

3.Solução Numérica

As simulações numéricas foram realizadas no programa comercial *POLYFLOW* da empresa *ANSYS*[®]. O software foi concebido para simular, principalmente, fluidos não-newtonianos, contemplando em seu acervo diferentes equações constitutivas. O programa inclui diferentes funções para a dependência da viscosidade com a taxa de cisalhamento incluindo modelos para fluidos viscoplásticos. Para fluidos viscoelásticos o programa divide em modelos viscoelásticos diferenciais, como Oldroyd-B e Maxwell, e em modelos viscoelásticos integrais, como Doi-Edwards e KBKZ [12]. O *POLYFLOW* ainda permite a simulação de escoamentos com gradientes de temperatura e em meios porosos.

Para simular qualquer problema no *POLYFLOW*, pelo menos os seguintes passos precisam ser implementados:

- Geração e leitura da malha;
- Definição dos parâmetros numéricos da simulação, condições de contorno, dados materiais e método de interpolação no arquivo de dados;
- Implementação dos cálculos para a solução do problema.

O *POLYFLOW* não possui capacidade de geração de malha. Assim, o desenvolvimento da malha deve ser realizado em programa específico de pré-processamento. O padrão é o *GAMBIT*, também da *ANSYS*[®], que foi utilizado neste trabalho.

Após a obtenção dos resultados da simulação, a visualização e o pós-processamento devem ser realizados em outro programa. Nesta dissertação utilizou-se o *TECPLOT* da empresa *TECPLOT, Inc*[®].

Os principais tópicos para a compreensão de como foram efetuadas as simulações numéricas serão descritos ao longo deste capítulo. Para maiores detalhes consultar o manual do programa [12] e referências sobre simulação numérica de escoamentos não-newtonianos [9,10,11].

Um esquema do procedimento utilizado na simulação numérica do escoamento deste trabalho é apresentado na figura 3.1.

