

DESENVOLVIMENTO DE METODOLOGIAS PARA SIMULAÇÃO DE
ESCOAMENTOS POLIDISPERSOS USANDO CÓDIGO LIVRE

Luiz Fernando Lopes Rodrigues Silva

TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO
DOS PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA
UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS
REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR
EM CIÊNCIAS EM ENGENHARIA QUÍMICA.

Aprovada por:

Prof. Paulo Laranjeira da Cunha Lage, D.Sc.

Prof. José Carlos Costa da Silva Pinto, D.Sc.

Prof. Alvaro Luiz Gayoso de Azeredo Coutinho, D.Sc.

Profa. Angela Ourivio Nieckele, D.Sc.

Prof. Hrvoje Jasak, Ph.D.

RIO DE JANEIRO – RJ, BRASIL

JUNHO DE 2008

SILVA, LUIZ FERNANDO LOPES
RODRIGUES

Desenvolvimento de Metodologias para
Simulação de Escoamentos Polidispersos
usando Código Livre [Rio de Janeiro] 2008

XVII, 262 p. 29,7cm (COPPE/UFRJ,
D.Sc., Engenharia Química, 2008)

Tese – Universidade Federal do Rio de
Janeiro, COPPE

1. Processos Multifásicos
2. Balanço Populacional
3. Fluidodinâmica Computacional
4. OpenFOAM
5. DQMOM
6. Modelagem e Simulação

I. COPPE/UFRJ II. Título (série)

— À minha família —

Agradecimentos

Agradeço em primeiro lugar ao orientador e amigo Paulo Lage que, desde os meus tempos de iniciação científica, contribui fortemente na minha formação, com seus conselhos, discussões, idéias e correções.

Agradeço aos membros da banca de doutorado, Prof. Jasak, Profa. Angela, Prof. Alvaro e Prof. José Carlos, pelo apoio e sugestões na avaliação deste documento. Em especial, gostaria de agradecer ao Dr. Jasak pelos vários conselhos e pelas longas e construtivas conversas que ocorreram durante sua estada na UFRJ.

Não posso deixar de agradecer aos amigos do Laboratório de Termofluidodinâmica, Antonio, Ricardo Carvalho, João Felipe, Marcelo, Fábio e Willian. A constante troca de idéias, sugestões, piadas, etc. foram de grande valia para o ótimo clima no laboratório e, por conseguinte, a evolução deste trabalho. Aos ex-integrantes do laboratório, Cláudio Patrício, Ricardo Damian, Flávio Campos, Jorge Palomino, Victor Ruiz e Carlos Eduardo, muito obrigado pelo apoio constante.

Meus sinceros agradecimentos aos amigos e colegas do PEQ pelo ótimo convívio. Em especial, agradeço à Clarissa por todo carinho em mim depositado.

Agradeço aos amigos que trilharam o curso de graduação em engenharia química comigo e hoje fazem parte da minha vida. Aos meus amigos de longa data e aos mais recentes, muito obrigado.

Agradeço ao CNPq e à Chemtech pelo suporte financeiro.

Por fim, agradeço à minha família pelo apoio e compreensão na escolha da minha preparação profissional.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciências (D.Sc.)

DESENVOLVIMENTO DE METODOLOGIAS PARA SIMULAÇÃO DE
ESCOAMENTOS POLIDISPERSOS USANDO CÓDIGO LIVRE

Luiz Fernando Lopes Rodrigues Silva

Junho/2008

Orientador: Paulo Laranjeira da Cunha Lage

Programa: Engenharia Química

Escoamentos multifásicos polidispersos estão presentes em várias aplicações da indústria química nas quais os processos de interação partícula-partícula podem afetar o perfil de escoamento. Portanto, os efeitos de interação entre partículas, como quebra e agregação, devem ser incluídos na simulação de escoamentos multifásicos polidispersos através do acoplamento da equação de balanço populacional (EBP) com o modelo Euleriano multi-fluido. O desenvolvimento das metodologias para a simulação de escoamentos multifásicos polidispersos é apresentado neste trabalho. Uma avaliação da acurácia dos métodos numéricos para a solução da EBP foi realizada e o *Direct Quadrature Method Of Moments* se mostrou o mais apropriado. As metodologias foram implementadas no pacote CFD OpenFOAM, escrito em C++ e de código aberto. O desenvolvimento foi baseado no modelo Euleriano bifásico incompressível já implementado no OpenFOAM, estendendo-o para a abordagem multifásica ($n + 1$ fases) e considerando o acoplamento CFD-EBP. Simulações considerando efeitos de quebra e agregação de partículas foram realizados de forma a verificar as implementações.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Doctor of Science (D.Sc.)

DEVELOPMENT OF METHODOLOGIES FOR THE SIMULATION OF
POLYDISPERSE FLOWS USING OPEN SOURCE CODE

Luiz Fernando Lopes Rodrigues Silva

June/2008

Advisor: Paulo Laranjeira da Cunha Lage

Department: Chemical Engineering

Polydisperse multiphase flows appear in several chemical industry applications in which the particle-particle interactions may affect the flow patterns. Thus, the particle interactions, as breakage and aggregation, should be included in the polydisperse multiphase flow simulations through the population balance equation (PBE) and the Eulerian multi-fluid model coupling. The development of the methodologies for the simulation of polydisperse multiphase flow is presented in this work. An evaluation of the accuracy of numerical methods to solve the PBE was accomplished and the *Direct Quadrature Method Of Moments* has shown as the best choice. The methodologies were implemented in the OpenFOAM open source CFD package written in C++. The development was based on the incompressible Eulerian two-phase model already implemented in OpenFOAM, extending it to a multiphase approach ($n + 1$ phases) and considering the CFD-PBE coupling. Numerical simulations considering breakage and aggregation effects were performed in order to verify the implementations.

Sumário

Lista de Figuras	xi
Lista de Tabelas	xiv
Nomenclatura	xv
1 Introdução e Objetivos	1
1.1 Contexto e Motivação	1
1.2 Objetivos	2
1.3 Justificativa e Relevância	3
1.3.1 Histórico do Grupo de Pesquisa	4
1.3.2 Reatores Multifásicos	4
1.3.3 Por que <i>Software</i> Livre?	7
1.4 Organização do texto	9
2 Fundamentos do Escoamento Multifásico	11
2.1 Fundamentos de Modelagem	11
2.1.1 Conceitos Básicos	12
2.2 Modelagem Euleriana-Euleriana	15
2.2.1 Modelo Multi-fluido	16
2.2.2 Modelo de Mistura	17
2.3 Abordagem para o Fechamento	18
2.3.1 Fechamento das Forças de Interação	20
2.3.2 Modelagem de Turbulência	26
2.4 Análise de Escoamentos Gás-Líquido	31

2.4.1	Colunas de Borbulhamento	31
2.4.2	Simulações Eulerianas: Regime Homogêneo	33
2.4.3	Simulações Eulerianas: Regime Heterogêneo	39
3	Modelagem de Balanço Populacional	42
3.1	Balanço Populacional	42
3.1.1	Equação de Balanço Populacional (EBP)	44
3.1.2	Processos de Agregação	45
3.1.3	Processos de Quebra	48
3.1.4	Hipóteses e Simplificações	49
3.2	Técnicas Numéricas para Solução de EBPs	51
3.2.1	Métodos Estocásticos	51
3.2.2	Método dos Resíduos Ponderados	53
3.2.3	Método das Classes	55
3.2.4	Método dos Momentos	61
3.2.5	Métodos Híbridos	62
3.3	Acoplamento CFD-EBP	69
3.3.1	Fundamentos do Acoplamento	70
3.3.2	Revisão de Aplicações usando CFD-EBP	73
4	Pacote CFD — OpenFOAM	81
4.1	Introdução	81
4.1.1	Origem do OpenFOAM	82
4.2	Estrutura da Metodologia Numérica	84
4.2.1	Discretização por Volumes Finitos	84
4.2.2	Sistema Algébrico de Equações	89
4.2.3	Notação da Formulação Discreta	90
4.3	Conceitos de Programação	92
4.3.1	Orientação a Objetos e C++	93
4.3.2	Interpretação da Linguagem pelo OpenFOAM	95
4.3.3	Implementação de Códigos CFD	97

4.4	Escoamento Multifásico no OpenFOAM	100
4.4.1	Abordagens Implementadas	100
4.4.2	Modelo Euleriano Bifásico	101
5	Desenvolvimento das Metodologias	104
5.1	Introdução	104
5.2	Formulação Numérica do Modelo Bifásico	105
5.2.1	Equações Multifásicas	106
5.2.2	Equação da Pressão	109
5.2.3	Equação da Fração Volumétrica	110
5.2.4	Modelo $k - \epsilon$ para Turbulência	112
5.2.5	Implementação e Algoritmo de Solução	112
5.3	Balanco Populacional no OpenFOAM	114
5.3.1	Formulação do DQMOM	114
5.3.2	Implementação Numérica	115
5.4	Abordagem MUSIG	117
5.4.1	Implementação Numérica	120
5.5	Modelagem Multi-fluido e EBP	120
5.5.1	Equação de Conservação Multifásica	122
5.5.2	Equação Multifásica da Pressão	124
5.5.3	Equação para a Fração Volumétrica Multifásica	125
5.5.4	Acoplamento CFD-EBP Multifásico	126
5.5.5	Implementação das Abordagens	127
6	Resultados e Discussão	131
6.1	Introdução	131
6.2	Comparação e Avaliação dos Métodos Híbridos	131
6.3	Avaliação do DQMOM e Acoplamento MUSIG	135
6.3.1	Solução da EBP no OpenFOAM	137
6.3.2	Escoamento em emulsão usando o modelo MUSIG	140
6.4	Escoamento Multifásico Polidisperso	145

6.4.1	Simulações Multifásicas	146
6.4.2	Simulações de Escoamentos Polidispersos	148
7	Conclusões e Sugestões	155
7.1	Resumo do Trabalho e Conclusões	155
7.2	Sugestões para Trabalhos Futuros	161
	Referências Bibliográficas	163
A	Comparação dos Métodos de Solução para EBP	195
A.1	Informações sobre o Apêndice	195
B	Avaliação e Implementação da EBP e MUSIG	227
B.1	Informações sobre o Apêndice	227
C	Modelagem e Implementação do Modelo Polidisperso	243
C.1	Informações sobre o Apêndice	243
C.1.1	Errata	243

Lista de Figuras

2.1	Coeficiente de arrasto para esferas rígidas em função do número de Reynolds (retirado de SCHLICHTING [44]).	22
2.2	Coeficiente de sustentação para uma esfera rígida e lisa em função do número de Reynolds. Linha sólida: solução analítica para escoamento invíscido; linhas pontilhadas: ajuste da curva para os resultados numéricos com $Sr = 0,02$ (\bullet) e $Sr = 0,2$ (\blacktriangle); linhas tracejadas: solução analítica para <i>creeping flow</i> com $Sr = 0,02$ e $Sr = 0,2$ (retirado de LEGENDRE e MAGNAUDET [51]).	24
2.3	Representações dos regimes de escoamento observados em colunas de borbulhamento.	32
3.1	Representação do processo de agregação iniciado pela drenagem do filme fluido que separa as partículas, podendo resultar (a) na agregação das partículas ou (b) na separação destas.	46
3.2	Representação de como a técnica do pivô fixo lida com a formação de novas partículas que não coincidem com um pivô existente; \bullet limites das classes, $ $ pivô (ξ_i), \triangle nova partícula formada por agregação ou quebra.	57
3.3	Representação dos n grupos de partícula separados em uma malha discreta em tamanho e suas respectivas velocidades.	71
3.4	Exemplo da falta de acurácia em representar a função de distribuição usando o método das classes [166] em um caso com quebra dominante.	72

4.1	Discretização nos domínios de espaço e tempo (retirado de RUS-CHE [52]).	85
4.2	Parâmetros na discretização por volumes finitos (retirado de RUS-CHE [52]).	85
4.3	Operações da classe geometricField<Type> usando operações implícitas (fvm) e explícitas (fv).	99
5.1	Representação de um <i>template</i> PtrList<volScalarField> com n elementos. Cada seta representa um ponteiro indicando o local onde cada volScalarField está alocado na memória.	116
6.1	Erro relativo dos momentos no caso de quebra dominante para os momentos de menor ordem, $\mu_0 - \mu_1$, (a) e maior ordem, $\mu_2 - \mu_3$, (b) usando o QMOM e DQMOM, PPDC (c) e MoC (d).	133
6.2	Comportamento de falsa convergência obtido pelo PPDC com 3 e 6 pontos de quadratura em um caso de quebra e agregação equivalentes.	134
6.3	Erro global médio obtido para os casos de (a) quebra dominante e (b) agregação dominante.	135
6.4	Dimensões paramétricas e nome dos contornos da geometria BFS, onde $L = 11H$, $l = H$ e $h = H/2$ sendo $H = 0,01 m$ para o caso de quebra e $H = 0,1 m$ para agregação dominante.	140
6.5	Contorno do diâmetro de Sauter, d_s , para os casos de (a) quebra e (b) agregação dominante.	142
6.6	Convergência em malha para os casos de (a) quebra e (b) agregação usando malhas hexaédricas e tetra-prismáticas.	143
6.7	Ilustração da conversão de um grupo de elementos tetraédricos em um volume de controle poliédrico.	144
6.8	Perfis verticais de $\sum_{\alpha=1}^n r_\alpha$ e p para o Casos I, (a) e (b), e II, (c) e (d), usando o <i>solver</i> twoPhaseEulerFoam (T-F Euler) e o código multifásico com diferentes números n de fases.	147

6.9	Perfis verticais dos componentes (a) x e (b) y da velocidade da mistura para o Caso II.	148
6.10	Gráficos de contorno para as (a) frações volumétricas e (b) diâmetros característicos das fases 1 e 3 no caso de quebra dominante.	150
6.11	Gráficos de contorno para as (a) frações volumétricas e (b) diâmetros característicos das fases 1 e 3 no caso de agregação dominante.	152
6.12	Tempo computacional de simulações seriais e paralelas (Ncpu) usando os códigos para escoamento multifásico com e sem o acoplamento com balanço populacional.	153

Lista de Tabelas

4.1	Notação da discretização por volumes finitos.	91
6.1	Erros médios nas simulações transientes para os casos de (a) quebra e agregação equivalentes, (b) quebra dominante e (c) agregação dominante.	138
6.2	Erro médio (%) para as simulações 1D no casos de (b) quebra e (c) agregação dominante usando diferentes métodos de interpolação.	139
6.3	Propriedades físicas e condições de entrada para a mistura bifásica.	141

Nomenclatura

A	área projetada normal à velocidade relativa da partícula
a	freqüência de agregação entre partículas de um estado de partícula a outro
B	termo fonte da equação de balanço populacional referente à taxa de nascimento de partículas
b	taxa específica de quebra de partículas
$C_{D,\alpha}$	coeficiente de arrasto da fase α
$C_{L,\alpha}$	coeficiente de sustentação da fase α
$C_{MV,\alpha}$	coeficiente de massa virtual da fase α
D_x	coeficiente de difusão anisotrópico da partícula
D	termo fonte da equação de balanço populacional referente à taxa de morte de partículas
d_α	diâmetro característico da fase α
d_s	diâmetro médio de Sauter da fase dispersa
D	taxa de deformação do fluido
f	função densidade de número de partículas
F_i	força atuante na partícula i
\mathbf{g}	campo gravitacional
G	termo fonte da equação de balanço populacional referente à taxa de variação das propriedades da partícula
H	termo fonte geral da equação de balanço populacional
J	termo fonte da equação de balanço populacional referente à taxa de nucleação de partículas

$\mathbf{M}_{I,\alpha}$	termo fonte da equação média de <i>momentum</i> referente à troca de <i>momentum</i> pela interface da fase α
m	massa da partícula
\mathbf{n}	vetor unitário normal à uma superfície específica
n	número de pontos de quadratura
N_T	número total de partículas
P	função de densidade de probabilidade na quebra de partículas
p	pressão
r_α	fração volumétrica da fase dispersa α
Re	número de Reynolds
\mathbf{S}	vetor normal à face do volume de controle
Sr_α	taxa adimensional de cisalhamento na fase α
\mathbf{T}	tensor tensão
t	tempo
\mathbf{u}	velocidade do fluido
$\mathbf{u}_{r,\alpha}$	velocidade relativa entre a fase contínua e a dispersa α
\dot{V}	taxa de variação das variáveis externas
\mathbf{v}	vetor de variáveis internas
v	volume da partícula
w_α	peso α para as variáveis internas
\dot{X}	taxa de variação das variáveis externas
\mathbf{x}	vetor de variáveis externas
\mathbf{y}	vetor de propriedades da fase contínua
Letras Gregas	
δ	número de pares idênticos na agregação
ϵ	taxa de dissipação de energia turbulenta
ϕ_α	fluxo volumétrico da fase α pela face do volume de controle, $\phi_\alpha = (\mathbf{u}_\alpha)_f \cdot \mathbf{S}$
k	energia cinética turbulenta
λ	eficiência de agregação
μ	viscosidade do fluido

μ_k	momento k da distribuição de partículas
Ω_v	domínio das variáveis internas
Ω_x	domínio das variáveis externas
Ψ	termo de produção de energia turbulenta
ρ	massa específica
ζ_α	abscissa ponderada da fase α ($\zeta_\alpha = w_\alpha \xi_\alpha$)
τ	tensão viscosa
ϑ	número médio de partículas geradas na quebra
ϖ	freqüência de colisão para agregação
χ_R	erro relativo
ξ_α	abscissa α no domínio das variáveis internas

Subescritos

a	referente aos processos de agregação
b	referente aos processos de quebra
f	referente à interpolação para o centro da face de um volume de controle
I	referente aos processos que ocorrem na interface
m	referente à mistura no ASMM

Sobrescritos

(a)	referente à solução analítica
a	referente à força de arrasto
\sim	referente à velocidade flutuante devido a interação entre bolhas
lam	referente ao escoamento laminar
mv	referente à força de massa virtual
$\sim, '$	referente à diferentes estados de partícula
s	referente à força de sustentação
$turb$	referente ao escoamento turbulento

Siglas

CFD	Fluidodinâmica Computacional, em inglês <i>Computational Fluid Dynamics</i>
EBP	Equação de Balanço Populacional
FOAM	<i>Field Operation And Manipulation</i>

Capítulo 1

Introdução e Objetivos

1.1 Contexto e Motivação

São vastas as aplicações de sistemas polidispersos em engenharia química e uma grande variedade de processos podem ser colocados. Entre estes podem ser citados a atomização de um líquido em uma corrente de ar quente (*spray-dryer*), a injeção de gotas de óleo em uma câmara de combustão, o crescimento de microorganismos em um fermentador, a queima de partículas de carvão em uma mistura gasosa. Em especial, os processos de borbulhamento e sistemas líquido-líquido são usados para operações de esgotamento, evaporação por contato direto, destilação, reação química, etc. Em tais equipamentos de operações unitárias e reação, as diferentes interações entre as partículas dispersas afetam o comportamento dinâmico do sistema. Essas interações tem influência direta no regime de escoamento de reatores gás-líquido e no grau de mistura da fase dispersa, afetando a separação, mistura e reação nos processos [1]. Deste modo, se torna necessário considerar uma abordagem detalhada para fins de modelagem e simulação de tais sistemas.

Normalmente, espera-se que os fenômenos associados à presença das partículas nos equipamentos sejam conhecidos ou controlados, permitindo acompanhar a

influência das partículas sobre o sistema. A análise apropriada de um sistema polidisperso procura representar o comportamento estatístico da população e de seu ambiente (fase contínua) a partir de uma partícula genérica e seu ambiente local. Como o comportamento das partículas dependem de variáveis associadas aos seus ambientes locais (campo de velocidade, temperatura, etc.), a modelagem para a população de partículas deve ser mutuamente acoplada com os modelos de conservação das variáveis da fase contínua. Isto ressalta a influência e a integração dos fenômenos de transporte que ocorrem na fase contínua sobre a população inteira e vice-versa. Por exemplo, a fluidodinâmica de um equipamento pode contribuir diretamente nos efeitos de quebra e agregação de partículas, enquanto que a presença destas mesmas partículas pode interferir no escoamento do fluido.

O estudo destes fenômenos associados à sistemas polidispersos vem sendo realizado com grande afinco pela comunidade acadêmica. De fato, alguns trabalhos [2, 3] apresentam uma revisão dos estudos e aplicações de sistemas polidispersos baseados na análise por balanço populacional, colocando-o como uma abordagem promissora no tratamento destes processos.

