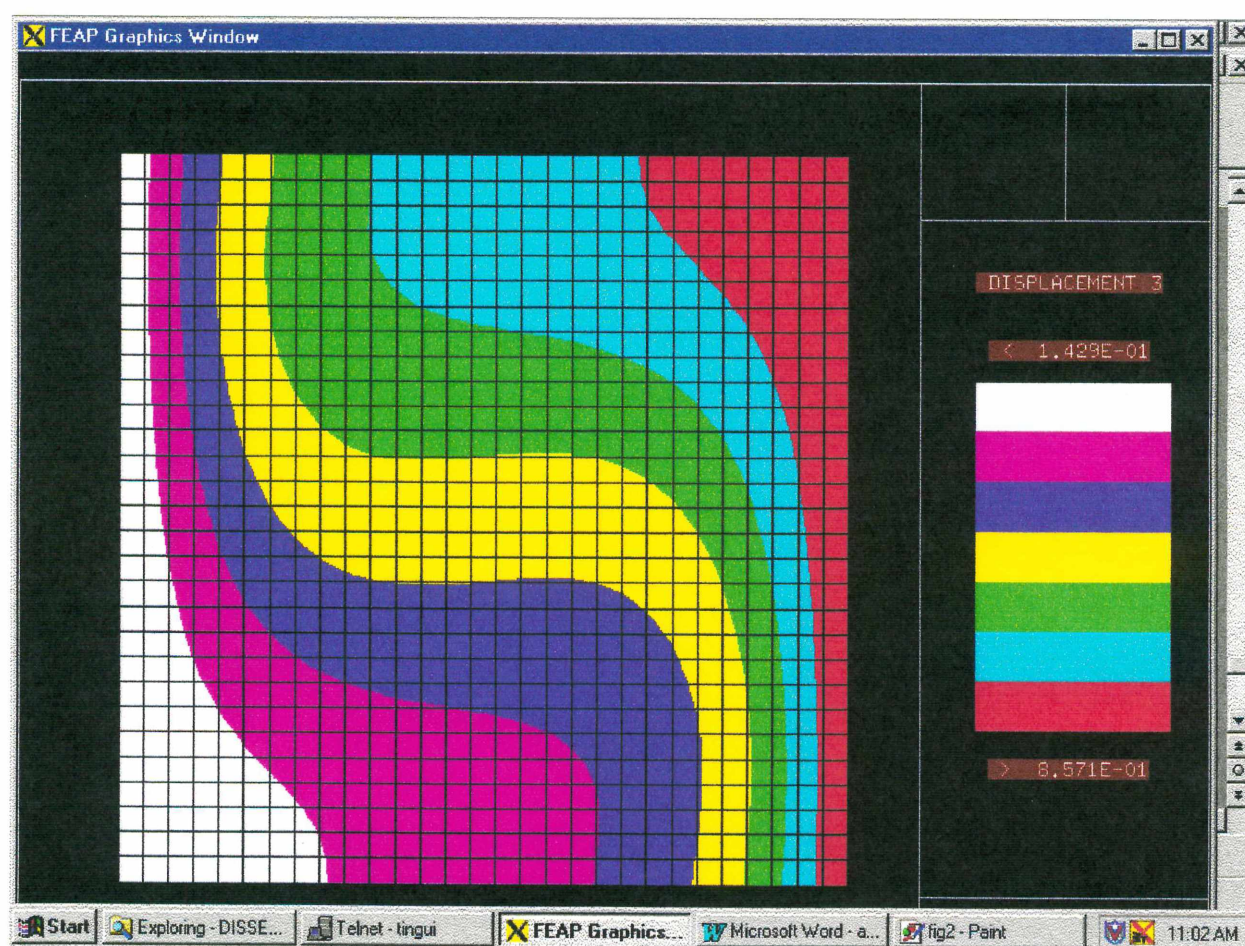


Figura 22: Gráfico das isothermas para uma cavidade quadrada.

**ANEXO 2****Input-file para uma cavidade quadrada**

feap \*\* NC in a square cavity - Ra=1.e4, Pr=0.71

900,841,1,2,3,4

bloc

4,29,29,1,1,1

1,0.,0.

2,0.,1.

3,1.,1.

4,1.,0.

boun

2,1,-1,-1,-1

29,0,1,1,1

1,30,-1,-1,0

871,0,1,1,1

872,1,-1,-1,-1

899,0,1,1,1

30,30,-1,-1,1

900,0,1,1,1

forc

871,1,0.,0.,1.

900,0,0.,0.,1.

mate

1,2

1.,1.e6,140.84507,2,2

0.,0.,-1.,0.71,0.71,1.

0.,0.

end

inter

stop

end

## ANEXO 3

**Input-file para 4 tubos no canal elementar,  $e = 1.0$ ,  $S/D = 1.0$  e  $Re_L = 620$** 

feap \*\*simulação numérica de tubos cilíndricos\*\*

5460,5180,1,2,3,4

bloc 1

4,9,20,1,1,1,250

1,0.,0.0000

2,0.9404,0.0000

3,0.9404,0.1613

4,0.,0.1613

bloc 2

4,10,20,10,181,1,249

1,0.9404,0.0000

2,0.9994,0.0000

3,0.9994,0.1613

4,0.9404,0.1613

bloc 3

4,5,20,20,381,1,254

1,0.9994,0.0000

2,1.0196,0.0000

3,1.0196,0.1080

4,0.9994,0.1613

bloc 4

4,5,20,25,481,1,254

1,1.0196,0.0000

2,1.0398,0.0000

3,1.0398,0.0915

4,1.0196,0.1080

bloc 5

4,5,20,30,581,1,254

1,1.0398,0.0000

2,1.0599,0.0000

3,1.0599,0.0832

4,1.0398,0.0915

bloc 6

4,5,20,35,681,1,254

1,1.0599,0.0000

2,1.0801,0.0000

3,1.0801,0.0807

4,1.0599,0.0832

bloc 7

4,5,20,40,781,1,254

1,1.0801,0.0000  
2,1.1003,0.0000  
3,1.1003,0.0832  
4,1.0801,0.0807

bloc 8

4,5,20,45,881,1,254  
1,1.1003,0.0000  
2,1.1204,0.0000  
3,1.1204,0.0915  
4,1.1003,0.0832

bloc 9

4,5,20,50,981,1,254  
1,1.1204,0.0000  
2,1.1406,0.0000  
3,1.1406,0.1080  
4,1.1204,0.0915

bloc 10

4,5,20,55,1081,1,254  
1,1.1406,0.0000  
2,1.1607,0.0000  
3,1.1607,0.1613  
4,1.1406,0.1080

bloc 11

4,20,20,60,1181,1,239  
1,1.1607,0.0000  
2,1.2788,0.0000  
3,1.2788,0.1613  
4,1.1607,0.1613

bloc 12

4,5,20,80,1581,1,254  
1,1.2788,0.0000  
2,1.2990,0.0533  
3,1.2990,0.1613  
4,1.2788,0.1613

bloc 13

4,5,20,85,1681,1,254  
1,1.2990,0.0533  
2,1.3191,0.0698  
3,1.3191,0.1613  
4,1.2990,0.1613

bloc 14

4,5,20,90,1781,1,254  
1,1.3191,0.0698  
2,1.3393,0.0781

3,1.3393,0.1613

4,1.3191,0.1613

bloc 15

4,5,20,95,1881,1,254

1,1.3393,0.0781

2,1.3595,0.0807

3,1.3595,0.1613

4,1.3393,0.1613

bloc 16

4,5,20,100,1981,1,254

1,1.3595,0.0807

2,1.3796,0.0781

3,1.3796,0.1613

4,1.3595,0.1613

bloc 17

4,5,20,105,2081,1,254

1,1.3796,0.0781

2,1.3998,0.0698

3,1.3998,0.1613

4,1.3796,0.1613

bloc 18

4,5,20,110,2181,1,254

1,1.3998,0.0698

2,1.4200,0.0533

3,1.4200,0.1613

4,1.3998,0.1613

bloc 19

4,5,20,115,2281,1,254

1,1.4200,0.0533

2,1.4401,0.0000

3,1.4401,0.1613

4,1.4200,0.1613

bloc 20

4,20,20,120,2381,1,239

1,1.4401,0.0000

2,1.5582,0.0000

3,1.5582,0.1613

4,1.4401,0.1613

bloc 21

4,5,20,140,2781,1,254

1,1.5582,0.0000

2,1.5784,0.0000

3,1.5784,0.1080

4,1.5582,0.1613

bloc 22

4,5,20,145,2881,1,254  
1,1.5784,0.0000  
2,1.5985,0.0000  
3,1.5985,0.0915  
4,1.5784,0.1080

bloc 23

4,5,20,150,2981,1,254  
1,1.5985,0.0000  
2,1.6187,0.0000  
3,1.6187,0.0832  
4,1.5985,0.0915

bloc 24

4,5,20,155,3081,1,254  
1,1.6187,0.0000  
2,1.6388,0.0000  
3,1.6388,0.0807  
4,1.6187,0.0832

bloc 25

4,5,20,160,3181,1,254  
1,1.6388,0.0000  
2,1.6590,0.0000  
3,1.6590,0.0832  
4,1.6388,0.0807

bloc 26

4,5,20,165,3281,1,254  
1,1.6590,0.0000  
2,1.6792,0.0000  
3,1.6792,0.0915  
4,1.6590,0.0832

bloc 27

4,5,20,170,3381,1,254  
1,1.6792,0.0000  
2,1.6993,0.0000  
3,1.6993,0.1080  
4,1.6792,0.0915

bloc 28

4,5,20,175,3481,1,254  
1,1.6993,0.0000  
2,1.7195,0.0000  
3,1.7195,0.1613  
4,1.6993,0.1080

bloc 29

4,20,20,180,3581,1,239  
1,1.7195,0.0000  
2,1.8376,0.0000  
3,1.8376,0.1613  
4,1.7195,0.1613

bloc 30

4,5,20,200,3981,1,254  
1,1.8376,0.0000  
2,1.8577,0.0533  
3,1.8577,0.1613  
4,1.8376,0.1613

bloc 31

4,5,20,205,4081,1,254  
1,1.8577,0.0533  
2,1.8779,0.0698  
3,1.8779,0.1613  
4,1.8577,0.1613

bloc 32

4,5,20,210,4181,1,254  
1,1.8779,0.0698  
2,1.8981,0.0781  
3,1.8981,0.1613  
4,1.8779,0.1613

bloc 33

4,5,20,215,4281,1,254  
1,1.8981,0.0781  
2,1.9182,0.0807  
3,1.9182,0.1613  
4,1.8981,0.1613

bloc 34

4,5,20,220,4381,1,254  
1,1.9182,0.0807  
2,1.9384,0.0781  
3,1.9384,0.1613  
4,1.9182,0.1613

bloc 35

4,5,20,225,4481,1,254  
1,1.9384,0.0781  
2,1.9586,0.0698  
3,1.9586,0.1613  
4,1.9384,0.1613

bloc 36

4,5,20,230,4581,1,254  
1,1.9586,0.0698



2,1.9787,0.0533  
 3,1.9787,0.1613  
 4,1.9586,0.1613

bloc 37

4,5,20,235,4681,1,254  
 1,1.9787,0.0533  
 2,1.9989,0.0000  
 3,1.9989,0.1613  
 4,1.9787,0.1613

bloc 38

4,10,20,240,4781,1,249  
 1,1.9989,0.0000  
 2,2.0579,0.0000  
 3,2.0579,0.1613  
 4,1.9989,0.1613

bloc 39

4,10,20,250,4981,1,249  
 1,2.0579,0.0000  
 2,2.9983,0.0000  
 3,2.9983,0.1613  
 4,2.0579,0.1613

boun

1,260,-1,-1,-1  
 5201,0,1,1,1  
 5202,1,0,-1,0  
 5219,0,0,1,0  
 5220,1,-1,-1,-1  
 5260,0,1,1,1  
 5261,1,0,-1,0  
 5339,0,0,1,0  
 5340,1,-1,-1,-1  
 5380,0,1,1,1  
 5381,1,0,-1,0  
 5460,0,0,1,0  
 2,1,0,-1,0  
 79,0,0,1,0  
 80,1,-1,-1,-1  
 120,0,1,1,1  
 121,1,0,-1,0  
 199,0,0,1,0  
 200,1,-1,-1,-1  
 240,0,1,1,1  
 241,1,0,-1,0  
 260,0,0,1,0

forc

1,260,1,0,0.