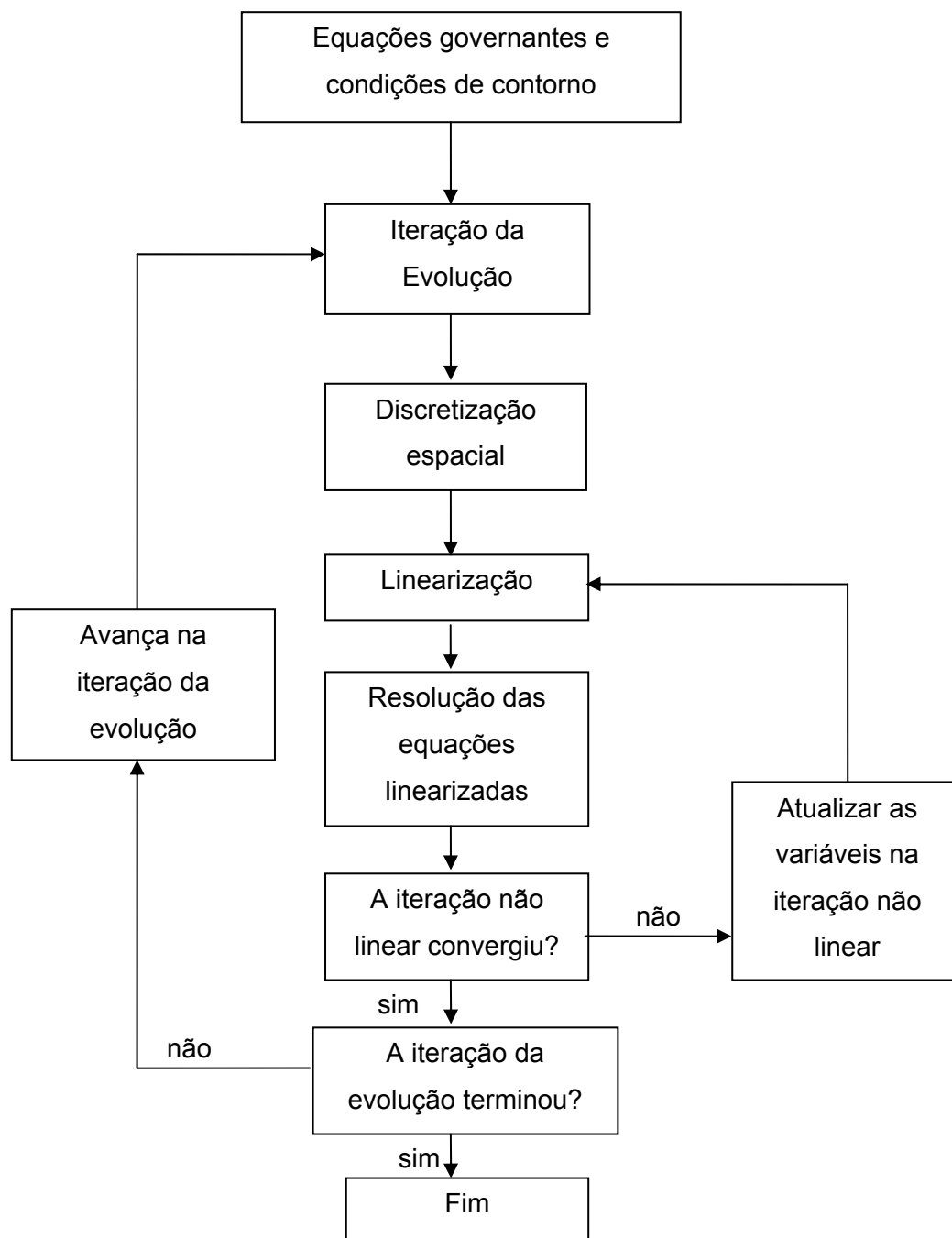


Figura 3.1: Seqüência de passos para solução numérica de escoamentos não newtonianos

Seguindo o esquema da figura 3.1, as equações governantes e condições de contorno foram discutidas no capítulo anterior.

A evolução citada no esquema da figura 3.1 é um método de discretização do *POLYFLOW* indicado quando se têm equações constitutivas não lineares (indicado para modelos viscoelásticos [12]) e que não convergem quando apenas se lineariza as funções. Este método é similar a discretização temporal quando se tem um escoamento transiente e será tratado em maiores detalhes na seção Parâmetros de Evolução.

A discretização espacial consiste na aproximação da velocidade, pressão e tensão em valores prescritos de r e x (coordenadas cilíndricas). Esta aproximação pode ser realizada de diversas maneiras, entre elas o método dos elementos finitos, que foi utilizado neste trabalho. Neste método, as variáveis são aproximadas por funções algébricas de r e x no espaço para cada polígono da discretização geométrica (neste trabalho utilizou-se o retângulo, como descrito na seção modelagem da malha). Os coeficientes da representação funcional das variáveis são calculados e representam a solução do problema.

O próximo passo é a linearização do sistema algébrico não linear para ser resolvido pelo programa computacional. O *POLYFLOW*, para o modelo diferencial viscoelástico, usa para a linearização das equações o clássico método de Newton [6,9].

A linearização pelo método de Newton leva a um processo iterativo para obtenção da solução como exposto no *loop* de iteração linear mostrado na figura 3.1. Em cada iteração do *loop* citado são resolvidos os sistemas de equações lineares. O *POLYFLOW* resolve o sistema de equações completamente acopladas, ou seja, a tensão, velocidade e pressão são calculadas simultaneamente. As matrizes destas equações são esparsas, o que leva a métodos para se estruturá-las e assim reduzir drasticamente o tempo de cálculo necessário para obter a solução. No *POLYFLOW*, a solução do sistema de equações lineares é realizada pela eliminação gaussiana.

Por fim, o programa avança na iteração da evolução. Quando alcançado o valor final especificado para o parâmetro evolutivo significa que o programa convergiu e a simulação está pronta para a fase de pós-processamento.

3.1. Resolução do Problema pelo *POLYFLOW*

Depois de mostrado de forma geral como se soluciona uma simulação de escoamento de fluidos não Newtonianos, será mostrado como se implementa e simula no *POLYFLOW*.

3.1.1. Definição da geometria e dependência temporal do escoamento

Inicialmente, para simular o problema, deve-se definir a geometria (no presente caso 2D axissimétrico) e se o escoamento é transiente ou não. Neste ponto, o *POLYFLOW* permite as opções:

- Permanente: O escoamento não varia nenhuma de suas propriedades ao longo de tempo;
- Transiente: O escoamento tem suas propriedades dependentes do tempo;
- Evolutivo: É uma técnica computacional utilizada pelo *POLYFLOW* que reduz a não-linearidade dos escoamentos, tais como as equações constitutivas de fluidos viscoelásticos.

Neste trabalho usou-se o método Evolutivo, tratado no tópico: Parâmetros Evolutivos.

3.1.2. Escolha do modelo viscoelástico

O passo seguinte é a definição da relação da tensão com a taxa de deformação. Aqui, como a equação constitutiva a ser simulada é inédita, ela não se encontra entre as opções disponíveis do programa. Para resolver este problema, escolheu-se a equação de Oldroyd-B (modelo viscoelástico diferencial e isotérmico) e substituiu-se as constantes η_0 , λ_1 e λ_2 desta equação (eq. 2.10) pelas funções da equação de Souza Mendes, eqs. 2.16a, 2.16b e 2.16c respectivamente.