1.2 Objetivos

O presente trabalho tem como objetivo geral a revisão, o desenvolvimento e a implementação de metodologias para a simulação de escoamentos polidispersos. Para atingir tal objetivo, algumas etapas tiveram que ser cumpridas e estão listadas a seguir.

- Estudo e revisão da modelagem multifásica e seus parâmetros.
- Estudo da equação de balanço populacional e dos métodos numéricos para sua solução.
- Avaliação crítica e comparação dos métodos de solução da equação de balanço

populacional.

- Revisão das técnicas para o acoplamento do modelo multifásico com balanço populacional.
- Estudo da ferramenta de Fluidodinâmica Computacional, CFD, visando a implementação e extensão de códigos multifásicos.
- Desenvolvimento de metodologias e algoritmos para a simulação de escoamentos polidispersos.
- Implementação das metodologias no pacote CFD.
- Avaliação e análise dos resultados simulados.

Por fim, uma ferramenta CFD gratuita e de código aberto será desenvolvida para simular escoamentos polidispersos utilizando as metodologias desenvolvidas no presente trabalho.

O desenvolvimento deste trabalho deixa para a comunidade científica uma ferramenta acessível para simulação de escoamentos polidispersos. O fato de ser desenvolvido em código aberto permite ainda a futura manipulação e possível aperfeiçoamento do código pelos pesquisadores e usuários do OpenFOAM.

1.3 Justificativa e Relevância

Nesta seção estão relatados os pontos que justificam o presente trabalho e a relevância do tema ao desenvolvimento da área de pesquisa de escoamentos multifásicos em sistemas polidispersos e sua aplicação na indústria. Além disso, os aspectos sobre o incentivo do governo brasileiro ao uso de programas de código livre são comentados nesta seção, uma vez que influenciaram na escolha do pacote CFD usado neste trabalho.

1.3.1 Histórico do Grupo de Pesquisa

A área de Termofluidodinâmica caracteriza-se como uma área de pesquisa da Engenharia Química que visa ao desenvolvimento teórico e experimental do estudo do comportamento dinâmico dos fluidos e dos fenômenos associados de transferência de calor e massa em equipamentos de separação, mistura ou reação. Em especial, as linhas de pesquisa no Laboratório de Termofluidodinâmica (LTFD) do Programa de Engenharia Química da COPPE/UFRJ incluem processos multifásicos e polidispersos, fluidodinâmica computacional e métodos numéricos. As linhas de pesquisa da área, algumas em andamento desde 1970, englobam estudos avançados que visam a elaboração de teses de mestrado e doutorado, possibilitando também o desenvolvimento de projetos científico-tecnológicos inovadores.

Os trabalhos mais recentes efetuados no LTFD incluem a análise experimental e modelagem de escoamentos gás-líquido [4, 5, 6, 7], evaporadores por contato direto em processos de borbulhamento [8, 9, 10], escoamentos multifásicos estratificados [11], escoamento em meios porosos [12, 13, 14] e escoamentos multifásicos polidispersos [15, 16, 17, 18]. O presente trabalho de doutorado foi proposto e desenvolvido em sequência natural à evolução da pesquisa do grupo referente ao campo de simulação de escoamentos multifásicos polidispersos.

1.3.2 Reatores Multifásicos

Reatores em colunas de borbulhamento são usados em uma grande variedade de processos industriais, tais como síntese de Fischer-Tropsch, produção de reagentes particulados em química fina, reações de oxidação e hidrogenação, fermentação, cultura de células e tratamento de efluentes [19, 20]. As grandes vantagens deste equipamento são a alta área interfacial entre as fases, boas taxas de transferência de massa e calor, uma alta retenção de líquido (favorecendo processos com reações lentas) e, por fim, a facilidade de construção e manutenção de uma coluna de bor-

bulhamento. O comportamento destes reatores é afetado por parâmetros globais de operação, como velocidade superficial do gás, altura do líquido, pressão e temperatura. As variáveis hidrodinâmicas que influenciam a performance da coluna de borbulhamento são a retenção gasosa, as taxas de quebra, coalescência e dispersão de bolhas, velocidade das bolhas, a distribuição de tamanho da fase dispersa, a distribuição da concentração da área interfacial entre gás-líquido, os coeficientes de troca de calor e massa entre as fases e o grau de mistura da fase líquida [21]. O entendimento da fluidodinâmica é vital para o aumento ou diminuição da escala de tais reatores. Além disso, é importante frisar que, do ponto de vista industrial, o escoamento em regime heterogêneo em colunas de bolhas é de grande importância pois garante uma alta produtividade volumétrica.

A dinâmica de escoamentos gás-líquido em colunas de borbulhamento tem sido objeto de estudo durante décadas. Existem três regimes que caracterizam o perfil de escoamento em colunas de borbulhamento e dependem da velocidade superficial do gás e do líquido, do distribuidor de gás e da geometria da coluna [22]. O primeiro, chamado de regime homogêneo, é caracterizado por baixas velocidades superficiais de gás, onde as bolhas possuem basicamente o mesmo tamanho e ascendem com velocidade constante em trajetória quase retilínea, com pouca ou nenhuma interação entre elas. Ao aumentar a velocidade superficial do gás, ocorre a transição para o regime heterogêneo. Neste caso, as interações entre as bolhas se intensificam e a faixa de distribuição de tamanho nas colunas aumenta. Por consequência, se forma um padrão de circulação interna no perfil de escoamento no líquido. O regime de golfada (usualmente chamado *slug*) caracteriza-se por altas velocidades superficiais de gás e a formação de grandes bolhas de Taylor. Este regime só é atingido em colunas com pequeno diâmetro.

Várias metodologias de simulação foram desenvolvidas e duas formas de modelagem se destacaram para tratar tais escoamentos. A primeira é a modelagem Euleriana-Lagrangeana, onde cada partícula (ou grupo de partículas) é simulada individualmente com base na segunda lei de Newton para o movimento e no cálculo

das forças atuantes sobre a mesma, considerando as interações com o fluido ou outras partículas (colisões). A abordagem Lagrangeana pode fornecer resultados precisos com uma modelagem relativamente simples. Contudo, o número de partículas representativas deve ser muito grande para que, em dado instante da simulação, existam partículas suficientes em dada região do equipamento simulado. Isto leva a uma abordagem com alto custo computacional, o que limita as simulações gás-líquido a valores de velocidade de gás e retenção gasosa relativamente baixos [23, 24]. A segunda abordagem é conhecida como Euleriana-Euleriana (modelo de dois fluidos), na qual todas as fases são tratadas como contínuas [25]. Neste caso, o escoamento é predito resolvendo-se equações de conservação médias para cada fase, onde aparecem termos de transporte entre as fases que precisam ser modelados. Devido ao seu menor custo computacional e sua boa representação média do sistema simulado, a modelagem Euleriana vem sendo muito empregada [22, 26, 27].

A grande maioria dos trabalhos de simulação de colunas de borbulhamento utiliza um único diâmetro médio de bolha em todo o equipamento para calcular termos presentes na modelagem Euleriana (por exemplo, o coeficiente de arraste). Esta hipótese é justificada para o regime homogêneo de borbulhamento onde as interações na fase dispersa tem menor importância e o tamanho das bolhas é praticamente uniforme. Contudo, no regime heterogêneo de borbulhamento, usado na grande maioria dos casos industriais, as interações entre as bolhas provocam uma ampla distribuição de tamanho de partículas no equipamento, o que invalida o uso de um único diâmetro médio. Para tratar esta situação, o uso de balanço populacional se torna necessário [28], pois permite o cálculo local da distribuição média de tamanho de partículas e pode aumentar consideravelmente a acurácia da predição do perfil de retenção gasosa no equipamento [27]. Assim, justifica-se a necessidade do acoplamento entre as equações médias Eulerianas para escoamentos multifásicos e a equação de balanço populacional. Existem várias metodologias de solução da equação de balanço populacional que apresentam diferentes resultados de acurácia e eficiência computacional. Portanto, é interessante realizar um estudo sobre as técnicas de solução da equação de balanço populacional devido ao constante surgimento

de novos métodos e aprimoramento dos já existentes.

Além da importância dos processos industriais, descritos acima, que envolvem escoamentos polidispersos este trabalho de pesquisa se justifica pela carência de ferramentas computacionais confiáveis para a simulação de tais processos.

1.3.3 Por que *Software Livre*?

Por se tratar de desenvolvimentos de códigos CFD com tecnologia de ponta, modelagem complexa e métodos numéricos avançados, um alto grau de confiabilidade só será alcançado usando ferramentas com código fonte aberto, ou seja, *software livre*.

São várias as considerações que devem ser avaliadas ao se escolher um pacote CFD como ferramenta base. Por exemplo, a facilidade de uso, a capacidade de gerar geometrias e malhas, a eficiência e robustez das técnicas numéricas implementadas e, por fim, a amplitude dos problemas físicos que o pacote pode resolver. Esses fatores dependem muito do grau de desenvolvimento e investimento inserido no pacote. Sem dúvida alguma, atualmente os pacotes CFD mais desenvolvidos são os comerciais, como CFX e FLUENT (ambos da ANSYS) e PHOENICS (da CHAM), justamente devido ao seu apelo comercial. Contudo, os preços dos pacotes comerciais somam uma quantia enorme ao usuário final - em média US\$ 25000 para uma única licença anual - fato que limita seu uso.

Contudo, na opinião do autor, uma das maiores limitações dos pacotes comerciais se refere à falta de detalhes na implementação de seus códigos. De fato, os manuais fornecidos com um pacote CFD descrevem os modelos e os métodos numéricos usados, porém não são suficientes para explicar a implementação numérica em seu código. Essas informações poderiam fornecer ao usuário maior controle sobre as capacidades do pacote e, sem dúvida, seria mais fácil evitar erros nas simulações. Desta forma, a falta do código fonte pode limitar o uso da ferramenta CFD a ponto de comprometer seus resultados. De fato, exemplo desta situação já ocorreu e está

comentada neste trabalho (Seção 3.3.1).

Existem muitos códigos CFD gratuitos disponíveis, sendo que a maioria foi desenvolvida no meio acadêmico. Contudo, estes normalmente possuem áreas de aplicação muito específicas. Certamente, uma ferramenta CFD deve fornecer ao usuário condições de analisar e implementar códigos e, ao mesmo tempo, ser geral o suficiente para ser aplicada em diversas áreas. Em conjunto com as qualidades enumeradas no início desta seção, as características próprias a uma ferramenta CFD são idealizadas enfim. Com isto em mente, o pacote de código livre OpenFOAM surge como uma opção viável e muito interessante para a simulação e desenvolvimento de códigos CFD.

Os fatores da escolha do OpenFOAM como ferramenta CFD neste trabalho estão listados a seguir:

- Código aberto e escrito em C++.
- Simplicidade no uso como uma ferramenta CFD.
- Ferramentas de geração de malha e visualização de dados incorporados ao pacote.
- Generalidade da malha (estruturada ou não-estruturada), inclusive com a possibilidade de importar malhas de outros programas (gratuitos e comerciais).
- Possibilidade de interação direta com os desenvolvedores do pacote.
- Ampla faixa de aplicações na engenharia, incluindo o escoamento multifásico.
- Multi-plataforma (Linux, Solaris, MacOS, etc).
- Solução das equações discretizadas implementadas para simulação em paralelo (SMP e DMP).
- Ferramentas de exportação de resultados para visualização em outros programas gráficos.

- Desenvolvimento tende a ser realizado pelo meio acadêmico.

Além das qualidades técnicas em se adotar um pacote CFD com código livre, o Governo do Brasil está apoiando de forma irrestrita o uso e desenvolvimento de *software* livre. Como opção estratégica, o governo federal está defendendo a democratização do conhecimento e a valorização cada vez maior da participação e inclusão digital da população. Em seu portal na internet¹ estão agrupados os conceitos e idéias em torno do *software* livre e os planos e ações do Brasil em sua adoção.

1.4 Organização do texto

Este trabalho apresenta um estudo sobre os diferentes métodos de simulação de escoamentos multifásicos e de solução do balanço populacional, de modo a propor a metodologia mais adequada a ser incorporada ao *software* de fluidodinâmica Open-FOAM. Três sub-áreas da engenharia fazem parte do tema deste trabalho: (i) modelagem de escoamentos multifásicos (termofluidodinâmica), (ii) métodos numéricos (fluidodinâmica computacional de escoamento multifásico e balanço populacional) e (iii) computação científica (implementação e desenvolvimento de *software*). De forma natural, a estrutura do trabalho se divide em tópicos relacionados a estas sub-áreas. Na seqüência, estão apresentadas a estrutura e o conteúdo dos capítulos deste trabalho.

Os fundamentos dos escoamentos multifásicos são mostrados no Capítulo 2, apresentando os conceitos básicos da sua modelagem Euleriana-Euleriana, incluindo a modelagem dos termos de interação entre as fases e da turbulência.

No Capítulo 3, a equação de balanço populacional e seus termos de interação são apresentados, definindo primeiramente os termos de quebra e coalescência de

¹Endereço do portal é: <http://www.softwarelivre.gov.br/>

forma geral e, em seguida, simplificando-os para uso em aplicações reais. Ainda neste capítulo, a revisão bibliográfica de diversas técnicas numéricas para resolver a equação de balanço populacional é feita, porém com uma ênfase maior nos métodos das classes e nos métodos dos momentos. Ao final do capítulo, se encontra uma revisão bibliográfica sobre o acoplamento da fluidodinâmica computacional às técnicas de balanço populacional para a simulação de escoamentos polidispersos.

O pacote de fluidodinâmica OpenFOAM, usado neste trabalho, é apresentado no Capítulo 4. Detalhes sobre a história e uso do *software* como ferramenta de fluidodinâmica são colocados. Uma breve introdução sobre a linguagem de programação C++ é colocada para que o leitor tenha base para acompanhar a programação interna do OpenFOAM e sua estrutura para manipulação e resolução de campos. Por fim, são colocados detalhes sobre a modelagem do escoamento multifásico no OpenFOAM, com ênfase na abordagem Euleriana.

O desenvolvimento das técnicas numéricas para simulação de escoamentos polidispersos avançaram em etapas graduais e distintas. Estas etapas estão reunidas no Capítulo 5, apresentando as particularidades sobre a formulação destas metodologias. Detalhes referentes à implementação dos códigos no OpenFOAM também estão presentes neste capítulo. Em seguida, os resultados obtidos usando as metodologias desenvolvidas no capítulo anterior são apresentados e discutidos no Capítulo 6.

Por fim, as conclusões deste trabalho e sugestões para trabalho futuro estão reunidos no Capítulo 7. Em algumas partes deste manuscrito, apenas os principais aspectos são ressaltados com referência aos trabalhos colocados nos Apêndices A, B e C, onde maiores detalhes são encontrados.

Capítulo 2

Fundamentos do Escoamento Multifásico

2.1 Fundamentos de Modelagem

Atualmente, o projeto de equipamentos na indústria química segue duas abordagens distintas. A primeira abordagem segue uma linha de modelagem de reatores com base em aspectos empíricos ou semi-empíricos devido a uma longa história de análise dimensional na indústria de equipamentos. Por outro lado, a fluidodinâmica computacional (CFD - *Computational Fluid Dynamics*), que é bastante utilizada em outros campos da ciência (como engenharia aeroespacial e automotiva), está se tornando cada vez mais interessante para os engenheiros químicos. Ambas abordagens para projeto de equipamentos ainda coexistem. Porém, com os recursos computacionais se tornando cada vez mais baratos e disponíveis, logo estas duas linhas se fundirão em uma única linha de simulação e projeto de reatores. Um primeiro passo nesse sentido foi o trabalho de BAUER e EIGENBERGER [29] que descreve a fluidodinâmica de um reator em colunas de borbulhamento por simulação CFD e inclui reações químicas e transferência de massa utilizando modelos zonais.

Para a solução CFD do escoamento multifásico, o uso de um único balanço de massa e três equações para quantidade de movimento não são suficientes para representar os campos de escoamento de todas as fases envolvidas. De fato, escoamentos multifásicos necessariamente apresentam uma fase contínua com elementos dispersos sólidos (partículas), líquidos (gotas) e/ou gás (bolhas) ou mesmo outra fase contínua (escoamento estratificado). De acordo com ISHII [30], cada tipo de escoamento pode ser classificado de acordo com a topologia da interface entre as fases em três classes distintas: escoamentos separados, mistos ou dispersos.

Em todas as abordagens propostas para solução CFD multifásica de escoamentos dispersos, as equações para a fase contínua devem ser resolvidas, existindo várias formas para tratar a fase dispersa. A modelagem multifásica vêm sendo pesquisada exaustivamente nos últimos anos e, apesar disso, ainda é necessário um grande esforço no desenvolvimento de novas técnicas e modelos mais confiáveis para a aplicação em processos existentes e no aumento de escala de equipamentos.

As próximas seções apresentam as considerações pertinentes aos modelos multifásicos existentes, com ênfase na abordagem Euleriana. Conceitos e detalhes na formulação das forças de troca de quantidade de movimento entre fases e dos modelos de turbulência também são mostrados neste capítulo.

2.1.1 Conceitos Básicos

Para a solução CFD do escoamento multifásico, duas abordagens se destacam na literatura [31]. Ambas calculam o campo de velocidade da fase contínua usando sua equação de conservação de quantidade de movimento, enquanto que o escoamento da fase dispersa é calculado de modo Lagrangeano, considerando entidades discretas (bolhas, gotas ou partículas), ou como uma fase contínua, onde todas as fases são vistas como meios interpenetrantes, na chamada abordagem Euleriana-Euleriana.

A abordagem Lagrangeana pode ser dividida de acordo com o tratamento dado

à fase dispersa.

- Modelagem Euleriana-Lagrangeana – as partículas são tratadas como fontes pontuais na modelagem do escoamento da fase contínua e o número de partículas no domínio computacional deve ser suficiente para representar o número real de partículas. Inicia-se o algoritmo de solução calculando as equações da fase contínua (Eulerianas) e, em seguida, as velocidades e posições das partículas para o próximo passo de tempo são obtidas usando a segunda lei de Newton para modelar cada partícula [32, 33]. Os modelos para o movimento e posição de cada partícula estão colocados nas Eqs. 2.1 e 2.2, onde m_i e \mathbf{u}_i representam respectivamente a massa (m_i constante) e a velocidade da partícula i e $\sum F_i$ é a soma das forças atuantes na partícula i .

$$m_i \frac{d\mathbf{u}_i}{dt} = \sum F_i \quad (2.1)$$

$$\frac{dx_i}{dt} = \mathbf{u}_i \quad (2.2)$$

Em uma terceira etapa, os efeitos das partículas sobre o campo de escoamento da fase contínua são computados em um procedimento iterativo. As interações partícula-partícula ou partícula-parede também podem ser incorporadas ao modelo. Contudo, o esforço computacional cresce drasticamente com o aumento do número de partículas simuladas. Deste modo, esta abordagem é mais usada para sistemas diluídos (com poucas partículas) ou em situações especiais, como o cálculo da distribuição do tempo de residência.

- Simulação Numérica Direta (DNS - *Direct Numerical Simulation*) – considera-se que a partícula ocupa apenas uma célula da malha em dado instante de tempo fornecendo somente uma velocidade relevante atuando na célula. Em uma abordagem mais refinada, as partículas podem ocupar mais de uma célula e diferentes velocidades são consideradas [34, 35]. Neste caso, a turbulência é resolvida diretamente sem a necessidade de adotar nenhum modelo adicional [36]. Isto leva a uma modelagem muito mais complexa para a locomoção das partículas, necessita de malhas muito mais refinadas e passos de tempo muito

pequenos resultando em um esforço computacional absurdamente alto. Como simulações DNS fornecem resultados muito precisos, esta técnica está sendo usada para verificar e validar outras formas de modelagem quando a realização de experimentos é impossível.

Pode-se notar pela descrição dos modelos acima que todas as abordagens Lagrangeanas necessitam de um alto poder de computação. Isto torna esta abordagem inapropriada para o cálculo de escoamentos multifásicos complexos e para aplicações em processos reais onde a fração volumétrica da fase dispersa é usualmente alta.

O Volume de Fluido (VOF - *Volume-of-Fluid*) é uma abordagem ainda mais refinada que as anteriores, onde as partículas fluidas (bolhas ou gotas) são consideradas deformáveis e o acompanhamento das superfícies pode ser modelado [24]. Um único campo de escoamento é considerado e a fase dispersa se distingue da contínua pela variação das propriedades do fluido através da interface. Esta classe de métodos foi implementada com sucesso para o cálculo de escoamento de frente livre em problemas de física simples [37, 11], mas ainda é proibitivo a simulação de reatores reais devido ao imenso custo computacional.

