

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA QUÍMICA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA QUÍMICA

**SIMULAÇÃO DE ESCOAMENTOS VISCOELÁSTICOS:
DESENVOLVIMENTO DE UMA METODOLOGIA DE ANÁLISE
UTILIZANDO O SOFTWARE *OpenFOAM* E EQUAÇÕES
CONSTITUTIVAS DIFERENCIAIS**

DISSERTAÇÃO DE MESTRADO

JOVANI LUIZ FAVERO

**PORTO ALEGRE, RS
2009**

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA QUÍMICA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA QUÍMICA

**SIMULAÇÃO DE ESCOAMENTOS VISCOELÁSTICOS:
DESENVOLVIMENTO DE UMA METODOLOGIA DE ANÁLISE
UTILIZANDO O SOFTWARE *OpenFOAM* E EQUAÇÕES
CONSTITUTIVAS DIFERENCIAIS**

JOVANI LUIZ FAVERO

Dissertação de Mestrado apresentada como requisito parcial para obtenção do título de Mestre em Engenharia.

Área de Concentração: Pesquisa e Desenvolvimento de Processos

Orientadores:

Prof. Argimiro Resende Secchi, D.Sc.

Prof. Nilo Sérgio Medeiros Cardozo, D.Sc.

Co-Orientador:

Prof. Hrvoje Jasak, D.Sc.

**PORTO ALEGRE, RS
2009**

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA QUÍMICA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA QUÍMICA

TIPO DE DOCUMENTO: **ERRATA**

TRABALHO: **SIMULAÇÃO DE ESCOAMENTOS VISCOELÁSTICOS:
DESENVOLVIMENTO DE UMA METODOLOGIA DE ANÁLISE UTILIZANDO
O SOFTWARE *OpenFOAM* E EQUAÇÕES CONSTITUTIVAS
DIFERENCIAIS**

JOVANI LUIZ FAVERO

**PORTO ALEGRE, RS
2010**

Enumeração de Erros

1. No final da página 27 substituir a frase:

"Se $\alpha_K \neq 0$ e $0 \leq \alpha_K \leq 2$ este modelo conduz a uma primeira diferença de tensões normais (N_2) diferente de zero."

por

"Se $\alpha_K \neq 0$ e $0 \leq \alpha_K \leq 2$ este modelo conduz a uma **segunda** diferença de tensões normais (N_2) diferente de zero."

2. Para os modelos Pom-Pom (PP), DXPP e DCPD a equação da evolução do estiramento dorsal do tubo Λ_{PK} possui um sinal errado no segundo termo do lado direito da equação.

Desse modo, as Eqs. 2.49, 2.52 e 2.60 da subseção 2.5.3.5 que se apresentam na forma:

$$\frac{D(\Lambda_{PK})}{Dt} = \Lambda_{PK} [D : S_{PK}] + \frac{1}{\lambda_{SK}} [\Lambda_{PK} - 1]$$

devem ser substituídas por:

$$\frac{D(\Lambda_{PK})}{Dt} = \Lambda_{PK} [D : S_{PK}] - \frac{1}{\lambda_{SK}} [\Lambda_{PK} - 1]$$

3. No modelo SXPP da página 34 a função extra (Eq. 2.56) que foi colocada como sendo:

$$\frac{1}{\lambda_{OBK}} f(\tau)^{-1} = \frac{2}{\lambda_{SK}} \left(1 - \frac{1}{\Lambda}\right) + \frac{2}{\lambda_{OBK} \Lambda^2} \left(1 - \frac{\alpha_K \lambda_{OBK}^2 I_{\tau,\tau}}{3\eta_{PK}^2}\right)$$

deve ser substituída por:

$$\frac{1}{\lambda_{OB_K}} f(\tau)^{-1} = \frac{2}{\lambda_{S_K}} \left(1 - \frac{1}{\Lambda}\right) + \frac{\mathbf{1}}{\lambda_{OB_K} \Lambda^2} \left(1 - \frac{\alpha_K \lambda_{OB_K}^2 I_{\tau \cdot \tau}}{3\eta_{P_K}^2}\right)$$

Neste mesmo modelo a equação para o tensor tempo de relaxação (Eq. 2.55) colocada como:

$$\lambda(\tau)^{-1} = \frac{1}{\lambda_{OB_K}} \left[\frac{\alpha_K \lambda_{OB_K}}{\eta_{P_K}} \tau_{P_K} + f(\tau)^{-1} \delta + \frac{\lambda_{OB_K}}{\eta_{P_K}} (f(\tau)^{-1} - 1) \tau_{P_K}^{-1} \right]$$

deve ser substituída por:

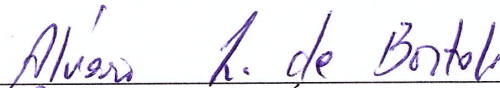
$$\lambda(\tau)^{-1} = \frac{1}{\lambda_{OB_K}} \left[\frac{\alpha_K \lambda_{OB_K}}{\eta_{P_K}} \tau_{P_K} + f(\tau)^{-1} \delta + \frac{\eta_{P_K}}{\lambda_{OB_K}} (f(\tau)^{-1} - 1) \tau_{P_K}^{-1} \right]$$

Obs.: Os erros reportados nas equações acima foram erros somente de digitação e não estão presentes no código do *solver viscoelasticFluidFoam* . Desse modo, nenhuma alteração no *solver* é necessária e os resultados mostrados estão corretos.

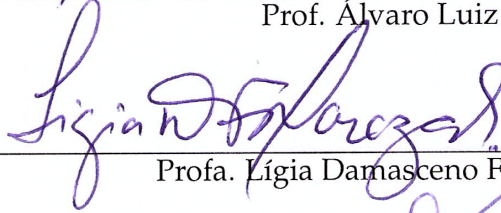
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA QUÍMICA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA QUÍMICA

A Comissão Examinadora, abaixo assinada, aprova a Dissertação *Simulação de Escoamentos Viscoelásticos: Desenvolvimento de uma Metodologia de Análise utilizando o Software OpenFOAM e Equações Constitutivas Diferenciais*, elaborada por Jovani Luiz Favero como requisito parcial para obtenção do Grau de Mestre em Engenharia.

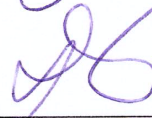
Comissão Examinadora:



Prof. Álvaro Luiz de Bortoli, D.Sc.



Profa. Lígia Damasceno Ferreira Marczak, Dra.Sc.



Profa. Rosário Elida Suman Bretas, Dra.Sc.

Uma vez tendo experimentado voar, caminharás para sempre sobre a Terra de olhos postos no Céu, pois é para lá que tencionas voltar.

Leonardo da Vinci

Agradecimentos

À Deus que me deu forças e coragem durante todo esse tempo.

Aos professores Argimiro Resende Secchi, Nilo Sérgio Medeiros Cardozo e Hrvoje Jasak pela valiosa e indispensável orientação e incentivo que possibilitou o desenvolvimento deste trabalho.

À minha família que sempre esteve presente em minha vida mesmo com a distância geográfica.

À Cristina pelas longas e animadas conversas que ajudaram a descontraír.

Ao Professor André Rodrigues Muniz pela ajuda e troca de idéias.

Aos colegas de mestrado e de sala que deram sinceras mostras de companheirismo e contribuíram para o aprendizado durante todo esse tempo.

À CAPES pelo apoio financeiro durante o desenvolvimento deste trabalho.

Resumo

A necessidade cada vez maior do uso de produtos poliméricos sintéticos, como para produção de embalagens, partes de eletrodomésticos, eletroeletrônicos, automóveis, etc., tem levado a indústria de polímeros a buscar cada vez mais a diminuição do desperdício e aumento da qualidade dos produtos. Para isso tem-se buscado entender melhor como as propriedades reológicas dos polímeros afetam seu processamento e a qualidade final dos produtos. Com o intuito de se obter resultados mais rápidos e com menor custo recorre-se cada vez mais a estudos de modelagem e simulação de processos de transformação de polímeros. Neste trabalho é apresentada uma nova ferramenta de CFD para simulações de escoamentos envolvendo fluidos viscoelásticos, o *viscoelasticFluidFoam* solver. A implementação do módulo foi feita no pacote de CFD *OpenFOAM* devido principalmente às vantagens oferecidas por esse *software*, como por exemplo, possibilidade de uso de geometrias complexas, malhas não-estruturadas, técnicas *multigrid* e paralelização do processamento de dados, além de ser um *software* gratuito e de código aberto. Foi feita a implementação do modelo de Maxwell, UCM, Oldroyd-B, Giesekus, FENE-P, FENE-CR, PTT na forma linear e exponencial, e DCP, todos na forma multimodo. Dentre as várias metodologias disponíveis para resolver o problema da obtenção de soluções estáveis a altos valores de *Weissenberg* foi escolhida a DEVSS devido a sua estabilidade e aplicação a modelos complexos. Para se fazer a validação do *solver* desenvolvido foi feita a comparação com resultados numéricos e experimentais obtidos da literatura. É mostrada uma comparação entre vários modelos para obtenção da velocidade e diferença de tensões normais para um escoamento em uma contração plana abrupta 4:1. Os resultados obtidos foram satisfatórios sendo possível dar credibilidade ao *solver* implementado e garantir a disponibilidade de uma boa ferramenta para estudo de fluidos viscoelásticos para ser usada tanto no meio acadêmico como no setor industrial.

Palavras-chave: Polímeros, fluidos viscoelásticos, simulação numérica, CFD, *OpenFOAM*.

Abstract

Synthetic polymer products are of great importance in several industrial sectors, such as for production of packaging, parts of appliances, electronics, and cars. Due to the increasing demand for this kind of material, reduction of waste and increase of quality has become a key issue in polymer industry. In this sense modeling and simulation of processing operations appears as a fundamental tool, leading to better understanding of how the rheological properties of polymers affect their processability and final product quality, and reducing time and costs related to the development of processes and products. This work presents a new Computational Fluid Dynamics (CFD) tool for the simulation of viscoelastic fluid flows, called *viscoelasticFluidFoam* solver, which consists of a viscoelastic fluid module to be used *OpenFOAM* CFD package. The advantages of using *OpenFOAM* as development platform include its characteristics with relation to flexibility to deal with complex geometries, unstructured and non orthogonal meshes, moving meshes, large variety of interpolation schemes and solvers for the linear discretized system, and the possibility of data processing parallelization. Linear Maxwell, Oldroyd-B, Giesekus, Phan-Thien-Tanner (PTT), the Finitely Extensible Nonlinear Elastic (FENE-P and FENE-CR), and DCCP constitutive equations have been implemented, in single and the multimode form. Among the various available methodologies to solve the problem of obtaining stable solutions to high *Weissenberg* values, the DEVSS was chosen due to its stability and application to complex models. The *viscoelasticFluidFoam* solver was tested by comparing its predictions with experimental and numerical data from literature for the analysis of a planar 4:1 contraction flow. These tests have shown the great potential of this solver for application both in academia and in industry.

Key-words: Polymers, viscoelastic fluids, numeric simulation, CFD, *OpenFOAM*.

Sumário

Lista de Figuras	xix
Lista de Tabelas	xxi
Lista de Símbolos	xxvii
Lista de Códigos	xxix
1 Introdução	1
1.1 Motivação do Trabalho	1
1.2 Objetivo	3
1.3 Estrutura da Dissertação	3
2 Modelos Constitutivos e Propriedades de Fluidos Poliméricos	5
2.1 Comportamento reológico de materiais poliméricos	5
2.2 Números Adimensionais importantes para Escoamentos de Fluidos Viscoelásticos	11
2.3 Funções Materiais para Fluidos Poliméricos	12
2.4 Modelagem Matemática	16
2.4.1 Equações Governantes para Fluidos Newtonianos	17
2.4.2 Formulação do Modelo Matemático para Fluidos Poliméricos	18
2.5 Equações Constitutivas para Fluidos Poliméricos	19

2.5.1	Fluido Newtoniano Generalizado (FNG)	19
2.5.2	Fluido Viscoelástico Linear (FVL)	21
2.5.3	Fluido Viscoelástico Não-Linear	23
2.5.3.1	Modelos de Oldroyd-B, UCM e White-Metzner (WM)	24
2.5.3.2	Modelos de Giesekus e Leonov	27
2.5.3.3	Modelos do tipo FENE	28
2.5.3.4	Modelos de Phan-Thien-Tanner (PTT) e Feta-PTT	29
2.5.3.5	Modelos de Pom-Pom (PP), SXPP, DXPP e DCP	31
3	Resolução Numérica de Escoamentos de Fluidos Viscoelásticos	37
3.1	Métodos Numéricos para Resolver um Problema de CFD	37
3.2	Método de Volumes Finitos	39
3.2.1	Tipo de arranjo utilizado para as variáveis	40
3.2.2	Esquemas de Interpolação	40
3.2.3	Solução das Equações Discretizadas	41
3.3	Metodologia para Resolver Escoamentos com Elevados valores de We	43
3.3.1	Formulação Viscosa	43
3.3.2	Formulação EVSS	44
3.3.3	Formulação DEVSS	46
3.3.4	Formulação AVSS, DAVSS e derivações destas Metodologias	46
4	Apresentação do Pacote de CFD <i>OpenFOAM</i>	49
4.1	O <i>OpenFOAM</i>	49
4.2	O <i>OpenFOAM</i> a nível de Usuário	51
4.2.1	Estrutura Necessária para Efetuar uma Simulação no <i>OpenFOAM</i>	51
4.2.2	Pré-Processamento	53

4.2.3	Etapa de Resolução Numérica	55
4.2.4	Pós-Processamento	55
4.3	O <i>OpenFOAM</i> a nível de Usuário Desenvolvedor	56
4.3.1	Orientação a Objetos e C++	56
4.3.2	Linguagem do <i>OpenFOAM</i>	58
4.3.3	Definição e nomenclatura dos operadores diferenciais no <i>Open-FOAM</i>	60
4.3.3.1	Avaliação do Operador Gradiente	61
4.3.3.2	Avaliação do Operador Divergente	62
4.3.3.3	Avaliação do Operador Laplaciano	63
4.3.3.4	Esquemas de Interpolação para Avaliação Temporal	63
4.3.4	Outros Operadores auxiliares no <i>OpenFOAM</i>	64
4.3.5	Definição e nomenclatura dos esquemas de interpolação no <i>Open-FOAM</i>	65
4.3.6	<i>Solvers</i> para Solução do Sistema Linear de Equações	68
4.3.7	Opções para Condições de Contorno	68
5	Desenvolvimento do Solver para fluidos viscoelásticos: <i>viscoelasticFluidFoam</i>	71
5.1	Escolha de uma Equação Constitutiva	71
5.2	Escolha de uma Metodologia	72
5.3	Algoritmo para Resolução de Escoamento de Fluido Viscoelástico	73
5.4	Detalhamento da implementação do <i>solver viscoelasticFluidFoam</i>	74
5.4.1	Código principal do <i>solver viscoelasticFluidFoam</i>	74
5.4.2	A biblioteca <code>createFields.H</code>	82
5.4.3	Implementação do modelo Phan-Thien-Tanner linear	83
5.5	Estudos de validação do código implementado	88

5.5.1	Escolha de uma geometria representativa	88
5.5.2	Parâmetros utilizados nos testes	89
6	Resultados: Validação do <i>Solver</i> Desenvolvido	91
6.1	Validação da implementação da estrutura básica do <i>solver</i> utilizando o modelo de Giesekus com um único modo	91
6.1.1	Estudo de Convergência de Malha	92
6.1.2	Teste de Esquemas de Interpolação	95
6.1.3	Comparação das predições com dados Numéricos e Experimentais	98
6.1.3.1	Dinâmica na seção anterior a contração	101
6.1.3.2	Dinâmica na seção posterior a contração	103
6.1.3.3	Efeito do valor de <i>Deborah</i>	105
6.2	Modelo de Giesekus na forma multimodo	106
6.3	Avaliação da implementação do modelo DCPD	108
6.4	Avaliação da Implementação de outros Modelos Constitutivos	110
6.4.1	Teste da implementação dos modelos LPTTS e FENE-P	111
6.4.2	Teste da implementação dos modelos EPTTS e FENE-CR	113
6.4.3	Teste da implementação dos modelos Maxwell linear e Oldroyd-B	115
7	Conclusões	117
	Referências Bibliográficas	127

Lista de Figuras

2.1	Representação do efeito de inchamento de extrudado (<i>die swell</i>) para um fluido newtoniano (esquerda) e um fluido polimérico (direita).	9
2.2	Experimento " <i>rod-climbing</i> ". A esquerda pode-se ver o comportamento de um fluido newtoniano e a direita o de um fluido polimérico.	10
2.3	Curvas típicas da viscosidade não-newtoniana para o polietileno de baixa densidade em função da taxa de deformação, sob diferentes temperaturas.	15
2.4	Viscosidade elongacional e viscosidade não-newtoniana para uma mistura de poliestireno.	16
2.5	Modelo físico mola-amortecedor, representando o modelo viscoelástico linear de Maxwell.	21
2.6	Representação de uma macromolécula de polímero como um " <i>dumbbell</i> ".	25
2.7	Estrutura esquemática da molécula do modelo Pom-Pom.	32
3.1	Volume de controle.	39
4.1	Estrutura de diretórios e arquivos necessários para uma simulação com o <i>OpenFOAM</i>	52
4.2	Parâmetros em uma discretização por volumes finitos.	60
5.1	Esboço da geometria utilizada.	90
6.1	Malha computacional (Malha 2).	92
6.2	Perfis de velocidade U em um corte transversal na seção anterior a contração usando as malhas 1, 2 e 3.	93

6.3	Perfis de tensão de cisalhamento τ_{xy} em um corte transversal na seção anterior a contração usando as malhas 1, 2 e 3.	93
6.4	Perfis da primeira diferença de tensões normais N_1 em um corte transversal na seção anterior a contração usando as malhas 1, 2 e 3.	94
6.5	Perfis de tensão de cisalhamento τ_{xy} na seção anterior a contração para alguns esquemas de interpolação (esquerda). Ampliação da região em destaque (direita).	96
6.6	Perfis da primeira diferença de tensões normais N_1 na seção anterior a contração para alguns esquemas de interpolação (esquerda). Ampliação da região 2 em destaque (direita).	96
6.7	Representação das oscilações na região compreendida entre 0,4 à 0,7 na Figura 6.6 à esquerda.	97
6.8	Campo de pressão.	99
6.9	Campo velocidade.	99
6.10	Campo de tensão τ_{xx}	99
6.11	Campo de tensão τ_{xy}	100
6.12	Campo de tensão τ_{yy}	100
6.13	Linhas de corrente.	100
6.14	Perfis de velocidade Ux em cortes transversais (esquerda) e paralelos (direita) ao escoamento na seção anterior a contração.	101
6.15	Perfis para a componente da tensão τ_{xy} em cortes transversais (esquerda) e paralelos (direita) ao escoamento na seção anterior a contração.	101
6.16	Perfis para a primeira diferença de tensões normais (N_1) em cortes transversais (esquerda) e paralelos (direita) ao escoamento na seção anterior a contração.	102
6.17	Perfis de velocidade Ux em um corte transversal ao escoamento na seção posterior a contração.	103
6.18	Perfis para a componente da tensão τ_{xy} em cortes transversais (esquerda) e paralelos (direita) ao escoamento na seção posterior a contração.	104
6.19	Perfis para a primeira diferença de tensões normais (N_1) em cortes transversais (esquerda) e paralelos (direita) ao escoamento na seção posterior a contração.	104

6.20	Perfis para a velocidade (esquerda) e primeira diferença de tensões normais (N_1) (direita) na linha central do escoamento e quando submetidos a diferentes valores de <i>Deborah</i>	105
6.21	Perfis para a primeira diferença de tensões normais (N_1) em um corte lateral (esquerda) e na linha de simetria (direita) ao escoamento na seção posterior a contração usando o modelo de Giesekus 4-modos.	107
6.22	Perfis para a velocidade usando o modelo DCP. Literatura (esquerda) e este trabalho (direita).	109
6.23	Perfis para a PSD usando o modelo DCP. Literatura (esquerda) e este trabalho (direita).	109
6.24	Perfis de velocidade Ux em um corte transversal ao escoamento comparando os modelos Giesekus, FENE-P e LPTTS.	112
6.25	Perfis para a componente da tensão τ_{xy} em um corte transversal (esquerda) e paralelo (direita) ao escoamento comparando os modelos Giesekus, FENE-P e LPTTS.	112
6.26	Perfis para a primeira diferença de tensões normais N_1 em um corte transversal (esquerda) e paralelo (direita) ao escoamento comparando os modelos Giesekus, FENE-P e LPTTS.	112
6.27	Perfis de velocidade Ux em um corte transversal ao escoamento comparando os modelos Giesekus, FENE-CR e EPTTS.	114
6.28	Perfis para a componente da tensão τ_{xy} em um corte transversal (esquerda) e paralelo (direita) ao escoamento comparando os modelos Giesekus, FENE-CR e EPTTS.	114
6.29	Perfis para a primeira diferença de tensões normais N_1 em um corte transversal (esquerda) e paralelo (direita) ao escoamento comparando os modelos Giesekus, FENE-CR e EPTTS.	114
6.30	Perfis de velocidade Ux em um corte transversal ao escoamento comparando os modelos Giesekus, Maxwell linear e Oldroyd-B.	116
6.31	Perfis para τ_{xy} em um corte transversal (esquerda) e paralelo (direita) ao escoamento comparando os modelos Giesekus, Maxwell linear e Oldroyd-B.	116
6.32	Perfis para N_1 em um corte transversal (esquerda) e paralelo (direita) ao escoamento comparando os modelos Giesekus, Maxwell linear e Oldroyd-B.	116

Lista de Tabelas

4.1	Principais palavras chaves usadas no arquivo <i>fvSchemes</i>	60
4.2	Esquemas de discretização possíveis em <i>gradSchemes</i>	62
4.3	Esquemas de discretização disponíveis em <i>ddtSchemes</i>	64
4.4	Alguns dos esquemas de interpolação presentes no <i>OpenFOAM</i>	66
4.5	Comportamento dos esquemas de interpolação usados em <i>divSchemes</i>	67
4.6	<i>Solvers</i> para o sistema linear.	68
4.7	Opções de pré-condicionadores.	68
4.8	Especificações primitivas para os contornos.	70
5.1	Representação de operadores matemáticos usando a linguagem do <i>OpenFOAM</i>	87
5.2	Condições do escoamento.	90
6.1	Características da malha.	92
6.2	Erro relativo, em percentual, das malhas 1 e 2 em relação a malha 3 para as curvas referenciadas com a letra "a" nas Figuras 6.2 - 6.4.	94
6.3	Erro relativo máximo e erro relativo médio, em percentual, do esquema <i>upwind</i> em relação ao <i>Gamma 1</i> para a curva "a".	97
6.4	Parâmetros para o modelo de Giesekus 4-modos.	106
6.5	Parâmetros para o modelo DCP 4-modos.	108
6.6	Parâmetros dos modelos.	111

Lista de Símbolos

Co	Número de <i>Courant</i>	—
D	Tensor taxa de deformação	s^{-1}
De	Número de <i>Deborah</i>	—
II_D	Segundo invariante do tensor taxa de deformação	s^{-1}
II_τ	Segundo invariante da tensão	$kg.m^{-2}.s^{-4}$
I_τ	Primeiro invariante da tensão	$kg.m^{-1}.s^{-2}$
L^2	Extensibilidade adimensional das moléculas	—
L_c	Comprimento característico	m
N_1	Primeira diferença de tensões normais	$kg.m^{-1}.s^{-2}$
N_2	Segunda diferença de tensões normais	$kg.m^{-1}.s^{-2}$
Re	Número de <i>Reynolds</i>	—
S	Tensor orientação	—
S_f	vetor que contem a área da face.	m^2
U	Vetor velocidade	$m.s^{-1}$
U_c	Velocidade característica	$m.s^{-1}$
V	Volume	m^3
We	Número de <i>Weissenberg</i>	—
\underline{U}	Velocidade média	$m.s^{-1}$
d	Vetor de P até N	m
g_b	Condição de contorno para a condição <i>fixed Gradient</i>	—
p	Pressão	$kg.m^{-1}.s^{-2}$

q	Quantidade de ramos existentes desde o começo até o fim da espinha dorsal do tubo	—
t	Tempo	s
t_c	Tempo característico do escoamento	s

Letras Gregas

α	Fator de mobilidade	—
δ	Tensor identidade	—
$\dot{\gamma}$	Taxa de deformação	s^{-1}
$\dot{\epsilon}$	Taxa de alongação	s^{-1}
$\eta(\dot{\gamma})$	Viscosidade não-newtoniana	$kg.m^{-1}.s^{-1}$
η	Viscosidade newtoniana	$kg.m^{-1}.s^{-1}$
η_0	Viscosidade a taxa de deformação nula	$kg.m^{-1}.s^{-1}$
γ_c	Taxa de deformação característica	s^{-1}
λ_{OB_K}	Tempo de relaxação para a orientação da espinha dorsal do tubo	s
λ_{OS_K}	Tempo de relaxação para o estiramento	s
Λ	Estiramento dorsal da molécula	—
λ	Tempo de relaxação	s
$\bar{\eta}$	Viscosidade elongacional	$kg.m^{-1}.s^{-1}$
$\bar{\eta}_1$	Primeira função viscosidade elongacional	$kg.m^{-1}.s^{-1}$
$\bar{\eta}_2$	Segunda função viscosidade elongacional	$kg.m^{-1}.s^{-1}$
ϕ_f	Variável arbitrária analisada na face da célula	—
Ψ_1	Primeiro coeficiente de tensões normais	$kg.m^{-1}$
Ψ_2	Segundo coeficiente de tensões normais	$kg.m^{-1}$
ρ	Massa específica	$kg.m^{-3}$
τ	Tensor das tensões	$kg.m^{-1}.s^{-2}$
τ_E	Parcela elástica da tensão polimérica	$kg.m^{-1}.s^{-2}$
τ_P	Tensor tensão para a contribuição polimérica	$kg.m^{-1}.s^{-2}$

τ_s	Tensor tensão para a contribuição do solvente	$kg.m^{-1}.s^{-2}$
τ_v	Parcela viscosa da tensão polimérica	$kg.m^{-1}.s^{-2}$
τ_{xx}	Tensões normais xx	$kg.m^{-1}.s^{-2}$
τ_{yx}	Tensão de cisalhamento	$kg.m^{-1}.s^{-2}$
τ_{yy}	Tensões normais yy	$kg.m^{-1}.s^{-2}$
τ_{zz}	Tensões normais zz	$kg.m^{-1}.s^{-2}$
ε	Parâmetro não linear do modelo PTT	—
ξ	Parâmetro não linear que relaciona as diferenças de tensões normais	—

Sobrescritos

q^n	Valor de uma variável q em um nível de tempo posterior
q^o	Valor de uma variável q em um nível de tempo anterior
q^{oo}	Valor de uma variável q em dois níveis de tempo anteriores

Subescritos

f	Face de comunicação entre as duas células
K	Índice para cada modo da formulação multimodo
P	Corresponde a contribuição polimérica
S	Corresponde a contribuição do solvente

Outros Símbolos

Δt	Intervalo de tempo
Δx	Variação de espaço
\square_{τ}	Derivada de Gordon-Schowalter
Δ_{τ}	Derivada convectiva inferior no tempo do tensor das tensões
∇_{τ}	Derivada convectiva superior no tempo do tensor das tensões

Siglas

<i>AVSS</i>	Adaptive Viscosity Stress Splitting Scheme
<i>CAD</i>	Computer-Aided Design
<i>CDS</i>	Central Differencing Scheme
<i>CFD</i>	Computational Fluid Dynamics
<i>CG</i>	Conjugate Gradient
<i>CUBISTA</i>	Convergent and Universally Bounded Interpolation Scheme for the Treatment of Advection
<i>DCPP</i>	Double Convected Pom-Pom
<i>DEVSS</i>	Discrete Elastic Viscous Split-Stress
<i>DIC</i>	Diagonal incomplete-Cholesky
<i>DILU</i>	Diagonal incomplete-LU
<i>DXPP</i>	Double equation eXtended Pom-Pom
<i>EEME</i>	Explicitly Elliptic Momentum Equation
<i>EPTT</i>	Exponential Phan-Thien-Tanner
<i>EVSS</i>	Elastic Viscous Split-Stress
<i>FEM</i>	Finite Element Method
<i>FENE</i>	Finitely Extensible Nonlinear Elastic
<i>FENE – CR</i>	Finitely Extensible Nonlinear Elastic-Chilcott and Rallison
<i>FENE – P</i>	Finitely Extensible Nonlinear Elastic-Peterlin
<i>Feta – PTT</i>	Fixed eta Phan-Thien-Tanner
<i>FIB</i>	Flow-Induced Birefringence
<i>FNG</i>	Fluido Newtoniano Generalizado
<i>FVL</i>	Fluido Viscoelástico Linear
<i>FVM</i>	Finite Volume Method
<i>GAMG</i>	Generalised geometric-algebraic multi-grid
<i>GLP</i>	Gnu Public License
<i>GMRES</i>	Generalized Minimal Residual

<i>HDS</i>	Hybrid Differencing Scheme
<i>HRS</i>	High Resolution Schemes
<i>HWNP</i>	High Weissenberg Number Problem
<i>LDV</i>	Laser-Doppler Velocimetry
<i>LPTT</i>	Linear Phan-Thien-Tanner
<i>OpenFOAM</i>	Open Source Field Operation and Manipulation
<i>PIB</i>	Poli-IsoButileno
<i>PISO</i>	Pressure Implicit Splitting of Operators
<i>POO</i>	Programação Orientada a Objeto
<i>PP</i>	Pom-Pom
<i>PSD</i>	Principal Stress Difference
<i>SIMPLE</i>	Semi-Implicit Method for Pressure Linked Equations
<i>SXPP</i>	Single equation eXtended Pom-Pom
<i>UCM</i>	Upper Convected Maxwell
<i>UDS</i>	Upwind Differencing Scheme
<i>WENO</i>	Weighted Essentially Non-Oscillatory

Lista de Códigos

5.1	<i>Solver viscoelasticFluidFoam</i> (arquivo <code>viscoelasticFluidFoam.C</code>).	75
5.2	Biblioteca <code>createFields.H</code> do <i>Solver viscoelasticFluidFoam</i>	82
5.3	Conteúdo do arquivo <code>LPTT.C</code> do <i>Solver viscoelasticFluidFoam</i>	84

Capítulo 1

Introdução

O uso de produtos de origem polimérica é uma necessidade da sociedade moderna e basta olharmos a nossa volta para vermos o quanto eles estão presentes em nossas vidas. Como exemplos, podem-se citar a maioria das embalagens para produtos alimentares, de limpeza, de beleza e de uso doméstico, peças para a indústria automobilística, de eletrodomésticos e eletroeletrônicos, produção de órgãos artificiais e artefatos para a indústria aeroespacial onde se requer materiais com alta qualidade, entre uma infinidade de outros exemplos que se poderia citar. Para se conseguir melhorar a qualidade dos produtos, diminuir o desperdício e conseguir as propriedades desejadas opta-se cada vez mais pelas simulações computacionais antes da execução de um projeto ou para melhorias dos já existentes ao invés de se usar os antigos protótipos de bancada que consomem mais tempo e envolvem maiores custos. Assim, não restam dúvidas da forte tendência de se usar cada vez mais a mecânica de fluidos computacional nas indústrias de processamento de polímeros.

1.1 Motivação do Trabalho

A existência de *softwares* para resolução de problemas envolvendo escoamento de fluidos poliméricos é ainda muito limitada sendo que a maioria deles não chegam às indústrias, ou seja, são utilizados quase que exclusivamente no meio acadêmico.

Dentre os poucos *softwares* comerciais que apresentam modelos constitutivos para fluidos poliméricos, podem-se citar:

- ***Polyflow***: é um módulo da ANSYS Inc. e possui alguns modelos para fluidos viscoelásticos implementados;
- ***Flow 2000***: bastante usado na indústria de polímeros para simular extrusão, contendo apenas modelos não-Newtonianos puramente viscosos e é comercializado pela Compuplast Inc.;
- ***Phoenix CFD***: permite algumas implementações de modelos e é comercializado pela CHAM Ltd.;
- ***Moldex3d***: bastante usado na indústria de polímeros para simular injeção usando modelos não-Newtonianos puramente viscosos e comercializado pela CoreTech System;
- ***Moldflow Plastics Insight***: usado para simular injeção usando modelos não-Newtonianos puramente viscosos e é comercializado pela Autodesk Inc.;

No entanto os *softwares* comerciais apresentam a desvantagem de possuírem custos elevados com obtenção e manutenção de licença e apresentam suas rotinas de cálculo escondidas, privando o usuário de saber quais as técnicas e artifícios numéricos que realmente são usados pelo *software*.

No ramo acadêmico geralmente são desenvolvidos *softwares* que visam resolver problemas bem específicos, sendo que a parte mais explorada é a questão numérica. Deste modo, esses códigos apresentam limitações para resolver os problemas complexos que são encontrados nas indústrias.

1.2 Objetivo

A proposta deste trabalho é a implementação e a validação de um módulo contendo modelos constitutivos para fluidos poliméricos em um pacote de CFD (*Computational Fluid Dynamics*) conhecido, confiável e de código aberto. O módulo poderá posteriormente ser usado tanto em estudos acadêmicos como dentro das indústrias para ajudar na implementação de novos projetos e na otimização de processos.

Visando atacar os dois problemas citados na [Seção 1.1](#), o módulo para fluidos poliméricos será de código aberto e permitirá o uso de recursos avançados de CFD, como por exemplo, a possibilidade de uso de geometrias complexas, malhas móveis e não-estruturadas, correções para malhas não-ortogonais, diferentes esquemas de interpolação, bons *solvers* para o sistema linear de equações discretizado, possibilidade de paralelização do processamento de dados entre outras vantagens que um bom pacote de CFD fornece.

Para atingir esses objetivos fez-se uma busca por um *software* que fosse o mais adequado possível. O *software* a ser usado deveria conduzir a soluções precisas, consumindo tempo e recursos computacionais não proibitivos. Além disso, teria que ser robusto e versátil. Por fim, o *software* deveria ter o melhor balanço possível entre estas propriedades e estar de acordo com os objetivos a serem alcançados pelo seu uso. O pacote de CFD que melhor atendeu os objetivos foi o *OpenFOAM* (*Open Source Field Operation and Manipulation*) ([OPENFOAM, 2008](#)), que além de ter seu código aberto permite o uso de muitos recursos avançados de CFD.

1.3 Estrutura da Dissertação

No [Capítulo 2](#) serão apresentados aspectos referentes à reologia, à modelagem matemática e à simulação de fluidos viscoelásticos. Será feito uma descrição das caracterís-

ticas dos fluidos viscoelásticos e serão apresentadas as principais funções materiais usadas para tratar o problema do escoamento desse tipo de fluido. Serão apresentadas as formulações matemáticas para fluidos newtonianos e fluidos viscoelásticos com o objetivo de mostrar as diferenças entre elas e também serão apresentados os modelos constitutivos usados para representação do comportamento de fluidos poliméricos.

No **Capítulo 3** é feita uma descrição das metodologias numéricas que são usadas para resolver escoamentos envolvendo fluidos viscoelásticos, pois devido às dificuldades de se resolver este tipo de escoamento, várias metodologias têm sido propostas na tentativa de contorná-las.

O **Capítulo 4** tem como objetivos justificar o porquê da escolha do *software OpenFOAM* (OPENFOAM, 2008) e também detalhar seu funcionamento como ferramenta de CFD. Será feita uma descrição do *OpenFOAM*, da sua estrutura e do procedimento básico para se fazer a simulação de um caso usando este *software*. Serão apresentados também os diferentes *solvers* disponíveis, esquemas de interpolação, etc., que o usuário pode escolher para simular um problema. Serão discutidos também aspectos relativos à implementação de novos *solvers*, linguagem de programação C++ e descrição de algumas classes importantes do *software*

No **Capítulo 5** será detalhado o algoritmo e a implementação do módulo *viscoelasticFluidFoam* feita durante a realização deste trabalho.

No **Capítulo 6** são apresentados resultados obtidos com o *solver* desenvolvido. Os resultados obtidos são comparados com dados numéricos e experimentais obtidos da literatura e são usados para validação do *solver*. É feita também a avaliação da implementação de vários modelos constitutivos e o uso de modelos com multimodos.

Por fim, no **Capítulo 7**, são apresentadas as conclusões e algumas sugestões para trabalhos futuros.

Capítulo 2

Modelos Constitutivos e Propriedades de Fluidos Poliméricos

Neste capítulo serão apresentados aspectos referentes à reologia, modelagem matemática e simulação de fluidos viscoelásticos.