3.1.2.1. Equação de Oldroyd-B

A equação de Oldroyd-B (eq. 2.10), no *POLYFLOW*, é apresentada como uma tensão total τ representada pela soma de uma tensão viscoelástica τ_1 , e uma puramente viscosa τ_2 :

$$\tau = \tau_1 + \tau_2 \quad (3.1)$$

$$\tau_1 + \lambda \ddot{\tau}_1 = 2\eta_1 \mathbf{D} \quad (3.1a)$$

$$\tau_2 = 2\eta_2 \mathbf{D} \quad (3.1b)$$

onde:

$$\mathbf{D} = \frac{\dot{\gamma}}{2} \quad (3.1c)$$

$$\eta_1 = (1 - \eta_r)\eta \quad (3.1d)$$

$$\eta_2 = \eta_r\eta \quad (3.1e)$$

Nas equações acima, η_2 representa o fator da viscosidade correspondente a parte do solvente e η_1 a viscosidade correspondente a parte polimérica. A viscosidade η é dada pela soma das duas partes viscosidades e $\eta_r = \frac{\eta_2}{\eta_1 + \eta_2}$.

No *POLYFLOW*, os dados de entrada constantes são η , λ e η_r . A transformação da equação 3.1 para a eq. 2.10 é mostrada a seguir.

Primeiro substitui-se a eq. 3.1b na eq. 3.1:

$$\tau = \tau_1 + \tau_2 \quad \Rightarrow \quad \tau = \tau_1 + 2\eta_2 \mathbf{D} \quad \Rightarrow \quad \tau_1 = \tau - 2\eta_2 \mathbf{D} \quad \Rightarrow \quad \tau_1 = \tau - \eta_2 \dot{\gamma}$$

Substituindo este resultado na equação 3.1a tem-se:

$$(\tau - \eta_2 \dot{\gamma}) + \lambda \overbrace{(\tau - \eta_2 \dot{\gamma})} = \eta_1 \dot{\gamma} \quad (3.1f)$$

onde $\overbrace{(\tau - \eta_2 \dot{\gamma})}$ representada a derivada convectada superior de $(\tau - \eta_2 \dot{\gamma})$.

Realizando a derivação da equação 3.1f torna-se:

$$\begin{aligned} (\tau - \eta_2 \dot{\gamma}) + \lambda (\ddot{\tau} - \eta_2 \ddot{\gamma}) &= \eta_1 \dot{\gamma} \Rightarrow \\ \tau + \lambda \ddot{\tau} &= (\eta_1 + \eta_2) \dot{\gamma} + \lambda \eta_2 \ddot{\gamma} \end{aligned} \quad (3.1g)$$

Substituindo as eq. 3.1d, 3.1e na equação 3.1g se tem:

$$\tau + \lambda \ddot{\tau} = \eta \dot{\gamma} + \lambda \eta_r \eta \ddot{\gamma} \Rightarrow \tau + \lambda \ddot{\tau} = \eta (\dot{\gamma} + \lambda \eta_r \ddot{\gamma})$$

logo, com

$$\lambda = \lambda_1; \quad (3.1h)$$

$$\lambda \eta_r = \lambda_2 \quad (3.1i)$$

a eq. 3.1 fica igual a eq. 2.10, onde ainda $\lambda_2 < \lambda_1$.

3.1.2.2. Modificação da equação de Oldroyd-B para Souza e Mendes

O *POLYFLOW* permite a introdução de funções no lugar de constantes dos modelos constitutivos existentes. Isto é possível através da programação da função desejada no software livre *CLIPS* [13]. A programação do *CLIPS* é feita programando-se em um documento de texto sem formatação qualquer (bloco de notas, por exemplo) e testada previamente no *CLIPS* antes de ser introduzida no *POLYFLOW*.

No caso da equação de Souza Mendes, as eqs. 2.16a e 2.16b foram inseridas no bloco de notas como mostrado no exemplo a seguir:

```
;=====
;
; FUNCTION NEW EQUATION
;
;=====
;
;
(defglobal ?*Jump* = 1000)
(defglobal ?*n* = 0.8)
(defglobal ?*small* = (** 10 -10))
(defglobal ?*etaInf* = 1)
(defglobal ?*lam_zero* = 1)
(defglobal ?*lam_inf* = 0)

( deffunction UDFeta ( ?g )
  ( + ( * ( - 1. ( exp ( * ( + ?*Jump* 1) ( + ?g ?*small*) -1.) ) ) ) (/
  1 ( + ?g ?*small*) ) ( + 1. ( ** ( + ?g ?*small*) ?*n* ) ) ) ( *
  ?*etaInf* ( - 1 (exp(* ?*etaInf* ( ** ( + ?g ?*small*) ( - 1 ?*n*)) -
  1)))) )

( deffunction UDFtrelax1 ( ?g )
  ( + ( * ( - ?*lam_zero* ?*lam_inf* ) (exp ( * ?*Jump* ?g -1) ) )
  ?*lam_inf* )
)

; (UDFeta (** 10 -16))
; (UDFtrelax1 (** 10 -16))
```

No texto de programação, *defglobal* é a definição das constantes a serem usadas nas funções dentro do *CLIPS*. *UDFeta* é a eq. 2.16a e *UDFtrelax1* é a eq. 2.16b.