O modelo Euleriano-Euleriano usa as equações de conservação de massa e *momentum* médias para descrever os perfis dinâmicos de ambas as fases em uma escala de tamanho superior à molecular e é baseada na promediação das equações de conservação usando médias volumétricas temporal ou amostral [30, 38, 39]. Em aplicações industriais é comum que as fases se encontrem bem misturadas e o número de partículas ou gotas (ou bolhas) seja muito grande e, portanto, o processo de média se torna necessário para caracterizar a mistura. Devido à sua generalidade, este trabalho trata o modelo Euleriano-Euleriano com maior enfoque e os detalhes sobre sua modelagem são abordados nas próximas seções.

2.2 Modelagem Euleriana-Euleriana

A formulação Euleriana mais geral é o modelo multi-fluido, derivado do processo de promediação das equações de conservação locais de cada fase e considerando que os meios são interpenetrantes. Em outras palavras, as diferentes fases podem possuir a mesma posição no espaço. Nesta abordagem, o processo de média introduz a fração volumétrica, r , no conjunto de variáveis, que é definida como a probabilidade de dada fase estar presente em certo instante de tempo e ponto no espaço (média amostral). Devido à perda de informação no processo de média, termos adicionais que precisam ser modelados aparecem nas equações de conservação de cada fase. Estes termos modelam os fenômenos que ocorrem em escalas menores que a escala da média [30, 40] e devem considerar a estrutura do campo de escoamento e as propriedades físicas dos materiais em sua modelagem. De fato, estes termos podem ser derivados empiricamente, analicamente ou numericamente.

O modelo de mistura, chamado ASMM (*algebraic slip mixture model*), é essencialmente uma simplificação do modelo Euleriano multi-fluido [41]. Como o último, utiliza propriedades médias de cada fase definidas em todo o domínio e permite que as fases se movimentem com velocidades diferentes entre si. Contudo, o modelo ASMM não requer a solução das equações da continuidade e de quantidade de movimento de cada fase, mas apenas a equação para a mistura. Esta simplificação basea-se na hipótese que as fases estão em equilíbrio local. Isto implica que a partícula presente na fase dispersa assume a velocidade terminal nas condições locais correspondentes a sua posição. Por fim, uma relação deve ser proposta para a velocidade de escorregamento entre as fases (*slip*).

Na sequência, a formulação dos modelos multi-fluido e de mistura são apresentados. Na descrição da modelagem multifásica são consideradas $n + 1$ fases onde existem apenas uma fase contínua, representada pelo subscrito $\alpha = 0$, e n fases dispersas, descritas com o subscrito $\alpha = 1, \dots, n$.

2.2.1 Modelo Multi-fluido

Para sistemas multifásicos isotérmicos, as equações de conservação médias são dadas por

$$\frac{\partial(r_\alpha \rho_\alpha)}{\partial t} + \nabla \cdot (r_\alpha \rho_\alpha \mathbf{u}_\alpha) = R_\alpha \quad (2.3)$$

$$\begin{aligned} \frac{\partial(r_\alpha \rho_\alpha \mathbf{u}_\alpha)}{\partial t} + \nabla \cdot (r_\alpha \rho_\alpha \mathbf{u}_\alpha \mathbf{u}_\alpha) &= \nabla \cdot (r_\alpha \mathbf{T}_\alpha^{eff}) \\ &+ \hat{\mathbf{M}}_{I,\alpha} + R_\alpha \mathbf{u}_{\alpha,I} + \mathbf{S}_\alpha + r_\alpha \rho_\alpha \mathbf{g} \end{aligned} \quad (2.4)$$

onde ρ_α , \mathbf{u}_α , r_α e \mathbf{T}_α^{eff} representam, respectivamente, a massa específica média, a velocidade média, a fração da fase e o tensor formado pela combinação das tensões médias viscosa e turbulenta da fase α ($\mathbf{T}_\alpha^{eff} = \mathbf{T}_\alpha^{lam} + \mathbf{T}_\alpha^{turb}$), R_α é o termo fonte referente à massa trocada entre a fase α e as outras fases, \mathbf{S}_α é o termo fonte de quantidade de momento na fase α devido a forças externas além da gravidade e $\hat{\mathbf{M}}_{I,\alpha}$ é o termo de troca de *momentum* pela interface da fase α . Usualmente, os principais componentes de $\hat{\mathbf{M}}_{I,\alpha}$ são devido às forças de arrasto, sustentação e massa virtual atuando na interface das fases. A formulação do termo referente à taxa de troca de massa entre fases, R_α , depende do processo considerado (evaporação, solidificação, condensação, etc.), onde $\mathbf{u}_{\alpha,I}$ é a velocidade com a qual a massa é transferida através da interface entre as fases [40].

A tensão viscosa, também chamada de tensão média da fase, usualmente é modelada pela relação para fluidos Newtonianos [42]. A Eq. 2.5 mostra o modelo de tensão viscosa adaptado para a abordagem multi-fluido.

$$\begin{aligned} \mathbf{T}_\alpha^{lam} &= -p_\alpha \mathbf{I} + \boldsymbol{\tau}_\alpha^{lam} \\ \boldsymbol{\tau}_\alpha^{lam} &= 2\mu_\alpha \mathbf{D}_\alpha + (\kappa_\alpha - \frac{2}{3}\mu_\alpha)(\nabla \cdot \mathbf{u}_\alpha) \mathbf{I} \\ \mathbf{D}_\alpha &= \frac{1}{2} [\nabla \mathbf{u}_\alpha + (\nabla \mathbf{u}_\alpha)^t] \end{aligned} \quad (2.5)$$

Quando considerado apenas o escoamento laminar, o termo de tensão de Reynolds devido à turbulência é considerado nulo. A modelagem do fechamento para a tensão turbulenta é abordada na Seção 2.3.

Um dos pontos mais importantes na modelagem de escoamentos multifásicos é a utilização correta dos termos de troca de *momentum* $\hat{\mathbf{M}}_{I,\alpha}$ pela interface da fase α . A formulação desta expressão possui grande influência no cálculo da distribuição da fase dispersa e do campo de velocidades no escoamento. A lei de conservação de quantidade de movimento mostra que a quantidade total de *momentum* transferida entre as fases é nula, ou seja, $\sum_{\alpha=0}^n \hat{\mathbf{M}}_{I,\alpha} = 0$. Desta forma, é possível fechar o sistema de equações ao definir a modelagem de apenas n fases. Os detalhes sobre a modelagem dos termos de troca de quantidade de movimento entre as fases são abordados na Seção 2.3.

Maiores detalhes sobre a teoria de formulação e modelagem multi-fluido podem ser encontrados em diversos trabalhos [30, 39, 43, 44, 45, 40].

2.2.2 Modelo de Mistura

A equação da continuidade para a mistura está expressa na Eq. 2.6, onde ρ_m é a massa específica de mistura definida na Eq. 2.7 e \mathbf{u}_m é a velocidade de mistura média mássica mostrada na Eq. 2.8.

$$\frac{\partial \rho_m}{\partial t} + \nabla \cdot (\rho_m \mathbf{u}_m) = 0 \quad (2.6)$$

$$\rho_m = \sum_{\alpha=0}^n r_\alpha \rho_\alpha \quad (2.7)$$

$$\mathbf{u}_m = \frac{\sum_{\alpha=0}^n r_\alpha \rho_\alpha \mathbf{u}_\alpha}{\rho_m} \quad (2.8)$$

A equação de quantidade de movimento para a mistura pode ser obtida ao somar as equações individuais para a quantidade de movimento de todas as fases. A Eq. 2.9 mostra o resultado dessa operação

$$\begin{aligned} \frac{\partial(\rho_m \mathbf{u}_m)}{\partial t} + \nabla \cdot (\rho_m \mathbf{u}_m) = \\ -\nabla p + \nabla \cdot (\boldsymbol{\tau}_m + \boldsymbol{\tau}_m^{turb}) + \nabla \cdot \boldsymbol{\tau}_{Dm} + \rho_m \mathbf{g} \end{aligned} \quad (2.9)$$

onde $\boldsymbol{\tau}_m$, $\boldsymbol{\tau}_m^{turb}$ e $\boldsymbol{\tau}_{Dm}$ são as tensões viscosa, turbulenta e difusiva devido ao escorregamento entre as fases definidas respectivamente nas Eqs. 2.10, 2.11 e 2.12

$$\boldsymbol{\tau}_m = \mu_m(\nabla \mathbf{u}_m + \nabla \mathbf{u}_m^t) - \frac{2}{3}\mu_m \nabla \cdot \mathbf{u}_m \mathbf{I} \quad (2.10)$$

$$\boldsymbol{\tau}_m^{turb} = \mu_m^{turb} \left[(\nabla \mathbf{u}_m + \nabla \mathbf{u}_m^t) - \frac{2}{3}\nabla \cdot \mathbf{u}_m \mathbf{I} \right] - \frac{2}{3}\rho_m k_m \mathbf{I} \quad (2.11)$$

$$\boldsymbol{\tau}_{Dm} = \sum_{\alpha=0}^n r_\alpha \rho_\alpha \mathbf{u}_{D,\alpha} \mathbf{u}_{D,\alpha} \quad (2.12)$$

$$\mu_m = \sum_{\alpha=0}^n r_\alpha \mu_\alpha \quad (2.13)$$

sendo que, μ_m é a viscosidade de mistura definida na Eq. 2.13, k_m representa a energia cinética turbulenta de mistura e $\mathbf{u}_{D,\alpha} = \mathbf{u}_\alpha - \mathbf{u}_m$ é a velocidade difusiva (*drift*) entre a fase α e o centro de massa da mistura. O termo de tensão turbulenta pode ser fechado resolvendo um modelo de turbulência para a mistura. Note que, como a equação para a mistura é a soma das equações para as fases e a soma dos termos de interação entre as fases é nula, a Eq. 2.9 não tem esse tipo de termo.

Contudo, a velocidade das fases, \mathbf{u}_α , ainda não é conhecida, mas pode ser derivada pela relação

$$\mathbf{u}_{D,\alpha} = \mathbf{u}_\alpha - \mathbf{u}_m = \sum_{i=0}^n c_i \mathbf{u}_{slip,\alpha i} \quad (2.14)$$

onde c_i é a fração mássica da fase i e $\mathbf{u}_{slip,\alpha i}$ é a velocidade de escorregamento da fase α em relação a fase i ($\mathbf{u}_{slip,\alpha i} = \mathbf{u}_\alpha - \mathbf{u}_i$). Existem vários modelos na literatura para descrever a velocidade de escorregamento [41, 46, 47, 48, 49]. Entre estes modelos, alguns são relacionados com o diâmetro da partícula [41] ou com as forças de interação entre fases [45].

2.3 Abordagem para o Fechamento

A abordagem Euleriana-Euleriana é aplicável a todos os regimes de escoamento já que a topologia deste não precisa ser prescrita. Contudo, a formulação dos termos de troca de *momentum* entre as fases e de tensão turbulenta depende do regime de

escoamento. Como consequência, a capacidade preditiva do modelo multifásico recai fortemente na modelagem destes termos, que são as chamadas leis de fechamento.

As leis de fechamento devem sempre se guiar pelos seguintes princípios:

- Equipresença;
- Solução bem-posta;
- Indiferença ao referencial;
- Determinismo;
- Segunda lei da termodinâmica.

A equipresença significa que qualquer variável descrita por uma lei de fechamento deve ser função de todas as outras variáveis. O princípio de solução bem-posta afirma que a solução das equações do modelo existe, é única e varia continuamente com as condições de contorno impostas. Contudo, DREW e PASSMAN [39] consideram este princípio não apropriado para sistemas multifásicos. A indiferença ao referencial significa que as expressões tensoriais das leis de fechamento não dependem do referencial do observador. O determinismo confirma a predibilidade do presente a partir de dados passados. Por fim, satisfazer a segunda lei da termodinâmica garante que a solução possui um significado físico.

Para resolver as equações para o escoamento multifásico, são ainda necessárias as condições iniciais, para especificar seu estado inicial, e as condições de contorno, para especificar como o escoamento interage com a sua vizinhança. Uma vez que as leis de fechamento, as condições iniciais e condições de contorno sejam conhecidas, o sistema de equações do modelo multifásico está fechado e pronto para ser resolvido. Na seqüência, são apresentados maiores detalhes sobre o fechamento da modelagem multi-fluido para os termos de troca de *momentum* entre fases e sobre a modelagem da turbulência em escoamentos multifásicos.

2.3.1 Fechamento das Forças de Interação

O fechamento do modelo Euleriano tem foco nos fenômenos que ocorrem na interface entre as fases. A força interfacial de uma fase α normalmente é decomposta em termos de força de interação entre fases e pressão e tensão de cisalhamento interfaciais médias.

$$\hat{\mathbf{M}}_{I,\alpha} = \mathbf{M}_{I,\alpha} + p_{I,\alpha} \nabla r_\alpha - \boldsymbol{\tau}_{I,\alpha} \cdot \nabla r_\alpha \quad (2.15)$$

A partir da decomposição colocada na Eq. 2.15 combinada à Eq. 2.5, o lado direito da Eq. 2.4 (sem considerar os efeitos de turbulência) pode ser reescrito como colocado abaixo.

$$\begin{aligned} & \nabla \cdot (r_\alpha \mathbf{T}_\alpha^{lam}) + \hat{\mathbf{M}}_{I,\alpha} + R_\alpha \mathbf{u}_\alpha + \mathbf{S}_\alpha + r_\alpha \rho_\alpha \mathbf{g} = \\ & - r_\alpha \nabla p_\alpha + \nabla \cdot (r_\alpha \boldsymbol{\tau}_\alpha^{lam}) + (p_{I,\alpha} - p_\alpha) \nabla r_\alpha - \boldsymbol{\tau}_{I,\alpha} \cdot \nabla r_\alpha \\ & + \mathbf{M}_{I,\alpha} + R_\alpha \mathbf{u}_{\alpha,I} + \mathbf{S}_\alpha + r_\alpha \rho_\alpha \mathbf{g} \end{aligned} \quad (2.16)$$

Ao admitir que os efeitos de tensão interfacial não são importantes ($\boldsymbol{\tau}_{I,\alpha} = 0$), é possível considerar que a pressão possui localmente o mesmo valor para todas as fases, $p_\alpha = p_{I,\alpha} = p$.

O termo de interação entre as fases pode considerar vários tipos de forças interfaciais. Contudo, usualmente as principais forças são devido ao arrasto, sustentação e massa virtual. Outras forças, como a força de Basset, são negligenciadas na grande maioria dos casos práticos [50].

O sistema multifásico pode ser modelado a partir de uma fase primária ($\alpha = 0$), consistindo da fase contínua, e n fases secundárias, onde cada uma delas representa uma classe de partícula com volume compreendido em um intervalo especificado. A Eq. 2.17 mostra a formulação do termo de interação entre fases para a fase primária, onde \mathbf{M}^a , \mathbf{M}^s e \mathbf{M}^{mv} representam respectivamente as forças devido ao arrasto,

sustentação e massa virtual.

$$\mathbf{M}_{I,0} = \sum_{\alpha=1}^n [\mathbf{M}_{I,0\alpha}^a + \mathbf{M}_{I,0\alpha}^s + \mathbf{M}_{I,0\alpha}^{mv}] \quad (2.17)$$

Considerando que as fases secundárias não interagem entre si, hipótese razoável para sistemas diluídos ($r_0 \rightarrow 1$), a formulação do termo de interação entre fases fica na forma colocada na Eq. 2.18.

$$\begin{aligned} \mathbf{M}_{I,\alpha} &= \mathbf{M}_{I,\alpha 0}^a + \mathbf{M}_{I,\alpha 0}^s + \mathbf{M}_{I,\alpha 0}^{mv} \\ \alpha &= 1, \dots, n \end{aligned} \quad (2.18)$$

Note que, a troca de quantidade de movimento entre as fases dispersa e contínua se inter-relaciona de forma oposta, como $\mathbf{M}_{I,\alpha 0}^a = -\mathbf{M}_{I,0\alpha}^a$ para a força de arrasto. Esta mesma relação é válida para as outras forças.

A decomposição apresentada nas Eqs. 2.17 e 2.18 não são sempre válidas devido à interações não lineares entre as forças. Apesar disso, estas interações ainda não são bem conhecidas e usualmente são pequenas o suficiente para serem desprezadas. Em geral, a força de arrasto é predominante na maioria dos escoamentos bifásicos e a inclusão das demais forças pode ser avaliada pela razão destas com a força de arrasto [51]. Uma breve discussão sobre essas forças está colocada na seqüência.

Modelagem da Força de Arrasto

A força de arrasto aparece sempre que existe um movimento relativo entre a partícula e o fluido e pode ser definida como a força exercida pela fase contínua sobre a partícula na direção do escoamento. A teoria em que a força de arrasto, $\mathbf{M}_{I,\alpha}^a$, é baseada começou com os estudos de NEWTON [52] sobre a queda livre no ar de esferas com diferentes tamanhos. Newton mediu a velocidade terminal das partículas ao soltá-las do alto da catedral de St. Patrick em Londres. Com estes dados ele foi capaz de relacionar o arrasto da partícula como função do diâmetro e da velocidade terminal. Assim, Newton concluiu experimentalmente que a força de arrasto sobre

a partícula é proporcional ao quadrado da velocidade terminal. Posteriormente, a relação obtida por Newton foi estendida por ALLEN [53] para incluir os efeitos da massa específica das partículas e do fluido na força de arrasto. STOKES [54] chegou a este mesmo resultado de forma analítica ao negligenciar a convecção na equação de Navier-Stokes, corroborando com os resultados obtidos em [52].

A maioria dos resultados de estudos referentes à força de arrasto são apresentados em forma adimensional através do coeficiente de arrasto, C_D , que é função do número de Reynolds, Re , como apresentado na Fig. 2.1 [55]. A definição do número de Reynolds é mostrada na Eq. 2.19, onde ρ_0 e μ_0 são a massa específica e viscosidade da fase contínua, $\mathbf{u}_r = \mathbf{u}_0 - \mathbf{u}_p$ é a velocidade relativa e d_α é o diâmetro equivalente da partícula (diâmetro da esfera com o mesmo volume da partícula).

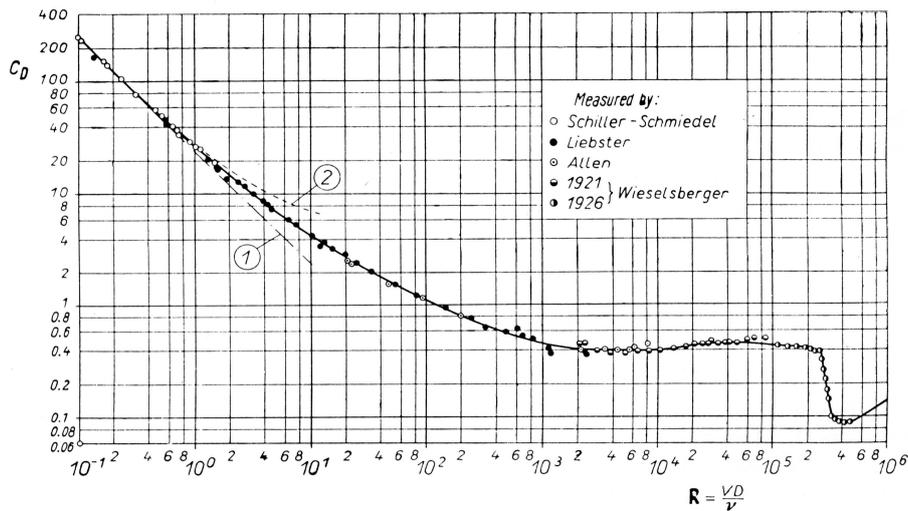


Figura 2.1: Coeficiente de arrasto para esferas rígidas em função do número de Reynolds (retirado de SCHLICHTING [44]).

$$Re = \frac{\rho_0 |\mathbf{u}_r| d_\alpha}{\mu_0} \quad (2.19)$$

A forma funcional da força de arrasto tem sido objeto de intensa pesquisa e a maioria dos escoamentos multifásicos utiliza a formulação colocada na Eq. 2.20 [45], onde $C_{D,\alpha}$ refere-se ao coeficiente de arrasto para a fase α , $\mathbf{u}_{r,\alpha} = \mathbf{u}_0 - \mathbf{u}_\alpha$ é a velocidade relativa e A_α é a área projetada normal à velocidade relativa da partícula

pertencente à fase α dividida pelo volume da mesma.

$$\mathbf{M}_{I,\alpha}^a = \frac{1}{2} \rho_0 r_\alpha A_\alpha C_{D,\alpha} |\mathbf{u}_{r,\alpha}| \mathbf{u}_{r,\alpha} \quad (2.20)$$

Sendo esferas: $A_\alpha = \frac{\pi d_\alpha^2/4}{\pi d_\alpha^3/6} = \frac{6}{d_\alpha}$.