5201,0,1,0,0.  
 5220,1,0,0,1.  
 5260,0,0,0,1.  
 80,1,0,0,1.  
 120,0,0,0,1.  
 5340,1,0,0,1.  
 5380,0,0,0,1.  
 200,1,0,0,1.  
 240,0,0,0,1.

mate

1,2

0.001613,1.e6,1.0,2,2  
 0,0,0,0,1,0.002240  
 0,0,0,0.08065,0.08065  
 476.,480.,576.,580.,676.,680.  
 776.,780.,876.,880.,976.,980.  
 1076.,1080.,1176.,1180.,1.0801,0.1613  
 1581.,1585.,1681.,1685.,1781.,1785.  
 1881.,1885.,1981.,1985.,2081.,2085.  
 2181.,2185.,2281.,2285.,1.3595,0.  
 2876.,2880.,2976.,2980.,3076.,3080.  
 3176.,3180.,3276.,3280.,3376.,3380.  
 3476.,3480.,3576.,3580.,1.6388,0.1613  
 3981.,3985.,4081.,4085.,4181.,4185.  
 4281.,4285.,4381.,4385.,4481.,4485.  
 4581.,4585.,4681.,4685.,1.9182,0.

end

inter

stop

## ANEXO 4

**Input-file para 4 tubos no canal elementar,  $e = 0.80$ ,  $S/2b = 1.0$  e  $Re_L = 620$**

feap \*\*simulação numérica de tubos elípticos\*\*

5460,5180,1,2,3,4

bloc 1

4,9,20,1,1,1,250

1,0.0000,0.0000

2,1.0013,0.0000

3,1.0013,0.1613

4,0.0000,0.1613

bloc 2

4,10,20,10,181,1,249

1,1.0013,0.0000

2,1.0403,0.0000

3,1.0403,0.1613

4,1.0013,0.1613

bloc 3

4,5,20,20,381,1,254

1,1.0403,0.0000

2,1.0655,0.0000

3,1.0655,0.1080

4,1.0403,0.1613

bloc 4

4,5,20,25,481,1,254

1,1.0655,0.0000

2,1.0907,0.0000

3,1.0907,0.0915

4,1.0655,0.1080

bloc 5

4,5,20,30,581,1,254

1,1.0907,0.0000

2,1.1159,0.0000

3,1.1159,0.0832

4,1.0907,0.0915

bloc 6

4,5,20,35,681,1,254

1,1.1159,0.0000

2,1.1411,0.0000

3,1.1411,0.0807

4,1.1159,0.0832

bloc 7

4,5,20,40,781,1,254

1,1.1411,0.0000  
2,1.1663,0.0000  
3,1.1663,0.0832  
4,1.1411,0.0807

bloc 8

4,5,20,45,881,1,254  
1,1.1663,0.0000  
2,1.1915,0.0000  
3,1.1915,0.0915  
4,1.1663,0.0832

bloc 9

4,5,20,50,981,1,254  
1,1.1915,0.0000  
2,1.2167,0.0000  
3,1.2167,0.1080  
4,1.1915,0.0915

bloc 10

4,5,20,55,1081,1,254  
1,1.2167,0.0000  
2,1.2419,0.0000  
3,1.2419,0.1613  
4,1.2167,0.1080

bloc 11

4,20,20,60,1181,1,239  
1,1.2419,0.0000  
2,1.3199,0.0000  
3,1.3199,0.1613  
4,1.2419,0.1613

bloc 12

4,5,20,80,1581,1,254  
1,1.3199,0.0000  
2,1.3451,0.0533  
3,1.3451,0.1613  
4,1.3199,0.1613

bloc 13

4,5,20,85,1681,1,254  
1,1.3451,0.0533  
2,1.3703,0.0698  
3,1.3703,0.1613  
4,1.3451,0.1613

bloc 14

4,5,20,90,1781,1,254  
1,1.3703,0.0698  
2,1.3955,0.0781

3,1.3955,0.1613  
4,1.3703,0.1613

bloc 15

4,5,20,95,1881,1,254  
1,1.3955,0.0781  
2,1.4207,0.0807  
3,1.4207,0.1613  
4,1.3955,0.1613

bloc 16

4,5,20,100,1981,1,254  
1,1.4207,0.0807  
2,1.4459,0.0781  
3,1.4459,0.1613  
4,1.4207,0.1613

bloc 17

4,5,20,105,2081,1,254  
1,1.4459,0.0781  
2,1.4711,0.0698  
3,1.4711,0.1613  
4,1.4459,0.1613

bloc 18

4,5,20,110,2181,1,254  
1,1.4711,0.0698  
2,1.4963,0.0533  
3,1.4963,0.1613  
4,1.4711,0.1613

bloc 19

4,5,20,115,2281,1,254  
1,1.4963,0.0533  
2,1.5215,0.0000  
3,1.5215,0.1613  
4,1.4963,0.1613

bloc 20

4,20,20,120,2381,1,239  
1,1.5215,0.0000  
2,1.5994,0.0000  
3,1.5994,0.1613  
4,1.5215,0.1613

bloc 21

4,5,20,140,2781,1,254  
1,1.5994,0.0000  
2,1.6246,0.0000  
3,1.6246,0.1080  
4,1.5994,0.1613

bloc 22

4,5,20,145,2881,1,254

1,1.6246,0.0000

2,1.6498,0.0000

3,1.6498,0.0915

4,1.6246,0.1080

bloc 23

4,5,20,150,2981,1,254

1,1.6498,0.0000

2,1.6750,0.0000

3,1.6750,0.0832

4,1.6498,0.0915

bloc 24

4,5,20,155,3081,1,254

1,1.6750,0.0000

2,1.7002,0.0000

3,1.7002,0.0807

4,1.6750,0.0832

bloc 25

4,5,20,160,3181,1,254

1,1.7002,0.0000

2,1.7254,0.0000

3,1.7254,0.0832

4,1.7002,0.0807

bloc 26

4,5,20,165,3281,1,254

1,1.7254,0.0000

2,1.7506,0.0000

3,1.7506,0.0915

4,1.7254,0.0832

bloc 27

4,5,20,170,3381,1,254

1,1.7506,0.0000

2,1.7758,0.0000

3,1.7758,0.1080

4,1.7506,0.0915

bloc 28

4,5,20,175,3481,1,254

1,1.7758,0.0000

2,1.8010,0.0000

3,1.8010,0.1613

4,1.7758,0.1080

bloc 29

4,20,20,180,3581,1,239  
1,1.8010,0.0000  
2,1.8790,0.0000  
3,1.8790,0.1613  
4,1.8010,0.1613

bloc 30

4,5,20,200,3981,1,254  
1,1.8790,0.0000  
2,1.9042,0.0533  
3,1.9042,0.1613  
4,1.8790,0.1613

bloc 31

4,5,20,205,4081,1,254  
1,1.9042,0.0533  
2,1.9294,0.0698  
3,1.9294,0.1613  
4,1.9042,0.1613

bloc 32

4,5,20,210,4181,1,254  
1,1.9294,0.0698  
2,1.9546,0.0781  
3,1.9546,0.1613  
4,1.9294,0.1613

bloc 33

4,5,20,215,4281,1,254  
1,1.9546,0.0781  
2,1.9798,0.0807  
3,1.9798,0.1613  
4,1.9546,0.1613

bloc 34

4,5,20,220,4381,1,254  
1,1.9798,0.0807  
2,2.0050,0.0781  
3,2.0050,0.1613  
4,1.9798,0.1613

bloc 35

4,5,20,225,4481,1,254  
1,2.0050,0.0781  
2,2.0302,0.0698  
3,2.0302,0.1613  
4,2.0050,0.1613

bloc 36

4,5,20,230,4581,1,254  
1,2.0302,0.0698

2,2.0554,0.0533  
 3,2.0554,0.1613  
 4,2.0302,0.1613

#### bloc 37

4,5,20,235,4681,1,254  
 1,2.0554,0.0533  
 2,2.0806,0.0000  
 3,2.0806,0.1613  
 4,2.0554,0.1613

#### bloc 38

4,10,20,240,4781,1,249  
 1,2.0806,0.0000  
 2,2.1196,0.0000  
 3,2.1196,0.1613  
 4,2.0806,0.1613

#### bloc 39

4,10,20,250,4981,1,249  
 1,2.1196,0.0000  
 2,3.1209,0.0000  
 3,3.1209,0.1613  
 4,2.1196,0.1613

#### boun

1,260,-1,-1,-1  
 5201,0,1,1,1  
 5202,1,0,-1,0  
 5219,0,0,1,0  
 5220,1,-1,-1,-1  
 5260,0,1,1,1  
 5261,1,0,-1,0  
 5339,0,0,1,0  
 5340,1,-1,-1,-1  
 5380,0,1,1,1  
 5381,1,0,-1,0  
 5460,0,0,1,0  
 2,1,0,-1,0  
 79,0,0,1,0  
 80,1,-1,-1,-1  
 120,0,1,1,1  
 121,1,0,-1,0  
 199,0,0,1,0  
 200,1,-1,-1,-1  
 240,0,1,1,1  
 241,1,0,-1,0  
 260,0,0,1,0

#### forc

1,260,1.,0.,0.



5201,0,1,0,0.  
 5220,1,0,0,1.  
 5260,0,0,0,1.  
 80,1,0,0,1.  
 120,0,0,0,1.  
 5340,1,0,0,1.  
 5380,0,0,0,1.  
 200,1,0,0,1.  
 240,0,0,0,1.

mate

1,2  
 0.00161,1.e6,1.0,2,2  
 0.,0.,0.,0.,1.,0.00224  
 0.,0.,0.,0.1008,0.08065  
 476.,480.,576.,580.,676.,680.  
 776.,780.,876.,880.,976.,980.  
 1076.,1080.,1176.,1180.,1.1411,0.1613  
 1581.,1585.,1681.,1685.,1781.,1785.  
 1881.,1885.,1981.,1985.,2081.,2085.  
 2181.,2185.,2281.,2285.,1.4207,0.  
 2876.,2880.,2976.,2980.,3076.,3080.  
 3176.,3180.,3276.,3280.,3376.,3380.  
 3476.,3480.,3576.,3580.,1.7002,0.1613  
 3981.,3985.,4081.,4085.,4181.,4185.  
 4281.,4285.,4381.,4385.,4481.,4485.  
 4581.,4585.,4681.,4685.,1.9798,0.

