

**ANSELMO MARTINEZ**

**APLICAÇÃO DO MÉTODO DOS VOLUMES FINITOS  
NA SIMULAÇÃO DE ESCOAMENTOS  
NÃO VISCOSOS**

Dissertação apresentada à  
Escola Politécnica da  
Universidade de São Paulo  
para obtenção do título de  
Mestre em Engenharia.

São Paulo

1997

Biblioteca da Escola Politécnica  
Universidade de São Paulo

FD-2014

FD-2017

DEDALUS - Acervo - EPBC



31200030126

**ANSELMO MARTINEZ**

**APLICAÇÃO DO MÉTODO DOS VOLUMES FINITOS  
NA SIMULAÇÃO DE ESCOAMENTOS  
NÃO VISCOSOS**

Dissertação apresentada à  
Escola Politécnica da  
Universidade de São Paulo  
para obtenção do título de  
Mestre em Engenharia.

Área de Concentração:  
Engenharia Mecânica

Orientador:  
Prof. Dr. Douglas Lauria

São Paulo

1997

OK

A Ciência é escrita num grande livro, que  
sempre se estende ante aos nossos olhos; mas  
não podemos entendê-la se não aprendermos  
primeiro a linguagem na qual é escrita.

Esta linguagem é a **Matemática**.

**GALILEO GALILEI (1564-1642)**

## **AGRADECIMENTOS**

Ao orientador Prof. Dr. Douglas Lauria pelas diretrizes seguras e ao seu permanente incentivo dado durante todo o desenvolvimento deste trabalho.

Ao Instituto Mauá de Tecnologia pela oportunidade e apoio dado, e a todos que direta ou indiretamente colaboraram na execução deste trabalho.

## ERRATA

Página	Linha	
7	1	sai: "(p. ex. falsa vorticidade)" entra: "que conduzem a soluções irreais do escoamento..."
11	5	sai: "A eq. 2.1 pode ser rescrita na forma integral." entra: "A mesma eq. pode ser escrita na forma integral."
11	18	sai: "as tensões de cisalhamento e os termos de condução de calor" entra: "os termos dissipativos"
46	1	sai: "se mostra rápida, já que em torno de" entra: "foi obtida com cerca de"
46	2	sai: "tem-se" entra: ", com"
51		entra após a fig. 5.5: "A não simetria nos valores de velocidade das linhas de corrente em relação ao centro, pode ser conseqüente à malha grosseira e ao reduzido trecho de conduto considerado na saída (apenas 3 subdivisões no sentido axial)."
53		entra após 1º parágrafo: "A pressão passou a ser representada pelo número de Euler, tomada a velocidade à saída do tubo como referência, e a velocidade foi adimensionalizada também pela velocidade na saída."
54	2	sai: "obtida" entra: "medida"
54		nas figuras 5.8 a 5.12 substituir no eixo das ordenadas e nos títulos "velocidade" por "coeficiente de velocidade";
58		nas figuras 5.13 a 5.15 substituir no eixo das ordenadas "P-Pentr" por "coeficiente diferencial de pressão" e nos títulos substituir "pressão" por coeficiente diferencial de pressão";
63	4	sai: "Porém, ele se mostrou incondicionalmente divergente." entra: ", o que não foi conseguido. Não foram feitas outras tentativas."
66	8	sai: "verificar os principais pontos positivos..." até a linha 16 "... -incondicional divergência para escoamentos com rotação." entra: "concluir que o método apresenta-se tão mais conveniente quanto menor a complexidade do escoamento. A partir das malhas analisadas, observa-se uma sensível influência do número de células na direção normal ao escoamento, evidenciado nos gráficos 5.8 a 5.15. Observa-se, a partir das malhas analisadas, problemas na determinação dos campos de pressão e velocidade na região de curvatura, evidenciados nas figuras 5.8 a 5.12. Com relação aos efeitos de rotação nada deve ser concluído, pois na única simulação efetuada não houve convergência."
67	7	sai todo o parágrafo; entra: "Deve-se aprofundar no estudo matemático de canais com fortes curvaturas para verificar qual a influência deste no escoamento, além do aprofundamento no estudo e análise de escoamentos com movimento de rotação, já que neste trabalho foi realizada somente uma simulação com movimento de rotação. Deve-se ainda discretizar os coeficientes métricos de tal maneira que se tornem de 2ª ordem, verificando a influência desta discretização nos resultados, já que da maneira como foram discretizados se tornaram de 1ª ordem."
70	14	sai: "a dissipação, o fenômeno do transporte de viscosidade, da difusão de massa e condutividade térmica." entra: "os termos dissipativos:"

## SUMÁRIO

Lista de Figuras .....	<i>i</i>
Lista de Tabelas .....	<i>ii</i>
Nomenclatura .....	<i>iii</i>
Resumo .....	<i>iv</i>
Abstract .....	<i>v</i>
<b>1. Introdução .....</b>	<b>1</b>
Referências .....	7
<b>2. Formulação Matemática .....</b>	<b>9</b>
2.1. Equação da Continuidade .....	10
2.2. Equação de Euler .....	11
2.3. Método da Compressibilidade Artificial .....	13
2.4. Transformação de Coordenadas .....	15
Referências .....	17
<b>3. Discretização .....</b>	<b>19</b>
3.1. Método dos Volumes Finitos .....	20
3.2. Método de Runge-Kutta .....	23
3.3. Dissipação Artificial .....	26
3.4. Condições de Contorno .....	28
Referências .....	30

<b>4. Desenvolvimento Computacional</b> .....	32
4.1. Função “Leitura de Coordenadas” .....	34
4.2. Função “Entrada de Dados” .....	37
4.3. Função “Processamento” .....	38
4.4. Função “Saída de Dados” .....	42
<b>5. Aplicações</b> .....	43
5.1. Bocal .....	44
5.2. Tubo de Stanitz .....	51
5.3. Rotor de Bomba Centrífuga .....	60
<b>6. Conclusões e Recomendações</b> .....	65
6.1. Desenvolvimentos Futuros .....	66
<b>Anexo I: Equação de Euler</b> .....	68
<b>Anexo II: Transformação de Coordenadas</b> .....	75
<b>Anexo III: Listagem do Programa</b> .....	80
<b>Bibliografia</b> .....	99
<b>Apêndice I: Coordenadas Dos Pontos Da Malha Do Bocal</b> .....	109

## LISTA DE FIGURAS

### CAPÍTULO 1

- 1.1 - Total de horas de ensaio em túnel de vento no desenvolvimento de aeronaves .....3  
 1.2 - Custo computacional relativo a uma dada simulação .....4

### CAPÍTULO 2

- 2.1 - Volume de Controle Finito .....10  
 2.2 - Forma Conservativa para uma Quantidade Escalar .....13  
 2.3 - Sistema de Coordenadas .....16

### CAPÍTULO 3

- 3.1 - Célula de Controle  $i,j$  .....21  
 3.2 - Região de Estabilidade para o Método de Runge-Kutta .....25  
 3.3 - Condições de Contorno .....29

### CAPÍTULO 4

- 4.1 - Fluxograma Geral .....34  
 4.2 - Malha Exemplo .....35  
 4.3 - Célula de Controle .....37  
 4.4 - Secção de Entrada .....41

### CAPÍTULO 5

- 5.1 - Malha para Bocal .....44  
 5.2 - Log da Tolerância  $\epsilon_p$  x Iteração .....45  
 5.3 - Log da Tolerância  $\epsilon_u$  ( $\epsilon_v$ ) x Iteração .....46  
 5.4 - Constante de Bernoulli .....50  
 5.5 - Velocidade ao longo das linhas de  $j$  constante .....51  
 5.6 - Tubo de Stanitz .....52  
 5.7 - Malha com 23x5 células e condições de contorno .....53  
 5.8 - Velocidades obtidas por Kramer/Stanitz .....54



5.9 - Velocidade sobre as paredes (23x5 células) .....	55
5.10 - Velocidade sobre as paredes (23x10 células).....	56
5.11 - Velocidade sobre as paredes (23x15 células).....	57
5.12 - Velocidade sobre as paredes (23x20 células).....	57
5.13 - Distribuição de pressão (Kramer/Stanitz) .....	58
5.14 - Distribuição de pressão (23x5 células) .....	59
5.15 - Distribuição de pressão (23x20 células) .....	59
5.16 - Plano meridiano do rotor .....	62
5.17 - Malha para o plano meridiano do rotor .....	63
<b>ANEXO I</b>	
AI.1. Partícula Fluida .....	69

## LISTA DE TABELAS

### CAPÍTULO 4

4.1 - Formato do Arquivo de Entrada .....36

4.2 - Formato do Arquivo de Saída .....42

### CAPÍTULO 5

5.1 - Arquivo de Resultados para o Bocal .....47

## NOMENCLATURA

### LETRAS LATINAS

$c$ :	pseudo-velocidade do som
$D$ :	dissipação artificial
$dS$ :	área de controle infinitesimal
$dV$ :	volume de controle infinitesimal
$f$ e $g$ :	componentes do vetor de fluxo $q$
$F$ e $G$ :	componentes do vetor de fluxo $Q$
$h$ :	acobiano da transformação
$n$ :	iteração
$p$ :	pressão
$\bar{q}$ e $\bar{Q}$ :	vetores de fluxo
$t$ :	tempo
$u$ e $v$ :	componentes cartesianas da velocidade
$U$ e $V$ :	contra-variantes da velocidade
$\bar{V}$ :	vetor velocidade
$w$ e $W$ :	variáveis conservativas
$x$ e $y$ :	coordenadas cartesianas
$I$ e $J$ :	número de pontos na direção do escoamento e normal ao escoamento

### LETRAS GREGAS

$\rho$ :	massa específica do fluido
$\beta$ :	$\rho \cdot c^2$
$\eta$ e $\xi$ :	coordenadas curvilíneas

### ÍNDICES

$i$ e $j$ :	posição do elemento na malha
-------------	------------------------------

## RESUMO

É apresentada neste trabalho uma metodologia para a determinação de campos de escoamento (velocidade e pressão) utilizando-se micro-computadores tipo PC (Personal Computer), hoje amplamente utilizados devido ao seu baixo custo.

Esta metodologia baseia-se na solução da equação de Euler para obter o campo de velocidades, que considera o escoamento como sendo não viscoso, porém rotacional. Considera-se ainda que o escoamento seja incompressível e o regime permanente. Na obtenção do campo de pressões é utilizada a equação da continuidade modificada.

A discretização das equações acima citadas é feita pelos seguintes métodos: o termo temporal da Equação de Euler é discretizado pelo Método de Runge-Kutta de 4ª ordem, sendo a solução obtida quando este termo tender a zero, e o termo espacial pelo Método dos Volumes Finitos.

Já para o campo de pressões, a discretização da equação da continuidade modificada é feita pelo Método da Compressibilidade Artificial, ou Pseudo-Compressibilidade, que considera a equação da continuidade para um escoamento compressível, cuja solução também será obtida quando o termo temporal tender a zero.

A metodologia descrita pode ser utilizada na determinação dos campos de pressões e velocidades de escoamentos considerados como não viscosos e incompressíveis, mas que sejam internos, como escoamentos em canais, em componentes de máquinas de fluxo trabalhando no ponto de projeto, etc.

## ABSTRACT

A methodology for resolution of the flow field (velocity and pressure) is presented at this explanation. Micro-Computers type PC (Personal Computer) are wide used today because of their low price.

This methodology is based on the Euler equation to obtain the velocity field solution, which considers the flow like inviscid and rotational. The flow is still considered incompressible and steady.

The discretization of the above equations is performed by the following methods: the term that depends on the time, in the Euler equations, is discretized by the 4th order Runge-Kutta method, whose solution is obtained when this term is tended to zero. Regarding to spatial term, it is discretized by the Finite Volume Method.

The discretization for the pressure flow is made by the modified Continuity Equation, knew like Artificial Compressibility or Pseudo-Compressibility. It is considered the continuity equation for a compressible flow, whose solution is obtained when the time term tend to zero.

The described methodology can be used for determination of the velocity and pressure fields, what is considered like inviscid and incompressible flow, but must be an internal flow, like channels, flood machines components working in the project conditions, and so on.

## INTRODUÇÃO

A utilização dos computadores digitais na Engenharia, principalmente para a solução de problemas na área da dinâmica dos fluidos, deu-se desde o seu aparecimento no final da década de 30 quando desenvolvido por Atanasoff. (Anderson, Tannehill e Pletcher, 1984). Porém, sua utilização só tomou um grande impulso quando os computadores digitais com grande velocidade de processamento foram disponibilizados na década de 60.

A solução analítica das equações diferenciais de derivadas parciais, que representam o fenômeno em estudo, é também conhecida como método teórico. A solução para este modelo matemático, se verificada na literatura clássica de mecânica dos fluidos, está restrita a uma parte dos problemas práticos, onde os mais complexos, modelados pela equação de Navier-Stokes, não possuem uma solução analítica exata.

Para estes problemas complexos, cuja representação matemática geralmente é realizada por equações diferenciais de derivadas parciais, se tem utilizado sistemas de equações aritméticas simples que representam o fenômeno em estudo.

Na investigação experimental tem-se o problema de realizar medições, onde para alguns ensaios é necessário construir um protótipo em escala reduzida e extrapolar os resultados, e em outros as medidas são impossíveis de serem feitas até mesmo em escala reduzida. A introdução de sondas num escoamento para medição de algumas grandezas, como por exemplo a pressão, pode levar a distúrbios do fenômeno e a medidas que não correspondem ao escoamento em estudo. Já com a introdução do laser evitou-se o problema da influência do equipamento nas medidas, mas deve-se observar que os instrumentos de medida não estão livres de erros e que as medidas são realizadas somente em alguns pontos de interesse, e não em todo o domínio em análise.

Os métodos computacionais, ou numéricos, empregados pela Dinâmica dos Fluidos Computacional têm mostrado uma série de vantagens sobre os outros: a solução numérica fornece informação detalhada e completa com resultados para todas as variáveis ao longo de todo o domínio de interesse, o que não acontece com o método experimental, além de não provocar distúrbios nos resultados devido aos equipamentos. O método numérico oferece condições de se fazer simulações dentro das condições reais do fenômeno, não importando se os valores de dimensão são pequenos ou grandes, se as temperaturas são baixas ou altas e se a escala de tempo do processo é em milissegundos ou horas.

Outro motivo tão importante quanto os já citados para a utilização do método numérico é o custo. Chapman (1975) cita que a quantidade de horas para a simulação experimental necessária para o projeto de uma nova aeronave vem crescendo numa taxa exponencial.

Como pode ser visto na fig. 1.1, para o projeto de uma nova aeronave seria necessário o equivalente a 10 anos de ensaios em túnel de vento antes do primeiro vôo, com um custo aproximado de US\$ 100 milhões de dólares.

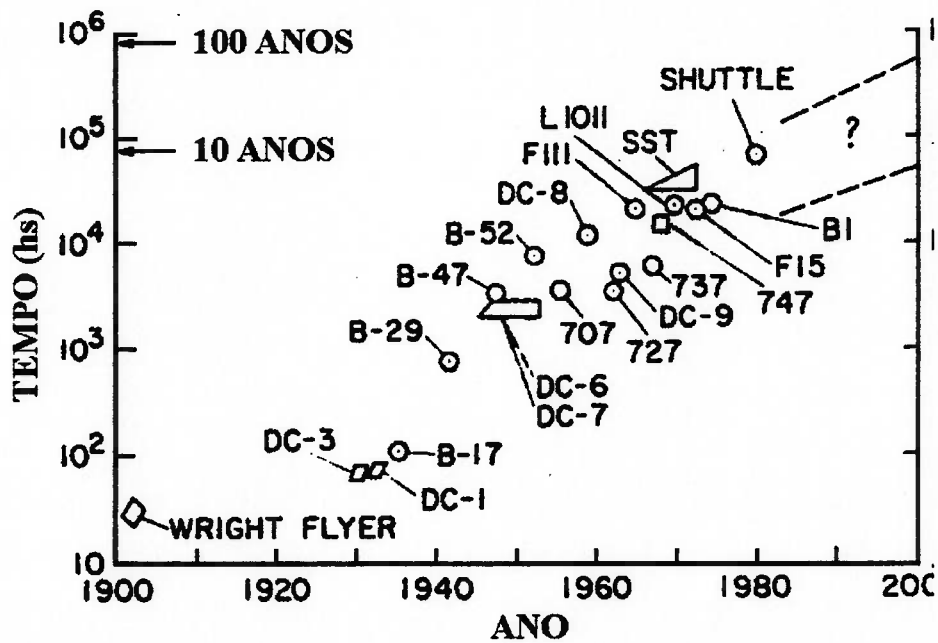


fig. 1.1: Total de horas de ensaio em túnel de vento no desenvolvimento de aeronaves

Em contraste ao crescimento do custo na utilização de laboratórios, o custo computacional relativo à simulação de um



dado escoamento vem decrescendo a um fator de 10 vezes a cada período de aproximadamente 5 anos (fig. 1.2).

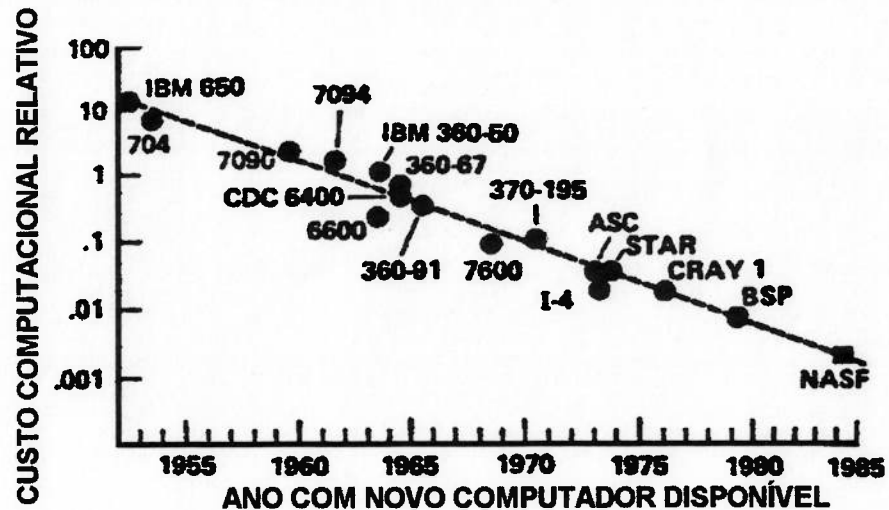


fig. 1.2: Custo computacional relativo a uma dada simulação

Apesar das vantagens já citadas dos métodos numéricos, verifica-se na prática a utilização do método experimental complementando o numérico. Com o método numérico se faz várias simulações até chegar ao projeto que é considerado ideal (por exemplo, determinando a geometria das pás de um rotor de uma bomba centrífuga) para em seguida ir ao laboratório ensaiar o modelo obtido na análise numérica e fazer medições nos pontos importantes ou que foram considerados críticos na simulação numérica.

Em aerodinâmica, por exemplo, as principais limitações na utilização do túnel de vento são o tamanho do modelo, velocidade, densidade, temperatura, interferência das paredes e dos suportes, etc., enquanto as limitações da Dinâmica dos Fluidos Computacional são velocidade de processamento, memória e a discretização das equações de movimento escolhidas para

representar o fenômeno. Desde que as limitações dos dois métodos são completamente diferentes, é óbvio que as facilidades dadas pelos métodos são inerentemente complementares. A importância relativa de cada método vai depender do fenômeno físico em particular que se está analisando.

Devido ao alto preço, a utilização dos computadores para simulação de escoamentos estava restrita somente as grandes empresas e centros de pesquisa no início de seu uso. Chapman (1979) observa que a simulação numérica de um escoamento ao redor de um perfil aerodinâmico, utilizando-se o modelo matemático proposto por Reynolds, podia ser conduzida em supercomputadores da época em menos de meia hora e por menos de US\$1,000 de custo em tempo de computador e que, 20 anos antes, o custo aproximado de utilização do computador para a mesma simulação era de aproximadamente US\$10,000, porém não se poderia esperar resultados antes de 10 anos de pesquisa.

Nota-se portanto que nos últimos anos houve um grande progresso na área da Dinâmica dos Fluidos Computacional devido tanto ao desenvolvimento dos computadores quanto das técnicas numéricas. A limitação imposta aos métodos computacionais está na velocidade de processamento e na memória dos computadores, além de algoritmos que possam produzir soluções para escoamentos complexos.

Mesmo com estes desenvolvimentos ainda se faz necessário reduzir o tempo de processamento através de esquemas e metodologias eficientes que descrevam de maneira satisfatória o fenômeno físico em máquinas de menor porte, ou seja, micro-computadores.

Neste trabalho é apresentada uma metodologia para a determinação de campos de escoamento (velocidade e pressão) utilizando-se micro-computadores tipo PC (Personal Computer), hoje amplamente utilizados devido ao seu baixo custo.

Esta metodologia baseia-se na solução da equação de Euler para obter o campo de velocidades, que considera o escoamento como sendo não viscoso, porém rotacional. Considera-se ainda que o escoamento seja incompressível e o regime permanente. Na obtenção do campo de pressões é utilizada a equação da continuidade modificada, como é descrito no Cap. 2.

A discretização das equações acima citadas é feita pelos seguintes métodos (descritos no Cap. 3): o termo temporal da Equação de Euler é discretizado pelo Método de Runge-Kutta de 4<sup>a</sup> ordem, sendo a solução obtida quando este termo tender a zero, e o termo espacial pelo Método dos Volumes Finitos.

Já para o campo de pressões, a discretização da equação da continuidade modificada é feita pelo Método da Compressibilidade Artificial, ou Pseudo-Compressibilidade, que considera a equação da continuidade para um escoamento compressível, cuja solução também será obtida quando o termo temporal tender a zero.

Como estes métodos utilizam o conceito de integrar as equações no tempo até que se chegue no regime permanente, os termos temporais são utilizados meramente como um dispositivo para gerar um esquema iterativo, podendo ser modificados a fim de aumentar ou controlar a taxa de convergência.

A utilização da Equação de Euler com os métodos numéricos só foi possível quando se desenvolveu métodos para

diminuir tanto a geração de grandes erros numéricos (p.ex. falsa vorticidade) como a excessiva morosidade na convergência da solução (geralmente divergindo da solução), que aconteciam devido à utilização dos métodos numéricos conhecidos para o escoamento potencial (Jameson, 1983).

O Método de Runge-Kutta e o da Pseudo-Compressibilidade são técnicas utilizadas para estabilizar a solução, tendo sido apresentadas por Jameson, et al. (1981), e desde então bastante aplicadas tanto para escoamento externo (para a qual foi utilizada pela primeira vez) quanto para interno, como escoamentos em turbo-máquinas.

A metodologia descrita pode ser utilizada na determinação dos campos de pressões e velocidades dos vários componentes de máquinas de fluxo (quando o escoamento for considerado como incompressível), tais como bombas centrífugas e turbinas hidráulicas para geração de energia elétrica; porém, nestes casos, considera-se que o ponto de operação destas máquinas seja o de projeto.

### **Referências**

ANDERSON, Dale A., TANNEHILL, John C. and PLETCHER, Richard H. "Computational Fluid Mechanics and Heat Transfer". New York: Hemisphere Publishing Corporation, 1984.

CHAPMAN, Dean R., MARK, Hans and PIRTLE, Melvin W. "Computers vs. Wind Tunnels". *Astronautics and Aeronautics*, v. 13, p. 22-35, April, 1975.

CHAPMAN, Dean R. "Computational Aerodynamics Development and Outlook". *AIAA Journal*, Washington, v. 17, n. 12, p. 1293-1313, Dec., 1979.

JAMESON, A., SCHMIDT, W., TURKEL, E. "Numerical Solution of the Euler Equations by Finite Volume Methods using Runge-Kutta Time Stepping Schemes". Proc. AIAA Paper 81-1259, **5 th Computational Fluid Dynamics Conference**, 1981.

---

## **FORMULAÇÃO MATEMÁTICA**

Para alguns escoamentos não há a necessidade de se utilizar as equações completas de Navier-Stokes, desde que o equacionamento com hipóteses simplificadoras descreva adequadamente o fenômeno físico: por exemplo, uma turbina hidráulica funcionando no ponto de máximo rendimento.

Com o desenvolvimento das técnicas numéricas e a evolução dos computadores em capacidade de processamento e armazenamento de dados, diminuiu-se cada vez mais as hipóteses simplificadoras no modelamento matemático que descreve o comportamento de um escoamento.

Nas condições de projeto de máquinas de fluxo, pode-se, sem perda de qualidade, usar as equações da Continuidade, que define a conservação de massa, e a da quantidade de movimento chamada de Equação Euler, que descreve um escoamento não viscoso, porém rotacional.

Esta formulação tem sido aplicada no projeto de máquinas de fluxo, como bombas centrífugas e turbinas hidráulicas, cujo ponto de operação é o de máximo rendimento. Alguns autores afirmam que esta formulação pode ser utilizada fora do ponto ótimo para rotores de turbinas hidráulicas, com bons resultados.

Goede, et al (1992), reporta que problemas como a separação do escoamento são provenientes principalmente da vorticidade e não tanto da viscosidade do fluido. Afirma ainda que no caso de bombas centrífugas o equacionamento acima não pode ser utilizado pois os efeitos de turbulência e de camada limite formada ao longo das pás tem grande influência sobre a região central do escoamento.

### 2.1. Equação da Continuidade

Considere um campo de escoamento representado pelas linhas de corrente conforme a fig. 2.1. Seja uma superfície de controle  $S$ , fechada, definindo o volume de controle  $V$  numa região finita do espaço.

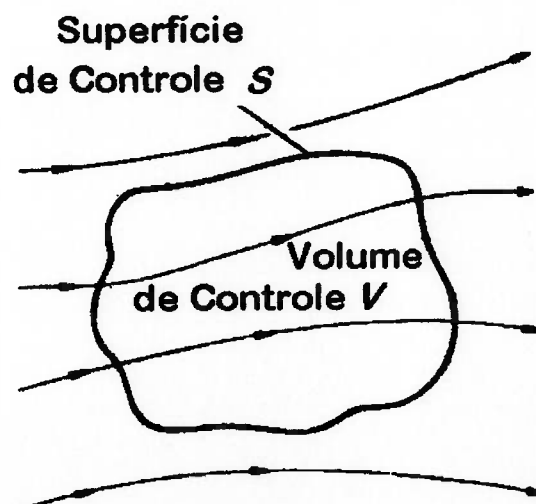


fig. 2.1: Volume de Controle Finito

Para este volume de controle que está fixo no espaço, a equação da continuidade na forma conservativa pode ser dada por:

$$\frac{\partial \rho}{\partial t} + \vec{\nabla} \cdot (\rho \vec{V}) = 0 \quad (2.1)$$

A eq. 2.1. pode ser reescrita na forma integral:

$$\frac{\partial}{\partial t} \iiint_{Vol} \rho dV + \iint_{Sup} \rho \vec{V} \cdot d\vec{S} = 0 \quad (2.2)$$

onde:

$\rho$  é a massa específica do fluido;

$dV$  representa um volume infinitesimal;

$\vec{V}$  é o vetor velocidade no sistema cartesiano;

$d\vec{S}$  é um elemento de área infinitesimal.