Porém, a física de escoamento ao redor de partículas fluidas difere de duas formas em relação ao escoamento em torno de partículas rígidas. De fato, a forma da partícula não é fixa. Efeitos desse tipo devem ser considerados ao calcular o coeficiente de arrasto para bolhas e gotas [56, 51, 57].

Modelagem da Força de Sustentação

Em um escoamento com campo uniforme e estacionário, a força de sustentação refere-se à força normal à velocidade relativa do fluido. O primeiro relato sobre a ação da força de sustentação, $\mathbf{M}_{I,\alpha}^s$, atuando sobre uma partícula foi registrado por POISEUILLE [58], ao constatar que as células do sangue permaneciam afastadas das paredes dos capilares. Porém, as causas deste fenômeno só foram esclarecidas mais de um século depois com o trabalho de SEGRÉ e SILBERBERG [59]. Os dois pesquisadores estudaram o leve deslocamento radial de esferas flutuantes com diferentes tamanhos suspensas em um tubo com condições de escoamento de Poiseuille devido à atuação desta “força lateral” sobre as partículas. A partir daí, vários trabalhos sobre a força de sustentação foram realizados, em grande parte para escoamento em tubos e canais [60, 61].

As origens da força de sustentação são diversas, como a presença de um gradiente de velocidade na fase contínua, a deformação e rotação da partícula (usualmente chamado de efeito Magnus) e o efeito de surfactantes presentes no meio contínuo. Em relação às aplicações com partículas fluidas, AUTON [62] mostrou que a força de sustentação aplicada a uma bolha esférica isolada em um escoamento estacionário invíscido é proporcional ao produto vetorial da vorticidade da fase contínua ($\nabla \times \mathbf{u}_0$) e a velocidade relativa $\mathbf{u}_{r,\alpha}$ entre as fases. O autor chegou à expressão colocada

na Eq. 2.21 para representar a força de sustentação, onde $C_{L,\alpha}$ é o coeficiente de sustentação da fase α .

$$\mathbf{M}_{I,\alpha}^s = -C_{L,\alpha}\rho_0 r_\alpha \mathbf{u}_{r,\alpha} \times (\nabla \times \mathbf{u}_0) \quad (2.21)$$

A Fig. 2.2 mostra a variação do coeficiente de sustentação sobre uma esfera lisa e esférica em função do número de Reynolds [63].

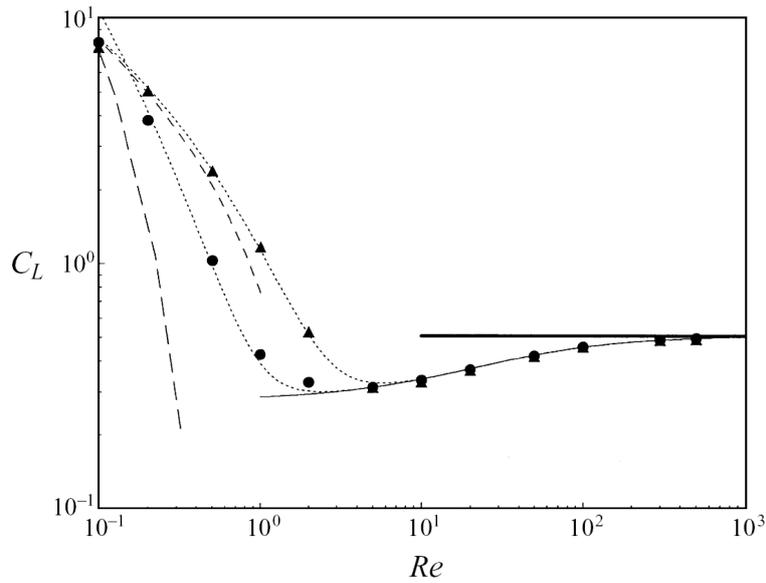


Figura 2.2: Coeficiente de sustentação para uma esfera rígida e lisa em função do número de Reynolds. Linha sólida: solução analítica para escoamento invíscido; linhas pontilhadas: ajuste da curva para os resultados numéricos com $Sr = 0,02$ (\bullet) e $Sr = 0,2$ (\blacktriangle); linhas tracejadas: solução analítica para *creeping flow* com $Sr = 0,02$ e $Sr = 0,2$ (retirado de LEGENDRE e MAGNAUDET [51]).

Pelos experimentos de LEGENDRE e MAGNAUDET [63], o coeficiente de sustentação se torna constante ($C_{L,\alpha} = 0,5$) para o caso de uma bolha esférica em um escoamento invíscido ($Re > 10^2$). Quando o número de Reynolds é inferior a 10^2 , o coeficiente de sustentação passa a ter uma forte dependência com as características do escoamento ao redor da partícula [64, 65], sendo primariamente função do Re e da taxa adimensional de cisalhamento na fase α , Sr_α , definida por:

$$Sr_\alpha = \frac{|\nabla \mathbf{u}_0| d_\alpha}{|\mathbf{u}_{r,\alpha}|} \quad (2.22)$$

Dado o grande número de fatores que influenciam a força de sustentação e as dúvidas ainda presentes no entendimento dos processos físicos envolvidos, não é de se espantar que não exista um modelo que realize previsões com acurácia satisfatória [66]. Usualmente, os limites apresentados na Fig. 2.2 são extrapolados e admite-se que o coeficiente de sustentação é constante e igual a 0,5.

Modelagem da Massa Virtual

A massa virtual de uma partícula se refere ao volume de fluido que deve ser acelerado junto com a partícula, de forma que a massa total sujeita a aceleração na equação de conservação de *momentum* da partícula seja $(\rho_\alpha + C_{MV,\alpha}\rho_0)v_\alpha$ [51], onde $C_{MV,\alpha}$ é o coeficiente de massa virtual devido à movimentação da partícula e v_α é o volume da partícula da fase dispersa α .

Considerando as teorias de escoamento invíscido, *creeping flow* e escoamento com Reynolds finito, todas levam ao valor de $C_{MV,\alpha} = 0,5$ para partículas esféricas, o que é consistente para muitos casos de partículas sólidas, bolhas e gotas. Mesmo para bolhas de ar com diâmetro acima de 3,5 mm e forma levemente elipsoidal, este coeficiente é aproximadamente válido [51].

A forma funcional da força de massa virtual está colocada na Eq. 2.23 [38].

$$\mathbf{M}_{I,\alpha}^{mv} = C_{MV,\alpha}\rho_0r_\alpha \left(\frac{D_0\mathbf{u}_0}{Dt} - \frac{D_\alpha\mathbf{u}_\alpha}{Dt} \right) \quad (2.23)$$

onde o operador $\frac{D(\cdot)}{Dt}$ é a chamada derivada material ou substantiva, definida por:

$$\frac{D(\cdot)}{Dt} = \frac{\partial(\cdot)}{\partial t} + \mathbf{u} \cdot \nabla(\cdot) \quad (2.24)$$

Contudo, DREW e LAHEY [67] concluíram que tanto a força de massa virtual (Eq. 2.23) quanto a força de sustentação (Eq. 2.21) não são indiferentes ao referencial. Porém, a sua soma é objetiva, considerando o caso em que os coeficientes são iguais ($C_{L,\alpha} = C_{MV,\alpha}$). ENWALD *et al.* [38] ressaltam que esta aproximação

é aplicável somente a fluidos com baixa vorticidade. Para este caso, a Eq. 2.25 é obtida para $C_{L,\alpha} = C_{MV,\alpha}$.

$$\mathbf{M}_{I,\alpha}^{mv} = \rho_0 r_\alpha C_{MV,\alpha} \left[\frac{D_0 \mathbf{u}_0}{Dt} - \frac{D_\alpha \mathbf{u}_\alpha}{Dt} - \mathbf{u}_{r,\alpha} \times (\nabla \times \mathbf{u}_0) \right] \quad (2.25)$$

Normalmente, os coeficientes $C_{D,\alpha}$, $C_{L,\alpha}$ e $C_{MV,\alpha}$ são determinados empiricamente com dependência das propriedades das partículas e das condições locais de escoamento. Maiores detalhes sobre a física e modelos para estes coeficientes adimensionais podem ser encontrados em CLIFT *et al.* [56], que apresenta uma extensa compilação de dados experimentais para a forma e a velocidade terminal de partículas (com maior ênfase no coeficiente de arrasto), em LOTH [51], que realizou um ótimo trabalho de revisão bibliográfica dos modelos existentes para estes coeficientes adimensionais para partículas sólidas e fluidas, e RUSCHE [66], que analisou os efeitos pertinentes às forças de arrasto e sustentação e comparou vários modelos para simular o escoamento bifásico gás-líquido com alta fração volumétrica de gás.

2.3.2 Modelagem de Turbulência

A modelagem da turbulência é de grande importância na descrição correta de escoamentos multifásicos em simulações CFD. Após vários trabalhos comparando simulações laminares e turbulentas para o escoamento gás-líquido, foi possível concluir que o modelo laminar não é apropriado para representar o perfil de escoamento, mesmo para baixos valores de velocidade e retenção gasosa [68, 47, 69, 70]. A análise destes trabalhos permite concluir que as simulações usando modelos laminares não são capazes de representar o fenômeno físico pois os resultados não convergem em malha. Os modelos de turbulência consideram os fenômenos de dissipação de energia que ocorrem na sub-escala da malha e, desta forma, representam os fenômenos que afetam o escoamento.

A modelagem da turbulência geralmente começa pela decomposição de Reynolds na qual a velocidade instantânea é dividida na soma de uma velocidade média \mathbf{u} e

uma velocidade flutuante \mathbf{u}' . Como as equações da continuidade e de balanço de *momentum* são escritas em função da velocidade instantânea, estas podem ser reescritas aplicando a média de Reynolds para uma escala de tempo Δt . Esta escala de tempo deve ser maior que a escala de tempo das flutuações de velocidade e menores que a escala de tempo do processo transiente a ser modelado. Desta forma, o termo de tensão de Reynolds ($\mathbf{T}_\alpha^{turb} = -\rho_\alpha \overline{\mathbf{u}'_\alpha \mathbf{u}'_\alpha}$) surge na equação de *momentum* a partir do processo de média, de modo a considerar os efeitos de turbulência. A modelagem da turbulência tende a focar determinados efeitos e, portanto, as equações de seus modelos podem variar de forma significativa dependendo das hipóteses assumidas em sua formulação.

Uma das abordagens para modelar a tensão de Reynolds é baseada na teoria de viscosidade de BOUSSINESQ [71] onde considera-se que, assim como as tensões viscosas em um escoamento laminar, as tensões turbulentas são proporcionais ao gradiente da velocidade média do escoamento. Baseando-se na viscosidade laminar, Boussineq introduziu o conceito de viscosidade turbulenta como uma propriedade que depende da turbulência local. Contudo, existe a necessidade de modelos para calculá-la. A Eq. 2.26 apresenta a formulação do tensor tensão turbulento, onde k é a energia cinética turbulenta e $\boldsymbol{\tau}_\alpha^{turb}$ é o tensor tensão residual de turbulência.

$$\begin{aligned}\mathbf{T}_\alpha^{turb} &= -\frac{2}{3}\rho_\alpha k \mathbf{I} + \boldsymbol{\tau}_\alpha^{turb} \\ \boldsymbol{\tau}_\alpha^{turb} &= 2\mu_\alpha^{turb} \mathbf{D}_\alpha\end{aligned}\tag{2.26}$$

A identificação dos fenômenos de turbulência em escoamentos multifásicos teve início a mais de duas décadas atrás. O trabalho de MICHELE [43] *apud* BRAUER [72] deu um dos primeiros passos na modelagem ao identificar diferentes fontes de turbulência e propôr suas formulações. Abaixo estão listadas as principais causas de geração de turbulência em escoamentos multifásicos.

- Turbulência de Reynolds — definida pelas flutuações de velocidade, como no caso monofásico. Também é válido para escoamentos multifásicos, porém outros efeitos, como a flutuação de fase, devem ser avaliados.

- Turbulência de Interface — ocorre pela diferença de tensão superficial na interface quando existe transferência de massa na última.
- Turbulência por Deformação — induzida pelo movimento da superfície das bolhas.
- Turbulência por Interação de População — devido às interações entre partículas ou bolhas em movimento [73].

Ao longo dos anos, foram desenvolvidos vários modelos para inclusão dos efeitos de turbulência em simulações CFD monofásicas [74, 75] e apenas alguns destes foram adaptados para os casos multifásicos.

Apesar de só tratar turbulência isotrópica, o modelo $k - \epsilon$ é sem dúvida a abordagem mais usada e aceita no tratamento de escoamentos turbulentos multifásicos [68, 47]. Este modelo introduz duas equações de transporte adicionais no cálculo, que são usadas para calcular a energia cinética turbulenta, k , e a taxa de dissipação de energia, ϵ . A energia cinética afeta diretamente as diferentes escalas da estrutura turbilhonar presentes no escoamento. As maiores estruturas são as portadores de maior quantidade de energia enquanto que as menores estruturas turbilhonares carregam menor quantidade de energia. Assim, a taxa de dissipação de energia ϵ deve ser resolvida de forma acoplada à energia cinética k , de modo a representar a variação do tamanho das estruturas turbilhonares. A partir de k e ϵ é possível calcular a viscosidade turbulenta local, usada nas equações de transporte de *momentum*.

A equação de conservação para a energia cinética turbulenta na fase contínua, k_0 , pode ser escrita como colocado na Eq. 2.27, onde Ψ_α é o termo de produção de energia turbulenta e $S_{0,k}$ é um termo fonte que pode ser usado para implementar efeitos de turbulência causados por outras fontes como, por exemplo, a interação entre bolhas.

$$\begin{aligned} \frac{\partial(r_0\rho_0k_0)}{\partial t} + \nabla \cdot (r_0\rho_0\mathbf{u}_0k_0) - \nabla \cdot \left[r_0 \left(\mu_0^{lam} + \frac{\mu_0^{turb}}{\sigma_\alpha} \right) \nabla k_0 \right] \\ = r_0(\Psi_\alpha - \rho_0\epsilon_0) + S_{0,k} \end{aligned} \quad (2.27)$$

A equação de conservação da dissipação de energia turbulenta ϵ_0 na fase contínua é mostrada na Eq. 2.28 e o termo fonte $S_{0,\epsilon}$ também pode ser usado para modelar outros efeitos de turbulência.

$$\begin{aligned} \frac{\partial(r_0\rho_0\epsilon_0)}{\partial t} + \nabla \cdot (r_0\rho_0\mathbf{u}_0\epsilon_0) - \nabla \cdot \left[r_0 \left(\mu_0^{lam} + \frac{\mu_0^{turb}}{\sigma_\epsilon} \right) \nabla \epsilon_0 \right] \\ = r_0 \frac{\epsilon_0}{k_0} (C_{\epsilon 1} \Psi_\alpha - C_{\epsilon 2} \rho_0 \epsilon_0) + S_{0,\epsilon} \end{aligned} \quad (2.28)$$

A viscosidade turbulenta μ_0^{turb} pode ser calculada como colocado abaixo.

$$\mu_0^{turb} = C_\mu \rho_0 \frac{k_0^2}{\epsilon_0} \quad (2.29)$$

Com isso, três incógnitas (k_0 , ϵ_0 e μ_0^{turb}) e três equações (Eqs. 2.27, 2.28 e 2.29) são adicionadas ao usar o modelo $k - \epsilon$ para considerar os efeitos de turbulência nas simulações, elevando o custo computacional, mas a um nível aceitável.

Ainda é possível incluir efeitos de turbulência causado por outras fontes, como por exemplo, a turbulência induzida pelas interações entre as partículas fluidas. Estes efeitos podem ser incluídos pela definição de velocidade flutuante devido à presença de bolhas ou introduzindo termos fontes adicionais. O modelo de Sato para turbulência induzida por bolhas [76, 77] assume que, em escoamentos gás-líquido, a velocidade do líquido pode ser decomposta em uma velocidade média ($\bar{\mathbf{u}}$), uma velocidade flutuante devido à turbulência inerente ao líquido e independente do movimento relativo entre as bolhas e o líquido (\mathbf{u}') e uma velocidade flutuante adicional devido à presença de bolhas no escoamento ($\check{\mathbf{u}}$).

$$\mathbf{u} = \bar{\mathbf{u}} + \mathbf{u}' + \check{\mathbf{u}} \quad (2.30)$$

Desta forma, a formulação de média de Reynolds para as equações de balanço inclui ainda novos termos devido aos efeitos de interação entre bolhas. Estes novos termos devem ser modelados de forma apropriada, fornecendo um componente adicional à expressão de viscosidade turbulenta. O modelo de tensão residual turbulenta induzida por bolhas $\tau_{\alpha,b}^{turb}$ [77] está colocado na Eq. 2.31

$$\tau_{\alpha,b}^{turb} = 2\mu_{\alpha,b}^{turb} \mathbf{D}_\alpha, \quad \mu_{\alpha,b}^{turb} = \Omega_b r_\alpha \rho_0 d_\alpha |\mathbf{u}_{r,\alpha}| \quad (2.31)$$

onde Ω_b é uma constante empírica, usualmente com valor de 1, 2. O modelo consiste de uma correção da turbulência pela viscosidade e é baseado no conceito de comprimento de mistura de Prandtl, onde o raio da bolha é usado como o tamanho de escala da turbulência induzida pelas bolhas.

A turbulência induzida por bolhas se mostra promissora, principalmente devido ao seu baixo custo computacional. Contudo, evidências experimentais [78] indicam que este modelo é especialmente aplicável para escoamentos com baixa retenção gasosa, onde a influência entre as duas componentes da Eq. 2.30 é pequeno, de forma que a superposição linear dos termos pode ser aplicada. Trabalhos mais recentes continuam avaliando os efeitos da presença de partículas sobre a turbulência em escoamentos multifásicos, seja por meio de análises experimentais [79] ou numéricas [80, 81].

Apenas dois trabalhos [82, 83] derivaram e estenderam o modelo de turbulência $k - \epsilon$ monofásico, desenvolvido por LAUNDER e SPALDING [74], para escoamento multifásico. Contudo, ao derivar as equações de k e ϵ , vários termos intercorrelacionados que surgem no processo de derivação foram desprezados, principalmente pela dificuldade de interpretá-los fisicamente.

O trabalho de CHAHED *et al.* [84] é o único na literatura que aplica o modelo de tensão de Reynolds para escoamentos multifásicos. Este modelo separa os efeitos da tensão turbulenta na fase contínua em duas partes: uma parte considerando a turbulência gerada pelo gradiente da velocidade média e pelos turbilhões formados pela movimentação das bolhas e outra parte pseudo-turbulenta induzida pelo deslocamento das partículas fluidas. Cada parte é determinada por uma equação de transporte. Contudo, a complexidade deste modelo traz a necessidade de se realizar mais testes de modo a verificar melhor a sua aplicabilidade.

2.4 Análise de Escoamentos Gás-Líquido

Desde o início do século XX, o estudo de colunas de borbulhamento têm atraído a atenção de vários pesquisadores. Apesar de tantos estudos na área, o entendimento completo da fluidodinâmica deste equipamento nunca foi alcançado de forma que a modelagem e o aumento de escala de reatores em colunas de borbulhamento ainda não estão bem desenvolvidos. Este fato pode ser atribuído à grande complexidade da hidrodinâmica e sua dependência com as propriedades físicas de transporte destes sistemas.

De fato, como comentado no Cap. 1, escoamentos de sistemas gás-líquido são tema de pesquisa do Laboratório de Termofluidodinâmica da COPPE/UFRJ, local onde este trabalho foi desenvolvido. As próximas seções apresentam os aspectos gerais da operação de uma coluna de borbulhamento e, na sequência, uma revisão sobre os trabalhos na literatura referentes a simulações Eulerianas deste equipamento.

2.4.1 Colunas de Borbulhamento

Colunas de borbulhamento apresentam três tipos de regime de escoamento, chamados de homogêneo, heterogêneo (*churn* ou turbulento) e em golfada (*slug*). Basicamente, para cada sistema gás-líquido, a formação e a estabilidade destes regimes dependem das velocidades superficiais do gás e do líquido, do distribuidor de gás e da geometria do reator de borbulhamento [85]. A Fig. 2.3 mostra um esquema qualitativo dos escoamentos homogêneo (bolhas dispersas), heterogêneo (*churn*) e *slug*.