Depois de definidas as funções, a introdução destas no *POLYFLOW* se dá pela ferramenta *User Defined Function* (UDF) do *POLYFLOW* [12]. Deve-se

escolher a constante que se deseja alterar e definir qual será a função que a substituirá. Neste contexto e no caso do problema analisado, as seguintes modificações foram feitas:

- A constante λ (da eq. 3.1a) foi modificada para representar a função De_1 (da eq. 2.16b);
- A função De_2 será uma razão de De_1 dada pela fórmula $De_2 = \eta_r De_1$ mostrada na eq 3.1f; e
- A constante η (da eq. 3.1d e 3.1e) foi modificada para representar a função $\eta^*(\dot{\gamma}^*)$.

3.1.3.Determinação do modelo de interpolação

No método de elementos finitos, um dos principais parâmetros que deve ser dado atenção é na escolha do modelo de discretização espacial das variáveis a serem calculadas. Neste caso são: velocidade, pressão e tensão. O *POLYFLOW* permite a escolha destes modelos no tópico interpolação.

Em todas as simulações realizadas neste trabalho utilizou-se o modelo quadrático para velocidade e linear para a pressão, respeitando a condição de Babuska-Brezzi [10,11], e quadrático para a tensão.

3.1.4.Parâmetros numéricos

Os parâmetros numéricos utilizados nas simulações do *POLYFLOW* podem ser descritos em dois grupos:

- Parâmetros numéricos para iteração;
- Parâmetros de evolução.

3.1.4.1.Parâmetros numéricos para iteração

Os parâmetros numéricos para iteração são apenas três:

- Número máximo de iterações;
- Critério de convergência; e
- Critério de divergência.

3.1.4.2. Parâmetros de evolução

A evolução é utilizada quando existe a presença de não-linearidade no problema e quando a simulação fora realizada sem nenhuma evolução e não tiver convergido.

A análise do problema pela evolução é similar à análise de um problema transiente. A diferença é que na simulação transiente uma ou mais variáveis são dependentes do tempo t , enquanto que na evolução se escolhe uma ou mais variáveis para mudarem em função de um parâmetro S evolutivo.

Com esta análise pode-se reduzir sobremaneira a não-linearidade de um problema no início da simulação, permitindo um número maior de convergência das simulações.

Por exemplo, um fluido *power-law* (equação 3.2) torna-se newtoniano quando se tem $n = 1$.

$$\tau = \mu \dot{\gamma}^n \quad (3.2)$$

Assim, ao se realizar diversas simulações de um fluido *power-law* com as seguintes condições:

- Variando-se o expoente n , fazendo-o iniciar com valor unitário, passando por valores intermediários e terminar com o seu valor real; e
- O resultado de uma simulação é usado como condição inicial para a próxima simulação.

Obtém-se uma simulação começando com um caso newtoniano e terminando como um fluido *power-law*, ou seja, a simulação começa com uma não-linearidade mais branda e termina altamente não-linear.

No caso deste trabalho, escolheu-se o trelax para variar com a evolução (correspondente ao parâmetro λ da equação 3.1a). Com isso, De_1 e De_2 (eqs. 2.16b e 2.16c) serão afetados pelo parâmetro evolutivo S .

Quando se determina a variável que será feita a evolução, esta variável passa a ser ela mesma multiplicada por uma função evolutiva. No caso das simulações deste trabalho escolheu-se trelax para variável a ter evolução e $f(S)=S$ para a função evolutiva, então trelax passa a ser $S.\text{trelax}$.

Assim como nos problemas transientes, o problema de evolução é governado por um sistema de equações diferenciais ordinárias:

$$M(X)\dot{X} + K(X)X + F(X) = 0 \quad (3.3a)$$

$$X(S_0) = X_0 \quad (3.3b)$$

Na equação 3.3, X é o vetor das variáveis (pressão, velocidade e tensão), \dot{X} representa a derivada do vetor X com relação ao parâmetro evolutivo S . As matrizes M e K são respectivamente definidas como as matrizes massa e rigidez e o vetor F representa o vetor das condições de contorno naturais.