Se considerarmos que o regime seja permanente e o fluido incompressível, chega-se à seguinte expressão:

$$\iint_{Sup} \rho \vec{V} \cdot d\vec{S} = 0 \quad (2.3)$$

## 2.2. Equação de Euler

Esta equação pode ser obtida a partir da Equação de Navier-Stokes, desprezando-se as tensões de cisalhamento e os termos de condução de calor. Esta aproximação é válida para escoamentos com alto número de Reynolds, fora das regiões viscosas desenvolvidas nas proximidades das superfícies sólidas, onde os termos viscosos da equação de Navier-Stokes tornam-se desprezíveis face aos convectivos (vide Anexo I).

Com a hipótese acima, as equações diferenciais de derivadas parciais se reduzem de segunda para primeira ordem,



simplificando sensivelmente a formulação matemática em termos de complexidade.

A equação de Euler na forma conservativa, para um escoamento incompressível, dependente do tempo e em um sistema de referência cartesiano bi-dimensional, pode ser expressa para a variável conservativa  $w$  e as componentes do vetor de fluxo  $f$  e  $g$  por:

$$\frac{\partial w}{\partial t} + \frac{\partial f}{\partial x} + \frac{\partial g}{\partial y} = 0 \quad (2.4)$$

ou então:

$$\frac{\partial w}{\partial t} + \vec{\nabla} \cdot \vec{q} = 0 \quad (2.5)$$

A equação hiperbólica acima forma um sistema de equações diferenciais de derivadas parciais de primeira ordem, onde  $\vec{q}$  representa o vetor de fluxo.

A variável conservativa  $w$  é definida por:

$$w = \begin{vmatrix} \rho u \\ \rho v \end{vmatrix} \quad (2.6)$$

onde:

$u$  e  $v$  são as componentes cartesianas da velocidade.

As componentes  $f$  e  $g$  do vetor de fluxo  $\vec{q}$  são definidas por:

$$f = \begin{vmatrix} \rho u^2 + p \\ \rho uv \end{vmatrix} \quad g = \begin{vmatrix} \rho uv \\ \rho v^2 + p \end{vmatrix} \quad (2.7)$$

onde:

$p$  é a pressão.

Integrando-se a equação (2.5) para um volume de controle e aplicando o Teorema de Gauss, pode-se obter a equação de Euler na forma conservativa e integral:

$$\frac{\partial}{\partial t} \int_{Vol} w dV + \oint_{Sup} \vec{q} \cdot d\vec{S} = 0 \quad (2.8)$$

Analisando a equação acima, pode-se verificar que a variação da quantidade  $w$  por unidade de tempo, no volume  $V$  deve ser igual à contribuição dos fluxos através da superfície  $\vec{S}$ . O vetor do elemento de superfície deve estar direcionado para fora do volume de controle.

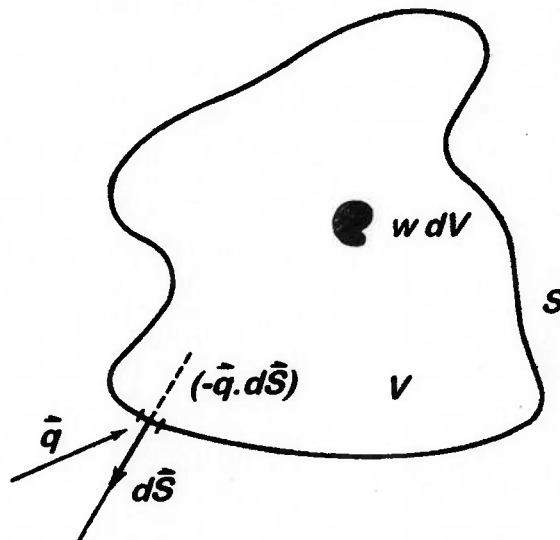


fig. 2.2: Forma Conservativa para uma Quantidade Escalar

### 2.3. Método da Compressibilidade Artificial

Segundo Lakshminarayana (1991), para escoamentos não viscosos e irrotacionais se tem utilizado eficientemente a equação de Poisson para obter o campo de pressão. Já para os

escoamentos rotacionais e incompressíveis o método não se mostrou eficiente, portanto, não recomendado.

Dentre os métodos para cálculo do campo de pressões tem se mostrado mais eficiente e promissor o da Compressibilidade Artificial ou Pseudo-Compressibilidade, introduzido independentemente, e de formas diferentes, por Vladimirova et al. (1965) e Chorin (1967).

O método consiste em se resolver um escoamento compressível e transitório, chegando-se ao regime permanente com o tempo de transitório tendendo ao infinito.

Definindo uma constante arbitrária  $c$ , com dimensão de velocidade, para um escoamento isoentrópico, temos:

$$\frac{\partial p}{\partial \rho} = c^2 \quad (2.9)$$

Substituindo (2.9) em (2.1), se obtém a equação da continuidade modificada:

$$\frac{\partial p}{\partial t} + \beta \vec{\nabla} \cdot \vec{V} = 0 \quad (2.10)$$

A equação (2.10) pode ser reescrita na forma integral:

$$\frac{\partial}{\partial t} \int_{Vol} p dV + \oint_{Sup} \beta \vec{V} \cdot d\vec{S} = 0 \quad (2.11)$$

onde:

$$\beta = \rho c^2;$$

e  $c$  corresponde a uma pseudo velocidade do som.

Deve-se lembrar que a equação (2.11) não possui qualquer significado físico enquanto a solução não convergir para o regime permanente, ou seja,  $\vec{\nabla} \cdot \vec{V} = 0$ .

## 2.4. Transformação de Coordenadas

Visto que a equação de Euler (2.8) e a equação para a pressão obtida pelo Método da Compressibilidade Artificial (2.11) possuem a mesma forma, pode-se reescrevê-las da seguinte maneira:

$$\frac{\partial}{\partial t} \int_{Vol} w \cdot dV + \oint_{Sup} \vec{q} \cdot d\vec{S} = 0 \quad (2.12)$$

onde:

a variável conservativa  $w$  é dada por:

$$w = \begin{vmatrix} p \\ \rho u \\ \rho v \end{vmatrix} \quad (2.13)$$

e as componentes do vetor de fluxo  $\vec{q}$ :

$$f = \begin{vmatrix} \beta u \\ \rho u^2 + p \\ \rho uv \end{vmatrix} \quad g = \begin{vmatrix} \beta v \\ \rho uv \\ \rho v^2 + p \end{vmatrix} \quad (2.14)$$

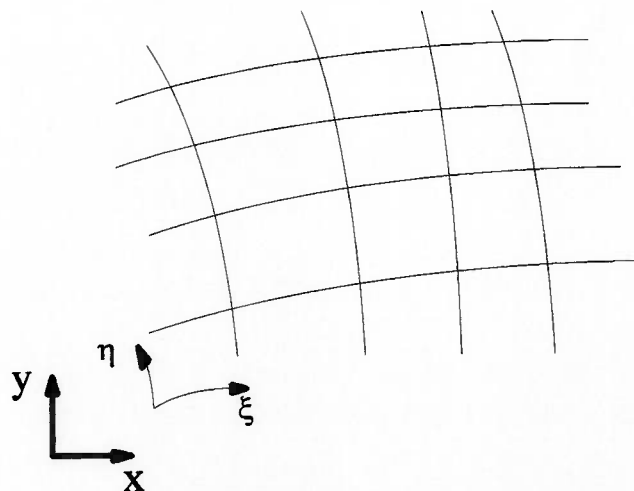


fig. 2.3: Sistema de Coordenadas

Na maior parte dos escoamentos (tanto interno quanto externo) a malha deve acompanhar as superfícies sólidas, criando um sério problema para sua obtenção em coordenadas retangulares.

Para sanar este inconveniente e facilitar a introdução das condições de contorno, será utilizada uma transformação das equações obtidas no sistema cartesiano para um curvilíneo não ortogonal (Anderson, et al., 1984).

A transformação das equações de um sistema cartesiano (2.12) para um curvilíneo (2.15) pode ser feita após algumas operações matemáticas (Vide Anexo II):

$$\frac{\partial}{\partial t} \int_{Vol} W \cdot dV + \oint_{Sup} \vec{Q} \cdot d\vec{S} = 0 \quad (2.15)$$

onde a variável primitiva  $W$  é dada por:

$$W = \begin{vmatrix} hp \\ h\rho u \\ h\rho v \end{vmatrix} \quad (2.16)$$

e as componentes  $F$  e  $G$  do vetor de fluxo  $\vec{Q}$ :

$$F = \begin{vmatrix} \beta h U \\ \rho h u U + y_{\eta} p \\ \rho h v U - x_{\eta} p \end{vmatrix} \quad G = \begin{vmatrix} \beta h V \\ \rho h u V - y_{\xi} p \\ \rho h v V + x_{\xi} p \end{vmatrix} \quad (2.17)$$

sendo:

$$\left. \begin{aligned} x_{\xi} &= \frac{\partial x}{\partial \xi} \\ x_{\eta} &= \frac{\partial x}{\partial \eta} \\ y_{\xi} &= \frac{\partial y}{\partial \xi} \\ y_{\eta} &= \frac{\partial y}{\partial \eta} \end{aligned} \right\} \text{coeficientes métricos de transformação; (2.18)}$$

$$h = x_{\zeta} y_{\eta} + x_{\eta} y_{\zeta} \} \text{jacobiano de transformação; (2.19)}$$

$$\left. \begin{aligned} U &= \frac{y_{\eta}}{h} u - \frac{x_{\eta}}{h} v \\ V &= -\frac{y_{\xi}}{h} u + \frac{x_{\xi}}{h} v \end{aligned} \right\} \text{contravariantes da velocidade. (2.20)}$$

As contravariantes da velocidade,  $U$  e  $V$ , representarão as velocidades no novo sistema de coordenadas, e não são necessariamente ortogonais.

### Referências

ANDERSON, Dale A., TANNEHILL, John C. and PLETCHER, Richard H. "Computational Fluid Mechanics and Heat Transfer". New York: Hemisphere Publishing Corporation, 1984.

- CHORIN, A. J. "A Numerical Method for Solving Incompressible Viscous Flow Problems". **Journal of Computational Physics**, v. 2, p. 12-26, 1967.
- GOEDE, E., SEBESTYEN, A. and SCHACHENMANN, A. "Navier-Stokes Flow Analysis for a Pump Impeller". **16th IAHR SYMPOSIUM, Section on Hydraulic Machinery and Cavitation**, São Paulo, p. 513-523, Sept., 1992.
- LAKSHMINARAYANA, B. "An Assessment of Computational Fluid Dynamic Techniques in the Analysis and Design of Turbomachinery - The 1990 Freeman Scholar Lecture". **Transactions of the ASME, Journal of Fluids Engineering**, New York, v. 113, n. 3, p. 315-352, Sept., 1991.
- PEYRET, Roger and TAYLOR, Thomas D. "**Computational Methods for Fluid Flow**". New York: Spring-Verlag, 1983.

## **DISCRETIZAÇÃO**

O Método dos Volumes Finitos tem tido grande aceitação devido à flexibilidade em se trabalhar com malhas irregulares, não exigindo sistemas ortogonais.

O método se utiliza do esquema de células centradas, onde as variáveis primitivas possuem valores médios para a célula em análise e podem ser consideradas como representativas de pontos interiores da célula (por exemplo, o ponto central).

Desta maneira, este método pode ser considerado como sendo o Método das Diferenças Finitas aplicado às equações diferenciais na forma conservativa e escritas em coordenadas arbitrárias, ou então como o Método dos Elementos Finitos com uma variação na formulação dos Resíduos Ponderados na forma integral (Hirsch, 1988).

Com relação à discretização apenas do termo temporal da equação (2.15), assegura-se que a solução seja a do regime



permanente quando esta parcela tender a zero. Com isto garante-se que a solução não dependerá do intervalo temporal utilizado.

### 3.1 Método dos Volumes Finitos

O Método dos Volumes Finitos foi apresentado independentemente por McDonald (1971) e MacCormack e Paullay (1972) no campo da Dinâmica dos Fluidos Computacional. Os trabalhos apresentados utilizavam a Equação de Euler dependente do tempo num sistema de referência bi-dimensional, e foi posteriormente estendido para escoamentos tri-dimensionais por Rizzi e Inouye (1973).

A característica mais atrativa da formulação do Método dos Volumes Finitos é que a solução resultante na forma conservativa das quantidades em análise, tais como massa e quantidade de movimento, é satisfeita para o grupo de volumes de controle e, logicamente, para o domínio de cálculo.

A idéia básica do Método conduz a uma interpretação direta do fenômeno físico. O domínio de cálculo é dividido em volumes de controle (células) que não se superpõem e que são rodeadas por pontos ao longo deste domínio.

O Método dos Volumes Finitos discretiza o termo espacial da equação (2.15), onde a integral de superfície no volume de controle é substituída por uma somatória ao longo da superfície de controle. A somatória destes termos de fluxo refere-se a todas as faces externas da célula de controle. Toma-se como exemplo a célula  $i,j$  mostrada na fig. 3.1 que possui um volume de controle  $V$  definido pela área ABCD e profundidade unitária:

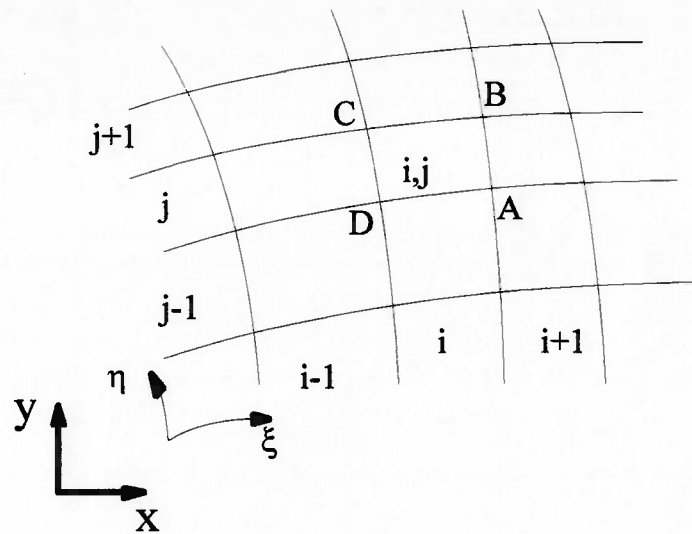


fig. 3.1: Célula de Controle  $i,j$

Os termos de fluxo serão somados através das quatro faces AB, BC, CD e DA da célula. Pode-se reescrever a equação (2.15) na forma discreta:

$$\frac{\partial}{\partial t} \int_{Vol} W \cdot dV + \sum_{Sup} \bar{Q} \cdot d\vec{S} = 0 \quad (3.1)$$

A aproximação feita pela substituição do termo  $\oint_{Sup} \bar{Q} \cdot d\vec{S}$  pela somatória  $\sum_{faces} \bar{Q} \cdot d\vec{S}$  pode ser feita obtendo-se os

fluxos para as quatro faces da célula:

$$\sum_{faces} \bar{Q} \cdot d\vec{S} = (\bar{Q} \cdot \vec{S})_{AB} + (\bar{Q} \cdot \vec{S})_{BC} + (\bar{Q} \cdot \vec{S})_{CD} + (\bar{Q} \cdot \vec{S})_{DA} \quad (3.2)$$

Deve-se notar que as coordenadas do ponto  $i,j$ , que é a localização precisa da variável  $W$  dentro do volume de controle  $V$ , não aparece explicitamente. Conseqüentemente,  $W_{i,j}$  não está necessariamente conectado a um ponto fixo dentro do volume de controle e pode ser considerado como um valor médio da variável  $W$  na célula de controle.

As coordenadas dos pontos da malha aparecem somente na determinação do volume da célula e da área da superfície de controle. Como a análise considerada é bi-dimensional, o volume da célula se reduzirá ao produto da área definida pelos vértices ABCD da célula pela profundidade unitária, enquanto a área da superfície de controle será o produto do comprimento da face da célula pela profundidade unitária.

Realizando o produto escalar do vetor de fluxo  $\vec{Q}$  pela área de cada face pode-se obter o termo espacial discretizado:

$$\sum_{\text{faces}} \vec{Q} \cdot d\vec{S} = (F \cdot \Delta\eta)_{AB} + (G \cdot \Delta\xi)_{BC} - (F \cdot \Delta\eta)_{CD} - (G \cdot \Delta\xi)_{DA} \quad (3.3)$$

A discretização das componentes do vetor de fluxo, por exemplo  $F_{AB}$  e  $G_{BC}$ , dependerá do esquema escolhido. Considerando a face AB e o esquema de célula centrada (variáveis primitivas  $p$ ,  $u$  e  $v$  no centro da célula de controle  $i,j$ ) pode-se realizar a discretização de três maneiras:

1. Calcular o fluxo para as células que possuem a face em comum para depois tomar a média:

$$F_{AB} = \frac{1}{2}(F_{i,j} + F_{i+1,j}) \quad (3.4)$$

2. Tomar a média das variáveis das células que possuem a face em comum para depois calcular o fluxo:

$$F_{AB} = F \left( \frac{W_{i,j} + W_{i+1,j}}{2} \right) \quad (3.5)$$

3. Calcular as variáveis primitivas nas extremidades das faces (vértices da célula) como sendo a média das células que as circundam; calcular o fluxo no vértice e depois calcular a média para obter o fluxo na face:

$$W_A = \frac{1}{4}(W_{i,j} + W_{i+1,j} + W_{i,j-1} + W_{i+1,j-1}) \quad (3.6)$$

$$F_A = F(W_A) \quad (3.7)$$

$$F_{AB} = \frac{1}{2}(F_A + F_B) \quad (3.8)$$

De maneira semelhante, pode-se aproximar as componentes do vetor de fluxo para todas as outras faces da célula. Dentre estas três opções, verifica-se que a primeira é a que demanda o menor esforço computacional e aparentemente o menor tempo de processamento (Anderson et al. - 1992), mas deve-se levar em consideração que este método iterativo é instável levando a oscilações durante o processamento, e que a opção que reduzir estes inconvenientes poderá conduzir a um número menor de iterações e a um tempo de processamento também menor.

### 3.2 Método de Runge-Kutta

O método explícito de Runge-Kutta para integração no tempo vem sendo utilizado para a solução da equação de Euler com grande sucesso, tendo sido aplicado inicialmente por Jameson et al. (1981). Códigos computacionais eficientes foram desenvolvidos com estes métodos, como os de Jameson e Baker (1983, 1984), Martinelli (1987), dentre outros.

A idéia básica do Método de Runge-Kutta é calcular diversos valores intermediários para a variável primitiva  $W$  nos intervalos  $n\Delta t$  e  $(n+1)\Delta t$  (onde o número da iteração  $n$  confunde-se com o tempo  $t$ ), e combiná-los a fim de obter uma aproximação para  $W^{n+1}$ .

Como a célula de controle  $i,j$  permanecerá constante ao longo do tempo, discretiza-se o termo de variação temporal por:

$$V \frac{\Delta W}{\Delta t} + \sum_{Sup} \bar{Q} \cdot d\bar{S} = 0 \quad (3.9)$$

ou melhor:

$$W^{n+1} = W^n - \frac{\Delta t}{V} \sum_{Sup} \bar{Q} \cdot d\bar{S} = 0 \quad (3.10)$$

Fazendo-se:

$$Pw = \sum_{Sup} \bar{Q} \cdot d\bar{S} \quad (3.11)$$

e tomando o Método de Runge-Kutta de quarta ordem:

$$\begin{aligned} W_{i,j}^0 &= W_{i,j}^n \\ W_{i,j}^1 &= W_{i,j}^0 - \frac{1}{2} \cdot \frac{\Delta t}{V_{i,j}} \cdot Pw^0 \\ W_{i,j}^2 &= W_{i,j}^0 - \frac{1}{2} \cdot \frac{\Delta t}{V_{i,j}} \cdot Pw^1 \\ W_{i,j}^3 &= W_{i,j}^0 - \frac{\Delta t}{V_{i,j}} \cdot Pw^2 \\ W_{i,j}^4 &= W_{i,j}^0 - \frac{\Delta t}{V_{i,j}} \cdot Pw^3 \\ W_{i,j}^{n+1} &= W_{i,j}^4 \end{aligned} \quad (3.12)$$

Substituindo-se a variável primitiva do primeiro, segundo e terceiro no quarto passo, pode-se então determinar a equação para a variável primitiva  $W$  na iteração  $n+1$ :

$$W_{i,j}^{n+1} = W_{i,j}^0 - \frac{\Delta t}{V_{i,j}} \left( \frac{Pw^0 + 2Pw^1 + 2Pw^2 + Pw^3}{6} \right) \quad (3.13)$$

Analisando a equação (3.9), verifica-se que a solução final dada pelo regime permanente será obtida quando a somatória dos fluxos nas faces da célula se anular, ou seja,  $Pw^n = 0$  e então  $W^1 = W^0$  e  $W^{n+1} = W^n$ .

Se a solução a ser obtida é a do regime permanente dada por  $Pw = 0$ , esta não dependerá do intervalo de tempo  $\Delta t$ . Isto permite um intervalo de tempo variável determinado pelo limite local do número de Courant (para cada célula) a ser usado para acelerar a convergência ao regime permanente sem alterar o próprio regime permanente.

A aproximação do método acima será de 4ª ordem no tempo e o fato de se utilizar quatro passos é para se melhorar a precisão e estender a região de estabilidade. A figura 3.2 mostra a região de estabilidade para o Método de Runge-Kutta com 1, 2, 3 e 4 passos, determinada pelo estudo da estabilidade.

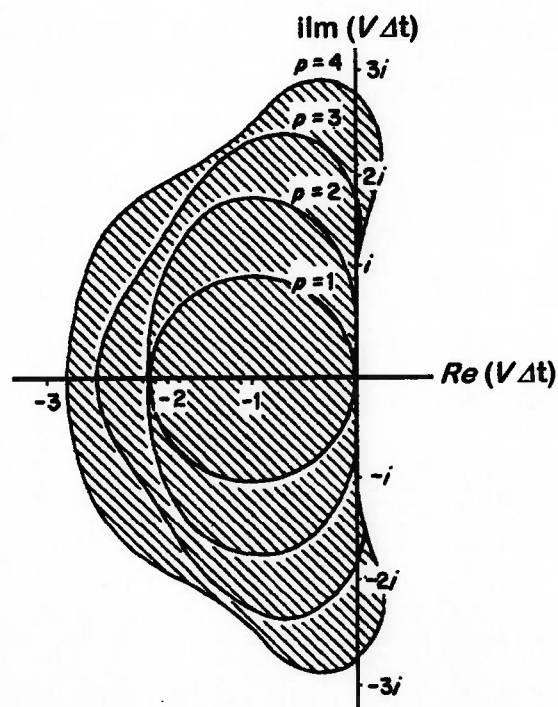


fig. 3.2: Região de Estabilidade para o Método de Runge-Kutta

A determinação do máximo intervalo temporal dependerá do número de Courant, ou CFL, que é a parte real do plano  $V\Delta t$ . A origem deste parâmetro reporta-se ao estudo realizado por Courant, Friedrichs e Lewy em 1928 (Courant et al. - 1967), que analisa a estabilidade da solução de equações discretas quando aplicadas na solução de equações de derivadas parciais.

O número de Courant para o qual a solução do método de Runge-Kutta de quarta ordem é estável é de  $2\sqrt{2}$  e a expressão que determina o máximo intervalo temporal para a região de estabilidade é dada por:

$$\left| c \frac{\Delta t}{\Delta x} \right| \leq 2\sqrt{2} \quad (3.14)$$

Desta maneira pode-se utilizar o máximo intervalo de tempo em função do espaçamento de cada célula, fazendo com que a solução venha a convergir o mais rápido possível para a solução final.

### 3.3 Dissipação Artificial

O método proposto por Jameson et al. (1981) faz uso do conceito de dissipação artificial que tem por objetivo evitar o desacoplamento das equações em células próximas a em análise (caracterizado pela oscilação das respostas), através da adição de um termo dissipativo ( $D$ ) à equação 3.9, estabilizando a solução (Caughey and Turkel, 1988), além de conseguir capturar choques (Jameson and Baker, 1984).

$$V \frac{\Delta W}{\Delta t} + \sum_{Sup} \bar{Q} \cdot d\bar{S} - D = 0 \quad (3.15)$$

O uso de células centradas no Método dos Volumes Finitos garante que o esquema tenha uma precisão de 2ª ordem, desde que a malha seja suficientemente regular sem mudanças

bruscas em sua forma e volume. Já o termo dissipativo introduzido por Jameson et al. (1981) possui precisão de 3ª ordem, o que não compromete a precisão de 2ª ordem do esquema (Jameson and Baker, 1984).

A construção do termo dissipativo é feita de maneira a se considerar a contribuição nas duas direções:

$$D = d_{i+1/2,j} - d_{i-1/2,j} + d_{i,j+1/2} - d_{i,j-1/2} \quad (3.16)$$

onde os termos a direita da igualdade da equação (3.16) podem ser calculados de maneira semelhante à abaixo:

$$d_{i+1/2,j} = \frac{V_{i,j}}{\Delta t} \left[ \varepsilon_{i+1/2,j}^{(2)} (W_{i+1,j} - W_{i,j}) - \varepsilon_{i+1/2,j}^{(4)} (W_{i+2,j} - 3W_{i+1,j} + 3W_{i,j} - W_{i-1,j}) \right] \quad (3.17)$$

Definindo-se:

$$v_{i,j} = \frac{|p_{i+1,j} - 2p_{i,j} + p_{i-1,j}|}{|p_{i+1,j} + 2p_{i,j} + p_{i-1,j}|} \quad (3.18)$$

Então:

$$\varepsilon_{i+1/2,j}^{(2)} = K^{(2)} \max(v_{i+1,j}, v_{i,j}) \quad (3.19)$$

$$\varepsilon_{i+1/2,j}^{(4)} = \max(0, (K^{(4)} - \varepsilon_{i+1/2,j}^{(2)})) \quad (3.20)$$

onde valores típicos para as constantes  $K^{(2)}$  e  $K^{(4)}$  encontrados na literatura são:

$$K^{(2)} = \frac{1}{4} \quad K^{(4)} = \frac{1}{256} \quad (3.21)$$



### 3.4 Condições de Contorno

O tratamento das condições de contorno é muito importante, pois pode conduzir a soluções que não representam o fenômeno em estudo, além de produzir instabilidades na solução. Este tratamento depende da natureza da equação que está sendo discretizada.