Em princípio, a retenção gasosa na coluna cresce com o aumento da velocidade superficial do gás e, no caso de um distribuidor eficiente (placa porosa, distribuidor com vários orifícios, etc.), um valor máximo de retenção pode ser obtido na transição entre os regimes homogêneo e heterogêneo. Usando um distribuidor menos eficiente

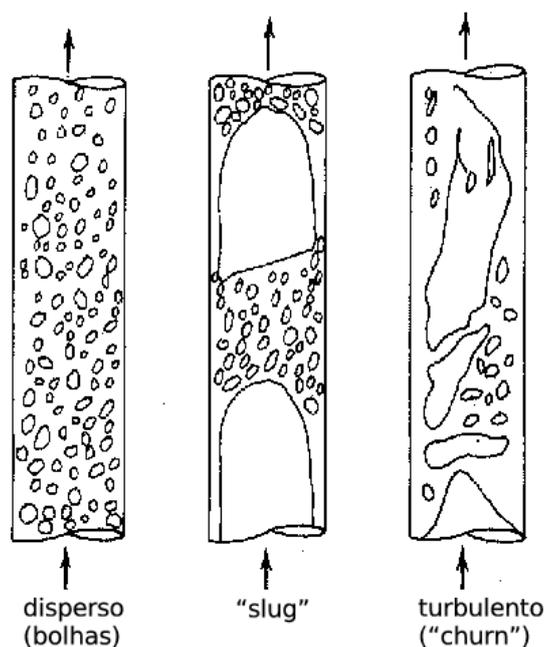


Figura 2.3: Representações dos regimes de escoamento observados em colunas de borbulhamento.

(por exemplo, distribuidor com único orifício), não se observa um ponto máximo de retenção gasosa e as características do regime heterogêneo prevalecem no sistema, mesmo para baixas velocidades superficiais de gás [86, 87].

O regime homogêneo é caracterizado por baixas velocidades superficiais de gás ($u_G < 4 \text{ cm/s}$), onde o tamanho de bolhas e a retenção gasosa são radialmente uniformes, sendo que as bolhas apresentam pequenos diâmetros e formas esféricas. Neste regime, a velocidade de ascensão das bolhas é praticamente uniforme, com trajetória ascendente quase retilínea. Além disso, as bolhas possuem pouca interação entre si. Desta forma, os fenômenos de quebra e coalescência de bolhas podem ser negligenciados, não havendo uma forte recirculação de líquido na coluna.

Com o aumento da vazão de gás, o escoamento se torna instável e o regime homogêneo não consegue se manter e passa por um estado de transição [88, 89]. A transição ocorre em uma pequena faixa de velocidades superficiais de gás, onde bolhas maiores formadas por coalescência ascendem com uma velocidade maior que a das bolhas menores. Neste regime, um padrão de circulação do líquido começa a se desenvolver. Para velocidades superficiais de gás superiores a 12 cm/s , é possível

observar uma tendência a se estabelecer um equilíbrio entre a quebra e a coalescência das bolhas. Neste ponto o regime passa a se chamar de heterogêneo e é caracterizado por bolhas com diferentes formas e tamanhos e uma intensa circulação de líquido no interior da coluna. As bolhas maiores tendem a ascender pelo centro da coluna e as pequenas pela região próxima a sua parede, mas devido à circulação de líquido na coluna as últimas podem até retornar.

Em colunas de pequeno diâmetro, o regime *slug* pode ser atingido ao aumentar a velocidade superficial do gás. Neste regime, enormes bolhas de gás, chamadas de bolhas de Taylor, ocupam quase toda a seção da coluna, sendo separadas da parede por um fino filme líquido. Normalmente, estas bolhas não se estabilizam em colunas com diâmetros maiores que 15 *cm* [86], ocorrendo a sua quebra em bolhas menores, ocorrendo, então, o regime heterogêneo.

2.4.2 Simulações Eulerianas: Regime Homogêneo

O grupo do Prof. Svendsen da Universidade de Trondheim, Noruega, foi responsável pelos primeiros trabalhos de simulação fluidodinâmica aplicada a colunas de borbulhamento na área de Engenharia Química. A evolução no desenvolvimento do modelo Euleriano bifásico (gás-líquido) pode ser acompanhada em uma série de trabalhos [90, 91, 92, 93, 94]. A modelagem considerava o arrasto e a sustentação como efeitos de interação entre as fases e utilizava o modelo de turbulência monofásico padrão $k - \epsilon$ [74]. Inicialmente, o modelo $k - \epsilon$ foi usado para ambas as fases e, posteriormente, apenas para a fase líquida, incluindo a turbulência induzida pelas bolhas de forma similar ao trabalho de SATO e SEKOGUCHI [76]. Todas as simulações foram realizadas em geometria bidimensional axialmente simétrica e em estado estacionário, o que foi o responsável pela falta de sucesso do modelo em vários aspectos.

Na mesma época, alguns trabalhos usaram modelos Eulerianos ao considerar a turbulência na fase gasosa empiricamente relacionada com a da fase líquida [95,

96], ou modelos do tipo *drift-flux*, onde há uma equação para a velocidade relativa (*slip*) entre as fases [97, 98]. Foram realizadas simulações adotando escoamento laminar [98, 99] e a modelagem dinâmica da coluna [99, 100]. De fato, comprovou-se experimentalmente que o escoamento dentro de uma coluna de borbulhamento tem comportamento caótico [101].

SOKOLICHIN e EIGENBERGER [99] e BECKER *et al.* [100] utilizaram um modelo Euleriano de dois fluidos considerando as equações de conservação de massa e *momentum* e os efeitos de arrasto, sustentação e massa virtual para interação entre fases. O modelo foi aplicado para uma coluna de seção retangular com espessura fina, permitindo considerar geometria bidimensional. Os autores realizaram, então, simulações dinâmicas com e sem considerar escoamento turbulento na fase líquida (com a turbulência dada via modelo $k - \epsilon$ padrão de LAUNDER e SPALDING [74]). A partir destes trabalhos, foi possível verificar que uma coluna de borbulhamento nunca atinge o estado estacionário e que o padrão de circulação na coluna pode ser obtido através da média temporal dos valores nas células. Determinaram ainda que o modelo $k - \epsilon$ introduz 100 vezes mais dissipação viscosa do que o necessário para o modelo simular os dados experimentais. A partir desta análise, os autores recomendaram a utilização de simulações dinâmicas laminares com a viscosidade da fase líquida ajustada aos dados experimentais.

SOKOLICHIN e EIGENBERGER [47] e BORCHERS *et al.* [102] utilizaram um modelo Euleriano para duas fases em simulações dinâmicas tridimensionais resolvendo as equações do movimento para ambas as fases, com ou sem turbulência (usando o modelo $k - \epsilon$) e a equação da continuidade para a fase gasosa com um termo de dispersão baseado na turbulência (nulo no caso laminar). A velocidade relativa entre as fases foi assumida constante. Os termos convectivos de todas as equações foram discretizados usando um esquema TVD (*Total Variation Diminishing*) de alta ordem. A partir de comparações com dados experimentais, foi possível chegar às seguintes conclusões:

- i. as simulações bidimensionais laminares não eram independentes da malha, apresentando maior detalhamento conforme a malha era refinada. Este comportamento é característico de um escoamento turbulento;
- ii. as simulações bidimensionais turbulentas não levam a soluções dinâmicas para o escoamento, contrapondo com os resultados obtidos experimentalmente [100];
- iii. o uso do modelo tridimensional turbulento fornece soluções dinâmicas independentes da malha com boa predição dos resultados experimentais. Daí concluí-se que a aproximação bidimensional era responsável pelos altos valores de dissipação viscosa obtidos pelo modelo $k - \epsilon$;
- iv. o uso do esquema *upwind* ao invés do TVD para discretizar os termos convectivos torna a solução do modelo tridimensional turbulento estacionária, eliminando o carácter dinâmico do escoamento e degradando a predibilidade da solução numérica.

SOKOLICHIN e EIGENBERGER [47] e BORCHERS *et al.* [102] recomendam o uso de um esquema de alta ordem para todos os termos convectivos, o modelo $k - \epsilon$ para a turbulência e simulações tridimensionais transientes. A necessidade do uso de simulação tridimensional transiente foi constatada também por MUDDE e SIMONIN [103] e PFLEGER *et al.* [68].

Outros aspectos na modelagem do escoamento em colunas de borbulhamento passaram a ser considerados com maior detalhamento. Por exemplo, as forças de interação entre as fases (arrasto, sustentação, massa virtual, dispersão de gás) precisam ser consideradas e como modelá-las é um fator muito importante [103, 68, 104, 70]. Outro ponto bastante analisado [105, 46, 104, 70, 106] refere-se à importância da turbulência induzida pelas bolhas e se a melhor forma de representação é através de um modelo do tipo *mixing-length* [76, 77] ou através de uma modificação do modelo $k - \epsilon$ para o escoamento bifásico [82, 83, 107].

O trabalho de PAN *et al.* [105] incluiu a modelagem da turbulência induzida pelas bolhas usando a formulação proposta por SATO *et al.* [77]. Contudo, os autores desprezaram os efeitos de turbulência induzida pelo cisalhamento, assumindo que para as baixas velocidades superficiais de gás fosse possível resolver todas as escalas do escoamento. Apesar da simulação ser bidimensional, PAN *et al.* [105] apresentaram resultados razoáveis. DEEN *et al.* [70] incluíram o modelo de SATO e SEKOGUCHI [76] para turbulência induzida pelas bolhas e compararam simulações utilizando os modelos $k-\epsilon$ e LES (*Large Eddy Simulation*) de SMAGORINSKY [108] para modelar a turbulência da fase líquida. Na abordagem por LES são resolvidas as equações dinâmicas do escoamento para obter o campo de velocidades médio e os maiores turbilhões, onde os efeitos dos menores turbilhões são modelados. Para isso, a escala da malha deve ser bem reduzida. Os resultados simulados usando a abordagem por LES foram muito superiores em relação ao $k-\epsilon$, reproduzindo muito bem os resultados experimentais apresentados por DEEN *et al.* [109]. Para o modelo $k-\epsilon$, a dissipação foi tão alta que a simulação tendeu para um estado estacionário, pois os detalhes transientes do escoamento ficaram contidos dentro do modelo de turbulência.

Os trabalhos de DEEN *et al.* [104, 70] obtiveram bons resultados ao usar o QUICK limitado para discretizar os termos convectivos das equações ao invés do esquema TVD usado em outros trabalhos [47, 68, 106]. Já MUDDE E SIMONIN [103] usaram uma discretização de terceira ordem (porém não citam detalhes), enquanto que BERTOLA *et al.* [110] utilizaram o FLUENT para testar os esquemas *Power Law*, QUICK e um método de segunda ordem. BOVE *et al.* [111] avaliaram os métodos *upwind* de primeira ordem e um método híbrido (FCT - combinação do *upwind* de primeira ordem e diferenças centrais de segunda ordem) com limitador de fluxo *superbee* na discretização dos termos convectivo. Ao comparar as simulações com os experimentos de DEEN *et al.* [70], fica clara a perda de definição dos resultados simulados usando o método de primeira ordem. Em todos os trabalhos supracitados, o efeito da difusão numérica sobre os resultados simulados é similar ao aumento da viscosidade da fase contínua. A recomendação de SOKOLICHIN

e EIGENBERGER [47] em usar simulações tridimensionais transientes se mostrou apropriada e foi adotada em diversos trabalhos [103, 68, 106, 104, 70, 111]. Desta forma, para analisar a modelagem dos escoamentos é necessário considerar a simulação tridimensional com discretização de alta ordem nos termos convectivos. Muitos trabalhos da literatura podem ser descartados usando este critério de seleção.

Nas simulações tridimensionais transientes do trabalho de MUDDE e SIMONIN [103] foi possível concluir que o termo de interação para massa virtual é essencial para modelar o período de oscilação do escoamento de uma pluma de bolhas em uma coluna de seção retangular. Como SOKOLINCHIN e EIGENBERGER [47] não incluíram este termo em sua modelagem, a predição do período de oscilação ficou comprometida.

PFLEGER *et al.* [68] levantaram a hipótese da turbulência induzida por bolhas ser importante, mas usaram apenas o modelo $k - \epsilon$ padrão. Já PFLEGER e BECKER [106] usaram um modelo $k - \epsilon$ modificado para incluir a turbulência induzida por bolhas, apresentando um impacto positivo sobre os perfis de velocidade e negativo sobre os perfis de retenção gasosa. DEEN *et al.* [70] analisaram a importância da turbulência induzida por bolhas usando o modelo de SATO e SEKOGUCHI [76] e embora os resultados fossem melhores com este termo incluído, o seu impacto era pequeno. O trabalho de BEHZADI *et al.* [112] estendeu o modelo $k - \epsilon$ de mistura proposto por GOSMAN *et al.* [113] para tratar a turbulência em sistemas com altas frações de gás. Com base em dados experimentais, os autores chegaram a uma formulação empírica para o coeficiente de resposta turbulenta C_t (razão entre as flutuações de velocidade entre as fases dispersa e contínua) em função da fração volumétrica. As simulações usando esta formulação obtiveram resultados mais acurados comparando com dados experimentais da literatura [114, 115]. De fato, a dedução fenomenológica de C_t provém das equações de transporte de mistura para k e ϵ e relaciona as viscosidades turbulentas, μ^{turb} , das fases contínua e dispersa. A Eq. 2.32 apresenta a formulação para o coeficiente de resposta turbulenta entre a

fase contínua 0 e a fase dispersa α , onde ν é a viscosidade cinemática.

$$C_t^2 = \frac{\mu_\alpha^{turb}}{\mu_0^{turb}} \left(\frac{\nu_\alpha \rho_0}{\nu_0 \rho_\alpha} \right) \quad (2.32)$$

Em trabalho recente, BECH [116] testou os modelos de comprimento de mistura de Prandtl, $k - \epsilon$ e $k - \omega$ para tratar a turbulência em uma coluna com pequena espessura, indicando o último para melhor predizer a pluma de bolhas devido ao seu tratamento a baixos números de Reynolds.

Segundo SOKOLICHIN e EIGENBERGER [47], o modelo $k - \epsilon$ fornece bons resultados para tratar a turbulência induzida pelo cisalhamento, porém DEEN *et al.* [70] não recomendam o seu uso em algumas escalas de escoamento, indicando o uso de LES. BOVE *et al.* [111] seguiram essa recomendação usando VLES (*Very Large Eddy Simulation*) em suas simulações, porém os resultados obtidos para a velocidade axial não foram satisfatórios, em parte devido a falhas na formulação de turbulência próximo à parede. Deve-se notar que os trabalhos acima estão limitados a baixas velocidades superficiais de gás (regime homogêneo).

A necessidade de usar um termo de dispersão na equação de conservação de massa de gás foi investigada por PFLEGER *et al.* [68] e DEEN *et al.* [70]. Ambos trabalhos concluíram que não é necessário incluir o termo de dispersão de gás, porém a força de sustentação é essencial para capturar os efeitos de dispersão de bolhas.

A formulação clássica para a força de arrasto é apropriada para colunas pressurizadas utilizando um fator de correção. KRISHNA e VAN BATEN [26] propuseram o uso de um fator de correção baseado na massa específica, $\rho_g/\rho_{g,1atm}$, na formulação do arrasto para bolhas grandes. Porém, a inclusão deste fator pode afetar significativamente a predição da fração de gás para altas velocidades superficiais de gás e pressões elevadas [26]. BEHZADI *et al.* [112] propuseram modificações para as expressões dos coeficientes de arrasto e sustentação em função da fração volumétrica de gás, estendendo a validade destas correlações para sistemas com alta retenção gasosa. LUCAS *et al.* [117] realizaram testes de estabilidade na equação de transporte

de *momentum* da fase líquida para avaliar os limites numéricos na modelagem do coeficiente de sustentação.

Em trabalho recente, EKAMBARA *et al.* [118] compararam as abordagens 1D, 2D e 3D na simulação de colunas de borbulhamento. Os autores realizaram simulações estacionárias em geometria cilíndrica, incluindo o modelo $k - \epsilon$ de turbulência e a transferência de *momentum* por arrasto, sustentação e massa virtual [20]. Os resultados simulados foram comparados com experimentos em regime homogêneo de borbulhamento encontrados na literatura [119, 120, 121, 122] para os perfis radiais de retenção gasosa, velocidade axial, viscosidade turbulenta e tensor de Reynolds. Todos os modelos conseguem acompanhar o perfil de velocidade de líquido, porém apenas o 3D é capaz de obter uma representação qualitativa das outras variáveis.

A questão mais importante no que concerne a modelagem de reatores em colunas de borbulhamento é que todos os trabalhos citados anteriormente nesta seção trataram apenas do regime de borbulhamento homogêneo, considerando que todas as bolhas tem o mesmo tamanho. Apenas um modelo que inclua a possibilidade de diferentes tamanhos de bolhas (e suas interações, como quebra e coalescência) pode ser usado para prever o regime de transição e simular o regime de borbulhamento heterogêneo.

2.4.3 Simulações Eulerianas: Regime Heterogêneo

De modo a tratar o regime heterogêneo, KRISHNA *et al.* [123] propuseram uma modelagem que assume a existência de duas classes de bolhas chamadas de bolhas grandes e bolhas pequenas. Esta abordagem necessita de informações sobre a distribuição de bolhas e velocidade superficial do gás, fornecidas através de correlações de cunho experimental. Isto pode parecer uma desvantagem a princípio, mas permite que este modelo seja utilizado para qualquer sistema bifásico. Este modelo teve um desenvolvimento em uma seqüência de trabalhos [123, 124, 26, 125] e suas características básicas estão colocadas na seqüência.

- A separação da velocidade superficial do gás entre as duas classes é realizado pela extensão do modelo de duas fases de leitos fluidizados a colunas de borbulhamento proposta por KRISHNA *et al.* [126] e ELLENBERGER e KRISHNA [127]. A fase líquida mais bolhas pequenas, chamada de fase densa, permanece com a sua retenção gasosa e velocidade superficial de gás do ponto de transição.
- A classe de bolhas pequenas tem um diâmetro assumido constante (usualmente 4 mm) e sua velocidade de ascensão é dada por uma correlação empírica [128, 129].
- A classe de bolhas grandes tem seu diâmetro dado por uma correlação empírica e dependente do sistema bifásico. A velocidade de ascensão vem da teoria não viscosa de DAVIES e TAYLOR [130] com correções para o efeito de parede [131], para o aumento da massa específica do gás com a pressão [26], e para efeito de população, dependente do sistema bifásico [22].
- O modelo consiste de suas equações de conservação de massa e quantidade de movimento para a fase líquida e para as duas classes de bolhas. Apenas a força de arrasto é considerada entre a fase líquida e cada classe de bolhas, usando os diâmetros e velocidades de ascensão das bolhas nos seus cálculos. Não há forças de interação entre as classes de bolhas.
- O modelo $k - \epsilon$ é usado para modelar a turbulência da fase líquida e as fases de bolhas são consideradas laminares.
- Para o regime homogêneo, a classe de bolhas grandes é eliminada. Para o regime heterogêneo em líquidos viscosos e em suspensões com alta concentração de sólidos, a classe de bolhas pequenas pode ser desconsiderada [132, 133, 22].

Este modelo foi usado inicialmente em casos 2D transientes [123, 124], mas depois verificou-se que apenas a simulação 3D transiente [26, 125] consegue prever adequadamente todos os aspectos do escoamento. A simulação 3D transiente mostrou um comportamento caótico e foi capaz de prever os perfis de retenção gasosa

e de velocidade axial da fase líquida e os coeficientes de dispersão axial de cada fase com relativo sucesso. A simulação 2D transiente pode ser usada em alguns cálculos de engenharia, mas não é adequada para a determinação dos coeficientes de dispersão axial.

O sucesso da modelagem proposta acima [123, 124, 26, 125] já indicava a tendência futura no desenvolvimento da modelagem de colunas de borbulhamento operando em regime heterogêneo. O uso de classes (limitadas a apenas duas nestes trabalhos) em sua modelagem podia ser estendida usando técnicas de balanço populacional, apresentada no Cap. 3.

Capítulo 3

Modelagem de Balanço Populacional

3.1 Balanço Populacional

Vários processos nas indústrias químicas tem a distribuição de tamanho de partículas (DTP) como um fator crucial sobre a performance do equipamento e a qualidade do produto final. Para sistemas onde houver pós, bolhas, gotas, lamas, sprays, etc., a população de partículas ao final do processo normalmente é diferente da população inicial. Isto torna necessário realizar cada vez mais estudos sobre a evolução da DTP e seus efeitos sobre o comportamento do sistema.