Primeiramente será feita uma descrição dos fluidos poliméricos e serão apresentadas as principais funções materiais usadas para tratar o problema de escoamento desse tipo de fluido. Após isso será tratada a modelagem matemática do problema. É apresentada a formulação matemática para fluidos newtonianos e para fluidos viscoelásticos visando mostrar as diferenças entre elas. Como existem diferentes teorias que são usadas para descrever o comportamento reológico de um fluido viscoelástico, são apresentados os modelos correspondentes a cada uma destas teorias.

2.1 Comportamento reológico de materiais poliméricos

Como já comentado, é enorme o uso de produtos de origem polimérica pela sociedade moderna e a tendência é que sua utilização cresça ainda mais; Isto por que os polímeros sintéticos estão cada vez mais substituindo materiais convencionais como metais e

madeira, devido, principalmente, ao seu menor custo, facilidade de processamento e propriedades físicas e mecânicas adequadas. Materiais bem distintos podem ser obtidos utilizando-se diferentes polímeros, por exemplo, o plástico utilizado na fabricação de sacolas de supermercados e o plástico que reveste televisores, monitores de computador ou os usados em automóveis. Para cada uma destas aplicações propriedades específicas do material polimérico são requeridas.

Todo o contingente de produtos de origem polimérica passa anteriormente por algum processo de transformação, ou seja, operações que transformam a matéria-prima (resina virgem) em produtos finais para consumo. Como exemplos desses processos de transformação têm-se a extrusão, moldagem por injeção, moldagem por sopro, termomoldagem, entre muitos outros, sendo cada um destes processos adequado à produção de um determinado tipo de produto. O que há de comum na grande maioria destes processos, é uma etapa na qual o material, originalmente no estado sólido, é fundido possibilitando assim que tome uma nova forma desejada (BRETAS; DAVILA, 2005).

Para o estudo, avaliação e entendimento de qualquer processo no qual o escoamento deste tipo de fluido seja de fundamental importância, é necessário conhecer este comportamento e em muitos casos, saber representá-lo de uma forma quantitativa, por meio de equações matemáticas. Exemplos de situações onde isto ocorre é na caracterização de polímeros pela obtenção de medidas reológicas e na modelagem, simulação e otimização de processos que envolvam o escoamento de fluidos poliméricos. Assim, um conhecimento das diferentes categorias de materiais que podemos encontrar, do ponto de vista de comportamento reológico, é indispensável. De modo geral, os diferentes tipos de comportamento reológico podem ser enquadrados em uma das seguintes categorias:

- **Sólidos de Hooke:** Sólidos, perfeitamente elásticos, que sofrem deformações finitas sob ação de uma tensão, retomando sua forma original com a remoção desta, sendo que a relação entre tensão e deformação é linear.

A função material que caracteriza estes materiais é o módulo de Young ou, simplesmente, módulo elástico, que define a proporcionalidade entre tensão e deformação.

- **Fluidos Newtonianos:** são fluidos puramente viscosos, ou seja, deformam-se continuamente sob a ação de uma tensão de cisalhamento, que apresentam relação linear entre tensão e taxa de deformação. A função material que caracteriza estes materiais é a viscosidade, que define a proporcionalidade entre tensão e taxa de deformação. Materiais compostos por moléculas de baixa massa molar (menor que 1000, monômeros, N_2 , O_2 , H_2O , etc.) são exemplos típicos de materiais que apresentam comportamento de fluido Newtoniano. A dinâmica dos fluidos newtonianos é representada pelas equações de Navier-Stokes e pode-se dizer que seu comportamento já é bem entendido nos dias de hoje.
- **Fluidos puramente viscosos não-Newtonianos:** são fluidos que apresentam resposta puramente viscosa, mas que não apresentam uma relação linear entre tensão e taxa de deformação. Enquadram-se nesta classificação materiais que apresentam tensão mínima de escoamento e materiais com viscosidade dependente do tempo e/ou da taxa de deformação. Para descrever a dinâmica do escoamento deste tipo de material é necessária uma equação constitutiva específica para viscosidade ou tensão, a qual é geralmente explícita em termos de taxa de deformação, de maneira que o número de incógnitas do problema não é alterado com relação ao necessário para resolver o escoamento de um fluido Newtoniano.
- **Fluidos viscoelásticos:** são materiais que podem apresentar as duas características anteriores, ou seja, apresentar propriedades viscosas e elásticas ao mesmo tempo. Estes materiais são constituídos por moléculas complexas e de elevada massa molar (moléculas muito extensas e estruturadas), como, por exemplo, soluções poliméricas ou polímeros fundidos, no qual

a dinâmica do escoamento não é descrita por completo pelas clássicas equações de Navier-Stokes. Para descrever a dinâmica do escoamento deste tipo de material é necessária uma equação constitutiva adicional para o campo de tensões, sendo que as componentes de tensão aparecem como incógnitas adicionais com relação às consideradas na análise de escoamentos de fluidos Newtonianos.

O estudo dos materiais que apresentam viscoelasticidade é cada vez mais necessário e é de grande importância, não só na indústria de processamento de artigos plásticos em geral, mas também na indústria de produção de tintas, processamento de alimentos, indústria de cosméticos, estudos sobre a eficiência de óleos lubrificantes, o movimento de fluidos biológicos, como por exemplo, o sangue e a saliva, estudos de moléculas biológicas complexas como as de DNA, entre outras. Como consequência, tanto do ponto de vista experimental como teórico, o estudo deste tipo de fluido teve um grande avanço a partir dos anos de 1950 e principalmente após os anos de 1970.

As características desse tipo de fluidos pode facilmente ser vista em experimentos (BIRD et al., 1987). Um desses experimentos consiste em submeter um fluido polimérico a uma taxa de deformação constante até o estado estacionário ser atingido, e então, em um dado instante, cessar o movimento. A resposta obtida para fluidos newtonianos é que a tensão cairia instantaneamente à zero. Porém, para fluidos poliméricos observa-se que a tensão terá um valor finito e decairá exponencialmente com o tempo, apresentando o que é conhecido como memória elástica. Esta tensão residual é causada pelo estiramento e alinhamento das moléculas poliméricas pela força de deformação aplicada. Assim, para fluidos poliméricos, o tempo necessário para a tensão se dissipar é definido pelo tempo de relaxação das moléculas.

A **Figura 2.1** mostra o efeito conhecido como inchamento do extrudado (*die swell*) no qual a seção vertical do escoamento aumenta em tamanho após a saída da matriz, devido à elasticidade do fluido (BIRD et al., 1987). No experimento ilustrado, considera-

se um fluido que sai de um capilar com diâmetro D para o ar formando um jato com diâmetro D_e . Para fluidos newtonianos D_e pode variar em 13 % para mais do diâmetro D no caso de baixos números de Reynolds ou 13 % para menos do diâmetro do capilar D quando se tem números de Reynolds maiores, porém ainda em escoamento laminar. Já para o caso de fluidos poliméricos o valor de D_e pode aumentar em mais de 300 % o valor de D .

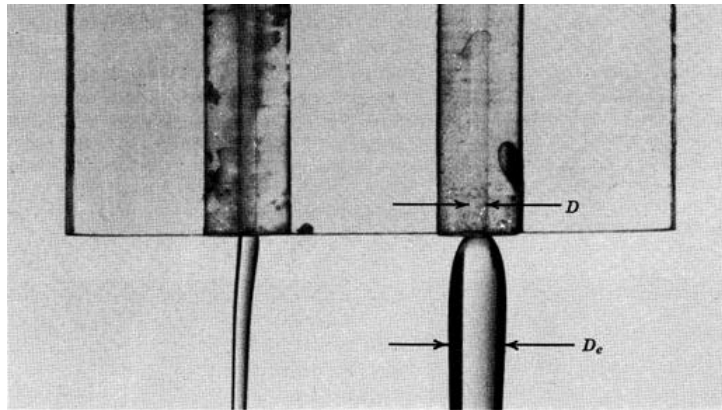


Figura 2.1: Representação do efeito de inchamento de extrudado (*die swell*) para um fluido newtoniano (esquerda) e um fluido polimérico (direita) (Fonte: Bird et al. (1987)).

Outro efeito decorrente da viscoelasticidade é a presença de diferenças de tensões normais em escoamentos por cisalhamento. Em adição às tensões de cisalhamento estes fluidos apresentam tensões extras ao longo das linhas de corrente, que derivam do estiramento e alinhamento das cadeias poliméricas ao longo das linhas de corrente. O experimento clássico "*rod-climbing*" permite visualizar este comportamento e consiste em submeter um fluido em um recipiente a uma agitação por meio de um eixo rotatório imerso no fluido. No recipiente contendo fluido newtoniano se formará uma depressão junto ao eixo do agitador. Já para um fluido polimérico, o resultado é intuitivamente inesperado: o fluido sobe junto ao eixo do agitador. Nesse experimento as linhas de corrente são círculos fechados e a tensão extra ao longo destas linhas de corrente "estrangulam" o fluido e forçam-no para dentro em oposição a força centrífuga e para cima em oposição à força gravitacional. Na Figura 2.2 pode-se visualizar o experimento (BIRD et al., 1987).

O comportamento reológico dos fluidos poliméricos se deve basicamente à sua constituição química. Estes materiais são constituídos por longas cadeias, que podem ser lineares ou ramificadas, com massas molares típicas entre 1000 a 1000000 g/mol . Estas cadeias geralmente estão entrelaçadas, formando estruturas complexas, que apresentam a capacidade de ser modificadas sob a ação de uma tensão, podendo retomar uma posição atingida em um passado recente após a remoção desta.

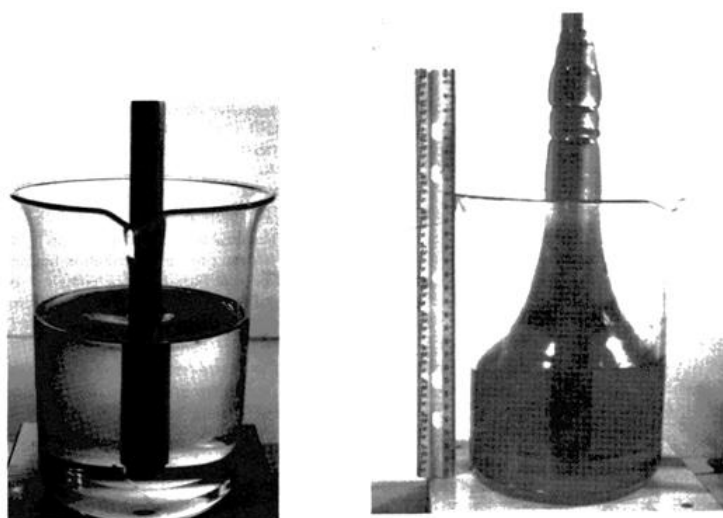


Figura 2.2: Experimento "rod-climbing". A esquerda pode-se ver o comportamento de um fluido newtoniano e a direita o de um fluido polimérico (Fonte: [Bird et al. \(1987\)](#)).

Além da viscoelasticidade, outra característica reológica dos fluidos poliméricos, e talvez a mais conhecida, é possuir uma viscosidade dependente da taxa de deformação aplicada sobre o material, também conhecida como viscosidade não-newtoniana. Baseado no comportamento da dependência da viscosidade com a taxa de deformação, os fluidos poliméricos na sua grande maioria são classificados como fluidos pseudoplásticos (*shear-thinning*), nos quais a viscosidade diminui com a taxa de deformação.

Muitos outros experimentos que demonstram o efeito da viscoelasticidade em fluidos poliméricos podem ser encontrados em [Bird et al. \(1987\)](#), [Macosko \(1994\)](#) e [Larson \(1988\)](#).

2.2 Números Adimensionais importantes para escoamentos de Fluidos Viscoelásticos

O processo transiente de relaxação das tensões possui um tempo característico conhecido como tempo de relaxação. Do tempo de relaxação podemos obter um número adimensional de fundamental importância no estudo de fluidos viscoelásticos, o número de *Deborah* (De) (Equação 2.1) que é dado pela razão entre o tempo de relaxação λ e o tempo característico do experimento, t_c .

$$De = \frac{\lambda}{t_c} \quad (2.1)$$

O número de *Deborah* nos fornece uma relação de quão pronunciado será o efeito elástico, ou seja, valores altos deste número indicam que o efeito elástico é maior, já quando esses valores tenderem a zero teremos escoamento puramente viscoso (LARSON, 1988; BIRD et al., 1987).

Na verdade, como a grande maioria dos polímeros comerciais se constitui por cadeias moleculares de diferentes tamanhos (polidispersos), apresentando uma distribuição de massa molar, não existe um único tempo de relaxação, mas sim um espectro de relaxação, constituído pelos tempos de relaxação individuais de cada molécula. O tempo de relaxação usado para o cálculo do número de *Deborah* será o que tiver maior importância dentro do espectro de relaxação.

Outro grupo adimensional utilizado para quantificar a importância dos efeitos elásticos em escoamentos é o número de *Weissenberg* (We) (Equação 2.2), que é obtido quando é feito o adimensionamento das equações constitutivas para fluidos viscoelásticos e define-se como:

$$We = \lambda \dot{\gamma}_c \quad (2.2)$$

onde, $\dot{\gamma}_c$ é a taxa de deformação característica. Para um duto de seção circular $\dot{\gamma}_c = U_c/R$, com U_c sendo igual à velocidade característica e R o raio do duto.

Em geral os valores de De e We somente são coincidentes para o caso de escoamentos estacionários, já que neste caso costuma-se utilizar o recíproco da taxa de deformação característica como tempo característico.

O número adimensional mais conhecido na área de CFD é o número de *Reynolds* (Re) (Equação 2.3). Este número surge naturalmente quando se faz o adimensionamento das equações de conservação de quantidade de movimento e serve para caracterizar um escoamento. Números baixos de Re indicam que o escoamento é laminar, ou seja, não existe turbilhonamento. Números de Re elevados indicam que o escoamento é turbulento e números intermediários caracterizam a zona de transição. Deve-se lembrar que este número é dependente da geometria onde ocorre o escoamento e por isso os valores que caracterizam cada regime de escoamento variam de acordo com a geometria.

$$Re = \frac{\rho U_c L_c}{\eta_0} \quad (2.3)$$

onde, L_c é o comprimento característico, ρ a massa específica e η_0 a viscosidade dinâmica a taxa de deformação nula.

Outro número importante em CFD é o número de *Courant* (Co). Este número é utilizado como critério para garantir a estabilidade numérica de métodos explícitos na resolução de processos transientes, ou seja, a partir desse número pode-se obter o valor do passo de tempo que garanta soluções estáveis. O número de Courant médio é dado pela relação entre a velocidade média \underline{U} , o passo de tempo Δt e o tamanho da célula de referência Δx como mostrado pela Equação 2.4.

$$Co = \frac{\underline{U} \Delta t}{\Delta x} \quad (2.4)$$

2.3 Funções Materiais para Fluidos Poliméricos

Para se poder representar o escoamento de um líquido polimérico é necessário conhecer, não somente as propriedades físicas do fluido, mas também parâmetros relaciona-

dos com a geometria e às condições do escoamento, como condições de contorno.

Para fluidos newtonianos, as propriedades físicas podem ser representadas por alguma constante material, como por exemplo, a viscosidade newtoniana η que é medida em pressão e temperaturas fixas. Sua definição vem da lei da viscosidade de Newton dada pela **Equação 2.5**:

$$\tau_{yx} = \eta \dot{\gamma}_{yx} \quad (2.5)$$

onde τ_{yx} é a tensão de cisalhamento e $\dot{\gamma}_{yx}$ é a taxa de deformação. Os sub-índices x e y , correspondem às direções características do escoamento, sendo x a direção do escoamento e y a direção de variação do perfil de velocidade (BIRD et al., 1987).

Para o caso dos fluidos poliméricos, não se tem uma constante material, já que as propriedades desses fluidos são funções da taxa de deformação, do tempo, etc. Assim, para os fluidos poliméricos, ao invés de se usar constantes materiais, é mais correto usar funções materiais.

As funções materiais para fluidos poliméricos podem ser classificadas em duas grandes classes: funções materiais para escoamentos por cisalhamento e funções materiais para escoamentos livres de cisalhamento. Estas funções materiais podem ser dependentes do tempo (em escoamentos transientes) ou não (em escoamentos estacionários).

Segundo Bird et al. (1987) a tensão em estado estacionário para um escoamento com cisalhamento puro depende somente da taxa de deformação. A viscosidade não-newtoniana $\eta(\dot{\gamma})$ é definida analogamente à viscosidade newtoniana como mostrado na **Equação 2.6**:

$$\tau_{yx} = \eta(\dot{\gamma}) \dot{\gamma}_{yx} \quad (2.6)$$

Para escoamentos por cisalhamento pode-se definir outras funções materiais importantes, como os coeficientes de tensões normais Ψ_1 e Ψ_2 que relacionam as

diferenças de tensões normais N_1 (Equação 2.7) e N_2 (Equação 2.8) no escoamento com a taxa de deformação:

$$N_1 = \tau_{xx} - \tau_{yy} = \Psi_1(\dot{\gamma})\dot{\gamma}_{yx}^2 \quad (2.7)$$

$$N_2 = \tau_{yy} - \tau_{zz} = \Psi_2(\dot{\gamma})\dot{\gamma}_{yx}^2 \quad (2.8)$$

É comum encontrar trabalhos que consideram $N_1 = -(\tau_{xx} - \tau_{yy})$ e $N_2 = -(\tau_{yy} - \tau_{zz})$. O primeiro coeficiente de tensões normais (Ψ_1) tem o mesmo comportamento da viscosidade frente à taxa de deformação, ou seja, geralmente diminui com o aumento da taxa de deformação. Este parâmetro tem um valor sempre positivo, exceto raras exceções. O segundo coeficiente de tensões normais (Ψ_2) é mais difícil de ser medido experimentalmente, e conhece-se pouco sobre valores experimentais deste parâmetro. Sabe-se que o coeficiente Ψ_2 é menor que Ψ_1 em módulo (10 % do valor de Ψ_1) e é sempre negativo (BIRD et al., 1987). Deve-se lembrar que estes coeficientes são ambos nulos para fluidos newtonianos sob cisalhamento puro.

Das três funções materiais apresentadas, a viscosidade é a mais conhecida pela maior facilidade de ser determinada experimentalmente. Em baixas taxas de deformação a tensão de cisalhamento é proporcional a taxa de deformação e a viscosidade nesta região é constante, sendo chamada de viscosidade a deformação nula η_0 (*zero-shear-rate viscosity*). Sob altas taxas de deformação a viscosidade diminui com o aumento da taxa de deformação para a maioria dos polímeros líquidos.

A viscosidade é uma propriedade muito importante no estudo de fluidos poliméricos, pois estes fluidos são geralmente pseudoplásticos, isto é, sua viscosidade diminui com a taxa de deformação aplicada. Este comportamento pode ser visto nas curvas típicas de viscosidade mostradas na Figura 2.3 para diferentes temperaturas, sendo que a viscosidade diminui com o aumento de temperatura, analogamente ao que ocorre com os fluidos de baixa massa molar no estado líquido em geral.

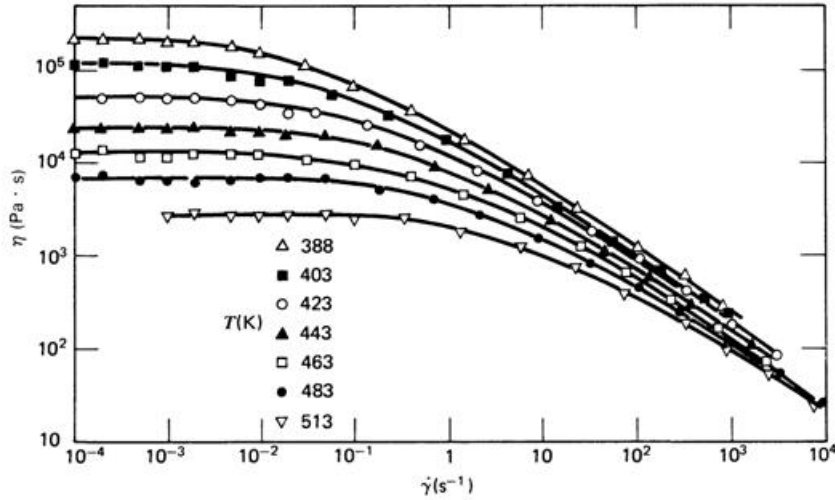


Figura 2.3: Curvas típicas da viscosidade não-newtoniana para o polietileno de baixa densidade em função da taxa de deformação, sob diferentes temperaturas (Fonte: Bird et al. (1987)).

Para escoamento em estado estacionário e para cisalhamento simples uma medida da elasticidade de um fluido pode ser obtida pela razão das tensões $(\tau_{xx} - \tau_{yy})/\tau_{xy}$. Esta medida é zero para fluidos newtonianos e também para não-newtonianos quando estão escoando a baixas taxas de deformação.

Em escoamentos livres de cisalhamento também podem ser definidas funções materiais. Para escoamentos estacionários, definem-se duas funções viscosidade $\bar{\eta}_1$ (Equação 2.9) e $\bar{\eta}_2$ (Equação 2.10) que são relacionadas às diferenças de tensões normais.

$$\tau_{zz} - \tau_{xx} = \bar{\eta}_1(\dot{\epsilon}, b)\dot{\epsilon} \quad (2.9)$$

$$\tau_{yy} - \tau_{xx} = \bar{\eta}_2(\dot{\epsilon}, b)\dot{\epsilon} \quad (2.10)$$

onde $\dot{\epsilon}$ é a taxa de alongação e b é um parâmetro que define o tipo de escoamento. Para o caso especial de escoamento livre de cisalhamento em estado estacionário $b = 0$ e $\bar{\eta}_2 = 0$ e $\bar{\eta}_1$ é igual a viscosidade elongacional $\bar{\eta}$ como representado pela Equação 2.11.

$$\bar{\eta}(\dot{\epsilon}) = \bar{\eta}_1(\dot{\epsilon}, 0)\dot{\epsilon}; \quad \bar{\eta}_2(\dot{\epsilon}, 0) = 0 \quad (2.11)$$

Para $\dot{\epsilon} > 0$, $\bar{\eta}$ representa o fluxo elongacional, já para $\dot{\epsilon} < 0$ representa a compressão biaxial. A viscosidade elongacional é também chamada algumas vezes de "viscosidade de Trouton" ou "viscosidade extensional".

Em baixas taxas de alongação a viscosidade elongacional assume um valor constante conhecido como viscosidade elongacional a taxa de alongação nula, que é igual a três vezes a viscosidade de cisalhamento à taxa de deformação nula. Com o aumento da taxa de alongação, esta viscosidade aumenta e depois diminui em altas taxas de alongação, como mostrado na **Figura 2.4**. Este comportamento é observado para a maioria dos polímeros, porém em alguns casos, se tem um comportamento inverso, como por exemplo, o polietileno de alta densidade (BIRD et al., 1987).

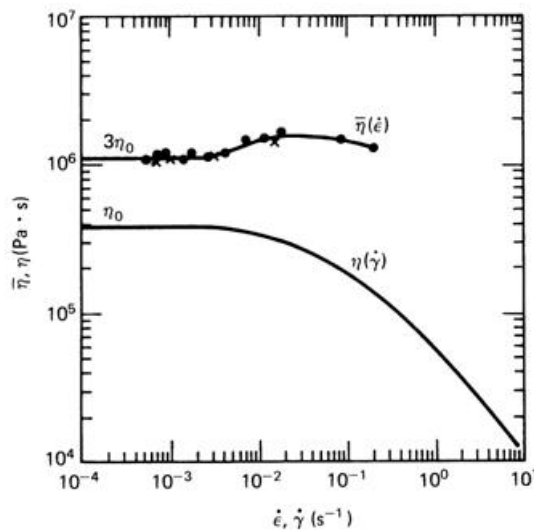


Figura 2.4: Viscosidade elongacional e viscosidade não-newtoniana para uma mistura de poliestireno (Fonte: Bird et al. (1987)).

2.4 Modelagem Matemática

Serão considerados neste estudo escoamentos incompressíveis e isotérmicos. Será apresentada a formulação newtoniana e a usada para descrever o comportamento dos fluidos viscoelásticos.

2.4.1 Equações Governantes para Fluidos Newtonianos

Em geral qualquer problema de mecânica de fluidos tem que satisfazer as equações de conservação de massa ou equação da continuidade, que para fluidos incompressíveis assume a forma da [Equação 2.12](#):

$$\nabla \cdot (U) = 0 \quad (2.12)$$

e de quantidade de movimento ([Equação 2.13](#)):

$$\frac{\partial(\rho U)}{\partial t} + \nabla \cdot (\rho U U) = -\nabla p + \nabla \cdot \tau \quad (2.13)$$

onde ρ é a massa específica, U é o vetor velocidade, p é a pressão e τ é o tensor das tensões. O termo referente a força gravitacional está incorporado no termo correspondente ao gradiente de pressão ([BIRD et al., 1987](#)).

Para completar o sistema de equações para o modelo, necessita-se de uma equação constitutiva mecânica. Para fluidos newtonianos incompressíveis para qualquer geometria tem-se a [Equação 2.14](#):

$$\tau = 2\eta D \quad (2.14)$$

onde η é o coeficiente de viscosidade newtoniana e D é o tensor taxa de deformação dado pela [Equação 2.15](#):

$$D = \frac{1}{2}(\nabla U + [\nabla U]^T) \quad (2.15)$$

Como esta equação constitutiva é explícita em termos de velocidade, ela pode ser substituída na [Equação 2.13](#), resultando na [Equação 2.16](#):

$$\frac{\partial(\rho U)}{\partial t} + \nabla \cdot (\rho U U) - \eta \nabla^2 U = -\nabla p \quad (2.16)$$

Assim, a análise de escoamentos de fluidos Newtonianos se resume à resolução do sistema de equações diferenciais formado pelas [Equações 2.12](#) e [2.16](#), tendo como incógnitas a pressão e as componentes da velocidade.

2.4.2 Formulação do Modelo Matemático para Fluidos Poliméricos

O ponto de partida para a análise de escoamentos incompressíveis de fluidos poliméricos também é a equação da continuidade e a equação de balanço de quantidade de movimento. No entanto, quando o interesse é resolver o problema de um escoamento de fluido não-newtoniano surgem algumas dificuldades adicionais, pois equações constitutivas mais complexas devem ser resolvidas simultaneamente às equações de conservação de massa e quantidade de movimento.

A equação da continuidade não tem sua forma alterada em decorrência do tipo de equação constitutiva mecânica utilizada, de maneira que a [Equação 2.12](#) também é utilizada na modelagem de escoamento de fluidos poliméricos.

Já no caso do balanço de quantidade de movimento, para fluidos viscoelásticos, a [Equação 2.13](#) costuma ser reescrita dividindo o termo de tensão em duas contribuições ([Equação 2.17](#)):

$$\frac{\partial(\rho U)}{\partial t} + \nabla \cdot (\rho U U) = -\nabla p + \nabla \cdot \tau_s + \nabla \cdot \tau_p \quad (2.17)$$

onde τ_s é a contribuição do solvente para o tensor das tensões e τ_p é a contribuição polimérica para este tensor. Desta forma se considera que os polímeros podem ser considerados como sendo uma mistura de um solvente e um soluto polimérico. O solvente possui comportamento newtoniano ([Equação 2.18](#)) ([BIRD et al., 1987](#)):

$$\tau_s = 2\eta_s D \quad (2.18)$$

onde η_s é a viscosidade dinâmica do solvente. O tensor das tensões adicionais τ_p (tensões "elásticas" ou "poliméricas") na [Equação 2.17](#) deve ser obtido através de equações constitutivas provenientes de teorias sobre reologia de fluidos, como por exemplo, a teoria cinética, a teoria de redes de soluções concentradas e polímeros fundidos e a teoria da reptação ([LARSON, 1988](#)).

Como este tensor não pode, geralmente, ser escrito explicitamente em função do gradiente de velocidades como no caso da contribuição Newtoniana, o sistema de equações a ser analisado passa a ser composto pelas Equações 2.12 e 2.17, juntamente com outra equação diferencial para a definição de τ_P . Além disso, as componentes de τ_P passam também a ser incógnitas do problema, juntamente com a pressão e as componentes da velocidade.

2.5 Equações Constitutivas para Fluidos Poliméricos

Existe um grande número de equações constitutivas, que buscam descrever o comportamento reológico dos fluidos poliméricos. Estas equações podem ser enquadradas em diferentes grupos, de acordo com a sua forma, sua natureza matemática e sua capacidade de predição de funções materiais.

2.5.1 Fluido Newtoniano Generalizado (FNG)

Consiste na generalização do modelo de fluido newtoniano para fluidos nos quais a viscosidade é uma função da magnitude do tensor taxa de deformação. Os modelos de FNG consistem na primeira generalização da mecânica de fluidos clássica para a mecânica dos fluidos não-newtonianos. Nesta situação os efeitos elásticos não são preditos, uma vez que essa categoria de modelos ainda não considera o cálculo de τ_P . Esses modelos podem ser aplicados satisfatoriamente somente em casos onde ocorrem escoamentos estacionários por cisalhamento puro e taxas de deformação elevadas. Para FNG tem-se que substituir a Equação 2.19 na Equação 2.17:

$$\tau_S = 2\eta_s(\dot{\gamma})D, \quad \tau_P = 0 \quad (2.19)$$

onde a viscosidade η_s é agora uma função de $\dot{\gamma}$ que é igual ao segundo invariante do tensor taxa de deformação D . Existem muitos modelos empíricos que fornecem

relações matemáticas para a viscosidade em função da taxa de deformação mas geralmente só são válidos para determinados fluidos ou em determinadas regiões de aplicação (BIRD et al., 1987).

O modelo mais simples e mais conhecido para a viscosidade dependente da taxa de deformação é a Lei da Potência (Equação 2.20) (OSTWALD, 1925; WAELE, 1923), dada por:

$$\eta_s(\dot{\gamma}) = K\dot{\gamma}^{n-1} \quad (2.20)$$

onde K é um parâmetro de consistência e n é o expoente do modelo Lei da Potência. Esses parâmetros são dependentes do fluido e são obtidos pelo ajuste de curvas a dados experimentais. Este modelo, por ter uma forma simples, permite a obtenção de soluções analíticas para uma grande variedade de escoamentos de fluidos. É possível representar o efeito pseudoplástico (*shear-thinning*), tão comum nos materiais de uso diário (por exemplo, a manteiga espalha-se mais facilmente quando a deformação imposta pela faca aumenta).

Outro modelo muito usado é o de Carreau-Yasuda (Equação 2.21) (CARREAU, 1968; YASUDA, 1979), o qual descreve bem a viscosidade para uma ampla faixa de taxa de deformação.

$$\frac{\eta_s - \eta_{s_\infty}}{\eta_{s_0} - \eta_{s_\infty}} = [1 + (\lambda\dot{\gamma})^a]^{\frac{n-1}{a}} \quad (2.21)$$

onde a viscosidade a baixas deformações η_{s_0} , a viscosidade em altas taxas de deformação η_{s_∞} , a constante de tempo λ e as constantes n e a são parâmetros característicos do fluido. O modelo original de Carreau considera $a = 2$, sendo este parâmetro introduzido na equação por Yasuda (BIRD et al., 1987).

O modelo de fluido newtoniano generalizado tem a deficiência de não predizer os efeitos elásticos característicos dos fluidos poliméricos. Do ponto de vista numérico o uso de FNG não apresenta dificuldades adicionais em comparação ao caso de fluido newtoniano. Devido a estas características estes modelos são muito utilizados no

estudo de aplicações industriais, como processos de extrusão e injeção, para predição de algumas etapas ou características dos referidos processos que estão associadas somente a fenômenos puramente viscosos e para os quais os efeitos da viscosidade não-newtoniana têm grande importância. Pode-se citar como exemplo destas aplicações, o cálculo de vazão em extrusoras de rosca simples e o cálculo de pressões e tempos para preenchimento de moldes em processos de injeção.

2.5.2 Fluido Viscoelástico Linear (FVL)

O modelo mais simples para fluido viscoelástico, ou seja, que contempla o caráter viscoso e elástico de um fluido em uma única equação é o modelo para fluido viscoelástico linear.

A primeira equação desenvolvida para descrever o comportamento viscoelástico foi o modelo linear de Maxwell (MAXWELL, 1867), representado esquematicamente na **Figura 2.5**. Este modelo pode ser encarado como uma combinação das equações de Hooke para sólido elástico $\dot{\tau}_E = G\dot{\gamma}_E$ e de Newton para a viscosidade $\tau_V = \mu\dot{\gamma}_V$, lembrando que ambas apresentam relação linear entre tensão e deformação ou taxa de deformação.

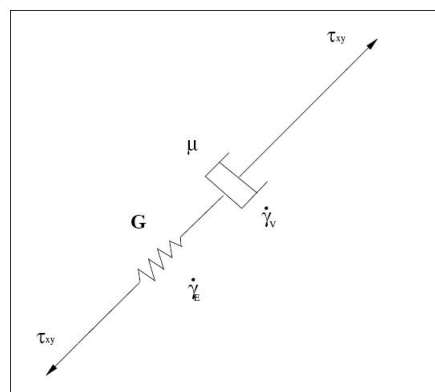


Figura 2.5: Modelo físico mola-amortecedor, representando o modelo viscoelástico linear de Maxwell.

Usando o modelo de Maxwell e considerando a [Equação 2.17](#) temos:

$$\tau_s = 2\eta_s D \quad (2.22)$$

$$\tau_{PK} + \lambda_K \frac{\partial \tau_{PK}}{\partial t} = 2\eta_{PK} D, \quad K = 1, 2, \dots, N \quad (2.23)$$

sendo

$$\tau_P = \sum_{K=1}^N \tau_{PK} \quad (2.24)$$

onde a [Equação 2.23](#) é o modelo linear de Maxwell e N é o número de modos. As constantes λ_K e η_{PK} são o tempo de relaxação e a viscosidade polimérica a taxa de deformação nula para cada modo de relaxação, respectivamente. Assim pode-se resolver a [Equação 2.23](#) para N modos (multimodo) de relaxação (espectro de relaxação) e obter a tensão τ_P pela superposição da tensão calculada para cada modo individual, como representado na [Equação 2.24](#).

A formulação multimodo será também usada para todos os modelos que serão apresentados nos itens a seguir. O uso desta formulação torna possível a obtenção de soluções mais realistas e condizentes com dados experimentais. A maioria dos materiais poliméricos compõe-se de estruturas moleculares de diferentes tamanhos (polidispersos) e conseqüentemente apresentam diferentes tempos de relaxação (espectro de relaxação). O espectro de relaxação pode ser considerado desde um até um número N de modos necessários para uma boa representação do fluido considerado.

Deve-se considerar que com o uso de multimodos é exigido um maior esforço computacional, pois cada modo adicional envolve a resolução de uma equação constitutiva a mais. Para se ter uma idéia do acréscimo do custo computacional considere que se para um modelo com um modo levaria em torno de 3 unidades de CPU de tempo de processamento o uso de 4 modos resultaria em um tempo computacional de 8 unidades de CPU ([AZAIEZ et al., 1996a](#)). No passado era totalmente inviável o uso de modelos na forma multimodo devido às restrições computacionais. No

entanto, recentemente, com o avanço na área computacional, o uso de modelos na forma multimodo é totalmente aceitável e vem sendo muito usado, conseguindo-se uma representação muito mais adequada de dados experimentais.

2.5.3 Fluido Viscoelástico Não-Linear

Os modelos para fluido viscoelástico não-linear são mais complexos, contudo permitem descrever, ao menos qualitativamente efeitos elásticos e características não-lineares, como diferenças de tensões normais e viscosidade não-newtoniana e por isso serão estudados neste trabalho. Existe uma grande variedade destes modelos, sendo que cada um é capaz de prever um determinado conjunto de fenômenos, podendo apresentar deficiências em outros.

Os modelos diferenciais não lineares podem ser obtidos a partir do modelo para fluido viscoelástico linear, na sua forma diferencial. As modificações realizadas consistem na substituição das derivadas em relação ao tempo pela derivada convectiva no tempo e/ou na inclusão de termos não-lineares e parâmetros nas equações. Essa derivada surgiu pela necessidade de obrigar as equações de estado constitutivas lineares a serem objetivas, isto é, a serem independentes do movimento dos eixos dos sistemas de coordenadas utilizados (BIRD et al., 1987).