end

inter  
 stop

## ANEXO 5

**Input-file para 4 tubos no canal elementar,  $e = 0.65$ ,  $S/2b = 1.0$  e  $Re_L = 620$**

feap \*\*simulação numérica de tubos elípticos\*\*  
5460,5180,1,2,3,4

bloc 1

4,9,20,1,1,1,250

1,0.0000,0.0000

2,1.0711,0.0000

3,1.0711,0.1613

4,0.0000,0.1613

bloc 2

4,10,20,10,181,1,249

1,1.0711,0.0000

2,1.0869,0.0000

3,1.0869,0.1613

4,1.0711,0.1613

bloc 3

4,5,20,20,381,1,254

1,1.0869,0.0000

2,1.1179,0.0000

3,1.1179,0.1080

4,1.0869,0.1613

bloc 4

4,5,20,25,481,1,254

1,1.1179,0.0000

2,1.1489,0.0000

3,1.1489,0.0915

4,1.1179,0.1080

bloc 5

4,5,20,30,581,1,254

1,1.1489,0.0000

2,1.1799,0.0000

3,1.1799,0.0832

4,1.1489,0.0915

bloc 6

4,5,20,35,681,1,254

1,1.1799,0.0000

2,1.2109,0.0000

3,1.2109,0.0807

4,1.1799,0.0832

bloc 7

4,5,20,40,781,1,254

1,1.2109,0.0000  
2,1.2420,0.0000  
3,1.2420,0.0832  
4,1.2109,0.0807

bloc 8

4,5,20,45,881,1,254  
1,1.2420,0.0000  
2,1.2730,0.0000  
3,1.2730,0.0915  
4,1.2420,0.0832

bloc 9

4,5,20,50,981,1,254  
1,1.2730,0.0000  
2,1.3040,0.0000  
3,1.3040,0.1080  
4,1.2730,0.0915

bloc 10

4,5,20,55,1081,1,254  
1,1.3040,0.0000  
2,1.3350,0.0000  
3,1.3350,0.1613  
4,1.3040,0.1080

bloc 11

4,20,20,60,1181,1,239  
1,1.3350,0.0000  
2,1.3664,0.0000  
3,1.3664,0.1613  
4,1.3350,0.1613

bloc 12

4,5,20,80,1581,1,254  
1,1.3664,0.0000  
2,1.3974,0.0533  
3,1.3974,0.1613  
4,1.3664,0.1613

bloc 13

4,5,20,85,1681,1,254  
1,1.3974,0.0533  
2,1.4285,0.0698  
3,1.4285,0.1613  
4,1.3974,0.1613

bloc 14

4,5,20,90,1781,1,254  
1,1.4285,0.0698  
2,1.4595,0.0781

3,1.4595,0.1613

4,1.4285,0.1613

bloc 15

4,5,20,95,1881,1,254

1,1.4595,0.0781

2,1.4905,0.0807

3,1.4905,0.1613

4,1.4595,0.1613

bloc 16

4,5,20,100,1981,1,254

1,1.4905,0.0807

2,1.5215,0.0781

3,1.5215,0.1613

4,1.4905,0.1613

bloc 17

4,5,20,105,2081,1,254

1,1.5215,0.0781

2,1.5525,0.0698

3,1.5525,0.1613

4,1.5215,0.1613

bloc 18

4,5,20,110,2181,1,254

1,1.5525,0.0698

2,1.5836,0.0533

3,1.5836,0.1613

4,1.5525,0.1613

bloc 19

4,5,20,115,2281,1,254

1,1.5836,0.0533

2,1.6146,0.0000

3,1.6146,0.1613

4,1.5836,0.1613

bloc 20

4,20,20,120,2381,1,239

1,1.6146,0.0000

2,1.6460,0.0000

3,1.6460,0.1613

4,1.6146,0.1613

bloc 21

4,5,20,140,2781,1,254

1,1.6460,0.0000

2,1.6770,0.0000

3,1.6770,0.1080

4,1.6460,0.1613

bloc 22

4,5,20,145,2881,1,254

1,1.6770,0.0000

2,1.7080,0.0000

3,1.7080,0.0915

4,1.6770,0.1080

bloc 23

4,5,20,150,2981,1,254

1,1.7080,0.0000

2,1.7390,0.0000

3,1.7390,0.0832

4,1.7080,0.0915

bloc 24

4,5,20,155,3081,1,254

1,1.7390,0.0000

2,1.7701,0.0000

3,1.7701,0.0807

4,1.7390,0.0832

bloc 25

4,5,20,160,3181,1,254

1,1.7701,0.0000

2,1.8011,0.0000

3,1.8011,0.0832

4,1.7701,0.0807

bloc 26

4,5,20,165,3281,1,254

1,1.8011,0.0000

2,1.8321,0.0000

3,1.8321,0.0915

4,1.8011,0.0832

bloc 27

4,5,20,170,3381,1,254

1,1.8321,0.0000

2,1.8631,0.0000

3,1.8631,0.1080

4,1.8321,0.0915

bloc 28

4,5,20,175,3481,1,254

1,1.8631,0.0000

2,1.8941,0.0000

3,1.8941,0.1613

4,1.8631,0.1080

bloc 29

4,20,20,180,3581,1,239  
1,1.8941,0.0000  
2,1.9256,0.0000  
3,1.9256,0.1613  
4,1.8941,0.1613

bloc 30  
4,5,20,200,3981,1,254  
1,1.9256,0.0000  
2,1.9566,0.0533  
3,1.9566,0.1613  
4,1.9256,0.1613

bloc 31  
4,5,20,205,4081,1,254  
1,1.9566,0.0533  
2,1.9876,0.0698  
3,1.9876,0.1613  
4,1.9566,0.1613

bloc 32  
4,5,20,210,4181,1,254  
1,1.9876,0.0698  
2,2.0186,0.0781  
3,2.0186,0.1613  
4,1.9876,0.1613

bloc 33  
4,5,20,215,4281,1,254  
1,2.0186,0.0781  
2,2.0496,0.0807  
3,2.0496,0.1613  
4,2.0186,0.1613

bloc 34  
4,5,20,220,4381,1,254  
1,2.0496,0.0807  
2,2.0807,0.0781  
3,2.0807,0.1613  
4,2.0496,0.1613

bloc 35  
4,5,20,225,4481,1,254  
1,2.0807,0.0781  
2,2.1117,0.0698  
3,2.1117,0.1613  
4,2.0807,0.1613

bloc 36  
4,5,20,230,4581,1,254  
1,2.1117,0.0698

2,2.1427,0.0533  
 3,2.1427,0.1613  
 4,2.1117,0.1613

#### bloc 37

4,5,20,235,4681,1,254  
 1,2.1427,0.0533  
 2,2.1737,0.0000  
 3,2.1737,0.1613  
 4,2.1427,0.1613

#### bloc 38

4,10,20,240,4781,1,249  
 1,2.1737,0.0000  
 2,2.1894,0.0000  
 3,2.1894,0.1613  
 4,2.1737,0.1613

#### bloc 39

4,10,20,250,4981,1,249  
 1,2.1894,0.0000  
 2,3.2606,0.0000  
 3,3.2606,0.1613  
 4,2.1894,0.1613

#### boun

1,260,-1,-1,-1  
 5201,0,1,1,1  
 5202,1,0,-1,0  
 5219,0,0,1,0  
 5220,1,-1,-1,-1  
 5260,0,1,1,1  
 5261,1,0,-1,0  
 5339,0,0,1,0  
 5340,1,-1,-1,-1  
 5380,0,1,1,1  
 5381,1,0,-1,0  
 5460,0,0,1,0  
 2,1,0,-1,0  
 79,0,0,1,0  
 80,1,-1,-1,-1  
 120,0,1,1,1  
 121,1,0,-1,0  
 199,0,0,1,0  
 200,1,-1,-1,-1  
 240,0,1,1,1  
 241,1,0,-1,0  
 260,0,0,1,0

#### forc

1,260,1,0,0.

5201,0,1,0,0.  
 5220,1,0,0,1.  
 5260,0,0,0,1.  
 80,1,0,0,1.  
 120,0,0,0,1.  
 5340,1,0,0,1.  
 5380,0,0,0,1.  
 200,1,0,0,1.  
 240,0,0,0,1.

mate

1,2

0.00161,1.e6,1.0,2,2  
 0.,0.,0.,0.,1.,0.00224  
 0.,0.,0.,0.1241,0.08065  
 476.,480.,576.,580.,676.,680.  
 776.,780.,876.,880.,976.,980.  
 1076.,1080.,1176.,1180.,1.2109,0.1613  
 1581.,1585.,1681.,1685.,1781.,1785.  
 1881.,1885.,1981.,1985.,2081.,2085.  
 2181.,2185.,2281.,2285.,1.4905,0.  
 2876.,2880.,2976.,2980.,3076.,3080.  
 3176.,3180.,3276.,3280.,3376.,3380.  
 3476.,3480.,3576.,3580.,1.7701,0.1613  
 3981.,3985.,4081.,4085.,4181.,4185.  
 4281.,4285.,4381.,4385.,4481.,4485.  
 4581.,4585.,4681.,4685.,2.0496,0.

end

inter

stop



## ANEXO 6

**Elemento Navier-Stokes e Energia para 4 tubos no canal elementar**

subroutine elmt02(d,ul,xl,ix,tl,s,p,ndf,ndm,nst,isw)  
implicit double precision (a-h,o-z)

```

c      #####
c      2-dimensional Non-Linear flow element for FEAP:
c      four node quadrilateral - Navier-Stokes and energy
c      equations
c      Hughes upwinding scheme (Int. J. Num. Meth. in Eng.,
c      Vol. 12, 1359-1365, 1978)
c      by JOSE VARGAS (August, 1996)
c      #####
c
c      Arguments in subroutine call:
c      d  - vector in which element material properties are stored
c      ul - localized nodal velocities
c      xl - localized nodal coordinates
c      ix - element connection array
c      tl - (not used in this element)
c      s  - element stiffness matrix
c      p  - element right-hand side (force) vector
c      ndf - number of dof per node
c      nst - dimensioned size of element stiffness matrix s
c      isw - integer switch=1-8 ; used to request various element
c           functions:
c           isw      function
c           ===      =====
c           1      input element data
c           2      check element for errors
c           3      compute element stiffness and force
c           4      compute fluxes for printed output
c           5      compute element mass matrix (dynamics)
c           6      compute internal force vector only
c           7      compute consistent load vector from
c                   surface tractions
c           8      compute fluxes AT NODES for plotting
c
c....Required format of element material properties in input file:
c
c      Record 1.
c
c      mu    =    dynamic viscosity
c      gama  =    penalty parameter
c      rho   =    Mass density
c      l     =    Quad. pts/dir to be used in elem stiffness
c                   and force vector calculations
c      k     =    Quad. pts/dir to be used in printed stress
c                   output

```

```

c   i   =   not used in this element
c           (enter 0 in all cases)
c
c   Record 2.
c
c   alfa  =   upwinding parameter
c   g_x   =   x-component of gravity
c   g_y   =   y-component of gravity
c   beta  =   coefficient of thermal expansion
c   c     =   specific heat
c   k     =   thermal conductivity
c
c   Record 3.
c
c   T0    =   reference temperature
c   Q     =   Internal heat generation
c   psi0  =   streamfunction value at node #1
c   sa    =   bigger semi-axis of the elliptical tube section
c   sb    =   smaller semi-axis of the elliptical tube section
c
c   Record 4.
c
c   nell1s1bini = first surface - first element
c   nell1s1bend = first surface - last element
c   nell1s2bini = second surface - first element
c   nell1s2bend = second surface - last element
c   nell1s3bini = third surface - first element
c   nell1s3bend = third surface - last element
c
c   Record 5.
c
c   nell1s4bini = fourth surface - first element
c   nell1s4bend = fourth surface - last element
c   nell1s5bini = fifth surface - first element
c   nell1s5bend = fifth surface - last element
c   nell1s6bini = sixth surface - first element
c   nell1s6bend = sixth surface - last element
c
c   Record 6.
c
c   nell1s7bini = seventh surface - first element
c   nell1s7bend = seventh surface - last element
c   nell1s8bini = eighth surface - first element
c   nell1s8bend = eighth surface - last element
c   xol        = first tube - origin of tube x
c   yol        = first tube - origin of tube y
c
c   Record 7.
c
c   nel2s1bini = first surface - first element
c   nel2s1bend = first surface - last element