Neste caso, como a equação de Euler é hiperbólica e o escoamento é interno, a condição de contorno a ser utilizada será a de Cauchy e o problema é do tipo valor inicial (*"initial value problem"*).

Conforme Marcum e Hoffman (1987), para tratar as condições de contorno deve-se dividi-la em quatro tipos básicos: região interior ao escoamento, superfícies sólidas, região de entrada e de saída do escoamento, sendo que somente as três últimas merecem tratamento especial.

Seja:

$$A = \frac{\partial F}{\partial W} \quad \text{e} \quad B = \frac{\partial G}{\partial W} \quad (3.22)$$

Como a superfície sólida é representada pela coordenada  $\xi$  constante, a imposição das condições de contorno dependerá da análise dos autovalores de  $B$  (Jameson et al., 1981). Se houver autovalor positivo deve-se especificar na região de entrada a variável primitiva, enquanto que se for negativo, especifica-se a propriedade na região de saída (Clark et al. - 1986, Jameson and Yoon - 1987).

Os autovalores de  $B$  são  $V$ ,  $U$  e  $V-c$ , onde  $U$  e  $V$  são as contravariantes da velocidade e  $c$  a pseudo-velocidade do som. Se considerar-se que o fluido utilizado seja a água, dois autovalores serão positivos e um negativo, logo especifica-se na região de entrada a velocidade (através das contravariantes da velocidade) e

na região de saída a pressão. Portanto, deve-se extrapolar os valores de pressão na entrada e velocidade na saída (fig. 3.3).

Já sobre a superfície sólida utiliza-se a condição de escorregamento, ou seja, o vetor velocidade deve ser paralelo a parede. Esta condição é dada pela equação abaixo, onde  $\vec{V}$  é o vetor velocidade e  $\vec{n}$  o vetor normal a superfície:

$$\vec{V} \cdot \vec{n} = 0 \quad (3.23)$$

Esta condição é facilmente implementada devido a utilização do sistema de coordenadas curvilíneo. Basta impor à célula que possui o contorno sólido a condição de que a contravariante da velocidade  $V$  seja nula, tendo-se desta maneira somente a contravariante  $U$  da velocidade, que representa a velocidade na direção paralela à parede.

Mas pela análise do fluxo na célula, a mesma terá contribuição da pressão na face dada pela parede; logo tem-se que estimar a pressão sobre a superfície sólida. Pode-se obtê-la fazendo com que o gradiente de pressão em relação a normal da parede sólida seja igual a zero, ou seja:

$$\frac{\partial p}{\partial n} = 0 \quad (3.24)$$

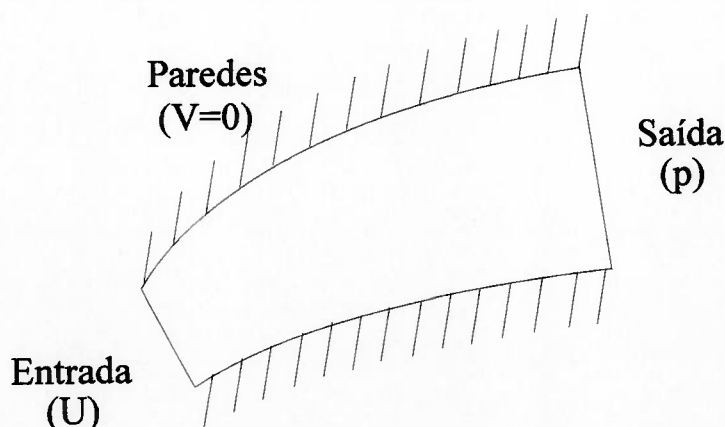


fig. 3.3: Condições de Contorno

## Referências

- ANDERSON Jr, J. D., DEGREGZ, G., DICK, E. and GRUNDMANN, R. "Computational Fluid Dynamics - An Introduction". Springer-Verlag, 1992.
- CAUGHEY, D. A. and TURKEL, E. "Effects of Numerical Dissipation of Finite Volume Solutions of Compressible Flow Problems". **Proc. AIAA Paper 88-0621, 26 th Aerospace Sciences Meeting**, p. 1-9, 1988.
- CLARKE, D. Keith, SALAS, M. D. and HASSAN, H. A. "Euler Equations for Multielement Airfoils Using Cartesian Grids". **AIAA Journal**, Washington, v. 24, n. 3, p. 353-358, Mar., 1986.
- COURANT, R., FRIEDRICHS, K. and LEWY, H. "On the Partial Difference Equations of Mathematical Physics". **IBM Journal Res. Dev.**, v. 11, p. 215-234, 1967.
- HIRSCH, Charles. "Numerical Computational of Internal and External flows". Volume 1: Fundamentals of Numerical Discretization. New York: Wiley-Interscience, 1988.
- JAMESON, A. and BAKER, T. J. "Solution of the Euler Equations for the Complex Configurations". **Proc. AIAA Paper 83-1929, 6 th Computational Fluid Dynamics Conference**, p. 293-302, 1983.
- JAMESON, A. and BAKER, T. J. "Multigrid Solution of the Euler Equations for Aircraft Configurations". **Proc. AIAA Paper 84-0093, 22nd Aerospace Sciences Meeting**, p. 1-11, 1984.

JAMESON, A., SCHMIDT, W., TURKEL, E. "Numerical Solution of the Euler Equations by Finite Volume Methods using Runge-Kutta Time Stepping Schemes". Proc. AIAA Paper 81-1259, **5 th Computational Fluid Dynamics Conference**, 1981.

JAMESON, Antony and YOON, Seokkwan. "Lower-Upper Implicit Schemes With Multiple Grids for the Euler Equations". **AIAA Journal**, Washington, v. 25, n. 7, p. 929-935, July., 1987.

MARCUM, David L. and HOFFMAN, Joe D. "Numerical Boundary Condition Procedures for Euler Solvers". **AIAA Journal**, Washington, v. 25, n. 8, p. 1054-1062, Aug., 1987.

---

## DESENVOLVIMENTO COMPUTACIONAL

O interesse deste trabalho está focado no processamento das informações, e não no desenvolvimento de pré/pós-processador. Porém, como é necessário a entrada e saída de informações, será utilizado como pré-processador um gerador de malha de um programa comercial de CAD ou de CAE, enquanto que para o pós-processamento dos resultados será gerado um arquivo formato texto a ser lido por uma planilha eletrônica.

Abaixo tem-se a relação das principais variáveis utilizadas no desenvolvimento computacional:

- $x, y$ : coordenadas cartesianas;
- $p$ : pressão;
- $u, v$ : velocidades no sistema cartesiano;

- ro*: massa específica;
- U, V*: contravariantes da velocidade;
- XQS*: coeficiente métrico da transformação ( $\partial x / \partial \xi$ );
- YQS*: coeficiente métrico da transformação ( $\partial y / \partial \xi$ );
- XET*: coeficiente métrico da transformação ( $\partial x / \partial \eta$ );
- YET*: coeficiente métrico da transformação ( $\partial y / \partial \eta$ );
- DQS*: variação em  $\xi$  ( $\Delta \xi$ );
- DET*: variação em  $\eta$  ( $\Delta \eta$ );
- H*: jacobiano da transformação;
- A*: área da célula;
- I, J*: número de pontos da malha na direção  $\xi$  e  $\eta$ , respectivamente.

O programa desenvolvido utilizando-se a linguagem C (Borland Turbo C Versão 2.0) foi subdividido em várias funções a fim de facilitar seu desenvolvimento. A fig. 4.1 apresenta um fluxograma geral do projeto denominado "VOLFIN", que mostra estas funções. Em seguida será descrito o conteúdo de cada uma destas funções.

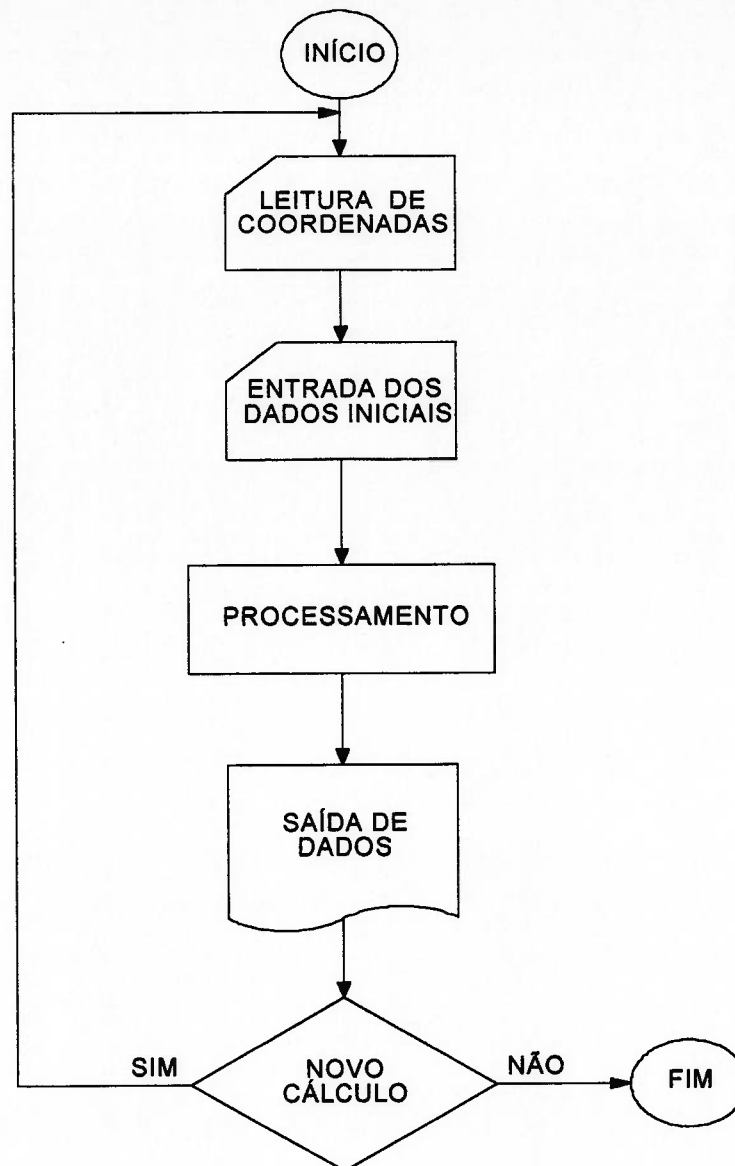


fig. 4.1: Fluxograma Geral

#### 4.1 Função “Leitura de Coordenadas”

A função “LEITURA” tem por objetivo fazer a leitura das coordenadas dos pontos que compõem a malha, além de calcular os coeficientes da transformação.

Para isto deve-se gerar um arquivo formato texto através de um programa gerador de malha de maneira a se ter na primeira linha o número de pontos na direção  $\xi$  (I) e na direção  $\eta$  (J), separados por um espaço em branco. Nas demais linhas do arquivo deve haver as coordenadas dos pontos num sistema cartesiano, sendo que cada linha deverá conter as coordenadas cartesianas  $x$  e  $y$  de um único ponto separadas por um espaço em branco.

A sequência dos pontos no arquivo deve ser dada na seguinte forma: mantém-se a coordenada  $\xi$  constante e varia-se a coordenada  $\eta$ . A fig. 4.1 mostra um exemplo:

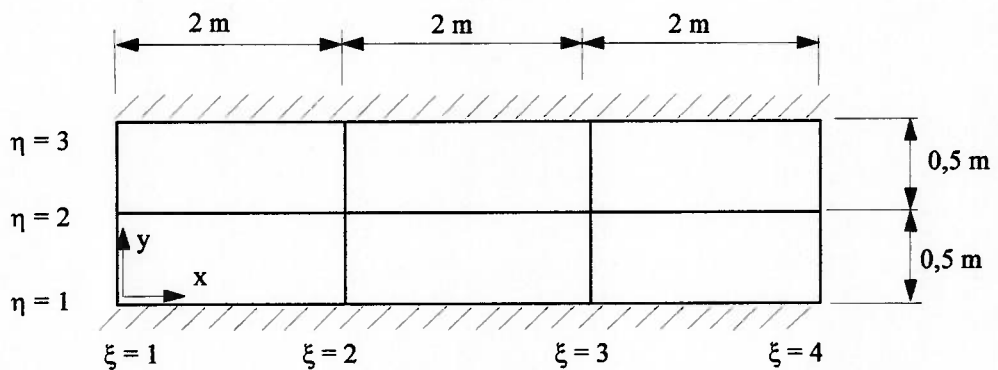


fig. 4.2: Malha Exemplo

O arquivo com as coordenadas da malha dada pela fig. 4.2 deverá conter as seguintes linhas:

```

4      3
0,0   0,0
    
```



0,0	0,5
0,0	1,0
2,0	0,0
2,0	0,5
2,0	1,0
4,0	0,0
4,0	0,5
4,0	1,0
6,0	0,0
6,0	0,5
6,0	1,0

tab. 4.1: Formato do Arquivo de Entrada

A função "LEITURA" irá ler a primeira linha do arquivo com o número de pontos da malha, fazendo então a alocação de memória necessária e, em seguida, a leitura das coordenadas. Após, irá calcular a área de cada célula e os coeficientes métricos da transformação.

Como o coeficiente métrico é dado em relação a célula de controle  $i,j$ , será considerado nos cálculos as coordenadas médias das faces da referida célula.

Considere a célula de controle  $i,j$  e seus vértices A, B, C, e D da figura 4.3. O coeficiente métrico  $\partial x / \partial \xi$  será aproximado por  $\Delta x / \Delta \xi$ , onde  $\Delta x$  da célula será  $x_{AB} - x_{CD}$ .

$$\Delta x = x_{AB} - x_{BC} = \left( \frac{x_A + x_B}{2} \right) - \left( \frac{x_B + x_C}{2} \right) \quad (4.1)$$

A coordenada  $x_{AB}$  da face AB será a média da coordenada  $X$  dos pontos A e B, enquanto que  $x_{BC}$  a média da coordenada de B e C. De maneira análoga determina-se a variação  $\Delta \xi$  e todos os outros coeficientes métricos.

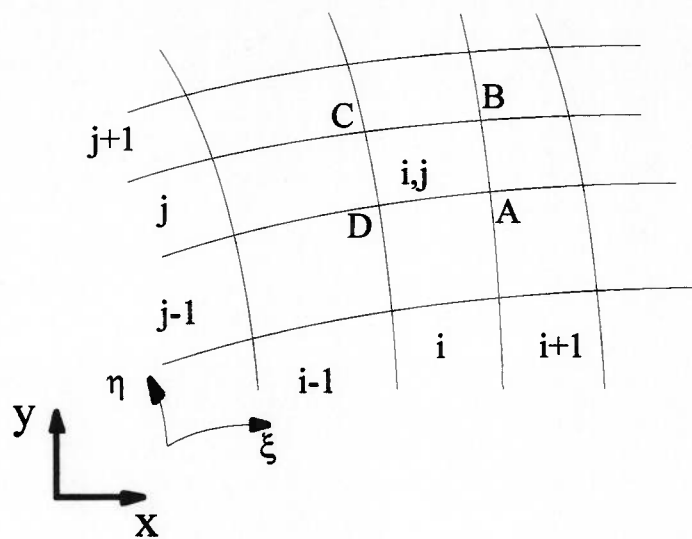


fig. 4.3: Célula de Controle  $i,j$

## 4.2 Função “Entrada de Dados”

A função entrada de dados faz com que se forneça os dados iniciais para poder se iniciar o processamento. Estes dados são a propriedade do fluido  $\rho_0$  (massa específica), as condições de contorno e iniciais.

As condições de contorno são: velocidade na entrada (na qual deve-se fornecer as contravariantes da velocidade no novo sistema) onde  $\xi$  será igual a 1 e  $\eta$  variável de 1 até o número de células nesta direção ( $J-1$ ), e a pressão na saída onde  $\xi$  será dado pela última carreira de células nesta direção ( $I-1$ ) e  $\eta$  variável de 1 até  $J-1$ .

Já para as condições iniciais de todas as células internas ao escoamento deve ser considerado iguais as de contorno, ou seja, velocidade em todas as células idênticas às da entrada e pressão idêntica à saída.

Desta forma inicia-se o processo iterativo suavemente, evitando-se as grandes variações na variável primitiva  $W$ . Se houver um forte gradiente da variável primitiva  $W$  o método se tornará mais lento na convergência, podendo até mesmo conduzir à divergência do processo.

### **4.3 Função "Processamento"**

A função se inicia calculando a vazão em volume na entrada do canal, obtida pela equação da continuidade dada por (2.3). Considerando-se que a carreira de células na direção  $\eta$  seja a secção transversal do canal, através da vazão em volume pode-se verificar e corrigir, se necessário, a velocidade nesta secção a fim de que se tenha a conservação de massa.

Na região de entrada a condição de contorno é dada pelas contra-variantes da velocidade ( $U$  e  $V$ ); porém, no

procedimento de cálculo utiliza-se a variável primitiva  $W$  (dada por 2.16). Como  $W$  depende da velocidade dada no sistema cartesiano, é necessário fazer a transformação das contra-variantes em velocidades no sistema cartesiano antes de se iniciar o método iterativo.

Assim, inicia-se o processamento do laço iterativo, que poderá terminar após um determinado número de iterações  $n$  ou quando se atingir a tolerância  $\varepsilon$  permitida, com valores já pré-estabelecidos. Esta tolerância  $\varepsilon$  será calculada, tanto para as velocidades quanto para a pressão.

Considerando a componente cartesiana  $u$  da velocidade, a tolerância é calculada da seguinte maneira:

a) Calcula-se a diferença relativa entre duas iterações consecutivas ( $n$  e  $n+1$ ) para a célula de controle  $i,j$ :

$$e_u = u^{n+1} - u^n \quad (4.2)$$

b) Calcula-se uma tolerância média, extraíndo-se a raiz quadrada da somatória do quadrado da diferença relativa para a célula de controle  $i,j$ :

$$\varepsilon_u = \frac{\sqrt{\sum_{i,j=1,1}^{I-1,J-1} e_u^2}}{(I-1).(J-1)} \quad (4.3)$$

A expressão (4.3) determina uma tolerância média relativa para toda a malha, servindo de referência para se verificar a taxa de convergência do método, mostrando se é divergente ou o quanto é convergente.

Quanto ao laço iterativo, inicia-se atribuindo o valor zero às variáveis de cálculo da tolerância média relativa. Na sequência é calculada a pressão na secção de entrada, pressão e velocidade para toda malha (exceto entrada e saída) incluindo-se a condição de contorno nas paredes (contra-variante  $V$  da velocidade nula), e por fim o cálculo da velocidade na saída.

Cabe ressaltar que para a célula de controle  $i,j$ , o cálculo da variável primitiva  $W$  na iteração  $n+1$  depende da variável primitiva  $W$  da iteração  $n$  (vide equação 3.13). Testes mostraram que o método é incondicionalmente instável se utilizar-se a variável primitiva  $W$  da iteração  $n$ .

Para resolver este problema basta calcular um valor médio de  $W$  da seguinte maneira:

$$W_{i,j}^n = \frac{W_{i+1,j} + W_{i-1,j} + W_{i,j+1} + W_{i,j-1}}{4} \quad (4.4)$$

Assim, deve-se tomar cuidado com as células que estão nas superfícies de fronteira da malha (dadas pelas condições de contorno). Considere a célula de controle  $i,j$  na secção de entrada dada pela fig. 4.4:

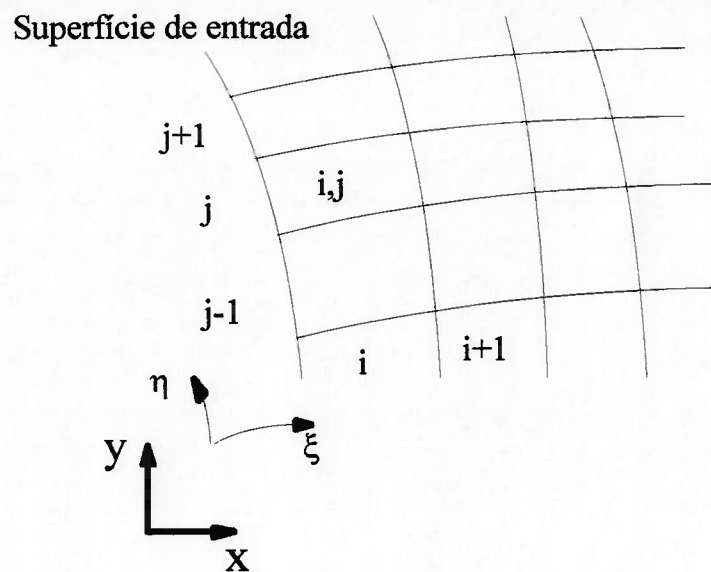


fig. 4.4: Seção de Entrada

Note-se que não há a carreira de células  $i-1$ . Para resolver este problema utiliza-se o conceito de falsas células (ou rebatimento), tomando como valor para a célula  $i-1,j$  o mesmo da célula  $i+1,j$ . Assim a equação 4.4 passa a ser reescrita da seguinte forma:

$$W_{i,j}^n = \frac{2.W_{i+1,j} + W_{i,j+1} + W_{i,j-1}}{4} \quad (4.5)$$

Para a superfície de saída e para a região das paredes utiliza-se o mesmo procedimento.

#### 4.4 Função "Saída de Dados"

A função "Saída de Dados" mostra na tela os resultados (pressão e componentes da velocidade no sistema cartesiano) de todas as células da malha, bem como gera um arquivo formato texto a ser lido por uma planilha eletrônica.

A disposição das informações no arquivo de saída seguirá variando  $i$  (número da célula na direção  $\xi$ ) de linha para linha, enquanto que numa mesma linha haverá a variação de  $j$  (número da célula na direção  $\eta$ ) para a pressão  $p$  e velocidades  $u$  e  $v$ , separados por um espaço em branco. A tabela abaixo mostra um exemplo de arquivo de saída para a malha da fig. 4.2.

```
p[1][1] u[1][1] v[1][1] p[1][2] u[1][2] v[1][2]
p[2][1] u[2][1] v[2][1] p[2][2] u[2][2] v[2][2]
p[3][1] u[3][1] v[3][1] p[3][2] u[3][2] v[3][2]
```

tab. 4.2: Formato do Arquivo de Saída

A listagem completa do projeto "VOLFIN" encontra-se no Anexo III.

#### Referências

- SCHILD, HERBERT. "Turbo C/C++ - The Complete Reference". Borland Osborne / McGraw-Hill, 1990.
- BORLAND. "Turbo C Version 2.0 - Reference Guide" Borland International, 1988.

## **APLICAÇÕES**

A metodologia será aplicada a três elementos diferentes. O primeiro elemento a ser aplicada a metodologia é um bocal convergente. O segundo é um tubo formando um ângulo de  $90^{\circ}$  entre as secções de entrada e saída, que foi desenvolvido com a velocidade sobre as paredes já prescritas a fim de não haver desaceleração ao longo das mesmas (Stanitz, 1952). O terceiro elemento é um rotor de bomba centrífuga, operando dentro das condições de projeto (Lauria, 1985).



## 5.1 Bocal

Para esta aplicação será feita uma verificação de sua taxa de convergência, tempo de processamento e resultados obtidos. Suas dimensões são mostradas na fig. 5.1, e como pode ser visto, sua malha possui 20 x 10 células (21 x 11 pontos na malha).

O fluido considerado no escoamento é ar que flui em regime permanente. As condições iniciais são velocidade na entrada de 20 m/s, enquanto que na saída a pressão é a atmosférica ( $p = 101325$  Pa). Nestas condições, o escoamento pode ser considerado como incompressível.

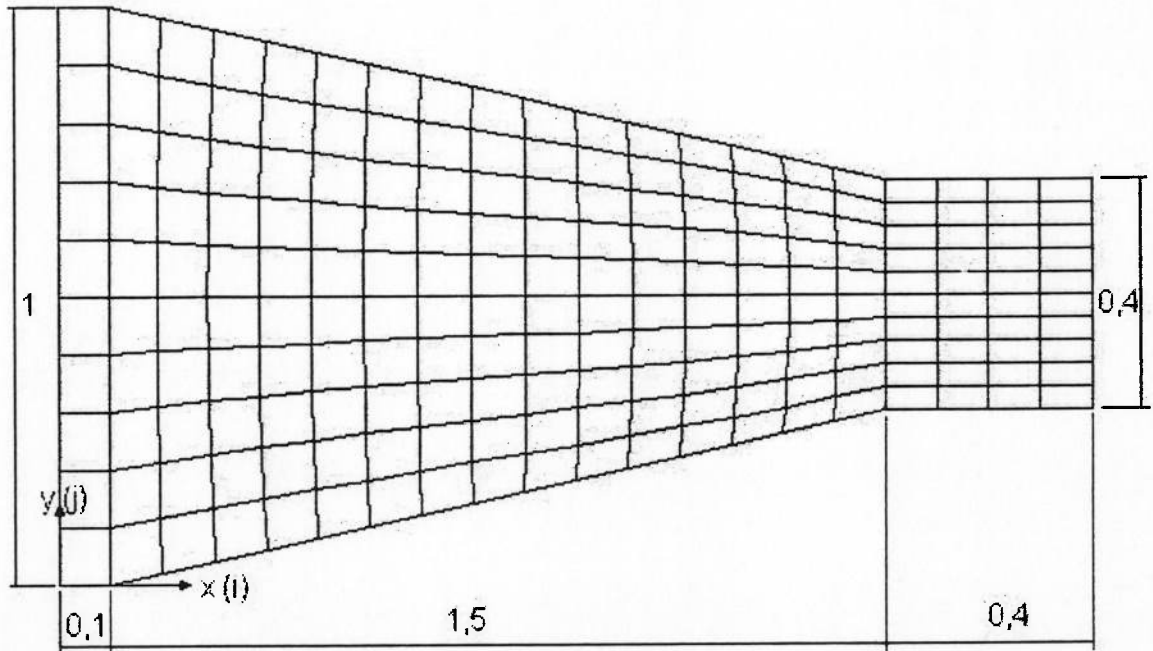


fig. 5.1: Malha para um Bocal (dimensões em m)

O arquivo de entrada com as coordenadas dos pontos da malha pode ser visto no Apêndice I.