A relação que define a derivada convectiva superior no tempo do tensor das tensões é dada por (Equação 2.25):

$$\overset{\nabla}{\tau}_{PK} = \frac{D}{Dt} \tau_{PK} - \left[\nabla U^T \cdot \tau_{PK} \right] - \left[\tau_{PK} \cdot \nabla U \right] \quad (2.25)$$

ou também, para tensores simétricos (Equação 2.26):

$$\overset{\nabla}{\tau}_{PK} = \frac{D}{Dt} \tau_{PK} - \left[\tau_{PK} \cdot \nabla U \right] - \left[\tau_{PK} \cdot \nabla U \right]^T \quad (2.26)$$

onde $\frac{D}{Dt}\tau_{PK}$ é a derivada material dada por (Equação 2.27):

$$\frac{D}{Dt}\tau_{PK} = \frac{\partial}{\partial t}\tau_{PK} + U \cdot \nabla\tau_{PK} \quad (2.27)$$

Estes modelos não estão limitados a pequenas deformações como é o caso do modelo de fluido viscoelástico linear. São modelos mais realistas, que permitem obter, no mínimo, informações qualitativas em relação a efeitos viscoelásticos lineares e não-lineares em diversos escoamentos, dos mais simples aos mais complexos. A escolha de uma equação constitutiva com um apropriado nível de sofisticação, para um determinado escoamento, depende de muitos fatores e requer o balanço entre a complexidade matemática, o número de parâmetros físicos adicionais a se estimar e a necessidade de se descrever as propriedades das macromoléculas do polímero (BIRD et al., 1987; LARSON, 1988; MACOSKO, 1994). Algumas vezes os modelos são encontrados usando o tensor configuração, que possui relação com a configuração espacial (extensão) das moléculas de polímero e pode ser facilmente relacionado com o tensor das tensões. No entanto, apesar de se ter algumas vantagens em relação a estabilidade, se tem algumas desvantagens com relação a maior quantidade de memória computacional requerida (WAPPEROM; HULSEN, 1995).

2.5.3.1 Modelos de Oldroyd-B, UCM e White-Metzner (WM)

Um modelo muito conhecido é o de Oldroyd-B (OLDROYD, 1950). Este modelo deriva da teoria cinética para soluções poliméricas concentradas e polímeros fundidos (BIRD et al., 1987).

A cadeia polimérica é representada por um conjunto de duas esferas unidas por uma mola como mostrado na Figura 2.6. Nesta configuração as esferas representam o centro de massa do sistema e estão relacionadas com a interação hidrodinâmica entre o solvente e as macromoléculas da solução polimérica (a força de arrasto viscoso do solvente sobre as macromoléculas). As molas representam o efeito de elasticidade das

macromoléculas ou o efeito restaurador do polímero. Esta configuração esfera/mola denominada "*dumbbell*" é simplificada assumindo-se um comportamento de mola linear ou mola de Hooke.

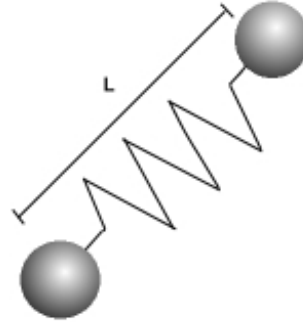


Figura 2.6: Representação de uma macromolécula de polímero como um "*dumbbell*".

A expressão matemática do modelo de Oldroyd-B é dado por (Equação 2.28):

$$\tau_{PK} + \lambda_K \overset{\nabla}{\tau}_{PK} = 2\eta_{PK} D \quad (2.28)$$

As constantes desta equação têm o mesmo significado do modelo linear descrito anteriormente. Este modelo produz valores constantes da viscosidade de cisalhamento em relação à taxa de deformação, estima a primeira diferença de tensões normais (N_1) como sendo uma função quadrática da taxa de cisalhamento e uma segunda diferença de tensões normais (N_2) nula. O modelo de Oldroyd-B consegue representar bem certos tipos de fluidos que apresentam elasticidade ideal, também conhecidos como fluidos de "Boger".

Para escoamentos extensionais o modelo de Oldroyd-B possui a deficiência de calcular uma viscosidade extensional infinita para valores de taxa de deformação tais que $(2\dot{\epsilon}\lambda) > 1$, onde λ é o maior tempo de relaxação do espectro de relaxação.

Se tomarmos a viscosidade do solvente como sendo nula, ou seja, desprezarmos a contribuição do solvente na Equação 2.17 o modelo de Oldroyd-B recai em um modelo também muito conhecido na literatura chamado de UCM (*Upper Convected*

Maxwell). Este modelo é muito usado para testar metodologias numéricas, uma vez que a ausência da parte correspondente a viscosidade do solvente torna mais crítica a estabilidade numérica do problema.

Para se conseguir uma melhor representação dos dados obtidos experimentalmente existe uma classe de modelos similares ao Oldroyd-B, mas que consideram que η_{PK} e λ_K são funções do segundo invariante do tensor taxa de deformação D dado por $II_D = \dot{\gamma} = \sqrt{2D : D}$. Neste caso a viscosidade $\eta_{PK}(II_D)$ e o tempo de relaxação $\lambda_K(II_D)$ podem ser representados por diferentes relações. Um modelo desse tipo que é muito conhecido é o de White-Metzner (WM), que é dado por (Equação 2.29) (LARSON, 1988):

$$\tau_{PK} + \lambda_K(II_D) \overset{\nabla}{\tau}_{PK} = 2\eta_{PK}(II_D)D \quad (2.29)$$

sendo que Larson (1988) apresenta as seguintes relações (Equação 2.30) de dependência para viscosidade polimérica e tempo de relaxação em função de $\dot{\gamma}$:

$$\eta_{PK}(II_D) = \frac{\eta_{0K}}{1 + a\lambda_{0K}II_D} \quad ; \quad \lambda_K(II_D) = \frac{\lambda_{0K}}{1 + a\lambda_{0K}II_D} \quad (2.30)$$

Os parâmetros η_{0K} e λ_{0K} são parâmetros lineares constantes obtidos do modelo de Maxwell e a é um parâmetro que deve ser ajustado a dados experimentais e que pode dar maior ou menor dependência entre a viscosidade polimérica e o tempo de relaxação em relação ao segundo invariante do tensor taxa de deformação.

Outras relações encontradas na literatura são o modelo de Cross (Equação 2.31) (KENNEDY, 1995):

$$\eta_{PK}(II_D) = \frac{\eta_{0K}}{1 + (kII_D)^{1-m}} \quad ; \quad \lambda_K(II_D) = \frac{\lambda_{0K}}{1 + (lII_D)^{1-n}} \quad (2.31)$$

e também o modelo de Carreau-Yasuda (Equação 2.32) (CARREAU, 1968; YASUDA, 1979):

$$\eta_{PK}(II_D) = \eta_{0K} [1 + (kII_D)^a]^{\frac{m-1}{a}} \quad ; \quad \lambda_K(II_D) = \lambda_{0K} [1 + (lII_D)^b]^{\frac{n-1}{b}} \quad (2.32)$$

onde os parâmetros k, m, a, l, n e b são obtidos pelo ajuste de dados experimentais.

Outras formulações podem ser encontradas na literatura, como por exemplo, a adotada no trabalho de [Zatloukal \(2003\)](#).

2.5.3.2 Modelos de Giesekus e Leonov

Outro modelo muito conhecido é o modelo reológico desenvolvido por [Giesekus \(1982\)](#). Este modelo também é baseado em considerações moleculares com sistemas do tipo esfera/mola onde a mola segue a lei de Hooke. Diferentemente do modelo de Oldroyd-B, no modelo de Giesekus foi introduzido um efeito de não-isotropia na definição da força de arrasto sobre as esferas. Este modelo ([Equação 2.33](#)) resulta em uma equação com forma ainda análoga às anteriores, porém contendo termos não lineares dados pelos produtos entre o tensor das tensões.

$$\tau_{PK} + \lambda_K \overset{\nabla}{\tau}_{PK} + \alpha_K \frac{\lambda_K}{\eta_{PK}} (\tau_{PK} \cdot \tau_{PK}) = 2\eta_{PK} D \quad (2.33)$$

A constante α_K é chamada de fator de mobilidade, e está associada com o movimento Browniano anisotrópico ou ao arrasto hidrodinâmico anisotrópico das moléculas de polímero no meio ([BIRD et al., 1987](#)). A inclusão do termo não-linear produz uma variação das propriedades cisalhantes frente à taxa de deformação. Com este modelo se obtém melhores resultados para escoamentos por cisalhamento quando comparado ao modelo de Oldroyd-B, porém, não traz bons resultados em escoamentos livres de cisalhamento ([BIRD et al., 1987](#); [MACOSKO, 1994](#); [SCHLEINIGER; WEINACHT, 1991](#)). Em um escoamento extensional uniaxial, o modelo de Giesekus mostra um comportamento monotônico da evolução da viscosidade extensional em função do tempo para altos valores da taxa de deformação, levando a patamares permanentes e finitos da viscosidade extensional uma vez atingido o estado permanente. Pode-se observar também que o modelo de Giesekus reduz-se ao modelo de Oldroyd-B no limite de pequenas deformações, o que é equivalente a fazer $\alpha_K = 0$. Se $\alpha_K \neq 0$ e $0 \leq \alpha_K \leq 2$ este modelo conduz a uma primeira diferença de tensões normais (N_2)

diferente de zero. No entanto, predições fisicamente coerentes são observadas com $0 \leq \alpha_K \leq 1/2$ (BIRD et al., 1987).

A equação constitutiva de Leonov para propriedades variáveis é dada pela seguinte expressão (Equação 2.34) (LARSON, 1988):

$$\tau_{PK} = \sigma_{PK} - \frac{\eta_{PK}}{\lambda_K} \delta \quad (2.34)$$

onde σ_{PK} é obtido resolvendo-se a Equação 2.35:

$$\nabla \sigma_{PK} + \frac{1}{2\eta_{PK}} \left\{ \sigma_{PK} \cdot \sigma_{PK} - \left(\frac{\eta_{PK}}{\lambda_K} \right)^2 \delta - \frac{1}{3} \left[\text{tr}(\sigma_{PK}) - \left(\frac{\eta_{PK}}{\lambda_K} \right)^2 \text{tr}(\sigma_{PK}^{-1}) \right] \sigma_{PK} \right\} = 0 \quad (2.35)$$

e δ é o tensor identidade, dado por (Equação 2.36):

$$\delta = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.36)$$

Para deformações incompressíveis e planas a equação de Leonov pode ser simplificada para a Equação 2.37:

$$\tau_{PK} + \lambda_K \nabla \tau_{PK} + \frac{1}{2} \frac{\lambda_K}{\eta_{PK}} (\tau_{PK} \cdot \tau_{PK}) = 2\eta_{PK} D \quad (2.37)$$

Esta equação corresponde a um caso particular da equação de Giesekus com $\alpha_K = 1/2$ (LARSON, 1988).

2.5.3.3 Modelos do tipo FENE

O modelo de mola linear tem a deficiência da macromolécula se deformar indefinidamente sem qualquer restrição. Por este motivo muitos modelos constitutivos de fluidos viscoelásticos passaram a se basear em descrições de molas não lineares nas quais uma restrição de deformação máxima L^2 é imposta. Esses modelos são conhecidos como "dumbbell-FENE" (*Finitely Extensible Nonlinear Elastic*) (WARNER, 1972). A partir do modelo original conhecido como FENE diversas derivações foram apresentadas

posteriormente, como por exemplo, o FENE-P, (*Finitely Extensible Nonlinear Elastic-Peterlin*) (BIRD et al., 1980) e o FENE-CR (*Finitely Extensible Nonlinear Elastic-Chilcott and Rallison*) (CHILCOTT; RALLISON, 1988) entre muitos outros que não serão discutidos neste trabalho mas que o leitor pode encontrar na literatura, como em Lielens et al. (1999).

O modelo FENE-P pode ser representado como sendo (Equação 2.38):

$$\left(1 + \frac{\frac{3}{(1-3/L_K^2)} + \frac{\lambda_K}{\eta_{PK}} \text{tr}(\tau_{PK})}{L_K^2}\right) \tau_{PK} + \lambda_K \overset{\nabla}{\tau}_{PK} = 2 \frac{1}{(1-3/L_K^2)} \eta_{PK} D \quad (2.38)$$

e o modelo FENE-CR (Equação 2.39):

$$\left(\frac{L_K^2 + \frac{\lambda_K}{\eta_{PK}} \text{tr}(\tau_{PK})}{(L_K^2 - 3)}\right) \tau_{PK} + \lambda_K \overset{\nabla}{\tau}_{PK} = 2 \left(\frac{L_K^2 + \frac{\lambda_K}{\eta_{PK}} \text{tr}(\tau_{PK})}{(L_K^2 - 3)}\right) \eta_{PK} D \quad (2.39)$$

O parâmetro L^2 do modelo representa a extensibilidade adimensional das moléculas (máximo comprimento possível a dividir pelo comprimento de equilíbrio). Este modelo reduz-se à conhecida equação de Oldroyd-B quando L^2 tende a infinito. Comparações entre diferentes versões de modelos do tipo FENE podem ser encontradas nos trabalhos de Herrchen e Öttinger (1997), Zhou e Akhavan (2003) e Lielens et al. (1999).

2.5.3.4 Modelos de Phan-Thien-Tanner (PTT) e Feta-PTT

Um modelo muito usado em simulações numéricas de fluidos viscoelásticos é o de Phan-Thien-Tanner (THIEN; TANNER, 1977), conhecido por PTT. Este modelo é derivado da teoria de rede de soluções concentradas e polímeros fundidos (*Network theory of concentrated solutions and melts*) (BIRD et al., 1987). O modelo PTT pode ser apresentado de diferentes formas:

PTT linear (LPTT) (Equação 2.40):

$$\left(1 + \frac{\varepsilon_K \lambda_K}{\eta_{PK}} \text{tr}(\tau_{PK})\right) \tau_{PK} + \lambda_K \overset{\square}{\tau}_{PK} = 2 \eta_{PK} D \quad (2.40)$$

PTT exponencial (EPTT) (Equação 2.41):

$$\exp\left(\frac{\varepsilon_K \lambda_K}{\eta_{PK}} tr(\tau_{PK})\right) \tau_{PK} + \lambda_K \overset{\square}{\tau}_{PK} = 2\eta_{PK} D \quad (2.41)$$

com

$$\overset{\square}{\tau}_{PK} = \frac{D}{Dt} \tau_{PK} - [\nabla U^T \cdot \tau_{PK}] - [\tau_{PK} \cdot \nabla U] + \xi_K (\tau_{PK} \cdot D + D \cdot \tau_{PK}) \quad (2.42)$$

onde $\overset{\square}{\tau}_{PK}$ é a derivada de Gordon-Schowalter, $tr(\tau_{PK})$ é o traço de τ_{PK} que leva em conta a energia elástica da rede. Se ao invés da derivada de Gordon-Schowalter fosse usada a derivada convectiva superior teríamos os conhecidos modelos PTT simplificados, também conhecidos como PTTS. Nestas expressões, ε_K é um parâmetro do modelo relacionado com as suas propriedades extensionais: quando um filamento de fluido é estirado axialmente, a oposição ao estiramento é tanto maior quanto menor for ε_K . Em outras palavras, um ε_K maior corresponde a um fluido com uma viscosidade elongacional menor. Do ponto de vista numérico uma solução é mais facilmente conseguida quando os fluidos apresentam um ε_K maior, contudo $\varepsilon_K < 1$. O parâmetro ξ_K relaciona as diferenças de tensões normais, e geralmente é usado um valor próximo de 0,2 para este parâmetro.

Recentemente vem sendo proposta uma nova classe de modelos constitutivos viscoelásticos que incorporam uma maior flexibilidade. Esses modelos fazem com que a viscosidade η_{PK} e o tempo de relaxação λ_K sejam função do tensor das tensões, assim a viscosidade $\eta_{PK}(\tau)$ e o tempo de relaxação $\lambda_K(\tau)$ são calculados por alguma função que os relaciona ao tensor das tensões. Um modelo que usa essas idéias é o Feta-PTT (*Fixed eta Phan-Thien-Tanner*). A expressão "Fixed eta" refere-se a viscosidade ser fixa pela Equação 2.44 e assim não depender do parâmetro não-linear ε_K . Este modelo é similar ao PTT com a diferença que considera a viscosidade e o tempo de relaxação dependentes da tensão (VERBEETEN et al., 2001). Assim tem-se (Equação 2.43):

$$\left(1 + \frac{\varepsilon_K \lambda_K(\tau)}{\eta_{PK}(\tau)} tr(\tau_{PK})\right) \tau_{PK} + \lambda_K(\tau) \overset{\square}{\tau}_{PK} = 2\eta_{PK}(\tau) D \quad (2.43)$$

Onde $\eta_{PK}(\tau)$ e $\lambda_K(\tau)$ são dados pelas equações 2.44 e 2.45:

$$\eta_{PK}(\tau) = \frac{\eta_{0K}}{\left\{ 1 + A \left[\frac{II_\tau \lambda_{0k}^2}{\eta_{0K}^2} \right]^a \right\}^b} \quad (2.44)$$

$$\lambda_K(\tau) = \frac{\lambda_{0K}}{1 + \frac{\varepsilon_K \lambda_{0K} I_\tau}{\eta_{0K}}} \quad (2.45)$$

Os termos I_τ e II_τ correspondem respectivamente ao primeiro e segundo invariantes da tensão e são dados por 2.46 e 2.47:

$$I_\tau = tr(\tau) \quad (2.46)$$

$$II_\tau = \frac{1}{2}(I_\tau^2 - tr(\tau \cdot \tau)) \quad (2.47)$$

e η_{0K} e λ_{0k} são os parâmetros lineares obtidos do modelo de Maxwell e A , a , b são parâmetros do modelo de Ellis (SCHOONEN, 1998). Neste modelo a viscosidade elongacional é ligeiramente mais sensível às variações do parâmetro ε_K se comparada com o primeiro coeficiente de tensões normais. Como resultado, propriedades viscosas e elongacionais podem ser controladas de forma mais independente. Maiores informações sobre este modelo podem ser conseguidas em Verbeeten et al. (2001).

2.5.3.5 Modelos de Pom-Pom (PP), SXPP, DXPP e DCP

O modelo Pom-Pom introduzido em 1998 por McLeish e Larson (1998) é considerado um novo passo rumo ao entendimento dos fluidos viscoelásticos. Com esse modelo, um comportamento não-linear coerente é conseguido ao mesmo tempo para fluxos elongacionais e cisalhantes. Levando em conta que as propriedades reológicas dos polímeros dependem da estrutura topológica das moléculas poliméricas, este modelo é baseado na teoria de reptação e em uma topologia simplificada para as moléculas ramificadas como esquematizado pela Figura 2.7.

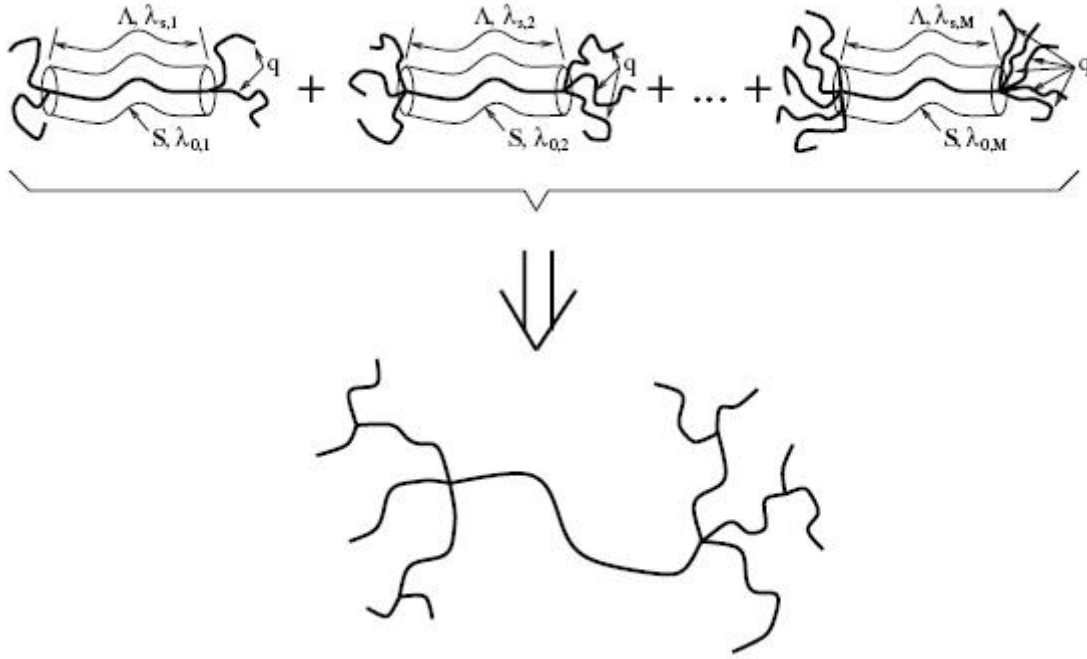


Figura 2.7: Estrutura esquemática da molécula do modelo Pom-Pom (Fonte: Verbeeten et al. (2001)).

O modelo consiste de duas equações desacopladas: uma para a orientação e uma para o estiramento:

$$\overset{\nabla}{S}_{PK} + 2[D : S_{PK}]S_{PK} + \frac{1}{\lambda_{OBK}} \left[S_{PK} - \frac{1}{3}\delta \right] = 0 \quad (2.48)$$

$$\frac{D(\Lambda_{PK})}{Dt} = \Lambda_{PK}[D : S_{PK}] + \frac{1}{\lambda_{SK}} [\Lambda_{PK} - 1], \quad \lambda_{SK} = \lambda_{OSK} e^{-\frac{2}{q}(\Lambda_{PK} - 1)}, \quad \forall \Lambda_{PK} \leq q \quad (2.49)$$

$$\tau_{PK} = \frac{\eta_{PK}}{\lambda_{OBK}} (3\Lambda_{PK}^2 S_{PK} - \delta) \quad (2.50)$$

A **Equação 2.48** é a equação para o tensor orientação S_{PK} , a **Equação 2.49** é a equação para o estiramento dorsal da molécula Λ_{PK} e representa a razão entre o comprimento do tubo e comprimento de equilíbrio e a **Equação 2.50** retorna o valor da tensão viscoelástica τ_{PK} . Nessas equações λ_{OBK} é o tempo de relaxação para a orientação da espinha dorsal do tubo e é obtido do espectro linear de relaxação através de medidas dinâmicas, λ_{OSK} é o tempo de relaxação para o estiramento e q

é a quantidade de ramos existentes desde o começo até o fim da espinha dorsal do tubo e representa a influência do meio sobre o tubo.

O modelo original de Pom-Pom apresenta três desvantagens principais:

- apresenta solução descontínua para o estado estacionário quando submetido a altas taxas de deformação.
- a equação evolutiva para o tensor orientação não apresenta limites quando submetida a altas taxas de alongação ($\dot{\epsilon}\lambda_{OB} > 1$), pois ele é do tipo UCM.
- não prediz a segunda diferença de tensões normais N_2 que é responsável por dar estabilidade ao sistema.

Por causa destas desvantagens, várias formulações baseadas no modelo PP foram posteriormente elaboradas. Todas mantendo as características originais do modelo e tentando resolver os problemas comentados. Uma análise de estabilidade do modelo PP é encontrada no trabalho de [Lee et al. \(2002\)](#).

O eXtended Pom-Pom foi um dos modelos propostos. Ele pode ser encontrado na forma de duas equações (*Double equation eXtended Pom-Pom - DXPP*) como o caso do modelo Pom-Pom e também na forma de uma única equação (*Single equation eXtended Pom-Pom - SXPP*) para o tensor das tensões viscoelásticas ([VERBEETEN, 2001](#)). Abaixo são representadas as duas formulações:

- DXPP:

Evolução do tensor orientação:

$$\begin{aligned} \overset{\nabla}{S}_{PK} + 2[D : S_{PK}]S_{PK} + \\ \frac{1}{\lambda_{OBK} \Lambda_{PK}^2} \left[3\alpha_K \Lambda_{PK}^4 S_{PK} \cdot S_{PK} + (1 - \alpha_K - 3\alpha_K \Lambda_{PK}^4 I_{S.S})S_{PK} - \frac{(1-\alpha_K)}{3}\delta \right] = 0 \end{aligned} \quad (2.51)$$

Evolução do estiramento dorsal do tubo:

$$\frac{D(\Lambda_{PK})}{Dt} = \Lambda_{PK} [D : S_{PK}] + \frac{1}{\lambda_{SK}} [\Lambda_{PK} - 1], \quad \lambda_{SK} = \lambda_{OSK} e^{-\frac{2}{q}(\Lambda_{PK} - 1)} \quad (2.52)$$

Tensão viscoelástica:

$$\tau_{PK} = \frac{\eta_{PK}}{\lambda_{OBK}} (3\Lambda_{PK}^2 S_{PK} - \delta) \quad (2.53)$$

- SXPP:

Tensão viscoelástica:

$$\nabla \tau_{PK} + \lambda(\tau)^{-1} \cdot \tau_{PK} = \frac{2\eta_{PK} D}{\lambda_{OBK}} \quad (2.54)$$

Tensor tempo de relaxação:

$$\lambda(\tau)^{-1} = \frac{1}{\lambda_{OBK}} \left[\frac{\alpha_K \lambda_{OBK}}{\eta_{PK}} \tau_{PK} + f(\tau)^{-1} \delta + \frac{\lambda_{OBK}}{\eta_{PK}} (f(\tau)^{-1} - 1) \tau_{PK}^{-1} \right] \quad (2.55)$$

Função extra:

$$\frac{1}{\lambda_{OBK}} f(\tau)^{-1} = \frac{2}{\lambda_{SK}} \left(1 - \frac{1}{\Lambda} \right) + \frac{2}{\lambda_{OBK} \Lambda^2} \left(1 - \frac{\alpha_K \lambda_{OBK}^2 I_{\tau \cdot \tau}}{3\eta_{PK}^2} \right) \quad (2.56)$$

Estiramento dorsal e tempo de relaxação do estiramento:

$$\Lambda = \sqrt{1 + \frac{\lambda_{OBK} I_{\tau}}{3\eta_{PK}}}, \quad \lambda_{SK} = \lambda_{OSK} e^{-\frac{2}{q}(\Lambda - 1)} \quad (2.57)$$

Devem-se ainda definir nesses modelos as seguintes grandezas:

$$I_{S \cdot S} = tr(S \cdot S), \quad I_{\tau} = tr(\tau), \quad I_{\tau \cdot \tau} = tr(\tau \cdot \tau) \quad (2.58)$$

Os parâmetros do modelo correspondem aos mesmos do modelo PP mais o parâmetro adicional α_K adicional. Se este parâmetro tem um valor não nulo ele confere o cálculo de uma segunda diferença de tensões normais não nula. Nos trabalhos de

Verbeeten et al. (2002), Bogaerds et al. (2002) e Aboubacar et al. (2005) são encontradas simulações com este modelo. Contudo, como comentado por Clemeur et al. (2003), a introdução deste parâmetro leva a singularidades analíticas. Da análise de soluções semi-analíticas encontrou-se o surgimento de pontos de bifurcação e soluções múltiplas para a viscosidade elongacional em estado estacionário.

O modelo DXPP, apesar de consumir mais recursos computacionais com armazenamento de variáveis, é mais adequado que o SXPP uma vez que este último gera problemas numéricos quando são obtidos valores de $1 + \frac{\lambda_{OB} I_\tau}{3\eta_P} \leq 0$.

Clemeur et al. (2003) propuseram um novo modelo baseado no modelo de PP e nas idéias que levaram Verbeeten et al. (2001) formularem o modelo DXPP. Eles incluíram duas derivadas convectivas, a superior e a inferior. O modelo proposto foi chamado de DCP (Double Convected Pom-Pom) e sua formulação é dada por:

$$\left[\left(1 - \frac{\xi_K}{2} \right) \nabla S_{PK} + \frac{\xi_K}{2} \Delta S_{PK} \right] + (1 - \xi_K) [2D : S_{PK}] S_{PK} + \frac{1}{\lambda_{OBK} \Lambda_{PK}^2} \left[S_{PK} - \frac{\delta}{3} \right] = 0 \quad (2.59)$$

$$\frac{D(\Lambda_{PK})}{Dt} = \Lambda_{PK} [D : S_{PK}] + \frac{1}{\lambda_{SK}} [\Lambda_{PK} - 1], \quad \lambda_{SK} = \lambda_{OSK} e^{-\frac{2}{q}(\Lambda_{PK} - 1)} \quad (2.60)$$

$$\tau_{PK} = \frac{\eta_{PK}}{(1 - \xi_K) \lambda_{OBK}} (3\Lambda_{PK}^2 S_{PK} - \delta) \quad (2.61)$$

onde a derivada convectiva inferior é dada por (Equação 2.62):

$$\Delta S_{PK} = \frac{D}{Dt} S_{PK} + [\nabla U \cdot S_{PK}] + [S_{PK} \cdot \nabla U^T] \quad (2.62)$$

O parâmetro adicional ξ_K controla a razão entre a primeira e a segunda diferença de tensões normais e, por razões físicas, assume tipicamente um valor próximo a 0,2. Maiores detalhes sobre este modelo podem ser encontrados em Clemeur et al. (2003). Simulações com este modelo também podem ser encontradas no trabalho de Clemeur et al. (2004).

Maiores informações sobre equações constitutivas e a teoria por trás desses modelos pode ser encontrada principalmente em [Bird et al. \(1987\)](#), [Larson \(1988\)](#) e [Verbeeten \(2001\)](#).

Capítulo 3

Resolução Numérica de Escoamentos de Fluidos Viscoelásticos

Devido às dificuldades de se resolver escoamentos de fluidos viscoelásticos com altos valores de Deborah, foram surgindo com o passar do tempo inúmeras metodologias que tentam contornar este problema. Será feita uma descrição destas metodologias e serão discutidas algumas delas.

3.1 Métodos Numéricos para Resolver um Problema de CFD

Como visto no capítulo anterior, a representação matemática de fluidos viscoelásticos consiste nas equações de conservação de massa (Equação 2.12), de quantidade de movimento (Equação 2.17) e equações constitutivas. Assim, deve ser resolvido um sistema de equações diferenciais parciais não-lineares, tal que a solução satisfaça as condições iniciais e de contorno impostas.

Como a obtenção de soluções analíticas para estas equações é na maioria dos casos impossível ou impraticável, torna-se necessária a aplicação de um método numérico para a resolução das equações.

O procedimento de resolução numérica deste problema consiste na discretização do domínio e das equações diferenciais parciais. Desse modo as equações diferenciais parciais são aproximadas por um sistema de equações algébrico-diferenciais ordinárias para um conjunto de pontos discretos no espaço (conhecido como método das linhas), compondo o que se chama de malha computacional, ou por um sistema de equações algébricas para um conjunto de pontos discretos no espaço e no tempo.

Os métodos numéricos mais utilizados para solução de escoamentos de fluidos viscoelásticos são diferenças finitas, volumes finitos e elementos finitos. O método das diferenças finitas foi o primeiro a ser empregado, nos trabalhos pioneiros de [Crochet e Pilate \(1976\)](#) e [Perera e Walters \(1977\)](#). Este método continuou a ser explorado nos anos seguintes, mas seu uso foi diminuindo, em favor do uso do método de elementos e volumes finitos.

O método mais utilizado e mais explorado para a simulação de escoamentos de fluidos viscoelásticos é sem dúvidas o método de elementos finitos. Encontram-se muitos trabalhos na literatura usando este método e o desenvolvimento de novas metodologias em elementos finitos é um assunto ainda muito explorado nos dias de hoje.

Após o uso dos métodos de diferenças finitas e elementos finitos terem se consolidado na área de simulação de escoamentos de fluidos viscoelásticos, uma nova alternativa passou a ser usada no início da década de 90. Esta alternativa foi o método dos volumes finitos, já consagrado nesta época dentro da mecânica de fluidos computacional para fluidos newtonianos. Esse novo método foi muito bem aceito uma vez que trazia diversas vantagens se comparado aos demais, como uma maior estabilidade numérica, menor quantidade de memória requerida e menor tempo computacional para soluções com mesma qualidade ([XUE et al., 1995](#); [XUE et al., 1999](#); [LUO, 1996](#)). Uma revisão detalhada de trabalhos envolvendo estes diferentes métodos pode ser encontrada em [Baaijens \(1998\)](#).

3.2 Método de Volumes Finitos

O método de volumes finitos consiste em obter a aproximação numérica da equação diferencial parcial a partir de sua integração no volume de controle elementar que está representado na [Figura 3.1](#). A discretização espacial é feita integrando todos os termos da equação no espaço para cada volume de controle do domínio. O resultado final é a equação discretizada para um grupo de pontos de uma malha.

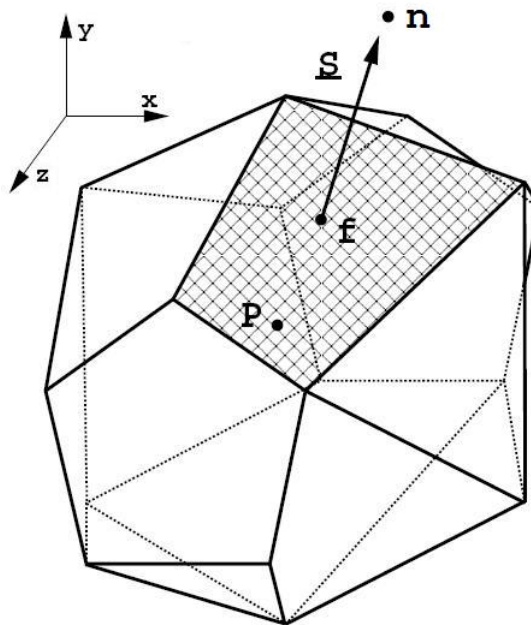


Figura 3.1: Volume de controle (Adaptado de [Jasak \(1996\)](#)).

Como existe vasta literatura sobre o assunto, não serão apresentados neste trabalho detalhes sobre os princípios básicos do método de volumes finitos, sugerindo-se para o leitor interessado neste tópico específico a consulta às obras de [Patankar \(1980\)](#), [Maliska \(2004\)](#), [Ferziger e Peric \(1999\)](#) e [Jasak \(1996\)](#), entre outros. A discussão nesta seção será limitada às opções de metodologias propostas e utilizadas na literatura para tratar passos ou problemas específicos na aplicação do método de volumes finitos.

3.2.1 Tipo de arranjo utilizado para as variáveis

Uma diferença muito comum encontrada em trabalhos que usam o método de volumes finitos é o uso do arranjo desencontrado ou o co-localizado das variáveis. O arranjo co-localizado armazena todas as variáveis do problema no centro dos volumes enquanto o desencontrado, duas ou mais variáveis estão localizadas em posições diferentes do volume. O arranjo desencontrado foi primeiramente o mais usado, contudo nos dias de hoje o arranjo co-localizado é muito mais usado uma vez que diversas vantagens são conseguidas com este arranjo, como por exemplo, menor uso de memória computacional e maior facilidade para se trabalhar com coordenadas generalizadas, indispensáveis em problemas que envolvem geometrias complexas. Torna também mais fácil a aplicação de condições de contorno, uma vez que não sobram "meios volumes" próximos às fronteiras. Além disso, o emprego de técnicas de malhas múltiplas (*multigrid*) só são viáveis usando este tipo de arranjo (PERIC et al., 1988). Deve-se lembrar que o arranjo co-localizado introduz oscilações na pressão e faz-se necessário usar o método de interpolação de momento também conhecido como método de correção de Rhie-Chow (RHIE; CHOW, 1983) para resolver este problema. O método de interpolação de momento consiste em fazer com que a velocidade nas interfaces, necessárias para o cálculo dos fluxos advectivos, dependam das pressões nos volumes vizinhos, "imitando" o arranjo desencontrado. Em problemas envolvendo fluidos viscoelásticos o mesmo problema acontece com a tensão, sendo necessário usar do mesmo artifício que o usado para a pressão para que não surjam oscilações na solução.

3.2.2 Esquemas de Interpolação

Uma questão importante a ser tratada no método de volumes finitos são as funções de interpolações usadas. Maior precisão pode ser conseguida usando esquemas de alta ordem mesmo usando malhas menos refinadas. Contudo, o aumento da ordem de interpolação pode causar problemas de instabilidades numéricas e soluções fisicamente

irreais com o surgimento de oscilações. Os primeiros trabalhos e a grande maioria dos trabalhos encontrados ainda hoje usam esquemas de 1ª ordem UDS (*Upwind Differencing Scheme*), 2ª ordem CDS (*Central Differencing Scheme*) ou uma mistura desses dois métodos gerando um esquema híbrido HDS (*Hybrid Differencing Scheme - UDS + CDS*). Esquemas de alta ordem ou também os HRS (*High Resolution Schemes*), como os usados no trabalho de [Muniz et al. \(2008\)](#), o MINMOD [Harten \(1983\)](#) e o esquema CUBISTA (*Convergent and Universally Bounded Interpolation Scheme for the Treatment of Advection*) introduzido por [Alves et al. \(2003b\)](#) também são alternativas. Contudo deve-se ter muito cuidado na hora da escolha de um esquema de alta ordem, pois podem ocorrer problemas de convergência e oscilações na solução. Apesar disso, sabe-se que as oscilações podem ser eliminadas com o uso de combinações convexas de diferentes esquemas de alta ordem, gerando os esquemas WENO (*Weighted Essentially Non-Oscillatory*) ([LIU et al., 1994](#)).