```

c nel2s2bini = second surface - first element  
 c nel2s2bend = second surface - last element  
 c nel2s3bini = third surface - first element  
 c nel2s3bend = third surface - last element

c

c Record 8.

c

c nel2s4bini = fourth surface - first element  
 c nel2s4bend = fourth surface - last element  
 c nel2s5bini = fifth surface - first element  
 c nel2s5bend = fifth surface - last element  
 c nel2s6bini = sixth surface - first element  
 c nel2s6bend = sixth surface - last element

c

c Record 9.

c

c nel2s7bini = seventh surface - first element  
 c nel2s7bend = seventh surface - last element  
 c nel2s8bini = eighth surface - first element  
 c nel2s8bend = eighth surface - last element  
 c xo2 = second tube - origin of tube x  
 c yo2 = second tube - origin of tube y

c

c Record 10.

c

c nel3s1bini = first surface - first element  
 c nel3s1bend = first surface - last element  
 c nel3s2bini = second surface - first element  
 c nel3s2bend = second surface - last element  
 c nel3s3bini = third surface - first element  
 c nel3s3bend = third surface - last element

c

c Record 11.

c

c nel3s4bini = fourth surface - first element  
 c nel3s4bend = fourth surface - last element  
 c nel3s5bini = fifth surface - first element  
 c nel3s5bend = fifth surface - last element  
 c nel3s6bini = sixth surface - first element  
 c nel3s6bend = sixth surface - last element

c

c Record 12.

c

c nel3s7bini = seventh surface - first element  
 c nel3s7bend = seventh surface - last element  
 c nel3s8bini = eighth surface - first element  
 c nel3s8bend = eighth surface - last element  
 c xo3 = third tube - origin of tube x  
 c yo3 = third tube - origin of tube y

c

c Record 13.

```

c
c   nel4s1bini = first surface - first element
c   nel4s1bend = first surface - last element
c   nel4s2bini = second surface - first element
c   nel4s2bend = second surface - last element
c   nel4s3bini = third surface - first element
c   nel4s3bend = third surface - last element
c
c   Record 14.
c
c   nel4s4bini = fourth surface - first element
c   nel4s4bend = fourth surface - last element
c   nel4s5bini = fifth surface - first element
c   nel4s5bend = fifth surface - last element
c   nel4s6bini = sixth surface - first element
c   nel4s6bend = sixth surface - last element
c
c   Record 15.
c
c   nel4s7bini = seventh surface - first element
c   nel4s7bend = seventh surface - last element
c   nel4s8bini = eighth surface - first element
c   nel4s8bend = eighth surface - last element
c   xo4      = fourth tube - origin of tube x
c   yo4      = fourth tube - origin of tube y
c
c   Other variables used include:
c   eps      = temp. derivs. vector
c   errck    = logical variable used to indicate error in input
c   head     = character variable containing the problem title
c   ipr      = precision of arithmetic: 1-single 2-double
c   ior,iow  = logical units for input and output files
c   lint     = total number of element integration points
c   m        = blank common array used throughout FEAP
c   ma       = material number
c   mct      = line counter for printed stress output
c   numnp    = total number of nodal points
c   numel    = total number of elements
c   numat    = total number of material sets
c   nen      = max number of nodes per element
c   neq      = total number of equations (or DOF) in problem
c   nel      = number of nodes per element
c   nsl      = number of dof per element
c   sig      = fluxes vector
c   shp      = shape function storage array; details in subroutine shape01
c   sg       = vector containing first natural coordinate of
c             Gauss points (i.e., xi)
c   td       = vector used to read from the input file
c   tg       = vector containing second natural coordinate of
c             Gauss points (i.e., eta)
c   wg       = vector of weights associated with Gauss points

```

c xsj = jacobian of global-local coordinate transformation

c

```
character*4 o,head,wd
logical errck,debug
common /bdata/ o,head(20)
common /cdata/ numnp,numel,nummat,nen,neq,ipr
common /debugs/ debug
common /eldata/ dm,n,ma,mct,iel,nel
common /errchk/ errck
common /iofile/ ior,iow
common /plstrs/ np
common m(1)
common /strnum/ istv
common /teste/ tetaia,tetaip,dteta,dtna,dtnp
common /ellipse/ sa,sb
save /bdata/,/cdata/,/debugs/,/eldata/,/plstrs/,/iofile/
```

```
dimension d(1),ul(ndf,1),xl(ndm,1),ix(1),tl(1),s(nst,1),p(1)
1 ,shp(3,9),sg(16),tg(16),wg(16),sig(6),eps(8),wd(2),td(6)
2 ,sd(27,27),taux(2,2),rkbar(2,2),cin(4),uaux(2),bous(2),
3 b(2),din(4),rl(2),shp1(3,9)
data wd /'flow','flow'/
```

c.... go to correct process (based on value of isw)

```
go to(1,2,3,4,5,4,7,8), isw
```

c

```
%%%%%%%%%%
%%%%%%%%%%
```

c isw=1: Read in material properties

c

```
%%%%%%%%%%
%%%%%%%%%%
```

```
1 if(ior.lt.0) write(*,3000)
```

c.... read a line as a string in the input file. Using

c.... commas or blanks as delimiters, dinput stores the

c.... numerical data needed in the vector td

c

c.... read record 1

```
call dinput(td,6)
```

```
if(errck) go to 1
```

c.... move properties

```
d(1) = td(1) ! viscosity
```

```
d(2) = td(2) ! penalty parameter
```

```
d(3) = td(3) ! density
```

```
l = td(4) ! Gauss points / dir for stiffness
```

```
k = td(5) ! Gauss points / dir for stresses
```

```
i = td(6) ! type of element (not used)
```

```
11 if(ior.lt.0) write(*,3001)
```

c

c.... read record 2

```
call dinput(td,6)
```

```
if(errck) go to 11
```

```

c.... move properties into vector d for later use
      d(7) = td(1) ! upwinding parameter (not used here)
      d(8) = td(2) ! g_x
      d(9) = td(3) ! g_y
      d(10) = td(4) ! beta (coefficient of thermal expansion)
      d(11) = td(5) ! c (specific heat)
      d(12) = td(6) ! k (thermal conductivity)
12  if(ior.lt.0) write(*,3002)
c.... read record 3
      call dinput(td,6)
      if(errck) go to 11
c.... move properties into vector d for later use
      d(13) = td(1) ! reference temperature
      d(14) = td(2) ! Internal heat generation
      d(15) = td(3) ! Streamfunction value at node #1
      sa  = td(4) ! bigger semi-axis of the elliptical tube section
      sb  = td(5) ! smaller semi-axis of the elliptical tube section
c.... read record 4
      call dinput(td,6)
      if(errck) go to 11
c.... move properties into variables for later use
      nel1s1bini = td(1) ! first surface - first element in block 1
      nel1s1bend = td(2) ! first surface - last element in block 1
      nel1s2bini = td(3) ! first surface - first element in block 2
      nel1s2bend = td(4) ! first surface - last element in block 2
      nel1s3bini = td(5) ! first surface - first element in block 3
      nel1s3bend = td(6) ! first surface - last element in block 3
c.... read record 5
      call dinput(td,6)
      if(errck) go to 11
c.... move properties into variables for later use
      nel1s4bini = td(1) ! first surface - first element in block 4
      nel1s4bend = td(2) ! first surface - last element in block 4
      nel1s5bini = td(3) ! first surface - first element in block 5
      nel1s5bend = td(4) ! first surface - last element in block 5
      nel1s6bini = td(5) ! first surface - first element in block 6
      nel1s6bend = td(6) ! first surface - last element in block 6
c.... read record 6
      call dinput(td,6)
      if(errck) go to 11
c.... move properties into variables for later use
      nel1s7bini = td(1) ! first surface - first element in block 7
      nel1s7bend = td(2) ! first surface - last element in block 7
      nel1s8bini = td(3) ! first surface - first element in block 8
      nel1s8bend = td(4) ! first surface - last element in block 8
      xo1 = td(5)    ! first surface - origin of tube x
      yo1 = td(6)    ! first surface - origin of tube y
c.... read record 7
      call dinput(td,6)
      if(errck) go to 11
c.... move properties into variables for later use

```

```

    nel2s1bini = td(1) ! first surface - first element in block 1
    nel2s1bend = td(2) ! first surface - last element in block 1
    nel2s2bini = td(3) ! first surface - first element in block 2
    nel2s2bend = td(4) ! first surface - last element in block 2
    nel2s3bini = td(5) ! first surface - first element in block 3
    nel2s3bend = td(6) ! first surface - last element in block 3
c.... read record 8
    call dinput(td,6)
    if(errck) go to 11
c.... move properties into variables for later use
    nel2s4bini = td(1) ! first surface - first element in block 4
    nel2s4bend = td(2) ! first surface - last element in block 4
    nel2s5bini = td(3) ! first surface - first element in block 5
    nel2s5bend = td(4) ! first surface - last element in block 5
    nel2s6bini = td(5) ! first surface - first element in block 6
    nel2s6bend = td(6) ! first surface - last element in block 6
c.... read record 9
    call dinput(td,6)
    if(errck) go to 11
c.... move properties into variables for later use
    nel2s7bini = td(1) ! first surface - first element in block 7
    nel2s7bend = td(2) ! first surface - last element in block 7
    nel2s8bini = td(3) ! first surface - first element in block 8
    nel2s8bend = td(4) ! first surface - last element in block 8
    xo2 = td(5)      ! second surface - orign of tube x
    yo2 = td(6)      ! second surface - orign of tube y
c.... read record 10
    call dinput(td,6)
    if(errck) go to 11
c.... move properties into variables for later use
    nel3s1bini = td(1) ! first surface - first element in block 1
    nel3s1bend = td(2) ! first surface - last element in block 1
    nel3s2bini = td(3) ! first surface - first element in block 2
    nel3s2bend = td(4) ! first surface - last element in block 2
    nel3s3bini = td(5) ! first surface - first element in block 3
    nel3s3bend = td(6) ! first surface - last element in block 3
c.... read record 11
    call dinput(td,6)
    if(errck) go to 11
c.... move properties into variables for later use
    nel3s4bini = td(1) ! first surface - first element in block 4
    nel3s4bend = td(2) ! first surface - last element in block 4
    nel3s5bini = td(3) ! first surface - first element in block 5
    nel3s5bend = td(4) ! first surface - last element in block 5
    nel3s6bini = td(5) ! first surface - first element in block 6
    nel3s6bend = td(6) ! first surface - last element in block 6
c.... read record 12
    call dinput(td,6)
    if(errck) go to 11
c.... move properties into variables for later use
    nel3s7bini = td(1) ! first surface - first element in block 7