Será considerado como valor inicial para toda malha, o mais desfavorável:  $U = V = 0$ , exceto nos contornos conforme já mencionado. Esta condição é a mais desfavorável pois na primeira iteração  $U$  é descontínuo, e o valor de  $\Delta W$  é elevado.

A convergência se mostra rápida quando as condições iniciais são "suaves", ou seja, quando o gradiente é pequeno entre a variável no contorno e na condição inicial. Também é rápida quando elas são próximas da solução final. Desta forma escolheu-se estes valores para demonstrar a convergência da metodologia sob condições desfavoráveis.

A fig. 5.2 apresenta o gráfico da tolerância média (conforme equações 4.2 e 4.3) ao longo das iterações para as condições já citadas. Verifica-se que o logaritmo da tolerância pode ser ajustado por uma função linear, o que era de se esperar para um método convergente.

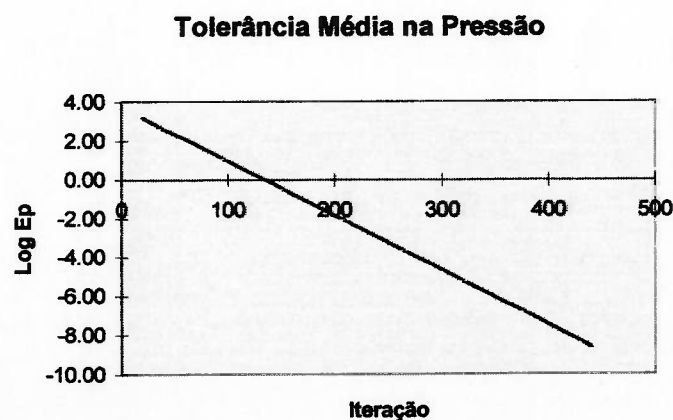


fig. 5.2: Log da Tolerância  $\epsilon_p$  x Iteração

A convergência se mostra rápida, já que em torno de 200 iterações tem-se uma diferença entre duas iterações consecutivas da ordem de grandeza de  $1. 10^{-2}$ .

O trabalho apresentado por Chorin (1967) tem como “problema teste” a análise de um escoamento num canal de paredes paralelas. As equações utilizadas foram as de Navier-Stokes, enquanto que o método de discretização o das diferenças finitas. Foi considerado o regime como permanente, e o número de iterações para se atingir a tolerância de  $1. 10^{-2}$  foi de 400.

Na fig. 5.3 tem-se as tolerâncias para as componentes da velocidade  $u$  e  $v$ . Pode-se notar que a convergência nestas componentes é mais rápida que na pressão.

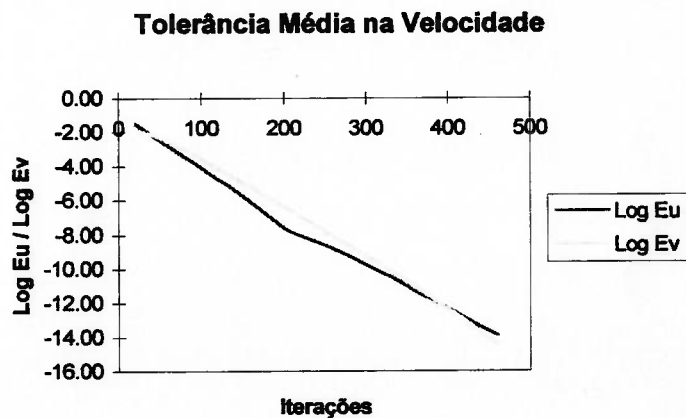


fig. 5.3: Log da Tolerância  $\epsilon_u$  ( $\epsilon_v$ ) x Iteração

Com relação ao tempo de processamento, utilizando-se um PC tipo pentium, com processador de 100 MHz e trabalhando sob o sistema operacional DOS, gastou-se 118 segundos para atingir 450 iterações, o que resulta num tempo de 0.262222 segundos para cada iteração.

Levando em conta que a malha possui 400 volumes, o tempo gasto num volume, em cada iteração, é de 0.656 milissegundos.

Os resultados podem ser vistos na tabela seguinte, na qual é o arquivo de saída gerado pelo aplicativo.

Número de células na direção i =							20
Número de células na direção j =							10
Número de iterações =							432
Tempo processam. =							112
		j=1			2		
	p[i][j]	u[i][j]	v[i][j]	p[i][j]	u[i][j]	v[i][j]	
i=1	102586.00	20.00	0.00	102538.00	20.00	0.00	
2	102581.00	17.31	3.32	102533.00	18.78	3.78	
3	102565.00	18.10	3.38	102516.00	19.37	4.41	
4	102530.00	19.70	3.59	102486.00	20.60	4.02	
5	102484.00	21.53	3.85	102446.00	22.07	3.38	
6	102432.00	23.37	4.12	102399.00	23.63	2.81	
7	102378.00	25.14	4.38	102348.00	25.20	2.41	
8	102323.00	26.84	4.65	102293.00	26.76	2.22	
9	102266.00	28.48	4.92	102234.00	28.33	2.25	
10	102208.00	30.07	5.21	102172.00	29.93	2.51	
11	102145.00	31.65	5.52	102104.00	31.61	2.97	
12	102076.00	33.30	5.87	102029.00	33.45	3.62	
13	101992.00	35.12	6.28	101941.00	35.56	4.41	
14	101882.00	37.35	6.82	101836.00	38.10	5.26	
15	101724.00	40.37	7.54	101712.00	41.33	5.92	
16	101542.00	44.82	8.61	101582.00	45.57	5.74	
17	101406.00	48.97	0.00	101473.00	49.06	2.29	
18	101338.00	49.26	0.00	101400.00	49.36	1.04	
19	101328.00	49.43	0.00	101357.00	49.51	0.57	
20	101325.00	49.49	0.00	101325.00	49.56	0.45	

3			4			5		
p[n]	u[n]	v[n]	p[n]	u[n]	v[n]	p[n]	u[n]	v[n]
102501.00	20.00	0.00	102476.00	20.00	0.00	102461.00	20.00	0.00
102496.00	20.18	3.25	102470.00	21.19	2.22	102455.00	21.73	0.94
102479.00	20.77	3.96	102454.00	21.85	2.78	102439.00	22.46	1.24
102452.00	21.50	3.52	102428.00	22.20	2.49	102412.00	22.63	1.18
102415.00	22.39	2.71	102392.00	22.61	1.88	102377.00	22.79	0.97
102371.00	23.45	1.91	102349.00	23.23	1.26	102333.00	23.18	0.73
102321.00	24.66	1.30	102299.00	24.13	0.76	102283.00	23.90	0.53
102266.00	26.02	0.95	102244.00	25.31	0.46	102227.00	24.99	0.40
102206.00	27.54	0.88	102183.00	26.80	0.37	102166.00	26.47	0.35
102142.00	29.24	1.08	102118.00	28.60	0.49	102099.00	28.33	0.38
102071.00	31.16	1.52	102046.00	30.74	0.80	102027.00	30.60	0.48
101994.00	33.36	2.17	101968.00	33.25	1.26	101949.00	33.30	0.64
101907.00	35.90	2.96	101883.00	36.17	1.81	101866.00	36.42	0.83
101810.00	38.88	3.75	101792.00	39.52	2.35	101778.00	39.98	1.01
101705.00	42.42	4.30	101698.00	43.34	2.72	101689.00	43.95	1.12
101599.00	46.64	4.16	101605.00	47.61	2.67	101603.00	48.26	1.08
101506.00	49.60	2.37	101520.00	50.18	1.71	101523.00	50.61	0.73
101432.00	49.68	1.32	101446.00	50.05	1.05	101450.00	50.34	0.48
101374.00	49.73	0.81	101383.00	49.99	0.70	101386.00	50.21	0.34
101325.00	49.75	0.67	101325.00	49.97	0.59	101325.00	50.17	0.30

6			7			8		
p[n]	u[n]	v[n]	p[n]	u[n]	v[n]	p[n]	u[n]	v[n]
102458.00	20.00	0.00	102466.00	20.00	0.00	102490.00	20.00	0.00
102451.00	21.78	-0.42	102459.00	21.37	-1.74	102479.00	20.60	-2.85
102434.00	22.57	-0.43	102439.00	22.23	-2.05	102455.00	21.59	-3.41
102406.00	22.80	-0.25	102408.00	22.75	-1.67	102419.00	22.61	-2.94
102369.00	23.00	-0.01	102368.00	23.29	-1.06	102375.00	23.70	-2.17
102324.00	23.42	0.19	102322.00	23.99	-0.48	102325.00	24.88	-1.43
102273.00	24.16	0.34	102269.00	24.94	-0.04	102270.00	26.16	-0.88
102216.00	25.26	0.42	102210.00	26.16	0.21	102210.00	27.56	-0.58
102154.00	26.74	0.43	102147.00	27.66	0.25	102146.00	29.10	-0.54
102086.00	28.61	0.37	102079.00	29.47	0.11	102078.00	30.82	-0.76
102013.00	30.89	0.24	102006.00	31.62	-0.22	102004.00	32.75	-1.21
101935.00	33.58	0.07	101927.00	34.13	-0.68	101924.00	34.95	-1.86
101852.00	36.70	-0.14	101842.00	37.03	-1.24	101837.00	37.46	-2.65
101766.00	40.25	-0.35	101754.00	40.35	-1.81	101743.00	40.37	-3.45
101679.00	44.20	-0.52	101665.00	44.09	-2.23	101647.00	43.76	-4.02
101595.00	48.46	-0.59	101580.00	48.25	-2.27	101557.00	47.73	-3.93
101517.00	50.78	-0.37	101504.00	50.68	-1.41	101482.00	50.39	-2.19
101447.00	50.48	-0.21	101437.00	50.45	-0.82	101419.00	50.29	-1.16
101384.00	50.33	-0.11	101379.00	50.33	-0.50	101369.00	50.24	-0.68
101325.00	50.27	-0.08	101325.00	50.29	-0.41	101325.00	50.23	-0.54

9			10		
$p[i]$	$u[i]$	$v[i]$	$p[i]$	$u[i]$	$v[i]$
102535.00	20.00	0.00	102624.00	20.00	0.00
102513.00	19.77	-3.60	102558.00	19.66	-3.78
102482.00	20.99	-4.22	102519.00	21.18	-3.96
102439.00	22.60	-3.88	102467.00	23.14	-4.22
102389.00	24.30	-3.31	102407.00	25.15	-4.50
102333.00	25.97	-2.80	102345.00	27.06	-4.77
102275.00	27.60	-2.45	102283.00	28.86	-5.03
102213.00	29.20	-2.29	102220.00	30.56	-5.29
102149.00	30.79	-2.35	102156.00	32.20	-5.56
102081.00	32.41	-2.61	102090.00	33.80	-5.85
102008.00	34.11	-3.07	102020.00	35.40	-6.17
101928.00	35.96	-3.72	101941.00	37.07	-6.53
101837.00	38.05	-4.51	101850.00	38.88	-6.96
101735.00	40.48	-5.34	101736.00	40.98	-7.48
101622.00	43.44	-5.98	101587.00	43.60	-8.15
101521.00	47.16	-5.78	101455.00	47.00	-9.03
101448.00	50.03	-2.24	101400.00	49.71	0.00
101388.00	50.10	-0.97	101332.00	49.99	0.00
101352.00	50.14	-0.51	101326.00	50.09	0.00
101325.00	50.15	-0.39	101325.00	50.12	0.00

tab. 5.2: Arquivo de Resultados para o Bocal

Tomada as linhas de corrente correspondentes ao centro das células de  $j$  constante, por exemplo  $j = 2, 5$  e  $9$ , pode-se aplicar às mesmas a equação de Bernoulli.

Para fluido incompressível em escoamento permanente e não viscoso sabe-se que:

$$\frac{Vel^2}{2} + \frac{p}{\rho} = constante \quad (5.1)$$

A figura 5.4 mostra o valor da constante de Bernoulli para as carreiras de células  $j = 2, 5$  e  $9$  ao longo de  $i$ . Verifica-se

que este valor se mantém praticamente constante, sendo válida então a equação (5.1).

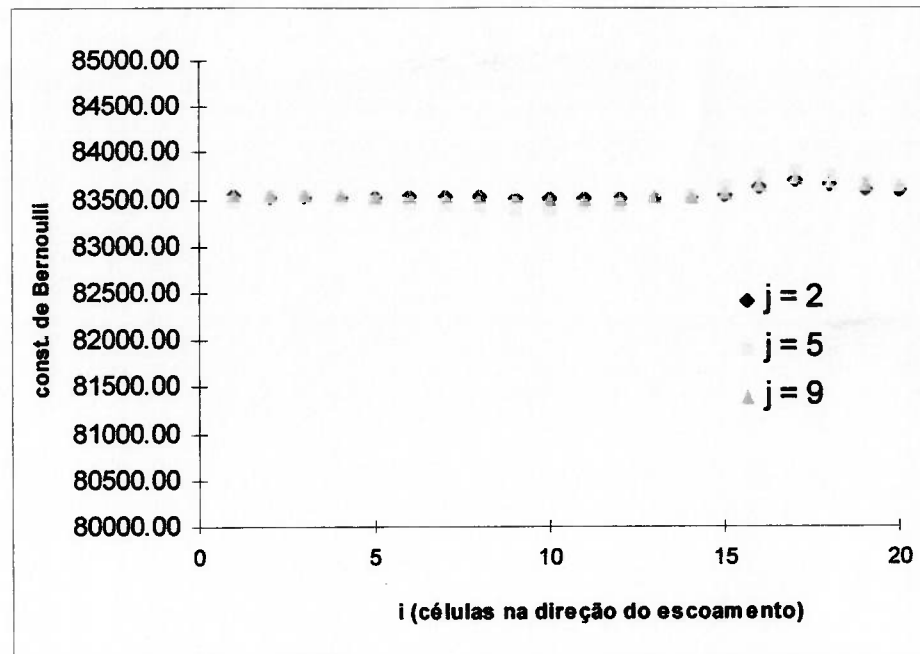


fig. 5.4: Constante de Bernoulli

Na fig. 5.5 pode-se verificar que nas células de  $j$  constante, o escoamento é acelerado, o que era de se esperar devido a redução de área.

Como esta redução se dá por uma função linear (parede dada por uma linha reta), o acréscimo da velocidade também deve ser linear. No caso, as células próximas a parede possuem uma função que melhor se ajustam a função linear.

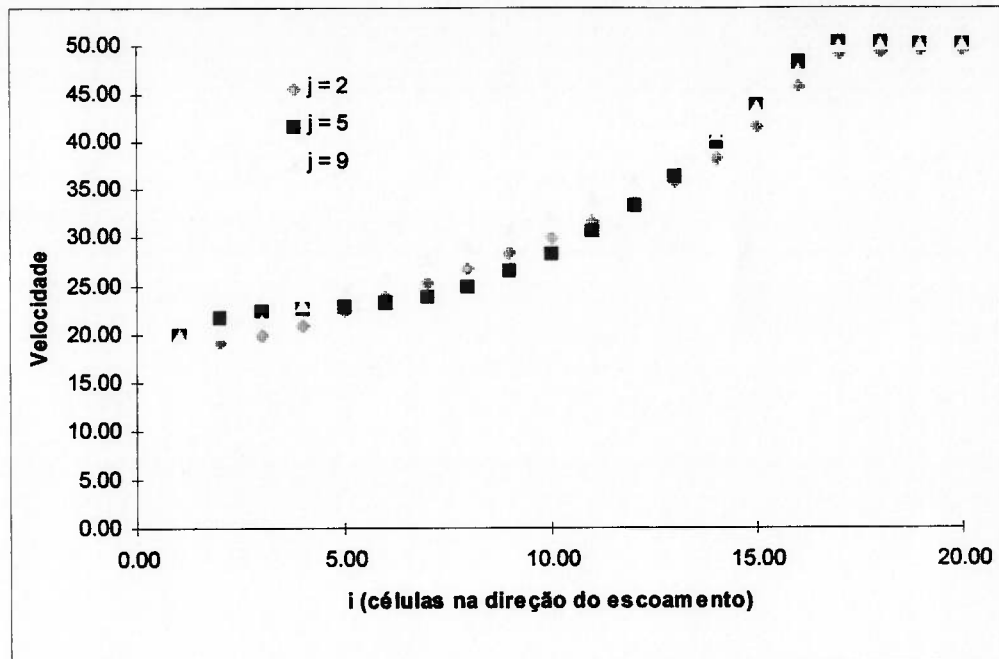


fig. 5.5: Velocidade ao longo das linhas de j constante

## 5.2 Tubo de Stanitz

J. D. Stanitz, em seu trabalho publicado em 1951, propôs um método de projeto para canais bi-dimensionais, sem ramificações, através da especificação de velocidades como uma função do comprimento de arco ao longo das paredes do canal. O escoamento considerado foi o potencial.

Em seu terceiro exemplo, Stanitz considera um escoamento incompressível, irrotacional e não viscoso. Prescreve uma velocidade de 0,5 sobre as paredes na entrada do canal e de 1,0 sobre as paredes na saída. Já as velocidades ao longo das paredes foram impostas de maneira a não haver desaceleração. Assim chegou-se a um tubo curvado de um ângulo de  $89.37^{\circ}$  entre a entrada e a saída (fig. 5.6).



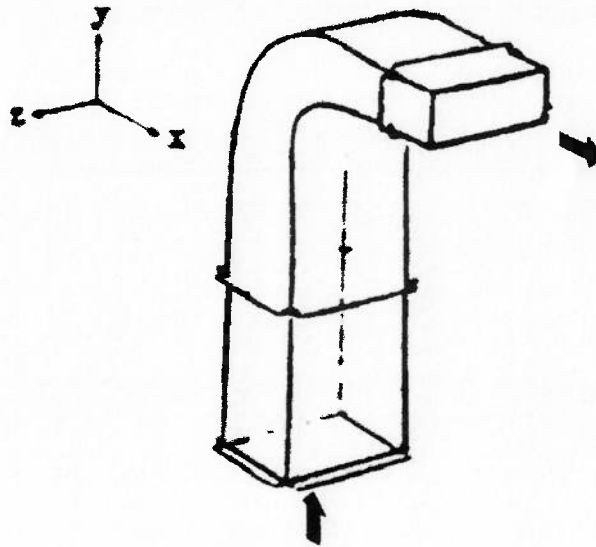


fig. 5.6: Tubo de Stanitz

Em 1952 J. J. Kramer e J. D. Stanitz, a fim de entender melhor o escoamento real em máquinas de fluxo, publicaram um estudo sobre escoamento no tubo citado, considerando-o como bi-dimensional, não viscoso, incompressível, porém rotacional.

Assim, será aplicada a metodologia proposta neste trabalho ao tubo da fig. 5.6 dentro das mesmas condições de Kramer e Stanitz (1952).

Serão realizadas 4 simulações com o número de células na direção do escoamento igual a 23 e com 5, 10, 15 e 20 células na direção normal ao escoamento.

A condição de contorno utilizada foi uma distribuição de velocidade linear na entrada, enquanto que na saída foi especificada a pressão obtida no trabalho de Kramer e Stanitz.

A velocidade especificada na entrada sobre a superfície interna (de menor raio) foi de 0,25, enquanto que sobre a superfície externa de 0,75. Já a pressão utilizada na saída foi de 21.96. Estes valores de velocidade e pressão foram adimensionalizados após os cálculos, conforme Kramer e Stanitz (1952).

Na fig. 5.7, pode-se ver a malha da primeira simulação, com 23x5 células junto com as condições de contorno.

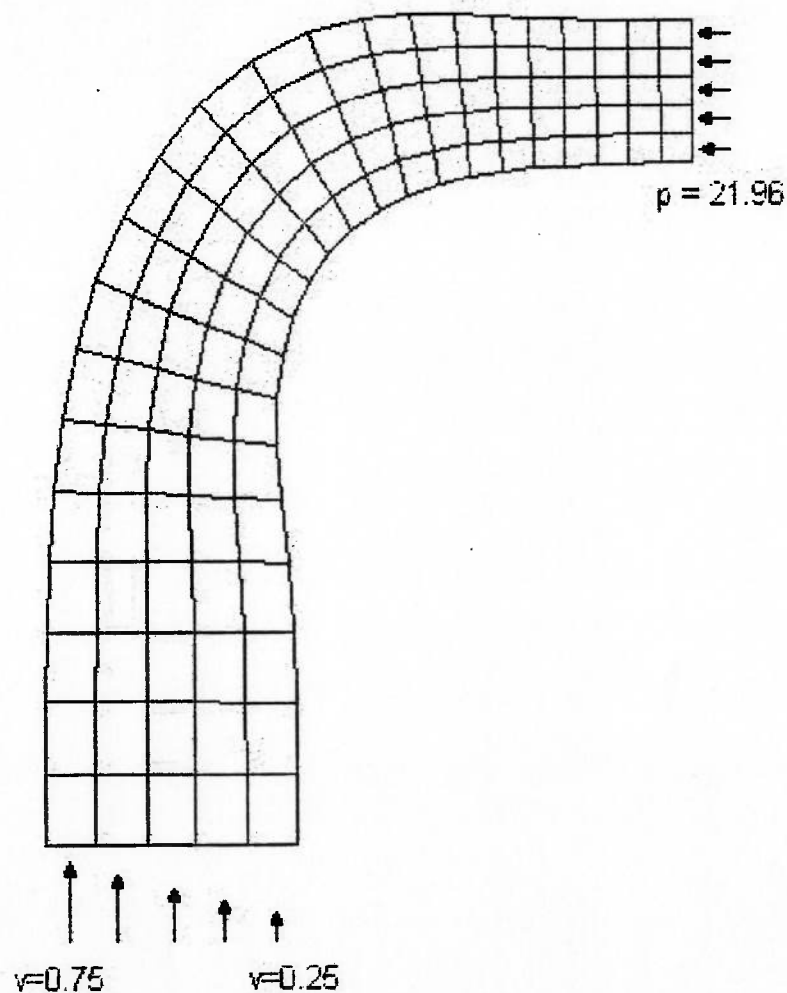


fig. 5.7: Malha com 23 x 5 células e condições de contorno

Na fig. 5.8 pode ser visto a distribuição de velocidade sobre as paredes interna ( $j=1$ ) e externa ( $j=5$ ) obtida por Kramer e Stanitz.

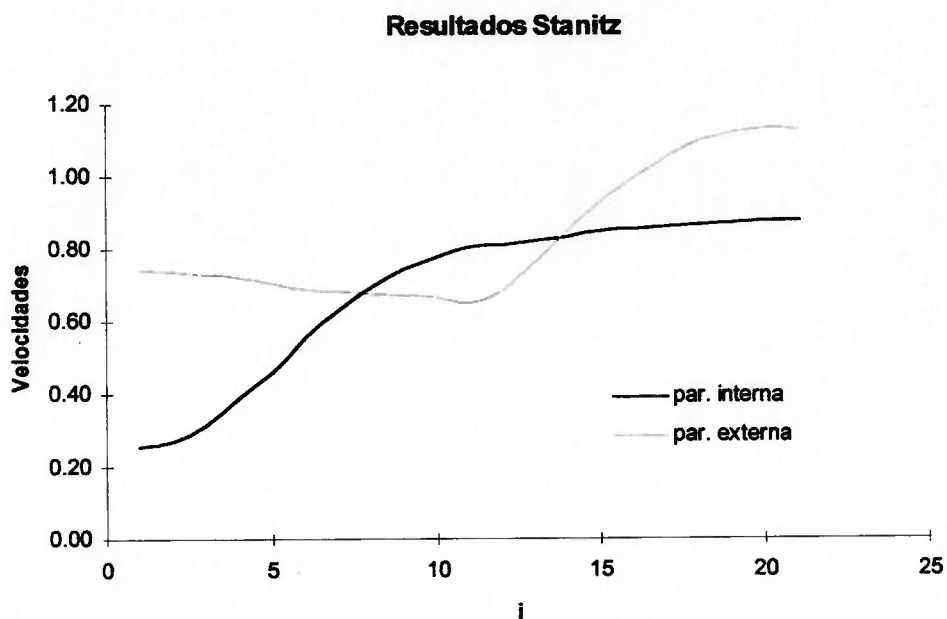


Fig. 5.8: Velocidades obtidas por Kramer e Stanitz

Já na fig. 5.9 pode-se ver os resultados obtidos na aplicação da metodologia proposta para a malha da fig. 5.7, com 23 x 5 células.

Pela análise deste gráfico, verifica-se que a velocidade tem seu valor praticamente constante logo após a entrada na secção transversal do tubo.

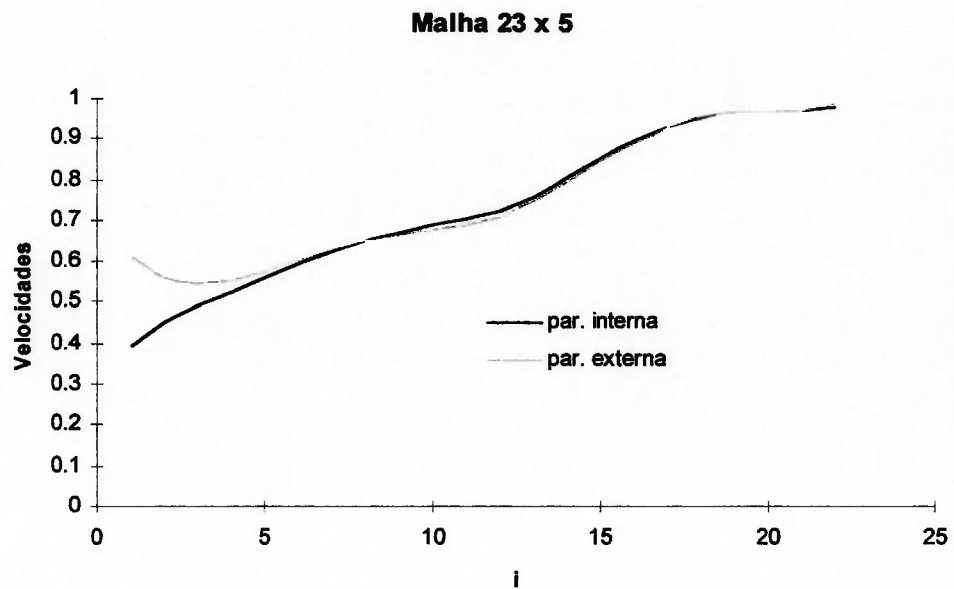


fig. 5.9: Velocidades sobre as paredes (23 x 5 células)

As fig. 5.10, 5.11 e 5.12 apresentam a distribuição de velocidades sobre as paredes para malhas com 23x10, 23x15 e 23x20 células, respectivamente.

