

Universidade Estadual de Campinas
Faculdade de Engenharia Química
Área de Concentração: Desenvolvimento de Processos Químicos

*Método de Colocação Ortogonal em
Elementos Finitos Aplicado à
Solução de Problemas Bidimensionais
de Escoamento Laminar em Dutos Cilíndricos*

Autor: Evandro Steffani

Orientador: Prof. Dr. Milton Mori

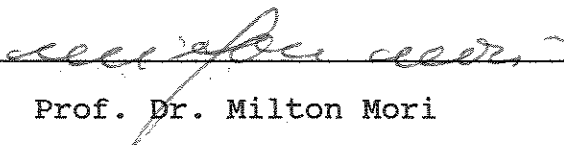
Tese apresentada à Faculdade de Engenharia Química como
parte dos requisitos exigidos para a obtenção do título de
Mestre em Engenharia Química

CAMPINAS - SP - BRASIL


Dezembro / 1993

UNICAMP
BIBLIOTECA CENTRAL

Tese defendida e aprovada, em 10 de dezembro de 1993, pela banca examinadora constituída pelos professores:


Prof. Dr. Milton Mori


Prof. Dr. Renato Machado Cotta


Prof. Dr. Paulo D. Castro Lobo

Agradecimentos

Ao meu orientador, professor Milton Mori, pela amizade e compreensão.

À minha família, pelo constante apoio.

Aos colegas de república Lúcio Cardozo, Luís Stragevitch, Atilano Vegini, Rodrigo de Bortoli e Georges Kaskantzis, pelo companheirismo e amizade.

Ao pessoal do Laboratório de Modelagem e Simulação de Processos Químicos, em especial aos colegas Frederico Resende de Oliveira e Luís Antônio R. Muniz, pelas discussões e sugestões.

Aos professores Márcia A. Gomes-Ruggiero e Gilmar Mompean Munhoz da Cruz pela grande contribuição que me prestaram.

Ao CNPq, PETROBRAS e FAEP pelo apoio financeiro.

E a todos que de alguma maneira contribuíram para o meu trabalho.

Conteúdo

1	INTRODUÇÃO	1
2	REVISÃO BIBLIOGRÁFICA	2
2.1	Métodos Numéricos	3
2.1.1	Método dos Resíduos Ponderados	4
2.1.2	Polinômios Ortogonais	5
2.2	Histórico do Método de Colocação Ortogonal	7
3	MODELAGEM MATEMÁTICA	10
3.1	A Equação da Continuidade	10
3.2	A Equação do Movimento	12
3.3	Descrição do Problema Físico	16
4	MÉTODOS NUMÉRICOS PARA A SIMULAÇÃO	20
4.1	Método de Colocação Ortogonal	20
4.1.1	Fundamentação do Método	21
4.1.2	Colocação Ortogonal em Elementos Finitos .	23
4.2	Aplicação do Método de Colocação Ortogonal em Elementos Finitos ao Problema de Escoamento Laminar	25
5	RESULTADOS E ANÁLISE	34
5.1	Simulação a Gradiente de Pressão Constante	34
5.2	Simulação a Gradiente de Pressão Variável	45
6	CONCLUSÕES E SUGESTÕES	47
	APÊNDICE A	48
	APÊNDICE B	49
	REFERÊNCIAS BIBLIOGRÁFICAS	88

Lista de Figuras

3.1	Volume de controle através do qual escoa um fluido	10
3.2	Elemento de volume onde as setas indicam a direção na qual a componente axial de momentum é transportada através das superfícies	13
3.3	Esquema do problema considerado	16
4.1	Distribuição geométrica dos pontos de colocação para $N=3$	21
4.2	Domínio para o método de colocação ortogonal em elementos finitos utilizando polinômios cúbicos de Lagrange	23
4.3	Sistema discreto oriundo da aplicação do método de colocação ortogonal em elementos finitos	24
4.4	Divisão do domínio em elementos finitos	27
4.5	Esquema de numeração de variáveis no tubo	28
5.1	Simulação para Reynolds = 500 e malha 5×10 elementos	36
5.2	Simulação para Reynolds = 700 e malha 5×10 elementos	37
5.3	Simulação para Reynolds = 1000 e malha 5×10 elementos	38
5.4	Simulação para Reynolds = 500 e malha 10×10 elementos	39
5.5	Simulação para Reynolds = 700 e malha 10×10 elementos	40
5.6	Simulação para Reynolds = 1000 e malha 10×10 elementos	41
5.7	Simulação para Reynolds = 500 e malha 10×15 elementos	42
5.8	Simulação para Reynolds = 700 e malha 10×15 elementos	43
5.9	Simulação para Reynolds = 1000 e malha 10×15 elementos	44

Lista de Tabelas

2.1	Coefficientes dos polinômios de Lagrange	6
2.2	Pontos de colocação (raízes dos polinômios de Lagrange)	7
5.1	Resultados obtidos e CPU utilizada (gradiente de pressão constante)	35
5.2	Resultados obtidos e CPU utilizada (gradiente de pressão variável)	45
A.1	Matrizes de colocação ortogonal	48

Nomenclatura

A	matriz de colocação representando a derivada primeira
A_j	coeficientes dos polinômios de Lagrange
B	matriz de colocação representando a derivada segunda
C	matriz auxiliar de colocação
d	vetor dos coeficientes do polinômio de aproximação
d_i	componentes de d
D	matriz auxiliar de colocação
$f(x)$	função que deseja-se aproximar
h_k	comprimento do elemento k
HR_k	comprimento do elemento k na direção radial
HZ_j	comprimento do elemento j na direção axial
HRI	comprimento dos elementos radiais
HZI	comprimento dos elementos axiais
IE	índice que dá a linha do elemento
JE	índice que dá a coluna do elemento
L	comprimento do tubo
L_d	comprimento de entrada
N	número de pontos de colocação internos
NTR	número de elementos na direção radial
NTZ	número de elementos na direção axial
P	pressão do fluido
P_L	pressão do fluido na saída do tubo
P_m	polinômio de Lagrange de grau m
P_o	pressão do fluido na entrada do tubo
Q	matriz auxiliar de colocação
r	coordenada radial
r_k	posição nodal no elemento k
R	coordenada radial normalizada
Re	número de Reynolds
R_t	raio do tubo
$R(x)$	residual

t	tempo
u	componente axial de velocidade
\bar{u}	velocidade axial média
u_{max}	velocidade axial máxima
u_0	velocidade axial na entrada do tubo
U	variável independente normalizada
v	componente radial de velocidade
$va(k)$	variável k do sistema de equações não-lineares obtido pela discretização
$W(x)$	função peso ortogonal a $R(x)$
x	variável independente
x_j	variável independente no ponto de colocação j
x_k	posição nodal no elemento k
$y(x)$	função a ser aproximada
y'	derivada primeira da função a ser aproximada
y''	derivada segunda da função a ser aproximada
y_t	vetor da função tentativa
y_t'	vetor da derivada primeira da função tentativa
y_t''	vetor da derivada segunda da função tentativa
z	coordenada axial
Z	coordenada axial normalizada
z_j	posição nodal no elemento j

Letras Gregas

$\Delta x, \Delta y, \Delta z$	dimensões do elemento de volume
μ	viscosidade dinâmica do fluido
ν	viscosidade cinemática do fluido
ϕ	componente axial ou radial de velocidade

Símbolos

∇	operador nabla
∇^2	operador laplaciano

CAPÍTULO 1

Introdução

O presente trabalho propõe a utilização do método de colocação ortogonal em elementos finitos para a solução do problema de escoamento laminar bidimensional em dutos cilíndricos. Este problema consiste na solução das equações diferenciais parciais da continuidade e da componente axial da equação de Navier-Stokes.

O conteúdo do trabalho é dividido em 6 capítulos, contendo os passos desenvolvidos ao longo de sua confecção.

O capítulo 2 aborda o problema de escoamento incompressível de fluidos, métodos usados para sua solução e contém um histórico sobre o método de colocação ortogonal.

A modelagem matemática é o objeto do capítulo 3, onde descreve-se a obtenção das equações através de balanços em volumes de controle e apresenta-se o problema físico a ser resolvido.

A fundamentação do método de colocação ortogonal é apresentada no capítulo 4, assim como a sua aplicação ao problema considerado.

Os resultados obtidos, assim como uma análise dos mesmos, podem ser apreciados no capítulo 5.

Para finalizar, o capítulo 6 contém as conclusões e sugestões do trabalho.

CAPÍTULO 2

Revisão Bibliográfica

A análise do escoamento incompressível de fluidos em estado estacionário ou transiente consiste na solução de problemas de valores de contorno ou problemas de valores inicial e de contorno dados pela equação de Navier-Stokes.

Vários métodos numéricos para a solução das equações de Navier-Stokes vêm sendo desenvolvidos nos últimos anos. Um dos mais utilizados é o método dos elementos finitos, cuja popularidade foi crescendo a partir da década de setenta como uma ferramenta potente para a aplicação em tais casos.

Dentre os métodos dos elementos finitos destaca-se a utilização do método de Galerkin. Huyakorn et al.(1978) fizeram um estudo comparativo onde simularam o escoamento viscoso incompressível em estado estacionário através de uma expansão súbita, utilizando quatro tipos diferentes de elementos de interpolação mista. Como conclusão deste trabalho verificaram que o elemento Lagrangeano de nove nós gerava os resultados mais apurados.

Um estudo comparativo entre os métodos de Galerkin e o método de colocação ortogonal em elementos finitos utilizando polinômios cúbicos de Hermite foi realizado por Chang et al.(1979). Estes autores simularam o escoamento entre duas placas planas paralelas, caracterizando-o como um problema de comprimento de entrada. Concluíram que, para fins práticos de engenharia, o método de Galerkin com polinômios quadráticos dá resultados com a mesma precisão global mas a um custo menor que o método de colocação com polinômios de Hermite cúbicos, embora este último gere resultados

mais acurados para a velocidade média.

Segundo Jensen e Finlayson (1980), num estudo sobre os limites de oscilação para os métodos dos resíduos ponderados aplicados a equações convectivas-difusivas, as equações de Navier-Stokes apresentam dificuldades para sua solução quando os termos convectivos predominam, o que acontece para números de Reynolds elevados. Estas dificuldades manifestam-se por soluções que oscilam no espaço em contraste com o comportamento real. Neste estudo foram analisados os métodos de Galerkin e de colocação ortogonal em elementos finitos.

Na tentativa de obter-se métodos capazes de gerar boas predições a um baixo custo computacional desenvolveu-se, mais recentemente, o método dos volumes finitos (MVF). Neste método as grandezas escalares são tratadas no centro dos volumes de controle principais e as velocidades são tratadas no centro das faces dos mesmos. Este procedimento, recomendado por Patankar (1980), constitui a chamada malha deslocada (staggered grid). Este método (MVF) foi aplicado, recentemente, na simulação do escoamento turbulento em dutos cilíndricos por Carvalho (1993), obtendo-se resultados bastante coerentes.

2.1 - Métodos Numéricos

Grande parte dos problemas de simulação numérica em Engenharia Química envolve a solução de equações diferenciais ordinárias não-lineares. Dentro desta categoria podemos distinguir dois tipos de problemas: problemas de valor inicial (PVI) e problemas de valor de contorno (PVC). Vamos nos ater aqui a este último tipo. Entre os métodos utilizados para a solução de um PVC estão os métodos dos resíduos ponderados. Segundo Finlayson (1980), o melhor deles é o método de colocação ortogonal.

2.1.1 - Método dos Resíduos Ponderados

Consiste em escolher uma função tentativa como solução da equação diferencial. Seja a equação :

$$f(x, y, y', y'', \dots) = 0 \quad , \quad (2.1)$$

com,

$$(0 \leq x \leq 1).$$

Escolhendo-se uma função tentativa $y_t(x)=f(x)$ e substituindo na equação diferencial, temos:

$$f(x, y_t, y_t', y_t'', \dots) = R(x) \quad , \quad (2.2)$$

onde $R(x)$ representa o residual. Fazendo-se a integração, temos:

$$\int_0^1 [W(x) f(x, y_t, y_t', y_t'', \dots)] dx = \int_0^1 W(x) R(x) dx \quad (2.3)$$

onde,

$W(x)$: função peso e ortogonal ao resíduo.

Como o produto escalar de vetores ortogonais é nulo tem-se:

$$\int_0^1 W(x) R(x) dx = 0 \quad (2.4)$$

e

$$\int_0^1 W(x) f(x, y_t, y_t', y_t'', \dots) dx = 0. \quad (2.5)$$

2.1.2 - Polinômios Ortogonais

Para se trabalhar com o método de colocação ortogonal é necessário escolher-se os polinômios ortogonais para a função tentativa. Nossa atenção estará voltada à série de polinômios de Lagrange:

$$P_m(x) = \sum_{j=0}^m A_j x^j, \quad 0 \leq x \leq 1. \quad (2.6)$$

Os coeficientes A_j na equação anterior são definidos pela condição de que P_1 seja ortogonal a P_0 , P_2 seja ortogonal a P_1 e P_0 e P_m seja ortogonal a cada P_k , onde $k \leq m-1$. A condição de ortogonalidade pode incluir uma função peso $W(x) \geq 0$. Temos então:

$$\int_a^b W(x) P_k(x) P_m(x) dx = 0, \quad (2.7)$$

onde:

$$k = 0, 1, 2, \dots, m-1.$$

Se tomarmos $W(x)=1$, $a=0$ e $b=1$ teremos:

$$P_0 = 1 \quad (2.8)$$

$$P_1 = 1 + bx \quad (2.9)$$

Da condição de ortogonalidade tem-se:

$$\int_0^1 P_0 P_1 dx = 0 \quad \text{ou} \quad \int_0^1 (1 + bx) dx = 0 \quad (2.10)$$

que resulta em $b=-2$. Com isto o polinômio P_1 fica definido:

$$P_1 = 1 - 2x$$

$$P_1(x) = 0 \text{ para } x = \frac{1}{2}$$
(2.11)

Os coeficientes dos demais polinômios podem ser obtidos fazendo-se as multiplicações entre eles e resolvendo-se as integrais definidas. Este procedimento gera um sistema de equações não-lineares cujas variáveis são os coeficientes dos polinômios. Outra maneira seria resolver-se um polinômio de cada vez, utilizando, na avaliação da integral definida, os coeficientes do polinômio anterior. A Tabela 2.1 mostra os valores dos coeficientes para os polinômios até o grau seis.

Tabela 2.1: Coeficientes dos polinômios de Lagrange.

Grau do Polinômio	Coeficientes					
	A_1	A_2	A_3	A_4	A_5	A_6
1	- 2					
2	- 6	6				
3	-12	30	-20			
4	-20	90	-140	70		
5	-30	210	-560	630	-252	
6	-42	420	-1680	3150	-2772	924

O polinômio $P_n(x)$ tem m raízes no intervalo a até b e estas raízes servem como escolhas convenientes para os pontos de colocação.

A Tabela 2.2 mostra os pontos de colocação obtidos pelas raízes dos polinômios cujos coeficientes são dados na Tabela 2.1.

Tabela 2.2: Pontos de colocação (raízes dos polinômios de Lagrange).

Grau do Polinômio	Pontos de Colocação							
	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
1	0.00000	0.50000	1.00000					
2	0.00000	0.21132	0.78868	1.00000				
3	0.00000	0.11270	0.50000	0.88730	1.00000			
4	0.00000	0.06943	0.33001	0.66999	0.93057	1.00000		
5	0.00000	0.04691	0.23077	0.50000	0.76923	0.95309	1.00000	
6	0.00000	0.03376	0.16939	0.38069	0.61931	0.83060	0.96623	1.00000

2.2 - Histórico do Método de Colocação Ortogonal

O primeiro artigo publicado sobre o método de colocação ortogonal foi escrito por Villadsen e Stewart (1967). Neste trabalho os autores aplicaram o método para resolver o problema de escoamento laminar de um fluido Newtoniano de densidade e viscosidade constantes através de um duto de seção quadrada.