```

```

    nel3s7bend = td(2) ! first surface - last element in block 7
    nel3s8bini = td(3) ! first surface - first element in block 8
    nel3s8bend = td(4) ! first surface - last element in block 8
    xo3 = td(5)      ! third surface - orign of tube x
    yo3 = td(6)      ! third surface - orign of tube y
c.... read record 13
    call dinput(td,6)
    if(errck) go to 11
c.... move properties into variables for later use
    nel4s1bini = td(1) ! first surface - first element in block 1
    nel4s1bend = td(2) ! first surface - last element in block 1
    nel4s2bini = td(3) ! first surface - first element in block 2
    nel4s2bend = td(4) ! first surface - last element in block 2
    nel4s3bini = td(5) ! first surface - first element in block 3
    nel4s3bend = td(6) ! first surface - last element in block 3
c.... read record 14
    call dinput(td,6)
    if(errck) go to 11
c.... move properties into variables for later use
    nel4s4bini = td(1) ! first surface - first element in block 4
    nel4s4bend = td(2) ! first surface - last element in block 4
    nel4s5bini = td(3) ! first surface - first element in block 5
    nel4s5bend = td(4) ! first surface - last element in block 5
    nel4s6bini = td(5) ! first surface - first element in block 6
    nel4s6bend = td(6) ! first surface - last element in block 6
c.... read record 15
    call dinput(td,6)
    if(errck) go to 11
c.... move properties into variables for later use
    nel4s7bini = td(1) ! first surface - first element in block 7
    nel4s7bend = td(2) ! first surface - last element in block 7
    nel4s8bini = td(3) ! first surface - first element in block 8
    nel4s8bend = td(4) ! first surface - last element in block 8
    xo4 = td(5)      ! fourth surface - orign of tube x
    yo4 = td(6)      ! fourth surface - orign of tube y
c.... output current parameters
    write(iow,2000) wd(i+1),d(1),d(2),d(3),l,k,d(14),d(11),d(12)
    if(ior.lt.0) then
        write(*,2000) wd(i+1),d(1),d(2),d(3),l,k,d(14),d(11),d(12)
    endif
    d(4) = d(3)
c.... integration properties
    d(5) = l
    d(6) = k
c
    if(i.eq.2) d(13) = d(2)
c.... total number of integration points initialized to zero
c    (pgauss01 called below to set it)
    lint = 0
    return
c

```



```

c
c isw=2: Check element for input errors
c
c.... subroutine ckisop checks for missing data and negative
c      jacobians at nodes
2      call ckisop(ix,xl,shp,ndm)
      return
c
c isw=3: Compute element stiffness matrix (s) and
c      stress-divergence vector (p) (residual force vector)
c
c.... generate natural coordinates and weights of integration
c      points (only done once when lint=0)
      if(1*.ne.lint) call pgauss01(l,lint,sg,tg,wg)
c.... Element jacobian computed as in Reddy formulation.
c.... Straight vector multiplication of Na,xl x Nb,xl for K
c
c.... perform integration by looping over int. points, summing
c.... integration loop calculates force vector due to body forces
c.... and integrals of N_a,i times N_b,j for later stiffness
c.... calculation
c
      do 111 ii=1,nst
        do 112 jj=1,nst
          s(ii,jj)=0.d0
112      continue
111      continue
      d2 = d(3)*d(8)*d(10)*d(13) ! x-force term
      d3 = d(3)*d(9)*d(10)*d(13) ! y-force term
      d4 = d(14) ! internal heat generation
      do 330 l = 1,lint
        call shape02(sg(l),tg(l),xl,shp,xsj,ndm,.false.,nalfa,shp1)
        xsj = xsj*wg(l)
c
c.... including weight and jacobian determinant in viscosity
c.... and in density
c
      dmu = d(1)*xsj
      rho = d(3)*xsj
      bous(1) = d(8)*d(10)*rho
      bous(2) = d(9)*d(10)*rho
      rk = d(12)*xsj

```

```

c
c.... big loop to assemble the jacobian matrix (12x12 for linear elements)
c
  nline = 0
  do 33 ii=1,nel ! loop over a
    ncol = 0
    do 44 jj=1,nel ! loop over b
      do 55 ki=1,ndm
        do 66 kj = 1,ndm
          taux(ki,kj) = shp(ki,ii)*shp(kj,jj)*dmu
66      continue
55      continue
    do ki = 1,ndm
      do kj = 1,ndm
        if(ki.eq.1.and.kj.eq.1) then
          rkbar(ki,kj) = 2.d0*taux(1,1)+taux(2,2)
        else
          if(ki.eq.2.and.kj.eq.2) then
            rkbar(ki,kj) = taux(1,1)+2.d0*taux(2,2)
          else
            rkbar(ki,kj) = taux(ki,kj)
          endif
        endif
      enddo
    enddo
  enddo

c
c.... compute thermal diffusion and
c Boussinesq vector
c
  do ik=1,ndm
    b(ik)= bous(ik)*shp(3,ii)*shp(3,jj) ! B_i
    rl(ik)= rk*shp(ik,ii)*shp(ik,jj) ! Lj(uj)
  enddo

c
c.... constructing jacobian initially with K, B and L (Reddy notation)
c
  s(ii+nline,jj+ncol) = rkbar(1,1)
  & +s(ii+nline,jj+ncol)
  s(ii+nline,jj+1+ncol) = rkbar(1,2)
  & +s(ii+nline,jj+1+ncol)
  s(ii+nline,jj+2+ncol) = b(1)+s(ii+nline,jj+2+ncol)
  s(ii+1+nline,jj+ncol) = rkbar(2,1)
  & +s(ii+1+nline,jj+ncol)
  s(ii+1+nline,jj+1+ncol) = rkbar(2,2)+s(ii+1+nline,jj+1+ncol)
  s(ii+1+nline,jj+2+ncol) = b(2)+s(ii+1+nline,jj+2+ncol)
  s(ii+2+nline,jj+2+ncol) = rl(1)+rl(2)
  & +s(ii+2+nline,jj+2+ncol)

c
c.... compute force vector with K,L and B (Reddy notation)
c
  p(ii+nline) = p(ii+nline)-(-

```

```

& rkbar(1,1)*ul(1,jj)+rkbar(1,2)*ul(2,jj)+b(1)*ul(3,jj))
p(ii+1+nline) = p(ii+1+nline)-(b(2)*ul(3,jj)
& +rkbar(2,2)*ul(2,jj)+rkbar(2,1)*ul(1,jj))
p(ii+2+nline) = p(ii+2+nline)-(rl(1)+rl(2))*
& ul(3,jj)
ncol = ncol+2
44 continue
nline = nline+2
33 continue
c
c
c.... now compute gravity and thermal loads
j1 = 1
do j=1,nel
p(j1) = p(j1)+d2*shp(3,j)*xsj
p(j1+1) = p(j1+1)+d3*shp(3,j)*xsj
p(j1+2) = p(j1+2)+d4*shp(3,j)*xsj
j1 = j1+ndf
enddo
330 continue ! end of first gaussian integration
c
c.... now we have the jacobian and residual vector with K
c.... and we have to add the K^ contribution, performing a
c.... reduced order Gaussian Integration (one level)
c
lint = 0
l = d(5)-1
if(l*.ne.lint) call pgauss01(l,lint,sg,tg,wg)
do 370 l=1,lint
call shape02(sg(l),tg(l),xl,shp,xsj,ndm,.false.,nalfa,shp1)
xsj = xsj*wg(l)
c
c.... including weight amd Jacobian determinant in the
c.... penalty parameter
c
gama = d(2)*xsj
nline = 0
do 34 ii=1,nel ! loop over a
ncol = 0
do 45 jj = 1, nel !loop over b
do 56 ki=1,ndm
do 67 kj=1,ndm
taux(ki,kj) = shp(ki,ii)*shp(kj,jj)*gama
67 continue
56 continue
c
c.... adding the new penalty contribution into "s" (big matrix)
c
s(ii+nline,jj+ncol) = s(ii+nline,jj+ncol) + taux(1,1)
s(ii+nline,jj+1+ncol) = s(ii+nline,jj+1+ncol) + taux(1,2)
s(ii+1+nline,jj+ncol) = s(ii+1+nline,jj+ncol) + taux(2,1)

```

```

      s(ii+1+nline,jj+1+ncol) = s(ii+1+nline,jj+1+ncol) + taux(2,2)
c
c....   update force vector with K^
c
      p(ii+nline) = p(ii+nline)-(taux(1,1)*ul(1,jj)+taux(1,2)*
&   ul(2,jj))
      p(ii+1+nline) = p(ii+1+nline)-(taux(2,1)*ul(1,jj)+taux(2,2)*
&   ul(2,jj))
c2345678901234567890123456789012345678901234567890123456789012
      ncol = ncol+2
45      continue
      nline = nline+2
34      continue
370 continue
c
c.... compute inertia terms with upwinding scheme
c
      nalfa = d(7)
      call pgauss01(1,lint,sg,tg,wg)
      call shape02(sg(1),tg(1),xl,shp,xsj,ndm,.false.,nalfa,shp1)
      xsj = xsj*wg(1)
      rho = d(3)*xsj
      du = d(3)*d(11)*xsj
      do jn=1,ndm
         uaux(jn)=0.d0
         uaux1=0.d0
c
c.... compute sum over c of N_c * u_c and N_c * 1
c
         do ir=1,nel
            uaux(jn)=uaux(jn)+shp(3,ir)*ul(jn,ir)
            uaux1=uaux1+shp(3,ir)
         enddo
      enddo
      nline = 0
      do 341 ii=1,nel ! loop over a
         ncol = 0
         do 345 jj = 1, nel !loop over b
c
c....   compute inertia contribution
c
            if (nalfa.eq.0) then
do ik=1,ndm
               cin(ik)= rho*shp(3,ii)*uaux(ik)*shp(ik,jj) ! Cj(uj)
               din(ik)= du*shp(3,ii)*uaux(ik)*shp(ik,jj) ! Dj(uj)
               cin(ik+ndm)=rho*shp(3,ii)*uaux1*shp(ik,jj) ! Cj(1)
               din(ik+ndm)=du*shp(3,ii)*uaux1*shp(ik,jj) ! Dj(1)
            enddo
            else
do ik=1,ndm
               cin(ik)= rho*shp1(3,ii)*uaux(ik)*shp1(ik,jj) ! Cj(uj)

```

```

        din(ik)= du*shp1(3,ii)*uaux1(ik)*shp1(ik,jj) ! Dj(uj)
        cin(ik+ndm)=rho*shp1(3,ii)*uaux1*shp1(ik,jj) ! Cj(1)
        din(ik+ndm)=du*shp1(3,ii)*uaux1*shp1(ik,jj) ! Dj(1)
    enddo
    endif
c
c.... updating jacobian with C and D
c
    s(ii+nline,jj+ncol) = cin(1)+cin(3)*ul(1,jj)+cin(2)
    & +s(ii+nline,jj+ncol)
    s(ii+nline,jj+1+ncol) = cin(4)*ul(1,jj)
    & +s(ii+nline,jj+1+ncol)
    s(ii+1+nline,jj+ncol) = cin(3)*ul(2,jj)
    & +s(ii+1+nline,jj+ncol)
    s(ii+1+nline,jj+1+ncol) = cin(1)+cin(2)+cin(4)*ul(2,jj)+
    & s(ii+1+nline,jj+1+ncol)
    s(ii+2+nline,jj+ncol) = din(3)*ul(3,jj)+s(ii+2+nline,jj+ncol)
    s(ii+2+nline,jj+1+ncol) = din(4)*ul(3,jj)
    & +s(ii+2+nline,jj+1+ncol)
    s(ii+2+nline,jj+2+ncol) = din(1)+din(2)
    & +s(ii+2+nline,jj+2+ncol)
c
c.... compute force vector with C and D (Reddy notation)
c
    p(ii+nline) = p(ii+nline)-(cin(1)*ul(1,jj)+cin(2)*ul(1,jj))
    p(ii+1+nline) = p(ii+1+nline)-(cin(1)*ul(2,jj)+cin(2)
    & *ul(2,jj))
    p(ii+2+nline) = p(ii+2+nline)-((din(1)+din(2))*
    & ul(3,jj))
    ncol = ncol+2
345 continue
    nline = nline+2
341 continue
c
c.... Now we have assembled the elemental jacobian matrix
c.... completely for the FLOW/HEAT TRANSFER problem with upwinding
c.... and the
c.... elemental residual vector
    return
c
c
c
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
c isw=4,6: compute element internal force vector (isw=6)
c         or element fluxes for printing (isw=4)
c
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
4    l = d(5)
    if(isw.eq.4) l = d(6)
c.... generate natural coordinates and weights of integration