3.2.3 Solução das Equações Discretizadas

Encontram-se na literatura trabalhos que usam a formulação de volumes finitos e resolvem o sistema de equações totalmente acoplado ([MUNIZ, 2003](#)), ou seja, o sistema de equações é resolvido simultaneamente, e trabalhos onde a resolução é feita de forma segregada, onde nesse caso usa-se um método de correção de pressão, tal como o SIMPLE (*Semi-Implicit Method for Pressure Linked Equations*) ([PATANKAR; SPALDING, 1972](#)) e o PISO (*Pressure Implicit Splitting of Operators*) ([ISSA, 1986](#)) entre outros.

Em qualquer destas metodologias, acoplada ou desacoplada, a resolução de sistemas de equações não-lineares geralmente é feita de duas maneiras. Uma delas é usar o método das substituições sucessivas com ou sem relaxação. A outra é usar o método de Newton para resolver o sistema não-linear. Em cada iteração deste método se torna necessário também resolver um sistema linear de equações algébricas.

Para resolver o sistema linear podem-se usar métodos diretos ou iterativos. Os métodos diretos fazem a inversão completa da matriz. Como exemplos podem-se citar a eliminação gaussiana e fatorizações (LU, Cholesky). Esses métodos possuem como característica o elevado custo computacional. Os métodos iterativos são baseados na transformação do sistema linear em um procedimento iterativo em que, a partir de uma estimativa inicial, pode conduzir à solução desejada. Esses métodos geram menor custo computacional, principalmente referente à memória necessária, e por isso são muito mais usados em CFD. Exemplos são os métodos de Gauss-Seidel, métodos de minimização, como o método dos resíduos generalizado (GMRES), método de CG (*Conjugate Gradient*) e seus derivados e métodos de malhas múltiplas como o GAMG (*Generalised geometric-algebraic multi-grid*), entre outros.

Outro ponto a se considerar para os métodos iterativos é o uso de alguns pré-condicionadores na hora da resolução do sistema de equações, pois, dependendo do procedimento iterativo aplicado, a matriz de iteração pode não ser diagonalmente dominante, que é a condição necessária para a convergência do método. Alguns pré-condicionadores como o DIC (*Diagonal incomplete-Cholesky*), DILU (*Diagonal incomplete-LU*) e Bi-CGSTAB (*Bi-Conjugate Gradient Stabilized*) são bem conhecidos na literatura (AJIZ; JENNINGS, 1984; LEE et al., 2003; JACOBS, 1980; VORST, 1992).

Deve-se ainda mencionar que o sistema de equações discretizado é caracterizado por apresentar alto nível de esparsidade. Assim o uso de algoritmos que levam em conta o fato dessas matrizes serem esparsas é indispensável para diminuir o custo computacional.

3.3 Metodologia para Resolver escoamentos com Elevados valores de We

Inúmeros trabalhos usando os modelos diferenciais não-lineares apresentados no [Capítulo 2](#) podem ser encontrados na literatura. Uma dificuldade presente em todos estes trabalhos, independente do método de discretização (elementos finitos ou volumes finitos), esquemas de interpolação, método iterativo de solução das equações ou equações constitutivas utilizadas, é o conhecido HWNP (*High Weissenberg Number Problem*). Este problema consiste na dificuldade de se obter soluções em números de *Weissenberg* (We) ou *Deborah* (De) elevados.

Para contornar esse problema surgiram várias metodologias numéricas com o passar do tempo. Essas metodologias caracterizam-se por usar algum artifício para estabilizar os métodos numéricos. Algumas dessas metodologias estabilizam o método numérico através do aumento do caráter elíptico da equação de movimento pela introdução de um operador elíptico. Entre essas metodologia pode-se citar a formulação viscosa (*Viscous Formulation*), a EVSS (*Elastic Viscous Split-Stress*) desenvolvida primeiramente por [Perera e Walters \(1977\)](#) para o método das diferenças finitas e posteriormente para o método de elementos finitos por [Rajagopalan et al. \(1990\)](#), a DEVSS (*Discrete Elastic Viscous Split-Stress*) de [Guénette e Fortin \(1995\)](#), a AVSS (*Adaptive Viscosity Stress Splitting Scheme*) de [Sun et al. \(1996\)](#) e a EEME (*Explicitly Elliptic Momentum Equation*) de [King et al. \(1988\)](#), entre muitas outras derivações destes métodos ([MATALLAH et al., 1998](#)). Nesse trabalho foram testadas algumas dessas metodologias e a seguir é feita uma descrição mais detalhada sobre elas.

3.3.1 Formulação Viscosa

A formulação viscosa consiste em dividir a tensão em uma contribuição viscosa e uma polimérica, onde a tensão viscosa é somente função da velocidade e a tensão polimérica

provém da equação constitutiva. Assim para a formulação viscosa temos a seguinte equação de conservação de quantidade de movimento:

$$\frac{\partial(\rho U)}{\partial t} + \nabla \cdot (\rho U U) - \eta_s \nabla \cdot (\nabla U) = -\nabla p + \nabla \cdot \tau_P \quad (3.1)$$

Essa formulação é a opção natural, pois a contribuição viscosa, do solvente ou também newtoniana da tensão entra diretamente na equação de conservação de quantidade de movimento e pode ser encarada como a contribuição dos componentes que possuem um tempo de relaxação desprezível perante os outros da mistura.

Sabe-se que a inclusão de um termo viscoso difusivo na [Equação 3.1](#) é essencial para a estabilidade numérica do problema e é perfeitamente aceitável do ponto de vista teórico. Geralmente a viscosidade do solvente η_s é menor que a viscosidade polimérica η_p . Essas medidas são obtidas a partir de dados experimentais, como em [Quinzani et al. \(1994\)](#), ou também, no caso de estudos numéricos para testar novos esquemas de interpolação ou novas metodologias numéricas, usa-se a razão de $\eta_s/\eta_p \sim 1/9$, pois esse valor sugere que a viscosidade do solvente é pequena em relação à viscosidade polimérica e, por outro lado, é suficiente para dar estabilidade ao problema ([ALVES et al., 2003a](#)).

3.3.2 Formulação EVSS

A formulação EVSS considera o tensor total das tensões poliméricas como sendo uma parcela viscosa e outra elástica ([Equação 3.2](#)):

$$\tau_P = \tau_V + \tau_E \quad (3.2)$$

onde,

$$\tau_V = 2\eta_a D \quad (3.3)$$

Substituindo τ_P na equação de quantidade de movimento resulta em:

$$\frac{\partial(\rho U)}{\partial t} + \nabla \cdot (\rho U U) - (\eta_a + \eta_s) \nabla \cdot (\nabla U) = -\nabla p + \nabla \cdot \tau_E \quad (3.4)$$

onde o operador elíptico é proporcional à constante η_a . No trabalho original de [Rajagopalan et al. \(1990\)](#) seu valor foi considerado como sendo $\eta_a = 1$, mas a forma mais usada é dada por $\eta_a = \eta_P$. Esta alteração é responsável por aumentar o caráter elíptico da equação da quantidade de movimento. Em termos matemáticos a inclusão desse operador não introduz mudança alguma uma vez que somente foi feita uma mudança de variável. Sendo assim, a mudança de variável deverá ser feita também na equação constitutiva.

Como exemplo, será ilustrada a mudança para a [Equação 2.40](#), equação de PTT linear simplificada (LPTTS). Considerando $\eta_a = \eta_P$ e introduzindo a mudança de variável tem-se:

$$\left(1 + \frac{\varepsilon_K \lambda_K}{\eta_{PK}} \text{tr}(\tau_{VK} + \tau_{EK})\right) (\tau_{VK} + \tau_{EK}) + \lambda_K (\overset{\nabla}{\tau}_{VK} + \overset{\nabla}{\tau}_{EK}) = 2\eta_{PK} D \quad (3.5)$$

Definindo:

$$g = \left(1 + \frac{\varepsilon_K \lambda_K}{\eta_{PK}} \text{tr}(\tau_{VK} + \tau_{EK})\right) \quad (3.6)$$

Assim,

$$g(\tau_{VK} + \tau_{EK}) + \lambda_K (\overset{\nabla}{\tau}_{VK} + \overset{\nabla}{\tau}_{EK}) = 2\eta_{PK} D \quad (3.7)$$

$$g\tau_{EK} + \lambda_K \overset{\nabla}{\tau}_{EK} = 2\eta_{PK} D - g\tau_{VK} - \lambda_K \overset{\nabla}{\tau}_{VK} \quad (3.8)$$

Substituindo a [Equação 3.3](#) na eq. [Equação 3.8](#) obtêm-se:

$$g\tau_{EK} + \lambda_K \overset{\nabla}{\tau}_{EK} = 2\eta_{PK} D - 2\eta_{PK} gD - 2\eta_{PK} \lambda_K \overset{\nabla}{D} \quad (3.9)$$

$$g\tau_{EK} + \lambda_K \overset{\nabla}{\tau}_{EK} = -2\eta_{PK} \lambda_K \overset{\nabla}{D} - (g - 1)2\eta_{PK} D \quad (3.10)$$

A [Equação 3.10](#) será a nova forma da equação constitutiva a ser resolvida. Percebe-se que esta equação inclui a derivada convectiva superior do tensor taxa de deformação e desse modo deverá ser resolvida também para esta variável.

Usando a formulação EVSS consegue-se soluções estáveis para números de We relativamente altos (MATALLAH et al., 1998). A principal desvantagem deste método é que a mudança de variável pode ser complicada em equações complexas.

3.3.3 Formulação DEVSS

Nesse método é introduzido um termo difusivo adicional em cada lado da equação de conservação de quantidade de movimento. A equação final assume a forma:

$$\frac{\partial(\rho U)}{\partial t} + \nabla \cdot (\rho U U) - (\eta_S + \kappa) \nabla \cdot (\nabla U) = -\nabla p + \nabla \cdot \tau_P - \kappa \nabla \cdot (\nabla U) \quad (3.11)$$

onde κ é um número positivo. O valor de κ está relacionado com os parâmetros do modelo constitutivo mas $\kappa = \eta_P$ tem se mostrado uma boa escolha (GUÉNETTE; FORTIN, 1995).

Percebe-se que a **Equação 3.11** é idêntica a equação de movimento original, a não ser quando está na forma discreta, pois, como é prática em CFD, os termos que estão do lado esquerdo da equação são resolvidos implicitamente e os do lado direito da equação são resolvidos explicitamente. Esse método é algumas vezes encontrado na literatura como BSD (*Both Sides Diffusion*).

Essa metodologia se mostrou muito atraente e vem sendo muito usada uma vez que proporciona vantagens similares a metodologia EVSS e não requer mudanças de variáveis nos modelos reológicos, sendo assim genérica e facilmente aplicável a todos os modelos constitutivos.

3.3.4 Formulação AVSS, DAVSS e derivações destas Metodologias

Essa metodologia é uma derivação das metodologias EVSS e DEVSS, porém possui uma característica distinta, pois considera a viscosidade variável. A metodologia AVSS

foi proposta por [Sun et al. \(1996\)](#) e a DAVSS mais tarde também por eles [Sun et al. \(1999\)](#). A finalidade dessa metodologia é aumentar a estabilidade do sistema fazendo com que a viscosidade seja um parâmetro dependente do escoamento. A [Equação 3.12](#) mostra um exemplo desse tipo de metodologia conhecida na literatura como DAVSS- ω ([DOU; PHAN-THIEN, 1999](#)):

$$\frac{\partial(\rho U)}{\partial t} + \nabla \cdot (\rho U U) - (\varphi + \eta_s) \nabla^2 U = -\nabla p + \nabla \cdot \tau_p + \varphi \nabla \times \omega - \varphi \nabla(\nabla \cdot U) \quad (3.12)$$

Onde se tem que:

$$\nabla^2 U = -\nabla \times \omega + \nabla(\nabla \cdot U) \quad (3.13)$$

Sendo o tensor vorticidade, ω , dado por:

$$\omega = \nabla \times U \quad (3.14)$$

A variável φ é definida como:

$$\varphi = \frac{\sqrt{a + b/2 \sum \tau_{ij}^2}}{\sqrt{a + b/2 \sum D_{ij}^2}} \quad (3.15)$$

onde os valores dos parâmetros a e b estão nos intervalos $[0,1 \ 1]$ e $[0, 1]$, respectivamente, como sugerido por [Dou e Phan-Thien \(1999\)](#). No caso particular em que $\varphi = \eta_p$ é obtida a metodologia conhecida como DEVSS- ω .

Capítulo 4

Apresentação do Pacote de CFD *OpenFOAM*

*Este capítulo tem como objetivos detalhar o funcionamento do software **OpenFOAM** como ferramenta de CFD e também discutir aspectos relativos a implementação de novos solvers.*

4.1 O *OpenFOAM*

O *OpenFOAM* surgiu em 1993, quando Henry Weller e Hrvoje Jasak uniram esforços para a criação do *FOAM* (*Field Operation and Manipulation*). A finalidade era ter uma ferramenta para se fazer operações com campos tensoriais. No ano de 2004 o *FOAM* teve seu código liberado, se tornou domínio público através da licença GLP (*Gnu Public License*) e começou a se chamar *OpenFOAM* (*Open Field Operation and Manipulation*). A partir deste momento houve um enorme crescimento de usuários que, além de poder usar os muitos *solvers* padrões que o pacote já possuía para o caso dos problemas mais gerais envolvendo fluidos newtonianos (escoamento compressível e incompressível, escoamento laminar e turbulento, escoamentos multifásicos, etc.), podiam também construir *solvers* específicos para os seus problemas de interesse.

O *OpenFOAM* é hoje um conjunto eficiente e flexível de módulos escritos em

linguagem C++ que pode ser usado para construir: (i) *solvers* para resolver problemas complexos de engenharia que envolvam operações e resoluções de campos tensoriais, (ii) utilitários para pré e pós-processamento de dados, (iii) bibliotecas para serem usadas pelos *solvers* e utilitários, tal como bibliotecas de modelos físicos.

Entre as principais vantagens que levaram a se escolher o *OpenFOAM* neste trabalho e que vem dando ao *software* cada vez mais adeptos podem-se citar:

- código aberto, que além de não envolver investimentos com a compra de licenças de uso também permite ao usuário verificar, manipular e incrementar o código.
- escrito em linguagem C++. Como esta linguagem é orientada a objetos torna-se muito mais fácil a criação de novos códigos, uma vez que propriedades como abstração e encapsulamento de dados, herança, polimorfismo, etc., facilitam muito a expansão do *software*.
- criador de malhas e visualizador de resultados incorporado ao *software*.
- possibilita fazer grandes simulações usando processamento paralelo.
- utilização de malhas móveis e não-ortogonais.
- possibilidade de importação e exportação de dados. O *software* possui ferramentas que possibilitam a importação de malhas estruturadas ou não-estruturadas de diferentes *softwares* gratuitos ou comerciais e também permite a exportação dos dados simulados para visualização em outros *softwares*.
- simplicidade de uso.
- aplicação em uma ampla faixa de problemas de engenharia.
- possui bons solvers para resolução de sistemas lineares de equações e dispõem de uma grande variedade de esquemas de interpolação.

4.2 O *OpenFOAM* a nível de Usuário

Para os usuários comuns do *OpenFOAM*, ou seja, aqueles que usam os *solvers* que já estão implementados no *software* a simulação pode ser dividida em nas seguintes etapas:

1. gerar a estrutura de diretórios necessária para efetuar a simulação.
2. pré-processamento: geração da geometria e da malha e definições de parâmetros da simulação.
3. solução numérica do problema: nesta etapa o modelo utilizado é resolvido de acordo com as condições impostas.
4. pós-processamento: visualização dos resultados.

4.2.1 Estrutura Necessária para Efetuar uma Simulação no *OpenFOAM*

Todos os *solvers* do *OpenFOAM* usam um conjunto de arquivos que armazenam as informações necessárias para se resolver um caso. Cada caso deve seguir uma estrutura de diretórios que contém os arquivos que armazenam as informações necessárias ao mesmo. Estes arquivos possuem as informações como a descrição da geometria, detalhes da malha, condições de contorno, parâmetros para os métodos numéricos e as propriedades físicas do problema. A estrutura de diretórios pode ser vista na [Figura 4.1](#), onde é representado um caso genérico (definido como **<Nome do Caso>**).

O diretório principal **<Nome do Caso>** é a "raiz" do caso e dentro deste estão incluídos os outros diretórios e arquivos de configuração. Uma breve descrição sobre o conteúdo destes diretórios é colocada na seqüência.

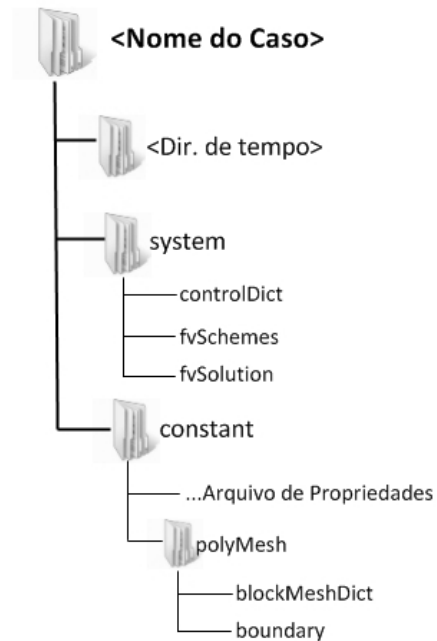


Figura 4.1: Estrutura de diretórios e arquivos necessários para uma simulação com o *OpenFOAM*.

- `<Dir.de Tempo>`: contém os arquivos individuais de dados para os campos das variáveis tratadas no caso (por exemplo, campo de velocidade, pressão, tensão, etc.). O nome associado ao diretório `<Dir.de Tempo>` refere-se ao instante simulado no qual os dados são escritos.
- `<system>`: os arquivos contidos neste diretório estão associados ao procedimento de solução do caso. Devem existir pelo menos 3 arquivos neste diretório: o `controlDict`, o `fvSolution` e o `fvSchemes`, discutidos a seguir.
- `<constant>`: deve conter os arquivos de propriedades físicas pertinentes ao caso, por exemplo, `transportProperties`. A descrição completa da geometria e da malha deve ser incluída no diretório `polyMesh`, nos arquivos `blockMeshDict` e `boundary`.

O usuário pode alterar os dados diretamente nos arquivos de configuração

usando um editor de texto ou usar a ferramenta gráfica *FoamX*. O *FoamX* acessa os arquivos de configuração alterando-os e organizando as informações pertinentes ao caso.

4.2.2 Pré-Processamento

Uma etapa muito importante em CFD é a da criação da geometria e principalmente da malha. A malha surge da discretização do domínio em uma quantidade definida de subdomínios. As equações diferenciais parciais do modelo também são discretizadas e após sua resolução será obtida a solução para os pontos discretos do domínio, ou seja, para cada subdomínio da malha. Uma malha mais fina, ou seja, com maior quantidade de subdomínios, produz resultados para uma maior quantidade de pontos do domínio e também fornece maior precisão. Contudo, quanto maior a quantidade de células maior será o problema a ser resolvido e maior o custo computacional. Assim deve-se optar por malhas que forneçam resultados com precisão adequada e que exijam um tempo de simulação aceitável.

O *OpenFOAM* não possui um editor CAD para construção da geometria do problema. As informações sobre a geometria são armazenadas e podem ser editadas no arquivo `blockMeshDict`. O comando `blockMesh` gera arquivos a partir do arquivo `blockMeshDict`, criando e estruturando os dados da malha em pontos, faces e células (arquivos `points`, `faces` e `cells`, respectivamente).

Para os casos em que se faz necessário o uso de malhas não-estruturadas pode-se fazer a importação de geometrias e malhas geradas em outros *softwares*. Assim, pode-se optar por *softwares* comerciais como *ICEM* e *GAMBIT* ou livres como *NETGEN*, *TETGEN*, *GMSH* e *SALOME*. O *OpenFOAM* aceita diferentes tipos de geometrias para as células, como hexaédrica, tetraédrica, prismática, piramidal, etc.

Além da definição da geometria e malha deve-se definir nos arquivos de con-

trole as condições da simulação no diretório `<system>` e as propriedades físicas e modelos adicionais do problema no diretório `<constant>`. Os arquivos de controle são:

- `controlDict`: controla o tempo de simulação, passo de tempo, intervalo de escrita de dados, etc.
- `fvSchemes`: especifica os métodos para discretização dos termos derivativos das equações. O método de discretização padrão adotado pelo *OpenFOAM* é a integração de Gauss para volumes finitos.
- `fvSolution`: é definido os métodos de solução do sistema de equações lineares e suas respectivas tolerâncias, assim como alguns parâmetros para o algoritmo de solução do campo de escoamento (correção pressão-velocidade e ortogonalidade da malha).

Além desses três arquivos de controle de simulação, outros podem ser colocados no diretório `<system>`. Um deles é o arquivo usado para simulações em paralelo (`decomposeParDict`). O *OpenFOAM* usa a biblioteca de domínio público *MPI* (*Message Passage Interface*) para a comunicação entre os computadores e a decomposição do domínio pode ser feita por quatro métodos diferentes, onde o *METIS* se destaca devido à grande eficiência no particionamento da malha. Outro ponto importante é a especificação das condições iniciais e de contorno, que será feito em arquivos com o mesmo nome da variável (por exemplo, `U` no caso da velocidade) dentro do diretório de tempo (para o tempo inicial zero tem-se o diretório `0`). Para condições iniciais o *OpenFOAM* permite a entrada de um campo uniforme e não-uniforme. O *OpenFOAM* apresenta implementadas as principais condições de contorno, como entrada de massa (`inlet`), saída de massa (`outlet`), parede fixa ou móvel (`wall`), condição atmosférica (`atmospheric`), simetria (`simetry`), entre outras. Porém nada impede ao usuário para criar sua própria condição de contorno, caso seja necessário. As condições de contorno também podem ser impostas como uniformes ou não-uniformes. Por fim,

devem-se definir as propriedades físicas e os modelos adicionais usados na simulação em arquivos específicos para cada caso (ex. `viscoelasticProperties`).

4.2.3 Etapa de Resolução Numérica

Nesta etapa é feita a resolução numérica das equações do modelo. O *OpenFOAM* utiliza arquivos executáveis chamados de *solvers* para realizar tal propósito. Os *solvers* contêm os modelos para resolver o caso pretendido e as informações de todas as rotinas de cálculo a serem executadas. Assim, quando eles são invocados acontece a leitura de todos os parâmetros da simulação definidos nos arquivos do caso e especificados na etapa de pré-processamento. Com estas informações o *solver* pode partir para a resolução numérica do problema.

O *OpenFOAM* apresenta uma grande diversidade de *solvers* que já vem com o pacote original. Se o usuário pretender usar um modelo que não se encontra na versão original é permitido a ele criar um novo *solver* para seu caso específico.

4.2.4 Pós-Processamento

O *OpenFOAM* possui uma ferramenta para o pós-processamento dos resultados que é denominada `paraFoam`, adaptada do *software ParaView* (PARAVIEW, 2008), para visualização científica e de código aberto.

As ferramentas básicas para visualização de resultados CFD estão incluídas no `paraFoam`, como a criação de gráficos de contorno, vetores e linhas de fluxo. Ainda é possível criar animações para analisar o transiente dos resultados. Outros *softwares* de visualização com recursos mais avançados também podem ser usados, pois é possível converter os resultados fornecidos pelo *OpenFOAM* para formatos lidos por *softwares* como *FLUENT*, *Fieldview*, *Ensign* e *Tecplot*. Existe ainda uma ferramenta de

conversão dos resultados do *OpenFOAM* para o formato VTK possibilitando a leitura dos dados em qualquer visualizador que use *VTK*.

4.3 O *OpenFOAM* a nível de Usuário Desenvolvedor

Os desenvolvedores necessitam ter conhecimento em linguagem C++ e um conhecimento básico do código do *OpenFOAM*, como os operadores, classes e funções importantes. Um bom conhecimento em CFD também é indispensável. Para linguagem C++ uma boa referencia é o trabalho de Yang (2001). As fontes sobre programação no *OpenFOAM* são seus manuais (*User's Guide* e *Programmer's Guide*) e fóruns de internet (FORUM, 2008). Sobre CFD encontra-se vasta literatura, contudo uma boa fonte é o trabalho de Jasak (1996) que apresenta detalhadamente vários aspectos sobre a formulação numérica, incluindo a metodologia de discretização, condições de contorno, etc., e a teoria dos algoritmos implementados no *OpenFOAM*, como o acoplamento pressão-velocidade, correção dos fluxos em malhas não estruturadas, etc..

4.3.1 Orientação a Objetos e C++

Para tentar solucionar o problema do baixo reaproveitamento de código, tomou corpo a idéia da Programação Orientada a Objeto (POO). A POO não é nova, sua formulação inicial data de 1960, porém, somente a partir dos anos 90 é que passou a ter seu uso difundido. Hoje, todas as grandes empresas de desenvolvimento de programas têm desenvolvido os seus *softwares* usando a programação orientada a objeto.

A programação orientada a objeto não é somente uma nova forma de programar é uma nova forma de pensar um problema utilizando conceitos do mundo real e não conceitos computacionais. Os conceitos de objetos devem acompanhar todo o ciclo de

desenvolvimento de um *software*.

Descreve-se a seguir os conceitos básicos da análise orientada a objeto, isto é, a abstração, o objeto, as classes, a herança e o polimorfismo.

- **Abstração:** Para a análise orientada a objeto, a abstração é o processo de identificação dos objetos e seus relacionamentos.
- **Objetos:** São coisas do mundo real ou imaginário que podemos de alguma forma identificar, como uma pedra, uma caneta, um copo. O objeto interage com o meio e, em função de excitações que sofre, realiza determinadas ações que alteram o seu estado.
- **Classe:** A classe contém toda a descrição da forma do objeto, é um molde para a criação do objeto, é uma matriz geradora de objetos.
- **Encapsulamento:** Para a análise orientada a objeto, encapsulamento é o ato de esconder do usuário informações que não são de seu interesse. O objeto atua como uma caixa preta, que realiza determinada operação, mas o usuário não sabe, e não precisa saber, exatamente como.
- **Herança:** Na análise orientada a objeto, herança é o mecanismo em que uma classe filha (subclasse) compartilha automaticamente todos os métodos e atributos de sua classe pai (superclasse). A herança permite implementar classes descendentes implementando os métodos e atributos que se diferenciam da classe pai.
- **Polimorfismo:** O conceito de polimorfismo é fundamental para a análise orientada a objeto; sua aplicação se fundamenta no uso de uma superclasse, através do qual vamos desenvolver nossa hierarquia de classes.

A linguagem de programação C++ é uma linguagem que contempla todas essas características de orientação a objetos. Seu surgimento data o ano de 1980, quando

Bjarne Stroustrup criou essa linguagem como um superconjunto da linguagem C sendo inicialmente chamado C com classes (STROUSTRUP, 1999). Hoje em dia, grande parte das empresas de desenvolvimento de softwares usam C++.

4.3.2 Linguagem do *OpenFOAM*

A técnica de orientação a objetos usada pelo *OpenFOAM* permitiu criar tipos de dados muito próximos aos usados na mecânica do contínuo. A técnica de sobrecarregamento de operadores permitiu que a simbologia matemática usual fosse aplicada para operações básicas. Assim as equações da mecânica do contínuo apresentadas como equações diferenciais parciais e os conceitos de escalares, vetores, tensores e seus respectivos campos, assim como a álgebra tensorial e sistemas de unidades podem ser invocados no *OpenFOAM* usando uma sintaxe parecida com a notação matemática usual. Isto, além de facilitar a implementação de novos *solvers*, também torna muito mais ágil o desenvolvimento.

As classes implementadas no *OpenFOAM* declaram tipos e operações que fazem parte da linguagem matemática utilizada na engenharia e no meio científico. O campo de velocidades pode ser representado no código de programação pelo símbolo \mathbf{U} e a magnitude do campo de velocidade pode ser obtido com a operação $\text{mag}(\mathbf{U})$. A velocidade é um campo vetorial e, portanto, deve existir um código com orientação a objetos para definir uma classe do tipo `vectorField`. Então, o campo de velocidade pode ser visto como um objeto da classe `vectorField`.

A estrutura das classes restringe o desenvolvimento do código dentro das próprias classes, tornando o código mais fácil de manipular. Novas classes podem herdar propriedades de outras classes, por exemplo, um `vectorField` pode ser derivado de uma classe `vector` e uma classe `Field`. O C++ fornece um mecanismo chamado de classes `template`, de forma que a classe `Field<Type>` pode representar um campo

de qualquer `<Type>`, como `scalar`, `vector` e `tensor`. As características gerais da classe `template` são passadas para qualquer classe criada a partir deste `template`. Os *templates* e a herança reduzem a duplicação de código e criam hierarquias de classe que impõe uma estrutura ao código.

O *OpenFOAM* usa o método dos volumes finitos para discretizar os campos geométricos e as bibliotecas `fvm.H` e `fvc.H` são responsáveis pelo processo de aproximação dos termos derivativos das variáveis tensoriais calculadas. Cada termo nas equações diferenciais parciais é representado individualmente no *OpenFOAM* usando as classes *finiteVolumeMethod* e *finiteVolumeCalculus*, abreviado por `fvm` e `fvc`, respectivamente. Tanto `fvm` como `fvc` contêm funções estáticas que representam os operadores diferenciais, como gradiente, divergente e laplaciano. Apesar destas bibliotecas possuírem o mesmo propósito, suas aplicações são diferentes.

A biblioteca `fvm.H` reúne funções para realizar operações implícitas de discretização pelo método dos volumes finitos e os resultados são armazenados em uma matriz definida pela classe `fvMatrix<Type>`. Em outras palavras, a classe `fvm` discretiza os termos que irão ser resolvidos na simulação e constrói um sistema de equações lineares. A biblioteca `fvm.H` ainda é capaz de montar a matriz utilizando termos fontes com discretização implícita ou explícita.

Já a biblioteca `fvc.H` agrupa funções para calcular operações explícitas de discretização. Os termos discretizados por essa classe não são armazenados em uma matriz, como no caso dos discretizados pela biblioteca `fvm.H`. Assim, as operações realizadas com a classe `fvc` retornam explicitamente um campo geométrico (classe `geometricField<Type>`). Por exemplo, essa biblioteca é particularmente útil na solução do gradiente da pressão pelo fato do *OpenFOAM* não incluir a pressão na matriz `fvMatrix<Type>`, já que utiliza um método segregado de solução para o acoplamento pressão-velocidade. Além das operações de discretização, essa biblioteca possui classes para integração de um campo tensorial sobre um volume ou superfície.

4.3.3 Definição e nomenclatura dos operadores diferenciais no *OpenFOAM*

Os operadores diferenciais no *OpenFOAM* são definidos com base nas duas classes mencionadas na seção anterior, *fvM* e *fvC*. Assim, o *OpenFOAM* possui uma linguagem que imita a linguagem matemática. A [Tabela 4.1](#) mostra alguns operadores importantes em CFD e suas palavras chaves no *OpenFOAM*.

Tabela 4.1: Principais palavras chaves usadas no arquivo *fvSchemes*.

Palavra Chave	Categoria do termo Matemático
snGradSchemes	Componente do gradiente normal a face da célula
gradSchemes	Gradiente ∇
divSchemes	Divergente $\nabla \cdot$
laplacianSchemes	Laplaciano ∇^2
timeSchemes	Primeira e segunda derivada temporal $\partial/\partial t, \partial^2/\partial t^2$

Na [Figura 4.2](#) é mostrada uma célula típica resultante da discretização do domínio. O ponto P corresponde à célula de interesse, o ponto N é uma célula vizinha, f é a face de comunicação entre as duas células, d é o vetor de P até N e S_f é um vetor que contém a área da face.

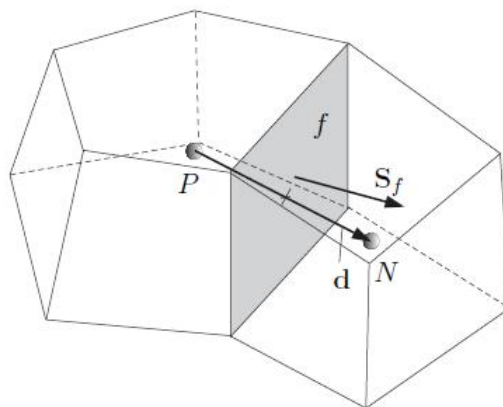


Figura 4.2: Parâmetros em uma discretização por volumes finitos.

4.3.3.1 Avaliação do Operador Gradiente

O termo gradiente é um termo explícito e pode ser avaliado de diversas formas:

- **Integração Gaussiana:** é feita usando a função `fvc::gGrad`. A discretização é obtida usando o método padrão de integração Gaussiana no volume de controle.

$$\int_V \nabla \phi dV = \int_S dS \phi = \sum_f S_f \phi_f \quad (4.1)$$

onde ϕ é uma variável arbitrária e o sub-índice f corresponde a face da célula.

- **Método dos mínimos quadrados:** é baseado nas seguintes idéias:
 1. O valor no ponto P pode ser extrapolado para seu vizinho N usando o gradiente no ponto P .
 2. O valor extrapolado em N pode ser comparado com o valor atual em N , a diferença é o erro.
 3. Se for agora minimizada a soma dos erros quadrados de todos os pontos vizinhos de P com respeito ao gradiente, esse terá uma boa aproximação.

A aproximação por mínimos quadrados é feita usando uma função conhecida como `fvc::lsGrad`. A discretização é feita calculando-se o valor do tensor G em todos os pontos P pela soma dos vizinhos N :

$$G = \sum_N w_N^2 dd \quad (4.2)$$

onde a função peso $w_N = 1/|d|$. O gradiente pode ser avaliado como:

$$(\nabla \phi)_P = \sum_N w_N^2 G^{-1} \cdot d(\phi_N - \phi_P) \quad (4.3)$$

- **Método do gradiente normal a superfície:** tem-se que $n_f \cdot (\nabla \phi_f)$ pode ser avaliado nas faces das células usando o seguinte esquema:

$$(\nabla \phi)_f = \frac{(\phi_N - \phi_P)}{|d|} \quad (4.4)$$

Neste caso, o operador gradiente é chamado pela função `fv::snGrad`. Este termo é análogo ao da discretização do Laplaciano. Um termo de correção para malhas não-ortogonais pode ser introduzido. Para usar o termo de correção usa-se a função `fv::snGradCorrection`.

Na [Tabela 4.2](#) apresentam-se mais informações referentes à avaliação do operador gradiente.

Tabela 4.2: Esquemas de discretização possíveis em `gradSchemes`.

Esquemas de discretização	Descrição
Gauss <interpolationScheme>	Segunda ordem, integração Gaussiana
leastSquares	Segunda ordem, mínimos quadrados
fourth	Quarta ordem, mínimos quadrados
Limited <gradScheme>	Versão com um limitador para alguns dos casos acima

4.3.3.2 Avaliação do Operador Divergente

O operador divergente é um termo explícito. Sua integração no volume de controle e linearização é feita conforme a [Equação 4.5](#):

$$\int_V \nabla \cdot \phi dV = \int_S dS \cdot \phi = \sum_f S_f \cdot \phi_f \quad (4.5)$$

Já o divergente convectivo é resolvido implicitamente, assim é integrado no volume de controle e linearizado como mostrado na [Equação 4.6](#):

$$\int_V \nabla \cdot (\rho U \phi) dV = \int_S dS \cdot (\rho U \phi) = \sum_f S_f \cdot (\rho U)_f \phi_f = \sum_f F \phi_f \quad (4.6)$$

O valor de ϕ_f pode ser calculado usando uma variedade de esquemas de interpolação, os quais são apresentados na seção seguinte.