Posteriormente, Villadsen e Sorensen (1969) aplicaram o método de colocação ortogonal para a solução de equações diferenciais parciais parabólicas, mostrando que o método é muito econômico e estável quando comparado aos métodos explícitos tais como o método de Runge-Kutta de quarta ordem.

Aplicações voltadas a reatores de leito recheado foram relatadas por Finlayson (1971). Neste trabalho, as equações de balanço de massa e de energia foram resolvidas, e o método demonstrou ser mais rápido e mais preciso do que o método das diferenças finitas.

A predição da performance de catalisadores de formato cilíndrico em reações com cinética não-linear foi o objeto de estudo de Sorensen et al.(1973). Foi considerada uma partícula de comprimento L e diâmetro D na qual ocorre uma reação química de cinética de primeira ordem. A temperatura e as concentrações foram consideradas constantes na superfície externa da partícula. O transporte interno e a reação foram estudados resolvendo-se as equações adimensionais da continuidade e energia.

Finlayson (1974) fez um estudo sobre a aplicação do método de colocação ortogonal em reatores químicos. Foram simulados três tipos de reatores: reator de oxidação de dióxido de enxofre, reator de produção de amônia com resfriamento contracorrente e a reação de redução de óxido nítrico em escapamento de automóveis. O autor concluiu que o método provou ser uma ferramenta muito eficiente para a engenharia das reações químicas.

Carey e Finlayson (1975) abordaram o assunto relativo à determinação do fator de efetividade em catalisadores, utilizando, porém, o método de colocação ortogonal em elementos finitos, para resolver a equação diferencial ordinária que governa a difusão e reação das espécies químicas numa partícula isotérmica. Como conclusão deste trabalho os autores verificaram que a eficiência aumenta à medida em que se aumenta o número de pontos de colocação internos em cada elemento. Concluíram também que para um problema de fator de efetividade correspondente a um módulo de Thiele pequeno, o método de colocação ortogonal em elementos finitos é mais eficiente do que o método das diferenças finitas e menos eficiente do que o método de colocação ortogonal que utiliza um polinômio no domínio inteiro. Para outro tipo de problema de fator de efetividade, com uma solução do tipo camada limite, o método de colocação ortogonal global não é satisfatório.

O método de colocação ortogonal em elementos finitos foi aplicado pela primeira vez a problemas bidimensionais por Prenter e Russel (1976). Neste trabalho o método é aplicado para a solução de uma equação diferencial parcial elíptica num domínio quadrado de

lado unitário.

Uma avaliação de métodos numéricos para a solução de equações diferenciais parciais elípticas lineares foi realizada por Houstis et al.(1978). Foram avaliados quatro métodos: diferenças finitas, colocação ortogonal, Galerkin e mínimos quadrados. Foram testados dezessete problemas que variavam de simples até moderadamente complexos. A principal conclusão foi a de que o método de colocação é o mais eficiente para uso geral.

Stevens et al.(1987) sugeriram que o método de colocação ortogonal seria uma técnica rápida e eficiente para o cálculo do fator de efetividade para a taxa de reação química ou bioquímica em cascas esféricas porosas baseada na cinética de Michaelis-Menten. Stevens (1988) aplicou o mesmo método para determinar o fator de efetividade para formas mais genéricas da cinética de Michaelis-Menten que incorpora os efeitos de inibição de substrato e de produto. Wang et al.(1991) mostraram que as soluções obtidas nos trabalhos de Stevens et al.(1987) e Stevens (1988), utilizando o método de colocação ortogonal, poderiam ser imprecisas e até errôneas no caso em que a inibição do substrato é considerada. Neste caso, a equação diferencial que governa o fenômeno físico é não-linear e ordinária e os autores concluíram que o método de colocação ortogonal não seria adequado a este tipo de problema físico.

Kurtanjek (1991) apresentou um método novo e simples para a otimização multidimensional de fermentadores em batelada baseado no uso da técnica de colocação ortogonal, abordando a determinação de fatores ótimos para o processo, tais como temperatura, concentração de substrato na alimentação e duração do processo.

A aplicação do método de colocação ortogonal em elementos finitos na simulação de cromatografia não-linear foi estudada por Ma e Guiochon (1991). Neste trabalho o método foi aplicado no cálculo dos perfis de banda de eluição para um ou dois componentes. Os resultados obtidos foram bons mas requeriam um tempo de computação alto.

CAPÍTULO 3

Modelagem Matemática

Neste capítulo apresentamos as equações que serão resolvidas para que se possa simular o escoamento laminar de um fluido em estado estacionário no interior de um duto cilíndrico. Estas equações são obtidas a partir de balanços de massa (equação da continuidade) e balanço de momentum (equação de Navier-Stokes) num elemento de volume.

O problema físico é apresentado em detalhe e discute-se as simplificações feitas na modelagem.

3.1 - A Equação da Continuidade

A equação da continuidade é derivada fazendo-se um balanço de massa num elemento de volume através do qual um fluido escoa, conforme a Figura 3.1:

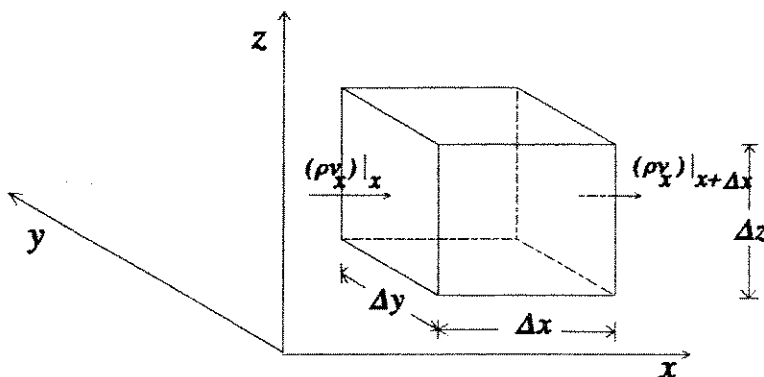


Figura 3.1: Volume de controle através do qual escoa um fluido.

$$\left\{ \begin{array}{l} \text{taxa de} \\ \text{acumulação} \\ \text{de massa} \end{array} \right\} = \left\{ \begin{array}{l} \text{taxa de} \\ \text{massa que} \\ \text{entra} \end{array} \right\} - \left\{ \begin{array}{l} \text{taxa de} \\ \text{massa que} \\ \text{sai} \end{array} \right\} \quad (3.1)$$

A taxa de massa que entra na face perpendicular ao eixo x é $(\rho v_x)|_x \Delta y \Delta z$ e a taxa de massa que sai é $(\rho v_x)|_{x+\Delta x} \Delta y \Delta z$. Para os outros pares de faces as expressões são análogas. A taxa de acumulação de massa no interior do elemento é $(\Delta x \Delta y \Delta z)(\partial \rho / \partial t)$. Substituindo-se estas expressões na equação (3.1) e dividindo-se a equação inteira por $(\Delta x \Delta y \Delta z)$, tomando-se o limite quando estas dimensões se aproximem de zero, temos:

$$\frac{\partial \rho}{\partial t} = - \left(\frac{\partial}{\partial x} \rho v_x + \frac{\partial}{\partial y} \rho v_y + \frac{\partial}{\partial z} \rho v_z \right) \quad (3.2)$$

A equação (3.2) é a equação da continuidade, que pode ser escrita da seguinte maneira:

$$\frac{\partial \rho}{\partial t} = -(\nabla \cdot \rho \mathbf{v}), \quad (3.3)$$

onde o termo $(\nabla \cdot \rho \mathbf{v})$ é chamado de divergente do vetor fluxo de massa $(\rho \mathbf{v})$. A equação (3.3) pode ser reescrita colocando-se todas as derivadas de ρ no membro esquerdo:

$$\frac{\partial \rho}{\partial t} + v_x \frac{\partial \rho}{\partial x} + v_y \frac{\partial \rho}{\partial y} + v_z \frac{\partial \rho}{\partial z} = - \rho \left(\frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y} + \frac{\partial v_z}{\partial z} \right). \quad (3.4)$$

O membro esquerdo da equação (3.4) representa a derivada substantiva da densidade. Portanto, temos:

$$\frac{D\rho}{Dt} = -\rho(\nabla \cdot \mathbf{v}) \quad (3.5)$$

Para fluidos incompressíveis esta equação reduz-se a:

$$(\nabla \cdot \mathbf{v}) = 0 \quad (3.6)$$

Como, na prática, muitas vezes trabalha-se com escoamento de fluidos em tubos é conveniente escrever a equação (3.6) em coordenadas cilíndricas:

$$\frac{\partial u}{\partial z} + \frac{\partial v}{\partial r} + \frac{v}{r} = 0 \quad (3.7)$$

3.2 - A Equação do Movimento

Para um elemento de volume ($\Delta x \Delta y \Delta z$), como o mostrado na Figura 3.2, podemos escrever um balanço de momentum da seguinte forma:

$$\left\{ \begin{array}{l} \text{taxa de} \\ \text{acumulação} \\ \text{de momentum} \end{array} \right\} = \left\{ \begin{array}{l} \text{taxa de} \\ \text{momentum} \\ \text{que entra} \end{array} \right\} - \left\{ \begin{array}{l} \text{taxa de} \\ \text{momentum} \\ \text{que sai} \end{array} \right\} + \left\{ \begin{array}{l} \text{soma das forças} \\ \text{que atuam} \\ \text{no sistema} \end{array} \right\} \quad (3.8)$$

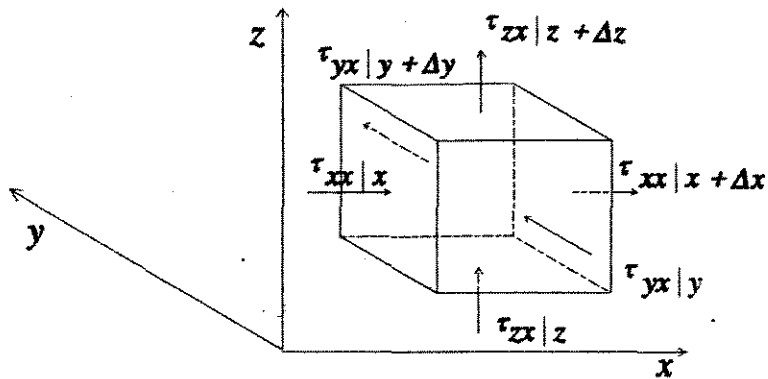


Figura 3.2: Elemento de volume onde as setas indicam a direção na qual a componente axial de momentum é transportada através das superfícies.

Na nossa análise, consideraremos apenas a componente x de cada termo na equação (3.8); as componentes y e z resultam de um procedimento análogo.

Primeiramente vamos considerar as taxas de fluxo da componente x de momentum que entra e sai do volume de controle mostrado na Figura 3.2. A entrada e saída de momentum no elemento de volume ocorre por dois mecanismos: por convecção (devido ao escoamento "bulk") e por transferência molecular (devido aos gradientes de velocidade).

A taxa com que a componente x de momentum entra por convecção na face em x é $\rho v_x v_x|_x \Delta y \Delta z$, e a taxa com que ele deixa a face $x+\Delta x$ é $\rho v_x v_x|_{x+\Delta x} \Delta y \Delta z$. A taxa com que a componente x de momentum entra na face y é $\rho v_y v_x|_y \Delta x \Delta z$. Para as outras três faces as expressões são similares.

O fluxo convectivo da componente x de momentum deve ser considerado em todas as seis faces do elemento de volume. O fluxo líquido desta componente no elemento de volume é:

$$\begin{aligned} \Delta y \Delta z (\rho v_x v_x|_x - \rho v_x v_x|_{x+\Delta x}) + \Delta x \Delta z (\rho v_y v_x|_y - \rho v_y v_x|_{y+\Delta y}) \\ + \Delta x \Delta y (\rho v_z v_x|_z - \rho v_z v_x|_{z+\Delta z}) \end{aligned} \quad (3.9)$$

De maneira análoga a taxa da componente x de momentum que entra na face em x por transporte molecular é $\tau_{xx}|_x \Delta y \Delta z$ e a que sai em $x+\Delta x$ é $\tau_{xx}|_{x+\Delta x} \Delta y \Delta z$. Para as outras quatro faces as expressões são similares. Somando-se as seis contribuições temos, para a componente x:

$$\begin{aligned} \Delta y \Delta z (\tau_{xx}|_x - \tau_{xx}|_{x+\Delta x}) + \Delta x \Delta z (\tau_{yx}|_y - \tau_{yx}|_{y+\Delta y}) \\ \Delta x \Delta y (\tau_{zx}|_z - \tau_{zx}|_{z+\Delta z}) \end{aligned} \quad (3.10)$$

Na maior parte dos casos as únicas forças importantes que atuam sobre o sistema são a pressão do fluido e a força gravitacional por unidade de massa (g). A resultante destas forças na direção x é:

$$\Delta y \Delta z (p|_x - p|_{x+\Delta x}) + \rho g_x \Delta x \Delta y \Delta z \quad (3.11)$$

Finalmente tem-se a taxa de acumulação da componente x de momentum no interior do elemento de volume, dada por $(\Delta x \Delta y \Delta z)(\partial \rho v_x / \partial t)$. Substituindo-se estas parcelas na equação (3.8), dividindo-se a equação inteira por $(\Delta x \Delta y \Delta z)$ e tomando-se o limite para Δx , Δy e Δz tendendo a zero, obtém-se a componente x da equação do movimento:

$$\begin{aligned} \frac{\partial}{\partial t} \rho v_x = - \left(\frac{\partial}{\partial x} \rho v_x v_x + \frac{\partial}{\partial y} \rho v_y v_x + \frac{\partial}{\partial z} \rho v_z v_x \right) \\ - \left(\frac{\partial}{\partial x} \tau_{xx} + \frac{\partial}{\partial y} \tau_{yx} + \frac{\partial}{\partial z} \tau_{zx} \right) - \frac{\partial \rho}{\partial x} + \rho g_x \end{aligned} \quad (3.12)$$

As componentes y e z são obtidas de maneira análoga e têm forma similar. A equação (3.12) pode ser rearranjada, utilizando a

equação da continuidade, resultando:

$$\rho \frac{Dv_x}{Dt} = - \frac{\partial p}{\partial x} - \left(\frac{\partial}{\partial x} \tau_{xx} + \frac{\partial}{\partial y} \tau_{yx} + \frac{\partial}{\partial z} \tau_{zx} \right) + \rho g_x \quad . \quad (3.13)$$

A equação (3.13), quando generalizada para qualquer das três componentes espaciais e para fluidos Newtonianos, pode ser escrita em termos de gradientes de velocidade e propriedades do fluido, originando, para o caso de um fluido com ρ e μ constantes:

$$\rho \frac{Dv}{Dt} = - \nabla p + \mu \nabla^2 v + \rho g \quad . \quad (3.14)$$

Esta equação é chamada de equação de Navier-Stokes, pois foi desenvolvida por Navier, na França, em 1822.