```

```

c    points (only done once when lint=0)
      if(l*.ne.lint) call pgauss01(l,lint,sg,tg,wg)
c.... compute element stresses, strains, and forces
      sumx=0.d0
      sumy=0.d0
      sumdx=0.d0
      sumdy=0.d0
      do 440 l = 1,lint
c.... compute element shape functions and derivatives
      call shape02(sg(l),tg(l),xl,shp,xsj,ndm,.false.,nalfa,shp1)
c.... compute strains and coordinates of current quadrature pt.
      do 410 i = 1,8
        eps(i) = 0.0
410    continue
      xx = 0.0
      yy = 0.0
      do 420 j = 1,nel
        xx = xx + shp(3,j)*xl(1,j) ! compute global coord.
        yy = yy + shp(3,j)*xl(2,j) ! of the gaussian points
        eps(1) = eps(1) + shp(1,j)*ul(1,j) ! du/dx
        eps(2) = eps(2) + shp(2,j)*ul(2,j) ! dv/dy
        eps(3) = eps(3) + shp(2,j)*ul(1,j) ! du/dy
        eps(4) = eps(4) + shp(1,j)*ul(2,j) ! dv/dx
        eps(5) = eps(5) + shp(3,j)*ul(1,j) ! u
        eps(6) = eps(6) + shp(3,j)*ul(2,j) ! v
        eps(7) = eps(7) + shp(1,j)*ul(3,j) ! dT/dx
        eps(8) = eps(8) + shp(2,j)*ul(3,j) ! dT/dy
420    continue
c
c    prepare gaussian coordinates and heat fluxes for averaging later
c    Done on Dec 8, 1999
c
      sumx=sumx+xx
      sumy=sumy+yy
      sumdx=sumdx+eps(7)
      sumdy=sumdy+eps(8)
c
c.... compute pressures, stresses and vorticity
      sig(1) = -d(2)*(eps(1)+eps(2)) ! pressure
      sig(2) = eps(1) ! dimensionless ux-flux
      sig(3) = eps(3) ! dimensionless uy-flux
c    sig(4) = d(1)*eps(4) ! vx-flux
      sig(4) = eps(4) ! dimensionless vx-flux
      sig(5) = eps(2) ! dimensionless vy-flux
      sig(6) = eps(4)-eps(3) ! vorticity
      psi = dsqrt(eps(5)**2+eps(6)**2) ! velo. contour
c    dtx = -d(12)*eps(7) ! x-heat flux
c    dty = -d(12)*eps(8) ! y-heat flux
      dtx = eps(7) ! dimensionless x-heat flux
      dty = eps(8) ! dimensionless y-heat flux
c.... Stream function computation

```

```

ll=iabs(ix(1))
if(ll.eq.1) then
  psi1=d(15)
  dx1=xl(1,2)-xl(1,1)
  dy1=xl(2,2)-xl(2,1)
else
  psi1=psi1+ul(1,1)*dy1-ul(2,1)*dx1 ! streamfunction
  dx1=xl(1,2)-xl(1,1)
  dy1=xl(2,2)-xl(2,1)
endif
if(isw.eq.4) then
c
c.... print pressures, stresses, vorticity, velo. contour and streamfunction
c
c.... output pressures, stresses, vorticity, velo. contour and streamfunction
  mct = mct - 2
  if(mct.le.0) then
c    write(iow,2001) o,head
    if(ior.lt.0) then
c      write(*,2001) o,head
    endif
    mct = 50
  endif

c
c    write(iow,2002) n,ma,(sig(ii),ii=1,5),xx,yy,sig(6),
c    &      psi,dtx,dty,psi1
  if(ior.lt.0) then
c    write(*,2002) n,ma,(sig(ii),ii=1,5),xx,yy,sig(6),
c    &      psi,dtx,dty,psi1
569  continue    ! only a dummy argument
  endif

c
c.... THIS PART BELOW IS NOT USED IN A FLUID ELEMENT
c
  elseif(isw.eq.6) then
c
c.... compute the element internal force vector
c    (via the integral of B^t sigma)
c
    dv = xsj*wg(l)*d(14)
    j1 = 1
    do 430 j = 1,nel
      p(j1 ) = p(j1 ) - (shp(1,j)*sig(1) + shp(2,j)*sig(2))*dv
1      + d(11)*shp(3,j)*dv
      p(j1+1) = p(j1+1) - (shp(1,j)*sig(2) + shp(2,j)*sig(3))*dv
1      + d(12)*shp(3,j)*dv
      j1 = j1 + ndf
430    continue
  endif
440  continue
c

```

```

c
c      compute average central gaussian coordinate at the element
c      and average central heat fluxes
c
c
      dtx4=dtx
      dty4=dty
      xx=sumx/4.d0
      yy=sumy/4.d0
      dtx=sumdx/4.d0
      dty=sumdy/4.d0
c
c*****
c Compute total heat fluxes at tubes
c*****
      pi=4.d0*atan(1.d0)
c
c***** first tube *****
c
      if(n.ge.nel1s1bini.and.n.le.nel1s1bend) then
        call sintegral02(yo1,xo1,yy,xx,dty,dtx,icont,pi,sum)
      endif
      if(n.eq.nel1s1bend) then
        write(*,*)'fluxt1=',sum
c      icont=0
c      sum=0.d0
      endif
      if(n.ge.nel1s2bini.and.n.le.nel1s2bend) then
        call sintegral02(yo1,xo1,yy,xx,dty,dtx,icont,pi,sum)
      endif
      if(n.eq.nel1s2bend) then
        write(*,*)'fluxt2=',sum
c      icont=0
c      sum=0.d0
      endif
      if(n.ge.nel1s3bini.and.n.le.nel1s3bend) then
        call sintegral02(yo1,xo1,yy,xx,dty,dtx,icont,pi,sum)
      endif
      if(n.eq.nel1s3bend) then
        write(*,*)'fluxt3=',sum
c      icont=0
c      sum=0.d0
      endif
      if(n.ge.nel1s4bini.and.n.le.nel1s4bend) then
        call sintegral02(yo1,xo1,yy,xx,dty,dtx,icont,pi,sum)
      endif
      if(n.eq.nel1s4bend) then
        write(*,*)'fluxt4=',sum
c      icont=0
c      sum=0.d0
      endif

```



```

        if(n.ge.nel1s5bini.and.n.le.nel1s5bend) then
            call sintegral02(yo1,xo1,yy,xx,dty,dtx,icont,pi,sum)
        endif
        if(n.eq.nel1s5bend) then
            write(*,*)'fluxt5=',sum
c         icont=0
c         sum=0.d0
        endif
        if(n.ge.nel1s6bini.and.n.le.nel1s6bend) then
            call sintegral02(yo1,xo1,yy,xx,dty,dtx,icont,pi,sum)
        endif
        if(n.eq.nel1s6bend) then
            write(*,*)'fluxt6=',sum
c         icont=0
c         sum=0.d0
        endif
        if(n.ge.nel1s7bini.and.n.le.nel1s7bend) then
            call sintegral02(yo1,xo1,yy,xx,dty,dtx,icont,pi,sum)
        endif
        if(n.eq.nel1s7bend) then
            write(*,*)'fluxt7=',sum
c         icont=0
c         sum=0.d0
        endif
        if(n.ge.nel1s8bini.and.n.le.nel1s8bend) then
            call sintegral02(yo1,xo1,yy,xx,dty,dtx,icont,pi,sum)
        endif
        if(n.eq.nel1s8bend) then
            write(*,*)'fluxt8=',sum
            icont=0
c         sum=0.d0
        endif
c
c***** second tube *****
c
        if(n.ge.nel2s1bini.and.n.le.nel2s1bend) then
            call sintegral02(yo2,xo2,yy,xx,dty,dtx,icont,pi,sum)
        endif
        if(n.eq.nel2s1bend) then
            write(*,*)'fluxt9=',sum
c         icont=0
c         sum=0.d0
        endif
        if(n.ge.nel2s2bini.and.n.le.nel2s2bend) then
            call sintegral02(yo2,xo2,yy,xx,dty,dtx,icont,pi,sum)
        endif
        if(n.eq.nel2s2bend) then
            write(*,*)'fluxt10=',sum
c         icont=0
c         sum=0.d0
        endif

```

```

        if(n.ge.nel2s3bini.and.n.le.nel2s3bend) then
            call sintegral02(yo2,xo2,yy,xx,dtz,dtx,icont,pi,sum)
        endif
        if(n.eq.nel2s3bend) then
            write(*,*)'flux11=',sum
c         icont=0
c         sum=0.d0
        endif
        if(n.ge.nel2s4bini.and.n.le.nel2s4bend) then
            call sintegral02(yo2,xo2,yy,xx,dtz,dtx,icont,pi,sum)
        endif
        if(n.eq.nel2s4bend) then
            write(*,*)'flux12=',sum
c         icont=0
c         sum=0.d0
        endif
        if(n.ge.nel2s5bini.and.n.le.nel2s5bend) then
            call sintegral02(yo2,xo2,yy,xx,dtz,dtx,icont,pi,sum)
        endif
        if(n.eq.nel2s5bend) then
            write(*,*)'flux13=',sum
c         icont=0
c         sum=0.d0
        endif
        if(n.ge.nel2s6bini.and.n.le.nel2s6bend) then
            call sintegral02(yo2,xo2,yy,xx,dtz,dtx,icont,pi,sum)
        endif
        if(n.eq.nel2s6bend) then
            write(*,*)'flux14=',sum
c         icont=0
c         sum=0.d0
        endif
        if(n.ge.nel2s7bini.and.n.le.nel2s7bend) then
            call sintegral02(yo2,xo2,yy,xx,dtz,dtx,icont,pi,sum)
        endif
        if(n.eq.nel2s7bend) then
            write(*,*)'flux15=',sum
c         icont=0
c         sum=0.d0
        endif
        if(n.ge.nel2s8bini.and.n.le.nel2s8bend) then
            call sintegral02(yo2,xo2,yy,xx,dtz,dtx,icont,pi,sum)
        endif
        if(n.eq.nel2s8bend) then
            write(*,*)'flux16=',sum
c         fluxf=sum/pi
c         write(*,*)'fluxf=',fluxf
            icont=0
c         sum=0.d0
        endif
c

```

```

c***** third tube *****
c
  if(n.ge.nel3s1bini.and.n.le.nel3s1bend) then
    call integral02(yo3,xo3,yy,xx,dy,dtx,icont,pi,sum)
  endif
  if(n.eq.nel3s1bend) then
    write(*,*)'fluxt17=',sum
c    icont=0
c    sum=0.d0
  endif
  if(n.ge.nel3s2bini.and.n.le.nel3s2bend) then
    call integral02(yo3,xo3,yy,xx,dy,dtx,icont,pi,sum)
  endif
  if(n.eq.nel3s2bend) then
    write(*,*)'fluxt18=',sum
c    icont=0
c    sum=0.d0
  endif
  if(n.ge.nel3s3bini.and.n.le.nel3s3bend) then
    call integral02(yo3,xo3,yy,xx,dy,dtx,icont,pi,sum)
  endif
  if(n.eq.nel3s3bend) then
    write(*,*)'fluxt19=',sum
c    icont=0
c    sum=0.d0
  endif
  if(n.ge.nel3s4bini.and.n.le.nel3s4bend) then
    call integral02(yo3,xo3,yy,xx,dy,dtx,icont,pi,sum)
  endif
  if(n.eq.nel3s4bend) then
    write(*,*)'fluxt20=',sum
c    icont=0
c    sum=0.d0
  endif
  if(n.ge.nel3s5bini.and.n.le.nel3s5bend) then
    call integral02(yo3,xo3,yy,xx,dy,dtx,icont,pi,sum)
  endif
  if(n.eq.nel3s5bend) then
    write(*,*)'fluxt21=',sum
c    icont=0
c    sum=0.d0
  endif
  if(n.ge.nel3s6bini.and.n.le.nel3s6bend) then
    call integral02(yo3,xo3,yy,xx,dy,dtx,icont,pi,sum)
  endif
  if(n.eq.nel3s6bend) then
    write(*,*)'fluxt22=',sum
c    icont=0
c    sum=0.d0
  endif
  if(n.ge.nel3s7bini.and.n.le.nel3s7bend) then