4.3.3.3 Avaliação do Operador Laplaciano

O operador Laplaciano é integrado no volume de controle e linearizado como mostrado a seguir:

$$\int_V \nabla \cdot (\Gamma \nabla \phi) dV = \int_S dS \cdot (\Gamma \nabla \phi) = \sum_f \Gamma_f S_f \cdot (\nabla \phi)_f \quad (4.7)$$

A discretização do gradiente na face é implícita quando o comprimento d entre o centro da célula de interesse P e o centro da célula vizinha N é ortogonal ao plano da face, ou seja, paralelo a S_f (**Equação 4.8**):

$$S_f \cdot (\nabla \phi)_f = |S_f| \frac{\phi_N - \phi_P}{|d|} \quad (4.8)$$

No caso de malhas não-ortogonais, um termo explícito adicional é introduzido, o qual é calculado através de interpolação dos valores dos gradientes no centro das células, usando diferenças centrais. No trabalho de [Jasak \(1996\)](#) está descrito mais detalhadamente como isto é feito no OpenFOAM.

4.3.3.4 Esquemas de Interpolação para Avaliação Temporal

Para a derivada temporal de primeira ordem $\partial/\partial t$ é feita a integração no volume de controle da seguinte forma (**Equação 4.9**):

$$\frac{\partial}{\partial t} \int_V \rho \phi dV \quad (4.9)$$

O termo é discretizado no tempo usando:

- **Valores novos:** $\phi^n \equiv \phi(t + \Delta t)$ para o passo de tempo que está sendo resolvido.
- **Valores do passo anterior:** $\phi^o \equiv \phi(t)$ valor do passo de tempo anterior.
- **Valores de dois passos anteriores:** $\phi^{oo} \equiv \phi(t - \Delta t)$ valor do passo de tempo anterior ao último passo de tempo.

Um dos dois esquemas de discretização abaixo, além do tradicional método de Crank-Nicholson, pode ser usado na hora da resolução do transiente de um problema:

Esquema Euler implícito é de primeira ordem ([Equação 4.10](#)):

$$\frac{\partial}{\partial t} \int_V \rho \phi dV = \frac{(\rho_P \phi_P V)^n - (\rho_P \phi_P V)^o}{\Delta t} \quad (4.10)$$

Esquema de diferenças Backward é de segunda ordem ([Equação 4.11](#)). Este esquema usa dois passos de tempo anteriores ao passo atual para o cálculo e necessita guardar uma maior quantidade de dados, se comparado com o método de Euler implícito. Na [Tabela 4.3](#) são apresentadas informações adicionais sobre cada um dos métodos para resolução do termo temporal.

$$\frac{\partial}{\partial t} \int_V \rho \phi dV = \frac{3(\rho_P \phi_P V)^n - 4(\rho_P \phi_P V)^o + (\rho_P \phi_P V)^{oo}}{2\Delta t} \quad (4.11)$$

Tabela 4.3: Esquemas de discretização disponíveis em `ddtSchemes`.

Esquema	Descrição
Euler	Primeira ordem, restrito, implícito
CrankNicholson	Segunda ordem, restrito, implícito
backward	Segunda ordem, implícito
steadyState	Não resolve a derivada temporal

4.3.4 Outros Operadores auxiliares no *OpenFOAM*

Além dos operadores diferenciais comentados no item anterior existe ainda uma grande diversidade de operadores auxiliares como, por exemplo:

- Operador `interpolationSchemes` que é usado para realocar a posição de um campo e tipicamente usado para interpolar valores do centro para a face das células.

- Operador `fluxRequired` que gera um fluxo a partir de um campo. Esse operador é necessário para gerar o fluxo após se resolver a equação da pressão para ser posteriormente usado na continuidade por exemplo.
- Operador `surfaceIntegrate` para calcular a integral sobre uma superfície, entre outros.

4.3.5 Definição e nomenclatura dos esquemas de interpolação no **OpenFOAM**

Uma questão muito importante em CFD é a escolha de esquemas de interpolação adequados. A [Tabela 4.4](#) mostra alguns esquemas de interpolação disponíveis no **OpenFOAM**.

Diferenças centrais (CD - *Central differencing*): é de segunda ordem de precisão, porém não é limitado, ou seja, dependendo das condições do escoamento pode ocorrer o surgimento de oscilações espúrias na solução. É expressa como mostrado na [Equação 4.12](#):

$$\phi_f = f_x \phi_P + (1 - f_x) \phi_N \quad (4.12)$$

onde $f_x = \overline{fN}/\overline{PN}$, sendo \overline{fN} a distância entre f e o centro da célula vizinha N e \overline{PN} a distância entre os centros das células N e P .

Diferenças upwind (UD - *Upwind differencing*): determina o valor da variável na face (ϕ_f) usando a direção do fluxo como mostrado pela [Equação 4.13](#). É um esquema limitado (não produz oscilações na solução) porém de primeira ordem de precisão.

$$\phi_f = \begin{cases} \phi_P & \text{para } F \geq 0 \\ \phi_N & \text{para } F < 0 \end{cases} \quad (4.13)$$

Diferenças mistas (BD - *Blended differencing*): esse esquema mistura o esquema UD com o CD ou outro esquema de alta ordem proporcionando propriedades de um e

Tabela 4.4: Alguns dos esquemas de interpolação presentes no *OpenFOAM*.

Esquemas Centrados	
Linear	Interpolação linear (diferenças centrais)
cubicCorrection	Esquema cúbico (correção cúbica)
midPoint	Interpolação linear com pesos simétricos
Esquemas para advecção (<i>Upwinded convection schemes</i>)	
upwind	Aproximação de um lado (<i>upwind differencing</i>)
linearUpwind	Upwind linear
skewLinear	Esquema linear com correção
QUICK	Diferenças upwind quadráticas (<i>Quadratic Upwind Differencing</i>)
Esquemas TVD (<i>Total Variation Diminishing</i>)	
limitedLinear	Diferenças lineares com limitador
vanLeer	Limitador van Leer
MUSCL	Limitador MUSCL
limitedCubic	Limitador cúbico
Esquemas NVD (<i>Normalised Variable Diagram</i>)	
SFCD	Diferenças centrais com filtro
Gamma ψ	Diferenças Gamma

do outro como pode ser observado na [Equação 4.14](#).

$$\phi_f = \phi_{UD} + \gamma[\phi_{HO} - \phi_{UD}] \quad (4.14)$$

onde γ é o limitador, o subscrito *HO* corresponde a esquemas de alta ordem e *UD* ao esquema *upwind* de baixa ordem. Para calcular os valores do limitador γ que garanta a ordem elevada sem presença de oscilações na solução pode-se usar diferentes esquemas limitadores de fluxo como van Leer, SUPERBEE, MINMOD, OSPRE, etc. A idéia principal nos esquemas limitadores de fluxo é limitar as derivadas espaciais em valores realistas. O limitador é uma função do valor de ϕ e seu valor pode ser medido

por (SWEBY, 1984; LEER, 1974):

$$r = \frac{\phi_C - \phi_U}{\phi_D - \phi_C} \quad (4.15)$$

onde

$$\gamma = \gamma(r) \quad (4.16)$$

Os pontos U (*up*), C (*center*) e D (*down*) são selecionados de acordo com a direção do fluxo na face f . As Equações 4.17, 4.18, 4.19 representam, respectivamente, os limitadores MINMOD, van Leer e SUPERBEE.

$$\gamma_{mm}(r) = \max[0, \min(1, r)], \lim_{r \rightarrow \infty} \gamma_{mm}(r) = 1 \quad (4.17)$$

$$\gamma_{vl}(r) = \frac{r + |r|}{1 + |r|}, \lim_{r \rightarrow \infty} \gamma_{vl}(r) = 2 \quad (4.18)$$

$$\gamma_{sb}(r) = \max[0, \min(2r, 1), \min(r, 2)], \lim_{r \rightarrow \infty} \gamma_{sb}(r) = 2 \quad (4.19)$$

A Tabela 4.5 traz algumas informações sobre alguns esquemas de interpolação.

Tabela 4.5: Comportamento dos esquemas de interpolação usados em `divSchemes`.

Esquema	Comportamento numérico
linear	Segunda ordem, sem limitador de fluxo
skewLinear	Segunda ordem, sem limitador e com correção
cubicCorrected	Quarta ordem, sem limitador e com correção
upwind	Primeira ordem, limitado
linearUpwind	Primeira/Segunda ordem, limitado
QUICK	Primeira/Segunda ordem, limitado
Esquemas TVD	Primeira/Segunda ordem, com limitador
SFCD	Segunda ordem, com limitador
Esquemas NVD	Primeira/Segunda ordem, com limitador

4.3.6 Solvers para Solução do Sistema Linear de Equações

O *OpenFOAM* possui diversos *solvers* para resolver o sistema linear discretizado. A [Tabela 4.6](#) apresenta alguns deles.

Tabela 4.6: *Solvers* para o sistema linear.

<i>Solver</i>	Palavra chave
Gradiente (bi-)conjugado com pré-condicionamento	PCG/PBiCG†
Gradiente (bi-)conjugado estabilizado	BiCGStab
Solver usando um smoother	smoothsolver
Multi-grid generalizado	GAMG

†PCG para matrizes simétricas, PBiCG para assimétricas

Em relação ao emprego de pré-condicionadores, as opções disponíveis atualmente no *OpenFOAM* são apresentadas na [Tabela 4.7](#).

Tabela 4.7: Opções de pré-condicionadores.

Pré-condicionadores	Palavra chave
Diagonal incomplete-Cholesky (simétrico)	DIC
Cholesky	Cholesky
Faster diagonal incomplete-Cholesky (DIC com caching)	FDIC
Diagonal incomplete-LU (assimétrico)	DILU
Diagonal	diagonal
Geometric-algebraic multi-grid	GAMG
Sem pré-condicionador	none

4.3.7 Opções para Condições de Contorno

As condições de contorno podem ser divididas em dois tipos:

Dirichlet onde é dado o valor da variável no contorno. O valor dado no contorno é chamado no **OpenFOAM** de `fixed value`.

Neumann onde é fornecido o gradiente da variável dependente na superfície do contorno. Esse valor é chamado de `fixed gradient`.

Fixed Value é especificado o valor na fronteira ϕ_b .

1. Pode-se simplesmente substituir a valor de ϕ_b no local onde a discretização necessita do seu valor na face do contorno ϕ_f . Este caso é típico do termo advectivo.
2. Em termos como o Laplaciano, nos quais sua discretização requer o valor do gradiente na face $(\nabla\phi)_f$, este pode ser calculado usando o valor na fronteira e o valor do centro da célula.

$$S_f \cdot (\nabla\phi)_f = |S_f| \frac{\phi_b - \phi_P}{|d|} \quad (4.20)$$

Fixed Gradient é a condição de contorno onde o valor do gradiente g_b é especificado na fronteira. g_b é o produto escalar entre o gradiente e a unidade normal a fronteira.

$$g_b = \left(\frac{S}{|S|} \cdot \nabla\phi \right)_f \quad (4.21)$$

1. Quando a discretização requer o valor de ϕ_f no contorno este pode ser interpolado dos valores do centro da célula da seguinte forma:

$$\phi_f = \phi_P + d \cdot (\nabla\phi)_f = \phi_P + d \cdot g_b \quad (4.22)$$

2. g_b pode ser substituído diretamente em casos onde a discretização requer o gradiente na face.

$$S_f \cdot (\nabla\phi)_f = |S_f| g_b \quad (4.23)$$

Na [Tabela 4.8](#) são listadas as condições de contorno primitivas. Existem também diversas condições de contorno que derivam destas, sendo chamadas no *OpenFOAM* de condições de contorno derivadas. Não serão apresentadas as condições de contorno derivadas, contudo o leitor interessado pode consultar o *User's e Programmer's Guide* e trabalhos como os de [Jasak \(1996\)](#) e [Rusche \(2002\)](#) para maiores informações.

Tabela 4.8: Especificações primitivas para os contornos.

Tipo	Descrição da Condição	Dados especificados
<code>fixedValue</code>	Valor de ϕ especificado	Valor dado
<code>fixedGradient</code>	Gradiente normal de ϕ especificado	Gradiente dado
<code>zeroGradient</code>	Gradiente normal de ϕ é zero	—
<code>calculated</code>	Valor de ϕ é calculado	—
<code>mixed</code>	Condição mista de <code>fixedValue</code> e <code>fixedGradient</code>	Valor de referência
<code>directionMixed</code>	Condição mista entre a direção normal ao contorno e a direção tangencial	Valor de referência

Capítulo 5

Desenvolvimento do Solver para fluidos viscoelásticos: *viscoelasticFluidFoam*

*Neste capítulo será apresentada a metodologia usada no desenvolvimento do solver para resolver escoamentos de fluidos viscoelásticos no **OpenFOAM**, que foi denominado de **viscoelasticFluidFoam**. Também será discutida a implementação feita durante a realização deste trabalho.*

5.1 Escolha de uma Equação Constitutiva

Como visto na [Subseção 2.5.3](#) existe uma grande variedade de equações constitutivas para fluidos poliméricos. Desde as mais simples até as mais complexas se deve sempre fazer uma análise a respeito de sua aplicabilidade a um determinado problema. Ainda não existe uma equação que seja universal e consiga representar todos os escoamentos de fluidos poliméricos com precisão adequada. Um modelo pode conseguir representar bem um tipo de fluido e não ser bom para representar outro ([RENARDY, 2005](#)). No geral os modelos não-lineares conseguem ser mais adequados para representar fluidos complexos, porém são mais difíceis de serem resolvidos, uma vez que são mais

complexos e às vezes possuem parâmetros difíceis de serem obtidos. Uma comparação para soluções poliméricas em escoamento extensional uniaxial usando alguns dos modelos não-lineares apresentados pode ser encontrada no trabalho de [Tirtaatmadja e Sridhar \(1995\)](#).

O ideal então é ter conhecimento das equações existentes e suas aplicações para que então se possa escolher a que melhor se adapte. O conhecimento do tipo de fluido a ser simulado também é indispensável.

Para dar flexibilidade ao *solver* desenvolvido em termos de opções de equações constitutivas, foi feita a implementação de todas as equações constitutivas apresentadas na [Subseção 2.5.3](#). No entanto, por questões de dimensão do documento, neste trabalho são apresentados resultados para os modelos Maxwell linear, Oldroyd-B, Giesekus, PTT linear e exponencial, FENE-P e FENE-CR e o modelo DCP. Com o uso desses modelos serão apresentados resultados para as três teorias para fluidos viscoelásticos, a teoria cinética, a teoria de redes e a de reptação.

5.2 Escolha de uma Metodologia

Após a análise das formulações descritas na [Seção 3.3](#) escolheu-se a que seria usada neste trabalho.

A formulação viscosa é a mais indicada quando se deseja obter soluções transientes precisas, pois não é feita alteração alguma na equação original. Contudo esta formulação é estável somente para baixos valores de números de *Deborah* ([XUE et al., 2004](#)).

Em uma etapa preliminar do presente trabalho foram feitas algumas comparações para alguns modelos usando a formulação EVSS e DEVSS. Chegou-se a conclusão,

para os casos analisados, que a metodologia DEVSS devolve resultados iguais ou melhores que as da metodologia EVSS, sem requerer mudanças de variáveis e sendo, assim, facilmente aplicável a todos os modelos constitutivos (GUÉNETTE; FORTIN, 1995). Por estes motivos, a DEVSS foi a metodologia escolhida para ser utilizada neste trabalho.

Cabe ainda salientar que a metodologia AVSS e suas derivações não foram testadas neste trabalho, em consequência dos resultados preliminares satisfatórios obtidos com a metodologia DEVSS e à limitação de tempo para completar a dissertação. No entanto, seria relevante testá-la em um trabalho futuro e comparar seu desempenho com o da DEVSS.

5.3 Algoritmo para Resolução de escoamento de Fluido Viscoelástico

O procedimento usado para resolver o problema de escoamentos de fluidos viscoelásticos no *solver viscoelasticFluidFoam* pode ser resumido em 4 passos:

1. Considerando campos conhecidos de velocidade U , pressão p e tensão τ a equação de quantidade de movimento é resolvida implicitamente para cada componente do vetor velocidade, obtendo-se U^* . O gradiente da pressão e o divergente da tensão são calculados explicitamente usando os valores do passo anterior.
2. Com os novos valores de velocidade U^* calcula-se o novo campo de pressão p^* e posteriormente corrige-se a velocidade resultando em U^{**} que satisfaz a equação da continuidade. Este procedimento de cálculo para o campo de pressão e a correção da velocidade é feito usando o algoritmo PISO.
3. É feito o cálculo de cada componente do tensor das tensões τ^* usando uma equação constitutiva.

4. Para soluções mais precisas em escoamentos transientes pode-se iterar algumas vezes os passos 1, 2 e 3 dentro de um mesmo instante de tempo, sendo que para isto os valores de U , p e τ são atualizados com os valores de U^{**} , pressão p^* e tensão τ^* .

5.4 Detalhamento da implementação do *solver viscoelasticFluidFoam*

5.4.1 Código principal do *solver viscoelasticFluidFoam*

Nesta Seção é feito o detalhamento da implementação do *solver viscoelasticFluidFoam*, mostrando-se aspectos básicos do código computacional desenvolvido. Este detalhamento será feito tomando como base, a título de exemplo, um dos casos implementados que consiste na aplicação da metodologia DEVSS e na utilização do modelo constitutivo LPTT. Procedimentos similares foram utilizados para os demais modelos constitutivos apresentados na [Subseção 2.5.3](#).

O equacionamento é dado pela equação da continuidade ([Equação 2.12](#) da [Subseção 2.4.1](#)), conservação de quantidade de movimento usando a formulação DEVSS ([Equação 3.11](#) da [Subseção 3.3.3](#)) e a equação constitutiva LPTT ([Equação 2.40](#) da [Subsubseção 2.5.3.4](#)).

Para fim de implementação a equação de conservação de quantidade de movimento foi deixada da seguinte forma:

$$\frac{\partial U}{\partial t} + \nabla \cdot (UU) - \frac{\eta_s + \eta_p}{\rho} \nabla \cdot (\nabla U) = -\nabla \frac{p}{\rho} + \nabla \cdot \frac{\tau_p}{\rho} - \frac{\eta_p}{\rho} \nabla \cdot (\nabla U) \quad (5.1)$$

O código fonte em C++ para resolver esse problema está no arquivo principal `viscoelasticFluidFoam.C` dado pelo [Código 5.1](#), no arquivo contendo o modelo constitutivo ilustrado no [Código 5.3](#) e alguns arquivos auxiliares. Na seqüência será

apresentado e discutido o arquivo principal `viscoelasticFluidFoam.C`.

Código 5.1: *Solver viscoelasticFluidFoam* (arquivo `viscoelasticFluidFoam.C`).

```

2 // Description
3 //   Transient solver for incompressible, laminar flow of viscoelastic
   fluids.
4
5 /*-----*/

7 #include "fvCFD.H"
8 #include "viscoelasticModel.H"

10 // * * * * *

12 int main(int argc, char *argv[])
13 {

15 #   include "setRootCase.H"

17 #   include "createTime.H"
18 #   include "createMesh.H"
19 #   include "createFields.H"
20 #   include "initContinuityErrs.H"

22 // * * * * *

24   Info<< "\nStarting time loop\n" << endl;

26   while (runTime.run())
27   {

29 #       include "readPISOControls.H"
30 #       include "readTimeControls.H"
31 #       include "CourantNo.H"
32 #       include "setDeltaT.H"

34       runTime++;

36       Info<< "Time = " << runTime.timeName() << nl << endl;

38       // Pressure-velocity PISO corrector loop
39       for (int corr = 0; corr < nCorr; corr++)
40       {
41           // Momentum predictor
42           tmp<fvVectorMatrix> UEqn
43           (
44               fvm::ddt(U)
45               + fvm::div(phi, U)
46               - visco.divTau(U)
47           );

49           UEqn().relax();

```

```

51     solve(UEqn() == -fvc::grad(p));

53     p.boundaryField().updateCoeffs();
54     volScalarField rUA = 1.0/UEqn().A();
55     U = rUA*UEqn().H();
56     UEqn.clear();
57     phi = fvc::interpolate(U) & mesh.Sf();
58     adjustPhi(phi, U, p);

60     // Store pressure for under-relaxation
61     p.storePrevIter();

63     // Non-orthogonal pressure corrector loop
64     for (int nonOrth=0; nonOrth<=nNonOrthCorr; nonOrth++)
65     {
66         fvScalarMatrix pEqn
67         (
68             fvm::laplacian(rUA, p) == fvc::div(phi)
69         );

71         pEqn.setReference(pRefCell, pRefValue);
72         pEqn.solve();

74         if (nonOrth == nNonOrthCorr)
75         {
76             phi -= pEqn.flux();
77         }
78     }

80 #     include "continuityErrs.H"

82     // Explicitly relax pressure for momentum corrector
83     p.relax();

85     // Momentum corrector
86     U -= rUA*fvc::grad(p);
87     U.correctBoundaryConditions();

89     visco.correct();
90 }

92     runTime.write();

94     Info<< "ExecutionTime = "
95         << runTime.elapsedCpuTime()
96         << " s\n\n" << endl;
97 }

99     Info<< "End\n" << endl;

101     return(0);
102 }

104 // *****

```

A primeira linha do código declara a biblioteca `fvcfd.H` que é a principal biblioteca do *OpenFOAM* e que permite acesso a todas as classes. Essa biblioteca é necessária em todos os *solvers*. Na segunda linha é declarada a biblioteca denominada `viscoelasticModel.H` e necessária para a escolha e a construção dos modelos constitutivos. Essa biblioteca armazena instruções para a troca de dados entre o arquivo principal do *solver* e os arquivos que definem os modelos constitutivos.

A função `main` engloba o código fonte principal e possui dois argumentos de entrada: o `argc` e o `argv`. Estes parâmetros armazenam informações a respeito das condições da simulação, como por exemplo, o diretório e o nome do caso estudado. O programa lê estes argumentos através da linha de comando usada na execução do *solver*.

A biblioteca `setRootCase.H` é usada para testar a validade dos argumentos `argc` e `argv` da simulação.

As bibliotecas `createTime.H` e `createMesh.H` são responsáveis pelo armazenamento das informações do caso simulado e informações sobre a malha, respectivamente. Assim, a primeira permite acessar dados relativos ao tempo inicial, passo de tempo, tempo final da simulação entre outros e a segunda possui informações a respeito de volumes de controle, condições de contorno, etc. Essas duas bibliotecas são gerais no *OpenFOAM* e podem ser usadas por qualquer *solver*.

Já a biblioteca `createFields.H` é usada para leitura e criação de campos iniciais para as variáveis do problema e propriedades físicas pertinentes ao caso e é específica para cada *solver*. Essa biblioteca deve ser criada pelo programador para atender as necessidades de cada *solver*. Assim, dentre as bibliotecas mencionadas nesta seção, esta é a única que é específica do *solver viscoelasticFluidFoam*, por este motivo, ela será discutida de forma mais detalhada na [Subseção 5.4.2](#).

Seguindo em diante encontramos a biblioteca `initContinuityErrs.H` que é

responsável por inicializar e armazenar o erro relativo da equação da continuidade.

Em seguida é iniciado o laço do tempo em que se resolve a dinâmica do problema. A biblioteca `readPISOControls.H` é responsável pela leitura do número de iterações para o método de correção pressão-velocidade (PISO) e dos parâmetros de não-ortogonalidade da malha. O *OpenFOAM* usa o método PISO para problemas transientes pois esse gera uma solução mais precisa que o SIMPLE.

No solver *viscoelasticFluidFoam* é usado o comando `while(runtime.run())`, ou seja, enquanto não chegar o tempo final da simulação esta não será interrompida a não ser por um erro ou pelo usuário. Em alguns solvers esse comando é definido como `for(runtime++; !runtime.end(); runtime++)` em que, neste caso, é definido e fixado o incremento no tempo a ser usado. O comando `while(runtime.run())` não requer o fornecimento do incremento de tempo para cada vez que o laço é executado. Esse incremento é calculado automaticamente com base no valor de *Courant* (Co) máximo aceitável. Para que isso seja possível devemos usar 3 bibliotecas: a biblioteca `readTimeControls.H` que lê informações a respeito do uso de passo de tempo ajustável, qual o valor máximo de Co e do passo de tempo máximo especificado pelo usuário para a simulação. A biblioteca `CourantNo.H` que calcula o número de Co médio e máximo no domínio e a biblioteca `setDeltaT.H` que faz o cálculo de qual o valor do incremento no tempo que satisfaça o valor máximo de Co definido pelo usuário. Uma vez calculado o valor do passo de tempo o comando `runtime++` se encarrega de incrementar esse valor. Caso o valor desse passo de tempo seja maior que o passo de tempo máximo definido pelo usuário, o incremento é controlado pelo valor máximo de passo de tempo e não pelo Co .

Na seqüência inicia-se o laço para correção pressão-velocidade PISO. O laço será repetido até ser atingido o número de correções especificado pelo usuário em `nCorr`.

Logo no início deste laço é feito o armazenamento de todos os termos discretiza-

dos da equação de conservação de quantidade de movimento na matriz `UEqn` (classe `fvVectorMatrix`) menos o termo da pressão. O comando `visco.divTau(U)` possui embutido todas as parcelas referentes aos termos viscosos e elásticos da tensão e seu conteúdo é obtido do arquivo que contém os modelos constitutivos e que será apresentado posteriormente. Neste instante basta aceitar que o conteúdo da matriz `UEqn` é toda a equação de conservação de quantidade de movimento menos o termo da pressão. O comando `UEqn().relax()` faz uma relaxação explícita baseado em valores de relaxação fornecidos pelo usuário. Ao igualar `UEqn` ao negativo do gradiente da pressão (discretizado explicitamente pela classe `fvC`), a equação de conservação de quantidade de movimento é resolvida posteriormente com o comando `solve(UEqn() == -fvC::grad(p))` e essa é a primeira etapa do algoritmo de acoplamento pressão-velocidade PISO, ou seja, a etapa conhecida como *momentum predictor* de U (ISSA, 1986). Esta etapa também corresponde ao item 1 do procedimento apresentado na Seção 5.3. O PISO parte de uma forma discretizada da equação de conservação de quantidade de movimento, podendo ser representada pela Equação 5.2:

$$a_p U_p = H(U) - \nabla p \Leftrightarrow U_p = \frac{H(U)}{a_p} - \frac{\nabla p}{a_p} \quad (5.2)$$

O termo $H(U)$ armazena parcelas referentes ao transporte nas células vizinhas, o termo transiente e os termos fontes (B), como mostrado na Equação 5.3.

$$H(U) = - \sum_N a_N U_N + \frac{U^o}{\Delta t} + B \quad (5.3)$$

Até esse momento os valores do campo de velocidade foram calculados tendo como base somente a equação de conservação de quantidade de movimento sem se preocupar em satisfazer a equação da continuidade.

Na seqüência temos a atualização das condições de contorno para a pressão com o comando `p.boundaryField().updateCoeffs()`. Já com o comando `rUA = 1.0/UEqn().A()` é calculado o valor de 1 dividido pelos coeficientes a_p que serão armazenados na variável `rUA` do tipo `volScalarField`.

No próximo comando `U = rUA*UEqn().H()` é feito o cálculo simplificado dos valores de U necessários para a equação da pressão. Para criar a equação para a pressão toma-se o divergente na [Equação 5.2](#). Com isso obtêm-se a seguinte equação:

$$\nabla \cdot (U_P) = \nabla \cdot (U_P^*) - \nabla \cdot \left(\frac{1}{a_P} \nabla p \right) \quad (5.4)$$

Onde os valores de U_P^* são os valores calculados pelo comando anterior `U = rUA*UEqn().H()`.

Para o caso de escoamentos incompressíveis a parcela da esquerda da [Equação 5.4](#) é zero (continuidade) e podemos reescrever a equação como sendo:

$$\nabla \cdot (U_P^*) = \nabla \cdot \left(\frac{1}{a_P} \nabla p \right) \quad (5.5)$$

A parcela da esquerda é calculada explicitamente e é feito o cálculo da pressão que satisfaça a relação. A velocidade U_P^* é substituída pelo fluxo `phi`, uma vez que é necessária a velocidade nas faces das células. O fluxo é calculado pelos seguintes comandos: `phi = fvc::interpolate(U) & mesh.Sf()` e `adjustPhi(phi, U, p)`.

O comando `p.storePrevIter()` é usado para guardar os valores de pressão calculados na iteração anterior uma vez que esses valores serão necessários para fazer a relaxação da pressão.

No laço que segue é feita a definição e o cálculo da pressão baseado na [Equação 5.5](#), assim como as correções do fluxo para malhas não-ortogonais. O comando `pEqn.setReference(pRefCell, pRefValue)` utiliza os valores de `pRefCell` e `pRefValue` para setar os valores de referência para a pressão. Com o uso do comando `pEqn.solve()` se resolve a equação da pressão e posteriormente é feita sua correção com o comando `phi -= pEqn.flux()`.

A biblioteca `continuityErrs.H` faz o cálculo dos erros associados a equa-

ção da continuidade. O comando `p.relax()` realiza um relaxamento explícito da pressão para ser usada na etapa *momentum corretor* do algoritmo PISO e o comando `U -= rUA*fvc::grad(p)` aplica a correção da velocidade. Esta etapa de cálculo da pressão e correção da velocidade corresponde ao passo 2 do procedimento apresentado na [Seção 5.3](#). Por fim o comando `U.correctBoundaryConditions()` corrige as condições de contorno para a velocidade.

Ainda dentro do laço PISO faz-se o cálculo das tensões usando um modelo viscoelástico definido pelo usuário. O comando `visco.correct()` é responsável por resolver a equação constitutiva e assim atualizar os valores das tensões para serem usados na equação de conservação de quantidade de movimento (passo 3 do procedimento apresentado na [Seção 5.3](#)). A função `correct()` está definida no arquivo que contém informações referentes ao modelo viscoelástico e será discutida na [Subseção 5.4.3](#). Para se conseguir uma solução transiente de melhor precisão podem-se repetir os passos correspondentes as etapas de predição e correção do algoritmo PISO por algumas vezes (passo 4 do procedimento apresentado na [Seção 5.3](#)).

Por fim, é feita a saída de resultados em arquivos utilizando-se o comando `runTime.write()` e o comando `runTime.elapsedCpuTime()` retorna o tempo de simulação.

Recomenda-se que o leitor interessado em desenvolver códigos no *OpenFOAM* estude a fundo o trabalho de [Jasak \(1996\)](#) onde são apresentados detalhadamente vários aspectos sobre a formulação numérica, incluindo a metodologia de discretização, condições de contorno, etc., e a teoria dos algoritmos implementados, como o acoplamento pressão-velocidade, correção dos fluxos em malhas não-ortogonais, etc., implementados no *OpenFOAM*.

5.4.2 A biblioteca createFields.H

O conteúdo desta biblioteca para o *solver viscoelasticFluidFoam* está mostrado no código [Código 5.2](#).

Código 5.2: Biblioteca createFields.H do *Solver viscoelasticFluidFoam*.

```

2   Info << "Reading field p\n" << endl;
3   volScalarField p
4   (
5       IOobject
6       (
7           "p",
8           runTime.timeName(),
9           mesh,
10          IOobject::MUST_READ,
11          IOobject::AUTO_WRITE
12      ),
13      mesh
14  );

16  Info << "Reading field U\n" << endl;
17  volVectorField U
18  (
19      IOobject
20      (
21          "U",
22          runTime.timeName(),
23          mesh,
24          IOobject::MUST_READ,
25          IOobject::AUTO_WRITE
26      ),
27      mesh
28  );

30  # include "createPhi.H"

32  label pRefCell = 0;
33  scalar pRefValue = 0.0;
34  setRefCell(p, mesh.solutionDict().subDict("PISO"), pRefCell, pRefValue)
35      ;

36  // Create viscoelastic model
37  viscoelasticModel visco(U, phi);

```

A classe `volScalarField` constrói um campo escalar para variáveis escalares como a pressão (p) e a classe `volVectorField` constrói um campo vetorial para vetores como a velocidade (U). A criação desses campos depende de dois parâmetros

de entrada, as classes `IOobject` e `mesh`. A classe `IOobject` constrói o objeto, armazena o registro das informações contidas em `runTime`, define o nome da variável e o arquivo em que se encontra. As informações de entrada e saída especificam que estes devem ser obrigatoriamente lidos (`MUST_READ`) e são automaticamente gravados ao longo do tempo (`AUTO_WRITE`). Já a classe definida como `mesh` é necessária para obter informações sobre onde o campo será alocado e inserido.

A biblioteca `createPhi.H` é responsável por ler e criar um campo escalar `phi` (`surfaceScalarField`) de fluxo normal a superfície das células.

Os comandos que seguem (`pRefCell`, `pRefValue` e `setRefCell`) servem para tomar o valor da pressão de uma célula como referência (pressão relativa) e buscar no sub-dicionário PISO seus valores. Um dicionário no *OpenFOAM* é um arquivo com dados sobre alguma etapa da simulação. Um dicionário pode conter muitas informações divididas entre vários sub-dicionários que armazenam informações sobre algum assunto mais específico.

O último comando `viscoelasticModel visco(U, phi)` serve para a criação do modelo viscoelástico e para a troca de informações entre a função principal e a função que contém os modelos constitutivos. A criação de `visco(U, phi)` do tipo `viscoelasticModel` dentro da função principal (`main()`) permitirá que todos os modelos constitutivos sejam chamados, executados, e ainda, que o resultado de tudo isto esteja disponível para ser usado dentro da função principal. Portanto, este será o *link* para a função principal ter acesso aos modelos constitutivos.

5.4.3 Implementação do modelo Phan-Thien-Tanner linear

Será mostrado e discutido neste instante o arquivo que contém o modelo viscoelástico implementado, o arquivo `LPTT.C` que está mostrado no [Código 5.3](#).


```

56     return
57     (
58         fvc::div(tau_/rho_, "div(tau) ")
59         - fvc::laplacian(etaPEff/rho_, U, "laplacian(etaPEff,U) ")
60         + fvm::laplacian( (etaPEff + etaS_)/rho_, U, "laplacian(etaPEff+etaS,
61             U) ")
62     );
63 }

64
65 void LPTT::correct ()
66 {
67     // Velocity gradient tensor
68     volTensorField L = fvc::grad( U() );

69
70     // Convected derivate term
71     volTensorField C = tau_ & L;

72
73     // Twice the rate of deformation tensor
74     volSymmTensorField twoD = twoSymm( L );

75
76     // Stress transport equation
77     tmp<fvSymmTensorMatrix> tauEqn
78     (
79         fvm::ddt(tau_)
80         + fvm::div(phi(), tau_)
81         ==
82         etaP_ / lambda_ * twoD
83         + twoSymm( C )
84         - zeta_ / 2 * ( (tau_ & twoD) + (twoD & tau_) )
85         - fvm::Sp( epsilon_ / etaP_ * tr(tau_) + 1/lambda_, tau_ )
86     );

87
88     tauEqn().relax();
89     solve(tauEqn);

90 }

91
92
93 // * * * * *
94
95 } // End namespace Foam

96
97 // *****

```

A biblioteca `LPTT.H` é uma biblioteca exclusiva do modelo `LPTT` e contém informações para a construção deste modelo. Neste arquivo é construída a classe `LPTT` e são declarados todos os parâmetros e incógnitas pertinentes ao modelo `LPTT`, como a incógnita τ e os parâmetros ρ , η_S , η_P , ε , λ e ξ . São declaradas também funções

necessárias como a `correct()` e a `divTau(volVectorField& U)` por exemplo.

A biblioteca `addToRunTimeSelectionTable.H` possui macros para permitir uma fácil inserção do tempo de execução dentro da classe.

Em seguida é construída a classe `LPTT`, e é criado o objeto `tau` sendo de leitura obrigatória e salvamento automático. Também é feita a leitura do valor dos parâmetros necessários ao modelo.

Após a criação da classe do modelo, segue-se a implementação da função dada por `divTau(volVectorField& U)`. Essa função tem o objetivo de retornar o valor de $\nabla \cdot \tau$, onde neste caso τ apresenta às parcelas correspondentes a tensão do solvente, a tensão polimérica e também, neste caso, os termos referentes à formulação DEVSS. A variável `etaPEff` neste caso serve para armazenar um valor que será usado pela formulação DEVSS, ou seja, armazena o valor que será dado ao parâmetro κ na [Equação 5.1](#). Como já foi comentado no item em que se falou sobre a formulação DEVSS, este parâmetro pode assumir diferentes valores, contudo têm-se percebido que a viscosidade polimérica η_P é um bom valor para esse parâmetro e, desse modo, este foi o valor adotado nesta implementação.