Quando escrita em coordenadas cilíndricas, para um escoamento bidimensional em estado estacionário de um fluido com propriedades físicas constantes, a componente axial da equação de Navier-Stokes apresenta a seguinte forma:

$$\rho \left(v \frac{\partial u}{\partial r} + u \frac{\partial u}{\partial z} \right) = - \frac{\partial p}{\partial z} + \mu \left[\frac{1}{r} \frac{\partial}{\partial r} \left(r \frac{\partial u}{\partial r} \right) + \frac{\partial^2 u}{\partial z^2} \right] \quad . \quad (3.15)$$

A equação (3.15) considera desprezíveis os efeitos do campo gravitacional.

3.3 - Descrição do Problema Físico

As equações (3.7) e (3.15) serão aplicadas à simulação de um escoamento bidimensional em estado estacionário, considerando-se um gradiente de pressão constante ao longo do tubo e para números de Reynolds 500, 700 e 1000. A simulação será feita considerando-se um tubo de dimensões $L=200R_t$, onde L é o comprimento e R_t é o raio do tubo (ver Figura 3.3).

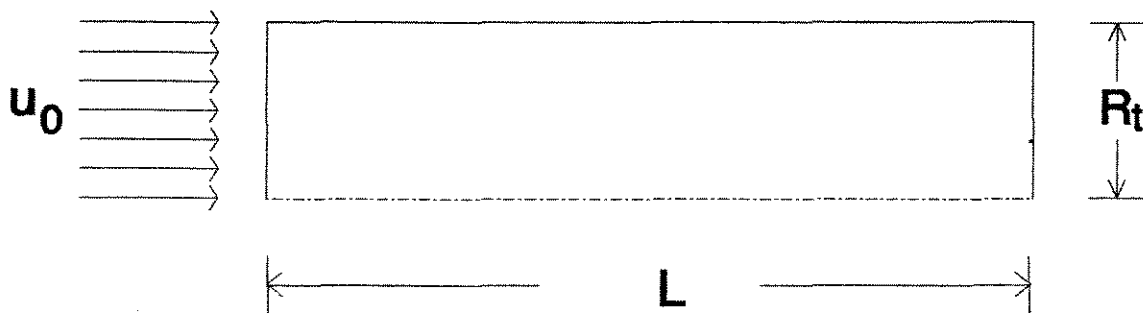


Figura 3.3: Esquema do problema considerado

Como o problema é axissimétrico, será considerado apenas a metade do domínio.

Na entrada o fluido apresenta um perfil uniforme de velocidade axial, com a componente axial calculada em função do número de Reynolds, utilizando a equação:

$$Re = \frac{2R_t u_0}{\nu} ,$$

(3.16)

onde:

- Re: número de Reynolds do escoamento;
- R_t : raio do tubo;
- u_o : velocidade axial na entrada;
- ν : viscosidade cinemática do fluido.

As condições de contorno para o problema são as seguintes:

eixo de simetria:

$$\frac{\partial u}{\partial r} = 0 \quad e \quad v = 0 \quad ; \quad (3.17)$$

parede do tubo:

$$u = 0 \quad e \quad v = 0 \quad ; \quad (3.18)$$

entrada do tubo:

$$u = u_o \text{ (função do número de Re)} \quad e \quad v = 0 \quad ; \quad (3.19)$$

saída do tubo:

$$\frac{\partial u}{\partial z} = 0 \quad e \quad v = 0 \quad . \quad (3.20)$$

Considera-se, portanto, que na saída o escoamento apresenta-se plenamente desenvolvido. O comprimento para o estabelecimento do perfil desenvolvido (L_d) é dado pela seguinte expressão:

$$\frac{L_d}{2R_t} \approx 0,06 Re \quad (3.21)$$

A equação para perfil desenvolvido de velocidade axial, para escoamento laminar, escrita em termos do gradiente de pressão axial é:

$$u = \frac{(P_o - P_L)}{4\mu L} R_t^2 \left[1 - \left(\frac{r}{R_t} \right)^2 \right] , \quad (3.22)$$

onde:

- P_o : pressão do fluido na entrada do tubo;
- P_L : pressão do fluido na saída do tubo;
- μ : viscosidade dinâmica do fluido;
- L : comprimento do tubo;
- R_t : raio do tubo.

A velocidade axial máxima, para o escoamento desenvolvido, ocorre no centro do tubo e é dada pela equação:

$$u_{\max} = \frac{(P_o - P_L)}{4\mu L} R_t^2 . \quad (3.23)$$

Para o escoamento laminar desenvolvido a velocidade axial máxima é igual ao dobro da velocidade axial média:

$$u_{\max} = 2\bar{u} . \quad (3.24)$$

Combinando as equações (3.22), (3.23) e (3.24) tem-se:

$$u = 2\bar{u} \left[1 - \left(\frac{r}{R_t} \right)^2 \right]. \quad (3.25)$$

O gradiente axial de pressão foi, inicialmente, considerado constante para efeitos de simulação, sendo calculado como função do número de Reynolds, combinando-se as equações (3.16), (3.22) e (3.25), obtendo-se:

$$-\frac{\Delta P}{L} = \frac{4\rho v^2 Re}{R_t^3}. \quad (3.26)$$

Posteriormente, com o intuito de melhorar a qualidade dos resultados, utilizou-se dados de gradiente de pressão obtidos através de simulações utilizando-se o método de volumes finitos, em virtude de não se dispor de dados experimentais. Estes dados estavam disponíveis para números de Reynolds 100, 300 e 500.

CAPÍTULO 4

Métodos Numéricos para a Simulação

Neste capítulo será apresentado o princípio matemático do método de colocação ortogonal utilizado na discretização das equações diferenciais do problema proposto.

O sistema, após a discretização, gerará um conjunto de equações algébricas não-lineares, que será resolvido utilizando-se uma subrotina do método de Newton adequada a sistemas esparsos de grande porte.

4.1 - Método de colocação Ortogonal

No método de colocação ortogonal os pontos de colocação são selecionados automaticamente, evitando com isto uma escolha arbitrária (e talvez não adequada) pelo usuário. Este método caracteriza-se por:

- a função tentativa é tomada como uma série de polinômios ortogonais;
- os pontos de colocação são as raízes de um destes polinômios;
- as variáveis dependentes são as soluções obtidas nos pontos de colocação ao invés de serem os coeficientes na expansão.

4.1.1 - Fundamentação do Método

Uma função qualquer $y(x)$ pode ser aproximada por um polinômio, conforme a equação:

$$y = \sum_{i=1}^{N+2} d_i x^{i-1} \quad (4.1)$$

Para a solução de uma equação diferencial, toma-se as derivadas primeira e segunda (conforme o caso) da equação (4.1), avaliando-as nos pontos de colocação. Toma-se os pontos de colocação como as N raízes de $P_N(x)=0$; estas raízes estão compreendidas entre zero e um. Os pontos de colocação são, então, $x_1=0, x_2, \dots, x_{N+1}$ são as raízes interiores e $x_{N+2}=1$ como mostrado na Figura 4.1:

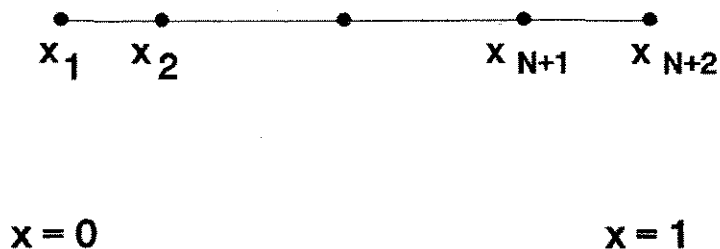


Figura 4.1: Distribuição geométrica dos pontos de colocação para $N=3$.

A função e as derivadas nos $N+2$ pontos de colocação são:

$$y(x_j) = \sum_{i=1}^{N+2} d_i x_j^{i-1} \quad (4.2)$$

$$\frac{dy}{dx}(x_j) = \sum_{i=1}^{N+2} d_i (i-1) x_j^{i-2} , \quad (4.3)$$

$$\frac{d^2y}{dx^2}(x_j) = \sum_{i=1}^{N+2} d_i (i-1) (i-2) x_j^{i-3} . \quad (4.4)$$

Estas equações podem ser escritas em notação matricial, onde Q, C e D são matrizes (N+2)x(N+2):

$$y = Q d , \quad (4.5)$$

$$\frac{dy}{dx} = C d , \quad (4.6)$$

$$\frac{d^2y}{dx^2} = D d . \quad (4.7)$$

Resolvendo-se a equação (4.5) para d pode-se reescrever as derivadas primeira e segunda como:

$$\frac{dy}{dx} = C Q^{-1} y \equiv A y , \quad (4.8)$$

$$\frac{d^2y}{dx^2} = D Q^{-1} y \equiv B y . \quad (4.9)$$

Conclui-se, portanto, que as derivadas em cada um dos pontos de colocação podem ser expressas em termos do valor da função nestes pontos.

As matrizes A e B, para os polinômios de grau um e dois, são mostradas no apêndice A.

4.1.2 - Colocação Ortogonal em Elementos Finitos

O método de colocação ortogonal tal como foi exposto até aqui utiliza uma função tentativa definida sobre o domínio inteiro. Quando a solução tem gradientes acentuados é mais vantajoso usar funções tentativa definidas em subdomínios, o que resulta em aproximações melhores. Isto levou ao método de colocação ortogonal em elementos finitos. As funções tentativa podem ser de dois tipos: um deles utiliza funções Lagrangeanas e adiciona condições que fazem com que as derivadas primeiras sejam contínuas entre os elementos; a outra forma utiliza polinômios de Hermite em que a continuidade da derivada primeira é assegurada automaticamente. Neste trabalho vamos nos ater aos polinômios de Lagrange.

Para a aplicação deste método o domínio é dividido em elementos como mostra a Figura 4.2:

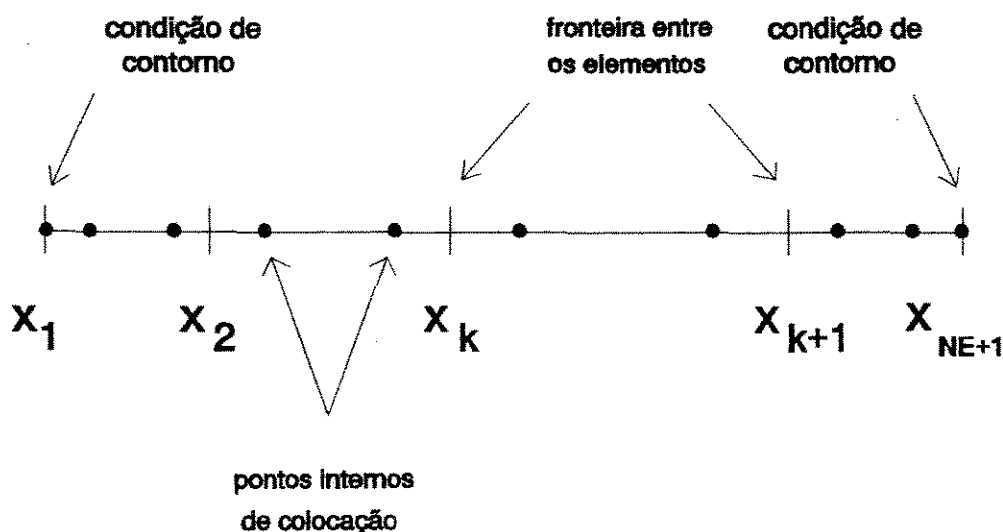


Figura 4.2: Domínio para o método de colocação ortogonal em elementos finitos utilizando polinômios cúbicos de Lagrange.

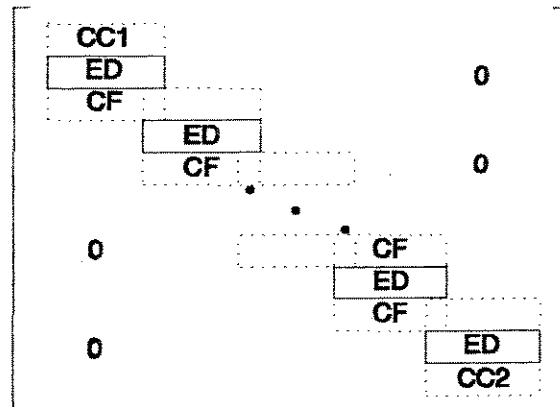
As equações são escritas para o késimo elemento, onde define-se:

$$U = \frac{x - x_k}{h_k} \quad (4.10)$$

$$h_k = x_{k+1} - x_k \quad (4.11)$$

Desta maneira a variável U varia de zero a um em cada elemento. Aplica-se o método de colocação ortogonal em cada um dos elementos, de maneira análoga ao caso de colocação global.

Para uma equação diferencial ordinária a estrutura da matriz resultante da discretização utilizando o método de colocação ortogonal em elementos finitos com polinômios de Lagrange é mostrada na Figura 4.3:



- ED - Eq. dif. discretizada nos pontos internos de colocação
- CC1 - Condição de contorno no extremo (x=0)
- CC2 - Condição de contorno no extremo (x=1)
- CF - Condição de fronteira

Figura 4.3: Sistema discreto oriundo da aplicação do método de colocação ortogonal em elementos finitos.

O sistema de equações algébricas resultante da discretização pode ser resolvido pelo método de Gauss ou pelo método de Newton-Raphson conforme as equações sejam, respectivamente, lineares ou não-lineares.

4.2 - Aplicação do Método de Colocação Ortogonal em Elementos Finitos ao Problema de Escoamento Laminar

Conforme o exposto no capítulo anterior o problema consiste em resolver as equações da continuidade e a componente axial da equação de Navier-Stokes, ambas escritas em coordenadas cilíndricas, considerando-se um domínio bidimensional e estado estacionário (equações 3.7 e 3.15).

Inicialmente define-se o tamanho de cada elemento finito nas direções radial (r) e axial (z) como:

$$HR_k = r_{k+1} - r_k \quad , \quad (4.12)$$

$$HZ_j = z_{j+1} - z_j \quad . \quad (4.13)$$

Para aplicar-se o método de colocação ortogonal o domínio de cada elemento deve estar compreendido entre zero e um. Portanto é necessário definir-se novas variáveis independentes:

$$R = \frac{r - r_k}{HR_k} \quad , \quad (4.14)$$

$$Z = \frac{z - z_j}{HZ_j} \quad . \quad (4.15)$$

Desta maneira, enquanto as variáveis r e z podem assumir quaisquer valores dentro do domínio do problema considerado, as variáveis R e Z variam sempre, em qualquer dos elementos considerados, entre os valores zero e um.