O POLYFLOW irá satisfazer as equações 3.3a e 3.3b em um grupo discreto de parâmetros evolutivos:

$$X_n = X(S_n) \quad (3.4a)$$

$$S_n = S_{n-1} + \Delta S_n \quad (3.4b)$$

Modificando a equação 3.3 para ficar em função de \dot{X} :

$$\dot{X} = -M^{-1}(KX + F) \quad (3.3c)$$

aproximando \dot{X} pela fórmula:

$$\dot{X} = \theta f(X_{n+1}) + (1 - \theta)f(X_n), \text{ com } 0 \leq \theta \leq 1 \quad (3.5)$$

Usando também a discretização de primeira ordem da derivada tem-se:

$$\dot{X} = \frac{X_{n+1} - X_n}{\Delta S_n} \quad (3.6)$$

juntando as equações 3.5 e 3.6 obtém-se:

$$X_{n+1} = X_n + \Delta S_n (\theta f(X_{n+1}) + (1 - \theta)f(X_n)) \quad (3.7)$$

Para cada valor escolhido de θ se obtém um método diferente de integração:

- Método explícito de *Euler*: $\theta = 0$;
- Método *Cranck-Nicolson*: $\theta = 1/2$;
- Método de *Galerkin*: $\theta = 2/3$;

- Método implícito de *Euler*: $\theta = 1$.

Nas simulações deste trabalho utilizou-se o método implícito de Euler.

Durante a simulação o primeiro ΔS é determinado pelo usuário e durante a simulação o próprio *POLYFLOW*, com um critério interno de escolha, determina os próximos ΔS . Este critério tende a aumentar o ΔS quando a iteração de evolução anterior convergiu e nas que não convergir o programa determina ΔS menores. Tanto ΔS_{\min} para interromper a simulação quanto ΔS_{\max} são determinados pelo usuário.

A estratégia de evolução utilizada foi começar com S_0 nulo, assim a primeira simulação seria um caso sem elasticidade o que garante uma convergência inicial. E para se alcançar o valor de trelax desejado usou-se S_{final} unitário.

3.2. Testes e Escolha da Malha

3.2.1. Modelagem da Malha

A geometria analisada neste trabalho é mostrada na figura 3.2. Como foi citado anteriormente, o domínio numérico usado foi a metade da geometria real, devido a simetria da mesma, como mostrado na figura 3.2.

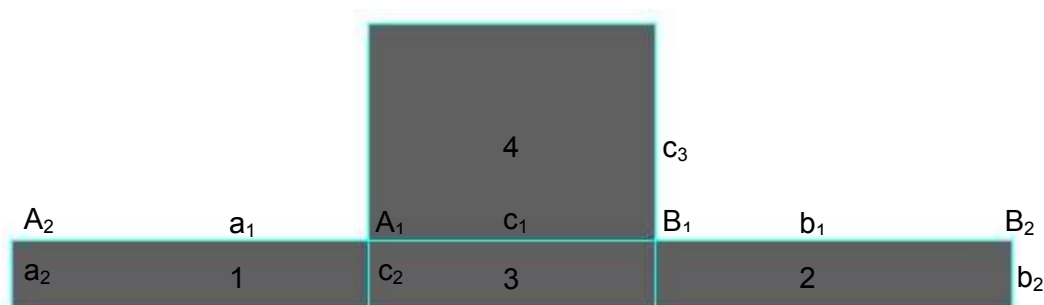


Figura 3.2: Geometria utilizada na simulação e parâmetros para a geração da malha

Para o desenvolvimento da malha, a geometria foi dividida em quatro seções. As seções 1 e 2 são respectivamente as seções de entrada e saída do fluido. A seção 3 corresponde à região central do duto e a seção 4 a concavidade onde o fluido se expande.

A malha na seção 1 foi desenvolvida a partir da divisão das arestas a_1 e a_2 . A aresta a_1 foi dividida partindo-se do vértice A_1 definindo-se o tamanho do menor elemento (mais próximo de A_1), a taxa de crescimento entre os elementos

subseqüentes e o tamanho máximo do elemento. A aresta a_2 foi dividida partindo-se do vértice A_2 com a condição de razão entre os elementos e fixado o número de elementos na aresta. As demais arestas na seção 1 foram divididas pelo mesmo procedimento das respectivas arestas opostas.

A malha na seção 2 foi desenvolvida a partir da divisão das arestas b_1 e b_2 . A aresta b_1 foi dividida partindo-se do vértice B_1 definindo-se o tamanho do menor elemento (mais próximo de B_1), a taxa de crescimento entre os elementos subseqüentes e o tamanho máximo do elemento. A aresta b_2 foi dividida partindo-se do vértice B_2 com a condição de razão entre os elementos e fixado o número de elementos na aresta. As demais arestas na seção 2 foram divididas pelo mesmo procedimento das respectivas arestas opostas.

A malha da seção 3 foi desenvolvida a partir da divisão das arestas c_1 e c_2 . A divisão da aresta c_1 foi realizada determinando o número de elementos existentes e com a condição de razão entre os elementos, com os menores elementos começando igualmente nas arestas A_1 e B_1 . As demais arestas na seção 3 foram divididas pelo mesmo procedimento das respectivas arestas opostas.

A malha da seção 4 foi desenvolvida a partir da divisão das arestas c_1 e c_3 . A aresta c_3 foi dividida partindo-se do vértice A_1 e assim como a aresta a_1 definiu-se o tamanho do menor elemento (mais próximo de A_1), a taxa de crescimento entre os elementos subseqüentes e o tamanho máximo do elemento.