```

```

        call sintegral02(yo3,xo3,yy,xx,dty,dtx,icont,pi,sum)
    endif
    if(n.eq.nel3s7bend) then
        write(*,*)'fluxt23=',sum
c        icont=0
c        sum=0.d0
    endif
        if(n.ge.nel3s8bini.and.n.le.nel3s8bend) then
            call sintegral02(yo3,xo3,yy,xx,dty,dtx,icont,pi,sum)
        endif
    if(n.eq.nel3s8bend) then
        write(*,*)'fluxt24=',sum
        icont=0
c        sum=0.d0
    endif

c***** fourth tube *****

        if(n.ge.nel4s1bini.and.n.le.nel4s1bend) then
            call sintegral02(yo4,xo4,yy,xx,dty,dtx,icont,pi,sum)
        endif
    if(n.eq.nel4s1bend) then
        write(*,*)'fluxt25=',sum
c        icont=0
c        sum=0.d0
    endif
        if(n.ge.nel4s2bini.and.n.le.nel4s2bend) then
            call sintegral02(yo4,xo4,yy,xx,dty,dtx,icont,pi,sum)
        endif
    if(n.eq.nel4s2bend) then
        write(*,*)'fluxt26=',sum
c        icont=0
c        sum=0.d0
    endif
        if(n.ge.nel4s3bini.and.n.le.nel4s3bend) then
            call sintegral02(yo4,xo4,yy,xx,dty,dtx,icont,pi,sum)
        endif
    if(n.eq.nel4s3bend) then
        write(*,*)'fluxt27=',sum
c        icont=0
c        sum=0.d0
    endif
        if(n.ge.nel4s4bini.and.n.le.nel4s4bend) then
            call sintegral02(yo4,xo4,yy,xx,dty,dtx,icont,pi,sum)
        endif
    if(n.eq.nel4s4bend) then
        write(*,*)'fluxt28=',sum
c        icont=0
c        sum=0.d0
    endif
        if(n.ge.nel4s5bini.and.n.le.nel4s5bend) then

```

```

        call sintegral02(yo4,xo4,yy,xx,dty,dtx,icont,pi,sum)
    endif
    if(n.eq.nel4s5bend) then
        write(*,*)'fluxt29=',sum
c      icont=0
c      sum=0.d0
    endif
    if(n.ge.nel4s6bini.and.n.le.nel4s6bend) then
        call sintegral02(yo4,xo4,yy,xx,dty,dtx,icont,pi,sum)
    endif
    if(n.eq.nel4s6bend) then
        write(*,*)'fluxt30=',sum
c      icont=0
c      sum=0.d0
    endif
    if(n.ge.nel4s7bini.and.n.le.nel4s7bend) then
        call sintegral02(yo4,xo4,yy,xx,dty,dtx,icont,pi,sum)
    endif
    if(n.eq.nel4s7bend) then
        write(*,*)'fluxt31=',sum
c      icont=0
c      sum=0.d0
    endif
    if(n.ge.nel4s8bini.and.n.le.nel4s8bend) then
        call sintegral02(yo4,xo4,yy,xx,dty,dtx,icont,pi,sum)
    endif
    if(n.eq.nel4s8bend) then
        write(*,*)'fluxt32=',sum
        fluxf=sum*0.013
        write(*,*)'fluxf=',fluxf
        write(*,*)'dtx4=',dtx4,'dtxavg=',dtx
        write(*,*)'dty4=',dty4,'dtyavg=',dty
        icont=0
        sum=0.d0
    endif
c
c*****
c  End of new part for computing total heat flux
c*****
    return
c
c
c
c%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
c%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
c isw=5: compute consistent mass matrix (for dynamics)
c
c%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
c%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
5    l = d(5)
c.... generate natural coordinates and weights of integration
c    points (only done once when lint=0)

```



```

c
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
c format statements for element 1
c
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
1000 format(3f10.0,3i10/6f10.0)
2000 format(/5x,'Fluid ',a4,' 4-NavierStokes flow Element'//
1 10x,'viscosity ',e12.5/10x,'penalty param',f8.5/
2 10x,'density ',e12.5/10x,'gauss pts/dir',i3/
3 10x,'fluxes pts ',i3/ 10x,' ',e12.5/
3 10x,' ',e12.5/10x,' ',e12.5/)
2001 format(a1,20a4//5x,'element fluxes'/' element material',
1 ' pressure ux-flux uy-flux vx-flux vy-flux ',
2 ' ' 1-coord 2-coord vorticity velocity modulus ',
3 'x-hflux y-hflux streamfunction')
2002 format(/2i9,5e12.3/2f9.3,5e12.3)
3000 format(' Input: mu, gama, rho, #-pts K, #-pts sig'/' >',$)
3001 format(' Input: thick, 1-body, 2-body'/' l>',$)
3002 format(' Input: thick, 1-body, 2-body'/' l>',$)
end
c #####
c subroutine integral02: computes values of integral Ii
c #####
subroutine sintegral02(yo1,xo1,yy,xx,dti,dtx,icont,pi,sum)
implicit real*8 (a-h,o-z)
common /teste/ tetaia,tetaip,dteta,dtna,dtnp
if(icont.eq.0) then
tetaia=atan(abs((yo1-yy)/(xo1-xx)))
aux1=tetaia
if(xx.gt.xo1) tetaia=pi-tetaia
dtna=fluxn(dti,dtx,aux1)
c write(*,*)'tetaia=',tetaia,'dtna=',dtna
icont=1
else
tetaip=atan(abs((yo1-yy)/(xo1-xx)))
aux2=tetaip
if(xx.gt.xo1) tetaip=pi-tetaip
dtnp=fluxn(dti,dtx,aux2)
dteta=tetaip-tetaia
sum=sum+((dtnp+dtna)/2)*dteta
c write(*,*)'tetaia,tetaip=',tetaia,tetaip
c write(*,*)'dtna=',dtna,'dtnp=',dtnp,'sum=',sum
tetaia=tetaip
dtna=dtnp
icont=1
endif
return
end
c #####

```

```

c    function to compute the normal heat flux to the
c    surface
c    #####
c    function fluxn(qy,qx,teta)
c    implicit real*8 (a-h,o-z)
c    common /ellipse/ sa,sb
c
c    circular or elliptical section
c
c    aux=sqrt(qx*qx+qy*qy)
c    aux=abs(qx)*cos(teta)+abs(qy)*sin(teta)
c    aux1=abs(qx)*1/(sqrt(1+(sa**2/sb**2)*(tan(teta))**2))
c    aux2=abs(qy)*1/(sqrt(1+(sb**2/sa**2)/(tan(teta))**2))
c    aux=aux1+aux2
c    fluxn=aux
c    return
c    end
c
c    #####
c    subroutine shape02: computes values of shape function
c    and its derivatives at a point
c    #####
c
c    subroutine shape02(s,t,xl,shp,xsj,ndm,flg,nalfa,shp1)
c    implicit real*8 (a-h,o-z)
c    logical flg
c.... shape function routine for 4-node isoparametric quadrilaterals
c
c.... Given data:
c    s:  first natural coordinate of point
c    t:  second natural coordinate of point
c    xl: array of localized coordinates
c        (dimensioned ndm by 4)
c
c.... Output:
c    shp: array containing evaluations of all element shape
c        functions and derivatives at (s,t)
c        shp(1,a)=N_a,x (s,t)
c        shp(2,a)=N_a,y (s,t)
c        shp(3,a)=N_a (s,t)
c        for a=1,2,3,4
c    xsj: jacobian of coordinate transformation
c        (i.e., det(dx/dxi))
c    shp1: array containing evaluations of all element shape
c        functions and derivatives at s=t=nalfa/3 (upwinding point)
c        shp1(1,a)=N_a,x (s,t)
c        shp1(2,a)=N_a,y (s,t)
c        shp1(3,a)=N_a (s,t)
c        for a=1,2,3,4
c
c    real*8 xl(ndm,4),shp(3,4),shp1(3,4)

```



```

sh = 0.5*s
th = 0.5*t
sp = 0.5 + sh
tp = 0.5 + th
sm = 0.5 - sh
tm = 0.5 - th
if (nalfa.ne.0) then
  s1 = nalfa / 3.d0
  t1 = s1
  sh1 = 0.5*s1
  th1 = 0.5*t1
  sp1 = 0.5 + sh1
  tp1 = 0.5 + th1
  sm1 = 0.5 - sh1
  tm1 = 0.5 - th1
  shp1(3,1) = sm1*tm1
  shp1(3,2) = sp1*tm1
  shp1(3,3) = sp1*tp1
  shp1(3,4) = sm1*tp1
endif
c.... evaluate the shape functions
shp(3,1) = sm*tm
shp(3,2) = sp*tm
shp(3,3) = sp*tp
shp(3,4) = sm*tp
c.... calculates the partial derivatives of x ,y with respect
c   to xi and eta (including a factor of 4) and put in xs,xt,ys,yt
xo = xl(1,1)-xl(1,2)+xl(1,3)-xl(1,4)
xs = -xl(1,1)+xl(1,2)+xl(1,3)-xl(1,4) + xo*t
xt = -xl(1,1)-xl(1,2)+xl(1,3)+xl(1,4) + xo*s
yo = xl(2,1)-xl(2,2)+xl(2,3)-xl(2,4)
ys = -xl(2,1)+xl(2,2)+xl(2,3)-xl(2,4) + yo*t
yt = -xl(2,1)-xl(2,2)+xl(2,3)+xl(2,4) + yo*s
if (nalfa.ne.0) then
  xol = xl(1,1)-xl(1,2)+xl(1,3)-xl(1,4)
  xs1 = -xl(1,1)+xl(1,2)+xl(1,3)-xl(1,4) + xol*t1
  xt1 = -xl(1,1)-xl(1,2)+xl(1,3)+xl(1,4) + xol*s1
  yol = xl(2,1)-xl(2,2)+xl(2,3)-xl(2,4)
  ys1 = -xl(2,1)+xl(2,2)+xl(2,3)-xl(2,4) + yol*t1
  yt1 = -xl(2,1)-xl(2,2)+xl(2,3)+xl(2,4) + yol*s1
endif
c.... compute jacobian, put in xsj
xsj1 = xs*yt - xt*ys
xsj = 0.0625*xsj1
if (nalfa.ne.0) xsj2 = xs1*yt1 - xt1*ys1
c.... compute shape function derivatives with respect to x and y
c   and store in shp (see pp. 146-147 of Hughes)
if(.not.flg) then
  if(xsj1.eq.0.0d0) xsj1 = 1.0
  xs = (xs+xs)/xsj1
  xt = (xt+xt)/xsj1