Subdividindo-se o domínio em NTR elementos na direção radial e NTZ elementos na direção axial e considerando-se que os elementos radiais tenham todos o mesmo comprimento, o mesmo acontecendo com os elementos na direção axial, tem-se:

$$HR_k = HRI \quad \text{para } k = 1, 2, \dots, NTR \quad , \quad (4.16)$$

$$HZ_j = HZI \quad \text{para } j = 1, 2, \dots, NTZ \quad . \quad (4.17)$$

Escrevendo-se a equação da continuidade nas novas variáveis independentes, tem-se:

$$\frac{1}{HZI} \frac{\partial u}{\partial Z} + \frac{1}{HRI} \frac{\partial v}{\partial R} + \frac{v}{R HRI + r_k} = 0 \quad . \quad (4.18)$$

Para a componente axial da equação de Navier-Stokes tem-se:

$$\rho \left(\frac{v}{HRI} \frac{\partial u}{\partial R} + \frac{u}{HZI} \frac{\partial u}{\partial Z} \right) = -\frac{\Delta P}{L} + \mu \left[\frac{1}{HRI^2} \frac{\partial^2 u}{\partial R^2} + \frac{1}{HZI^2} \frac{\partial^2 u}{\partial Z^2} + \frac{1}{R HRI + r_k} \frac{1}{HRI} \frac{\partial u}{\partial R} \right] \quad . \quad (4.19)$$

Como iremos utilizar um ponto de colocação em cada direção, por elemento, a variável R nestes pontos fica determinada pela raiz do polinômio de Lagrange P_1 , cujo valor é 0,5. Logo obtém-se:

- equação da continuidade:

$$\frac{1}{HZI} \frac{\partial u}{\partial Z} + \frac{1}{HRI} \frac{\partial v}{\partial R} + \frac{v}{0,5 HRI + r_k} = 0 \quad ; \quad (4.20)$$

- equação de Navier-Stokes (componente axial):

$$\rho \left(\frac{v}{HRI} \frac{\partial u}{\partial R} + \frac{u}{HZI} \frac{\partial u}{\partial Z} \right) = -\frac{\Delta P}{L} + \mu \left[\frac{1}{HRI^2} \frac{\partial^2 u}{\partial R^2} + \frac{1}{HZI^2} \frac{\partial^2 u}{\partial Z^2} + \frac{1}{0,5 HRI + r_k} \frac{1}{HRI} \frac{\partial u}{\partial R} \right] \quad (4.21)$$

A Figura 4.4 mostra a divisão do domínio considerado em elementos finitos, para o caso em que NTR = 2 e NTZ = 3:

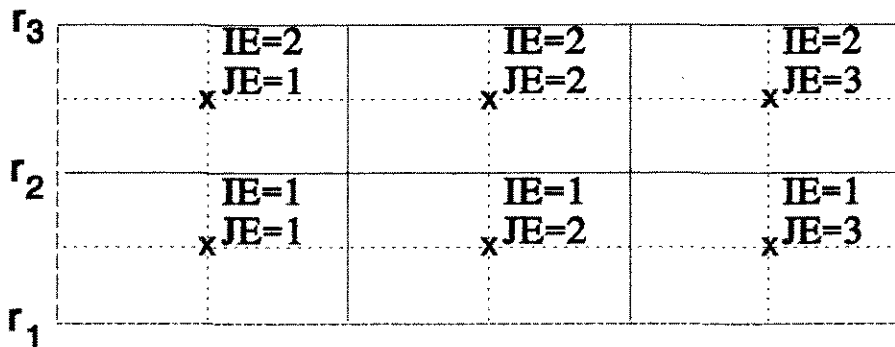


Figura 4.4: Divisão do domínio em elementos finitos.

As linhas cheias representam os limites entre os elementos. Os pontos de colocação internos são representados por x. Os elementos são numerados pela respectiva posição (IE,JE) onde:

$$\begin{aligned} IE &= 1, 2, \dots, NTR \quad , \\ JE &= 1, 2, \dots, NTZ \quad . \end{aligned} \quad (4.22)$$

A numeração das variáveis segue o esquema mostrado na Figura 4.5:

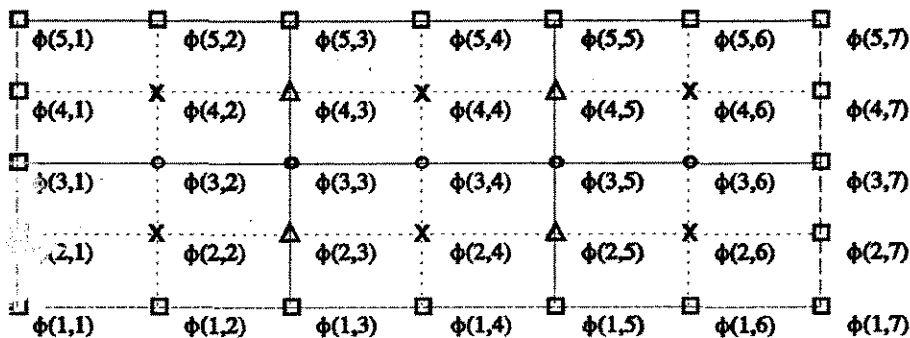


Figura 4.5: Esquema de numeração de variáveis no tubo.

onde:

$$\phi = \{u, v\}.$$

O valor de r_k na equação (4.21) fica definido de acordo com a posição do ponto de colocação interno onde a equação diferencial está sendo discretizada:

$$r_k = r_{IE} = HRI(IE - 1) \quad , \quad \text{para } IE = 1, 2, \dots, NTR \quad . \quad (4.23)$$

Por exemplo, na discretização da equação de Navier-Stokes no elemento ($IE=2, JE=2$) o valor de r_k é HRI .

Na Figura 4.5 os pontos marcados com quadrados representam a discretização das condições de contorno. Por exemplo, a condição de que $\partial u / \partial r = 0$ no eixo de simetria no ponto (1,2) nos leva a:

$$A(1,1)u(1,2) + A(1,2)u(2,2) + A(1,3)u(3,2) = 0 . \quad (4.24)$$

Assim, para um determinado ponto de posição (1,L), tem-se:

$$\sum_{M=1}^3 A(1,M)u(M,L) = 0 \quad , \quad 1 < L < (2NTZ + 1) . \quad (4.25)$$

Na saída do tubo, onde a condição de contorno para a componente axial é $\partial u / \partial z = 0$, tem-se para o ponto (2,7):

$$A(3,1)u(2,5) + A(3,2)u(2,6) + A(3,3)u(2,7) = 0 . \quad (4.26)$$

Generalizando-se para um ponto qualquer (N,(2NTZ+1)) tem-se:

$$\sum_{M=1}^3 A(3,M)u(N, (2NTZ+M-2)) = 0 \quad (4.27)$$

para $1 \leq N \leq (2NTR+1)$.

Na entrada do tubo a componente axial de velocidade é definida pelo número de Reynolds do escoamento, considerando-se um perfil uniforme:

$$u(kk,1) = f(Re) \quad (4.28)$$

para $1 \leq kk \leq (2NTR + 1)$.

Na parede do tubo a componente axial de velocidade é nula:

$$u(2NTR+1, jj) = 0 \quad (4.29)$$

para $2 \leq jj \leq 2NTZ$.

A componente radial de velocidade (v) é nula em todas as quatro faces do domínio.

Os pontos marcados com círculos e triângulos na Figura 4.5 representam, respectivamente, as fronteiras entre elementos nas direções radial e axial. Ao passar-se de um elemento (IE, JE) para outro, deve-se fixar a condição de continuidade da derivada de primeira ordem em relação à direção considerada, de forma que o perfil seja suave. Isto é feito para pontos que pertençam, simultaneamente, a mais de um elemento finito.

Em notação matemática:

- continuidade na direção radial:

$$\frac{\partial \phi}{\partial r} \Big|_{IE, JE} = \frac{\partial \phi}{\partial r} \Big|_{IE+1, JE} \quad ; \quad (4.30)$$

- continuidade na direção axial:

$$\frac{\partial \phi}{\partial z} \Big|_{IE, JE} = \frac{\partial \phi}{\partial z} \Big|_{IE, JE+1} \quad , \quad (4.31)$$

onde $\phi = \{u, v\}$.

Por exemplo, para o ponto (2,3) temos que assegurar a continuidade da derivada na direção z . Para tanto tem-se a seguinte equação:

$$\begin{aligned} A(3,1)\phi(2,1) + A(3,2)\phi(2,2) + A(3,3)\phi(2,3) = \\ A(1,1)\phi(2,3) + A(1,2)\phi(2,4) + A(1,3)\phi(2,5) \quad , \end{aligned} \quad (4.32)$$

onde $\phi = \{u, v\}$. Para outros pontos o procedimento é análogo.

A continuidade da derivada na direção radial, para o ponto (3,3) leva à seguinte equação:

$$\begin{aligned} A(3,1)\phi(1,3) + A(3,2)\phi(2,3) + A(3,3)\phi(3,3) = \\ A(1,1)\phi(3,3) + A(1,2)\phi(4,3) + A(1,3)\phi(5,3) \end{aligned} \quad (4.33)$$

onde $\phi = (u,v)$. Para os pontos restantes o procedimento é análogo.

A discretização das equações diferenciais é feita nos pontos marcados com x na Figura 4.5. A equação da continuidade discretizada no elemento (IE=1, JE=1) apresenta a seguinte forma:

$$\begin{aligned} \frac{1}{HZI} [A(2,1)u(2,1) + A(2,2)u(2,2) + A(2,3)u(2,3)] + \\ + \frac{1}{HRI} [A(2,1)v(1,2) + A(2,2)v(2,2) + A(2,3)v(3,2)] + \\ + \frac{v(2,2)}{0,5HRI + r_1} = 0 \end{aligned} \quad (4.34)$$

Expressões análogas são obtidas para os outros pontos de colocação internos.

A discretização da componente axial da equação de Navier-Stokes no mesmo ponto apresenta a seguinte forma:

$$\begin{aligned}
& \frac{v(2,2)}{HRI} [A(2,1)u(1,2) + A(2,2)u(2,2) + A(2,3)u(3,2)] + \\
& + \frac{u(2,2)}{HZI} [A(2,1)u(2,1) + A(2,2)u(2,2) + A(2,3)u(2,3)] - \\
& - \frac{v}{HRI^2} [B(2,1)u(1,2) + B(2,2)u(2,2) + B(2,3)u(3,2)] - \\
& - \frac{v}{HRI} \frac{1}{0,5HRI + I_1} [A(2,1)u(1,2) + A(2,2)u(2,2) + A(2,3)u(3,2)] \\
& - \frac{v}{HZI^2} [B(2,1)u(2,1) + B(2,2)u(2,2) + B(2,3)u(2,3)] + \frac{\Delta P}{L} = 0.
\end{aligned}
\tag{4.35}$$

Para os pontos de colocação internos restantes as expressões obtidas são análogas.

Como resultado da discretização obtém-se um sistema de equações algébricas não-lineares constituído por $[2x(2NTR+1)x(2NTZ+1)]$ equações. Para resolvê-lo foi utilizada uma subrotina do Método de Newton desenvolvida pelo IMECC (UNICAMP) para sistemas de grande porte cuja matriz jacobiana é esparsa, subrotina esta denominada ROUXINOL. Para fazer uso deste programa foi preciso construir-se outras três subrotinas, necessárias, respectivamente, para o cálculo da estrutura da matriz jacobiana (STRTUBO.FOR), cálculo do jacobiano analítico (JCBTUBO.FOR) e avaliação das funções, escritas na forma $f(x)=0$ (FNTUBO.FOR). Estas subrotinas, assim como o programa gerenciador, estão listadas em anexo.

Para facilitar a confecção do código computacional as variáveis dependentes foram numeradas de maneira a transformar matrizes em vetores. O sistema apresenta $[2x(2NTR+1)x(2NTZ+1)]$ variáveis no total, sendo metade delas referentes à componente axial e a outra metade referentes à componente radial de

velocidade. A variável $va(1)$, com a nova numeração, assume a variável $u(1,1)$, a variável $va(2)$ assume a variável $u(1,2)$ e assim por diante, até chegar-se à variável $va((2NTZ+1) \times (2NTR+1))$ que assume a variável $u(2NTZ+1, 2NTR+1)$. Com relação à componente radial de velocidade (v) o procedimento é análogo, partindo-se de $va((2NTZ+1) \times (2NTR+1)+1)$ assumindo a variável $v(1,1)$ até chegar-se à variável $va(2 \times (2NTZ+1) \times (2NTR+1))$ que assumirá a variável $v(2NTZ+1, 2NTR+1)$.

O programa foi construído de maneira a utilizar como dados de entrada o número de elementos nas direções radial e axial (respectivamente NTR e NTZ), o tamanho de cada elemento nas direções radial e axial (respectivamente HRI e HZI), a viscosidade cinemática e a densidade do fluido e o número de Reynolds do escoamento.

CAPÍTULO 5

Resultados e Análise

Conforme o que foi visto no capítulo 3, as simulações foram divididas em dois tipos: simulações a gradiente de pressão constante e simulações a gradiente de pressão variável.

5.1 - Simulação a Gradiente de Pressão Constante

Para a validação do código proposto foram feitas simulações num tubo com as proporções $L=200R_t$ (L =comprimento; R_t =raio). Utilizou-se malhas 5×10 , 10×10 e 10×15 elementos (direção axial x direção radial) e números de Reynolds 500, 700 e 1000. O fluido utilizado foi o ar a 25°C .

Os resultados obtidos foram colocados em forma gráfica e apresentam-se a seguir. Cada gráfico mostra as curvas de variação da componente de velocidade axial com a variável raio adimensionalizada, para diferentes posições axiais ao longo do tubo. Apresenta-se, também, o perfil plenamente desenvolvido, obtido pela solução analítica.

A estimativa inicial utilizada pelo método de Newton para a solução dos sistemas de equações algébricas não-lineares foi feita da seguinte maneira: para as variáveis correspondentes à componente axial o valor utilizado foi o valor de entrada, definido pelo número de Reynolds; para as variáveis referentes à componente radial de velocidade a estimativa inicial utilizada foi o valor zero. Com isto verificou-se uma convergência relativamente rápida, principalmente para o caso da malha 5×10 . Com relação à malha 10×10

a convergência foi obtida com 5 iterações para $Re=500$ e 7 iterações para $Re=700$ e $Re=1000$. O número máximo de iterações exigido foi para o caso da malha 10×15 e $Reynolds=700$, onde foram necessárias 11 iterações. O programa utiliza como critério de parada a condição de que a diferença entre duas aproximações consecutivas seja menor que uma determinada tolerância estabelecida pelo usuário.

De acordo com os resultados obtidos, verificou-se que, para uma determinada malha, o resultado do perfil estabelecido é tanto melhor quanto menor for o número de Reynolds do escoamento, conforme observa-se na Tabela 5.1:

Tabela 5.1: Resultados obtidos e CPU utilizada (gradiente de pressão constante).

malha/Reynolds	$(u_{max}/\bar{u})_{simul.}$	$(u_{max}/\bar{u})_{analit.}$	CPU(s)	Desvio(%)
5x10 / 500	1,942	2,000	36	- 2,90
5x10 / 700	1,905	2,000	48	- 4,75
5x10 /1000	1,860	2,000	72	- 7,00
10x10 / 500	1,947	2,000	300	- 2,65
10x10 / 700	1,912	2,000	425	- 4,40
10x10 /1000	1,868	2,000	425	- 6,60
10x15 / 500	1,958	2,000	1006	- 2,10
10x15 / 700	1,921	2,000	1390	- 3,95
10x15 /1000	1,871	2,000	1134	- 6,45

Com o refinamento da malha, observou-se que o resultado não apresenta grandes alterações, ou seja, mesmo com um reduzido número de elementos o resultado já é razoável.

As simulações foram feitas utilizando uma estação de trabalho Sun SPARCstation 1+.