Normalmente, as variáveis que afetam a distribuição de partículas podem ser divididas em dois grupos, nomeados como variáveis externas e internas. As variáveis externas, \mathbf{x} , $\{\mathbf{x} \in \Omega_{\mathbf{x}}\}$, são confinadas ao campo geométrico (ou espaço físico) analisado, ou seja ao espaço de coordenadas em \mathfrak{R}^3 (no máximo), enquanto que as variáveis internas, \mathbf{v} , $\{\mathbf{v} \in \Omega_{\mathbf{v}}\}$, são as propriedades intrínsecas da partícula como tamanho, composição, energia interna, etc. Deste modo, o espaço de estado da partícula é formado pelo produto cartesiano dos espaços das variáveis internas e externas. A partir desta definição, HULBURTZ e KATZ [134] e VALENTAS e AMUNDSON [135] foram os primeiros a introduzir a equação de balanço populaci-

onal para a modelagem de processos em engenharia química envolvendo operações com fases dispersas.

Uma população de partículas em um sistema fechado pode ser afetada pela nucleação, agregação, quebra e crescimento. Estes fenômenos estão associados à forma como as partículas interagem entre si e com a fase contínua, incluindo o efeito do campo de escoamento sobre as propriedades das partículas. Isto leva a heterogeneidades espaciais e, portanto, a modelagem destes processos é um fator importante para projetos de engenharia confiáveis.

O trabalho pioneiro utilizando balanço populacional deduziu o modelo para agregação pura e aplicou-o a um sistema biológico [136, 137]. A popularidade do balanço populacional cresceu devido ao livro de RANDOLPH e LARSEN [138]. Apesar de estar basicamente voltado para cristalização, este livro expôs uma abordagem geral que pode ser aplicada em diversos casos, inclusive para processos de pulverização, coagulação de aerossóis e granulação. Em um livro mais recente, RAMKRISHNA [28] incluiu ainda os processos de crescimento de células. Seu livro estabeleceu a modelagem de balanço populacional como uma ferramenta básica para ser aplicada a todos os processos que envolvem interação entre partículas.

De fato, as áreas de aplicação para balanço populacional são diversas e muito vastas. Por exemplo, pode-se citar sistemas biológicos e ambientais [139], nanopartículas [140], processos de polimerização [141] e, por fim, aplicações à fluidodinâmica de processos polidispersos [142], foco deste trabalho.

Esta seção apresenta os fundamentos da equação de balanço populacional (EBP), fornecendo a sua formulação completa, incluindo os termos de agregação e quebra de partículas. Devido à grande complexidade da modelagem da EBP, apresentam-se algumas hipóteses simplificadoras de modo a aplicar esta técnica a casos práticos.

3.1.1 Equação de Balanço Populacional (EBP)

Neste momento, é necessário introduzir uma nova variável, $f(\mathbf{x}, \mathbf{v}, t)$, conhecida como função de densidade numérica da partícula, que quantifica o número médio de partículas no tempo t por unidade de volume no espaço de estado. A partir da função de densidade numérica, variáveis importantes para sistemas polidispersos podem ser calculadas. Por exemplo, o número total de partículas, N_T , no domínio $\Omega_{\mathbf{x}} \times \Omega_{\mathbf{v}}$ do espaço de estado de partículas (\mathbf{x}, \mathbf{v}) é dado pela Eq. 3.1.

$$N_T(t) \doteq \int_{\Omega_{\mathbf{x}}} \int_{\Omega_{\mathbf{v}}} f(\mathbf{x}, \mathbf{v}, t) dV_{\mathbf{v}} dV_{\mathbf{x}} \quad (3.1)$$

De fato, outras densidades numéricas podem ser definidas para a população. Considerando $v(\mathbf{v})$ como o volume da partícula no espaço de estado interno \mathbf{v} , a densidade de volume pode ser definida como o produto $v(\mathbf{v})f(\mathbf{x}, \mathbf{v}, t)$ e a fração volumétrica da fase dispersa incluindo todas as partículas, r , é mostrada na Eq. 3.2.

$$r(\mathbf{x}, t) \doteq \int_{\Omega_{\mathbf{v}}} v(\mathbf{v})f(\mathbf{x}, \mathbf{v}, t) dV_{\mathbf{v}} \quad (3.2)$$

Ao contrário da densidade numérica, as densidades de massa e volume se referem à quantidade de material disperso e, conseqüentemente, são variáveis mais relevantes em aplicações práticas.

A equação de balanço populacional está representada na Eq. 3.3 [28], onde $\dot{\mathbf{X}}$ é a taxa de variação da variável externa, \mathbf{D}_x é o coeficiente de difusão anisotrópico e $H(\mathbf{x}, \mathbf{v}, t)$ é o termo fonte da EBP que inclui os efeitos de nucleação, crescimento, agregação e quebra. O segundo e terceiro termos da Eq. 3.3 representam o transporte convectivo e dispersivo de partículas no espaço físico.

$$\begin{aligned} \frac{\partial f(\mathbf{x}, \mathbf{v}, t)}{\partial t} &= -\nabla_{\mathbf{x}} \cdot [\dot{\mathbf{X}}f(\mathbf{x}, \mathbf{v}, t)] + \nabla_{\mathbf{x}} \cdot [\mathbf{D}_x \cdot (\nabla_{\mathbf{x}} \cdot \mathbf{D}_x^T f(\mathbf{x}, \mathbf{v}, t))] \\ &+ H(\mathbf{x}, \mathbf{v}, t) \end{aligned} \quad (3.3)$$

O termo fonte $H(\mathbf{x}, \mathbf{v}, t)$ pode ser dividido nos seguintes termos: a taxa de nucleação $J(\mathbf{x}, \mathbf{v}, t)$, a taxa de variação das propriedades da partícula $G(\mathbf{x}, \mathbf{v}, t)$ e

as taxas de nascimento e morte, $B(\mathbf{x}, \mathbf{v}, t)$ e $D(\mathbf{x}, \mathbf{v}, t)$ respectivamente, que podem ocorrer por processos de agregação, definido pelo subscrito a , e quebra, definido pelo subscrito b .

$$\begin{aligned} H(\mathbf{x}, \mathbf{v}, t) = & B_a(\mathbf{x}, \mathbf{v}, t) - D_a(\mathbf{x}, \mathbf{v}, t) + B_b(\mathbf{x}, \mathbf{v}, t) - D_b(\mathbf{x}, \mathbf{v}, t) \\ & + J(\mathbf{x}, \mathbf{v}, t) + G(\mathbf{x}, \mathbf{v}, t) \end{aligned} \quad (3.4)$$

O termo $G(\mathbf{x}, \mathbf{v}, t)$ refere-se a mudanças no estado das variáveis internas e pode ser interpretado como o divergente do fluxo destas variáveis em seu respectivo estado. A equação geral para este termo se encontra na Eq. 3.5, sendo que $\dot{\mathbf{V}}$ representa a taxa determinística de variação da variável interna e o segundo termo da equação representa a contribuição estocástica da variação.

$$G(\mathbf{x}, \mathbf{v}, t) = -\nabla_{\mathbf{v}} \cdot [\dot{\mathbf{V}}f(\mathbf{x}, \mathbf{v}, t)] + \nabla_{\mathbf{v}} \cdot [\mathbf{D}_v \cdot (\nabla_{\mathbf{v}} \cdot \mathbf{D}_v^T f(\mathbf{x}, \mathbf{v}, t))] \quad (3.5)$$

Esta notação é útil pois permite quantificar os processos separadamente e, portanto, modelá-los de forma mais fácil e eficiente. Para um equacionamento completo da EBP, um conhecimento destes processos e de sua modelagem é essencial. Na seqüência são mostrados aspectos da física e da modelagem dos fenômenos de agregação e quebra. Detalhes sobre nucleação e crescimento de partículas serão colocados quando for apropriado.

3.1.2 Processos de Agregação

Em uma dispersão em um escoamento turbulento, as partículas se movem aleatoriamente e colidem continuamente umas com as outras. Para que a coalescência ocorra, o fluido que está entre as partículas em colisão deve ser drenado até a espessura de ruptura deste filme fluido. Durante o processo de drenagem, é possível que a flutuação turbulenta na fase contínua seja tão intensa que as partículas possam se separar por completo. Assim, nem toda colisão entre partículas necessariamente resulta em agregação. Deste modo, uma eficiência de agregação deve ser anexada ao processo

físico de colisão entre partículas. A Fig. 3.1 apresenta um diagrama simplificado do processo de agregação.

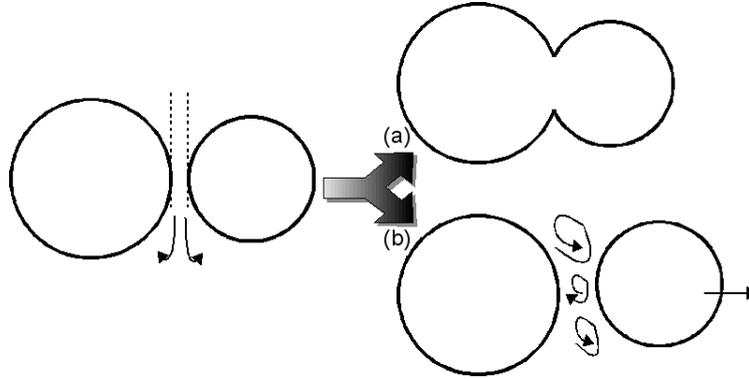


Figura 3.1: Representação do processo de agregação iniciado pela drenagem do filme fluido que separa as partículas, podendo resultar (a) na agregação das partículas ou (b) na separação destas.

A frequência de agregação a , descrita abaixo, é a probabilidade de duas partículas no estado $(\tilde{\mathbf{x}}, \tilde{\mathbf{v}})$ e $(\mathbf{x}', \mathbf{v}')$ no instante t com as propriedades da fase contínua, \mathbf{y} , se agreguem em um período de tempo de t a $t + dt$.

frequência de agregação = frequência de colisão \times eficiência de agregação

ou

$$a(\tilde{\mathbf{x}}, \tilde{\mathbf{v}}; \mathbf{x}', \mathbf{v}'; \mathbf{y}, t) = \varpi(\tilde{\mathbf{x}}, \tilde{\mathbf{v}}; \mathbf{x}', \mathbf{v}'; \mathbf{y}, t) \lambda(\tilde{\mathbf{x}}, \tilde{\mathbf{v}}; \mathbf{x}', \mathbf{v}'; \mathbf{y}, t) \quad (3.6)$$

Foram propostas diversas abordagens para modelar a frequência de agregação em diferentes sistemas polidispersos [143, 144, 145, 146]. Estes autores propuseram diferentes hipóteses e teorias para basear seus modelos e, apesar disso, todos concordam com os fenômenos físicos que afetam a agregação. A frequência de agregação é afetada principalmente por dois fatores: a frequência de colisão de um par específico de partículas e a probabilidade da agregação ocorrer após a colisão. O primeiro fator depende principalmente da distribuição de tamanho da partícula e do padrão de escoamento. O segundo fator depende dos aspectos que afetam o sucesso da agregação como a camada limite viscosa (ou a drenagem do filme do fluido), a

deformação das partículas e o balanço de energia [147].

As taxas de nascimento e morte por agregação, que fazem parte do termo fonte da EBP, mostrado na Eq. 3.4, são dadas por RAMKRISHNA [28] na forma:

$$B_a(\mathbf{x}, \mathbf{v}, \mathbf{y}, t) = \frac{1}{\delta} \int_{\Omega_{\mathbf{x}'}} \int_{\Omega_{\mathbf{v}'}} a(\tilde{\mathbf{x}}, \tilde{\mathbf{v}}; \mathbf{x}', \mathbf{v}'; \mathbf{y}, t) f(\tilde{\mathbf{x}}, \tilde{\mathbf{v}}, t) f(\mathbf{x}', \mathbf{v}', t) \frac{\partial(\tilde{\mathbf{x}}, \tilde{\mathbf{v}})}{\partial(\mathbf{x}, \mathbf{v})} dV_{\mathbf{v}'} dV_{\mathbf{x}'} \quad (3.7)$$

$$D_a(\mathbf{x}, \mathbf{v}, \mathbf{y}, t) = \int_{\Omega_{\mathbf{x}'}} \int_{\Omega_{\mathbf{v}'}} a(\mathbf{x}, \mathbf{v}; \mathbf{x}', \mathbf{v}'; \mathbf{y}, t) f(\mathbf{x}, \mathbf{v}, t) f(\mathbf{x}', \mathbf{v}', t) dV_{\mathbf{v}'} dV_{\mathbf{x}'} \quad (3.8)$$

onde δ representa o número de vezes que pares idênticos foram considerados no intervalo de integração, de forma que $1/\delta$ corrige a redundância. O termo $\partial(\tilde{\mathbf{x}}, \tilde{\mathbf{v}})/\partial(\mathbf{x}, \mathbf{v})$ corresponde ao jacobiano da transformação de coordenadas de modo que o par de partículas que colide com as coordenadas $[\tilde{\mathbf{x}}, \tilde{\mathbf{v}}]$ e $[\mathbf{x}', \mathbf{v}']$ gerem partículas no estado $[\mathbf{x}, \mathbf{v}]$, sendo dado por:

$$\frac{\partial(\tilde{\mathbf{x}}, \tilde{\mathbf{v}})}{\partial(\mathbf{x}, \mathbf{v})} = \begin{vmatrix} \frac{\partial \tilde{x}_1}{\partial x_1} & \frac{\partial \tilde{x}_1}{\partial x_2} & \frac{\partial \tilde{x}_1}{\partial x_3} & \frac{\partial \tilde{x}_1}{\partial v_1} & \dots & \frac{\partial \tilde{x}_1}{\partial v_n} \\ \frac{\partial \tilde{x}_2}{\partial x_1} & \frac{\partial \tilde{x}_2}{\partial x_2} & \frac{\partial \tilde{x}_2}{\partial x_3} & \frac{\partial \tilde{x}_2}{\partial v_1} & \dots & \frac{\partial \tilde{x}_2}{\partial v_n} \\ \frac{\partial \tilde{x}_3}{\partial x_1} & \frac{\partial \tilde{x}_3}{\partial x_2} & \frac{\partial \tilde{x}_3}{\partial x_3} & \frac{\partial \tilde{x}_3}{\partial v_1} & \dots & \frac{\partial \tilde{x}_3}{\partial v_n} \\ \frac{\partial \tilde{v}_1}{\partial x_1} & \frac{\partial \tilde{v}_1}{\partial x_2} & \frac{\partial \tilde{v}_1}{\partial x_3} & \frac{\partial \tilde{v}_1}{\partial v_1} & \dots & \frac{\partial \tilde{v}_1}{\partial v_n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial \tilde{v}_n}{\partial x_1} & \frac{\partial \tilde{v}_n}{\partial x_2} & \frac{\partial \tilde{v}_n}{\partial x_3} & \frac{\partial \tilde{v}_n}{\partial v_1} & \dots & \frac{\partial \tilde{v}_n}{\partial v_n} \end{vmatrix} \quad (3.9)$$

Vários modelos para frequência de agregação foram propostos na literatura e é imperativo escolher aquele que é apropriado para uma dada situação física. Contudo, a descrição sobre modelos de frequência de agregação não será tópico deste trabalho e detalhes sobre os mesmos serão colocados quando for necessário.

3.1.3 Processos de Quebra

Diversos trabalhos na literatura tratam dos mecanismos de quebra de partículas, concluindo que a quebra pode ocorrer devido à colisões ou influência do campo de escoamento. A primeira ocorrência é mais comum em partículas sólidas e seu mecanismo de quebra envolve o atrito na superfície da partícula, onde esta é lascada pela colisão com outras partículas, parede ou impelidor. Alguns modelos de quebra usando esse conceito foram propostos [148, 149, 150]. O segundo mecanismo se aplica a partículas fluidas, como bolhas ou gotas, e ocorre pela exposição destas a um campo de escoamento turbulento, o qual provoca oscilações de forma pela ação das forças inerciais e coesivas [151, 152, 145, 153, 154, 155]. Deste modo, a quebra por turbulência é induzida pela ação de pequenos vórtices (também chamados de turbilhões) que bombardeiam a superfície da partícula, causando deformações nesta. Isto é, a energia cinética do movimento turbulento da fase contínua provoca um aumento de energia superficial da partícula através das deformações. A fragmentação da partícula ocorre quando o movimento turbulento fornece um aumento suficiente da energia superficial para causar a quebra. O presente trabalho terá enfoque no segundo mecanismo, considerando a quebra causada pela turbulência.

As taxas de nascimento e morte por quebra na EBP são dadas por [28]:

$$B_b(\mathbf{x}, \mathbf{v}, \mathbf{y}, t) = \int_{\Omega_{\mathbf{x}'}} \int_{\Omega_{\mathbf{v}'}} \vartheta(\mathbf{x}', \mathbf{v}', \mathbf{y}, t) b(\mathbf{x}', \mathbf{v}', \mathbf{y}, t) \times P(\mathbf{x}, \mathbf{v} \mid \mathbf{x}', \mathbf{v}', \mathbf{y}, t) f(\mathbf{x}', \mathbf{v}', t) dV_{\mathbf{v}'} dV_{\mathbf{x}'} \quad (3.10)$$

$$D_b(\mathbf{x}, \mathbf{v}, \mathbf{y}, t) = b(\mathbf{x}, \mathbf{v}, \mathbf{y}, t) f(\mathbf{x}, \mathbf{v}, t) \quad (3.11)$$

onde $\vartheta(\mathbf{x}', \mathbf{v}', \mathbf{y}, t)$ é o número médio de partículas formado pela quebra da partícula de estado $(\mathbf{x}', \mathbf{v}')$ ($\vartheta \geq 2$), $P(\mathbf{x}, \mathbf{v} \mid \mathbf{x}', \mathbf{v}', \mathbf{y}, t)$ é a função de densidade de probabilidade das partículas formadas pela quebra de uma partícula de estado $(\mathbf{x}', \mathbf{v}')$ vir a possuir o estado (\mathbf{x}, \mathbf{v}) e $b(\mathbf{x}, \mathbf{v}, \mathbf{y}, t)$ é a taxa específica de quebra, ou seja, é a fração de partículas no estado (\mathbf{x}, \mathbf{v}) que quebra em uma unidade de tempo.

3.1.4 Hipóteses e Simplificações

O problema geral de balanço populacional é, certamente, de extrema complexidade, sendo vital considerar hipóteses que simplifiquem o equacionamento original mas mantendo os detalhes desejados. Algumas destas hipóteses são intrínsecas à física do problema, sendo natural o processo de simplificação. As hipóteses físicas e de modelagem apresentadas abaixo são amplamente usadas na solução da EBP e no desenvolvimento de funções de agregação e quebra [28].

Em primeiro lugar, é importante destacar as simplificações na nomenclatura das equações. As funções de quebra e agregação possuem dependência com as propriedades da fase contínua \mathbf{y} , que podem ser, por exemplo, o campo de velocidades, a dissipação de energia turbulenta ou mesmo as propriedades físicas do meio contínuo. O vetor \mathbf{y} não será mais mostrado como um parâmetro das funções, sendo considerada implícita sua dependência nas funções.