3.2.2. Malhas Analisadas

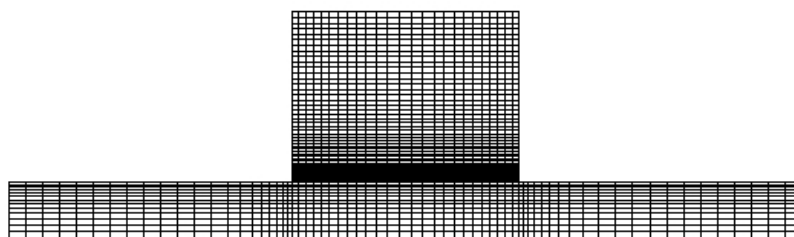
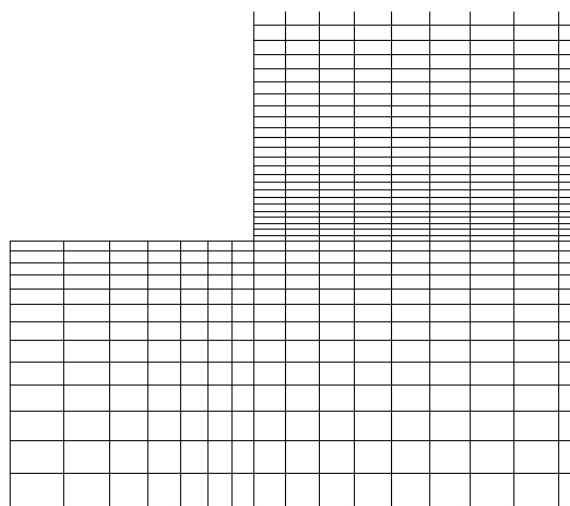
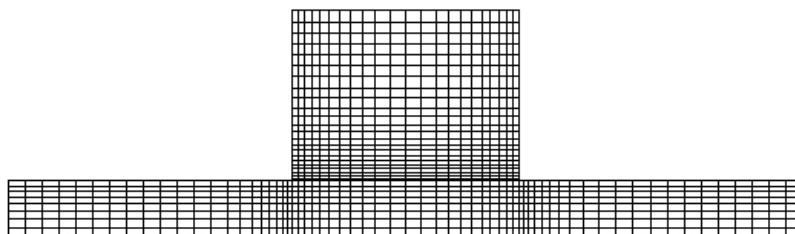
Neste trabalho foram geradas cinco malhas, mas apenas duas análises se mostraram necessárias nos testes. As malhas analisadas foram *malha003* e *malha005*, como mostrado na tabela 3.1. As malhas foram geradas para o caso $R_0/R = 4$ e $L_0/R_0 = 1$ (ver figura 2.1).

As equações foram adimensionalizadas antes de serem introduzidas no POLYFLOW, assim os tamanhos citados na tabela 3.1 são também adimensionais para fim da simulação.

Aresta	Parâmetro	<i>malha003</i>	<i>malha005</i>
a_1	Tamanho inicial	0,08	0,01
	Taxa de crescimento	1,2	1,3
	Tamanho limite	0,3	0,6
b_1	Tamanho inicial	0,08	0,01
	Taxa de crescimento	1,2	1,3
	Tamanho limite	0,3	0,6

c_3	Tamanho inicial	0,02	0,09
	Taxa de crescimento	1,05	1,1
	Tamanho limite	0,1	0,3
a_2	Número de elementos	13	8
	Razão de sucessão	0,9	0,9
c_1	Número de elementos	25	22
	Razão de sucessão	1,05	1,1

Tabela 3.1: Parâmetros de divisão das malhas

Figura 3.3: *malha003* com 2071 elementosFigura 3.4: Detalhe do refinamento da *malha003* na entrada da cavidadeFigura 3.5: *malha005* com 1018 elementos

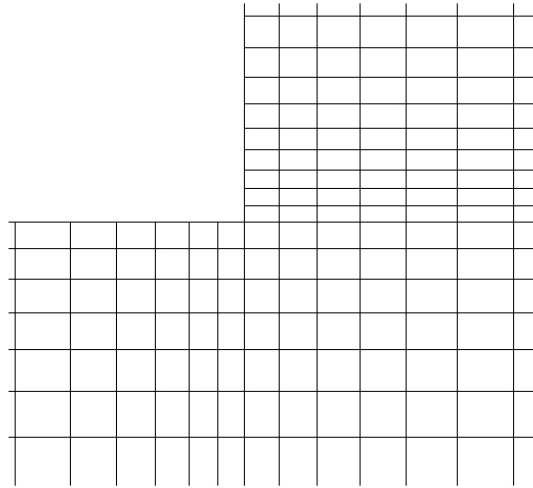


Figura 3.6: Detalhe do refinamento da *malha005* na entrada da cavidade

3.2.3. Testes de Malha

Para se garantir que o resultado obtido nas simulações seja independente da malha usada realizaram-se alguns testes com tolerâncias de 10^{-3} , 10^{-5} , 10^{-8} ., mantendo-se sempre o mesmo fluido simulado.