Assim a parte de código correspondente aos comandos:

```

1   fvc::div(tau_/rho_, "div(tau) ")
2   - fvc::laplacian(etaPEff/rho_, U, "laplacian(etaPEff,U) ")
3   + fvm::laplacian( (etaPEff + etaS_)/rho_, U, "laplacian(etaPEff+etaS,
   U) ")

```

retornam o valor de:

$$\nabla \cdot \frac{\tau_P}{\rho} - \frac{\eta_P}{\rho} \nabla \cdot (\nabla U) + \frac{\eta_S + \eta_P}{\rho} \nabla \cdot (\nabla U)$$

que são as parcelas correspondentes ao divergente da tensão polimérica calculada explicitamente com o uso da classe `fvc`, mais a parcela correspondente ao Laplaciano da velocidade multiplicado por η_P , também calculado explicitamente, e o Laplaciano

da velocidade multiplicado pela soma de η_s com η_p calculado implicitamente pela classe `fvm`. A função `divTau(volVectorField& U)` será usada dentro da função principal (`main()`). Assim, os dados de tensão calculados usando o modelo viscoelástico são repassados para a equação de conservação de quantidade de movimento por esta função.

Seguindo adiante no código encontramos a função `correct()` que é onde está definido o modelo constitutivo. Para facilitar e tornar mais clara a implementação e o entendimento do código foram declarados os parâmetros `L`, `C` e `twoD`. O parâmetro `L` armazena o valor do gradiente da velocidade calculado explicitamente. No *OpenFOAM* o operador `&` é responsável por executar a operação produto escalar. O parâmetro `C` armazena o valor do produto escalar entre `L` e `tau`, ou matematicamente $C = L \cdot \tau$. Essa parcela entra no cálculo da derivada convectiva superior de τ . Já o parâmetro `twoD` recebe a operação da função `twoSymm()` sobre o valor de `L`. Essa função calcula o valor de $twoD = 2D = 2(1/2)(L + L^T)$. Na **Tabela 5.1** é mostrada cada parte do código com seu respectivo valor matemático no modelo LPTT.

Tabela 5.1: Representação de operadores matemáticos usando a linguagem do *OpenFOAM*.

Operação Matemática	Linha de código para esta operação
$\frac{D}{Dt} \tau_{PK}$	<code>fvm::ddt(tau_)</code> <code>+ fvm::div(phi(), tau_)</code>
$2 \frac{\eta_{PK}}{\lambda_K} D$	<code>etaP_ / lambda_ * twoD</code>
$[\tau_{PK} \cdot \nabla U] + [\tau_{PK} \cdot \nabla U]^T = 2 \left(\frac{1}{2} \right) (C + C^T)$	<code>twoSymm(C)</code>
$\xi_K (\tau_{PK} \cdot D + D \cdot \tau_{PK})$	<code>zeta_ / 2 * ((tau_ & twoD)</code> <code>+ (twoD & tau_)</code>
$\left(\frac{\varepsilon_K}{\eta_{PK}} tr(\tau_{PK}) + \frac{1}{\lambda_K} \right) \tau_{PK}$	<code>fvm::Sp(epsilon_ / etaP_ * tr(tau_)</code> <code>+ 1/lambda_, tau_)</code>

Por fim tem-se o comando `tauEqn().relax()` usado para fazer uma relaxação da variável `tau` e `solve(tauEqn)` para resolver o sistema discretizado contido em `tauEqn`. Maiores informações a respeito de bibliotecas usadas e classes do *OpenFOAM* podem ser consultadas no *Programmer's Guide* e no próprio endereço eletrônico do *OpenFOAM*.

5.5 Estudos de validação do código implementado

5.5.1 Escolha de uma geometria representativa

A geometria representada por uma contração plana foi adotada como padrão em 1987, durante o quinto *workshop* internacional sobre métodos numéricos para fluidos não-newtonianos (HASSAGER, 1988), para o caso específico de uma razão de contração 4:1. Desde então esta geometria foi enormemente usada e a quantidade de dados de literatura para essa geometria é imensa (ALVES et al., 2004). Essa geometria também é padrão e de fundamental importância em medidas experimentais de propriedades físicas. Consegue-se obter, para diferentes posições dessa geometria, diferentes condições de escoamento, como escoamento por cisalhamento e escoamento elongacional próximo a contração. Além disso, é possível fazer análises sobre o tamanho do vórtice no canto lateral e reentrante da geometria (AZAIEZ et al., 1996b).

Escoamentos de fluidos viscoelásticos por uma contração são também muito encontradas em indústrias, como no processo de moldagem e extrusão de polímeros.

Portanto, esta geometria foi escolhida para os estudos de validação do solver implementado, tomando como base de comparação os resultados de (QUINZANI et al., 1994) e de (AZAIEZ et al., 1996b).

Quinzani et al. (1994) realizaram experimentos em uma contração plana, com

seção anterior a contração possuindo largura de $2H = 0,0254m$ e seção posterior de $2h = 0,0064m$, conforme mostra a [Figura 5.1](#). Conseqüentemente a razão da contração é de 3,97:1. Nestes experimentos o fluido viscoelástico utilizado foi uma solução polimérica de 5% em peso de poli-isobutileno (PIB) em tetradecano (C_{14}) e foram feitas medidas de campos de tensão e velocidade a partir de medidas com LDV (Laser-Doppler Velocimetry) e FIB (Flow-Induced Birefringence).

[Azaiez et al. \(1996b\)](#) fizeram uma comparação entre resultados de simulações numéricas usando o método de elementos finitos (FEM) e o modelo de Giesekus ([GIESEKUS, 1982](#)) com os dados experimentais obtidos por [Quinzani et al. \(1994\)](#).

5.5.2 Parâmetros utilizados nos testes

Um estudo detalhado para o campo de velocidade e tensão é feito para uma única condição de fluxo, correspondendo a corrida (5) do artigo de [Quinzani et al. \(1994\)](#). O fluxo é caracterizado por dois números adimensionais, o número de Reynolds (Re) e o número de Deborah (De) definidos na [Equação 5.6](#).

$$Re_0 = \frac{2\rho\langle U \rangle h}{\eta_0}, \quad De_0 = \frac{\lambda\langle U \rangle}{h} = \lambda\langle \dot{\gamma} \rangle \quad (5.6)$$

onde se tem que $\langle U \rangle$ é a velocidade média no canal menor, η_0 é a soma entre a viscosidade polimérica e do solvente $\eta_0 = \eta_p + \eta_s$ e $\langle \dot{\gamma} \rangle$ é a taxa de deformação característica.

$$\langle \dot{\gamma} \rangle = \frac{\langle U \rangle}{h}, \quad \langle U \rangle = \frac{Q}{2Wh} \quad (5.7)$$

com $W = 0,254m$ sendo a profundidade do canal e Q a vazão volumétrica. A [Tabela 5.2](#) contém os dados referente a corrida 5 do trabalho de [Quinzani et al. \(1994\)](#). Na [Figura 5.1](#) o comprimento usado para L_1 foi de $80h$ e de $50h$ para L_2 . As condições de contorno são de não escorregamento nas paredes (*wall*), ou seja, velocidade zero e de simetria (*symmetry*) na linha central. Na entrada (*inlet*) será dada uma condição uniforme de entrada de massa que corresponde a velocidade média no canal menor

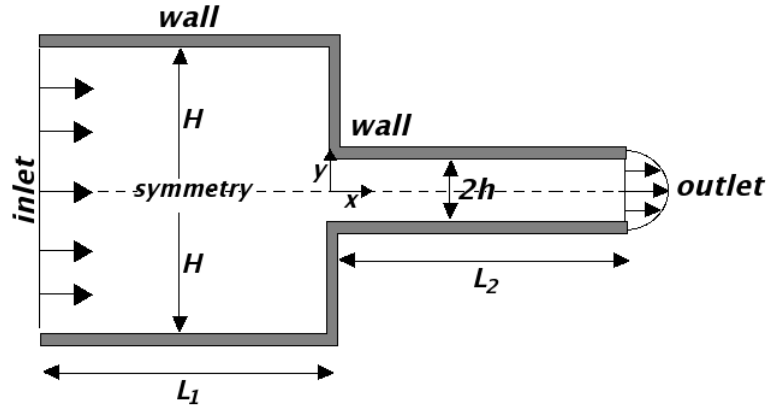


Figura 5.1: Esboço da geometria utilizada.

Tabela 5.2: Condições do escoamento.

$Q[cm^3s^{-1}]$	$\langle U \rangle [cms^{-1}]$	$\langle \dot{\gamma} \rangle [s^{-1}]$	Re_0	De_0
252	15,5	48,4	0,56	1,45 [†]

[†]Quinzani et al. (1994) usaram o modelo UCM (*Upper Convected Maxwell*) para encontrar as propriedades viscoelásticas e encontraram um tempo de relaxação $\lambda = 0,06s$ obtendo então um $De_0 = 2,90$ (RYSSEL; BRUNN, 1999).

divida por quatro. Na saída (*outlet*) é usada a condição de *Newmann*, onde se tem que o escoamento está plenamente desenvolvido.

Como condição inicial foi dado um valor igual a zero para todas as variáveis. Foi utilizado Crank-Nicholson para o termo temporal. Como *solvers* para o sistema linear discretizado usou-se CG com pré-condicionamento GAMG para a pressão. Para a velocidade e a tensão usou-se BiCGSTAB com um pré-condicionamento Cholesky. Serão mostrados resultados para a solução estacionária, ou seja, quando a variação da solução entre 2 passos de tempo consecutivos está dentro de uma tolerância admitida. A tolerância absoluta para a pressão foi de $1,0^{-7}$ e para a velocidade e tensão $1,0^{-6}$ dentro de um mesmo passo de tempo.

Capítulo 6

Resultados: Validação do Solver Desenvolvido

Neste capítulo são apresentados os resultados obtidos com o solver desenvolvido. É mostrada uma comparação com dados numéricos e experimentais da literatura. É feita também uma investigação da implementação para alguns dos modelos constitutivos implementados (Maxwell linear, Oldroyd-B, Giesekus, e os do tipo PTT e FENE).

6.1 Validação da implementação da estrutura básica do solver utilizando o modelo de Giesekus com um único modo

A avaliação da implementação da estrutura básica do solver foi feita por meio de testes de convergência de malha (Subseção 6.1.1), análise do desempenho dos esquemas de interpolação (Subseção 6.1.2) e comparação das predições do solver com dados experimentais e numéricos apresentados nos trabalhos de Quinzani et al. (1994) e Azaiez et al. (1996b), respectivamente, para uma contração plana 4:1 (Subseção 6.1.3). Para isto foram utilizados o mesmo modelo constitutivo e os parâmetros empregados por Azaiez et al. (1996b), ou seja, o modelo de Giesekus com o seguinte conjunto de parâmetros: $\alpha = 0,15$, $\lambda = 0,03s$, $\eta_p = 1,422Pa.s$ e $\eta_s = 0,002Pa.s$. A massa específica

usada foi $\rho = 803,87 \text{kg.m}^{-3}$.

6.1.1 Estudo de Convergência de Malha

Os testes de convergência de malha foram feitos com três diferentes malhas. As informações a respeito das malhas estão descritas na [Tabela 6.1](#). O refinamento foi feito junto às paredes e próximo à contração, como mostrado na [Figura 6.1](#). O refinamento deve ser feito nestes locais porque é onde ocorrem os maiores gradientes das variáveis. A malha usada foi toda hexaédrica.

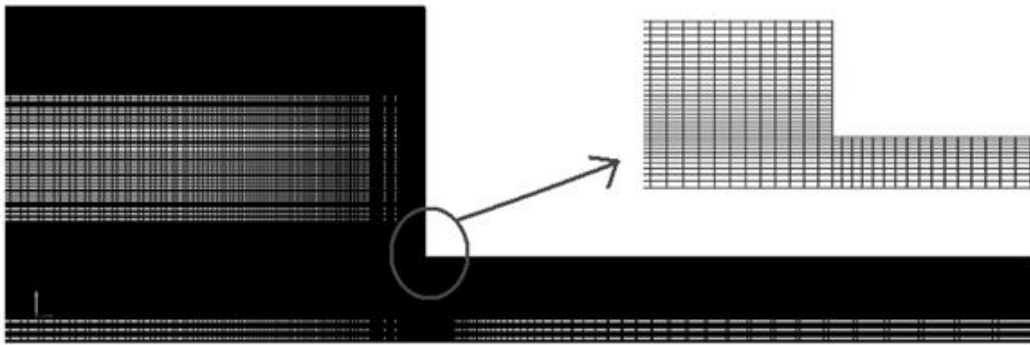


Figura 6.1: Malha computacional (Malha 2).

Tabela 6.1: Características da malha.

Malha	Número de CVs	$\Delta x_{min}/h$	$\Delta y_{min}/h$ †
1	9200	0,0098	0,026
2	20700	0,0065	0,017
3	36800	0,0049	0,013

† Os valores $\Delta x_{min}/h$ e $\Delta y_{min}/h$ correspondem a medidas no menor volume de controle (CV) próximo ao canto reentrante normalizado com o valor da metade da altura do canal na seção posterior a contração.

De acordo com os resultados mostrados percebe-se que a velocidade quase não

apresenta variação entre as três malhas diferentes, conforme [Figura 6.2](#). Para a tensão de cisalhamento, [Figura 6.3](#), e a primeira diferença de tensões normais, [Figura 6.4](#), são percebidas diferenças mais significativas, principalmente se comparado os valores obtidos com a malha mais grossa e a mais fina. Contudo, não são percebidas diferenças consideráveis entre a malha 2 e 3.

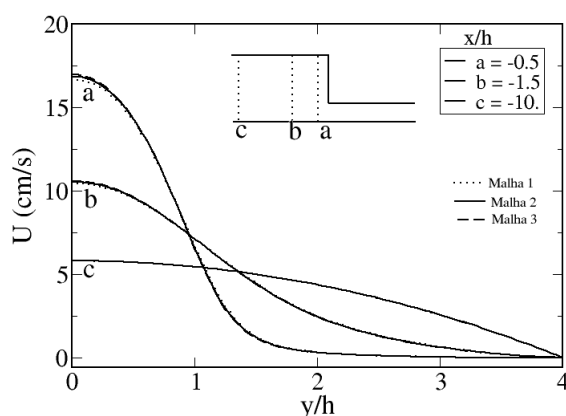


Figura 6.2: Perfis de velocidade U em um corte transversal na seção anterior a contração usando as malhas 1, 2 e 3.

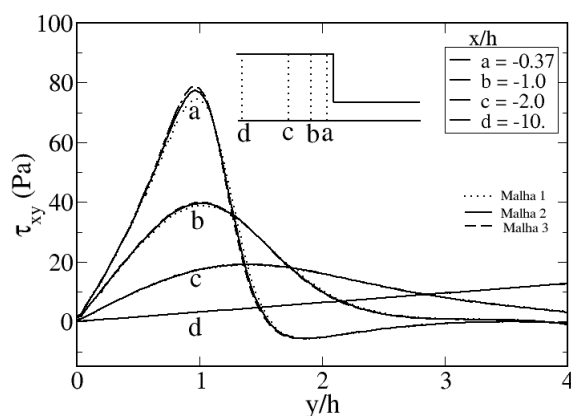


Figura 6.3: Perfis de tensão de cisalhamento τ_{xy} em um corte transversal na seção anterior a contração usando as malhas 1, 2 e 3.

A [Tabela 6.2](#) traz uma comparação quantitativa de um erro relativo máximo, em percentual, que os resultados das malhas 1 e 2 apresentam em relação a malha 3. O

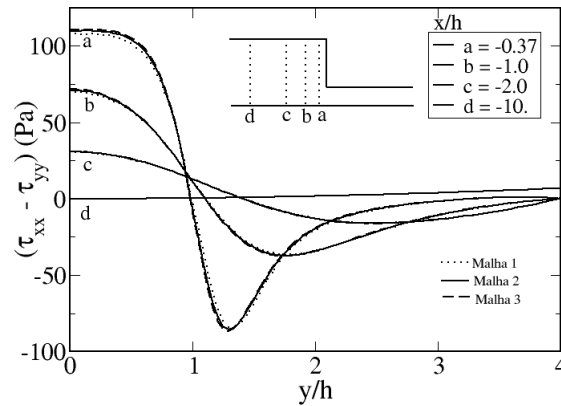


Figura 6.4: Perfis da primeira diferença de tensões normais N_1 em um corte transversal na seção anterior a contração usando as malhas 1, 2 e 3.

corte analisado corresponde às curvas referenciadas com a letra "a" nas Figuras 6.2 - 6.4, pois foram as que mostraram as maiores diferenças.

Tabela 6.2: Erro relativo, em percentual, das malhas 1 e 2 em relação a malha 3 para as curvas referenciadas com a letra "a" nas Figuras 6.2 - 6.4.

Malha	Erro relativo máximo (%)		
	U	τ_{xy}	N_1
1	2,034	3,068	6,329
2	0,729	1,054	2,118

O erro relativo máximo, em percentual, é dado por:

$$\% \text{ ERMax} = \max_{j=1}^N \left(\frac{|X_j^i - X_j^{ref}|}{\max(|X^{ref}|)} \right) \times 100 \quad (6.1)$$

onde X_j^i corresponde a pontos da curva "a", o índice i corresponde às malhas 1 ou 2 e j corresponde aos pontos da curva que vão de 1 a N . O termo X_j^{ref} corresponde a pontos da malha de referência, ou seja, a malha 3. O termo $\max(|X^{ref}|)$ retorna o valor máximo do módulo de X^{ref} e $\max_{j=1}^N$ retorna o valor de desvio máximo que ocorre dentre todos os pontos analisados.

Percebe-se que a variável que mais é afetada com a malha é a primeira diferença de tensões normais. Tomando como padrão a malha 3 tem-se um erro absoluto normalizado de mais de 6% para variável N_1 usando a malha 1. Para esta mesma análise usando a malha 2 tem-se um erro relativo máximo de pouco mais de 2%. A variação do menor volume de controle da malha 1 com a 2 é a mesma que a variação do menor volume da malha 2 para a 3, contudo percebe-se que o desvio da malha 1 em relação a 2 é de mais de 4% (6,329 - 2,118). Já a malha 2 em relação a 3 é de pouco mais de 2% indicando que as soluções estão se tornando independente da malha. Com essas informações pôde-se decidir em utilizar uma malha que garantisse a precisão desejada com o menor custo computacional. Levando em consideração que a malha 3 contém quase o dobro da quantidade de células da malha 2 e que, apesar disso, as diferenças de predição entre a malha 2 e 3 estão dentro de uma tolerância aceitável, optou-se pela utilização da malha 2 na obtenção do restante dos resultados apresentados neste trabalho.

6.1.2 Teste de Esquemas de Interpolação

O termo mais crítico das equações constitutivas utilizadas com relação à estabilidade numérica é o advectivo. Esse termo é responsável por introduzir instabilidades e oscilações na solução se não for usado um esquema de interpolação adequado.

Foi feita uma comparação entre alguns esquemas para o termo advectivo:

- *upwind*: primeira ordem (Equação 4.13).
- **MINMOD**: esquema de alta resolução que combina diferenças centrais com outros esquemas baseados em *upwind* de segunda ordem (HARTEN, 1983).
- *Gamma differencing*: Esquema baseado em NVD (*Normalised Variable Diagram*) e de primeira/segunda ordem (JASAK et al., 1999).

- **SFCD (Self-Filtered Central Differencing):** Esquema baseado em NVD de segunda ordem (STAR-CD, 2002).

As Figuras 6.5 e 6.6 ilustram o efeito que o uso de diferentes esquemas de interpolação causa na variável tensão de cisalhamento e primeira diferença de tensões normais em diferentes regiões da geometria estudada. O campo de velocidade praticamente não foi influenciado e assim não será mostrado.

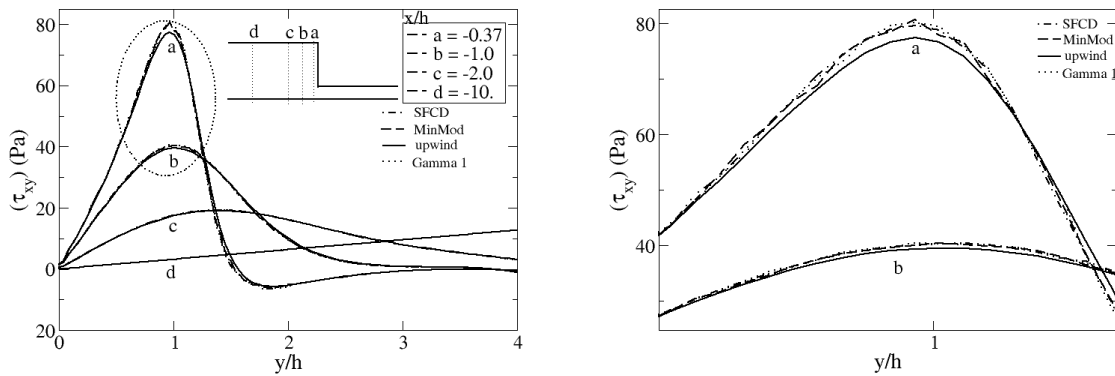


Figura 6.5: Perfis de tensão de cisalhamento τ_{xy} na seção anterior a contração para alguns esquemas de interpolação (esquerda). Ampliação da região em destaque (direita).

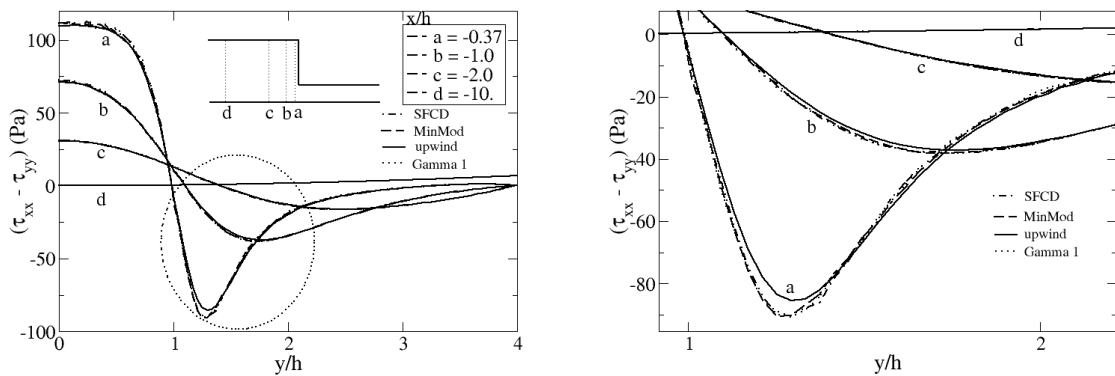


Figura 6.6: Perfis da primeira diferença de tensões normais N_1 na seção anterior a contração para alguns esquemas de interpolação (esquerda). Ampliação da região 2 em destaque (direita).

Pode-se perceber que os esquemas MINMOD, *Gamma 1* e SFCD apresentam resultados muito parecidos entre si. O esquema *upwind* difere destes principalmente onde ocorrem as maiores taxas de deformação e, conseqüentemente, os picos de tensões. Contudo, o *upwind* foi o único esquema totalmente livre de oscilações nos locais de elevadas taxas de deformação. O esquema SFCD foi o que mais introduziu oscilações nestas regiões. As oscilações podem melhor ser visualizadas na [Figura 6.7](#) onde foi feito um aumento (*zoom*) na região compreendida entre 0,4 à 0,7 na [Figura 6.6](#) da esquerda.

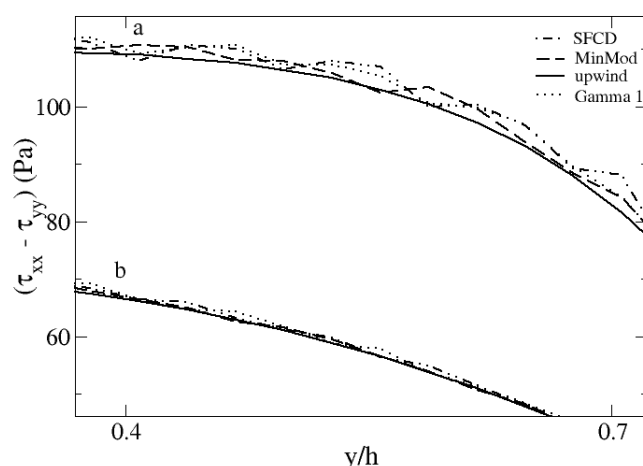


Figura 6.7: Representação das oscilações na região compreendida entre 0,4 à 0,7 na [Figura 6.6](#) à esquerda.

A [Tabela 6.3](#) mostra uma comparação numérica de um erro relativo máximo e um erro relativo médio, em percentual, do esquema *upwind* em relação ao esquema *Gamma 1* para a curva "a" que apresenta os maiores desvios.

Tabela 6.3: Erro relativo máximo e erro relativo médio, em percentual, do esquema *upwind* em relação ao *Gamma 1* para a curva "a".

Esquema	Erro Rel. Máx (%)		Erro Rel. Médio (%)	
	τ_{xy}	N_1	τ_{xy}	N_1
<i>upwind</i>	4,441	6,922	0,741	1,143

O erro relativo máximo foi calculado usando a [Equação 6.1](#), onde X_j^i corresponde a pontos para a curva "a" e o esquema *upwind* e X^{ref} corresponde a pontos para esta mesma curva e o esquema *Gamma 1*. Já o erro relativo médio é calculado segundo a [Equação 6.2](#).

$$\% \text{ ERM}_{ed} = \frac{\sum_{j=1}^N \left(\frac{|X_j^i - X_j^{ref}|}{\max(|X^{ref}|)} \right)}{N} \times 100 \quad (6.2)$$

Como pode ser observado o erro relativo máximo do esquema *upwind* em relação aos esquemas de alta resolução é considerável. O erro relativo máximo é maior para a primeira diferença de tensões normais chegando a quase 7%. Contudo, apesar de se obter um erro relativo máximo elevado, o erro relativo médio possui um valor aceitável sendo de pouco mais de 1%. Cabe mencionar que isto se deve ao fato de que desvios elevados ocorrem somente em algumas regiões específicas do escoamento sendo desprezível nas outras. Apesar de conseguir maior precisão usando os esquemas *Gamma 1* ou MINMOD optou-se por usar *upwind* uma vez que as diferenças observadas não afetarão a análise feita neste trabalho. Assim, os resultados mostrados nas seções seguintes foram obtidos usando o esquema *upwind* para os termos advectivos.

6.1.3 Comparação das predições com dados Numéricos e Experimentais

Nas Figuras [6.8 - 6.13](#) são ilustradas as superfícies de contorno para pressão, magnitude da velocidade, componentes de tensões τ_{xx} , τ_{xy} , τ_{yy} e as curvas de nível de linhas de corrente. Ao contrário das tensões τ_{xx} e τ_{xy} , que apresentam valores significativos em toda a seção posterior à contração, a tensão τ_{yy} é principalmente percebida junto ao canto reentrante. As linhas de corrente ([Figura 6.13](#)) mostram que surge um vórtice no canto superior da geometria.

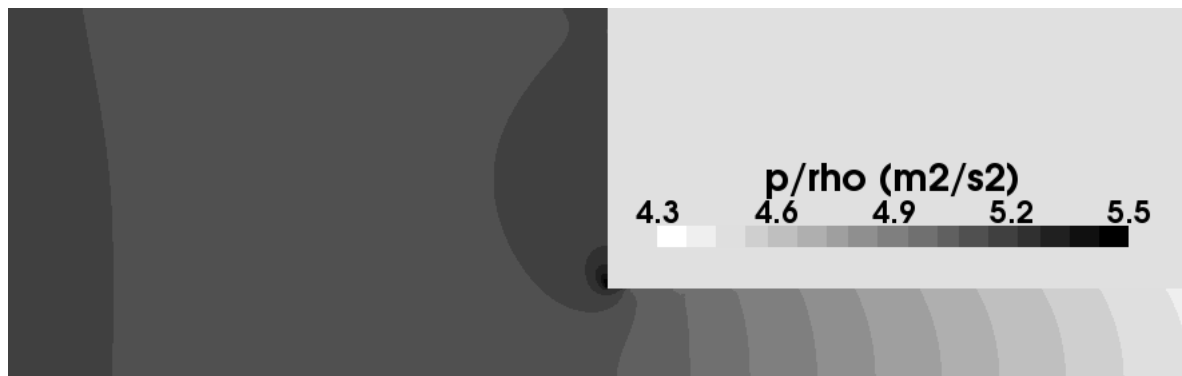


Figura 6.8: Campo de pressão.

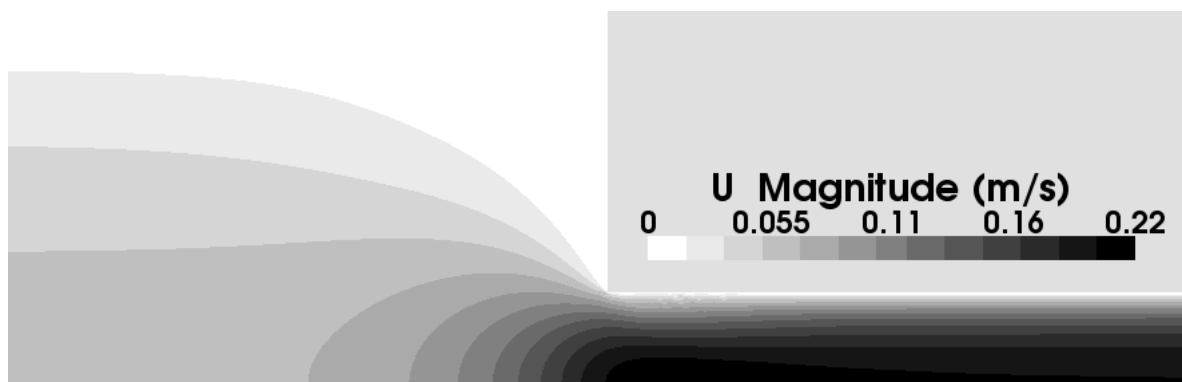


Figura 6.9: Campo velocidade.



Figura 6.10: Campo de tensão τ_{xx} .

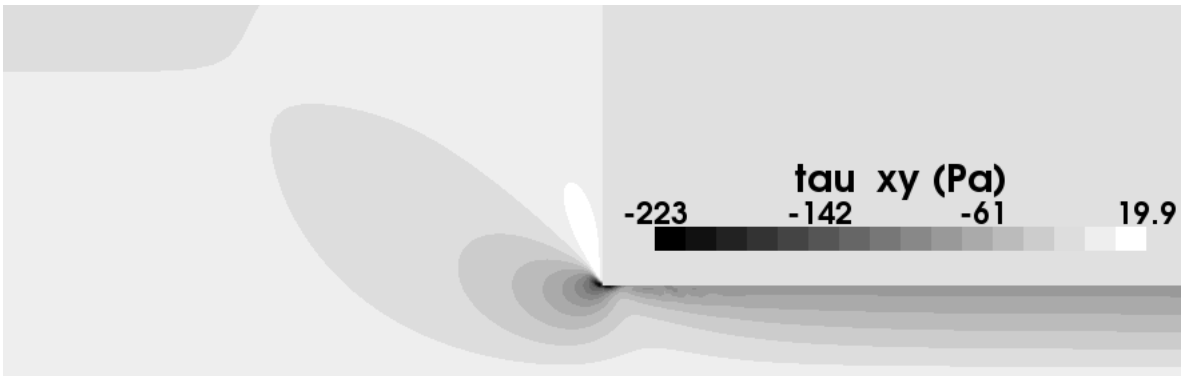


Figura 6.11: Campo de tensão τ_{xy} .

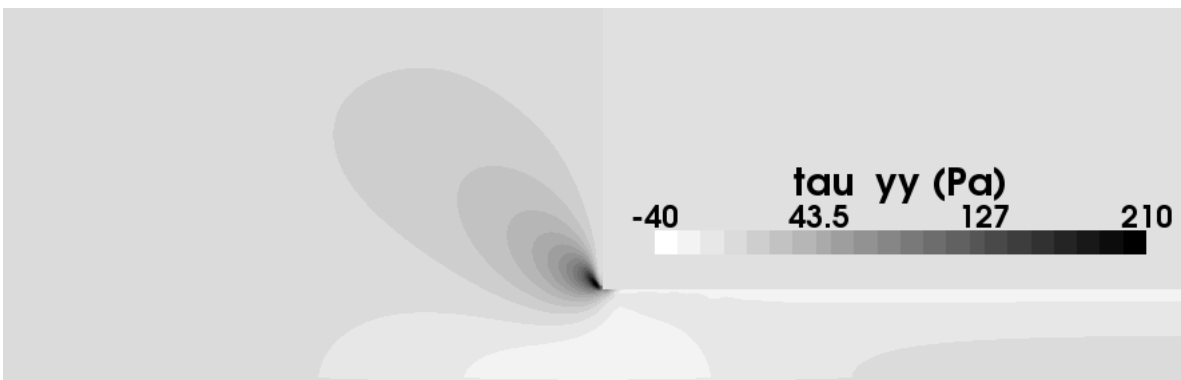


Figura 6.12: Campo de tensão τ_{yy} .

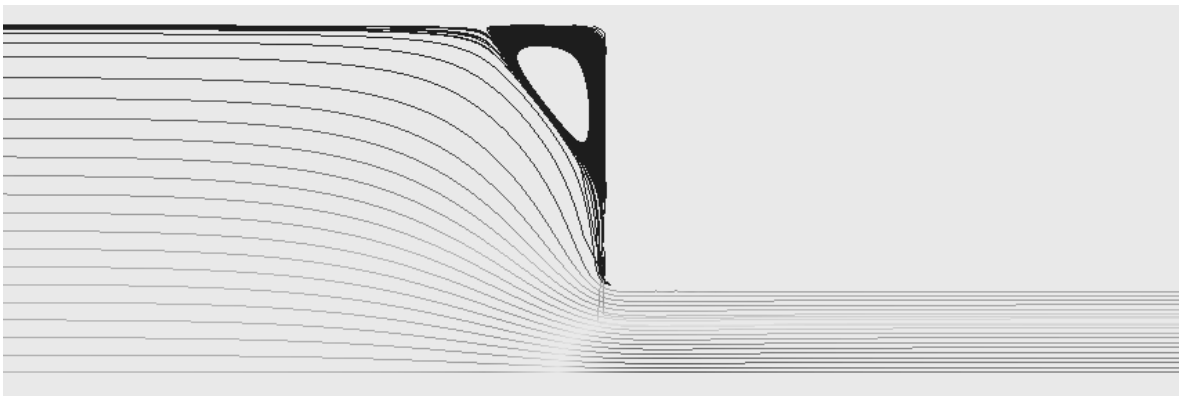


Figura 6.13: Linhas de corrente.

6.1.3.1 Dinâmica na seção anterior a contração

Na [Figura 6.14](#) é feita uma comparação com os resultados obtidos por [Quinzani et al. \(1994\)](#) e [Azaiez et al. \(1996b\)](#) para o perfil de velocidade U_x em cortes transversais e paralelos ao escoamento na seção anterior a contração. A [Figura 6.15](#) faz esta mesma comparação para a componente τ_{xy} da tensão e a [Figura 6.16](#) para a primeira diferença de tensões normais ($N_1 = \tau_{xx} - \tau_{yy}$).

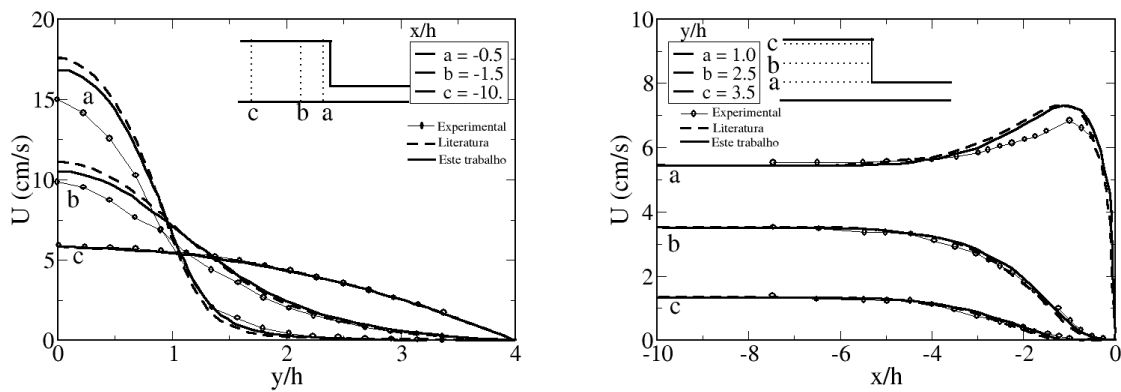


Figura 6.14: Perfis de velocidade U_x em cortes transversais (esquerda) e paralelos (direita) ao escoamento na seção anterior a contração.