```

```

ys = (ys+ys)/xsj1
yt = (yt+yt)/xsj1
ytm = yt*tm
ysm = ys*sm
ytp = yt*tp
ysp = ys*sp
xtm = xt*tm
xsm = xs*sm
xtp = xt*tp
xsp = xs*sp
shp(1,1) = - ytm+ysm
shp(1,2) = ytm+ysp
shp(1,3) = ytp-ysp
shp(1,4) = - ytp-ysm
shp(2,1) = xtm-xsm
shp(2,2) = - xtm-xsp
shp(2,3) = - xtp+xsp
shp(2,4) = xtp+xsm
endif
if (nalfa.ne.0) then
  if(.not.flg) then
    if(xsj2.eq.0.0d0) xsj2 = 1.0
    xs1 = (xs1+xs1)/xsj2
    xt1 = (xt1+xt1)/xsj2
    ys1 = (ys1+ys1)/xsj2
    yt1 = (yt1+yt1)/xsj2
    ytm1 = yt1*tm1
    ysm1 = ys1*sm1
    ytp1 = yt1*tp1
    ysp1 = ys1*sp1
    xtm1 = xt1*tm1
    xsm1 = xs1*sm1
    xtp1 = xt1*tp1
    xsp1 = xs1*sp1
    shp1(1,1) = - ytm1+ysm1
    shp1(1,2) = ytm1+ysp1
    shp1(1,3) = ytp1-ysp1
    shp1(1,4) = - ytp1-ysm1
    shp1(2,1) = xtm1-xsm1
    shp1(2,2) = - xtm1-xsp1
    shp1(2,3) = - xtp1+xsp1
    shp1(2,4) = xtp1+xsm1
  endif
endif
end
c
c
c #####
c  subroutine stcn02: computes quantities necessary for
c    nodal projection of element stresses to nodes
c  #####

```

```

c
c *** representation of stress field in terms of NODAL
c   stresses requires a reprojection of the stress
c   field. The technique used is a least squares
c   method (see Hughes, pp. 226-228 for a brief and
c   somewhat cryptic presentation)
c
c *** primary output variables of this routine:
c
c   s: consistent projection matrix
c   dt: lumped (diagonal) projection matrix, stored
c       in vector form
c   st: weighted stress vector

subroutine stcn02(ix,d,xl,ul,tl,s,shp,dt,st,ndf,ndm,nel,numnp)
implicit double precision (a-h,o-z)
dimension dt(numnp),st(numnp,1),xl(ndm,1),shp(3,4),sig(6),
1   d(1),eps(6),ul(ndf,1),tl(1),ix(1),s(nel,1),sg(16),tg(16),
2   wg(16),shp1(3,4)
c   data ss/-1.d0, 1.d0,1.d0,-1.d0, 0.d0,1.d0,0.d0,-1.d0,0.d0/
c   data tt/-1.d0,-1.d0,1.d0, 1.d0,-1.d0,0.d0, 1.d0, 0.d0,0.d0/
c..... lumped and consistent projection routine
1 = d(5)
if(1*.ne.lint) call pgauss01(1,lint,sg,tg,wg)
call pzero(s,nel*nel)
c
c..... Note: do 300 loop is a loop over the integration points
c
do 300 l = 1,lint
  call shape01(sg(l),tg(l),xl,shp,xsj,ndm,.false.,nalfa,shp1)
  xsj=xsj*wg(l)
  do 100 i = 1,8
    eps(i) = 0.0d0
100  continue
  xx = 0.0
  yy = 0.0
  do 110 i = 1,nel
    xx = xx + shp(3,i)*xl(1,i) ! compute global coord.
    yy = yy + shp(3,i)*xl(2,i) ! of the gaussian points
    eps(1) = eps(1) + shp(1,i)*ul(1,i) !pressure
    eps(2) = eps(2) + shp(2,i)*ul(2,i)
    eps(3) = eps(3) + shp(2,i)*ul(1,i)
    eps(4) = eps(4) + shp(1,i)*ul(2,i)
    eps(5) = eps(5) + shp(3,i)*ul(1,i) ! u
    eps(6) = eps(6) + shp(3,i)*ul(2,i) ! v
    eps(7) = eps(7) + shp(1,i)*ul(3,i) ! dT/dx
    eps(8) = eps(8) + shp(2,i)*ul(3,i) ! dT/dy
    ss = shp(3,i)*xsj
c.... compute consistent projection matrix
    do 105 j = 1,nel
      s(i,j) = s(i,j) + ss*shp(3,j)
    105  continue
  300  continue

```

```

105    continue
110    continue
c.... compute stresses
c
    sig(1) = -d(2)*(eps(1)+eps(2)) ! pressure
    sig(2) = d(1)*eps(1) ! ux-flux
    sig(3) = d(1)*eps(3) ! uy-flux
    sig(4) = d(1)*eps(4) ! vx-flux
    sig(5) = d(1)*eps(2) ! vy-flux
    sig(6) = eps(4)-eps(3) ! vorticity
    psi = dsqrt(eps(5)**2+eps(6)**2)
    dtx = eps(7) ! x-hflux
    dty = eps(8) ! y-hflux
c.... Stream function computation
    ll=iabs(ix(1))
    if(ll.eq.1) then
        psi1=0.d0
        dx1=xl(1,2)-xl(1,1)
        dy1=xl(2,2)-xl(2,1)
    else
        psi1=psi1+ul(1,1)*dy1-ul(2,1)*dx1 ! streamfunction
        dx1=xl(1,2)-xl(1,1)
        dy1=xl(2,2)-xl(2,1)
    endif
c.... compute lumped projection and assemble the stress integrals
    do 120 j = 1,nel
        ll = iabs(ix(j))
        if(ll.gt.0) then
            xg = xsj*shp(3,j)
            dt(ll) = dt(ll) + xg
            st(ll,1) = st(ll,1) + sig(1)*xg ! pressure
            st(ll,2) = st(ll,2) + sig(6)*xg ! vorticity
            st(ll,3) = st(ll,3) + sig(2)*xg ! du/dx
            st(ll,4) = st(ll,4) + psi*xg ! velo. contour
            st(ll,9) = st(ll,9) + dtx*xg ! x-hflux
            st(ll,10) = st(ll,10) + dty*xg ! y-hflux
            st(ll,11) = st(ll,11) + psi1*xg ! streamfunction
        endif
120    continue
300    continue
    return
end

```

## REFERÊNCIAS BIBLIOGRÁFICAS

- Anand, N. K.; Kim, S. H.; Fletcher, L. S. The effect of plate spacing on free convection between heated parallel plates. *J. Heat Transfer*, 114: 515-518 (1992).
- Bar-Cohen, A. & Rohsenow, W. M. Thermally optimum spacing of vertical, natural convection cooled, parallel plates. *J. Heat Transfer*, 106: 116-123 (1984).
- Bar-Cohen, A. & Kraus, A. D. *Advances in Thermal Modeling of Electronic Components and Systems*. Vol. 2 ASME Press, New York (1990).
- Bejan, A. & Sciubba, E. The optimal spacing of parallel plates cooled by forced convection. *Int. J. Heat Mass Transfer*, 35: 3259-3264 (1992).
- Bejan, A. & Morega, A. M. Optimal arrays of pin fins in laminar forced convection. *J. Heat Transfer*, 115: 75-81 (1993).
- Bejan, A. *Convection Heat Transfer*. 2nd Edition, Wiley, New York (1995).
- Bejan, A.; Fowler, A. J.; Stanescu, G. The optimal spacing between horizontal cylinders in a fixed volume cooled by natural convection. *Int. J. Heat Mass Transfer*, 38: 2047-2055 (1995).
- Bordalo, S. N. & Saboya, F. E. M. Determinação experimental dos coeficientes de perda de carga em trocadores de calor de tubos circulares e elípticos aletados. *Proc. 13º COBEM, Conferência Brasileira em Engenharia Mecânica*, Belo Horizonte, Brasil (1995).
- Brauer, H. Compact heat exchangers. *Chem. Process Eng.*, 451-460 (August 1964).
- Dongarra, J.J.; Bunch, J. R.; Moler, C. B.; Stewart, G. W. *LINPAK Users' Guide*, Society for Industrial and Applied Mathematics, Philadelphia (1979).

- Fowler, A. J.; Ledezma, G. A.; Bejan, A. Optimal geometric arrangement of staggered plates in forced convection, *Int. J. Heat Mass Transfer*, 8: 1795-1805 (1997).
- Hirata, M.; Kakita, Y.; Yada, Y.; Hirose, Y.; Morikawa, T.; Ecomoto, H. Temperature distribution of finned integrated circuits, *Fijitsu Sci. Technol. J.*, 6: 91-115 (1970).
- Hughes, T. J. R., A simple scheme for developing upwind finite elements, *Int. J. Numerical Methods in Engineering*, Vol. 12, pp. 1359-1365 (1978).
- Hughes, T. J. R., Liu, W. K., e Brooks, A., "Review finite element analysis of incompressible viscous flows by penalty function formulation", *Journal of Computational Physics*, Vol. 30, pp. 1- 60 (1979).
- Kays, W. A. & London, A. L. Compact heat transfer characteristics of air cooled heat exchangers for air conditioning devices Bull, *JSME*, 11: 303-311 (1968).
- Kim, S. H.; Anand, N. K.; Fletcher, L. S. Free convection between series of vertical parallel plates with embedded line heat sources, *J. Heat Transfer*, 113: 108-115 (1991).
- Knight, R. W.; Goodling, J. S.; Hall, D. J. Optimal thermal design of forced convection heat sinks-analytical, *J. Electronic Packaging*, 113: 313-321 (1991).
- Knight, R. W.; Goodling, J. S.; Gross, B. E. Optimal thermal design of air cooled forced convection finned heat sinks-experimental verification, *IEEE Trans. Components, Hybrids Manufacturing Technol.* 15: 754-760 (1992).
- Matsushima, H.; Yanagida, T.; Kondo, Y. Algorithm for predicting the thermal resistance of finned LSI packages mounted on a circuit board, *Heat Transfer-Jap. Res.*, 21: 504-517 (1992).
- Nakayama, W.; Matsushima, H.; Goel, P. Forced convective heat transfer from arrays of finned packages, *Em Cooling Technology for Electronic Equipment*, (Editado por W. Aung), pp. 195-210, Hemisphere, New York (1988).

- Peterson, G. P. & Ortega, A. Thermal control of electronic equipment and devices, *Adv. Heat Transfer*, 20: 181-314 (1990).
- Reddy, J. N.; Gartling, D. K. The Finite Element Method in Heat Transfer and Fluid Dynamics, CRC (1994).
- Rocha, L. A. O.; Saboya, F. E. M.; Vargas, J. V. C. A comparative study of elliptical and circular sections in one and two-row tubes and plate fin heat exchangers, *Int. J. Heat Fluid Flow*, 18: 247-252 (1997).
- Rosman, E. C. Performance of tube and plate fin heat exchangers. Tese de Mestrado, Department of Mechanical Engineering, Pontificia Universidade Católica do Rio de Janeiro, Brazil, (1979).
- Rosman, E. C.; Carajilescov, P.; Saboya, F. E. M. Performance of tube of one and two-row tube and plate fin heat exchangers. *J. Heat Transfer*, 106: 627-632 (1984).
- Saboya, F. E. M. Local and average transfer coefficients in plate fin and tube configurations. Ph.D. thesis, Department of Mechanical Engineering, University of Minnesota, Minneapolis, MN. (1974).
- Saboya, F. E. M. & Sparrow, E. M. Experiments on a three-row fin and tube heat exchangers, *J. Heat Transfer*, 98: 520-522 (1976).
- Schulemberg, F. Finned elliptical tubes and their applications in air-cooled heat exchangers. *J. Eng. Ind.*, 88: 179-190 (1966).
- Shepherd, D. G. Performance of one-row tube coils with thin, plate fins, low velocity forced convection. *Heating, Piping and Air Conditioning*, 28: 137-144 (1956).
- Stanescu, G.; Fowler, A. J.; Bejan, A. The optimal spacing of cylinders in free-stream cross-flow forced convection, *Int. J. Heat Mass Transfer*, em impressão (1996).

Ximenes, M. P. Heat and mass transfer in elliptical tubes and plate fin heat exchangers. Ph.D. thesis, Mechanical Engineering Division, Air Force Institute of Technology, São José dos Campos, Brazil, (1981).

Zienkiewicz, O. C.; Taylor, R. L. The Finite Element Method, McGraw-Hill, London (1989).