As Eqs. 3.7 e 3.8 para nascimento e morte por agregação envolvem três posições diferentes, considerando duas partículas que interagem com posições \mathbf{x}' e $\tilde{\mathbf{x}}$ e a partícula aglomerada em \mathbf{x} . Apesar de ser um tratamento matematicamente rigoroso, o mesmo é impraticável e desnecessariamente complicado para aplicações práticas de engenharia. Normalmente, considera-se que estas três posições estão muito próximas entre si durante o processo de agregação, como

$$\tilde{\mathbf{x}} \approx \mathbf{x}' \approx \mathbf{x} \quad (3.12)$$

sob a hipótese de que

$$a(\mathbf{x}, \mathbf{v}; \mathbf{x}', \mathbf{v}'; \mathbf{y}, t) = a(\mathbf{x}, \tilde{\mathbf{v}}, \mathbf{v}', t) \delta(\mathbf{x}' - \mathbf{x}) \delta(\tilde{\mathbf{x}} - \mathbf{x}). \quad (3.13)$$

Ao assumir que as três posições são próximas, as Eqs. 3.7 e 3.8 são simplificadas resultando nas Eqs. 3.14 e 3.15, respectivamente.

$$B_a(\mathbf{x}, \mathbf{v}, t) = \frac{1}{\delta} \int_{\Omega_{\mathbf{v}'}} a(\mathbf{x}, \tilde{\mathbf{v}}, \mathbf{v}', t) f(\mathbf{x}, \tilde{\mathbf{v}}, t) f(\mathbf{x}, \mathbf{v}', t) \frac{\partial(\tilde{\mathbf{v}})}{\partial(\mathbf{v})} dV_{\mathbf{v}'} \quad (3.14)$$

$$D_a(\mathbf{x}, \mathbf{v}, t) = \int_{\Omega_{\mathbf{v}'}} a(\mathbf{x}, \mathbf{v}, \mathbf{v}', t) f(\mathbf{x}, \mathbf{v}, t) f(\mathbf{x}, \mathbf{v}', t) dV_{\mathbf{v}'} \quad (3.15)$$

Considerando o problema multivariado, ou seja, um vetor de n variáveis internas, e a hipótese colocada na Eq. 3.13, o jacobiano de transformação de coordenadas pode ser simplificado como colocado na Eq. 3.16.

$$\frac{\partial(\tilde{\mathbf{v}})}{\partial(\mathbf{v})} = \begin{vmatrix} \frac{\partial \tilde{v}_1}{\partial v_1} & \dots & \frac{\partial \tilde{v}_1}{\partial v_n} \\ \vdots & \vdots & \vdots \\ \frac{\partial \tilde{v}_n}{\partial v_1} & \dots & \frac{\partial \tilde{v}_n}{\partial v_n} \end{vmatrix} \quad (3.16)$$

Note que quando o problema é monovariado, isto é, só possui uma variável interna, o jacobiano de transformação de coordenadas se torna unitário quando se trata de uma propriedade aditiva na agregação. Considere como exemplo o jacobiano para a massa m da partícula.

$$\begin{aligned} m' + \tilde{m} &= m, \\ \frac{\partial \tilde{m}}{\partial m} &= \frac{\partial(m - m')}{\partial(m)} = 1 \end{aligned} \quad (3.17)$$

Uma hipótese semelhante a mostrada na Eq. 3.13 pode ser estendida para a quebra. O processo de quebra ocorre em uma escala de tempo muito menor do que a evolução da população e, portanto, a quebra pode ser considerada instantânea. Desta maneira, as posições das partículas mãe \mathbf{x} e filha \mathbf{x}' podem ser consideradas como iguais. Esta hipótese aplicada à Eq. 3.10 para nascimento por quebra resulta na Eq. 3.18.

$$B_b(\mathbf{x}, \mathbf{v}, \mathbf{y}, t) = \int_{\Omega_{\mathbf{v}'}} \vartheta(\mathbf{x}, \mathbf{v}', t) b(\mathbf{x}, \mathbf{v}', t) P(\mathbf{x}, \mathbf{v} | \mathbf{x}, \mathbf{v}', t) f(\mathbf{x}, \mathbf{v}', t) dV_{\mathbf{v}'} \quad (3.18)$$

Por fim, a dedução dos métodos no presente trabalho lida principalmente com a EBP monovariada. Com a finalidade de apresentar a forma final da EBP utilizando o volume da partícula, v , como variável interna, a Eq. 3.19 reproduz sua versão

simplificada com os termos de quebra e coalescência, considerando quebra binária ($\vartheta(\mathbf{x}, \mathbf{v}', t) = 2$), difusão isotrópica e $\bar{\mathbf{u}}$ como a velocidade característica associada ao espaço físico ($\bar{\mathbf{u}} = \dot{\mathbf{X}}$).

$$\begin{aligned}
\frac{\partial f(\mathbf{x}, v, t)}{\partial t} &+ \nabla \cdot (\bar{\mathbf{u}}f(\mathbf{x}, v, t)) - \nabla \cdot [D_x \nabla f(\mathbf{x}, v, t)] \\
&= \frac{1}{2} \int_0^v a(\mathbf{x}, \tilde{v}, v', t) f(\mathbf{x}, \tilde{v}, t) f(\mathbf{x}, v', t) dv' \\
&- \int_0^\infty a(\mathbf{x}, v, v', t) f(\mathbf{x}, v, t) f(\mathbf{x}, v', t) dv' \\
&+ 2 \int_v^\infty b(\mathbf{x}, v', t) P(\mathbf{x}, v | \mathbf{x}, v', t) f(\mathbf{x}, v', t) dv' \\
&- b(\mathbf{x}, v, t) f(\mathbf{x}, v, t)
\end{aligned} \tag{3.19}$$

3.2 Técnicas Numéricas para Solução de EBPs

Existem várias maneiras de resolver a equação de balanço populacional e a metodologia apropriada depende da natureza do problema. A EBP é uma equação integro-diferencial que nem sempre pode ser resolvida analiticamente e, portanto uma aproximação numérica deve ser aplicada. Soluções analíticas só podem ser obtidas para as situações mais simples, usualmente não realísticas [156, 157, 158, 159, 160]. Assim, para simular casos reais, deve-se utilizar técnicas numéricas para a solução da EBP. Estas podem ser divididas em três categorias que são descritas e discutidas na seqüência.

3.2.1 Métodos Estocásticos

Os métodos estocásticos (ou de Monte Carlo) podem simular o comportamento do sistema usando técnicas de geração de números aleatórios para calcular as funções de probabilidade do sistema [28]. Deste modo, o cálculo direto das funções integrais e diferenciais é substituído pela simulação estocástica do sistema. Ao contrário

da abordagem determinística, a abordagem estocástica modela os fenômenos de nucleação, crescimento, quebra e agregação como processos discretos e aleatórios. RAMKRISHNA e BORWANKER [161, 162] introduziram a abordagem estocástica para modelar processos particulados. Estes autores mostraram que a EBP determinística é a primeira de uma série de infinitas equações derivadas da equação para a densidade de probabilidade principal. A EBP determinística calcula, de fato, a expectativa da função de distribuição e descreve o comportamento médio.

SHAH *et al.* [163] aplicou o conceito introduzido por KENDALL [164] para simular quebra e agregação em sistemas dispersos, onde os intervalos de tempo são exponencialmente distribuídos entre eventos de nascimento e morte em populações monovariadas. As taxas de quebra e agregação eram proporcionais ao número de partículas no sistema e com mecanismos dependentes do tamanho da partícula. SWEET *et al.* [165] estendeu o equacionamento descrito em SHAH *et al.* [163] incorporando reações químicas de primeira ordem para simular um leito fluidizado não isotérmico considerando apenas a coalescência das bolhas. Este trabalho usou velocidades superficiais relativamente baixas (duas vezes menor que a velocidade mínima de fluidização) pois seu objetivo foi mostrar a capacidade de simular sistemas complexos. DAS [166] desenvolveu uma solução semi-analítica para processos de quebra e aplicou com sucesso o método de Monte Carlo. DAS [166] sugere que, na ausência de uma solução analítica, seu algoritmo pode ser usado sem perda de generalidade dos resultados.

A solução da EBP foi avaliada usando diferentes algoritmos estocásticos em diferentes trabalhos [167, 168]. Os testes de comparação entre as metodologias analisaram a velocidade de simulação e a magnitude dos erros sistemáticos e estatísticos para determinar qual método é capaz de prever a solução com um dado erro em menor tempo.

Apesar de ser uma técnica extremamente flexível, a quantidade de realizações para simular adequadamente a evolução de uma população é extremamente alto, o

que eleva muito o custo computacional. Contudo, uma vantagem desta abordagem é a capacidade de simular EBPs multivariadas em respeito à variáveis internas quando outros métodos numéricos se tornam computacionalmente exaustivos [169, 170].

3.2.2 Método dos Resíduos Ponderados

O método dos resíduos ponderados é uma técnica geral para a solução de equações diferenciais parciais. Neste caso, em vez de tentar aproximar a função que é a solução da equação, aproxima-se a função a partir de funções conhecidas. A solução é dada pela expansão funcional da função de distribuição em funções de base, cujos coeficientes são calculados pela ponderação dos resíduos das equações usando funções peso. Forma-se, assim, um conjunto de equações algébricas (ou diferenciais ordinárias) linearmente independentes para obter os coeficientes da expansão da solução na base de funções usada. As formulações se diferenciam no uso de bases globais ou locais. As funções globais normalmente são usadas em problemas estacionários e quando uma forma aproximada da solução está disponível. A formulação com expansão global foi usada por BHATIA e CHAKRABORTY [171] para resolver o problema da EBP estacionária com agregação e crescimento. A técnica de elementos finitos usa bases locais e permite soluções mais gerais. Escolhas possíveis para a função peso incluem as funções delta de Dirac, resultando nos métodos de colocação, ou as próprias funções da base resultando na formulação de Galerkin.

O primeiro trabalho utilizando a técnica de elementos finitos para a EBP foi de GELBARD e SEINFELD [172]. Estes autores incorporaram funções polinomiais cúbicas em uma formulação de colocação para resolver uma EBP dinâmica com termos de agregação e crescimento de gotas em tanque agitado. O domínio da variável interna (diâmetro das gotas) foi escalonado de forma logarítmica e distribuído entre elementos de tamanhos iguais. Mais tarde, este mesmo problema foi abordado por EYRE *et al.* [173] agora usando *B-splines* como funções da base na formulação de colocação.

NICMANIS e HOUNSLOW [174] aplicaram o método dos elementos finitos para EBP estacionárias para evitar o mau condicionamento da matriz de solução em problemas com crescimento. A metodologia usada neste trabalho é sensível à análise de erros e a técnica de refinamento de malha das variáveis internas fornece um aumento significativo na acurácia da solução e na eficiência computacional. Contudo, NICMANIS e HOUNSLOW [174] não abordaram problemas dinâmicos.

LIU e CAMERON [175] resolveram a EBP dinâmica aproximando a frequência de agregação por uma série de *wavelets* em sistemas com agregação, crescimento e nucleação simultânea. O foco deste trabalho foi a predição acurada de descontinuidades, fornecendo bons resultados quando pontos suficientes de colocação (acima de 67 para o caso de frequência de agregação constante) foram usados.

MAHONEY e RAMKRISHNA [176] aplicaram o método de Galerkin em elementos finitos para sistemas com precipitação e ressaltaram as dificuldades associadas às descontinuidades no integrando. RIGOPOULOS e JONES [177] resolveram a EBP dinâmica unidimensional incluindo nucleação, crescimento, quebra e agregação usando uma formulação de colocação com elementos lineares e uma aproximação *upwind* no termo de crescimento.

Recentemente, o método dos resíduos ponderados com base nos mínimos quadrados foi aplicado com sucesso na solução da EBP [178, 179], considerando problemas estacionários puramente advectivos no espaço [180] e com dependência no tempo e espaço [181]. Esta metodologia se encontra em uma etapa avançada de desenvolvimento, sendo possível considerar o acoplamento deste método com modelos multifásicos.

O método de elementos finitos possui as vantagens de conseguir obter uma ótima convergência numérica com boa acurácia da solução e existe ainda a possibilidade de adaptar a malha ao problema. Contudo, o alto custo computacional imposto pelo cálculo das integrais duplas na formulação de Galerkin é crítico. Estes problemas aumentam quando as funções de quebra e agregação dependem de uma propriedade

associada à população, como tempo ou espaço físico. Uma revisão sobre os métodos de resíduos ponderados pode ser encontrada na literatura [28, 182].

3.2.3 Método das Classes

O método das classes ou de ordem zero, como chamado por KOSTOGLU e KARABELAS [183], se baseia em representar a distribuição da população em um número finito de classes, formado pela divisão do domínio do tamanho das partículas e adotar um valor constante de tamanho em cada classe. Desta forma é possível transformar a EBP integro-diferencial em um sistema de equações diferenciais com solução numérica bem estabelecida. De acordo com KUMAR e RAMKRISHNA [184], os métodos de ordem zero podem ser separados em duas vertentes: os internamente consistentes e as metodologias de discretização inconsistentes com respeito à propriedade integral analisada. A consistência interna implica que a discretização da EBP submetida a um operador integral é igual a forma discreta deste operador aplicada a EBP discretizada. A eficácia desta técnica decorre da obtenção rápida e acurada de propriedades selecionadas da população, como distribuição de massa, tamanho, etc., de acordo com a aplicação. Assim, o método das classes se mostra apropriado para problemas CFD e uma revisão mais detalhada sobre este assunto é descrita na seqüência.

O primeiro trabalho a usar uma discretização da EBP utilizou uma metodologia relativamente simples, empregando o teorema do valor médio. HOUNSLOW *et al.* [185] estudou a coalescência de gotas em tanques agitados usando o volume da partícula como variável interna discretizada em uma malha geométrica com fator de expansão igual a 2 ($v_{i+1} = 2v_i$). O autor desenvolveu um conjunto de equações para conservar número e massa de gotas em sistemas com agregação pura, como mostrado na Eq. 3.20, onde N_i é o número de partículas no ponto i da malha e $a_{i,j}$

é a frequência de agregação aplicada a partículas nos pontos i e j da malha.

$$\begin{aligned} \frac{dN_i}{dt} = & N_{i-1} \sum_{j=1}^{i-2} 2^{j-i+1} a_{i-1,j} N_j + \frac{1}{2} a_{i-1,i-1} N_{i-1}^2 \\ & - N_i \sum_{j=1}^{i-1} 2^{j-i} a_{i,j} N_j - N_i \sum_{j=1}^{i_{max}} a_{i,j} N_j \end{aligned} \quad (3.20)$$

A principal desvantagem deste método é a sua inflexibilidade em termos de malha, que impossibilita o refinamento das classes. Outro fator negativo é a necessidade de desenvolver novas equações para conservar propriedades da distribuição diferentes de massa e número de partículas. LISTER *et al.* [186] estenderam o método de Hounslow possibilitando o refino em progressão geométrica das classes ($v_{i+1}/v_i = 2^{1/q}$, onde q é um número inteiro maior que zero) e HILL e NG [187] desenvolveram equações similares à Eq. 3.20 para quebra com refino de malha. A flexibilidade da malha utilizando uma formulação variada do método de Hounslow foi alvo do trabalho de WYNN [188], que aplicou o método a problemas com agregação e reação. Porém, WYNN [188] comenta sobre a necessidade de realizar testes de malha para verificar se existe perda de massa no último intervalo de volume. Recentemente, ALEXOPOULOS *et al.* [189, 190] estudaram esquemas numéricos para resolver a EBP com nucleação, crescimento e agregação simultânea usando colocação ortogonal em elementos finitos e o método de Hounslow estendido por LISTER *et al.* [186]. Nos casos onde a taxa de crescimento de partículas foi alta, os autores apontaram algumas limitações na acurácia da abordagem numérica proposta por HOUNSLOW *et al.* [185] quando comparado ao método de elementos finitos.

Para driblar o problema de flexibilidade da malha, KUMAR e RAMKRISHNA [184] propuseram um método de discretização com uma abordagem mais geral e flexível devido à possibilidade de usar malhas mais grossas ou finas e com diferentes faixas de tamanho. Esta técnica concentra a população de partículas com determinada faixa de tamanho em torno de um único ponto (chamado pivô ou abscissa), fixo no espaço das variáveis internas, usando uma função delta de Dirac (δ). O subdomínio entre dois pontos da malha v_i e v_{i+1} é chamado de seção I_i e o tamanho de partícula nesta seção é representado pelo ponto na malha ξ_i (abs-

cissa) tal que $v_i < \xi_i < v_{i+1}$. Uma representação esquemática da malha pode ser analisada na Fig. 3.2. Contudo, esta abordagem possui um problema intrínseco já que os fenômenos de quebra e agregação podem levar a partículas cuja propriedade não pertence a nenhum dos pivôs representativos da malha, gerando problemas de inacurácia numérica da solução.

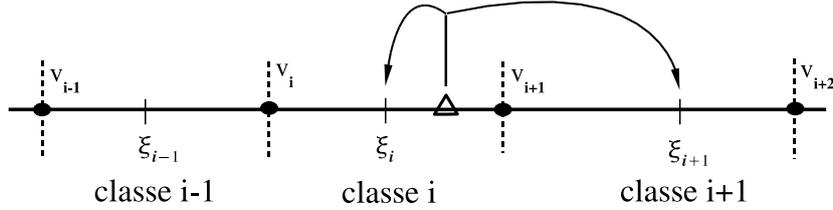


Figura 3.2: Representação de como a técnica do pivô fixo lida com a formação de novas partículas que não coincidem com um pivô existente; • limites das classes, | pivô (ξ_i), Δ nova partícula formada por agregação ou quebra.

O conjunto de equações necessárias para conservar o número e a massa das partículas usando a técnica do pivô fixo está colocado abaixo, onde $\delta'_{j,k}$ representa o delta de Kronecker e n é o número de pontos discretizados na malha.

$$\begin{aligned} \frac{dN_i}{dt} = & \sum_{\substack{j \geq k \\ \xi_{i-1} \leq (\xi_j + \xi_k) \leq \xi_{i+1}}} [1 - \frac{1}{2} \delta'_{j,k}] \eta_i a(\xi_j, \xi_k) N_j N_k \\ & - N_i \sum_k^n a(\xi_i, \xi_k) N_k + \sum_{j \geq k}^n \kappa_{i,k} b(\xi_i) N_j - b(\xi_i) N_i \end{aligned} \quad (3.21)$$

onde

$$\eta_i = \begin{cases} \frac{\xi_{i+1} - (\xi_j + \xi_k)}{\xi_{i+1} - \xi_i} & \text{quando } \xi_i \leq (\xi_j + \xi_k) \leq \xi_{i+1} \\ \frac{(\xi_j + \xi_k) - \xi_{i-1}}{\xi_i - \xi_{i-1}} & \text{quando } \xi_{i-1} \leq (\xi_j + \xi_k) \leq \xi_i \end{cases} \quad (3.22)$$

e

$$\kappa_{i,k} = \int_{\xi_i}^{\xi_{i+1}} \frac{\xi_{i+1} - v}{\xi_{i+1} - \xi_i} P(v | \xi_k) dv + \int_{\xi_{i-1}}^{\xi_i} \frac{v - \xi_{i-1}}{\xi_i - \xi_{i-1}} P(v | \xi_k) dv \quad (3.23)$$

Vale ressaltar que, quando uma malha discretizada em volume com fator geométrico igual a 2 é usada, as Eqs. 3.21, 3.22 e 3.23 levam a um equacionamento idêntico ao descrito em HOUNSLOW *et al.* [185].

KUMAR *et al.* [191] estenderam a metodologia do pivô fixo [184] ao considerar uma nova abordagem para a redistribuição de partículas geradas por agregação na malha. Este método considera a média do volume das partículas para definir o local onde a nova partícula gerada será alocada. Apesar de melhorar a acurácia e possibilitar o uso de malhas mais gerais em processos de agregação pura, o método é limitado pois usa a formulação original [184] para tratar a quebra. Recentemente, este método foi estendido para casos multivariados nas propriedades internas da partícula [192].

VANNI [193] realizou um extenso estudo comparando a acurácia, facilidade de implementação e velocidade de cálculo de diversos métodos de solução para EBP envolvendo quebra e agregação. Os métodos foram divididos devido à flexibilidade na discretização: (i) métodos baseados em discretização fixa [194, 185], (ii) métodos que permitem refinamento de malha [195, 196, 186, 184, 197]. Os testes de comparação adotaram diferentes combinações de funções de quebra e agregação montando um total de 10 casos. Os métodos propostos por GELBARD *et al.* [195] e o pivô fixo de KUMAR e RAMKRISHNA [184] se mostraram mais robustos e com maior versatilidade em relação aos outros métodos. Enquanto a implementação do método de GELBARD *et al.* [195] é complexa e custosa, a técnica pivô fixo é mais simples e apresentou maior acurácia nos casos simulados.