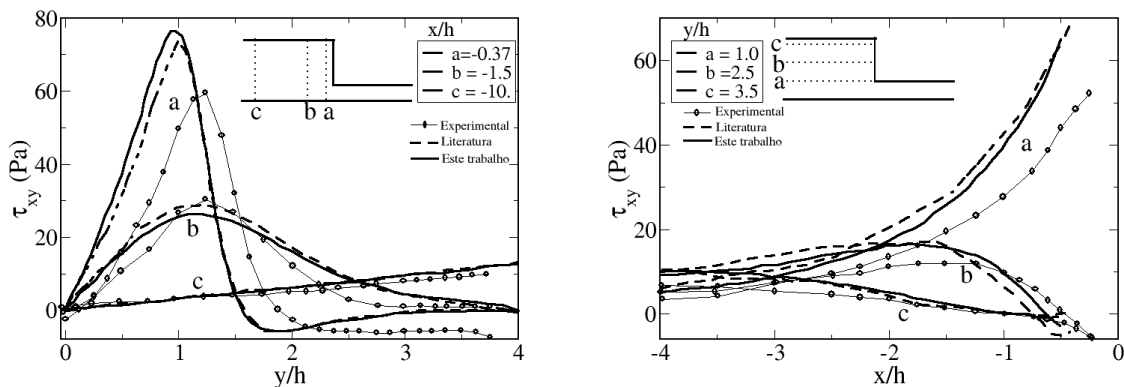


Figura 6.15: Perfis para a componente da tensão τ_{xy} em cortes transversais (esquerda) e paralelos (direita) ao escoamento na seção anterior a contração.

Percebe-se que os resultados obtidos para o perfil de velocidades são pratica-

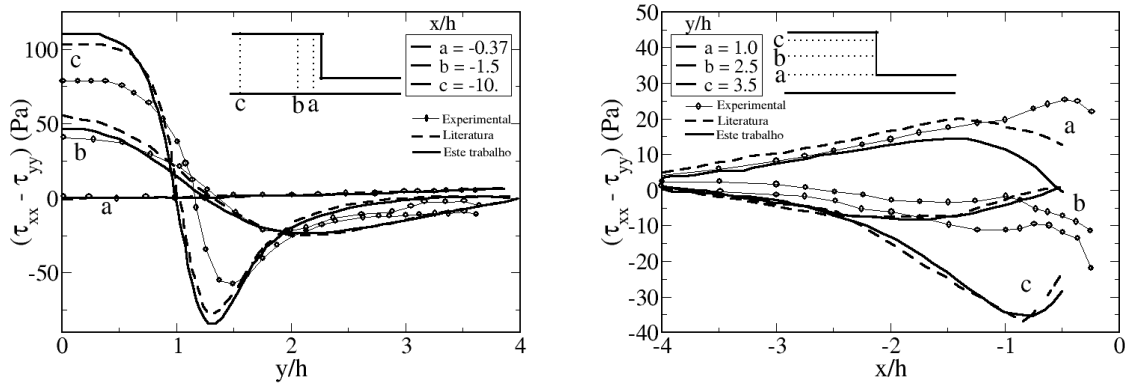


Figura 6.16: Perfis para a primeira diferença de tensões normais (N_1) em cortes transversais (esquerda) e paralelos (direita) ao escoamento na seção anterior a contração.

mente idênticos aos obtidos por [Azaiez et al. \(1996b\)](#) e que ambos apresentam boa concordância com os dados experimentais. Como já era de se esperar as maiores diferenças em relação aos dados experimentais ocorrem junto à contração, onde acontecem os valores extremos dos gradientes.

Para a tensão τ_{xy} na região próxima à contração obtiveram-se valores mais elevados do que os obtidos experimentalmente. Já para regiões onde o escoamento não sofre influência da contração houve uma concordância muito boa entre resultados numéricos e experimentais.

A primeira diferença de tensões normais é a variável mais crítica do problema, pois as elevadas taxas de alongação que ocorrem próximo à contração exigem da equação constitutiva boa capacidade de predição para se conseguir bons resultados. A [Figura 6.16](#) (direita) é a que apresenta um maior desvio entre resultados simulados e experimentais, contudo deve-se levar em conta que nesses locais é onde existem os maiores erros das medidas experimentais. Outro fator que pode influenciar nos resultados simulados é o uso de um único modo de relaxação podendo ser insuficiente para caracterizar bem a tensão em regiões onde existe uma elevada taxa de alongação. Deve-se lembrar que o uso de mais de um modo permite uma representação mais realista do comportamento reológico de amostras de polímero, já que estas geralmente

apresentam mais de um tempo de relaxação característico devido à heterogeneidade do tamanho de suas moléculas (distribuição de massa molar) e à possibilidade da presença de diferentes mecanismos de relaxação. Assim, o uso do modelo em sua versão multimodo pode ser importante para uma boa representação de regiões que apresentam altas taxas de alongação. Esta última hipótese foi testada e confirmada por [Azaiez et al. \(1996a\)](#). Assim, no presente trabalho também se efetuou a implementação da versão multimodo das diferentes equações constitutivas utilizadas. Os resultados relativos ao teste da versão multimodo do modelo de Giesekus para o caso de teste aqui analisado é apresentado mais adiante na [Seção 6.2](#).

6.1.3.2 Dinâmica na seção posterior a contração

Nas Figuras [6.17](#), [6.18](#) e [6.19](#) é apresentada a comparação entre as predições obtidas neste trabalho e os dados de literatura para os perfis de velocidades, da componente τ_{xy} da tensão e da primeira diferença de tensões normais (N_1), respectivamente.

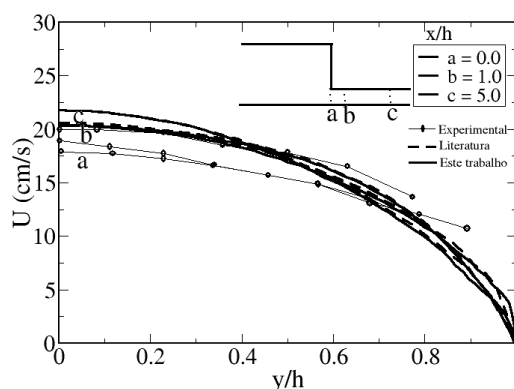


Figura 6.17: Perfis de velocidade Ux em um corte transversal ao escoamento na seção posterior a contração.

Na seção posterior à contração a taxa de deformação é maior e a diferença entre resultados numéricos e experimentais tornam-se mais aparentes. Os resultados numéricos apresentam boa concordância entre si e alguma diferença existente pode ser

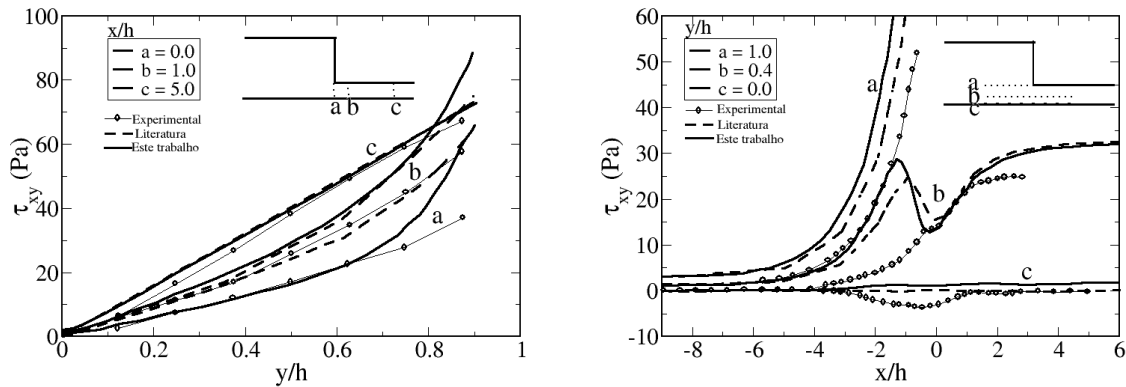


Figura 6.18: Perfis para a componente da tensão τ_{xy} em cortes transversais (esquerda) e paralelos (direita) ao escoamento na seção posterior a contração.

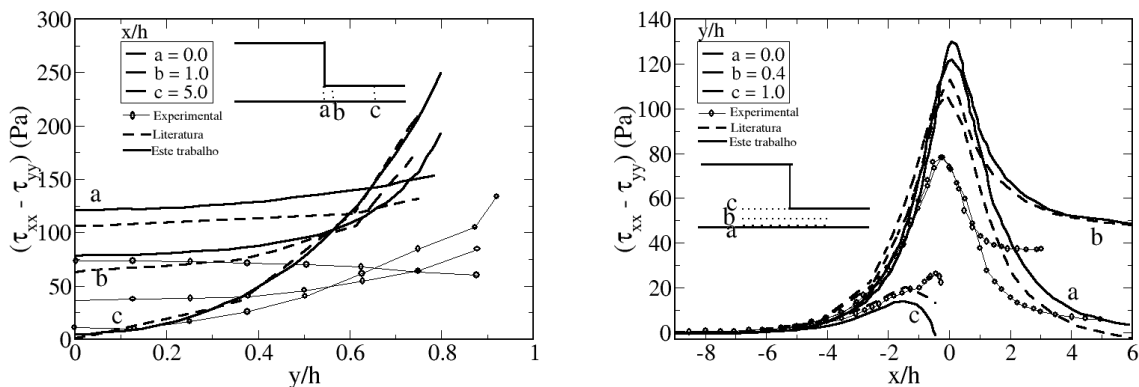


Figura 6.19: Perfis para a primeira diferença de tensões normais (N_1) em cortes transversais (esquerda) e paralelos (direita) ao escoamento na seção posterior a contração.

atribuída à diferença de malha e esquemas de interpolação usados. As previsões de velocidade concordam bem com os valores experimentais. Para a tensão foi encontrado o mesmo problema que na seção anterior a contração, onde se obteve, para altas taxas de deformação, valores super estimados.

A maior diferença entre os resultados numéricos e experimentais é observada para a primeira diferença de tensões normais, para a qual se obteve resultados com pouca concordância mesmo qualitativa para o corte transversal (Figura 6.19 (esquerda)). Uma vez desconsiderado os erros das medidas experimentais, acredita-se que este problema possa ser solucionado usando os modelos na versão multimodo, pelas mesmas

razões discutidas na seção anterior.

6.1.3.3 Efeito do valor de *Deborah*

Os valores de *Deborah* afetam o escoamento principalmente quando estes valores são elevados. Na [Figura 6.20](#) é feita uma análise de como os valores de *Deborah* afetam o perfil de velocidade e a primeira diferença de tensões normais na linha central do escoamento.

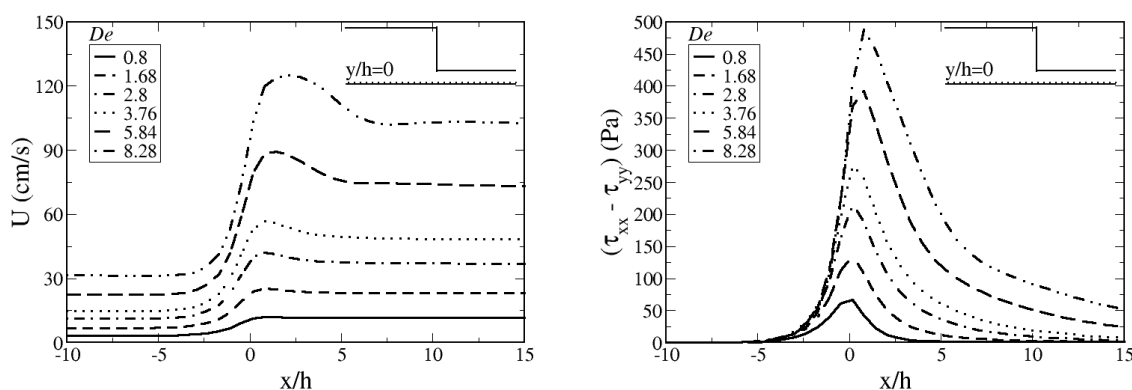


Figura 6.20: Perfis para a velocidade (esquerda) e primeira diferença de tensões normais (N_1) (direita) na linha central do escoamento e quando submetidos a diferentes valores de *Deborah*.

O fluxo na linha do centro do escoamento tem uma natureza alongacional e as propriedades alongacionais previstas pelos modelos reológicos representam uma característica muito importante para a correta predição dos fenômenos que ocorrem próximo ao canto reentrante. O modelo de Giesekus prediz um overshoot da velocidade próximo ao canto reentrante para valores de *Deborah* elevados. Percebe-se que o *overshoot* é tanto maior quanto maior for o valor de De .

A primeira diferença de tensões normais também tem seus valores aumentados com o aumento do número de *Deborah*. Os maiores valores, como já esperado, acontecem próximos à contração. Estes resultados estão de acordo com os mostrados

em [Azaiez et al. \(1996b\)](#).

6.2 Modelo de Giesekus na forma multimodo

Para testar a eficiência da estratégia utilizada na implementação da forma multimodo dos modelos diferenciais foi feita uma nova análise do caso descrito na seção anterior usando o modelo de Giesekus 4-modos. O conjunto de parâmetros utilizado nos testes é apresentado na [Tabela 6.4](#).

Tabela 6.4: Parâmetros para o modelo de Giesekus 4-modos (Fonte: [Azaiez et al. \(1996a\)](#)).

Modo	$\alpha[-]$	$\lambda[s]$	$\eta_p[Pa.s]$	$\eta_s[Pa.s]$	De
1	0,5	0,6855	0,0400	0,002	15,94
2	0,2	0,1396	0,2324	0,002	3,25
3	0,3	0,0389	0,5664	0,002	0,90
4	0,2	0,0059	0,5850	0,002	0,14

Os valores obtidos usando o modelo de Giesekus 4-modos serão comparados com os dados experimentais da corrida 3 do trabalho de [Quinzani et al. \(1994\)](#), correspondendo a um valor de *Reynolds* igual a 0,27 e valores de *Deborah* listados na [Tabela 6.4](#). Como os maiores desvios são encontrados na seção posterior a contração e próximo ao canto reentrante a análise será feita para esta região somente. Da mesma forma, será considerada nesta análise somente a primeira diferença de tensões normais, que foi identificada como a variável mais crítica. Os resultados estão ilustrados na [Figura 6.21](#).

Os resultados mostram uma concordância muito boa entre os dados numéricos obtidos da literatura e os obtidos neste trabalho, sendo que as diferenças podem ser atribuídas ao uso de malhas e metodologias diferentes.

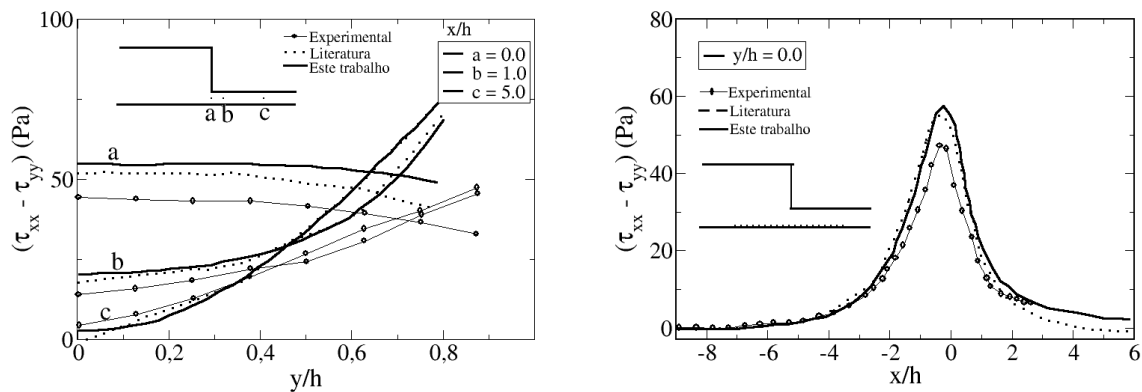


Figura 6.21: Perfis para a primeira diferença de tensões normais (N_1) em um corte lateral (esquerda) e na linha de simetria (direita) ao escoamento na seção posterior a contração usando o modelo de Giesekus 4-modos.

Como esperado os resultados obtidos na [Figura 6.21](#) (direita) são muito melhores que os obtidos anteriormente usando o modelo de Giesekus com um único modo para este mesmo corte. Essa comparação mostra que pelo menos qualitativamente os resultados concordam muito bem com os dados experimentais, ao contrário do que foi obtido usando um único modo [Figura 6.19](#). Como os resultados de medidas experimentais de tensões sempre carregam erros de medições, os resultados obtidos com o modelo de Giesekus 4-modos são satisfatórios para representar o fluido aqui analisado e fica justificado porque se obteve resultados pouco satisfatórios em relação aos dados experimentais na [Subsubseção 6.1.3.2](#).

Em relação ao custo computacional não se obteve um acréscimo muito grande (pouco mais de 2 vezes superior, confirmando as estimativas de [\(AZAIEZ et al., 1996a\)](#)) de tempo de cômputo com relação ao referente ao uso do modelo com um único modo. Isso pode ser explicado se considerado que o maior custo computacional está associado à resolução da equação de quantidade de movimento e à etapa de correção pressão-velocidade (PISO). Além do mais, os modos com baixos valores de De convergiram rapidamente e assim desde cedo não envolviam mais etapas iterativas demoradas.

6.3 Avaliação da implementação do modelo DCP

Para testar a implementação do modelo DCP foram usados os dados do trabalho de [Clemeur et al. \(2004\)](#) como base de comparação. A geometria usada foi uma contração plana abrupta similar a mostrada na [Figura 5.1](#). As dimensões foram de $2H = 4,10\text{mm}$ e $2h = 1\text{mm}$, dando uma contração de 4, 1 : 1. Tanto o canal anterior como o posterior à contração possuem medidas de 80mm . A malha usada foi similar à Malha 2 da [Subseção 6.1.1](#). O número de volumes de controle total foi de 22500.

No trabalho de [Clemeur et al. \(2004\)](#) foram feitas simulações para três diferentes taxas de deformações e algumas variações para os parâmetros não-lineares do modelo DCP também foram analisadas. Uma análise mais detalhada foi feita para a taxa de deformação aparente $\dot{\gamma}_a = 12,4\text{s}^{-1}$ e o caso denominado DCP1, cujos parâmetros são apresentados na [Tabela 6.5](#). Assim serão feitas comparações para esta taxa de deformação aparente e os parâmetros deste caso. Para se conseguir uma taxa aparente igual a $12,4\text{s}^{-1}$ no canal posterior à contração, usou-se um perfil uniforme com velocidade $U = 5,04 \times 10^{-4}\text{m}\cdot\text{s}^{-1}$ na entrada. Para estes parâmetros e a taxa de deformação aparente usada se obtém um valor de *Weissenberg* igual a 50,0. Maiores informações a respeito deste caso o leitor pode buscar diretamente do trabalho de [Clemeur et al. \(2004\)](#).

Tabela 6.5: Parâmetros para o modelo DCP 4-modos (Fonte: [Clemeur et al. \(2004\)](#)).

Modo	$\lambda_{ob}[s]$	$\lambda_{os}[s]$	$\eta_p[Pa\cdot s]$	$\xi[-]$	$q[-]$
1	0,02	0,01	$1,03 \times 10^3$	0,2	1,0
2	0,2	0,1	$2,22 \times 10^3$	0,2	1,0
3	2,0	1,0	$4,16 \times 10^3$	0,07	6,0
4	20,0	20,0	$1,322 \times 10^3$	0,05	18,0

Na [Figura 6.22](#) é comparado o perfil de velocidade obtido neste trabalho com o obtido na literatura. Na [Figura 6.23](#) é feita esta mesma análise para a variável PSD

(Principal Stress Difference) definida na Equação 6.3.

$$PSD = \sqrt{(\tau_{yy} - \tau_{xx})^2 + 4\tau_{xy}^2} \quad (6.3)$$

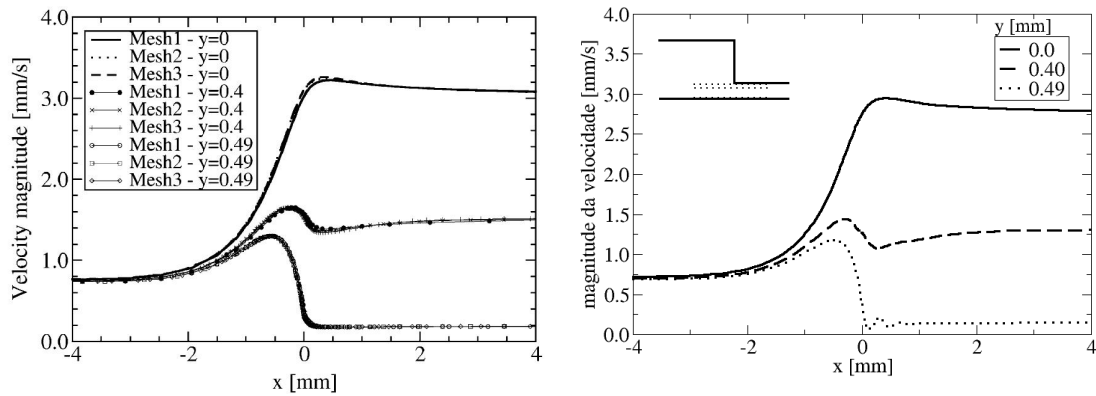


Figura 6.22: Perfis para a velocidade usando o modelo DCPD. Literatura (esquerda) e este trabalho (direita).

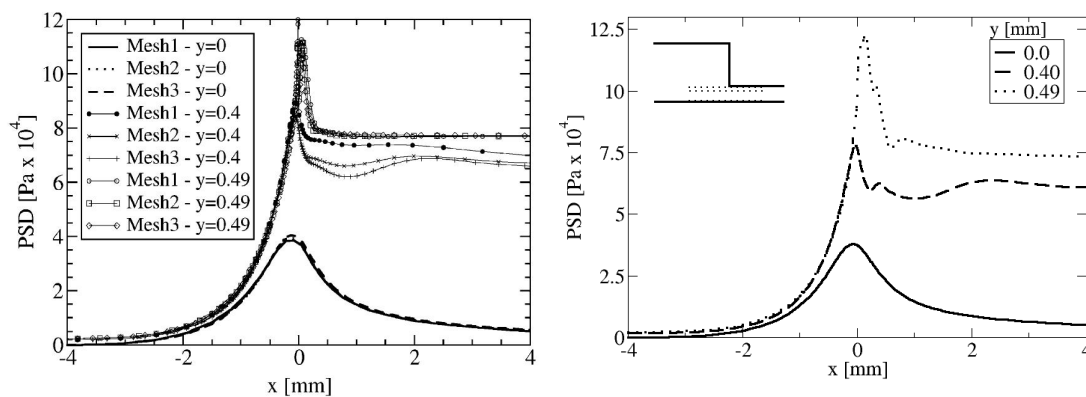


Figura 6.23: Perfis para a PSD usando o modelo DCPD. Literatura (esquerda) e este trabalho (direita).

Tanto os resultados para a velocidade como os de PSD concordam muito bem com os dados da literatura. Algumas diferenças podem ser atribuídas principalmente às diferentes malhas e esquemas de interpolação usados. A resolução do modelo DCPD é complicada uma vez que envolve tanto a derivada convectiva superior como a inferior em sua formulação. Além disso, duas equações devem ser resolvidas para posteriormente se obter a tensão viscoelástica, uma para o tensor orientação e outra para o estiramento. Contudo, não foram encontradas dificuldades numéricas para este

problema.

Com os resultados obtidos pode-se concluir que o modelo está corretamente implementado e a metodologia usada é satisfatória.

6.4 Avaliação da Implementação de outros Modelos Constitutivos

Nesta seção são descritos os resultados dos testes de validação da implementação dos seguintes modelos constitutivos: LPTTS, FENE-P, EPTTS, FENE-CR, Maxwell linear e Oldroyd- B. A geometria e as condições de escoamento são as mesmas que foram apresentadas na [Subseção 5.5.2](#) e utilizou-se como base de comparação os resultados obtidos com o modelo de Giesekus com os parâmetros apresentados na [Seção 6.1](#), já que tais resultados foram validados pela comparação com dados de literatura.

Os parâmetros para os modelos de LPTTS e FENE-P foram obtidos do trabalho de [Azaiez et al. \(1996b\)](#). Para os outros modelos não se tinha em mãos os parâmetros ajustados. No entanto, como a intenção principal é a avaliação da implementação considerou-se os parâmetros para o modelo EPTTS iguais ao do modelo LPTTS, os do modelo FENE-CR iguais ao do modelo FENE-P, apesar de que tal escolha não assegura que as previsões das funções materiais do fluido representado sejam iguais às do fluido utilizado na [Seção 6.1](#). Para os modelos de Maxwell linear e Oldroyd-B utilizou-se os mesmos parâmetros do modelo de Giesekus. Para o modelo de Maxwell linear os parâmetros lineares realmente correspondem aos do modelo de Giesekus, uma vez que os parâmetros lineares foram obtidos pelo ajuste dos parâmetros do modelo de Maxwell linear aos dados experimentais. Os parâmetros usados para cada modelo estão descritos na [Tabela 6.6](#). A análise dos modelos foi dividida em três grupos: (i) avaliação dos modelos LPTTS e FENE-P no qual os parâmetros foram obtidos diretamente do ajuste a dados experimentais, (ii) os modelos EPTTS e FENE-CR em que foram usados os parâmetros dos modelos LPTTS e FENE-P, respectivamente para as simulações, (iii) os modelos Maxwell linear e Oldroyd-B que não possuem

parâmetro não-linear adicional.

Tabela 6.6: Parâmetros dos modelos.

Modelo	<i>Parâmetro</i> [-]	λ [s]	η_P [Pa.s]	η_S [Pa.s]
LPTTS	0,25	0,03	1,422	0,002
FENE-P	6,0	0,04	1,422	0,002
Oldroyd-B	-	0,03	1,422	0,002
Maxwell linear	-	0,03	1,422	0,002
EPTTS	0,25	0,03	1,422	0,002
FENE-CR	6,0	0,04	1,422	0,002

6.4.1 Teste da implementação dos modelos LPTTS e FENE-P

As Figuras de 6.24 à 6.26 trazem uma comparação entre as previsões dos modelos LPTTS, FENE-P e de Giesekus. Nas são percebidas grandes diferenças no perfil de velocidade com o uso dos diferentes modelos analisados na Figura 6.24. Já a tensão τ_{xy} é muito mais sensível à mudança de equação constitutiva como pode ser observado na Figura 6.25. Nessa mesma figura percebe-se para o corte axial (esquerda) e a curva "b" uma tendência nos resultados obtidos, onde o modelo de LPTTS prediz valores de tensão maiores que o modelo de Giesekus e o modelo FENE-P prediz valores de tensão maiores que o modelo de LPTTS. O modelo de Giesekus previu valores maiores que os experimentais (Figura 6.15) para esta mesma análise e deste modo os modelos LPTTS e FENE-P mostram erros ainda maiores em relação ao experimental.

Para a primeira diferença de tensões normais (Figura 6.26) percebe-se algo parecido ao que aconteceu para a tensão τ_{xy} . Os modelos de Giesekus e LPTTS possuem resultados semelhantes na Figura 6.26 (esquerda). O modelo FENE-P é o que mais difere dentre os modelos analisados e o que pior representaria os dados experimentais. A curva "b" da Figura 6.26 (direita) ilustra bem o fato do modelo FENE-P superestimar os valores de tensão em relação aos outros modelos. Contudo, o modelo FENE-P apresenta resultados com uma boa concordância qualitativa.

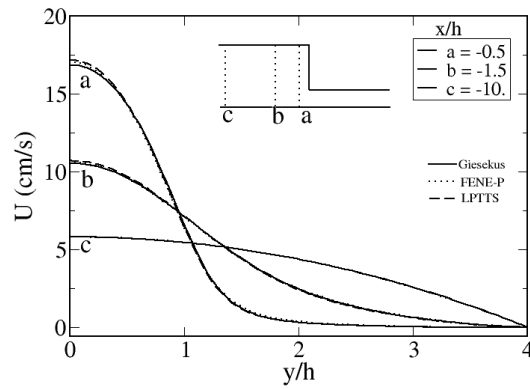


Figura 6.24: Perfis de velocidade U_x em um corte transversal ao escoamento comparando os modelos Giesekus, FENE-P e LPTTS.

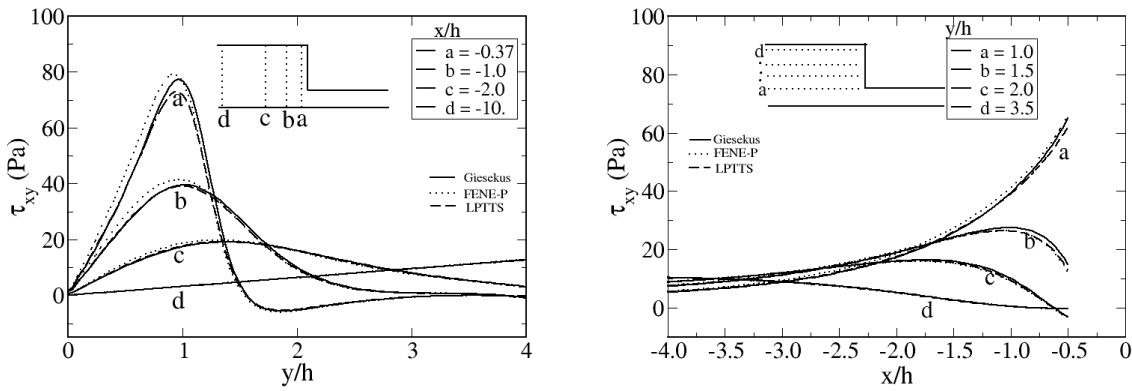


Figura 6.25: Perfis para a componente da tensão τ_{xy} em um corte transversal (esquerda) e paralelo (direita) ao escoamento comparando os modelos Giesekus, FENE-P e LPTTS.

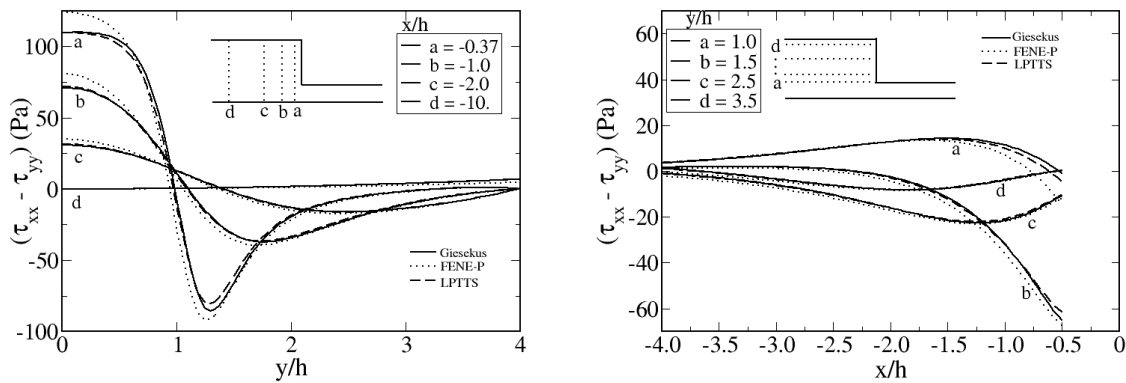


Figura 6.26: Perfis para a primeira diferença de tensões normais N_1 em um corte transversal (esquerda) e paralelo (direita) ao escoamento comparando os modelos Giesekus, FENE-P e LPTTS.

6.4.2 Teste da implementação dos modelos EPTTS e FENE-CR

As Figuras de 6.27 à 6.29 trazem uma comparação entre as previsões dos modelos EPTTS, FENE-CR e de Giesekus.

O perfil de velocidades representado pela Figura 6.27 não foi severamente afetado, porém teve maiores variações que a comparação feita usando os modelos da Subseção 6.4.1.

O perfil de tensão τ_{xy} (Figura 6.28) sofreu maiores variações principalmente quando comparado o modelo de Giesekus com o modelo FENE-CR. Os modelos Giesekus e EPTTS previram resultados muito parecidos para esta variável.

Para a primeira diferença de tensões normais (Figura 6.29) tem-se um resultado similar ao obtido para a tensão τ_{xy} . O modelo FENE-CR superestimou os resultados em relação aos dois outros modelos analisados. Uma observação importante é que o modelo EPTTS previu resultados melhores que os obtidos com o modelo de Giesekus na comparação com os dados experimentais (Figura 6.16).

Em uma análise qualitativa todos os modelos analisados apresentam boa concordância. O fato do modelo FENE-CR ter trazido resultados quantitativos ruins em relação aos outros modelos pode ser atribuída ao parâmetro não-linear usado, pois apesar dos parâmetros do modelo FENE-CR e FENE-P terem o mesmo significado, o ajuste dos parâmetros dependem de cada modelo e o uso dos parâmetros obtidos para o modelo FENE-P pode não ser adequado.

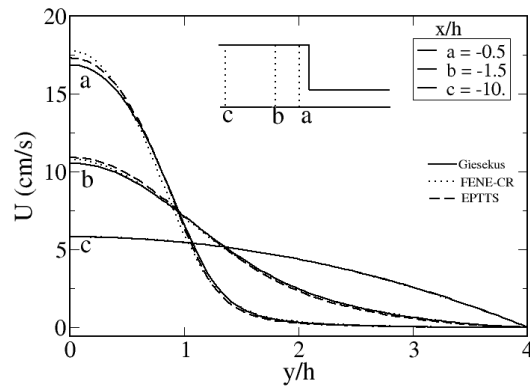


Figura 6.27: Perfis de velocidade U_x em um corte transversal ao escoamento comparando os modelos Giesekus, FENE-CR e EPTTS.

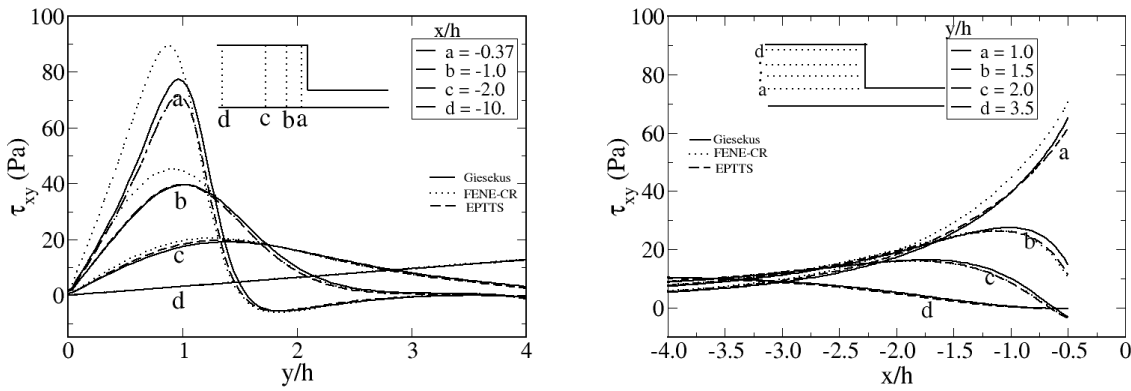


Figura 6.28: Perfis para a componente da tensão τ_{xy} em um corte transversal (esquerda) e paralelo (direita) ao escoamento comparando os modelos Giesekus, FENE-CR e EPTTS.

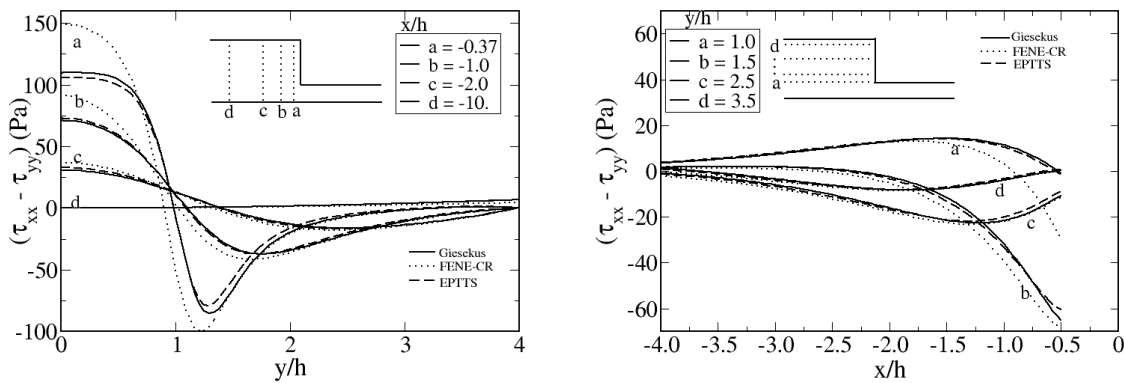


Figura 6.29: Perfis para a primeira diferença de tensões normais N_1 em um corte transversal (esquerda) e paralelo (direita) ao escoamento comparando os modelos Giesekus, FENE-CR e EPTTS.