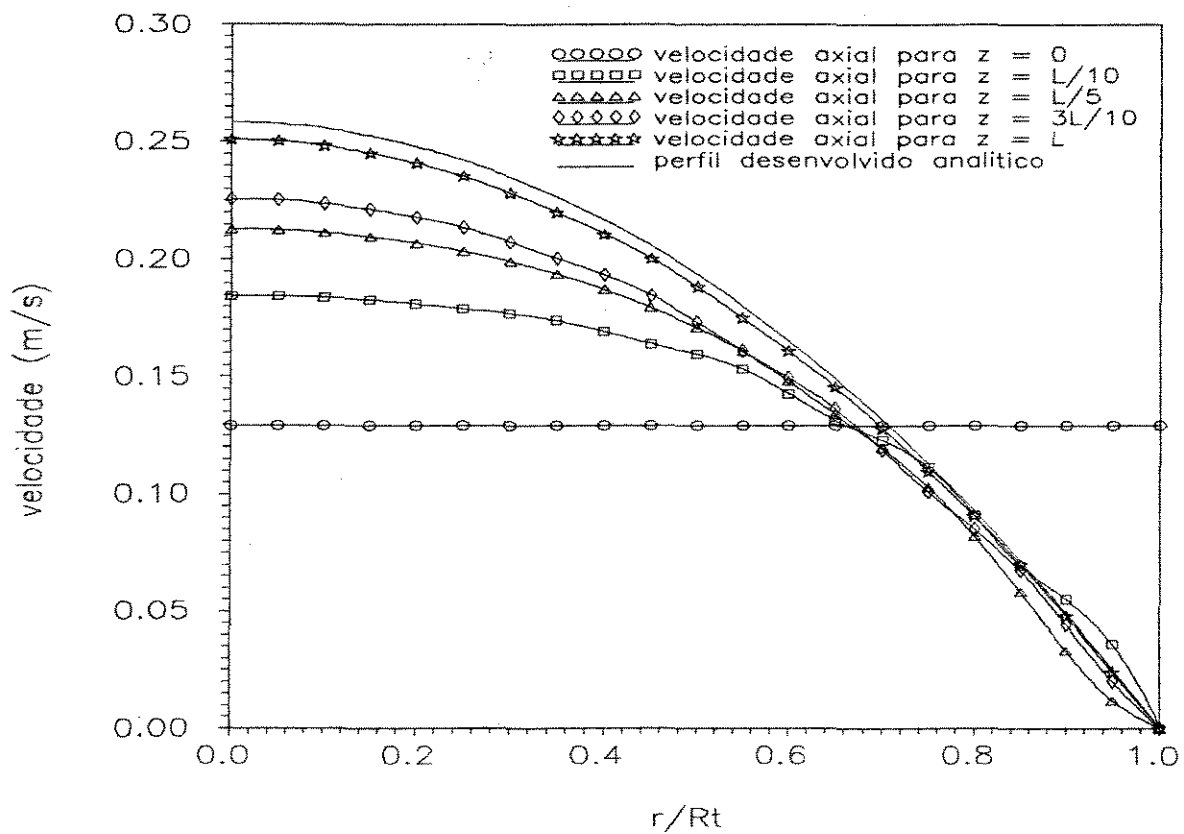


Figura 5.1: Simulação para Reynolds = 500 e malha 5x10 elementos.

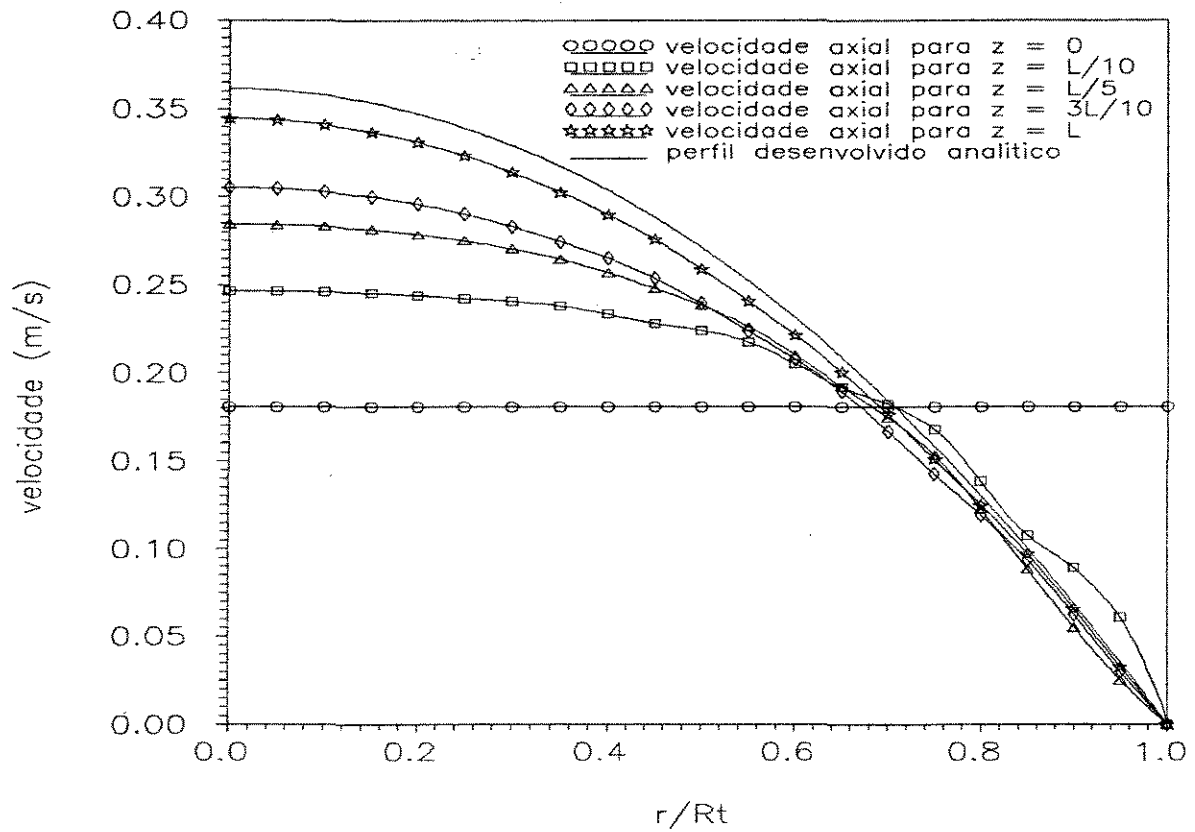


Figura 5.2: Simulação para Reynolds = 700 e malha 5x10 elementos.

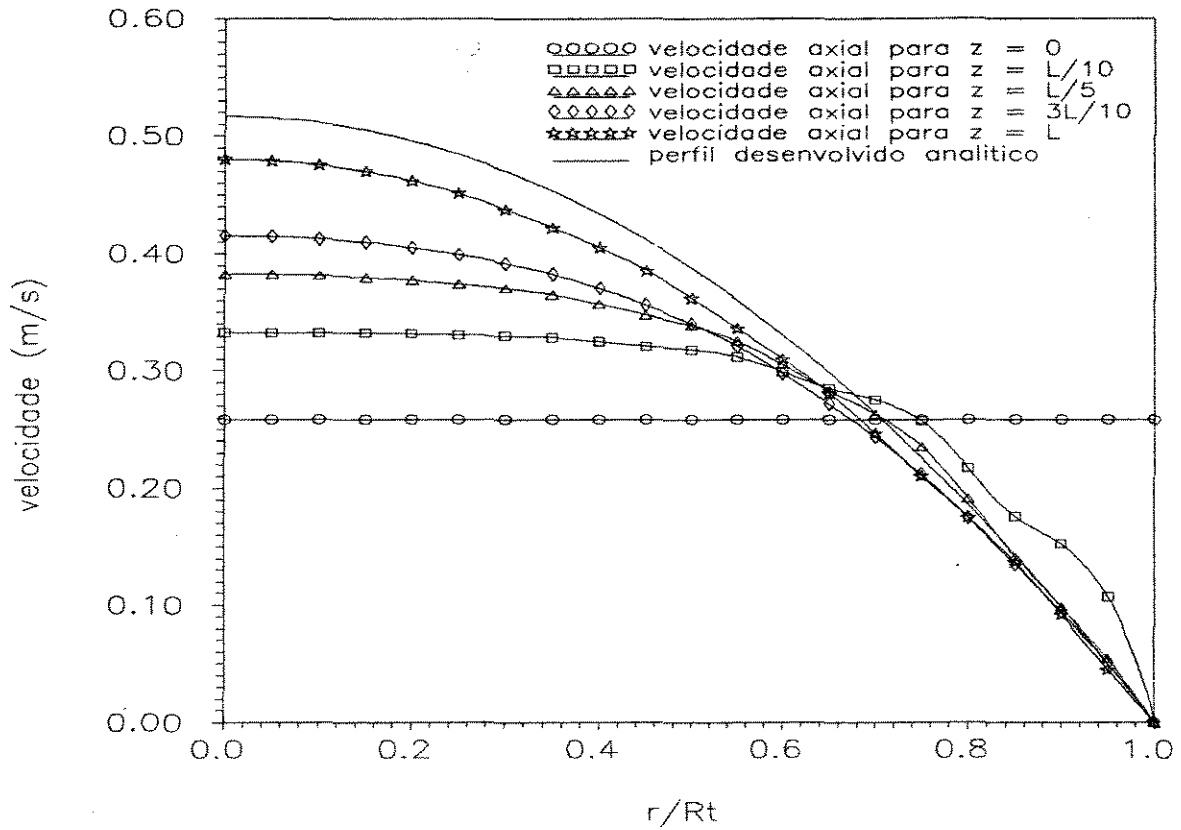


Figura 5.3: Simulação para Reynolds = 1000 e malha 5x10 elementos.

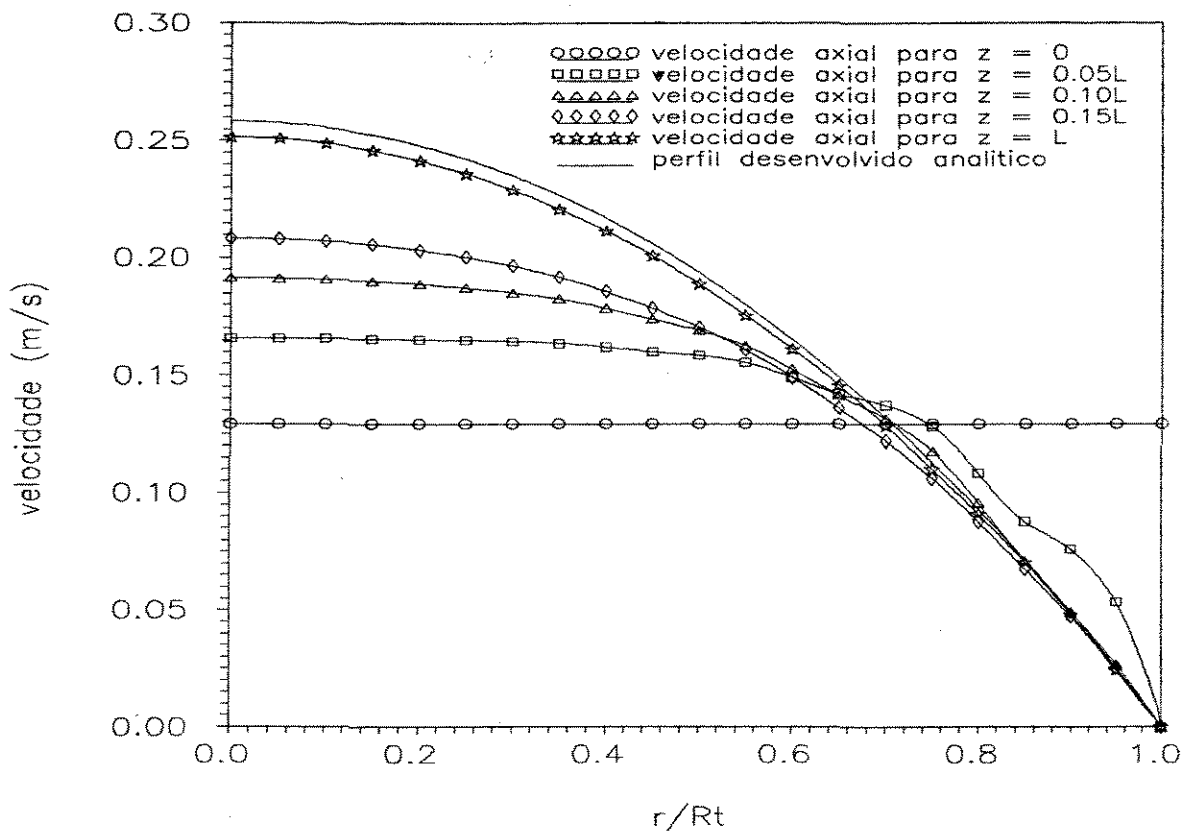


Figura 5.4: Simulação para Reynolds = 500 e malha 10x10 elementos.

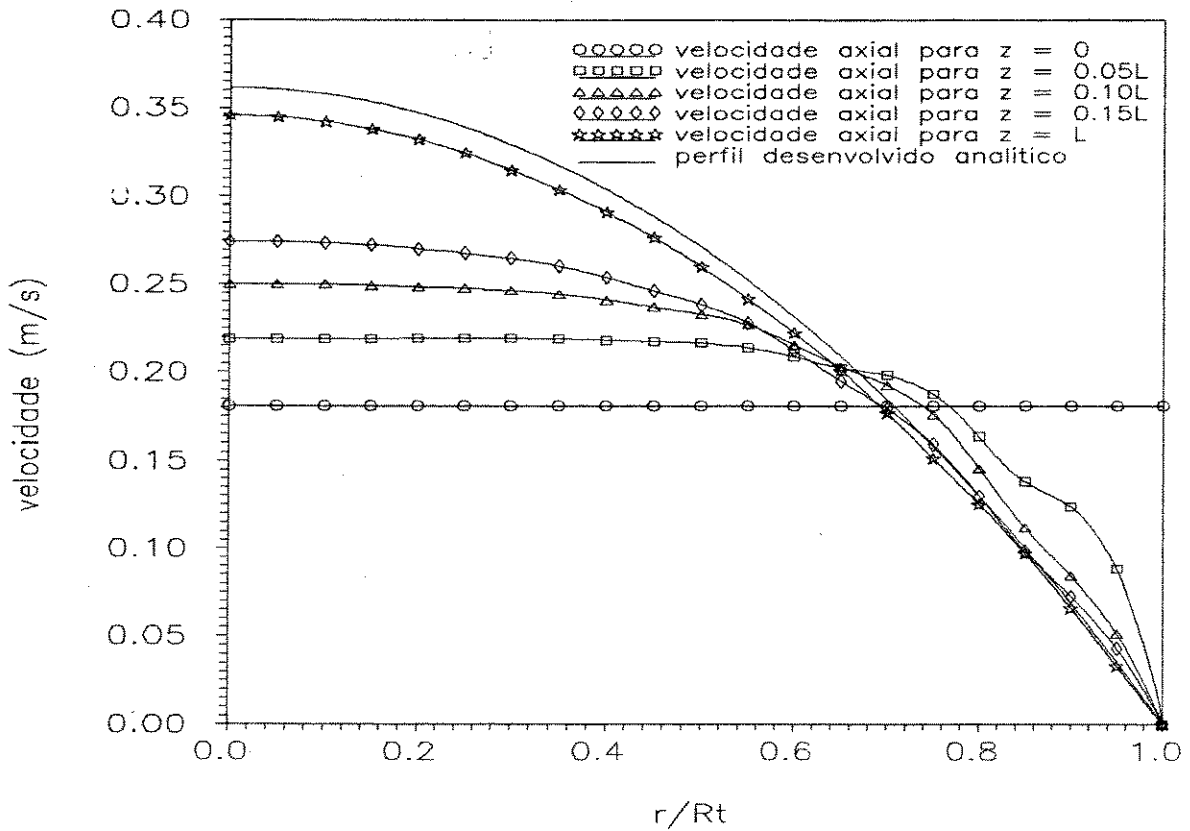


Figura 5.5: Simulação para Reynolds = 700 e malha 10x10 elementos.