Conforme se aumenta o número de células na direção normal ao escoamento, verifica-se que a velocidade sobre a parede externa sofre alteração significativa, no sentido de se aproximar dos valores obtidos por Kramer/Stanitz,.

A velocidade sobre esta parede tem seu valor levemente decrescente até a secção dada pela carreira de células de  $i = 12$ , onde passa a ser crescente (fig. 5.12).

Já sobre a parede interna não há alteração significativa nos valores da velocidade com o aumento de número de células na direção normal ao escoamento.

Verifica-se também que na região onde se tem a curvatura, os valores de velocidade sobre as paredes, que deveriam ter seus valores invertidos (velocidade maior sobre a parede interna) praticamente se mantêm iguais.

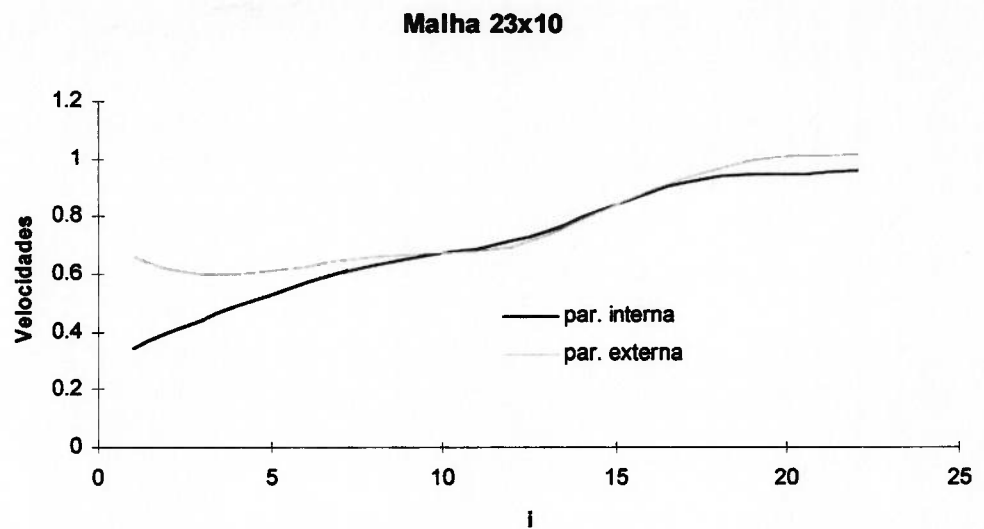


fig. 5.10: Velocidades sobre as paredes (23 x 10 células)

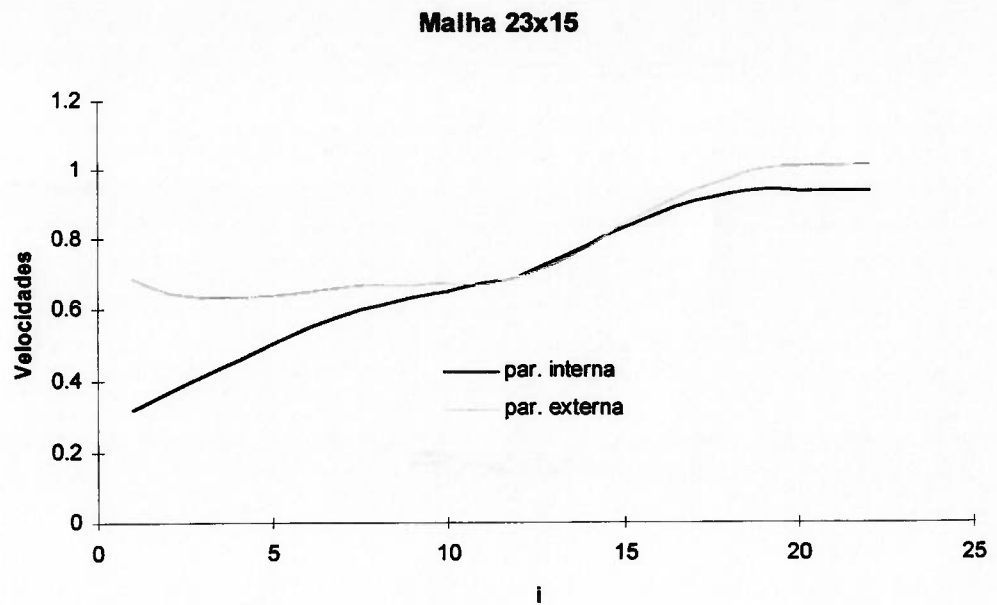


fig. 5.11: Velocidades sobre as paredes (23 x 15 células)

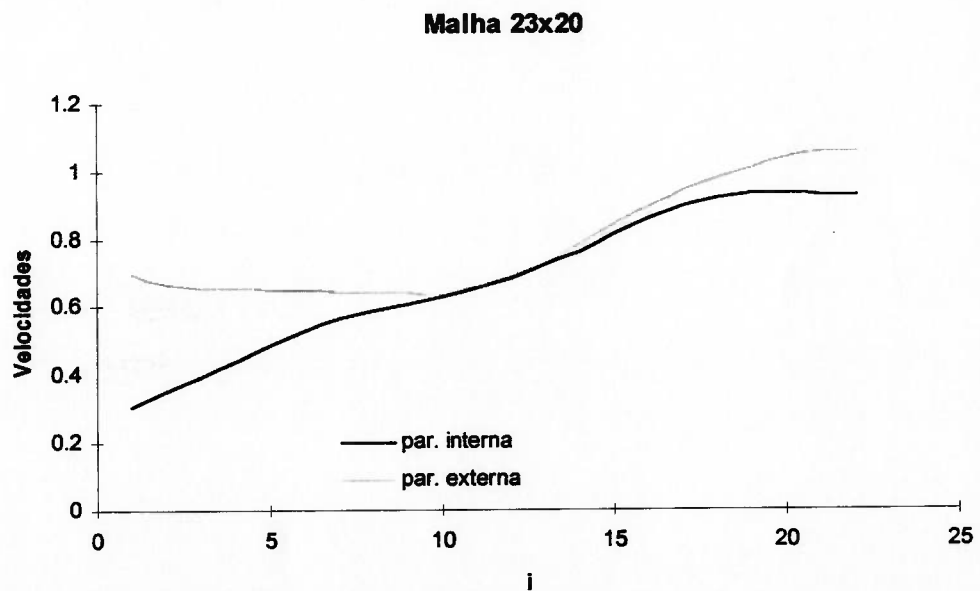


fig. 5.12: Velocidades sobre as paredes (23 x 20 células)

Na figura 5.13 pode-se ver a distribuição de pressão sobre as paredes interna e externa obtida por Kramer e Stanitz, enquanto que nas fig's 5.14 e 5.15, os resultados obtidos pela aplicação da metodologia proposta para malhas com 23x5 e 23x20 células.

Estes gráficos foram traçados tomando a diferença entre a pressão adimensionalizada no ponto considerado e a pressão na entrada.

Como a pressão está acoplada a velocidade, era de se esperar que houvesse uma discrepância entre os resultados obtidos e os de Kramer/Stanitz. Porém, o aumento do número de células na direção normal ao escoamento conduz à uniformização da pressão na secção de entrada.

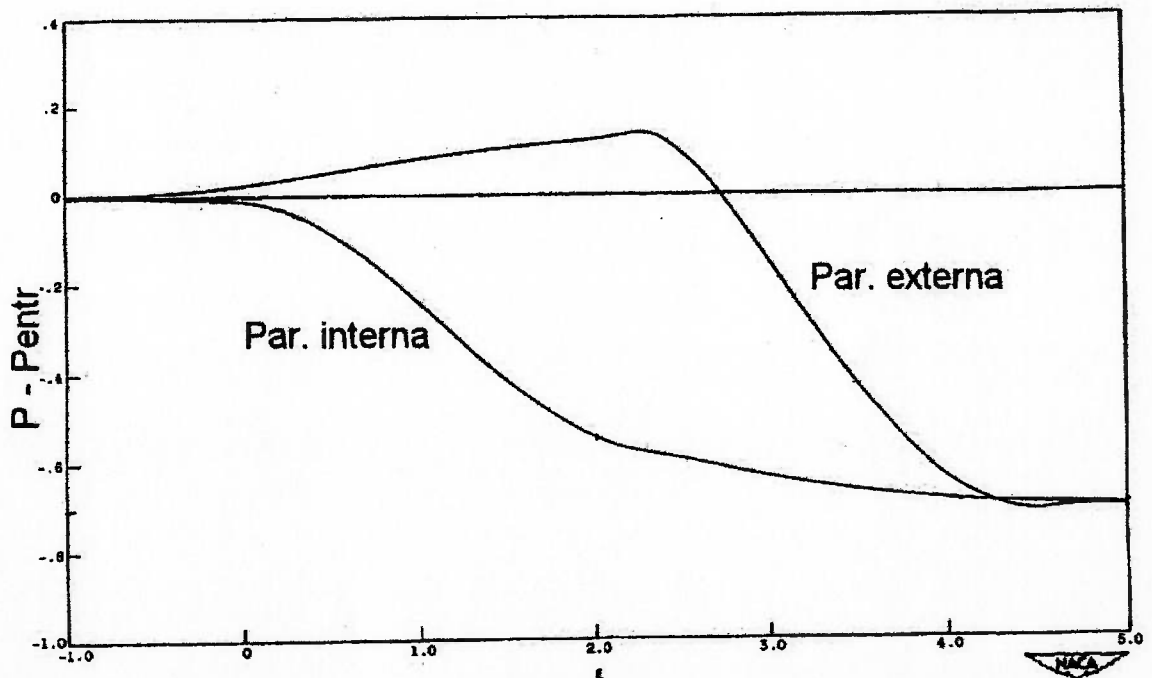


Fig. 5.13: Distribuição de Pressão (Kramer/Stanitz)

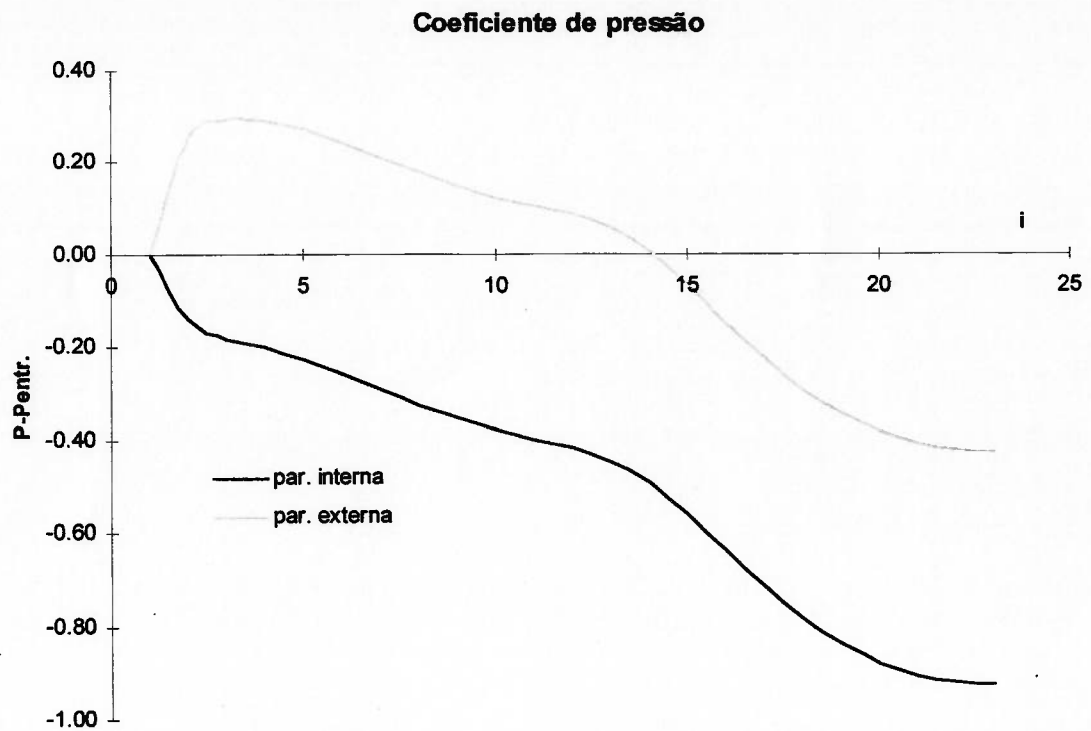


Fig. 5.14: Distribuição de Pressão (23x5 células)

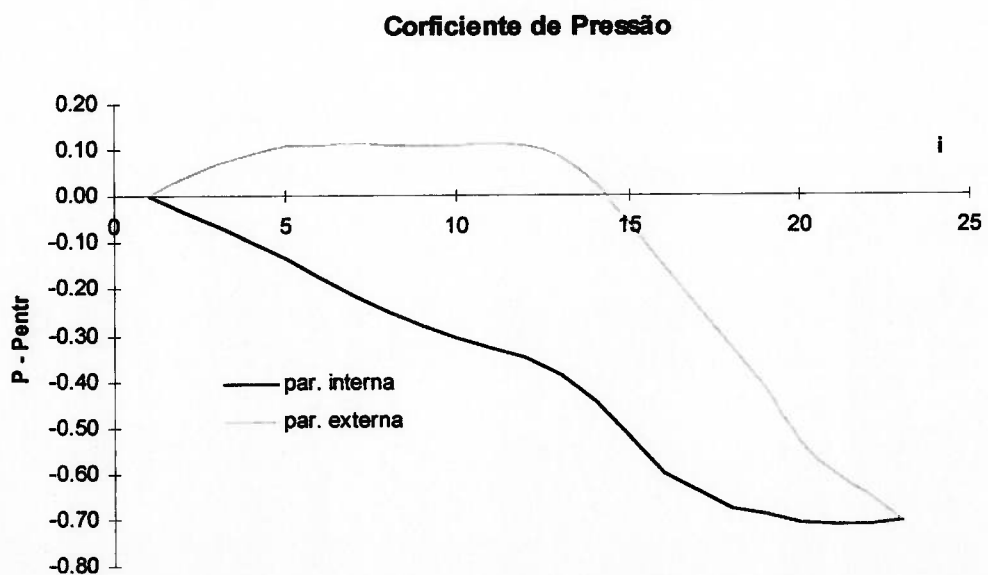


Fig. 5.15: Distribuição de Pressão (23x20 células)



Em face dos resultados obtidos, torna-se necessário refinar ainda mais a malha, devendo-se aumentar também o número de células na direção do escoamento.

Foi realizada uma simulação com uma malha de 23x22 células, e os resultados praticamente não diferiram com os da malha de 23x20. Porém, ao tentar utilizar uma malha com 23x23 células, o sistema operacional se tornou instável, perdendo-se o controle sobre a máquina, e não havendo a possibilidade de terminar esta simulação.

Isto aconteceu pois, para se agilizar as iterações, foi utilizado no cálculo da área das células, dos coeficientes métricos e do jacobiano da transformação, variáveis com espaço de memória já reservado, evitando-se o cálculo destas grandezas a cada nova iteração. Assim, sob o sistema operacional DOS, não foi possível realizar esta simulação devido a problemas no gerenciamento de memória.

### **5.3 Rotor de Bomba Centrífuga**

Nesta terceira análise procurou-se verificar, além da influência da curvatura no escoamento, os efeitos do movimento de rotação. Para isto, deve-se acrescentar às equações os termos provenientes deste efeito, que são as forças de coriolis e centrífuga.

A equação 2.15 pode ser reescrita acrescentando-se estes termos:

$$\frac{\partial}{\partial t} \int_{Vol} W \cdot dV + \oint_{Sup} \vec{Q} \cdot d\vec{S} = \int \rho \cdot \vec{f}_e \cdot dV \quad (5.2)$$

onde  $\vec{f}_e$  são as forças por unidade de massa, definidas por:

$$\vec{f}_e = \vec{f}_{cor} + \vec{f}_{centr} = -2\vec{\Omega} \times \vec{V} + \Omega^2 \vec{R} \quad (5.3)$$

sendo:

$\vec{\Omega}$ - vetor velocidade angular;

$\vec{V}$  - vetor velocidade;

$\vec{R}$  - vetor posição.

O rotor a ser analisado já foi objeto de estudo de Lauria (1985). Em sua análise, utilizou-se as equações de Navier-Stokes, com o método  $k-\varepsilon$ .

O plano meridiano do rotor que será feita a simulação pode ser visto na fig. 5.16. Esta máquina tem como condição de projeto:

Altura de recalque:	10 metros;
Vazão em volume:	0,10 m <sup>3</sup> /s;
Rotação do rotor:	500 rpm;
Rotação específica ( $n_q$ ):	29;
Número de pás:	7;
Espessura das pás:	6,0 mm.

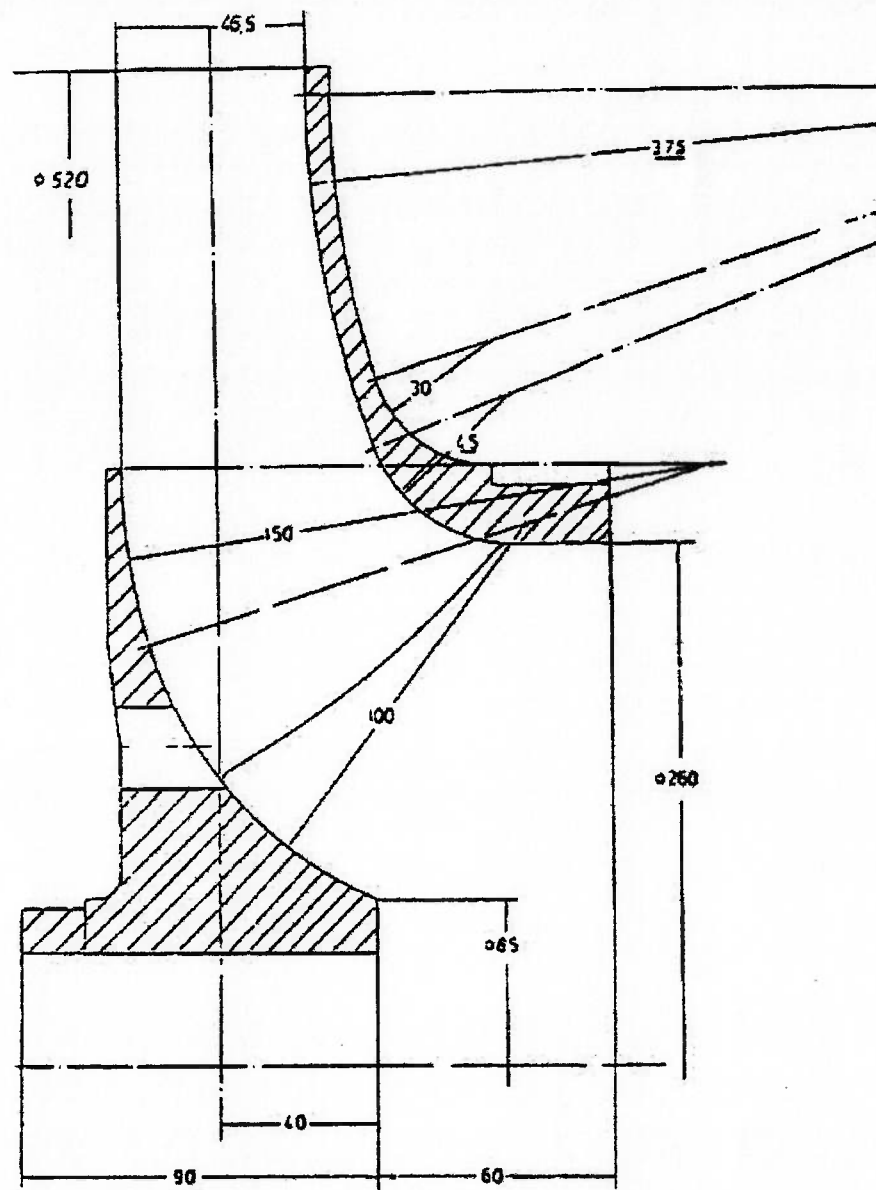


Fig. 5.16: Plano meridiano do rotor

A malha a ser utilizada possui 24 células na direção do escoamento e 10 na direção normal ao escoamento, e pode ser vista na fig. 5.17.

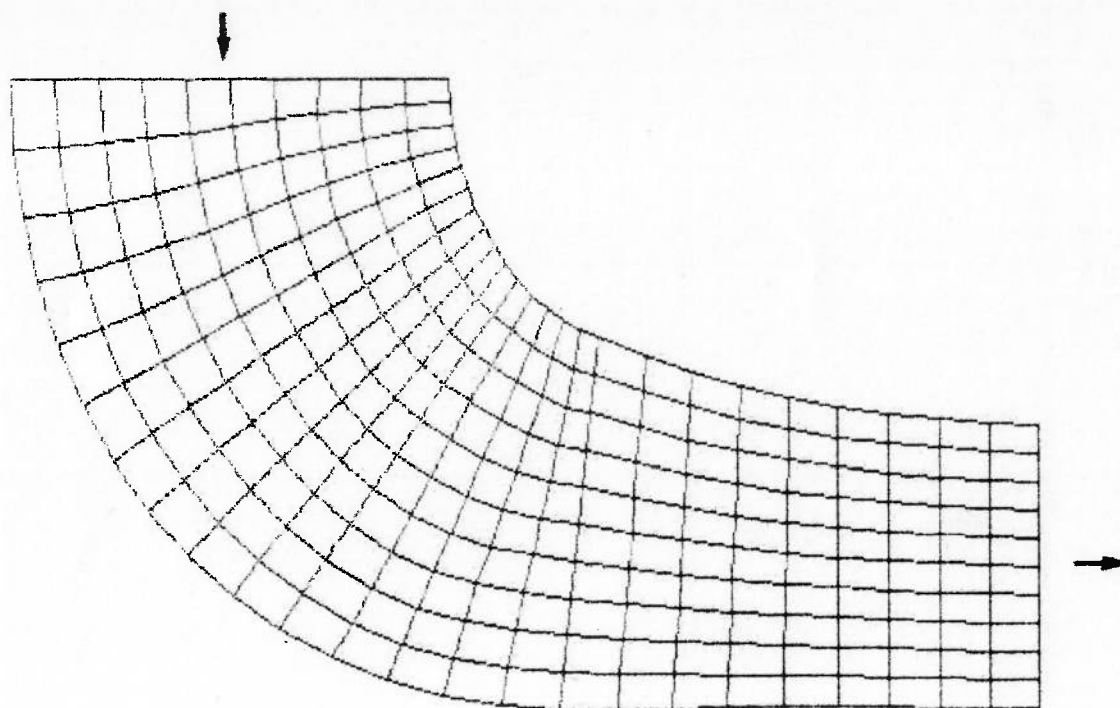


fig. 5.17: Malha para o plano meridiano do rotor

Ao se iniciar os cálculos, o método se mostrou divergente. Tentou-se alterar o intervalo de tempo impondo-se valores, a fim de controlar o avanço das variáveis ao longo das iterações, e tornar o método convergente. Porém, ele se mostrou incondicionalmente divergente.

### Referências

CHORIN, A. J. "A Numerical Method for Solving Incompressible Viscous Flow Problems". *Journal of Computational Physics*, v. 2, p. 12-26, 1967.

KRAMER, J. J., STANITZ, J. D. "Two Dimensional Shear Flow in a 90° Elbow". NACA, TN 2736, 1952.

LAURIA, D. "Ein Beitrag zur Berechnung des gesamten turbulenten Stromungsfeldes in Laufradern hydraulischer Maschinen". Dissertation, Technischen Universität München, 1985.

STANITZ, J. D. " Design of Two Dimensional Channels with Prescribed Velocity Distribution along the Channels Walls. I - Relaxation Solutions". NACA, TN 2593, 1951.

## **CONCLUSÕES E RECOMENDAÇÕES**

Foi apresentada uma metodologia para a determinação de campos de escoamento (velocidade e pressão). Baseia-se na solução da equação de Euler para obter o campo de velocidades. Considera-se que o escoamento seja incompressível e o regime permanente. Na obtenção do campo de pressões é utilizada a equação da continuidade modificada.

O termo temporal da Equação de Euler é discretizado pelo Método de Runge-Kutta de 4<sup>a</sup> ordem, sendo a solução obtida quando este termo tender a zero, enquanto que o termo espacial pelo Método dos Volumes Finitos.

Já para o campo de pressões, a discretização da equação da continuidade modificada é feita pelo Método da Compressibilidade Artificial, ou Pseudo-Compressibilidade, que considera a equação da continuidade para um escoamento compressível, cuja solução também será obtida quando o termo temporal tender a zero.

A partir dos resultados obtidos nas aplicações, pode-se verificar os principais pontos positivos apresentados pelo método:

- alta taxa de convergência;
- reduzido tempo de processamento;
- excelentes resultados para canais sem curvatura.

Como pontos negativos:

- elevada influência do número de células na direção normal ao escoamento no campo de velocidades;
- extrema sensibilidade a curvatura;
- incondicional divergência para escoamentos com rotação.

### **6.1. Desenvolvimentos Futuros**

Recomenda-se que a metodologia apresentada seja implementada utilizando-se novos sistemas operacionais, com plataformas de 32 bits, como os sistemas operacionais "Windows 95" ou "OS/2 Warp".

Desta forma pode-se utilizar das vantagens destes sistemas, principalmente na manipulação de memória e alocação

para seu uso, sem a barreira imposta pelo DOS de 640 Kb, e a instabilidade no uso da memória alta (extendida e/ou expandida).

Outra vantagem no uso destes sistemas são suas interfaces gráficas. No caso pode-se adicionar ao aplicativo a possibilidade de visualização da malha e dos resultados mais facilmente.

Com relação à metodologia proposta, deve-se aprofundar no estudo de escoamentos com rotação e se verificar a influência dos termos no equacionamento, já que o método se mostrou incondicionalmente divergente.