Para se comparar as malhas e as tolerâncias adotadas utilizou-se três variáveis e verificou-se a diferença percentual entre os resultados obtidos. As variáveis analisadas são:

- Perda de Carga: $\Delta p^* = \frac{\Delta p}{4\tau_R}$ (3.8)

Onde Δp é a perda de carga devido a presença da concavidade (figura 3.7) e τ_R é a tensão na parede do duto de raio R .

- Eficiência de deslocamento: $\varphi = \frac{V_{c,y}}{V_c}$ (3.9)

Onde $V_{c,y}$ é o volume yielded dentro da cavidade e V_c é o volume total da cavidade. Nos resultados apresentados, a taxa de deformação $\dot{\gamma}_0$ foi considerada como a transição entre a região *yielded* e *unyielded*. Assim, o volume *unyielded* é o que está com o escoamento $\dot{\gamma} < \dot{\gamma}_0$, e o volume yielded $\dot{\gamma} > \dot{\gamma}_0$.

- Assimetria: $As = \frac{V_{c,y,x^-}}{V_{c,y}}$ (3.10)

Onde V_{c,y,x^-} é o volume da região yielded a esquerda ao eixo r (figura 2.1) e V_c é o volume total da cavidade. Esta variável identifica a simetria do escoamento em relação ao eixo r .

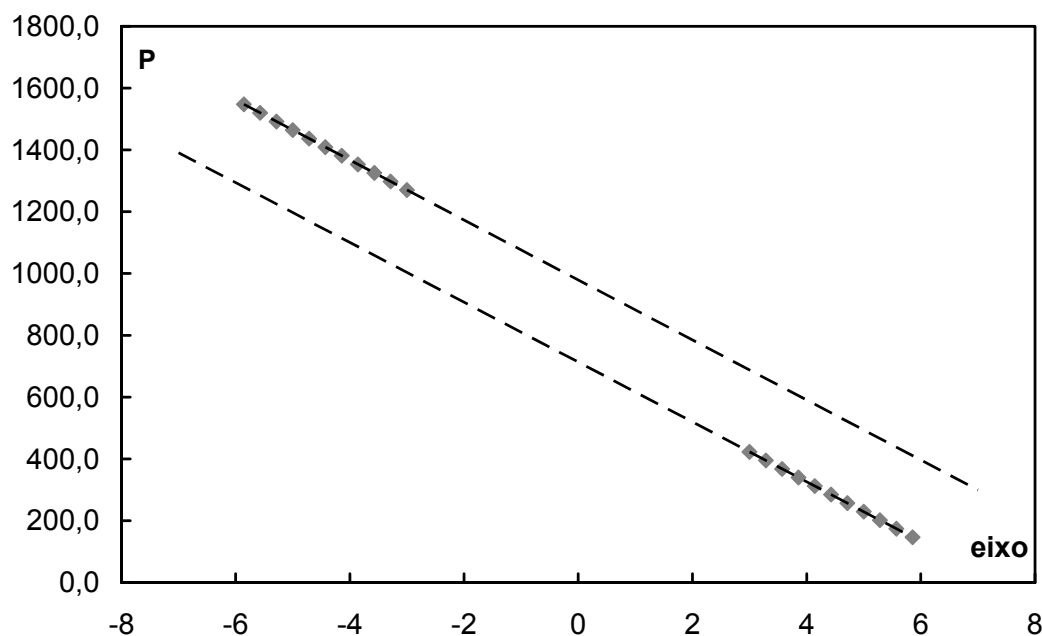


Figura 3.7: Exemplo da pressão em função do eixo x da geometria do escoamento

Os resultados dos testes e a comparação percentual são mostrados nas tabelas 3.2 e 3.3 respectivamente. A diferença percentual foi calculada tomando-se a diferença entre dois resultados com tolerâncias diferentes e dividindo-se esta diferença pelo resultado obtido da simulação com menor tolerância.