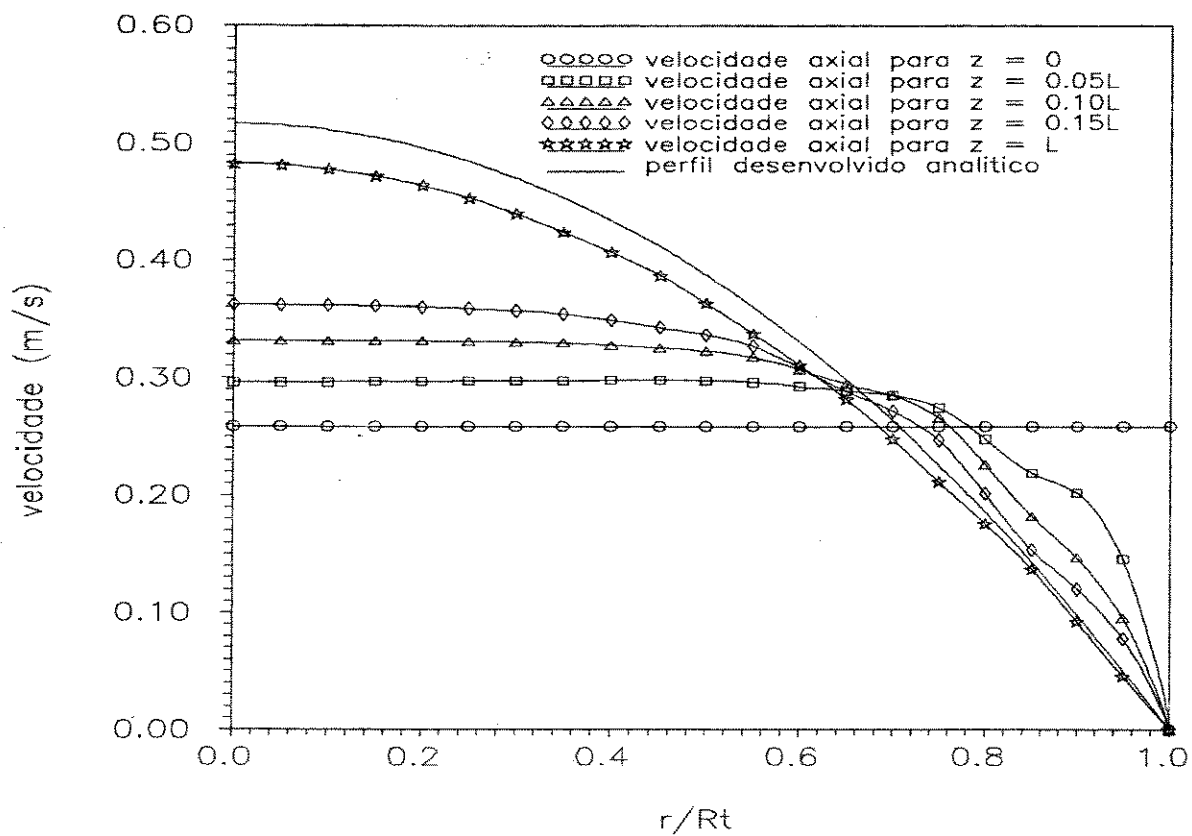


Figura 5.6: Simulação para Reynolds = 1000 e malha 10x10 elementos.

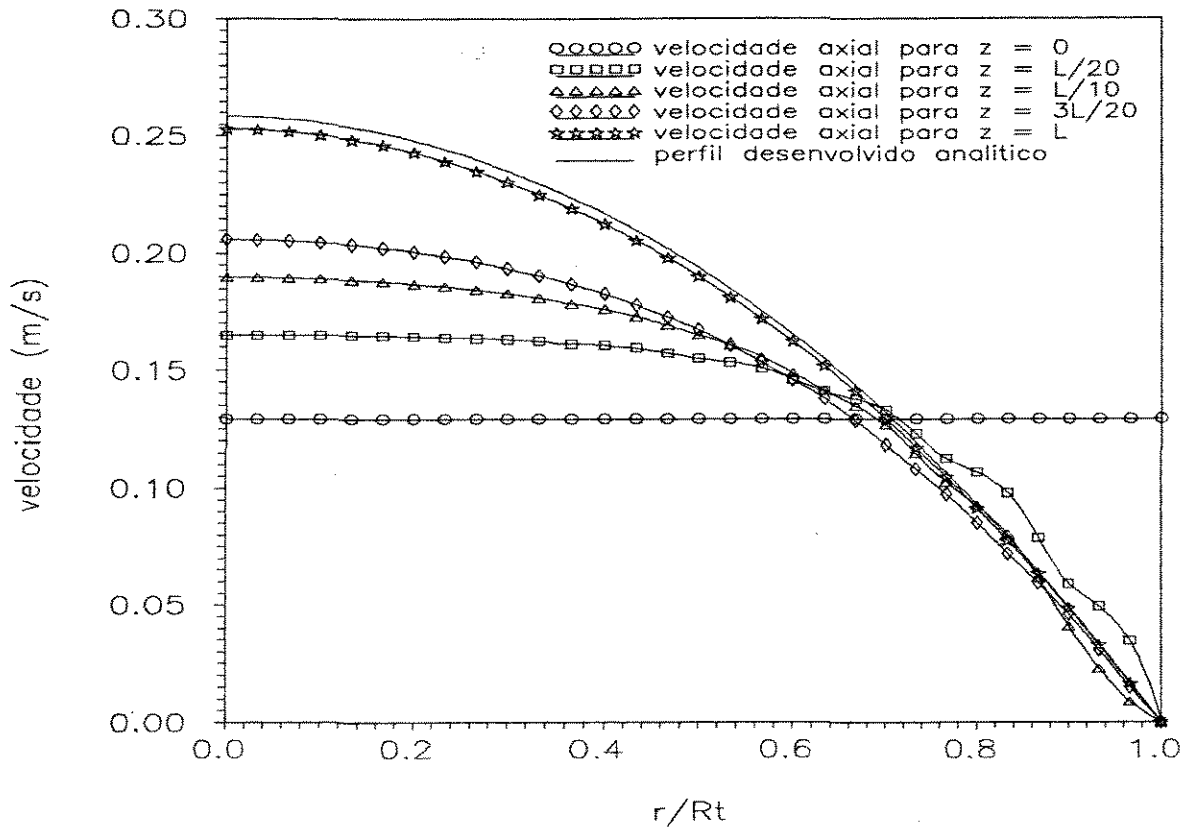


Figura 5.7: Simulação para Reynolds = 500 e malha 10x15 elementos.

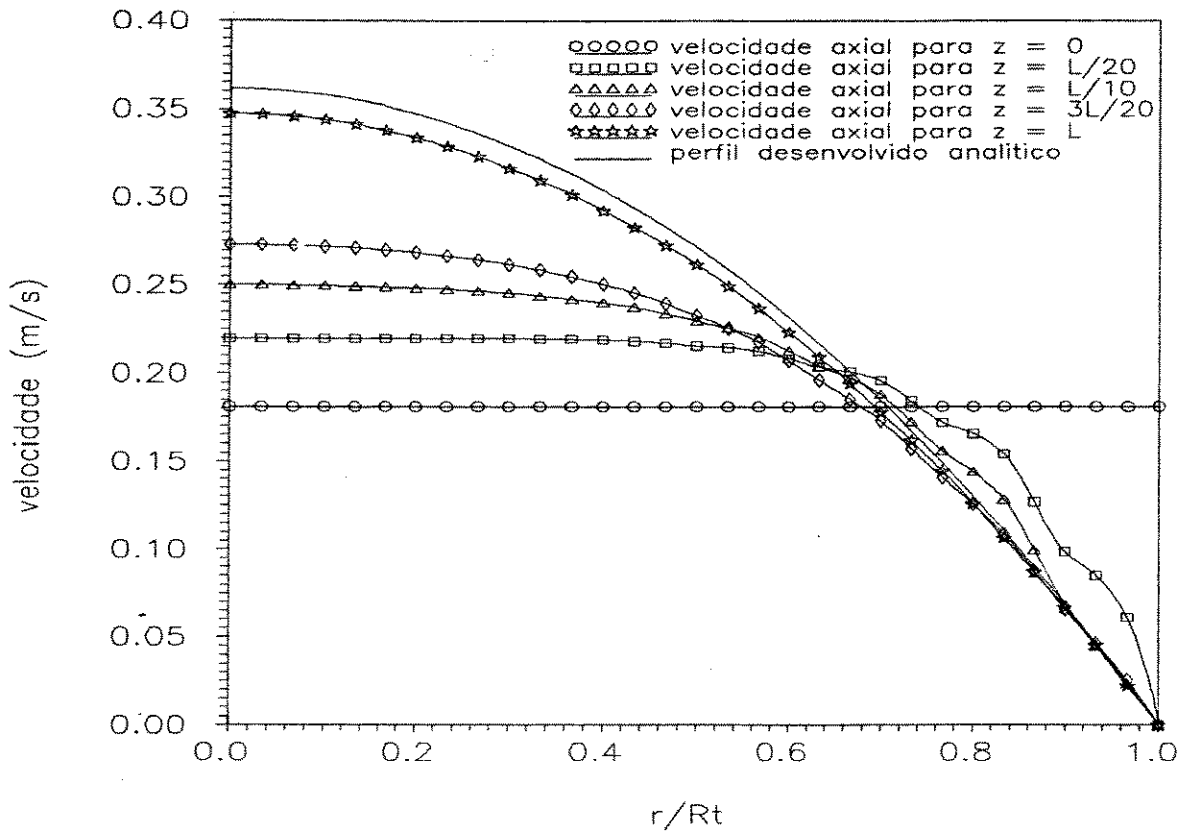


Figura 5.8: Simulação para Reynolds = 700 e malha 10x15 elementos.

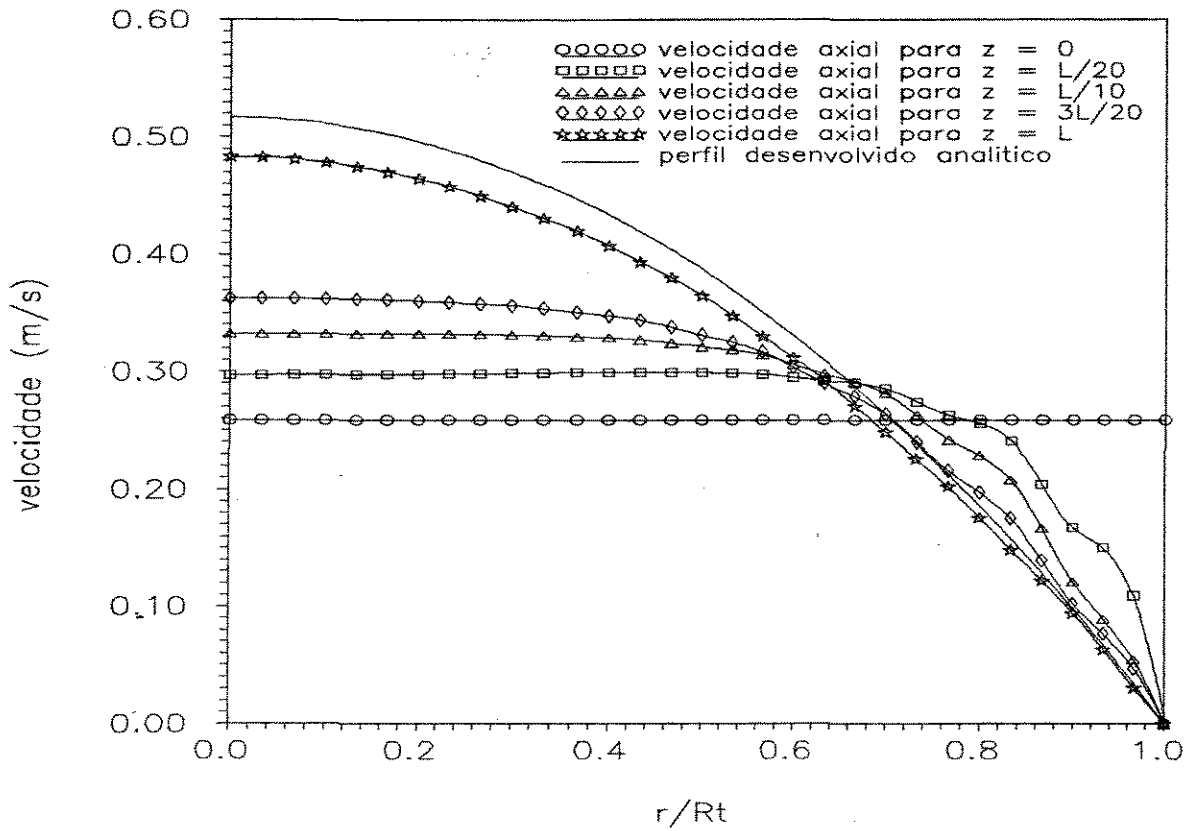


Figura 5.9: Simulação para Reynolds = 1000 e malha 10x15 elementos.

5.2 - Simulação a Gradiente de Pressão Variável

Na simulação com os gradientes de pressão obtidos através de simulações utilizando o método dos volumes finitos, o resultado, como seria de se esperar, apresentou significativa melhora. As simulações foram feitas para números de Reynolds 100, 300 e 500. O fluido utilizado é, como no caso anterior, ar a 25°C. Os resultados são apresentados na Tabela 5.2:

Tabela 5.2: Resultados obtidos e CPU utilizada (gradiente de pressão variável).

malha/Reynolds	$(u_{max}/\bar{u})_{simul.}$	$(u_{max}/\bar{u})_{analit.}$	CPU(s)	Desvio(%)
5x10 / 100	1,998	2,000	48	- 0,10
5x10 / 300	1,992	2,000	60	- 0,40
5x10 / 500	1,984	2,000	60	- 3,00
10x10 / 100	1,997	2,000	240	- 0,15
10x10 / 300	1,998	2,000	300	- 0,10
10x10 / 500	1,996	2,000	495	- 0,20
10x15 / 100	1,997	2,000	2040	- 0,15
10x15 / 300	2,000	2,000	1270	0,00
10x15 / 500	2,001	2,000	1660	+ 0,05

A estimativa inicial para a solução do sistema de equações algébricas, assim como o critério de parada utilizado, são as mesmas da simulação a gradiente de pressão constante.

Com relação a malha 5x10 a convergência foi alcançada com 4 iterações para Re=100 e 5 iterações para Re=300 e Re=500. Verificou-se que o desvio percentual em relação ao valor analítico aumenta à medida em que aumenta o número de Reynolds do escoamento.

No caso da malha 10x10 foram necessárias 4, 5 e 8 iterações para os números de Reynolds 100, 300 e 500, respectivamente. Com esta malha o melhor resultado obtido foi para $Re=300$.

A malha 10x15 apresentou um comportamento atípico no que diz respeito ao número de iterações necessárias para a convergência. Observou-se que o maior número de iterações, no caso 16, ocorreu para o menor número de Reynolds, ou seja, 100 e isto justifica o maior tempo de CPU necessário para este caso. Para os números de Reynolds 300 e 500 o número de iterações foi, respectivamente, 10 e 13.

Quanto aos resultados obtidos, verificou-se que a malha 10x15 apresenta o menor desvio médio, como seria de se esperar, embora seguida de muito perto pela malha 10x10.

CAPÍTULO 6

Conclusões e Sugestões

O método utilizado (colocação ortogonal em elementos finitos) é relativamente simples e mostrou-se eficiente, gerando resultados para o perfil desenvolvido coerentes e próximos aos analíticos, mesmo quando se trabalhou com malhas pouco refinadas (poucos elementos).

Verificou-se que o gradiente de pressão exerce uma influência marcante na determinação dos perfis de velocidade. Utilizando-se um gradiente de pressão mais realístico os resultados obtidos apresentaram sensível melhora.