---

## EQUAÇÃO DE EULER

A equação de Navier-Stokes pode ser deduzida a partir da 2ª Lei de Newton, que expressa a proporcionalidade entre uma força aplicada a uma partícula fluida e a aceleração resultante. A equação possui este nome em homenagem ao francês M. Navier e ao inglês G. Stokes, que obtiveram as equações de maneira independente um ao outro.

Na forma não conservativa pode ser escrita da seguinte maneira:

componente na direção x:

$$\rho \frac{Du}{Dt} = -\frac{\partial p}{\partial x} + \frac{\partial \tau_{xx}}{\partial x} + \frac{\partial \tau_{yx}}{\partial y} + \frac{\partial \tau_{zx}}{\partial z} + \rho f_x \quad (\text{A1-1a})$$

direção y:

$$\rho \frac{Dv}{Dt} = -\frac{\partial p}{\partial y} + \frac{\partial \tau_{xy}}{\partial x} + \frac{\partial \tau_{yy}}{\partial y} + \frac{\partial \tau_{zy}}{\partial z} + \rho f_y \quad (\text{A1-1b})$$

direção z:

$$\rho \frac{Dw}{Dt} = -\frac{\partial p}{\partial z} + \frac{\partial \tau_{xz}}{\partial x} + \frac{\partial \tau_{yz}}{\partial y} + \frac{\partial \tau_{zz}}{\partial z} + \rho f_z \quad (\text{AI-1c})$$

onde:

$f_x, f_y$  e  $f_z$  - forças de corpo nas respectivas direções;

$\tau_{i,j}$  - tensões de cisalhamento que agem na face de um cubo elementar representante de uma partícula fluida;

$\tau_{i,i}$  - tensões normais; vide figura AI-1.

$u, v$  e  $w$  - velocidades nas respectivas direções;

$p$  - pressão;

$\rho$  - massa específica.

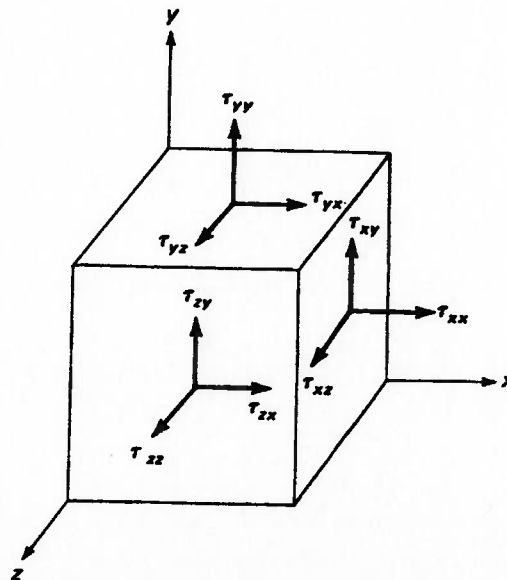


fig. AI-1: partícula fluida

Esta equação que descreve um escoamento viscoso possui termos para representar o fenômeno de transporte de viscosidade, de condução térmica e dissipativos. Não estão inclusos os termos que descrevem o fenômeno da difusão de massa, pois considera-se o fluido homogêneo e sem reações químicas.

Porém, apenas para facilitar a manipulação e sua discretização, é conveniente reescrevê-la na forma conservativa:

$$\frac{\partial(\rho u)}{\partial t} + \nabla(\rho u \vec{V}) = -\frac{\partial p}{\partial x} + \frac{\partial \tau_{xx}}{\partial x} + \frac{\partial \tau_{yx}}{\partial y} + \frac{\partial \tau_{zx}}{\partial z} + \rho f_x \quad (\text{AI-2})$$

onde  $\vec{V}$  representa o vetor velocidade da partícula. Pode-se reescrever as equações nas outras direções de maneira análoga.

A equação de Euler, que descreve um escoamento invíscido, pode ser obtida a partir da equação de Navier-Stokes desprezando-se a dissipação, o fenômeno do transporte de viscosidade, da difusão de massa e condutividade térmica:

direção x:

$$\frac{\partial(\rho u)}{\partial t} + \nabla(\rho u \vec{V}) = -\frac{\partial p}{\partial x} + \rho f_x \quad (\text{AI-3a})$$

direção y:

$$\frac{\partial(\rho v)}{\partial t} + \nabla(\rho v \vec{V}) = -\frac{\partial p}{\partial y} + \rho f_y \quad (\text{AI-3b})$$

direção z:

$$\frac{\partial(\rho w)}{\partial t} + \nabla(\rho w \vec{V}) = -\frac{\partial p}{\partial z} + \rho f_z \quad (\text{AI-3c})$$

Pode-se reescrevê-la da seguinte forma:

$$\frac{\partial W}{\partial t} + \frac{\partial f}{\partial x} + \frac{\partial g}{\partial y} + \frac{\partial h}{\partial z} = q \quad (\text{A1-4})$$

onde a variável primitiva  $W$  é dada por:

$$W = \begin{vmatrix} \rho u \\ \rho v \\ \rho w \end{vmatrix} \quad (\text{A1-5a})$$

e as componentes  $f$ ,  $g$  e  $h$ :

$$f = \begin{vmatrix} \rho u^2 + p \\ \rho uv \\ \rho uw \end{vmatrix} \quad g = \begin{vmatrix} \rho uv \\ \rho v^2 + p \\ \rho vw \end{vmatrix} \quad h = \begin{vmatrix} \rho uw \\ \rho vw \\ \rho w^2 + p \end{vmatrix} \quad (\text{A1-5b})$$

e:

$$q = \begin{vmatrix} \rho f_x \\ \rho f_y \\ \rho f_z \end{vmatrix} \quad (\text{A1-5c})$$

Considerando-se o escoamento bi-dimensional e desprezando as forças de corpo, a equação (A1-4) se reduz à:

$$\frac{\partial W}{\partial t} + \frac{\partial f}{\partial x} + \frac{\partial g}{\partial y} = 0 \quad (\text{A1-5})$$

onde a variável primitiva  $W$  passa a ser dada por:

$$W = \begin{vmatrix} \rho u \\ \rho v \end{vmatrix} \quad (\text{A1-6a})$$

e as componentes:

$$f = \left| \begin{array}{c} \rho u^2 + p \\ \rho uv \end{array} \right| \quad g = \left| \begin{array}{c} \rho uv \\ \rho v^2 + p \end{array} \right| \quad (\text{AI-6b})$$

---

## TRANSFORMAÇÃO DE COORDENADAS

Considere-se uma transformação da forma:

$$\begin{aligned}\xi &= \xi(x, y) \\ \eta &= \eta(x, y)\end{aligned}\tag{AII-1}$$

que pode ser utilizada para transformar as equações de movimento de um sistema de referência cartesiano para um curvilíneo não ortogonal.

Utilizando-se a regra da cadeia para diferenciação parcial, tem-se:

$$\begin{aligned}\frac{\partial}{\partial x} &= \frac{\partial \xi}{\partial x} \frac{\partial}{\partial \xi} + \frac{\partial \eta}{\partial x} \frac{\partial}{\partial \eta} \\ \frac{\partial}{\partial y} &= \frac{\partial \xi}{\partial y} \frac{\partial}{\partial \xi} + \frac{\partial \eta}{\partial y} \frac{\partial}{\partial \eta}\end{aligned}\tag{AII-2a}$$

ou de uma maneira simplificada para representar os coeficientes métricos:

$$\begin{aligned}\frac{\partial}{\partial x} &= \xi_x \frac{\partial}{\partial \xi} + \eta_x \frac{\partial}{\partial \eta} \\ \frac{\partial}{\partial y} &= \xi_y \frac{\partial}{\partial \xi} + \eta_y \frac{\partial}{\partial \eta}\end{aligned}\tag{All-2b}$$

Os coeficientes métricos ( $\xi_x$ ,  $\xi_y$ ,  $\eta_x$  e  $\eta_y$ ) que aparecem na equação (All-2) podem ser determinados da seguinte maneira:

Escreve-se as expressões diferenciais:

$$\begin{aligned}d\xi &= \xi_x dx + \xi_y dy \\ d\eta &= \eta_x dx + \eta_y dy\end{aligned}\tag{All-3}$$

Da mesma maneira pode-se escrever:

$$\begin{aligned}dx &= x_\xi d\xi + x_\eta d\eta \\ dy &= y_\xi d\xi + y_\eta d\eta\end{aligned}\tag{All-4}$$

As equações (All-3) e (All-4) podem ser reescritas na forma matricial:

$$\begin{bmatrix} d\xi \\ d\eta \end{bmatrix} = \begin{bmatrix} \xi_x & \xi_y \\ \eta_x & \eta_y \end{bmatrix} \begin{bmatrix} dx \\ dy \end{bmatrix}\tag{All-5}$$

$$\begin{bmatrix} dx \\ dy \end{bmatrix} = \begin{bmatrix} x_\xi & x_\eta \\ y_\xi & y_\eta \end{bmatrix} \begin{bmatrix} d\xi \\ d\eta \end{bmatrix}\tag{All-6}$$

Comparando-se as equações (All-5) e (All-6), temos:

$$\begin{bmatrix} \xi_x & \xi_y \\ \eta_x & \eta_y \end{bmatrix} = \begin{bmatrix} x_\xi & x_\eta \\ y_\xi & y_\eta \end{bmatrix}^{-1}\tag{All-7}$$

Fazendo-se  $h = x_\xi y_\eta - x_\eta y_\xi$ , pode-se determinar a inversa da matriz à direita da igualdade da equação (All-7):

$$\frac{1}{h} \begin{bmatrix} y_\eta & -x_\eta \\ -y_\xi & x_\xi \end{bmatrix}\tag{All-8}$$

Comparando-se a matriz a esquerda da igualdade de (All-7) com a de (All-8), pode-se determinar os coeficientes métricos:

$$\begin{aligned}\xi_x &= \frac{1}{h} y_\eta \\ \xi_y &= -\frac{1}{h} x_\eta \\ \eta_x &= -\frac{1}{h} y_\xi \\ \eta_y &= \frac{1}{h} x_\xi\end{aligned}\tag{All-9}$$

onde  $h$  é o jacobiano da transformação do sistema de referência cartesiano para o curvilíneo não ortogonal.

Aplicando-se à equação (2.12) (Equações de Euler e para a pressão) na forma diferencial a transformação dada pela equação (All-2) e os coeficientes métricos dados por (All-9), chega-se à equação (All-10):

$$\frac{\partial w}{\partial t} + \frac{\partial f}{\partial x} + \frac{\partial g}{\partial y} = 0\tag{2.12}$$

$$\frac{\partial w}{\partial t} + \xi_x \frac{\partial f}{\partial \xi} + \eta_x \frac{\partial f}{\partial \eta} + \xi_y \frac{\partial g}{\partial \xi} + \eta_y \frac{\partial g}{\partial \eta} = 0\tag{All-10}$$

Pode-se somar e subtrair os termos  $f \frac{\partial (\xi_x)}{\partial \xi}$ ,  $g \frac{\partial (\xi_y)}{\partial \xi}$ ,

$f \frac{\partial (\eta_x)}{\partial \eta}$  e  $g \frac{\partial (\eta_y)}{\partial \eta}$  à equação (All-10), chegando-se à equação

(All-11):

$$\frac{\partial w}{\partial t} + \left[ \frac{\partial (f \xi_x)}{\partial \xi} + \frac{\partial (g \xi_y)}{\partial \xi} - f \frac{\partial (\xi_x)}{\partial \xi} - g \frac{\partial (\xi_y)}{\partial \xi} \right] + \left[ \frac{\partial (f \eta_x)}{\partial \eta} + \frac{\partial (g \eta_y)}{\partial \eta} - f \frac{\partial (\eta_x)}{\partial \eta} - g \frac{\partial (\eta_y)}{\partial \eta} \right] = 0\tag{All-11}$$



A equação (All-11) pode ainda ser rearranjada da seguinte maneira:

$$\frac{\partial w}{\partial t} + \frac{\partial}{\partial \xi} [f \xi_x + g \xi_y] + \frac{\partial}{\partial \eta} [f \eta_x + g \eta_y] - K = 0 \quad (\text{All-12a})$$

onde  $K$  é dado por:

$$K = f \left( \frac{\partial(\xi_x)}{\partial \xi} + \frac{\partial(\eta_x)}{\partial \eta} \right) + g \left( \frac{\partial(\xi_y)}{\partial \xi} + \frac{\partial(\eta_y)}{\partial \eta} \right) \quad (\text{All-12b})$$

Verifica-se que os termos da equação (All-12b) são nulos, reduzindo a equação (All-12a) para:

$$\frac{\partial w}{\partial t} + \frac{\partial}{\partial \xi} [f \xi_x + g \xi_y] + \frac{\partial}{\partial \eta} [f \eta_x + g \eta_y] = 0 \quad (\text{All-13})$$

Substituindo-se os coeficientes métricos na equação (All-13), pode-se chegar à seguinte expressão:

$$\frac{\partial w}{\partial t} + \frac{\partial}{\partial \xi} \left[ \frac{1}{h} (f y_\eta - g x_\eta) \right] + \frac{\partial}{\partial \eta} \left[ \frac{1}{h} (-f y_\xi + g x_\xi) \right] = 0 \quad (\text{All-14})$$

Multiplicando-se a equação (All-14) por  $h$  e fazendo-se:

$$F = \left[ (f y_\eta - g x_\eta) \right] = \begin{bmatrix} \beta u y_\eta - \beta v x_\eta \\ y_\eta (\rho u^2 + p) - x_\eta \rho u v \\ y_\eta \rho u v - x_\eta (\rho v^2 + p) \end{bmatrix} = \begin{bmatrix} \beta (u y_\eta - v x_\eta) \\ y_\eta p + \rho u (u y_\eta - v x_\eta) \\ -x_\eta p + \rho v (u y_\eta - v x_\eta) \end{bmatrix}$$

$$G = \left[ (-f y_\xi + g x_\xi) \right] = \begin{bmatrix} -\beta u y_\xi + \beta v x_\xi \\ -y_\xi (\rho u^2 + p) + x_\xi \rho u v \\ -y_\xi \rho u v + x_\xi (\rho v^2 + p) \end{bmatrix} = \begin{bmatrix} \beta (-u y_\xi + v x_\xi) \\ -y_\xi p + \rho u (-u y_\xi + v x_\xi) \\ x_\xi p + \rho v (-u y_\xi + v x_\xi) \end{bmatrix}$$

(All-15)

pode-se reescrevê-la na seguinte forma:

$$\frac{\partial W}{\partial t} + \frac{\partial F}{\partial \xi} + \frac{\partial G}{\partial \eta} = 0 \quad (\text{AII-16})$$

Integrando-se a equação (AII-16) num volume de controle e aplicando-se o Teorema de Gauss, pode-se obter a equação na forma conservativa e integral:

$$\frac{\partial}{\partial t} \int_{Vol} W \cdot dV + \oint_{Sup} \vec{Q} \cdot d\vec{S} = 0 \quad (\text{AII-17})$$

onde a variável primitiva  $W$  é dada por:

$$W = \begin{vmatrix} hp \\ h\rho u \\ h\rho v \end{vmatrix} \quad (\text{AII-17a})$$

e as componentes  $F$  e  $G$  do vetor de fluxo  $\vec{Q}$ :

$$F = \begin{vmatrix} \beta h U \\ \rho h u U + y_\eta p \\ \rho h v U - x_\eta p \end{vmatrix} \quad G = \begin{vmatrix} \beta h V \\ \rho h u V - y_\xi p \\ \rho h v V + x_\xi p \end{vmatrix} \quad (\text{AII-17b})$$

sendo:

$$\left. \begin{aligned} U &= \frac{1}{h}(uy_\eta - vx_\eta) \\ V &= \frac{1}{h}(-uy_x + vx_x) \end{aligned} \right\} \text{contravariantes da velocidade.}$$

As contravariantes da velocidade,  $U$  e  $V$ , representarão as velocidades no novo sistema de coordenadas, e não são necessariamente ortogonais.

---

## LISTAGEM DO PROGRAMA

Como já citado, a linguagem de programação utilizada para o desenvolvimento do projeto *VOLFIN* foi o Borland Turbo C, versão 2.0. Esta linguagem permite, além de se utilizar funções, dividi-las em vários arquivos a fim de facilitar sua organização.

Arquivos com extensão "H" permitem a definição das variáveis globais e das funções do projeto. No arquivo "V\_VAR.C" é feita a inicialização das variáveis. Já no arquivo "VOLFIN.C" tem-se a função principal e nos demais arquivos o restante das funções.

### Arquivo "VOLFIN.H"

```
extern double    **x,      /* coordenada x */
                 **y,      /* coordenada y */
                 **p        /* variavel para pressao */
                 **u        /* variavel para veloc. dir. x */
                 **v        /* variavel para veloc. dir. y */
                 **U        /* contravar. da veloc. dir. escoam. */
```

```

**V      /* contravar. da veloc. dir. normal ao
          escoamento */
**A,     /* area da celula */
**XQS,   /* delta x por delta qsi */
**YQS,   /* delta y por delta qsi */
**XET,   /* delta x por delta eta */
**YET,   /* delta y por delta eta */
**H,     /* jacobiano de transformacao */
**DQS,   /* distancia de 2 linhas qsi constantes */
**DET,   /* distancia de 2 linhas eta constantes */
Q,       /* vazao na entrada */
ro;      /* massa especifica */

extern int I,J; /* numero de pontos em qsi e eta respectivamente */

void leitura(); /* leitura de coordenadas */
void erro();   /* erro na alocao */
void entrada(); /* entrada de dados ro e cond. inic. */
void desaloca(double **,int); /* desaloca memoria p. variav. double */
void processa(); /* calcula equacao principal */
void infproces(); /* mostra tela inform. durante processam. */
void result(long,int,double,
            double,double); /* mostra tela c/ resultados */
void saida(); /* abre arquivo com resultados */

double **aloca2d(int,int); /* aloca memoria */
double f(int,int,int,int,int); /* calcula fluxo F */
double g(int,int,int,int,int); /* calcula fluxo G */
double FS(int,int,int); /* calcula variacao de fluxo F.S */
double dt(int,int); /* calcula intervalo de tempo */
double beta(int,int); /* calcula produto ro.c.c */
double c(int,int); /* calcula pseudo vel. do som */
double dxy(int,int,int,int); /* calc. dist. entre 2 pontos */
double uns(int,int); /* transf. cont-var. U e V para vel. u */
double vns(int,int); /* transf. cont-var. U e V para vel. v */
double UNS(int,int); /* transf. vel. u e v para cont-var. U */

```

```

double VNS(int,int);      /* transf. vel. u e v para cont-var. V */
double DN(int,int,int);  /* dissipacao artificial */
double ni(int,int);      /* gradiente de pressao local */
double dni(int,int,int,int); /* dissipacao na dir. qsi */
double dnj(int,int,int,int); /* dissipacao na dir. eta */

```

### Arquivo "V\_VAR.C":

```
#include <alloc.h>
```

```

double  **x=NULL, /* coordenadas x */
        **y=NULL, /* coordenadas y */
        **p=NULL, /* pressao */
        **u=NULL, /* veloc. dir. x */
        **v=NULL, /* veloc. dir. y */
        **U=NULL, /* contravariante da vel. dir. escoam. */
        **V=NULL, /* contravariante da vel. dir. normal escoam. */
        **A=NULL, /* area da celula */
        **DQS=NULL, /* distancia de 2 qsi constantes */
        **DET=NULL, /* distancia de 2 eta constantes */
        **XQS=NULL, /* delta x por delta qsi */
        **YQS=NULL, /* delta y por delta qsi */
        **XET=NULL, /* delta x por delta eta */
        **YET=NULL, /* delta y por delta eta */
        **H=NULL, /* jacobiano de transformacao */
        Q=0.0, /* vazao na entrada */
        ro; /* massa especifica */

int I,J; /* num. pontos dir. escoam. e normal escoam. */

```

### Arquivo "VOLFIN.C":

```

#include <stdlib.h>
#include <stdio.h>
#include <dos.h>
#include <conio.h>
#include "volfin.h"

```

```

int main()
{
char a;

```

```
do
{
    leitura();
    entrada();
    processa();
    saida();
    clrscr();
    printf("\n\n\n Deseja novo calculo? (S/N)");
    a=getchar();
    if (a == "N") exit(0);
}
while (a=="S");

return 0;
}
```

### Arquivo "V\_LEARQ.C":

```
#include <stdio.h>
#include <io.h>
#include <stdlib.h>
#include <conio.h>
#include <math.h>
#include "volfin.h"

void leitura()
{
    double z=0;
    int i=0,j=0;
    FILE *arq;

    /* desalocacao de memoria se for segundo calculo */
    if(x!=NULL)
    {
        desaloca(x,I);
        desaloca(y,I);
        desaloca(p,I);
        desaloca(u,I);
        desaloca(v,I);
        desaloca(U,I);
        desaloca(V,I);
        desaloca(A,I);
        desaloca(DQS,I);
        desaloca(DET,I);
        desaloca(XQS,I);
    }
}
```

```
    desaloca(YQS,I);
    desaloca(XET,I);
    desaloca(YET,I);
    desaloca(H,I);
}

/* abertura do arquivo */
if ((arq = fopen("c:\\temp\\bocal.txt", "rt")) == NULL)
{
    clrscr();
    printf("\n\n Arquivo nao aberto\n");
    printf("\n\n Tecele algo para continuar...\n\n");
    getch();
    return;
}
else
{
    clrscr();

/* leitura do numero de pontos em x e y */
    fscanf(arq,"%d %d",&I,&J);
    printf("\n\n Numero de pontos na direcao i = %d",I);
    printf("\n\n Numero de pontos na direcao j = %d\n\n",J);
    printf(" P(i,j)    x    y\n\n");

/* alocao de memoria para x y p u v A DQS DET XQS YQS XET YET
H */
    x=aloca2d(I,J); if(x==NULL) return;
    y=aloca2d(I,J); if(y==NULL) return;
    p=aloca2d(I-1,J-1); if(p==NULL) return;
    u=aloca2d(I-1,J-1); if(u==NULL) return;
    v=aloca2d(I-1,J-1); if(v==NULL) return;
    U=aloca2d(I-1,J-1); if(U==NULL) return;
    V=aloca2d(I-1,J-1); if(V==NULL) return;
    A=aloca2d(I-1,J-1); if(A==NULL) return;
    DQS=aloca2d(I-1,J-1); if(DQS==NULL) return;
    DET=aloca2d(I-1,J-1); if(DET==NULL) return;
    XQS=aloca2d(I-1,J-1); if(XQS==NULL) return;
    YQS=aloca2d(I-1,J-1); if(YQS==NULL) return;
    XET=aloca2d(I-1,J-1); if(XET==NULL) return;
    YET=aloca2d(I-1,J-1); if(YET==NULL) return;
    H=aloca2d(I-1,J-1); if(H==NULL) return;

/* leitura das coordenadas x y */
    for(i=1;i<=I;i++)
    {
```

```

for(j=1;j<=J;j++)
{
    fscanf(arq,"%lf",&x[i][j]);
    fscanf(arq,"%lf",&y[i][j]);
}
}

for(i=1;i<=I;i++)
{
    for(j=1;j<=J;j++)
    {
        printf(" P(%d,%d) %lf %lf\n",i,j,x[i][j],y[i][j]);
    }
    printf("\n          Tecle algo para continuar...\n\n");
    getch();
    clrscr();
    printf("\n\n\n");
}
}
fclose(arq);

/* calculo da area e dos coef. métricos da transf. de coordenadas */
for(i=1;i<=I-1;i++)
{
    for(j=1;j<=J-1;j++)
    {
        A[i][j]=.5*fabs((x[i ][j ]-x[i+1][j ])*(y[i ][j ]+y[i+1][j ])+
            (x[i+1][j ]-x[i+1][j+1])*(y[i+1][j ]+y[i+1][j+1])+
            (x[i+1][j+1]-x[i ][j+1])*(y[i+1][j+1]+y[i ][j+1])+
            (x[i ][j+1]-x[i ][j ])*(y[i ][j+1]+y[i ][j ]));
        DQS[i][j]=sqrt((.5*(x[i+1][j ]+x[i+1][j+1])-.5*(x[i ][j ]+x[i ][j+1]))*
            (.5*(x[i+1][j ]+x[i+1][j+1])-.5*(x[i ][j ]+x[i ][j+1]))+
            (.5*(y[i+1][j ]+y[i+1][j+1])-.5*(y[i ][j ]+y[i ][j+1]))*
            (.5*(y[i+1][j ]+y[i+1][j+1])-.5*(y[i ][j ]+y[i ][j+1])));
        DET[i][j]=sqrt((.5*(x[i ][j+1]+x[i+1][j+1])-.5*(x[i ][j ]+x[i+1][j ]))*
            (.5*(x[i ][j+1]+x[i+1][j+1])-.5*(x[i ][j ]+x[i+1][j ]))+
            (.5*(y[i ][j+1]+y[i+1][j+1])-.5*(y[i ][j ]+y[i+1][j ]))*
            (.5*(y[i ][j+1]+y[i+1][j+1])-.5*(y[i ][j ]+y[i+1][j ])));
        XQS[i][j]=(.5*(x[i+1][j ]+x[i+1][j+1])-.5*(x[i ][j ]+x[i
            ][j+1]))/DQS[i][j];
        YQS[i][j]=(.5*(y[i+1][j ]+y[i+1][j+1])-.5*(y[i ][j ]+y[i
            ][j+1]))/DQS[i][j];
        XET[i][j]=(.5*(x[i ][j+1]+x[i+1][j+1])-.5*(x[i ][j ]+x[i+1][j
            ]))/DET[i][j];
        YET[i][j]=(.5*(y[i ][j+1]+y[i+1][j+1])-.5*(y[i ][j ]+y[i+1][j
            ]))/DET[i][j];
    }
}