6.4.3 Teste da implementação dos modelos Maxwell linear e Oldroyd-B

Os modelos Maxwell linear e Oldroyd-B não apresentam parâmetros não-lineares. Os únicos parâmetros necessários para estes modelos são a viscosidade polimérica e o tempo de relaxação. Para esta análise usou-se os parâmetros obtidos para o modelo de Giesekus.

Como era de se esperar os maiores desvios acontecem para o modelo de Maxwell linear. Para o perfil de velocidade [Figura 6.30](#) o modelo Maxwell linear previu valores maiores que o modelo de Giesekus já o modelo Oldroyd-B previu valores menores. Para a tensão τ_{xy} ([Figura 6.31](#)) os resultados apesar de ter uma resposta qualitativa correta apresentam desvios muito grandes. Para a primeira diferença de tensões normais ([Figura 6.32](#)) o mesmo é observado.

Cabe mencionar também que a introdução de um parâmetro não-linear no modelo de Oldroyd-B, como ocorre para o modelo de Giesekus, produz resultados mais precisos do ponto de vista da representação da física do problema. Fica nítida a melhor qualidade preditiva do modelo de Giesekus em relação ao de Oldroyd-B para representação dos dados experimentais. Assim, pode-se afirmar que os desvios apresentados pelas predições dos modelos de Maxwell e Oldroyd-B com relação às do modelo de Giesekus são consequência das limitações inerentes a estes dois modelos e que suas implementações também estão corretas.

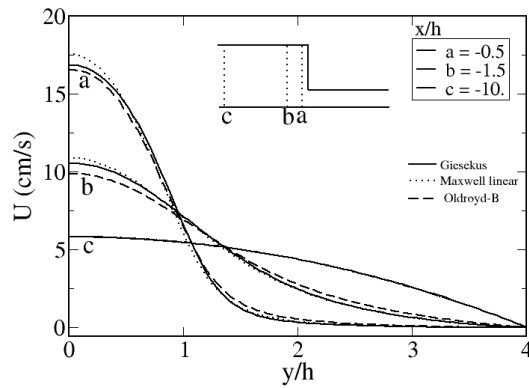


Figura 6.30: Perfis de velocidade Ux em um corte transversal ao escoamento comparando os modelos Giesekus, Maxwell linear e Oldroyd-B.

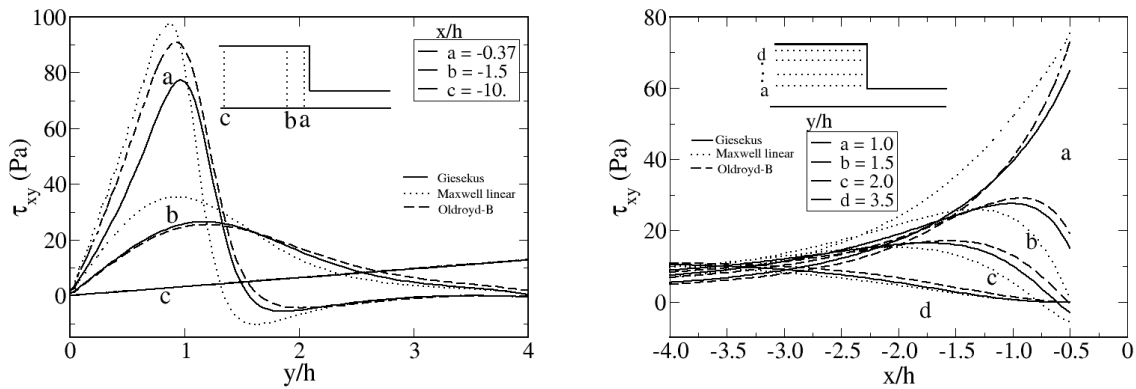


Figura 6.31: Perfis para τ_{xy} em um corte transversal (esquerda) e paralelo (direita) ao escoamento comparando os modelos Giesekus, Maxwell linear e Oldroyd-B.

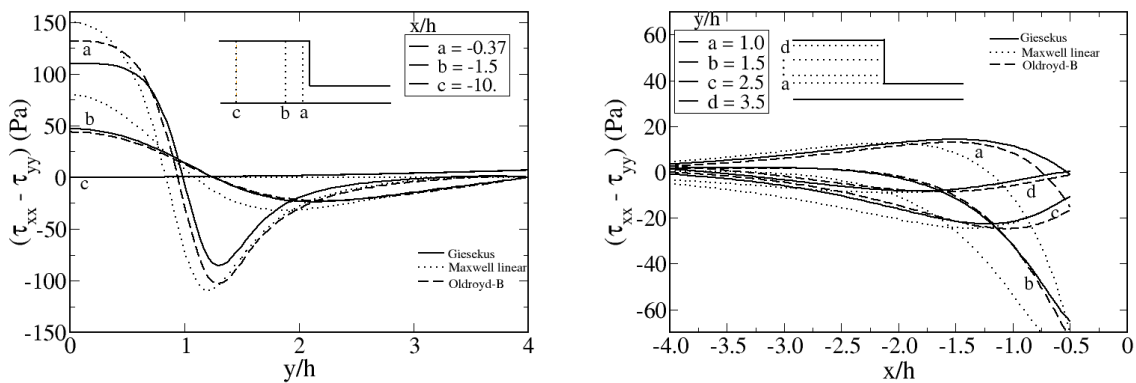


Figura 6.32: Perfis para N_1 em um corte transversal (esquerda) e paralelo (direita) ao escoamento comparando os modelos Giesekus, Maxwell linear e Oldroyd-B.

Capítulo 7

Conclusões

Foi proposto neste trabalho o desenvolvimento de um *solver* para fluidos viscoelásticos no pacote de CFD *OpenFOAM*. Foram apresentadas as principais vantagens que levaram a escolha deste pacote no qual foi feita a implementação do *solver* chamado *viscoelasticFluidFoam*. Apresentou-se também a metodologia usada para se resolver escoamentos com elevados valores de *Deborah* e foi feita uma revisão sobre as características dos fluidos viscoelásticos, equações constitutivas para fluidos viscoelásticos e uma breve descrição do *software OpenFOAM*.

Foram implementados os modelos Maxwell linear, UCM, Oldroyd-B, Giesekus, LPTTS, EPTTS, FENE-P, FENE-CR, Pom-Pom, WM, XPP e DCPP todos na forma multimodo. Para todos estes modelos foram apresentados resultados exceto para os modelos de WM, Pom-Pom e XPP. A implementação foi detalhada somente para o modelo constitutivo LPTT, porém pode-se tomar esta como base. O uso da linguagem do *OpenFOAM* facilitou bastante na criação de um *solver* com um grande potencial de aplicação.

No **Capítulo 6** foram apresentadas comparações de simulações usando o *solver* desenvolvido e dados numéricos e experimentais obtidos da literatura. Foi usada uma geometria padrão conhecida como contração plana abrupta com razão 4:1 e buscou-se

utilizar uma malha fina o suficiente para se obter resultados com boa precisão. Foi feito um elevado refinamento próximo ao canto reentrante para conseguir captar bem os efeitos que ocorrem nesta região que também é a mais crítica para esta geometria. As comparações feitas usando o modelo de Giesekus com apenas um modo de relaxação mostraram que o *solver* desenvolvido produz resultados satisfatórios quando comparado com os dados numéricos e experimentais da literatura. Tanto os dados numéricos da literatura quanto os simulados neste trabalho apresentam um desvio em relação ao experimental. Para um caso apresentado o desvio foi significativo e até mesmo uma má concordância qualitativa foi percebida. Essa discordância pode ser observada principalmente em locais sujeitos a elevadas taxas de alongação e a explicação para isto pode ser em parte atribuída a erros de medidas experimentais e também à própria capacidade preditiva do modelo de Giesekus com um único modo.

Para tentar melhorar os resultados numéricos em relação ao experimental foram feitas novas simulações usando o modelo de Giesekus com 4-modos. Os resultados obtidos tiveram uma melhora significativa sendo que somente algumas diferenças quantitativas foram observadas. Assim, ficou comprovado que para uma boa representação de resultados experimentais é indispensável o uso de multimodos. A explicação para se ter uma melhora nos resultados é evidente, uma vez que se consegue uma representação mais realista do comportamento reológico de amostras de polímero, já que estas geralmente apresentam mais de um tempo de relaxação característico devido à heterogeneidade do tamanho de suas moléculas (distribuição de massa molar) e à possibilidade da presença de diferentes mecanismos de relaxação. O custo computacional teve um incremento de pouco mais de 2 vezes quando comparado ao uso de um único modo.

Uma avaliação da implementação de diferentes modelos também foi apresentada. O modelo escolhido como referência foi o modelo de Giesekus. A avaliação dos diferentes modelos mostrou que diferenças significativas podem ser obtidas. Os modelos Maxwell linear, Oldroyd-B e FENE-CR foram os que mais se afastaram dos

resultados preditos pelo modelo de Giesekus. O modelo de Maxwell linear é limitado a baixas taxas de deformação e não prediz fenômenos não-lineares. O modelo de Oldroyd-B gera bons resultados para uma classe de fluidos conhecidos como "fluidos de Boger", porém a falta de parâmetros adicionais o torna muito limitado. Assim a obtenção de resultados piores em relação ao modelo de Giesekus é perfeitamente justificável. O modelo FENE-CR costuma gerar bons resultados, porém não foi o que foi visto neste trabalho. Provavelmente o parâmetro não-linear usado para este modelo não tenha sido adequado. Como não se tinham os parâmetros ajustados para este modelo foram usados os mesmos do modelo FENE-P. Acredita-se que tenha sido esta a causa dos valores serem bastante diferentes para este modelo. Já os modelos FENE-P e principalmente LPTTS e EPTTS mostraram resultados bastante próximos dos obtidos pelo modelo Giesekus. O modelo EPTTS se destacou por predizer resultados melhores que o modelo de Giesekus se comparado com os dados experimentais. Contudo, o principal objetivo era a avaliação da implementação feita e com os resultados obtidos pode-se garantir que os modelos estão corretamente implementados e a metodologia usada é adequada.

O efeito do valor do número de *Deborah* sobre um escoamento é assunto de muitos estudos. Foram feitas simulações usando o modelo de Giesekus sujeito a diferentes valores de *Deborah*. Os resultados mostraram que a velocidade é afetada com o surgimento de um *overshoot* próximo à contração. O *overshoot* se torna cada vez maior com o acréscimo do valor de *Deborah*. A primeira diferença de tensões normais também é afetada obtendo-se valores mais elevados com valores de *Deborah* maiores.

Foi feito também uma análise usando o modelo DCP 4-modos. Comparações para o campo de velocidade e a principal diferença de tensões (PSD) foram feitas com dados obtidos da literatura. O modelo DCP é um modelo que envolve a resolução de duas equações, uma para o tensor orientação e outra para o estiramento. Na equação para o tensor orientação existem muitos termos não-lineares representados

pelas derivadas convectivas superior e inferior. Este modelo é um dos mais recentes e existem grandes expectativas quanto ao seu uso. Os resultados simulados com este modelo mostraram uma concordância muito boa e é possível validar a implementação deste modelo e a metodologia utilizada.

Como conclusão geral pode-se afirmar que foi uma ótima escolha o uso do *software OpenFOAM* para se fazer o presente trabalho com fluidos viscoelásticos. O *solver* desenvolvido se mostrou muito eficiente e assim será disponibilizado na próxima versão do *software* e estará disponível livremente para todos os interessados. O código é aberto e os usuários poderão, além de executarem suas simulações, visualizar a implementação e toda a metodologia usada. Com o *solver* desenvolvido, é possível a simulação de uma grande diversidade de problemas dentro da indústria de processamento de polímeros e pode ser usado para outros fins, como simulação de fluidos biológicos por exemplo. Com a inclusão de todos os modelos de fluidos viscoelásticos apresentados o *OpenFOAM* passa a ser umas das melhores opções para se estudar escoamentos desse tipo de fluido.

Uma sugestão para um trabalho futuro é testar também novas metodologias como a DAVSS- ω , por exemplo, que parece ser muito boa. Outra sugestão é o desenvolvimento de uma metodologia para resolver escoamentos com superfície livre. O *OpenFOAM* já apresenta *solvers* para resolver superfície livre quando for usado fluido newtoniano ou fluido newtoniano generalizado. Assim seria necessário incluir o uso de modelos para fluidos viscoelásticos. Alguns testes preliminares neste sentido já foram feitos e os resultados foram satisfatórios. Com esta análise seria possível simular o preenchimento de moldes que ocorrem em processos de injeção e extrusão e também efeitos como o inchamento de extrudado, entre outros.

Referências Bibliográficas

- ABOUBACAR, M.; AGUAYO, J.; PHILLIPS, P.; PHILLIPS, T.; TAMADDON-JAHROMI, H.; SNIGEREV, B.; WEBSTER, M. Modelling pom-pom type models with high-order finite volume schemes. *Journal of Non-Newtonian Fluid Mechanics*, v. 126, n. 2-3, p. 207 – 220, 2005. ISSN 0377-0257. Annual European Rheology Conference 2003. Disponível em: <<http://www.sciencedirect.com/science/article/B6TGV-4FG4V9V-2/2/a254922bc1ed3cfcf968d66203093f29>>.
- AJIZ, M. A.; JENNINGS, A. A robust incomplete cholesky-conjugate gradient algorithm. *J. Numer. Methods. Engrg.*, v. 20, p. 949 – 966, 1984.
- ALVES, M. A.; OLIVEIRA, P. J.; PINHO, F. T. Benchmark solutions for the flow of oldroyd-b and ptt fluids in planar contractions. *Journal of Non-Newtonian Fluid Mechanics*, v. 110, n. 1, p. 45 – 75, 2003. ISSN 0377-0257. Disponível em: <<http://www.sciencedirect.com/science/article/B6TGV-47WD97B-2/2/09c59ded063c5cfba7f399f2ed918556>>.
- ALVES, M. A.; OLIVEIRA, P. J.; PINHO, F. T. A convergent and universally bounded interpolation scheme for the treatment of advection. *Int. J. Numer. Meth. Fluids*, v. 41, p. 47–75, 2003.
- ALVES, M. A.; OLIVEIRA, P. J.; PINHO, F. T. On the effect of contraction ratio in viscoelastic flow through abrupt contractions. *Journal of Non-Newtonian Fluid Mechanics*, v. 122, n. 1-3, p. 117 – 130, 2004. ISSN 0377-0257. XIIIth International Workshop on Numerical Methods for Non-Newtonian Flows. Disponível em: <<http://www.sciencedirect.com/science/article/B6TGV-4D8MKJT-2/2/4da95fcc6bc6fb898e545a69a62fd0c8>>.
- AZAIEZ, J.; GUÉNETTE, R.; AIT-KADI, A. Entry flow calculations using multi-mode models. *Journal of Non-Newtonian Fluid Mechanics*, v. 66, n. 2-3, p. 271 – 281, 1996. ISSN 0377-0257. Disponível em: <<http://www.sciencedirect.com/science/article/B6TGV-3VVMNS1-V/2/e80ddcde87f949cba70577242c8a9465>>.
- AZAIEZ, J.; GUÉNETTE, R.; AIT-KADI, A. Numerical simulation of viscoelastic flows through a planar contraction. *Journal of Non-Newtonian Fluid Mechanics*, v. 62, n. 2-3, p. 253 – 277, 1996. ISSN 0377-0257. Disponível em: <<http://www.sciencedirect.com/science/article/B6TGV-3TKMMG1-8/2/b9b59e81347a3b5b3c185a2d9f2289af>>.
- BAAIJENS, F. P. Mixed finite element methods for viscoelastic flow analysis: a review. *Journal of Non-Newtonian Fluid Mechanics*, v. 79, n. 2-3, p. 361 – 385, 1998. ISSN 0377-0257. Disponível em: <<http://www.sciencedirect.com/science/article/B6TGV-3V8KXCD-G/2/747950876caa60182aad51ec57559670>>.

- BIRD, R. B.; ARMSTRONG, R. C.; HASSAGER, O. *Dynamics of Polymeric Liquids*. 2nd. ed. New York: John Wiley & Sons, Inc., 1987.
- BIRD, R. B.; DOTSON, P. J.; JOHNSON, N. L. Polymer solution rheology based on a finitely extensible bead-spring chain model. *Journal of Non-Newtonian Fluid Mechanics*, v. 7, n. 2-3, p. 213 – 235, 1980. ISSN 0377-0257. Disponível em: <<http://www.sciencedirect.com/science/article/B6TGV-43PJ2RP-R/2/2b687c9c4bad907c1af8bbb31fb6393a>>.
- BOGAERDS, A. C. B.; GRILLET, A. M.; PETERS, G. W. M.; BAAIJENS, F. P. T. Stability analysis of polymer shear flows using the extended pom-pom constitutive equations. *Journal of Non-Newtonian Fluid Mechanics*, v. 108, n. 1-3, p. 187 – 208, 2002. ISSN 0377-0257. Disponível em: <<http://www.sciencedirect.com/science/article/B6TGV-479TJ1V-1/2/d0d7082505f4886c02adf01c102bcfe8>>.
- BRETAS, R. E. S.; DAVILA, M. A. *Reologia de Polímeros Fundidos*. 2nd. ed. [S.l.]: Ed. São Carlos: EDUFSCar, 2005.
- CARREAU, P. Tese (Doutorado) — University of Wisconsin, Madison, 1968.
- CHILCOTT, M. D.; RALLISON, J. M. Creeping flow of dilute polymer solutions past cylinders and spheres. *Journal of Non-Newtonian Fluid Mechanics*, v. 29, p. 381 – 432, 1988. ISSN 0377-0257. Disponível em: <<http://www.sciencedirect.com/science/article/B6TGV-43T9JJ5-2P/2/fb7bfe1b9f84a314b42b7a9e30c0434c>>.
- CLEMEUR, N.; RUTGERS, R. P. G.; DEBBAUT, B. On the evaluation of some differential formulations for the pom-pom constitutive model. *Rheologica Acta*, v. 42, n. 3, p. 217–231, maio 2003. Disponível em: <<http://dx.doi.org/10.1007/s00397-002-0279-2>>.
- CLEMEUR, N.; RUTGERS, R. P. G.; DEBBAUT, B. Numerical simulation of abrupt contraction flows using the double convected pom-pom model. *Journal of Non-Newtonian Fluid Mechanics*, v. 117, n. 2-3, p. 193 – 209, 2004. ISSN 0377-0257. Disponível em: <<http://www.sciencedirect.com/science/article/B6TGV-4C47GVS-3/2/d9d170224e2a3ee5f6560b5ec5875662>>.
- CROCHET, M. J.; PILATE, G. Plane flow of a fluid of second grade through a contraction. *Journal of Non-Newtonian Fluid Mechanics*, v. 1, n. 1, p. 247–258, 1976.
- DOU, H.-S.; PHAN-THIEN, N. The flow of an oldroyd-b fluid past a cylinder in a channel: adaptive viscosity vorticity (davss-[omega]) formulation. *Journal of Non-Newtonian Fluid Mechanics*, v. 87, n. 1, p. 47 – 73, 1999. ISSN 0377-0257. Disponível em: <<http://www.sciencedirect.com/science/article/B6TGV-3XM2YBR-3/2/9ef22465712cd6a0e342354e70696110>>.
- FERZIGER, J. H.; PERIC, M. *Computational methods for fluid dynamics*. Springer, 1999.
- FORUM, O. 2008. Disponível em: <<http://www.cfd-online.com/Forums/openfoam%-%/>>.
- GIESEKUS, H. A simple constitutive equation for polymer fluids based on the concept of deformation-dependent tensorial mobility. *Journal of Non-Newtonian Fluid Mechanics*, v. 11, n. 1-2, p. 69 – 109, 1982. ISSN 0377-0257. Disponível em: <<http://www.sciencedirect.com/science/article/B6TGV-446TT7G-J/2/47b8220a2b55346a2632d247345c7db3>>.

- GUÉNETTE, R.; FORTIN, M. A new mixed finite element method for computing viscoelastic flows. *Journal of Non-Newtonian Fluid Mechanics*, v. 60, n. 1, p. 27 – 52, 1995. ISSN 0377-0257. Disponível em: <<http://www.sciencedirect.com/science/article/B6TGV-3YYTV3H-G/2/6f2f21562d3207c6cfa92bcd34453ead>>.
- HARTEN, A. High-resolution schemes for hyperbolic conservation laws. *Journal of Computational Physics*, v. 49, p. 357–393, 1983.
- HASSAGER, O. in: Working group on numerical techniques, in: Proceedings of the fifth workshop on numerical methods in non-newtonian flow. *J. Non-Newton. Fluid Mech.*, v. 29, p. 2–5, 1988.
- HERRCHEN, M.; ÖTTINGER, H. C. A detailed comparison of various fene dumbbell models. *Journal of Non-Newtonian Fluid Mechanics*, v. 68, n. 1, p. 17 – 42, 1997. ISSN 0377-0257. Disponível em: <<http://www.sciencedirect.com/science/article/B6TGV-3S9DKTH-2/2/cc2e8ccacf222d9459886fa87714e765>>.
- ISSA, R. I. Solution of the implicitly discretised fluid flow equations by operator-splitting. *Journal of Computational Physics*, v. 62, n. 1, p. 40 – 65, 1986. ISSN 0021-9991. Disponível em: <<http://www.sciencedirect.com/science/article/B6WHY-4DD1XK9-1NN/2/230aa54fb7c104cb77120f1e210f6f0a>>.
- JACOBS, D. A. H. Preconditioned conjugate gradient methods for solving systems of algebraic equations. *Central Electricity Research Laboratories RD/L/N193*, v. 62, 1980.
- JASAK, H. *Error Analysis and Estimation for the Finite Volume Method with Applications to Fluid Flows*. Tese (Doutorado) — Imperial College of Science, Technology and Medicine, University of London., 1996.
- JASAK, H.; WELLER, H.; GOSMAN, A. High resolution nvd differencing scheme for arbitrarily unstructured meshes. *International Journal for Numerical Methods in Fluids*, v. 31, n. 1, p. 431 – 449, 1999. Disponível em: <[http://dx.doi.org/10.1002/\(SICI\)1097-0363\(19990930\)31:2<431::AID-FLD884>3.0.CO;2-T](http://dx.doi.org/10.1002/(SICI)1097-0363(19990930)31:2<431::AID-FLD884>3.0.CO;2-T)>.
- KENNEDY, P. *Flow Analysis of Injection Molds*. Munich: Hanser Gardner Publications, 1995. ISBN 1569901813.
- KING, R.; APELIAN, M.; ARMSTRONG, R.; BROWN, R. Numerically stable finite element techniques for viscoelastic calculations in smooth and singular geometries. *Journal of Non-Newtonian Fluid Mechanics*, v. 29, p. 147–216, 1988.
- LARSON, R. G. *Constitutive equations for polymer melts and solutions*. Department of Polymer Engineering, The University of Akron, Akron, OH 44325: Butterworths, 1988. 364 p. Boston.
- LEE, A. G.; SHAQFEH, E. S. G.; KHOMAMI, B. A study of viscoelastic free surface flows by the finite element method: Hele-shaw and slot coating flows. *Journal of Non-Newtonian Fluid Mechanics*, v. 108, n. 1-3, p. 327 – 362, 2002. ISSN 0377-0257. Disponível em: <<http://www.sciencedirect.com/science/article/B6TGV-47CXCC5-1/2/08541ad51e3456dcc997940951e20139>>.
- LEE, J.; ZHANG, J.; LU, C.-C. Incomplete lu preconditioning for large scale dense complex linear systems from electromagnetic wave scattering problems. *Journal of Non-Newtonian Fluid Mechanics*, v. 185, n. 1, p. 158 – 175, 2003. ISSN 00021-9991.

- LEER, B. V. Towards the ultimate conservative differencing scheme. ii monotonicity and conservation combined in a second-order scheme. *J. Comp. Physics*, v. 14, p. 361–370, 1974.
- LIELENS, G.; KEUNINGS, R.; LEGAT, V. The fene-l and fene-ls closure approximations to the kinetic theory of finitely extensible dumbbells. *Journal of Non-Newtonian Fluid Mechanics*, v. 87, n. 2-3, p. 179 – 196, 1999. ISSN 0377-0257. Disponível em: <<http://www.sciencedirect.com/science/article/B6TGV-3Y9N4D5-K/2/7bdd82b9acdc89fc14ad6101063a36b3>>.
- LIU, X.-D.; OSHER, S.; CHAN, T. Weighted essentially non-oscillatory schemes. *Journal of Computational Physics*, v. 115, n. 1, p. 200 – 212, 1994.
- LUO, X. L. A control volume approach for integral viscoelastic models and its application to contraction flow of polymer melts. *Journal of non-newtonian fluid mechanics. Journal of Computational Physics*, v. 64, n. 1, p. 173–189, 1996.
- MACOSKO, C. *Rheology: Principles, Measurements and Applications*. [S.l.]: VHC, 1994.
- MALISKA, C. R. *Transferência de Calor e Mecânica dos Fluidos Computacional*. 2. ed. Rio de Janeiro: LTC, 2004.
- MATALLAH, H.; TOWNSEND, P.; WEBSTER, M. F. Recovery and stress-splitting schemes for viscoelastic flows. *Journal of Non-Newtonian Fluid Mechanics*, v. 75, n. 2-3, p. 139 – 166, 1998. ISSN 0377-0257. Disponível em: <<http://www.sciencedirect.com/science/article/B6TGV-3TYNXBB-2/2/a1df5288b0ef8bedfb93a835bd7af75a>>.
- MAXWELL, J. *Phil. Trans. Roy. Soc.*, A157, p. 49–88, 1867.
- MCLEISH, T. C. B.; LARSON, R. G. Molecular constitutive equations for a class of branched polymers: The pom-pom polymer. *Journal of Rheology*, SOR, v. 42, n. 1, p. 81–110, 1998. Disponível em: <<http://link.aip.org/link/?JOR/42/81/1>>.
- MUNIZ, A. R. *Desenvolvimento de um Método de Volumes Finitos de Alta Ordem para a Simulação de Escoamentos de Fluidos Viscoelásticos*. Dissertação (Mestrado) — Universidade Federal do Rio Grande do Sul, Porto Alegre, 2003.
- MUNIZ, A. R.; SECCHI, A. R.; CARDOZO, N. S. High-order finite volume method for solving viscoelastic fluid flows. *Braz. J. Chem. Eng.*, São Paulo, Brasil, v. 25, n. 1, p. 53–58, mar. 2008. Disponível em: <<http://www.scielo.br/pdf/bjce/v25n1/a16v25n1.pdf>>.
- OLDROYD, J. *Proc. Roy. Soc.*, A200, p. 523–54, 1950.
- OPENFOAM. *OpenFOAM*. 9 Albert Road, Caversham, Reading, Berkshire RG4 7AN UK, 2008. Disponível em: <<http://www.opencfd.co.uk/openfoam/>>.
- OSTWALD, W. *Kolloid-Z*, v. 36, p. 99–117, 1925.
- PARAVIEW. *ParaView*. [S.l.], 2008. Disponível em: <<http://www.paraview.org/>>.
- PATANKAR, S. V. *Numerical heat transfer and fluid flow*. [S.l.]: Hemisphere Publishing Corporation, 1980.
- PATANKAR, S. V.; SPALDING, D. B. A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows. *Int. Heat and Mass Transfer*, v. 115, p. 1787–1803, 1972. Simple algorithm.

- PERERA, M. G. N.; WALTERS, K. Long-range memory effects in flows involving abrupt changes in geometry : Part i: flows associated with i-shaped and t-shaped geometries. *Journal of Non-Newtonian Fluid Mechanics*, v. 2, n. 1, p. 49 – 81, 1977. ISSN 0377-0257. Disponível em: <<http://www.sciencedirect.com/science/article/B6TGV-44FMMM-W/2/8d1c43aecb3d7b72597daab3f00ab954>>.
- PERIC, M.; KESSLER, R.; SCHEURER, G. Comparison of finite-volume numerical methods with staggered and colocated grids. *Computers & Fluids*, v. 16(4), p. 389–403, 1988.
- QUINZANI, L. M.; ARMSTRONG, R. C.; BROWN, R. A. Birefringence and laser-doppler velocimetry (ldv) studies of viscoelastic flow through a planar contraction. *Journal of Non-Newtonian Fluid Mechanics*, v. 52, n. 1, p. 1 – 36, 1994. ISSN 0377-0257. Disponível em: <<http://www.sciencedirect.com/science/article/B6TGV-44V49BT-68/2/cb383793226ddb02381ce4e8af81185b>>.
- RAJAGOPALAN, D.; ARMSTRONG, R. C.; BROWN, R. A. Finite element methods for calculation of steady, viscoelastic flow using constitutive equations with a newtonian viscosity. *Journal of Non-Newtonian Fluid Mechanics*, v. 36, p. 159 – 192, 1990. ISSN 0377-0257. Disponível em: <<http://www.sciencedirect.com/science/article/B6TGV-43T9S56-93/2/2e0d517dfefa92681d1d9066934c82de>>.
- RENARDY, M. Are viscoelastic flows under control or out of control? *Systems & Control Letters*, v. 54, n. 12, p. 1183 – 1193, 2005. ISSN 0167-6911. Disponível em: <<http://www.sciencedirect.com/science/article/B6V4X-4G9GN1D-2/2/dbd9130cf31c7d15b57ba02532058553>>.
- RHIE, C. M.; CHOW, W. L. Numerical study of the turbulent flow past an airfoil with trailing edge separation. *AIAA Journal*, v. 21, n. 11, p. 1523–1532, 1983.
- RUSCHE, H. *Computational fluid dynamics of dispersed two-phase flows at high phase fractions*. Tese (Doutorado) — Imperial College of Science, Technology and Medicine, Londres, Reino Unido, 2002.
- RYSSEL, E.; BRUNN, P. O. Comparison of a quasi-newtonian fluid with a viscoelastic fluid in planar contraction flow. *Journal of Non-Newtonian Fluid Mechanics*, v. 86, n. 3, p. 309 – 335, 1999. ISSN 0377-0257. Disponível em: <<http://www.sciencedirect.com/science/article/B6TGV-3XF07SM-2/2/9f786871bedcc9b2fe6f0d28dae1e758>>.
- SCHLEINIGER, G.; WEINACHT, R. J. A remark on the giesekus viscoelastic fluid. *Journal of Rheology*, SOR, v. 35, n. 6, p. 1157–1170, 1991. Disponível em: <<http://link.aip.org/link/?JOR/35/1157%-%/1>>.
- SCHOONEN, J. F. M. *Determination of Rheological Constitutive Equations using Complex Flows*. Tese (Doutorado) — Eindhoven University of Technology, 1998.
- STAR-CD. *Computational Dynamics Ltd., Star-CD Methodology Manual*. UK, 2002.
- STROUSTRUP, B. *C++ The Programming Language*. 3. ed. [S.l.]: John Wiley Sons, 1999.
- SUN, J.; PHAN-THIEN, N.; TANNER, R. I. An adaptive viscoelastic stress splitting scheme and its applications: Avss/si and avss/supg. *Journal of Non-Newtonian Fluid Mechanics*, v. 65, n. 1, p. 75 – 91, 1996. ISSN 0377-0257. Disponível em: <<http://www.sciencedirect.com/science/article/B6TGV-3V9D06J-4/2/7a3529cd2ab9672c3115c0594ed8f870>>.

- SUN, J.; SMITH, M. D.; ARMSTRONG, R. C.; BROWN, R. A. Finite element method for viscoelastic flows based on the discrete adaptive viscoelastic stress splitting and the discontinuous galerkin method: Davss-g/dg. *Journal of Non-Newtonian Fluid Mechanics*, v. 86, n. 3, p. 281 – 307, 1999. ISSN 0377-0257. Disponível em: <<http://www.sciencedirect.com/science/article/B6TGV-3XF07SM-1/2/1ce734853d4d4261e3fc5b820d6e7497>>.
- SWEBY, P. K. High resolution schemes using flux limiters for hyperbolic conservation laws. *SIAM J. Numer. Analysis*, v. 21, p. 995 – 1011, 1984.
- THIEN, N. P.; TANNER, R. I. A new constitutive equation derived from network theory. *Journal of Non-Newtonian Fluid Mechanics*, v. 2, n. 4, p. 353 – 365, 1977. ISSN 0377-0257. Disponível em: <<http://www.sciencedirect.com/science/article/B6TGV-44FMMPF-1B/2/cb7dacb8aeefad93c9916c106bbeb50b>>.
- TIRTAATMADJA, V.; SRIDHAR, T. Comparison of constitutive equations for polymer solutions in uniaxial extension. *Journal of Rheology*, SOR, v. 39, n. 6, p. 1133–1160, 1995. Disponível em: <<http://link.aip.org/link/?JOR/39/1133%-%/1>>.
- VERBEETEN, W. M. H. *Computational polymer melt rheology*. [S.l.]: CIP-Data Library Technische Universiteit Eindhoven, 2001.
- VERBEETEN, W. M. H.; PETERS, G. W. M.; BAAIJENS, F. P. T. Differential constitutive equations for polymer melts: The extended pom-pom model. *Journal of Rheology*, SOR, v. 45, n. 4, p. 823–843, 2001. Disponível em: <<http://link.aip.org/link/?JOR/45/823/1>>.
- VERBEETEN, W. M. H.; PETERS, G. W. M.; BAAIJENS, F. P. T. Viscoelastic analysis of complex polymer melt flows using the extended pom-pom model. *Journal of Non-Newtonian Fluid Mechanics*, v. 108, n. 1-3, p. 301 – 326, 2002. ISSN 0377-0257. Disponível em: <<http://www.sciencedirect.com/science/article/B6TGV-47GR4MK-3/2/f05f2c3f2c20d29a7b0e868fb2e1baee>>.
- VORST, H. A. V. D. Bi-cgstab: A fast and smoothly converging variant of bi-cg for the solution of nonsymmetric linear systems. *SIAM J. Scientific Computing*, v. 13, n. 2, p. 631–644, 1992.
- WAELE, A. de. *Oil Color Chem. Assoc. Journal*, v. 6, p. 33–88, 1923.
- WAPPEROM, P.; HULSEN, M. A. A lower bound for the invariants of the configuration tensor for some well-known differential models. *Journal of Non-Newtonian Fluid Mechanics*, v. 60, n. 2-3, p. 349 – 355, 1995. ISSN 0377-0257. Disponível em: <<http://www.sciencedirect.com/science/article/B6TGV-3YYTV2V-C/2/2a74ef72c2f7644df09ba9e778a97363>>.
- WARNER, H. *Ind. Eng. Chem. Fundam.*, v. 11, p. 379–387, 1972.
- XUE, S.-C.; TANNER, R.; PHAN-THIEN, N. Numerical study of secondary flows of viscoelastic fluid in straight pipes by an implicit finite volume method. *Journal of Non-Newtonian Fluid Mechanics*, v. 123, n. 1, p. 192, 1995.
- XUE, S.-C.; TANNER, R.; PHAN-THIEN, N. Three-dimensional numerical simulations of viscoelastic flows - predictability and accuracy. *Computer Methods in Applied Mechanics and Engineering*, v. 180, n. 1, p. 305–331, 1999.

- XUE, S.-C.; TANNER, R.; PHAN-THIEN, N. Numerical modelling of transient viscoelastic flows. *Journal of Non-Newtonian Fluid Mechanics*, v. 123, n. 1, p. 33 – 58, 2004. ISSN 0377-0257. Disponível em: <<http://www.sciencedirect.com/science/article/B6TGV-4D8MKJT-1/2/d1d159a5e3987b5d7737b20797ac0159>>.
- YANG, D. *C++ and Object-Oriented Numeric Computing for Scientists and Engineers*. Nova York: Springer, 2001.
- YASUDA, K. Tese (Doutorado) — Massachussets Institute of Technology, Cambridge, 1979.
- ZATLOUKAL, M. Differential viscoelastic constitutive equations for polymer melts in steady shear and elongational flows. *Journal of Non-Newtonian Fluid Mechanics*, v. 113, n. 2-3, p. 209 – 227, 2003. ISSN 0377-0257. Disponível em: <<http://www.sciencedirect.com/science/article/B6TGV-496FT8B-2/2/b005e5f1831d3b430c15430feb9633fe>>.
- ZHOU, Q.; AKHAVAN, R. A comparison of fene and fene-p dumbbell and chain models in turbulent flow. *Journal of Non-Newtonian Fluid Mechanics*, v. 109, n. 2-3, p. 115 – 155, 2003. ISSN 0377-0257. Disponível em: <<http://www.sciencedirect.com/science/article/B6TGV-47K2TYK-2/2/08a6ef8fa2c6e3678b49881146c83302>>.