O programa simulador foi construído de maneira a considerar o mesmo tamanho para todos os elementos axiais, o mesmo acontecendo com relação aos elementos radiais. Para poder-se fazer uma melhor avaliação do comportamento do escoamento na região próxima à entrada do tubo o programa poderia ser modificado de modo a permitir uma redução do comprimento dos elementos axiais nesta região.

Como sugestão propõe-se a aplicação do método a problemas de escoamento turbulento em tubos e posterior utilização na simulação de separadores ciclônicos. Poder-se-ia, nestes casos, devido aos gradientes serem maiores, utilizar-se dois pontos de colocação internos em cada elemento para cada uma das duas direções (axial e radial) ao invés de apenas um.

APÊNDICE A

Matrizes de
Colocação Ortogonal

A Tabela A.1 mostra as matrizes A e B de colocação ortogonal para N=1 e N=2:

Tabela A.1 - Matrizes de colocação ortogonal.

N	A	B
1	$\begin{pmatrix} -3 & 4 & -1 \\ -1 & 0 & 1 \\ 1 & -4 & 3 \end{pmatrix}$	$\begin{pmatrix} 4 & -8 & 4 \\ 4 & -8 & 4 \\ 4 & -8 & 4 \end{pmatrix}$
2	$\begin{pmatrix} -7,000 & 8,196 & -2,196 & 1,000 \\ -2,732 & 1,732 & 1,732 & -0,732 \\ 0,732 & -1,732 & -1,732 & 2,732 \\ -1,000 & 2,196 & -8,196 & 7,000 \end{pmatrix}$	$\begin{pmatrix} 24,000 & -37,180 & 25,180 & -12,000 \\ 16,390 & -24,000 & 12,000 & -4,392 \\ -4,392 & 12,000 & -24,000 & 16,390 \\ -12,000 & 25,180 & -37,180 & 24,000 \end{pmatrix}$

APÊNDICE B

Codificação do Programa Simulador

As páginas a seguir contêm a listagem do programa gerenciador (programa RODI), assim como das três subrotinas principais: STRTUBO.FOR, JCBTUBO.FOR e FNTUBO.FOR.

```

C      ****
C      ***
C      ***          Codigo de calculo que utiliza o metodo          ***
C      ***          de colocacao ortogonal em elementos            ***
C      ***          finitos para solucao de problemas              ***
C      ***          bidimensionais de escoamento laminar          ***
C      ***          em dutos cilindricos                            ***
C      ***
C      ***          -----
C      ***
C      ***          Autor : Evandro Steffani                        ***
C      ***
C      ***
C      ***          UNICAMP / FEQ / LMSPQ                            ***
C      ***
C      ***
C      ****
C
C      ****
C      ***          Definicoes Iniciais                             ***
C      ****
C
C      program rodi
C      common / Arquivos      / ArquInfo,ArqComp,ArqIter,ArqMat,ArqSaida
C      &                      ,ArqSol
C      common / ChuteInic     / Ctex
C      common / Dimensoes    / NYG,Nel,Nmila,Nmica,Ncol
C      common / Nomes        / NomeTeste
C      common / Opcoes       / Icrit,Sing
C      common / Tolerancias  / Xtol,Ftol,Alfa,Fmax,Itmax,Distx
C      common / Paramexec    / Nme,Iopt,IterQN,IterGB,Mtes
C      common / Paramglobal  / Alfabl,Ftolgb,Tolgradf
C      common / tenta       / Ntenta
C
C      real*4 T(1000000),X0(882)
C      integer*4 Ap1,Ap2,Ap3
C
C      ****
C      ***          Supondo-se a Dimensao = 882                    ***
C      ****
C
C      character*20 ArquInfo,ArqComp,ArqIter,ArqMat,ArqSaida
C      character*20 ArqSol
C      character*50 NomeTeste
C      external funcao,jacobiano
C
C      NomeTeste = 'Modelo Laminar'
    
```

```
C
C
C *****
C ***      NYG deve ser exatamente a dimensao do sistema      ***
C *****
```

```
C
C
C NYG = 882
```

```
C
C *****
C ***      O parametro Nmila e' sempre igual a NYG+1      ***
C *****
```

```
C
C
C Nmila = NYG + 1
```

```
C
C *****
C ***      Tolerancias / Parametros Globais      ***
C *****
```

```
C
C
C Itmax = 3000
C Alfa = 1.0e-04
C Xtol = 1.0e-4
C Ftol = 1.0e-04
C Fmax = 1.0e+20
C Sing = 2.0
C Alfabl = 1.0e-04
C Ftolgb = 0.9
C Tolgradf = 1.0e-04
C Distx = 1000.
```

```
C
C *****
C ***      Nomes dos Arquivos      ***
C *****
```

```
C
C
C ArquInfo = 'TesteInfo'
C ArqComp  = 'TesteComp'
C ArqIter  = 'TesteIter'
C ArqMat   = 'N'
C ArqSaida = 'Saida'
C ArqSol   = 'Solucao'
```

```
C
C *****
C ***      Ntm = dimensao do vetor T      ***
C *****
```

```
C
C
C Ntm = 1000000
```

```
C
C *****
C ***      Apontadores      ***
C *****
```

```

Ap1 = 1
Ap2 = Ap1 + NYG
Ap3 = Ap2 + Nmila
Nel = Ntm - NYG - Nmila

```

```

*****
*** Chamada da Subrotina de Estrutura ***
*****

```

```

call str2(T(Ap1),T(Ap2),T(Ap3))
Nel = Ntenta

```

```

write(*,*) 'Nel=',Nel

```

```

*****
*** Se for resolver o sistema so uma vez Numteste = 1 ***
*****

```

```

NumTeste = 1

```

```

*****
*** Metodo e Forma de Execucao ***
*****

```

```

*****
*** Metodo de Newton ***
*****

```

```

Nme = 0
Iopt = 4

```

```

*****
*** Chamada do Pacote Rouxinol ***
*****

```

```

call ROUXINOL(T,X0,Ntm,NumTeste,Funcao,Jacobiano,Iflag)

```

```

*****
*** Fim de Programa ***
*****

```

```

stop
end

```

Subrotina STRTUBO.FOR


```

        write(*,61)
61  format(' Forneca hzi')
    read (8,*)hzi

        write(*,62)
62  format(' Forneca hri')
    read (8,*)hri

        write(*,63)
63  format(' Forneca a viscosidade cinematica do ar')
    read (8,*)xnul

        write(*,67)
67  format(' Forneca a densidade do ar')
    read (8,*)rho

        write(*,65)
65  format(' Forneca o numero de Reynolds ')
    read (8,*) rey

c *****
    raio = ntr * hri
    uo = rey * xnul /2/raio
    dpdz = - (8.* rho * xnul * uo / (raio ** 2))
c *****

    do i = 1,ntz
        hz(i) = hzi
    enddo
    z(1) = 0.
    do i = 2,ntz+1
        z(i) = z(i-1) + hz(i-1)
    enddo
    do j = 1,ntr
        hr(j) = hri
    enddo
    r(1) = 0.
    do j = 2,ntr+1
        r(j) = r(j-1) + hr(j-1)
    enddo

    rr(1) = 0.

    do i=2,(2*ntr+1)
        rr(i) = rr(i-1) + 0.5 * hri
    enddo

```

```
C *****
C                                     Matriz A
C *****

a(1,1)= -3.0
a(1,2)=  4.0
a(1,3)= -1.0
a(2,1)= -1.0
a(2,2)=  0.0
a(2,3)=  1.0
a(3,1)=  1.0
a(3,2)= -4.0
a(3,3)=  3.0

C *****
C                                     Matriz B
C *****

b(1,1)= 4.
b(1,2)= -8.
b(1,3)= 4.
b(2,1)= 4.
b(2,2)=-8.
b(2,3)=4.
b(3,1)=4.
b(3,2)=-8.
b(3,3)=4.

C *****
C      Calculo do Numero de Pontos de Colocacao da Parte Retangular
C *****
naux1 = 2*ntz + 1
naux2 = 2*ntr + 1

C *****
C      Calculo do Numero de Pontos de Colocacao Total por Equacao
C *****
n3 = naux1 * naux2

C *****
C      Estimativa Inicial
C *****

do i=1,n3
  va(i)= uo
enddo

do i=(n3+1),(2*n3)
  va(i) = 0.0
enddo

kk = 1
```



```

C *****
C                               Varredura da Parte Retangular
C *****
do n = 1,2

do i = 1,(naux1*naux2)
  il = float((i-1)/naux1)+1
  jl = i - (il-1)*naux1
  do j = 1,2*n3
    jj = j - float((j-1)/n3)*n3
    ill = float((jj-1)/naux1)+1
    jll = jj - naux1*(ill-1)

    if ((i.gt.1).and.(i.lt.naux1)) then
C *****
      call tipo1(va,ncom,nquem)
C *****
    else

      if (i.eq.((il-1)*naux1 + 1)) then
C *****
        call tipo2(va,ncom,nquem)
C *****
      else

        if (i.eq.(naux1*il)) then
C *****
          call tipo3(va,ncom,nquem)
C *****
        else

          if (il.eq.(2*ntr+1)) then
C *****
            call tipo10(va,ncom,nquem)
C *****
          else

            if ((il-int(il/2.)*2.).ne.0.) then
C *****
              call tipo4(va,ncom,nquem)
C *****
            else

              if ((jl-int(jl/2.)*2.).ne.0.) then

```

```

C *****
      call tipo5(va,ncom,nquem)
C *****
      else
C *****
      call tipo6(va,ncom,nquem)
C *****
      endif
      endif
      endif
      endif
      enddo
      enddo

      enddo
      ncom(2*n3+1) = kk
      Ntenta = kk -1

*****
C                                     Impressao
C *****

      do i=1,kk-1
        write(10,77)i,nquem(i)
77      format('QUEM [',i5,'] = ',i5)
      enddo

      do n = 1,2
        do i = 1,n3
          nn = (n-1)*n3 + i
          write(10,78)n,i,ncom(nn)
78      format('COM [',i4,',',i4,'] = ',i5)
        enddo
      enddo
      nn = nn + 1
      i = i + 1
79      write(10,79)n,i,ncom(nn)
      format('COM [',i4,',',i4,'] = ',i5)

      close(8)
      close(10)
      return
      end

```

```

C *****
C ***
C ***          TIPO - 01          ***
C ***          ***
C *****
C *****
C          Definicoes Iniciais
C *****
C subroutine tipol(va,ncom,nquem)
C parameter (npar=883)
C parameter (npar1=4100)
C implicit real*4(a-h,o-z)
C implicit integer (i,n)
C dimension va(npar-1)
C dimension ncom(npar),nquem(npar1)
C
C common / indices / i,j,l,il,jl,ill,jll,ncont,kk,n
C common / paramet / inter,n3
C
C     if (n.eq.2) then
C         if ((i+(n-1)*n3).eq.j) then
C             ncom(i+(n-1)*n3) = kk
C             nquem(kk) = j
C             kk = kk + 1
C         endif
C     else
C         if ((jll.eq.jl).and.((ill.ge.(il)).and.(ill.le.(il+2)))
C & .and.(float((j-1)/n3).eq.(n-1))) then
C             if (ill.eq.(il)) then
C                 ncom(i+(n-1)*n3) = kk
C             endif
C             nquem(kk) = j
C             kk = kk + 1
C         endif
C     endif
C
C     return
C     end

```



```

C *****
C ***
C ***          TIPO - 06          ***
C ***
C *****
C *****
C          Definicoes Iniciais
C *****
subroutine tipo6(va,ncom,nquem)
parameter (npar=883)
parameter (npar1=4100)
implicit real*4(a-h,o-z)
implicit integer (i,n)
dimension va(npar-1)
dimension ncom(npar),nquem(npar1)
common / indices / i,j,l,il,jl,ill,jll,ncont,kk,n
common / paramet / inter,n3

C *****
C ***          equacao da continuidade          ***
C *****
      if (n.eq.1) then
        if ((jll.eq.jl).and.((ill.ge.(il-1)).and.(ill.le.(il+1)))
& .and.(float((j-1)/n3).eq.1)) then
          nquem(kk) = j
          kk = kk + 1
        endif
        if ((jll.eq.jl).and.(ill.eq.il).and.
& (float((j-1)/n3).eq.0)) then
          do l = j-1,j+1
            if (l.eq.(j-1)) then
              ncom(i) = kk
            endif
            nquem(kk) = l
            kk = kk + 1
          enddo
        endif
      endif
endif

```

```
C *****
C ***                               Navier-Stokes em z                               ***
C *****
  if (n.eq.2) then
C
C         *** Variavel ---> u ***
C
  if ((j11.eq.j1).and.(i11.eq.i1).and.
& ((float((j-1)/n3)).eq.0)) then
    do l = j-1,j+1
      nquem(kk) = l
      kk = kk + 1
    enddo
  endif
  if ((j11.eq.j1).and.(((i11.eq.(i1-1)).or.(i11.eq.(i1+1))))
& .and.((float((j-1)/n3)).eq.0)) then
    if (i11.eq.(i1-1)) then
      ncom(i+n3) = kk
    endif
    nquem(kk) = j
    kk = kk + 1
  endif
  endif
C
C         *** Variavel ---> v ***
C
  if ((j11.eq.j1).and.(i11.eq.i1).and.
& ((float((j-1)/n3)).eq.1)) then
    nquem(kk) = j
    kk = kk + 1
  endif

endif

return
end
```


Subrotina JCBTUBO.FOR

```

C *****
C ***
C ***          SUBROTINA JCBTUBO.FOR          ***
C ***
C *****
C *
C *          Esta subrotina calcula o          *
C *          jacobiano analitico            *
C *
C *
C *****
C          Definicoes Iniciais
C *****
C subroutine jacobiano(va,an)
C parameter (npar=883)
C parameter (npar1=4100)
C implicit real*4(a-h,o-z)
C implicit integer (i,n)
C dimension va(npar-1), an(npar1)
C common / indices / i,j,l,il,jl,ill,jll,ncont,kk,n
C common / paramet / inter,n3
C common / interv / hz(100),hr(100)
C common / points / z(100),r(100)
C common / juca / xnul
C common / matriz / a(3,3)
C common / matrizb / b(3,3)
C common / tamanho / hzi,hri
C common / dimensao / ntz,ntr,naux1,naux2
C common / tenta / Ntenta

C
C open(2,file ='arqjcb' , status ='unknown')
C
C *****          Zerando o vetor an          *****
C
C do i=1,Ntenta
C     an(i) = 0.
C enddo
C *****
C
C kk = 1
C
C *****
C          Varredura da Parte Retangular
C *****
C do n = 1,2
C
C do i = 1,(naux1*naux2)
C     il = float((i-1)/naux1)+1
C     jl = i - (il-1)*naux1