```



```
H[i][j]=XQS[i][j]*YET[i][j]-XET[i][j]*YQS[i][j];
    }
  }
}
```

**Arquivo "V\_ENTRDA.C":**

```
#include <stdio.h>
#include <io.h>
#include <stdlib.h>
#include <conio.h>
#include "volfin.h"

void entrada()
{
  int i,j;

  /* propriedades do fluido */
  ro = getche();

  /* condicao de contorno na entrada */
  i= getche();
  for(j=1;j<=J-1;j++)
  {
    U[1][j] = i
    V[1][j] = i;
  }

  /* condicao de contorno na saida */
  i=getche();
  for(j=1;j<=J-1;j++)
  {
    p[I-1][j]= i;
  }

  /* condicao inicial para toda malha */
  for(i=2;i<=I-1;i++)
  {
    for(j=1;j<=J-1;j++)
    {
      U[i][j]=U[1][1];
      V[i][j]=V[1][1];
    }
  }
}
```

```
for(i=1;i<=I-2;i++)
{
  for(j=1;j<=J-1;j++)
  {
    p[i][j]=p[I-1][1];
  }
}

/* mostrar na tela as condicoes iniciais */
clrscr();
printf("\n      Condicoes iniciais \n");
printf(" Na entrada \n");
for(j=1;j<=J-1;j++)
{
  printf("  U[1][%d] = %lf",j,U[1][j]);
  printf("  V[1][%d] = %lf\n",j,V[1][j]);
}
printf("\n      Tecle algo para continuar...\n");
getch();
clrscr();

printf(" Na saida \n");
for(j=1;j<=J-1;j++)
{
  printf("  p[%d][%d] = %lf \n",I-1,j,p[I-1][j]);
}

printf("\n      Tecle algo para continuar...\n");
getch();
clrscr();

printf("\n Para toda malha \n\n");
printf("  p = %lf\n",p[2][2]);
printf("  U = %lf\n",U[2][2]);
printf("  V = %lf\n",V[2][2]);
printf("\n      Tecle algo para continuar...\n");
getch();
}
```

#### Arquivo "V\_PROCES.C":

```
#include <stdlib.h>
#include <stdio.h>
#include <math.h>
#include <conio.h>
```

```

#include <time.h>
#include <dos.h>
#include "volfin.h"

void processa()
{
int i,j,k=0; /* variaveis para lacos e iteracao */
double rp,ru,rv; /* variaveis para residuo */
double pa,*ua,*va, /* var. p. guardar p u v no dt anterior */
      pm,um,vm, /* valores medios das variaveis para iteracao n */
      a,b, /* variaveis para acelerar produtos */
      c,d,e, /* variaveis para acelerar dissipacao */
      QWP[4], /* produto FS para pressao no runge kutta */
      QWU[4], /* produto FS para vel. u no runge kutta */
      QWV[4], /* produto FS para vel. v no runge kutta */
      q; /* vazao na seccao */
long ti,tf;
  ua = (double *) malloc((unsigned) (J+1)*sizeof(double *));
  if (!ua) erro();
  ua -= 1;
  va = (double *) malloc((unsigned) (J+1)*sizeof(double *));
  if (!va) erro();
  va -= 1;

  clrscr();
  ti = time(0);
  n=0;
  infproces();

  /* vazao na entrada */
  for(k=1;k<=J-1;k++)
  {
  Q += U[1][k]*.5*(dxy(1,k,1,k+1)+dxy(2,k,2,k+1));
  }

  /* atualizacao da velocidade */
  for(i=1;i<=I-1;i++)
  {
  for(j=1;j<=J-1;j++)
  {
  u[i][j] = uns(i,j);
  v[i][j] = vns(i,j);
  }
  }
}
do

```

```

{
rp=0.0;
ru=0.0;
rv=0.0;
n++;

gotoxy(21,12);
cprintf("%4d",n);

/* calculo de p na entrada */
gotoxy(51,14);
cprintf("%4d",1);
for(j=1;j<=J-1;j++)
{
i = 1;
gotoxy(51,15);
cprintf("%4d",j);
/* calculo de p no tempo explicitamente */
pa = p[i][j];
a = A[i][j]*H[i][j];
c = DN(1,i,j);
if(j==1) { pm = .5*(p[i][j+1]+p[i+1][j]); }
else if(j==J-1) { pm = .5*(p[i][j-1]+p[i+1][j]); }
else { pm = .25*(p[i][j-1]+p[i][j+1]+2*p[i+1][j]); }
QWP[0] = FS(1,i,j);
p[i][j] = pm-dt(i,j)*QWP[0]/(2*a)+dt(i,j)*c/(2*a);
QWP[1] = FS(1,i,j);
p[i][j] = pm-dt(i,j)*QWP[1]/(2*a)+dt(i,j)*c/(2*a);
QWP[2] = FS(1,i,j);
p[i][j] = pm-dt(i,j)*QWP[2]/a+dt(i,j)*c/a;
QWP[3] = FS(1,i,j);
p[i][j] = pm-(dt(i,j)/(6*a))*(QWP[0]+2*QWP[1]+2*QWP[2]+QWP[3])+dt(i,j)*c/a;
rp += (p[i][j]-pa)*(p[i][j]-pa);
}

/* calculo de u v p para toda malha exceto entrada e saida */
for(i=2;i<=I-2;i++)
{
q = 0;
gotoxy(51,14);
cprintf("%4d",i);
for(j=1;j<=J-1;j++)
{
gotoxy(51,15);
cprintf("%4d",j);
gotoxy(51,15);

```

```

    cprintf("%4d", j);
/* calculo de u v p no tempo explicitamente */
    pa = p[i][j];
    ua[j]=u[i][j];
    va[j]=v[i][j];
    a = A[i][j]*H[i][j];
    b = a*ro;
    c = DN(1,i,j);
    d = DN(2,i,j);
    e = DN(3,i,j);
    if (j==1) { pm = .25*(p[i-1][j]+2*p[i][j+1]+p[i+1][j]);
                um = .25*(u[i-1][j]+2*u[i][j+1]+u[i+1][j]);
                vm = .25*(v[i-1][j]+2*v[i][j+1]+v[i+1][j]); }
    else if (j==J-1) { pm = .25*(p[i-1][j]+2*p[i][j-1]+p[i+1][j]);
                       um = .25*(u[i-1][j]+2*u[i][j-1]+u[i+1][j]);
                       vm = .25*(v[i-1][j]+2*v[i][j-1]+v[i+1][j]); }
    else { pm = .25*(p[i-1][j]+p[i][j-1]+p[i][j+1]+p[i+1][j]);
           um = .25*(u[i-1][j]+u[i][j-1]+u[i][j+1]+u[i+1][j]);
           vm = .25*(v[i-1][j]+v[i][j-1]+v[i][j+1]+v[i+1][j]); }
    QWP[0] = FS(1,i,j); QWU[0] = FS(2,i,j); QWV[0] = FS(3,i,j);
    p[i][j] = pm-dt(i,j)*QWP[0]/(2*a)+dt(i,j)*c/(2*a);
    u[i][j] = um-dt(i,j)*QWU[0]/(2*b)+dt(i,j)*d/(2*b);
    v[i][j] = vm-dt(i,j)*QWV[0]/(2*b)+dt(i,j)*e/(2*b);
    QWP[1] = FS(1,i,j); QWU[1] = FS(2,i,j); QWV[1] = FS(3,i,j);
    p[i][j] = pm-dt(i,j)*QWP[1]/(2*a)+dt(i,j)*c/(2*a);
    u[i][j] = um-dt(i,j)*QWU[1]/(2*b)+dt(i,j)*d/(2*b);
    v[i][j] = vm-dt(i,j)*QWV[1]/(2*b)+dt(i,j)*e/(2*b);
    QWP[2] = FS(1,i,j); QWU[2] = FS(2,i,j); QWV[2] = FS(3,i,j);
    p[i][j] = pm-dt(i,j)*QWP[2]/a+dt(i,j)*c/a;
    u[i][j] = um-dt(i,j)*QWU[2]/b+dt(i,j)*d/b;
    v[i][j] = vm-dt(i,j)*QWV[2]/b+dt(i,j)*e/b;
    QWP[3] = FS(1,i,j); QWU[3] = FS(2,i,j); QWV[3] = FS(3,i,j);
    p[i][j] = pm-(dt(i,j)/(6*a))*(QWP[0]+2*QWP[1]+2*QWP[2]+QWP[3])+dt(i,j)*c/a;
    u[i][j] = um-
(dt(i,j)/(6*b))*(QWU[0]+2*QWU[1]+2*QWU[2]+QWU[3])+dt(i,j)*d/b;
    v[i][j] = vm-(dt(i,j)/(6*b))*(QWV[0]+2*QWV[1]+2*QWV[2]+QWV[3])+dt(i,j)*e/b;
    U[i][j] = UNS(i,j);
    if (j==1 || j==J-1)
        { V[i][j] = 0;
          }
    else
        { V[i][j] = VNS(i,j); }
    u[i][j] = uns(i,j);
    v[i][j] = vns(i,j);
    q += U[i][j]*.5*(dxy(i,j,i,j+1)+dxy(i+1,j,i+1,j+1));
/* residuo */

```

```

    rp += (p[i][j]-pa)*(p[i][j]-pa);
}

/* correcao da velocidade devido vazao na seccao */
for(k=1;k<=J-1;k++)
{
    U[i][k] += (Q-q)/((J-1)*.5*(dxy(i,k,i,k+1)+dxy(i+1,k,i+1,k+1)));
    u[i][k] = uns(i,k);
    v[i][k] = vns(i,k);
    ru += (u[i][k]-ua[k])*(u[i][k]-ua[k]);
    rv += (v[i][k]-va[k])*(v[i][k]-va[k]);
}
}

/* calculo de u v para saida */
i=I-1;
q = 0;
gotoxy(51,14);
cprintf("%4d",i);
for(j=1;j<=J-1;j++)
{
    ua[j]=u[i][j];
    va[j]=v[i][j];
    gotoxy(51,15);
    cprintf("%4d",j);
    gotoxy(51,15);
    cprintf("%4d",j);
}
/* calculo de u v no tempo explicitamente */
a = A[i][j]*H[i][j];
b = a*ro;
d = DN(2,i,j);
e = DN(3,i,j);
if (j==1) { um = .5*(u[i-1][j]+u[i][j+1]);
            vm = .5*(v[i-1][j]+v[i][j+1]); }
else if (j==J-1) { um = .5*(u[i-1][j]+u[i][j-1]);
                  vm = .5*(v[i-1][j]+v[i][j-1]); }
else { um = .25*(2*u[i-1][j]+u[i][j-1]+u[i][j+1]);
      vm = .25*(2*v[i-1][j]+v[i][j-1]+v[i][j+1]); }
QWU[0] = FS(2,i,j); QWV[0] = FS(3,i,j);
u[i][j] = um-dt(i,j)*QWU[0]/(2*b)+dt(i,j)*d/(2*b);
v[i][j] = vm-dt(i,j)*QWV[0]/(2*b)+dt(i,j)*e/(2*b);
QWU[1] = FS(2,i,j); QWV[1] = FS(3,i,j);
u[i][j] = um-dt(i,j)*QWU[1]/(2*b)+dt(i,j)*d/(2*b);
v[i][j] = vm-dt(i,j)*QWV[1]/(2*b)+dt(i,j)*e/(2*b);
QWU[2] = FS(2,i,j); QWV[2] = FS(3,i,j);
u[i][j] = um-dt(i,j)*QWU[2]/(b)+dt(i,j)*d/b;

```

```

v[i][j] = vm-dt(i,j)*QWV[2]/(b)+dt(i,j)*e/b;
QWU[3] = FS(2,i,j); QWV[3] = FS(3,i,j);
u[i][j] = um-
(dt(i,j)/(6*b))*(QWU[0]+2*QWU[1]+2*QWU[2]+QWU[3])+dt(i,j)*d/b;
v[i][j] = vm-(dt(i,j)/(6*b))*(QWV[0]+2*QWV[1]+2*QWV[2]+QWV[3])+dt(i,j)*e/b;
U[i][j] = UNS(i,j);
if (j==1 || j==J-1)
  { V[i][j] = 0;}
else
  { V[i][j] = VNS(i,j); }
u[i][j] = uns(i,j);
v[i][j] = vns(i,j);
q += U[i][j]*.5*(dxy(i,j,i,j+1)+dxy(i+1,j,i+1,j+1));
}

/* correcao da velocidade devido vazao */
for(k=1;k<=J-1;k++)
{
  U[i][k] += (Q-q)/((J-1)*.5*(dxy(i,k,i,k+1)+dxy(i+1,k,i+1,k+1)));
  u[i][k] = uns(i,k);
  v[i][k] = vns(i,k);
  ru += (u[i][k]-ua[k])*(u[i][k]-ua[k]);
  rv += (v[i][k]-va[k])*(v[i][k]-va[k]);
}

ru = (sqrt(ru/((I-2)*(J-1))));
rv = (sqrt(rv/((I-2)*(J-1))));
rp = (sqrt(rp/((I-2)*(J-1))));
gotoxy(29,17);cprintf("%7.4e",ru);
gotoxy(29,18);cprintf("%7.4e",rv);
gotoxy(29,19);cprintf("%7.4e",rp);

tf=time(0);
tp=tf-ti;
if(kbhit()!=0)
{
  result(tp,n,ru,rv,rp);
  infproces();
}

}

while (rp > 1e-10);
saida();
}

void infproces()

```

```

{
clrscr();
printf("\n\n\n\n Numero de pontos na direcao i = %d\n",I);
printf(" Numero de pontos na direcao j = %d\n",J);
printf(" Numero total de pontos = %d\n",I*J);
printf(" Numero de celulas na direcao i = %d\n",I-1);
printf(" Numero de celulas na direcao j = %d\n",J-1);
printf(" Numero total de celulas = %d\n",(I-1)*(J-1));
printf("\n Iteracao numero \n");
printf("\n Processando calculo para celula na direcao I = ");
printf("\n Processando calculo para celula na direcao J = \n");
printf("\n Erro na velocidade u = \n");
printf(" Erro na velocidade v = \n");
printf(" Erro na pressao p = \n");
}

void result(long tp,int n,double ru,double rv,double rp)
{
int i,j;
delay(0);
sound(440);
delay(500);
nosound();
getch();
clrscr();
printf(" Resultados para a iteracao %d \n",n);
printf(" Residuo na velocidade u = %7.4e \n",ru);
printf(" Residuo na velocidade v = %7.4e \n",rv);
printf(" Residuo na pressao p = %7.4e \n",rp);
printf(" Tempo de processamento = %d seg. \n",tp);
for(i=1;i<=I-1;i++)
{
for(j=1;j<=J-1;j++)
{
printf("p(%d,%d)=%lf",i,j,p[i][j]);
printf(" u(%d,%d)=%lf",i,j,u[i][j]);
printf(" v(%d,%d)=%lf",i,j,v[i][j]);
printf(" q(%d,%d)=%lf\n",i,j,sqrt(u[i][j]*u[i][j]+v[i][j]*v[i][j]));
}
}
getch();
clrscr();
printf("\n\n\n\n");
}
saida();
}

```



**Arquivo "V\_FUNCFS.C":**

```

#include <math.h>
#include <stdio.h>
#include <io.h>
#include <stdlib.h>
#include <conio.h>
#include "volfin.h"

double FS(int a,int i,int j)
{
double fs;
fs = f(a,i ,j ,i+1,j )*dxy(i+1,j ,i+1,j+1)
    +g(a,i ,j ,i ,j+1)*dxy(i ,j+1,i+1,j+1)
    -f(a,i ,j ,i-1,j )*dxy(i ,j ,i ,j+1)
    -g(a,i ,j ,i ,j-1)*dxy(i ,j ,i+1,j );
return (fs);
}

double f(int a,int i,int j,int k,int l)
{
double fu;
if(i<1) i=1; if(i>I-1) i=I-1;
if(j<1) j=1; if(j>J-1) j=J-1;
if(k<1) k=1; if(k>I-1) k=I-1;
if(l<1) l=1; if(l>J-1) l=J-1;
switch(a)
{
case 1:
fu = beta(i,j)*H[i][j]*(.5*U[i][j]+.5*U[k][l]);
return (fu);
case 2:
fu =
ro*H[i][j]*.5*(U[i][j]+U[k][l])* .5*(u[i][j]+u[k][l])+YET[i][j]*.5*(p[i][j]+p[k][l]);
return (fu);
case 3:
fu = ro*H[i][j]*.5*(U[i][j]+U[k][l])* .5*(v[i][j]+v[k][l])-
XET[i][j]*.5*(p[i][j]+p[k][l]);
return (fu);
}
return (fu);
}

double g(int a,int i,int j,int k,int l)

```

```

{
double gu;
if(i<1) i=1; if(i>I-1) i=I-1;
if(j<1) j=1; if(j>J-1) j=J-1;
if(k<1) k=1; if(k>I-1) k=I-1;
if(l<1) l=1; if(l>J-1) l=J-1;
switch(a)
{
case 1:
gu = beta(i,j)*H[i][j]*(.5*V[i][j]+.5*V[k][l]);
return (gu);
case 2:
gu = ro*H[i][j]*.5*(V[i][j]+V[k][l])* .5*(u[i][j]+u[k][l])-
YQS[i][j]*.5*(p[i][j]+p[k][l]);
return (gu);
case 3:
gu =
ro*H[i][j]*.5*(V[i][j]+V[k][l])* .5*(v[i][j]+v[k][l])+XQS[i][j]*.5*(p[i][j]+p[k][l]);
return (gu);
}
return (gu);
}

```

```

double DN(int a, int i, int j)
{
double D=0.0;
D = (dni(a,i,j,i+1)-dni(a,i,j,i-1))+(dnj(a,i,j,j+1)-dnj(a,i,j,j-1));
return (D);
}

```

```

double dni(int a, int i, int j, int k)
{
double d=0, kd=.25, kq=(1/256), epsd=0, epsq=0, nu=0, nd=0, l, m, n, o;
if(i<1) i=1; if(i>I-1) i=I-1;
if(j<1) j=1; if(j>J-1) j=J-1;
if(k<1) k=1; if(k>I-1) k=I-1;
l=i+1; m=i-1; n=i+2; o=i-2;
if(l<1) l=1; if(l>I-1) l=I-1;
if(m<1) m=1; if(m>I-1) m=I-1;
if(n<1) n=1; if(n>I-1) n=I-1;
if(o<1) o=1; if(o>I-1) o=I-1;
nu = ni(i,j);
nd = ni(k,j);
if (nu>nd)
{ epsd = kd*nu; }
else

```

```

    { epsd = kd*nd; }
    if ((kq-epsd)>0)
    { epsq = kq-epsd; }
    else
    { epsq = 0; }

    switch(a)
    {
    case 1:
        if (k>=i)
        { d = ((A[i][j]/dt(i,j))*(epsd*(p[k][j]-p[i][j])
            -epsq*(p[n][j]-3*p[l][j]+3*p[i][j]-p[m][j]))) *H[i][j]; }
        else
        { d = ((A[i][j]/dt(i,j))*(epsd*(p[k][j]-p[i][j])
            -epsq*(p[o][j]-3*p[m][j]+3*p[i][j]-p[l][j]))) *H[i][j]; }
        return (d);
    case 2:
        if (k>=i)
        { d = ((A[i][j]/dt(i,j))*(epsd*(u[k][j]-u[i][j])
            -epsq*(u[n][j]-3*u[l][j]+3*u[i][j]-u[m][j]))) * (ro *H[i][j]); }
        else
        { d = ((A[i][j]/dt(i,j))*(epsd*(u[k][j]-u[i][j])
            -epsq*(u[o][j]-3*u[m][j]+3*u[i][j]-u[l][j]))) * (ro *H[i][j]); }
        return (d);
    case 3:
        if (k>=i)
        { d = ((A[i][j]/dt(i,j))*(epsd*(v[k][j]-v[i][j])
            -epsq*(v[n][j]-3*v[l][j]+3*v[i][j]-v[m][j]))) * (ro *H[i][j]); }
        else
        { d = ((A[i][j]/dt(i,j))*(epsd*(v[k][j]-v[i][j])
            -epsq*(v[o][j]-3*v[m][j]+3*v[i][j]-v[l][j]))) * (ro *H[i][j]); }
        return (d);
    }
    return (d);
}

double dnj(int a, int i, int j, int k)
{
    double d=0, kd=.25, kq=(1/256), epsd=0, epsq=0, nu=0, nd=0, l, m, n, o;
    iff(i<1) i=1; iff(i>I-1) i=I-1;
    iff(j<1) j=1; iff(j>J-1) j=J-1;
    iff(k<1) k=1; iff(k>J-1) k=J-1;
    l=j+1; m=j-1; n=j+2; o=j-2;
    iff(l<1) l=1; iff(l>J-1) l=J-1;
    iff(m<1) m=1; iff(m>J-1) m=J-1;
    iff(n<1) n=1; iff(n>J-1) n=J-1;
}

```

```

if(o<1) o=1; if(o>J-1) o=J-1;
nu = ni(i,j);
nd = ni(i,k);
if (nu>nd)
  { epsd = kd*nu; }
else
  { epsd = kd*nd; }
if ((kq-epsd)>0)
  { epsq = kq-epsd; }
else
  { epsq = 0; }

switch(a)
{
case 1:
  if (k>=j)
    { d = ((A[i][j]/dt(i,j))*epsd*(p[i][k]-p[i][j])
           -epsq*(p[i][n]-3*p[i][l]+3*p[i][j]-p[i][m])))*H[i][j]; }
  else
    { d = ((A[i][j]/dt(i,j))*epsd*(p[i][k]-p[i][j])
           -epsq*(p[i][o]-3*p[i][m]+3*p[i][j]-p[i][l])))*H[i][j]; }
  return (d);
case 2:
  if (k>=j)
    { d = ((A[i][j]/dt(i,j))*epsd*(u[i][k]-u[i][j])
           -epsq*(u[i][n]-3*u[i][l]+3*u[i][j]-u[i][m]))*(ro*H[i][j]); }
  else
    { d = ((A[i][j]/dt(i,j))*epsd*(u[i][k]-u[i][j])
           -epsq*(u[i][o]-3*u[i][m]+3*u[i][j]-u[i][l]))*(ro*H[i][j]); }
  return (d);
case 3:
  if (k>=j)
    { d = ((A[i][j]/dt(i,j))*epsd*(v[i][k]-v[i][j])
           -epsq*(v[i][n]-3*v[i][l]+3*v[i][j]-v[i][m]))*(ro*H[i][j]); }
  else
    { d = ((A[i][j]/dt(i,j))*epsd*(v[i][k]-v[i][j])
           -epsq*(v[i][o]-3*v[i][m]+3*v[i][j]-v[i][l]))*(ro*H[i][j]); }
  return (d);
}
return (d);
}

double ni(int i, int j)
{
double n=0, k=0, l=0;
k=i+1; l=i-1;

```

```

if(k<1) k=1; if(k>I-1) k=I-1;
if(l<1) l=1; if(l>I-1) l=I-1;
if (p[i][j]==0) p[i][j]=1e-5;
n = fabs(p[k][j]-2*p[i][j]+p[l][j])/(fabs(p[k][j])+2*fabs(p[i][j])+fabs(p[l][j]));
return (n);
}

```

### Arquivo "V\_ALOCA.C":

```

#include <alloc.h>
#include <stdlib.h>
#include <stdio.h>
#include <math.h>
#include <io.h>
#include <conio.h>
#include "volfin.h"

double **aloca2d(int nrh,int nch)
{
    int i;
    double **m;
    m = (double **) malloc((unsigned) (nrh+1)*sizeof(double *));
    if (!m) erro();
    m -= 1;
    for(i=1;i<=nrh;i++)
    {
        m[i] = (double *) malloc((unsigned) (nch+1)*sizeof(double));
        if (!m[i]) erro();
        m[i] -= 1;
    }
    return (m);
}

void desaloca(m,nrh)
double **m;
{
    int i;
    for(i=nrh;i>=1;i--) free(m[i]);
    free(m);
}

double dxy(int i,int j, int k, int l)
{
    double d=0.0;
    d = sqrt((x[i][j]-x[k][l])*(x[i][j]-x[k][l])+

```

```
        (y[i][j]-y[k][l])*(y[i][j]-y[k][l]));
return (d);
}

void erro()
{
printf("\n nao alocou a memoria");
}
```

### Arquivo "V\_DT.C":

```
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <conio.h>
#include "volfin.h"
double dt(int i, int j)
{
double t,CFL,q,ds,a,b;
CFL = 2*sqrt(2);
a = .25*(dxy(i,j,i+1,j)+dxy(i,j+1,i+1,j+1));
b = .25*(dxy(i,j,i,j+1)+dxy(i+1,j,i+1,j+1));
if (a < b) {ds = a;} else {ds = b;}
t = CFL*ds/c(i,j);
return (t);
}

double beta(int i,int j)
{
double bt;
bt = ro*c(i,j)*c(i,j);
return (bt);
}

double c(int i, int j)
{
double cs;
cs=sqrt(((fabs(p[i+1][j]-p[i][j])))/ro);
return (cs);
}
```

### Arquivo "V\_AREA.C":

```
#include <stdio.h>
#include <math.h>
#include <io.h>
#include <stdlib.h>
#include <conio.h>
#include "volfin.h"

double UNS(int i,int j)
{
double a;
a=(YET[i][j]/H[i][j])*u[i][j]-(XET[i][j]/H[i][j])*v[i][j];
return (a);
}

double VNS(int i,int j)
{
double a;
a=-(YQS[i][j]/H[i][j])*u[i][j]+(XQS[i][j]/H[i][j])*v[i][j];
return (a);
}

double uns(int i,int j)
{
double a;
a = XQS[i][j]*U[i][j]+XET[i][j]*V[i][j];
return (a);
}

double vns(int i,int j)
{
double a;
a = YQS[i][j]*U[i][j]+YET[i][j]*V[i][j];
return (a);
}
```

## **BIBLIOGRAFIA**

---

- ADKINS, D. R. and BRENNEM, C. E. "Analysis of Hydrodynamic Radial Forces on Centrifugal Pump Impellers". **Transactions of the ASME, Journal of Fluids Engineering**, New York, v. 110, n. 1, p. 20-28, Mar., 1988.
- AGOUZOUL, M., REGGIO, M. and CAMARERO, R. "Calculation of Turbulent Flows in a Hydraulic Turbine Draft Tube". **Transactions of the ASME, Journal of Fluids Engineering**, New York, v. 112, n. 3, p. 257-263, Sept., 1990.
- ANDERSON Jr, J. D., "Computational Fluid Dynamics", Mc Graw-Hill, 1995.
- ANDERSON, W. Kyle, THOMAS, James L. and WHITFIELD, David L. "Multigrid Acceleration of the Flux-Split Euler Equations". **AIAA Journal**, Washington, v. 26, n.6, p. 649-654, June, 1988.