Resultados dos Testes de Malha			
φ	10^{-3}	10^{-5}	10^{-8}
<i>malha005</i>	0,511	0,508	0,509
<i>malha003</i>	0,497	0,496	0,497
Δp^*	10^{-3}	10^{-5}	10^{-8}
<i>malha005</i>	1,573	1,572	1,572
<i>malha003</i>	1,568	1,568	1,568
A_1	10^{-3}	10^{-5}	10^{-8}
<i>malha005</i>	0,514	0,513	0,510
<i>malha003</i>	0,515	0,515	0,514

Tabela 3.2: Resultados dos Testes de Malha

Diferença Percentual entre os Resultados do Teste de Malha		
φ	10^{-3} e 10^{-5}	10^{-5} e 10^{-8}
<i>malha005</i>	0,59%	-0,20%
<i>malha003</i>	0,20%	0,20%
Δp^*	10^{-3} e 10^{-5}	10^{-5} e 10^{-8}
<i>malha005</i>	0,19%	0,59%
<i>malha003</i>	0%	0,19%
A_1	10^{-3} e 10^{-5}	10^{-5} e 10^{-8}
<i>malha005</i>	0,06%	0%
<i>malha003</i>	0%	0%

Tabela 3.3: Diferença Percentual entre os Resultados do Teste de Malha

Com base nesses resultados, ambas as malhas analisadas, com qualquer tolerância assumida, a simulação tem um erro inferior a 1%, o que garante um resultado preciso para as variáveis analisadas. Porém, em uma análise do formato da interface entre os volumes *yielded* e *unyielded*, nota-se uma melhor convergência quando a simulação é implementada com tolerância 10^{-5} . Neste caso, considerando as aproximações da parte gráfica do *TECPLOT*, resulta que, considerando a *malha005*, com tolerância de 10^{-5} na simulação do *POLYFLOW*, a precisão de φ é da ordem de 10^{-4} , como mostrado nas figuras 3.8 e 3.9. Na figura 3.8, com precisão de 10^{-5} no *TECPLOT*, não se distingue a região de φ entre 0,99999 e 1, como mostrado no detalhe (figura 3.8b). Quando se analisa a interface com precisão de 10^{-4} no *TECPLOT*, há uma faixa distinguível entre a região de φ entre 0,9999 e 1, como mostrado no detalhe (figura 3.9b). Assim, considerando os erros de arredondamento do *TECPLOT* e do próprio *POLYFLOW*, verifica-se que a imprecisão na determinação da região de interface é menor do que o valor admitido anteriormente de 1%.

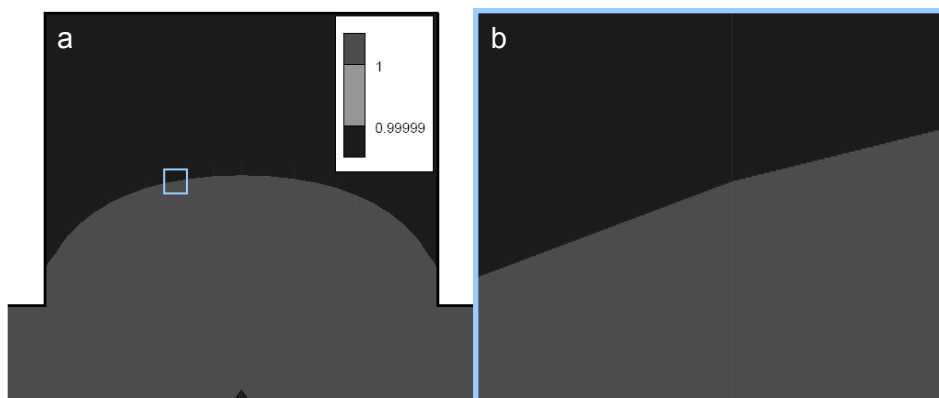


Figura 3.8: a) Análise da interface com precisão do *TECPLOT* de 10^{-5} . b) detalhe da interface.

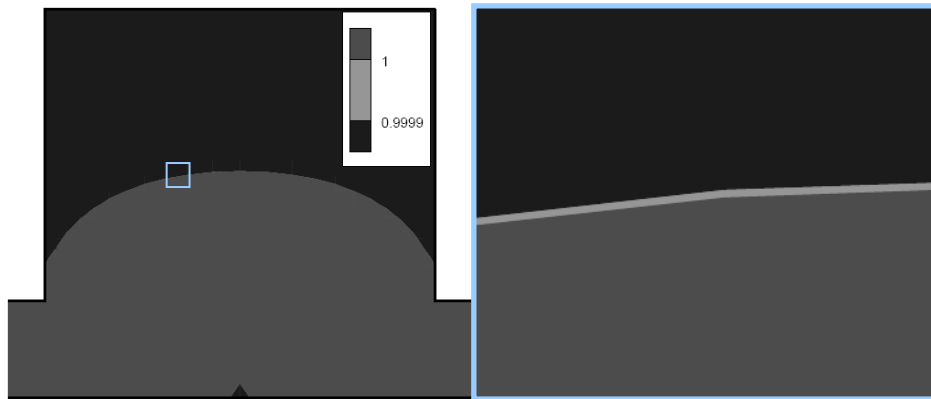


Figura 3.9: a) Análise da interface com precisão do *TECPLOT* de 10^{-4} . b) detalhe da interface.

Desta forma, foram adotadas como padrão para a simulação a *malha005* e a tolerância de 10^{-5} .