```

```
do j = 1,2*n3
  jj = j - float((j-1)/n3)*n3
  ill = float((jj-1)/naux1)+1
  jll = jj - naux1*(ill-1)

  if ((i.gt.1).and.(i.lt.naux1)) then
c *****
      call ntipol(va,an)
c *****
    else
      if (i.eq.((il-1)*naux1 + 1)) then
c *****
          call ntipo2(va,an)
c *****
      else
        if (i.eq.(naux1*il)) then
c *****
            call ntipo3(va,an)
c *****
        else
          if (il.eq.(2*ntr+1)) then
c *****
              call ntipol0(va,an)
c *****
          else
            if ((il-int(il/2.)*2.).ne.0.) then
c *****
                call ntipo4(va,an)
c *****
            else
              if ((jll-int(jll/2.)*2.).ne.0.) then
c *****
                  call ntipo5(va,an)
c *****
              else
                call ntipo6(va,an)
c *****
              endif
            endif
          endif
        endif
      endif
    endif
  endif
endif
```

```

                endif
            endif
        endif

        endif
    endif
enddo
enddo

enddo

C *****
C                                     Impressao
C *****

do i=1,kk-1
write(2,*)'an (' ,i,') =' ,an(i)
enddo

do i=1,2*n3
    write(2,*)'va(' ,i,')=' ,va(i)
enddo

close(2)

return
end

C *****
C ***                                     ***
C ***                                     ***
C ***                                     ***
C ***                                     ***
C *****

C *****
C                                     Definicoes Iniciais
C *****
subroutine ntipol(va,an)
parameter (npar=883)
parameter(npar1 = 4100)
implicit real*4 (a-h,o-z)
implicit integer (i,n)
dimension va(npar-1),an(npar1)
common / indices / i,j,l,il,jl,ill,jll,ncont,kk,n
common / paramet / inter,n3
common / matriz / a(3,3)

```



```

C *****
C ***                               Navier Stokes em z                               ***
C *****

      if (n.eq.2) then

C
C ***   variavel u ***
C
      if ((j11.eq.j1).and.(i11.eq.il)
& .and.(float((j-1)/n3).eq.0)) then
        valor = va(j+n3)
        inter = (j1/2)
        do l = 1,3
          an(kk) = (a(2,l)*va(j))/hz(inter)-xnul*b(2,l)/(hzi**2)
          if (l.eq.2) then
            an(kk) = an(kk) + (a(2,1)*va(j-1)+a(2,3)*va(j+1))/hz(j1/2)
          endif
          kk = kk + 1
        enddo
      endif
      if((j1.eq.j11).and.((i11.eq.(il-1)).or.(i11.eq.(il+1))).and.
& (float((j-1)/n3).eq.0)) then
        if (i11.eq.(il-1)) then
          valor0 = va(j)
          kk = kk + 1
        else
&          an(kk) = (a(2,(i11-il+2))*valor)/hr(il/2)
          -xnul*a(2,3)/hri/(0.5*hri+r(il/2))
          chips = -xnul*b(2,3)/(hri**2)
          valor2 = va(j)
          kk = kk + 1
          an(kk-5) = - an(kk-1) + chips
          an(kk-1) = an(kk-1) + chips
          an(kk-3) = an(kk-3) - xnul*b(2,2)/(hri**2)
        endif
      endif

C
C ***   variavel v ***
C
      if((j1.eq.j11).and.(i11.eq.il).and.
& (float((j-1)/n3).eq.1)) then
        inter = il/2
        an(kk) = (a(2,1)*valor0+a(2,3)*valor2)/hr(inter)
        kk = kk + 1
      endif

      endif
      return
      end

```


Subrotina FNTUBO.FOR

```

C *****
C ***
C ***          SUBROTINA FNTUBO.FOR          ***
C ***
C *****
C *
C *          Esta subrotina avalia o          *
C *          valor das funcoes                *
C *
C *
C *****
C          Definicoes Iniciais
C *****
C subroutine funcao(va,f)
C parameter (npar=883)
C parameter (npar1=4100)
C implicit real*4(a-h,o-z)
C implicit integer (i,n)
C dimension va(npar-1), f(npar-1)
C common / indices / i,j,l,il,jl,ill,jll,ncont,kk,n
C common / paramet / inter,n3
C common / interv / hz(100),hr(100)
C common / points / z(100),r(100)
C common / juca / xnul
C common / matriz / a(3,3)
C common / matrizb / b(3,3)
C common / tamanho / hzi,hri
C common / entrada / uo
C common / gradp / dpdz
C common / densidade / rho
C common / dimensao / ntz,ntr,naux1,naux2
C common / fgh / rr(30)

C
C open(30,file ='arqfn', status ='unknown')
C open(9,file ='resposta' , status ='unknown')

C *****          Zerando o vetor f          *****
C
C do i=1,2*n3
C     f(i) = 0.
C enddo
C *****

```

```

C *****
C                               Varredura da Parte Retangular
C *****
do n = 1,2

do i = 1,(naux1*naux2)
  kk = i+(n-1)*n3
  il = float((i-1)/naux1)+1
  jl = i - (il-1)*naux1
  do j = 1,2*n3
    jj = j - float((j-1)/n3)*n3
    ill = float((jj-1)/naux1)+1
    jll = jj - naux1*(ill-1)

    if ((i.gt.1).and.(i.lt.naux1)) then
C *****
C                               call nntipol(va,f)
C *****
    else
        if (i.eq.((il-1)*naux1 + 1)) then
C *****
C                               call nntipo2(va,f)
C *****
    else
        if (i.eq.(naux1*il)) then
C *****
C                               call nntipo3(va,f)
C *****
    else

        if (il.eq.(2*ntr+1)) then
C *****
C                               call nntipol0(va,f)
C *****
    else

        if ((il-int(il/2.)*2.).ne.0.) then
C *****
C                               call nntipo4(va,f)
C *****

    else
        if ((jl-int(jl/2.)*2.).ne.0.) then

```

```
C *****
      call nntipo5(va,f)
C *****
      else
C *****
      call nntipo6(va,f)
C *****
      endif
      endif
      endif
      endif
      endif
      enddo
      enddo

      enddo

C *****
C                               Impressao
C *****

      do n = 1,2
        do i=1,n3
          kaux = i + (n-1)*n3
          write(30,1001) n,i,f(i+(n-1)*n3),kaux
1001    format('X [' ,i5,' , ' ,i5,' ] = ' ,g14.7,2x,'kk = ' ,i5)
          enddo
        enddo
        do i=1,2
          do j = 1,n3
            write(30,1111)j+(i-1)*n3, va(j+(i-1)*n3)
1111    format('va ( ' ,i5,' ) = ' ,g14.7)
            enddo
          enddo

          jj = 1
          do j=1, (n3-2*ntz) , (2*ntz+1)
            write(9,1120)rr(jj),va(j+1),va(j+2),va(j+3),va(j+20)
1120    format(g14.7,2x,g14.7,2x,g14.7,2x,g14.7,2x,g14.7)
            jj=jj+1
          enddo

          close(30)
          close(9)

          return
        end
```



```

C *****
C ***
C ***
C ***
C *****
C *****
C Definicoes Iniciais
C *****
C subroutine nntipo2(va,f)
C parameter (npar=883)
C parameter(npar1 = 4100)
C implicit real*4 (a-h,o-z)
C implicit integer (i,n)
C dimension va(npar-1),f(npar-1)
C common / indices / i,j,l,il,jl,ill,jll,ncont,kk,n
C common / paramet / inter,n3
C common / entrada / uo
C
C if ((i+(n-1)*n3).eq.j) then
C   if (n.eq.1) then
C     f(kk) = va(j) - uo
C   else
C     f(kk) = va(j)
C   endif
C endif
C
C return
C end
C *****
C ***
C ***
C ***
C *****
C *****
C Definicoes Iniciais
C *****
C subroutine nntipo3(va,f)
C parameter (npar=883)
C parameter(npar1 = 4100)
C implicit real*4 (a-h,o-z)
C implicit integer (i,n)
C dimension va(npar-1),f(npar-1)
C common / indices / i,j,l,il,jl,ill,jll,ncont,kk,n
C common / paramet / inter,n3
C common / matriz / a(3,3)

```



```

C *****
C ***
C ***
C ***
C *****
C *****
C Definicoes Iniciais
C *****
subroutine nntipo5(va,f)
parameter (npar=883)
parameter(npar1 = 4100)
implicit real*4 (a-h,o-z)
implicit integer (i,n)
dimension va(npar-1),f(npar-1)
common / indices / i,j,l,il,jl,ill,jll,ncont,kk,n
common / paramet / inter,n3
common / matriz / a(3,3)
common / tamanho / hzi,hri

    if (j.eq.(i+(n-1)*n3)) then
        ncont = 0
        do l = i - 2 , i
            ncont = ncont + 1
            f(kk) = f(kk) + a(3,ncont)*va(1)/hzi
        enddo
        ncont = 0
        do l = i , i + 2
            ncont = ncont + 1
            f(kk) = f(kk) - a(1,ncont)*va(1)/hzi
        enddo
    endif

return
end

```



```

C *****
C ***                               Navier Stokes em z                               ***
C *****

      if (n.eq.2) then

C
C ***   variavel u   ***
C
      if ((j11.eq.j1).and.(i11.eq.i1)
& .and.(float((j-1)/n3).eq.0)) then
          f(kk) = f(kk) + (a(2,1)*va(j-1)+a(2,3)*va(j+1))/
&      hz(j1/2)*va(j) - xnul*(b(2,1)*va(j-1)+b(2,2)*va(j)+
&      b(2,3)*va(j+1))/(hzi**2)

      endif

      if((j1.eq.j11).and.((i11.eq.(i1-1)).or.(i11.eq.(i1)).
& or.(i11.eq.(i1+1))).and.
& (float((j-1)/n3).eq.0)) then
          f(kk) = f(kk) - xnul*b(2,i11-i1+2)*va(j)/(hri**2)
&      - xnul*a(2,i11-i1+2)*va(j)/hri/(0.5*hri+r(i1/2))
      endif

C
C ***   variavel v   ***
C
      if ((j11.eq.j1).and.(i11.eq.i1)
& .and.(float((j-1)/n3).eq.1)) then
          valor = va(j)
      endif

      if((j1.eq.j11).and.((i11.eq.(i1-1)).or.(i11.eq.(i1+1))).and.
& (float((j-1)/n3).eq.1)) then
          sum = sum + (a(2,(i11-i1+2))*va(j-n3))/hr(i1/2)
          if (i11.eq.(i1+1)) then
              f(kk) = f(kk) + sum * valor + dpdz/rho
              sum = 0.
          endif
      endif

      endif

      endif

      return
      end

```


Referências bibliográficas

[1] Bird, R. B., Stewart, W. E. and Lightfoot, E. N. - Transport Phenomena, John Wiley & Sons, New York, 1960.

[2] Carey, G.F. and Finlayson, B.A. - Orthogonal Collocation on Finite Elements, Chem. Engng. Sci., vol. 30, 587-596, 1975.

[3] Carvalho, C. B. - Desenvolvimento de um Código de Cálculo Utilizando o Método dos Volumes Finitos e o Modelo de Turbulência $k-\epsilon$ para Solução de Problemas Bidimensionais; Tese (Mestrado em Engenharia Mecânica), Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas, 1993.

[4] Chang, P. W., Patten, T. W. and Finlayson, B. A. - Collocation and Galerkin Finite Element Methods for Viscoelastic Fluid Flow, Part I. Description of Method and Problems with Fixed Geometry. Comp. Fluids, vol. 7, 267-283, 1979.

[5] Finlayson, B. A. - Packed Bed Reactor Analysis by Orthogonal Collocation, Chem. Engng. Sci., vol. 26, 1081-1091, 1971.

[6] Finlayson, B. A. - Orthogonal Collocation in Chemical Reaction Engineering, Cat. Rev. - Sci. Eng., 10 (1), 69-138, 1974.

[7] Finlayson, B. A. - Nonlinear Analysis in Chemical Engineering, McGraw-Hill, 1980.

[8] Houstis, E. N., Lynch, R. E., Rice, J. R. and Papatheodorou, T. S.- Evaluation of Numerical Methods for Elliptic Partial Differential Equations, J. Comp. Phys., vol. 27, 323-350, 1978.

[9] Huyakorn, P. S., Taylor, C., Lee, R. L. and Gresho, P. M.- A Comparison of Various Mixed-interpolation Finite Elements in the Velocity-Pressure Formulation of the Navier-Stokes Equations, *Comp. Fluids*, vol. 6, 25-35, 1978.

[10] Jensen, O. K. and Finlayson, B. A. - Oscillation Limits for Weighted Residual Methods Applied to Convective Diffusion Equations, *Int. J. Num. Methods Eng.*, vol. 15, 1681-1689, 1980.

[11] Kurtanjek, Z. - Optimal Nonsingular Control of Fed-batch Fermentation, *Biotechnology and Bioengineering*, vol. 37, 814-823, 1991.

[12] Ma, Z. and Guichon, G. - Application of Orthogonal Collocation on Finite Elements in the Simulation of Non-linear Chromatography, *Computers chem. Engng.*, 15 (6), 415-426, 1991.

[13] Meier, H. F. - Modelagem, Simulação e Otimização de um Reator de Leito Móvel para Pirólise de Finos de Xisto; Tese (Mestrado em Engenharia Química) - Faculdade de Engenharia Química, Universidade Estadual de Campinas, 1990.

[14] Patankar, S. V. - *Numerical Heat Transfer and Fluid Flow*, Hemisphere Publishing Corporation, 1980.

[15] Prenter, P. M. and Russell, R. D. - Orthogonal Collocation for Elliptic Partial Differential Equations, *SIAM J. Num. Anal.*, vol. 13, 923-939, 1976.

[16] Sorensen, J. P., Guertin, E. W. and Stewart, W. E. - Computational Models for Cylindrical Catalyst Particles, *AIChE Journal*, 19 (5), 969-975, 1973.

[17] Stevens, D. K., Berthouex, P. M. and Chapman, T. W. - Calculation of Effectiveness Factors in Spherical Shells, *J. Envir. Engng.*, 113 (5), 1149-1155, 1987.

[18] Stevens, D. K. - Interaction of Mass Transfer and Inhibition in Biofilms, *J. Envir. Engng.*, 114 (6), 1352-1358, 1988.

[19] Villadsen, J. V. and Stewart, W. E. - Solution of Boundary-value Problems by Orthogonal Collocation, *Chem. Engng. Sci.*, vol. 22, 1483-1501, 1967.

[20] Villadsen, J. and Sorensen, J. P. - Solution of Parabolic Partial Differential Equations by a Double Collocation Method, *Chem. Engng. Sci.*, vol. 24, 1337-1349, 1969.

[21] Villadsen, J. V. and Michelsen, M. L. - Solution of Differential Equation Models by Polynomial Approximation, Prentice-Hall, Englewood Cliffs, New Jersey, 1978.

[22] Wang, C. M., Ong, S. L. and Ang, K. K. - Calculation of Effectiveness Factors for Spherical Shells Using Shooting Technique, *J. Envir. Engng.*, 117 (6), 859-864, 1991.

Resumo

Steffani, Evandro. Método de Colocação Ortogonal em Elementos Finitos Aplicado à Simulação de Problemas Bidimensionais de Escoamento Laminar em Dutos Cilíndricos.

Este trabalho consiste na elaboração de uma rotina de cálculo para a simulação do escoamento laminar de fluidos em dutos cilíndricos. As equações da continuidade e a componente axial da equação de Navier-Stokes são resolvidas utilizando-se o método de colocação ortogonal em elementos finitos. Como resultado deste trabalho tem-se um programa fácil de se utilizar que calcula os perfis de velocidade, para o escoamento de um fluido newtoniano, no interior de tubos. O programa é testado fazendo-se simulações do escoamento laminar de ar a 25°C a diferentes números de Reynolds, variando-se também o tamanho da malha, ou seja, o número de elementos finitos utilizados nas direções axial e radial. Os resultados obtidos são comparados à solução analítica para perfil desenvolvido.

Abstract

This work presents a computer program for the simulation of the laminar flow of fluids in cylindrical ducts. The continuity equation and the axial component of Navier-Stokes equation are solved using the orthogonal collocation on finite elements method. As a result we have a simple program which gives the velocity profiles for the flow of a Newtonian fluid. The program is tested using air at 25°C at different Reynolds numbers and different grid sizes. The results are compared to the analytical solution for fully developed flow.