- ARNONE, A. and STECCO, S. S. "Multigrid Calculation of Incompressible Flows for Turbomachinery Applications". **XXIV IAHR Congress**, Madrid, p. C361-C368, 1991.
- ARTHUR, M. T., BLAYLOCK, T. A. and ANDERSON, J. M. "Cell-Vertex, Multigrid Euler Scheme for use with Multiblock Grids". **AIAA Journal**, Washington, v. 29, n. 4, p. 507-514, Apr., 1991.
- BATINA, John T. "Unsteady Euler Airfoil Solutions Using Unstructured Dynamic Meshes". **AIAA Journal**, Washington, v. 28, n. 8, p.1381-1388, Aug., 1990.
- BELK, Dave M., JANUS, J. Mark and WHITFIELD, David L. "Three-Dimensional Unsteady Euler Equations Solution on Dynamic Grids". **AIAA Journal**, Washington, v. 25, n. 9, p. 1160-1161, Sept., 1987.
- BOSMAN, C. and EL-SHAARAWI, M. A. I. "Quasi-Three-Dimensional Numerical Solution of Flow in Turbomachines". **Transactions of the ASME, Journal of Fluids Engineering**, New York, v. 99, n. 1, p. 132-140, Mar., 1977.
- BRADSHAW, P. "Effects of Streamline Curvature on Turbulent Flow". **AGARD**, n. 169, 1973.
- BRENNEIS, A. and EBERLE, A. "Application of an Implicit Relaxation Method Solving the Euler Equations for Time-Accurate Unsteady Problems". **Transactions of the ASME, Journal of Fluids Engineering**, New York, v. 112, n. 4, p. 510-520, Dec., 1990.
- CABUK, H., SUNG, C.-H and MODI, V. "Explicit Runge-Kutta Method for Three-Dimensional Internal Incompressible Flows". **AIAA Journal**, Washington, v. 30, n. 8, p. 2024-2031, Aug., 1992.
- CAUDIU, E., GRIMBERT, I., EL GHAZZANI, E. M., VERRY, A. and PLILIBERT "3D Flow Computation in Turbomachinery". **13th IAHR**

- SYMPOSIUM, Section on Hydraulic Machinery and Cavitation, Montréal, 1986.**
- CAUGHEY, D. A. and JAMESON, Antony. "Numerical Calculation of Transonic Potential Flow about Wing-Body Combinations". **AIAA Journal**, Washington, v. 17, n.2, p. 175-181, Feb., 1979.
- CAUGHEY, D.A. and JAMESON, Antony. "Progress in Finite-Volume Calculations for Wing-Fuselage Combinations". **AIAA Journal**, Washington, v. 18, n. 11, p. 1281-1288, Nov., 1980.
- CAUGHEY, D. A. "Multi-Grid Calculation of Three-Dimensional Transonic Potential Flows". **Applied Mathematics and Computation**, n.13, p. 241-260, 1983.
- CEBECI, Tuncer, Chang, K. C., LI, C. and WHITELAW, J. H. "Turbulence Models for Wall Boundary Layers". **AIAA Journal**, Washington, v. 24, n.3, p. 359-360, Mar., 1986.
- CHAKRAVARTHY, Sukumar R. "Euler Equations - Implicit Schemes and Boundary Conditions". **AIAA Journal**, Washington, v. 21, n. 5, p. 699-706, May, 1983.
- CHAKRABARTTY, Sunil Kumar. "Vertex-Based Finite-Volume Solution of the Two-Dimensional Navier-Stokes Equations". **AIAA Journal**, Washington, v. 28, n. 10, p. 1829-1831, Oct., 1990.
- CHENEY, Ward and KINCAID, David. "Numerical Mathematics and Computing". 2nd ed. California: Brooks/Cole Publishing Company, 1985.
- CHORIN, A. J. "Random Choice Solution of Hyperbolic Systems". **Journal of Computational Physics**, v. 22, p. 517-533, 1976.

- DANG, Thong Q. and CHEN, Lee-Tzong. "Euler Correction Method for Two- and Three-Dimensional Transonic Flows". **AIAA Journal**, Washington, v. 27, n.10, p. 1377-1386, Oct., 1989.
- DANNENHOFFER III, John F., and GILES, Michael B. "Accelerated Convergence of Euler Solutions Using Time Inclining". **AIAA Journal**, Washington, v. 28, n. 8, p. 1457-1463, Aug., 1990.
- EDWARDS, David E., CARTER, James E. and SMITH, Frank T. "Analysis of Three-Dimensional Separated Flow with the Boundary-Layer Equations". **AIAA Journal**, Washington, v. 25, n. 3, p. 380-387, Mar., 1987.
- ERIKSSON, Lars E. and RIZZI, Arthur. "Computer-Aided Analysis of the Convergence to Steady State of Discrete Approximations to the Euler Equations". **Journal of Computational Physics**, n. 57, p. 90-128, 1985.
- FANG, Zhigang and SABER, Aaron Jaan. "Marching Control Volume Finite-Element Calculation for Developing Entrance Flow". **AIAA Journal**, Washington, v. 25, n.2, p. 346-348, Feb., 1987.
- GNOFO, Peter A. "A Vectorized, Finite-Volume, Adaptative-Grid Algorithm for Navier-Stokes Calculations". **Applied Mathematics and Computation, Numerical Grid Generation**, n. 12, p. 819-835, 1982.
- GÖDE, E. "3-D Computational of the Flow in a Francis Runner". **Sulzer Technical Review**, n. 4, p. 31-35, 1987.
- GOEDE, E. and PESTALOZZI, J. "Flow Calculation in a Bulb Turbine". **13th IAHR SYMPOSIUM, Section on Hydraulic Machinery and Cavitation**, Montréal, p. 1-15, 1986.

- HIRSCH, Ch. and WARZEE, G. "A Finite-Element Method for Through Flow Calculations in Turbomachines". **Transactions of the ASME, Journal of Fluids Engineering**, New York, v. 98, . n. 3, p. 403-421, Sept., 1976.
- HIRSCH, Charles. "Numerical Computational of Internal and External Flows". Volume 2: Computational Methods for Inviscid and Viscous Flows. Chichester: Wiley-Interscience, 1990.
- HOFFMANN, K. A., CHIANG, S. T. "Computational Fluid Dynamics for Engineers". 2nd Ed., Vol. 1 and 2, Engineering Education System, 1993.
- HUMPHREY, J. A. C., TAYLOR, A. M. K., WHITELOW, J. H. "Laminar Flow in a Square Duct of Strong Curvature". **Journal of Fluid Mechanics**, v. 83, p. 509-527, 1977.
- HUMPHREY, J. A. C., WHITELOW, J. H., YEE, G. "Turbulent Flow in a Square Duct with Strong Curvature". **Journal of Fluid Mechanics**, v. 103, p. 443-463, 1981.
- JAMESON, A. and CAUGHEY, D. E. "A Finite Volume Method for Transonic Potential Flow Calculations". **Proc. AIAA Third Computational Fluid Dynamics Conference**, p. 35-54, 1977.
- JAMESON, Antony. "Solution of the Euler Equations for Two Dimensional Transonic Flow by a Multigrid Method". **Applied Mathematics and Computation**, n. 13, p. 327-355, 1983.
- JAMESON, A. "The Evolution of Computational Methods in Aerodynamics". **Transactions of the ASME, Journal of Applied Mechanics**, New York, v. 50, p. 1052-1070, Dec., 1983.

- JOU, Wen-Huei. "Finite Volume Calculation of Three-Dimensional Potential Flow Around a Propeller". **AIAA Journal**, Washington, v. 21, n. 10, p. 1360-1364, Oct., 1983.
- KRAUS, Allan D. "**Matrices for Engineers**". New York: Hemisphere Publishing Corporation, 1987.
- KRAMER, J. J. "Analysis of Incompressible, Nonviscous Blade to Blade Flow in Rotating Blade Rows". **Transactions of the ASME**, p. 263-275, Feb., 1958.
- KRAMER, J. J. "Theoretical Analysis of Incompressible Flow Through a Radial Inlet Centrifugal Impeller at Various Weight Flows. II - Solution in Leading Edge Region by Relaxation Methods". **NACA**, TN 3449, 1955.
- LAKSHMINARAYANA, B. "An Assessment of Computational Fluid Dynamic Techniques in the Analysis and Design of Turbomachinery - The 1990 Freeman Scholar Lecture". **Transactions of the ASME, Journal of Fluids Engineering**, New York, v. 113, n. 3, p. 315-352, Sept., 1991.
- LAX, P. D. "Weak Solutions of Nonlinear Hyperbolic Equations and Their Numerical Computation". **Comm. on Pure and Applied Math.**, v. VII, p. 159-193, 1954.
- LONG, Lyle N., KHAN, M. M. S. and SHARP, H. Thomas. "Massively Parallel Three-Dimensional Euler/Navier-Stokes Method". **AIAA Journal**, Washington, v. 29, n. 5, p. 657-666, May, 1991.
- MAVRIPLIS, D. J. "Multigrid Solutions of the Two-Dimensional Euler Equations on Unstructured Triangular Meshes". **AIAA Journal**, Washington, v. 26, n. 7, p. 824-831, July, 1988.

- MAVRIPLIS, Dimitri J. "Accurate Multigrid Solution of the Euler Equations on Unstructured and Adaptive Meshes". **AIAA Journal**, Washington, v. 28, n. 2, p. 213-221, Feb., 1990.
- MORETTI, Gino. "A Technique for Integrating Two-Dimensional Euler Equations". **Computers & Fluids**, v. 15, n. 1, p. 59-75, 1987.
- MORETTI, G. "Efficient Euler Solver with Many Applications". **AIAA Journal**, Washington, v. 26, n. 6, p. 655-660, June, 1988.
- PATANKAR, S. V. "Numerical Heat Transfer and Fluid Flow". Series in Computational Methods in Mechanics and Thermal Sciences. New York: Hemisphere Publishing Corporation, 1980.
- PRIAN, V. D., KRAMER, J. J., WU, C. H. "Theoretical Analysis of Incompressible Flow Through a Radial Inlet Centrifugal Impeller at Various Weight Flows. I - Solution by a Matrix Method and Comparison with an Approximate Method". **NACA**, TN 3448, 1955.
- RICHTMYER, R. D. "Difference Methods for Initial Value Problems". New York: Interscience Publishers, Wiley, 1957.
- RIZZI, A. and ERIKSSON, L. E. "Computation of Inviscid Incompressible Flow with Rotation". **Journal of Fluid Mechanics**, v. 153, p. 275-312, 1985.
- ROACHE, Patrick J. "Computational Fluid Dynamics". Albuquerque: Hermosa Publishers, 1972.
- SANTAL, O. and AVELLAN, F. "Hydraulic Analysis of Flow Computation Results". **16th IAHR SYMPOSIUM, Section on Hydraulic Machinery and Cavitation**, São Paulo, p. 545-554, Sept., 1992.

- SCHILLING, R. "Numerical Calculation of the Quasi-3D Incompressible Inviscid Flow in Turbomachines". **11th IAHR SYMPOSIUM, Section on Hydraulic Machinery and Cavitation**, Amsterdam, p. 1-13, 1982.
- SHOUP, Terry E. "Applied Numerical Methods for the Micro-computer". New Jersey: Prentice-Hall, 1984.
- SICLARI, M. J. and GUIDICE, P. Del. "Hybrid Finite Volume Approach to Euler Solutions for Supersonic Flows". **AIAA Journal**, Washington, v. 28, n. 1, p. 66-74, Jan., 1990.
- TAYLOR, A. M. K. P., WHITELAW, J. H., YIANNESKIS, X. "Curved Ducts with Strong Secondary Motion: Velocity Measurements of Developing Laminar and Turbulent Flow". **Transactions of the ASME, Journal of Fluids Engineering**, New York, v. 104, n. 3, p.350-359, Sept., 1982.
- THOMPSON, J. F., THAMES, F. C. and MASTIN, C. W. "Automatic Numerical Generation of Body-Fitted Curvilinear Coordinate System for Field Containing Any Number of Arbitrary Two-Dimensional Bodies". **Journal of Computational Physics**, v. 15, p. 299-319, 1974.
- TURKEL, Eli and VATSA, Veer N. "Effect of Artificial Viscosity on Three-Dimensional Flow Solutions". **AIAA Journal**, Washington, v. 32, n. 1, p. 39-45, Jan., 1994.
- von LAVANTE, E., HAERTL, A. and CLAES, D. "Numerical Solutions of Euler Equations using Simplified Flux Vector Splitting". **AIAA Journal**, Washington, v. 25, n. 8, p. 1050-1051, Aug., 1987.
- WU, CHUNG-HUA "A General Theory of Three-Dimensional Flow in Subsonic and Supersonic Turbomachines of Axial, Radial, and Mixed Flow Types". **NACA, TN 2604**, 1952.

## COORDENADAS DOS PONTOS DA MALHA DO BOCAL

A tabela abaixo mostra o arquivo de entrada com as coordenadas cartesianas do bocal, utilizado para cálculo no Cap. 5.

```

21 11
0.000000e+000 0.000000e+000 0.000000e+000
0.000000e+000 9.999999e-002 0.000000e+000
0.000000e+000 2.000000e-001 0.000000e+000
0.000000e+000 3.000000e-001 0.000000e+000
0.000000e+000 4.000000e-001 0.000000e+000
0.000000e+000 5.000000e-001 0.000000e+000
0.000000e+000 6.000000e-001 0.000000e+000
0.000000e+000 7.000000e-001 0.000000e+000
0.000000e+000 8.000001e-001 0.000000e+000
0.000000e+000 9.000001e-001 0.000000e+000
0.000000e+000 1.000000e+000 0.000000e+000
1.000000e-001 0.000000e+000 0.000000e+000
1.000000e-001 1.000000e-001 0.000000e+000
1.000000e-001 2.000000e-001 0.000000e+000
1.000000e-001 3.000000e-001 0.000000e+000
1.000000e-001 4.000000e-001 0.000000e+000
1.000000e-001 5.000000e-001 0.000000e+000
1.000000e-001 6.000000e-001 0.000000e+000
1.000000e-001 7.000000e-001 0.000000e+000
1.000000e-001 8.000001e-001 0.000000e+000
1.000000e-001 9.000001e-001 0.000000e+000
1.000000e-001 1.000000e+000 0.000000e+000
2.000000e-001 2.000000e-002 0.000000e+000
1.971445e-001 1.178727e-001 0.000000e+000
1.949236e-001 2.144970e-001 0.000000e+000
1.933373e-001 3.101849e-001 0.000000e+000
    
```



1.923855e-001 4.052485e-001 0.000000e+000  
1.920682e-001 5.000000e-001 0.000000e+000  
1.923855e-001 5.947515e-001 0.000000e+000  
1.933373e-001 6.898152e-001 0.000000e+000  
1.949236e-001 7.855031e-001 0.000000e+000  
1.971445e-001 8.821273e-001 0.000000e+000  
2.000000e-001 9.800001e-001 0.000000e+000  
3.000001e-001 4.000001e-002 0.000000e+000  
2.955129e-001 1.349429e-001 0.000000e+000  
2.920229e-001 2.279239e-001 0.000000e+000  
2.895301e-001 3.194334e-001 0.000000e+000  
2.880344e-001 4.099619e-001 0.000000e+000  
2.875358e-001 5.000000e-001 0.000000e+000  
2.880344e-001 5.900381e-001 0.000000e+000  
2.895301e-001 6.805667e-001 0.000000e+000  
2.920229e-001 7.720762e-001 0.000000e+000  
2.955129e-001 8.650572e-001 0.000000e+000  
3.000001e-001 9.600001e-001 0.000000e+000  
4.000000e-001 6.000001e-002 0.000000e+000  
3.949167e-001 1.513339e-001 0.000000e+000  
3.909630e-001 2.404452e-001 0.000000e+000  
3.881389e-001 3.278895e-001 0.000000e+000  
3.864445e-001 4.142226e-001 0.000000e+000  
3.858797e-001 5.000000e-001 0.000000e+000  
3.864445e-001 5.857774e-001 0.000000e+000  
3.881389e-001 6.721105e-001 0.000000e+000  
3.909630e-001 7.595549e-001 0.000000e+000  
3.949167e-001 8.486662e-001 0.000000e+000  
4.000000e-001 9.400001e-001 0.000000e+000  
5.000000e-001 8.000001e-002 0.000000e+000  
4.951677e-001 1.671692e-001 0.000000e+000  
4.914092e-001 2.522257e-001 0.000000e+000  
4.887246e-001 3.356974e-001 0.000000e+000  
4.871139e-001 4.181128e-001 0.000000e+000  
4.865769e-001 5.000000e-001 0.000000e+000  
4.871139e-001 5.818872e-001 0.000000e+000  
4.887246e-001 6.643026e-001 0.000000e+000  
4.914092e-001 7.477745e-001 0.000000e+000  
4.951677e-001 8.328309e-001 0.000000e+000  
5.000000e-001 9.200001e-001 0.000000e+000  
6.000001e-001 1.000000e-001 0.000000e+000  
5.960777e-001 1.825724e-001 0.000000e+000  
5.930271e-001 2.634299e-001 0.000000e+000  
5.908480e-001 3.430012e-001 0.000000e+000  
5.895405e-001 4.217149e-001 0.000000e+000  
5.891047e-001 5.000000e-001 0.000000e+000  
5.895405e-001 5.782850e-001 0.000000e+000  
5.908480e-001 6.569989e-001 0.000000e+000  
5.930271e-001 7.365702e-001 0.000000e+000  
5.960777e-001 8.174276e-001 0.000000e+000  
6.000001e-001 9.000001e-001 0.000000e+000  
7.000000e-001 1.200000e-001 0.000000e+000  
6.974583e-001 1.976669e-001 0.000000e+000  
6.954815e-001 2.742226e-001 0.000000e+000  
6.940694e-001 3.499448e-001 0.000000e+000  
6.932222e-001 4.251113e-001 0.000000e+000  
6.929398e-001 5.000000e-001 0.000000e+000  
6.932222e-001 5.748886e-001 0.000000e+000  
6.940694e-001 6.500553e-001 0.000000e+000  
6.954815e-001 7.257774e-001 0.000000e+000  
6.974583e-001 8.023332e-001 0.000000e+000  
7.000000e-001 8.800001e-001 0.000000e+000

8.000000e-001 1.400000e-001 0.000000e+000  
7.991214e-001 2.125762e-001 0.000000e+000  
7.984381e-001 2.847683e-001 0.000000e+000  
7.979500e-001 3.566723e-001 0.000000e+000  
7.976571e-001 4.283842e-001 0.000000e+000  
7.975594e-001 5.000000e-001 0.000000e+000  
7.976570e-001 5.716158e-001 0.000000e+000  
7.979500e-001 6.433278e-001 0.000000e+000  
7.984381e-001 7.152318e-001 0.000000e+000  
7.991214e-001 7.874239e-001 0.000000e+000  
8.000000e-001 8.600001e-001 0.000000e+000  
9.000000e-001 1.600000e-001 0.000000e+000  
9.008785e-001 2.274238e-001 0.000000e+000  
9.015619e-001 2.952317e-001 0.000000e+000  
9.020500e-001 3.633278e-001 0.000000e+000  
9.023429e-001 4.316158e-001 0.000000e+000  
9.024405e-001 5.000000e-001 0.000000e+000  
9.023429e-001 5.683842e-001 0.000000e+000  
9.020500e-001 6.366723e-001 0.000000e+000  
9.015619e-001 7.047683e-001 0.000000e+000  
9.008785e-001 7.725763e-001 0.000000e+000  
9.000000e-001 8.400000e-001 0.000000e+000  
9.999999e-001 1.800000e-001 0.000000e+000  
1.002542e+000 2.423331e-001 0.000000e+000  
1.004519e+000 3.057775e-001 0.000000e+000  
1.005931e+000 3.700553e-001 0.000000e+000  
1.006778e+000 4.348887e-001 0.000000e+000  
1.007060e+000 5.000000e-001 0.000000e+000  
1.006778e+000 5.651113e-001 0.000000e+000  
1.005930e+000 6.299447e-001 0.000000e+000  
1.004518e+000 6.942227e-001 0.000000e+000  
1.002542e+000 7.576670e-001 0.000000e+000  
9.999999e-001 8.200001e-001 0.000000e+000  
1.100000e+000 2.000000e-001 0.000000e+000  
1.103922e+000 2.574276e-001 0.000000e+000  
1.106973e+000 3.165701e-001 0.000000e+000  
1.109152e+000 3.769989e-001 0.000000e+000  
1.110460e+000 4.382850e-001 0.000000e+000  
1.110895e+000 5.000000e-001 0.000000e+000  
1.110460e+000 5.617149e-001 0.000000e+000  
1.109152e+000 6.230011e-001 0.000000e+000  
1.106973e+000 6.834300e-001 0.000000e+000  
1.103922e+000 7.425724e-001 0.000000e+000  
1.100000e+000 8.000000e-001 0.000000e+000  
1.200000e+000 2.200000e-001 0.000000e+000  
1.204832e+000 2.728307e-001 0.000000e+000  
1.208591e+000 3.277744e-001 0.000000e+000  
1.211275e+000 3.843026e-001 0.000000e+000  
1.212886e+000 4.418872e-001 0.000000e+000  
1.213423e+000 5.000000e-001 0.000000e+000  
1.212886e+000 5.581128e-001 0.000000e+000  
1.211275e+000 6.156974e-001 0.000000e+000  
1.208591e+000 6.722257e-001 0.000000e+000  
1.204832e+000 7.271693e-001 0.000000e+000  
1.200000e+000 7.800000e-001 0.000000e+000  
1.300000e+000 2.400000e-001 0.000000e+000  
1.305083e+000 2.886662e-001 0.000000e+000  
1.309037e+000 3.395549e-001 0.000000e+000  
1.311861e+000 3.921106e-001 0.000000e+000  
1.313555e+000 4.457774e-001 0.000000e+000  
1.314120e+000 5.000000e-001 0.000000e+000  
1.313555e+000 5.542226e-001 0.000000e+000

1.311861e+000 6.078895e-001 0.000000e+000  
1.309037e+000 6.604452e-001 0.000000e+000  
1.305083e+000 7.113339e-001 0.000000e+000  
1.300000e+000 7.600001e-001 0.000000e+000  
1.400000e+000 2.600000e-001 0.000000e+000  
1.404487e+000 3.050572e-001 0.000000e+000  
1.407977e+000 3.520762e-001 0.000000e+000  
1.410470e+000 4.005667e-001 0.000000e+000  
1.411966e+000 4.500381e-001 0.000000e+000  
1.412464e+000 5.000000e-001 0.000000e+000  
1.411966e+000 5.499619e-001 0.000000e+000  
1.410470e+000 5.994334e-001 0.000000e+000  
1.407977e+000 6.479238e-001 0.000000e+000  
1.404487e+000 6.949430e-001 0.000000e+000  
1.400000e+000 7.400001e-001 0.000000e+000  
1.500000e+000 2.800000e-001 0.000000e+000  
1.502856e+000 3.221273e-001 0.000000e+000  
1.505077e+000 3.655031e-001 0.000000e+000  
1.506663e+000 4.098152e-001 0.000000e+000  
1.507615e+000 4.547515e-001 0.000000e+000  
1.507932e+000 5.000000e-001 0.000000e+000  
1.507615e+000 5.452484e-001 0.000000e+000  
1.506663e+000 5.901849e-001 0.000000e+000  
1.505077e+000 6.344970e-001 0.000000e+000  
1.502856e+000 6.778727e-001 0.000000e+000  
1.500000e+000 7.200000e-001 0.000000e+000  
1.600000e+000 3.000000e-001 0.000000e+000  
1.600000e+000 3.400000e-001 0.000000e+000  
1.600000e+000 3.800000e-001 0.000000e+000  
1.600000e+000 4.200000e-001 0.000000e+000  
1.600000e+000 4.600000e-001 0.000000e+000  
1.600000e+000 5.000000e-001 0.000000e+000  
1.600000e+000 5.400000e-001 0.000000e+000  
1.600000e+000 5.800000e-001 0.000000e+000  
1.600000e+000 6.200001e-001 0.000000e+000  
1.600000e+000 6.600001e-001 0.000000e+000  
1.600000e+000 7.000000e-001 0.000000e+000  
1.700000e+000 3.000000e-001 0.000000e+000  
1.700000e+000 3.400000e-001 0.000000e+000  
1.700000e+000 3.800000e-001 0.000000e+000  
1.700000e+000 4.200000e-001 0.000000e+000  
1.700000e+000 4.600000e-001 0.000000e+000  
1.700000e+000 5.000000e-001 0.000000e+000  
1.700000e+000 5.400000e-001 0.000000e+000  
1.700000e+000 5.800000e-001 0.000000e+000  
1.700000e+000 6.200001e-001 0.000000e+000  
1.700000e+000 6.600001e-001 0.000000e+000  
1.700000e+000 7.000000e-001 0.000000e+000  
1.800000e+000 3.000000e-001 0.000000e+000  
1.800000e+000 3.400000e-001 0.000000e+000  
1.800000e+000 3.800000e-001 0.000000e+000  
1.800000e+000 4.200000e-001 0.000000e+000  
1.800000e+000 4.600000e-001 0.000000e+000  
1.800000e+000 5.000000e-001 0.000000e+000  
1.800000e+000 5.400000e-001 0.000000e+000  
1.800000e+000 5.800000e-001 0.000000e+000  
1.800000e+000 6.200001e-001 0.000000e+000  
1.800000e+000 6.600001e-001 0.000000e+000  
1.800000e+000 7.000000e-001 0.000000e+000  
1.900000e+000 3.000000e-001 0.000000e+000  
1.900000e+000 3.400000e-001 0.000000e+000  
1.900000e+000 3.800000e-001 0.000000e+000

1.900000e+000	4.200000e-001	0.000000e+000
1.900000e+000	4.600000e-001	0.000000e+000
1.900000e+000	5.000000e-001	0.000000e+000
1.900000e+000	5.400000e-001	0.000000e+000
1.900000e+000	5.800000e-001	0.000000e+000
1.900000e+000	6.200001e-001	0.000000e+000
1.900000e+000	6.600001e-001	0.000000e+000
1.900000e+000	7.000000e-001	0.000000e+000
2.000000e+000	3.000000e-001	0.000000e+000
2.000000e+000	3.400000e-001	0.000000e+000
2.000000e+000	3.800000e-001	0.000000e+000
2.000000e+000	4.200000e-001	0.000000e+000
2.000000e+000	4.600000e-001	0.000000e+000
2.000000e+000	5.000000e-001	0.000000e+000
2.000000e+000	5.400000e-001	0.000000e+000
2.000000e+000	5.800000e-001	0.000000e+000
2.000000e+000	6.200001e-001	0.000000e+000
2.000000e+000	6.600001e-001	0.000000e+000
2.000000e+000	7.000000e-001	0.000000e+000