KUMAR e RAMKRISHNA [198] propuseram uma nova abordagem, chamada de método do pivô móvel, para superar a falta de acurácia nas predições de gradientes altamente não lineares presentes na distribuição. Esta técnica permite que a não uniformidade da distribuição em cada seção I_i , provocada pelos efeitos de quebra ou agregação, sejam considerados mudando a localização do pivô. Em outras palavras, agora os pivôs são quantidades dinâmicas e suas posições mudam conforme a distribuição varia. Acoplado à equação de conservação do número de partículas, é necessário deduzir as equações para as mudanças de posição dos pivôs, que se movimentam de forma a compensar a evolução da função de distribuição devido as novas partículas formadas. As equações que conservam massa e número pela técnica

do pivô móvel estão colocadas abaixo.

$$\begin{aligned} \frac{dN_i}{dt} = & \sum_{\substack{j \geq k \\ v_i \leq (\xi_j + \xi_k) \leq v_{i+1}}} \left[1 - \frac{1}{2} \delta'_{j,k} \right] a(\xi_j, \xi_k) N_j N_k \\ & - N_i \sum_{k=1}^{i_{max}} a(\xi_i, \xi_k) N_k + \sum_{j \geq i}^{i_{max}} b(\xi_i) N_j \bar{B}_{i,j}^{(1)} - b(\xi_i) N_i \end{aligned} \quad (3.24)$$

$$\begin{aligned} \frac{d\xi_i}{dt} = & \frac{1}{N_i} \sum_{\substack{j \geq k \\ v_i \leq (\xi_j + \xi_k) \leq v_{i+1}}} \left[1 - \frac{1}{2} \delta'_{j,k} \right] [(\xi_j + \xi_k) - \xi_i] a(\xi_j, \xi_k) N_j N_k \\ & - \frac{1}{N_i} \sum_{j \geq i}^{i_{max}} b(\xi_i) N_j [\bar{B}_{i,j}^{(v)} - \xi_i \bar{B}_{i,j}^{(1)}] \end{aligned} \quad (3.25)$$

onde

$$\begin{aligned} \bar{B}_{i,j}^{(1)} &= \int_{v_i}^{v_{i+1}} P(v | \xi_j) dv \\ \bar{B}_{i,j}^{(v)} &= \int_{v_i}^{v_{i+1}} v P(v | \xi_j) dv \end{aligned} \quad (3.26)$$

Apesar da alta acurácia do método, este tem alto custo computacional quando a variação da distribuição no espaço físico é considerada. ATTARAKI *et al.* [199] propuseram dois algoritmos para otimizar a localização dos pivôs na malha e minimizar o erro da discretização da EBP aplicada a sistemas de tanques agitados em batelada ou contínuos. Estes autores usaram a técnica do pivô móvel e mostraram que as equações de densidade numérica e as equações do pivô (Eqs. 3.24 e 3.25) poderiam ser resolvidas seqüencialmente no tempo quando o número de classes fosse suficientemente alto.

NOPENS *et al.* [200] realizaram simulações para comparar a velocidade de cálculo dos métodos de Hounslow, pivô fixo e pivô móvel em processos de agregação ou quebra pura e agregação e quebra simultânea usando diferentes condições iniciais. Contudo, a simplicidade das funções que representam estes fenômenos são questionáveis para serem usados na avaliação da acurácia dos métodos. Os autores não realizaram comparações dos resultados numéricos simulados com soluções analíticas. Para comparar as metodologias de solução, NOPENS *et al.* [200] usaram três

tamanhos de malha com 25 (grosseira), 31 e 46 (refinada) classes e mediram o tempo computacional até que a simulação atingisse o estado estacionário. Os autores destacaram a dependência dos resultados com o método numérico utilizado e o grau de refino da malha devido ao fenômeno de encarceramento de partículas em alguma classe (fora a última), onde a agregação e a quebra não são mais possíveis. Para o caso de agregação e quebra combinadas, o método de Hounslow atingiu o estado estacionário mais rápido do que todos os métodos, enquanto que o método do pivô móvel apresentou um alto custo computacional, aumentado proporcionalmente com o número de classes usadas.

KUMAR e RAMKRISHNA [201] combinaram o método das características com a técnica do pivô móvel que permite a solução da EBP transiente considerando crescimento, nucleação e agregação simultânea sem dependência espacial. Eles mostraram os problemas de difusão/dispersão numérica que ocorrem ao aplicar métodos de discretização simples para o termo de crescimento e que tais problemas são eliminados ao usar uma malha móvel no espaço de coordenadas internas. Esta formulação foi aplicada por LEE *et al.* [202] para simular os efeitos de crescimento, quebra e agregação de partículas em um cristalizador de sulfato de potássio. Estes autores também acoplaram um método de adaptação da malha [203] para problemas com nucleação de modo a aumentar a acurácia da solução. Uma formulação mais geral foi desenvolvida por CAMPOS e LAGE [15], que inclui a dependência espacial da EBP.

O trabalho de HU *et al.* [204] desenvolve uma formulação baseada em diferenças finitas para resolver EBPs homogêneas e monovariadas. O algoritmo parte de aproximações por série de Taylor da densidade de partículas para obter expressões algébricas para a evolução temporal do número e volume das partículas da população em um domínio discretizado em classes. As integrais dos termos de quebra e coalescência devem ser avaliadas no intervalo discretizado de cada classe e, para tal, os autores usaram a quadratura de Gauss-Legendre. Foram apresentados resultados de casos com agregação e crescimento simultâneos, quebra pura e crescimento e nu-

cleação simultânea, onde todos possuem solução analítica. Uma ótima concordância entre os resultados numéricos e analíticos foi obtida, porém os fenômenos simulados e as funções de quebra e coalescência foram muito simples. Para avaliar melhor este método, mais simulações com funções de quebra e coalescência e situações físicas mais realísticas devem ser realizadas.

3.2.4 Método dos Momentos

Uma outra abordagem faz uso dos momentos da distribuição de partículas para simular sistemas dispersos de forma apropriada. De modo geral, a função de distribuição de partículas é sistematicamente substituída por um conjunto de momentos de baixa ordem que são, na grande maioria dos casos, suficientes para estimar as propriedades físicas do sistema disperso simulado [205]. O momento k de uma distribuição está definido na Eq. 3.27.

$$\mu_k = \int_0^\infty \mathbf{v}^k f(\mathbf{x}, \mathbf{v}, t) d\mathbf{v} \quad (3.27)$$

Os momentos fornecem detalhes importantes sobre a descrição estatística da população. O momento de ordem zero ($k = 0$) representa a densidade numérica total da população (número total de partículas por unidade de volume), o momento de primeira ordem ($k = 1$) é a conservação das variáveis internas da população, e os momentos fracionais, $k = \frac{1}{3}$ e $k = \frac{2}{3}$ fornecem respectivamente informações sobre o diâmetro numérico médio e a área superficial média.

O método dos momentos (MOM) é uma técnica promissora que possui vantagens únicas ao simular sistemas dispersos dinâmicos [134, 206]. MOM é uma metodologia que acompanha diretamente os momentos de baixa ordem ao invés da distribuição propriamente dita e, portanto, não necessita de maiores conhecimentos sobre a última. O MOM convencional formula as equações para acompanhar a evolução em uma forma fechada, ou seja, são equações que só envolvem funções dos momentos. Infelizmente, este método só possui forma fechada para poucos casos. Na grande

maioria das situações, as equações que descrevem a evolução do momento μ_k envolvem momentos de maior ordem como μ_{k+1} , deixando o problema sem fechamento [134] ou com necessidade de formas matemáticas especiais para representar alguns fenômenos, como, por exemplo, o crescimento controlado pela difusão e evaporação e crescimento simultâneos [206]. A necessidade de uma formulação de fechamento é uma restrição severa deste método e é a principal razão deste não ter recebido mais atenção da comunidade acadêmica. Trabalhos mais recentes resgataram o método dos momentos formulando novas propostas para o fechamento das equações.

BARRETT e JHEETA *et al.* [207] formularam uma forma funcional dos momentos para o fechamento das equações e aplicaram para o problema de agregação, condensação e remoção de partículas em aerossóis. Os autores assumiram que o logaritmo neperiano do momento $\ln(\mu_k)$ pode ser expresso por um polinômio de ordem p em k e seus coeficientes são calculados ao resolver um sistema de $p + 1$ equações diferenciais numericamente. Contudo, sabe-se que as aproximações polinomiais possuem a tendência de oscilar entre os valores ajustados e nem sempre os momentos da distribuição podem ser aproximados pela forma colocada pelos autores.

3.2.5 Métodos Híbridos

Os métodos híbridos são assim chamados pois possuem características dos métodos das classes (discretização da variável interna) e dos momentos. Nesta classificação, uma abordagem intitulada *quadrature method of moments* (QMOM) foi proposta por MCGRAW [208], onde as equações dinâmicas para a evolução dos momentos são substituídas por um conjunto de aproximações baseadas em quadratura. Esta formulação satisfaz a condição de fechamento para uma ampla faixa de problemas sem a necessidade de assumir uma forma matemática especial para representar a distribuição de tamanho ou o crescimento de partículas. A essência do fechamento baseado em quadratura encontra-se no fato de que as abscissas ξ_α e os pesos w_α sejam completamente especificados em termos dos momentos de menor ordem da

função de distribuição. Desta forma, os pesos e abscissas representam a função de distribuição da melhor forma possível e, em contraste com o método das classes, as abscissas podem se mover livremente por todo o domínio da variável interna sem limitações ou restrições.

A formulação básica proposta por MCGRAW [208] admite uma distribuição monovariada sem dependência com variáveis externas. A Eq. 3.28 mostra a aproximação dos momentos por uma quadratura com n pontos para $2n$ momentos ($k = 0 \dots 2n - 1$).

$$\mu_k = \int_0^\infty v^k f(v, t) dv = \sum_{\alpha=1}^n \xi_\alpha^k w_\alpha \quad (3.28)$$

Deste modo, os primeiros $2n$ momentos podem ser determinados a partir de n abscissas e n pesos. Contudo, o inverso, isto é, a obtenção dos pesos e abscissas a partir dos momentos, como colocado na Eq. 3.28, requer um método de solução para equações não lineares e este procedimento não é recomendado devido ao custo computacional. Uma abordagem mais apropriada é o algoritmo Produto-Diferença (PD) [209], que usa os momentos para construir uma matriz tridiagonal da qual os pesos e abscissas podem ser obtidos.

A idéia básica do QMOM é calcular os termos fonte da EBP usando a quadratura, avaliando os pesos e abscissas ao longo da evolução temporal dos momentos. MCGRAW [208] estudou a simulação monovariada do crescimento de gotas de ácido sulfúrico diluído em aerossóis e comparou os resultados fornecidos pela solução exata do problema com as soluções numéricas obtidas pelo método dos momentos com fechamentos por série de Laguerre [134] e por quadratura (QMOM) utilizando 6 momentos (μ_0 – μ_5) para representar a distribuição de tamanho. A evolução dos momentos de menor ordem, μ_1 – μ_2 , em ambos os métodos numéricos acompanharam a solução exata, porém somente o QMOM conseguiu acompanhar a evolução dos momentos de maior ordem, μ_3 – μ_5 . O erro em relação à solução exata pelo fechamento de Laguerre aumenta junto com a ordem dos momentos.

MARCHISIO *et al.* [210] estenderam o QMOM para incluir a quebra de partí-

culas e analisaram diferentes combinações de condições iniciais de distribuição de partículas e funções de agregação e quebra. As predições de densidade numérica total de partículas (μ_0) e tamanho médio de partículas ($d_{34} = \mu_4/\mu_3$) foram comparadas com a solução rigorosa da EBP fornecida por VANNI [193] utilizando um grande número de classes (aproximadamente 2000). O número de variáveis escalares necessárias para resolver a EBP pelo QMOM é muito inferior comparado ao método das classes obtendo a mesma acurácia.

A formulação do QMOM obtida por MARCHISIO *et al.* [210] para a evolução de $2n$ momentos contendo os termos de quebra e agregação está colocada na Eq. 3.29 e a Eq. 3.30 contém os momentos da densidade de probabilidade de quebra.

$$\begin{aligned} \frac{\partial \mu_k}{\partial t} = & \frac{1}{2} \sum_{\alpha=1}^n \sum_{\beta=1}^n [(\xi_\alpha + \xi_\beta)^k - \xi_\alpha^k - \xi_\beta^k] a(\xi_\alpha, \xi_\beta) w_\alpha w_\beta \\ & + \sum_{\alpha=1}^n b(\xi_\alpha) w_\alpha [\vartheta(\xi_\alpha) \pi_k(\xi_\alpha) - \xi_\alpha^k] \end{aligned} \quad (3.29)$$

onde

$$\pi_k(\xi_\alpha) = \int_0^{\xi_\alpha} v^k P(v | \xi_\alpha) dv \quad (3.30)$$

Apesar de tudo, QMOM possui dois fatores limitantes: (i) se aplicado a distribuições multivariadas, o método perde sua simplicidade e eficiência computacional, e (ii) ao acompanhar apenas os momentos da distribuição de partículas, o método não representa de forma realística sistemas polidispersos com forte acoplamento entre as variáveis internas e as velocidades das fases. De forma a driblar esses fatores, MARCHISIO e FOX [211] apresentam uma metodologia derivada a partir do QMOM, intitulada *Direct Quadrature Method Of Moments* (DQMOM). O DQMOM é baseado na idéia de acompanhar diretamente os pesos e abscissas presentes na aproximação de quadratura ao invés de acompanhar os momentos da distribuição propriamente ditos. Como no QMOM, a função de distribuição é unicamente aproximada como o somatório de funções delta de Dirac multidimensionais. A Eq. 3.31 mostra a aproximação monovariada para a função de distribuição de tamanho pela

formulação do DQMOM.

$$f(\mathbf{x}, v, t) = \sum_{\alpha=1}^n w_{\alpha}(\mathbf{x}, t) \delta[v - \xi_{\alpha}(\mathbf{x}, t)] \quad (3.31)$$

Esta forma funcional pode ser analisada como um conjunto de n fases dispersas e cada fase é caracterizada por um peso w_{α} e uma abscissa ξ_{α} , que representam a densidade numérica e o valor da propriedade considerada em cada fase dispersa. A formulação do DQMOM se baseia na substituição da Eq. 3.31 diretamente na EBP (Eq. 3.3) que após alguma manipulação fica na forma mostrada na Eq. 3.32, onde $\varsigma_{\alpha} = w_{\alpha}\xi_{\alpha}$ é a abscissa ponderada, \mathbf{u}_{α} é a velocidade característica associada com a fase α e δ' e δ'' são as primeiras e segundas derivadas da função generalizada δ .

$$\begin{aligned} & \sum_{\alpha=1}^n \delta[v - \xi_{\alpha}(\mathbf{x}, t)] \left[\frac{\partial w_{\alpha}}{\partial t} + \nabla \cdot (\mathbf{u}_{\alpha} w_{\alpha}) - \nabla \cdot [D_x \nabla w_{\alpha}] \right] \\ & - \sum_{\alpha=1}^n \delta'[v - \xi_{\alpha}(\mathbf{x}, t)] \left[\frac{\partial \varsigma_{\alpha}}{\partial t} + \nabla \cdot (\mathbf{u}_{\alpha} \varsigma_{\alpha}) - \nabla \cdot [D_x \nabla \varsigma_{\alpha}] \right] \\ & + \sum_{\alpha=1}^n \delta'[v - \xi_{\alpha}(\mathbf{x}, t)] \xi_{\alpha} \left[\frac{\partial w_{\alpha}}{\partial t} + \nabla \cdot (\mathbf{u}_{\alpha} w_{\alpha}) - \nabla \cdot [D_x \nabla w_{\alpha}] \right] \\ & - \sum_{\alpha=1}^n \delta''[v - \xi_{\alpha}(\mathbf{x}, t)] \{ D_x w_{\alpha} (\nabla \xi_{\alpha}) \cdot (\nabla \xi_{\alpha}) \} = H(\mathbf{x}, v, t) \end{aligned} \quad (3.32)$$

Define-se θ_{α} e κ_{α} como termos fonte das equações de transporte do DQMOM para os pesos e abscissas ponderadas.

$$\begin{aligned} \frac{\partial w_{\alpha}}{\partial t} + \nabla \cdot (\mathbf{u}_{\alpha} w_{\alpha}) - \nabla \cdot [D_x \nabla w_{\alpha}] &= \theta_{\alpha} \\ \frac{\partial \varsigma_{\alpha}}{\partial t} + \nabla \cdot (\mathbf{u}_{\alpha} \varsigma_{\alpha}) - \nabla \cdot [D_x \nabla \varsigma_{\alpha}] &= \kappa_{\alpha} \end{aligned} \quad (3.33)$$

Define-se ainda C_{α} na seqüência.

$$C_{\alpha} = D_x w_{\alpha} (\nabla \xi_{\alpha}) \cdot (\nabla \xi_{\alpha}) \quad (3.34)$$

Note que, quando não existe termo dispersivo, C_{α} assume valor nulo para todas as fases.

Usando as definições das Eqs. 3.33 e 3.34 e aplicando uma transformação na Eq. 3.32 que consiste em multiplicá-la por v^k e integrá-la, chega-se à Eq. 3.35.

$$(1 - k) \sum_{\alpha=1}^n \xi_{\alpha}^k \theta_{\alpha} + k \sum_{\alpha=1}^n \xi_{\alpha}^{k-1} \kappa_{\alpha} = \bar{H}_k^{(n)} + \bar{C}_k \quad (3.35)$$

onde os termos fonte são definidos por

$$\bar{H}_k^{(n)} = \int_0^{\infty} v^k H(\mathbf{x}, v, t) dv \quad (3.36)$$

e

$$\bar{C}_k = k(k - 1) \sum_{\alpha=1}^n \xi_{\alpha}^{k-2} C_{\alpha} \quad (3.37)$$

A Eq. 3.35 forma um sistema linear de equações que deve ser resolvido numericamente em cada nó da malha das variáveis externas em cada passo de tempo para obter os termos fontes θ_{α} e κ_{α} , necessários para resolver as Eqs. 3.33. O trabalho de MARCHISIO e FOX [211] desenvolve a formulação para os casos de crescimento, dispersão, nucleação, agregação e quebra homogêneos monovariados, difusão pura monovariada e estende a agregação homogênea e o crescimento com dependência espacial para casos bivariados. Os autores ainda colocam a formulação do DQMOM para casos multivariados, contudo não apresentam testes com esta última.

Uma metodologia derivada do método das classes chamada *Parallel Parent and Daughter Classes* (PPDC) foi desenvolvida recentemente por BOVE *et al.* [212]. O conceito deste método é uma decomposição de operadores aliada ao uso de várias malhas para as partículas nascidas. Desta forma, os efeitos de nascimento e morte (por quebra e agregação) de partículas são separados em diferentes malhas. Os autores utilizaram funções de Dirac para decompor a função de distribuição em três componentes, que se referem a uma malha para as classes de partícula mãe, M malhas para as classes de filhas geradas por quebra e $M(M + 1)/2$ malhas para as classes filhas geradas pelo processo de agregação. Essas malhas estão superpostas (paralelas) no domínio da coordenada interna. BOVE *et al.* [212] partiram de uma discretização por Euler explícito no tempo para obter as Eqs. 3.38, 3.39 e 3.40 referentes a cada componente da função de distribuição, onde N_{α} , $A_{\alpha\beta}$ e $B_{\gamma}^{(\alpha)}$ são

respectivamente as densidades numéricas de partículas para a classe mãe α , para a classe de filha $\alpha\beta$ gerada por agregação de partículas presentes nas classes mãe α e β e para a classe de filha γ gerada por quebra de uma partícula presente na classe mãe α .

$$\frac{N_\alpha(t^{n+1}) - N_\alpha(t^n)}{\Delta t} = -N_\alpha(t^n) \sum_{\beta=1}^M a(\xi_\alpha, \xi_\beta) N_\beta(t^n) - b(\xi_\alpha) N_\alpha(t^n), \quad (3.38)$$

$$\alpha = 1, \dots, M$$

$$\frac{A_{\alpha\beta}(t^{n+1}) - A_{\alpha\beta}(t^n)}{\Delta t} = \left(1 - \frac{1}{2}\delta_{\alpha\beta}\right) a(\xi_\alpha, \xi_\beta) N_\alpha(t^n), \quad (3.39)$$

$$\alpha, \beta = 1, \dots, M, \beta \geq \alpha$$

$$\frac{B_\gamma^{(\alpha)}(t^{n+1}) - B_\gamma^{(\alpha)}(t^n)}{\Delta t} = \vartheta_\alpha b(\xi_\alpha) N_\alpha(t^n) \int_{v_\gamma^{(\alpha)}}^{v_{\gamma+1}^{(\alpha)}} P(v | \xi_\alpha) dv, \quad (3.40)$$

$$\alpha = 1, \dots, M, \gamma = 1, \dots, NB(\alpha)$$

Após resolver o sistema das Eqs. 3.38 à 3.40, a função de distribuição expandida de tamanho de partículas no tempo t^{n+1} pode ser calculada pela Eq. 3.41, onde $y_{\alpha\beta}$ e $z_\gamma^{(\alpha)}$ são as abscissas nas malhas de agregação e quebra respectivamente.

$$f_E(v, t^{n+1}) = \sum_{\alpha=1}^M N_\alpha(t^{n+1}) \delta(v - \xi_\alpha) + \sum_{\alpha=1}^M \sum_{\beta=1}^M A_{\alpha\beta}(t^{n+1}) \delta(v - y_{\alpha\beta})$$

$$+ \sum_{\alpha=1}^M \sum_{\gamma=1}^{NB(\alpha)} B_\gamma^{(\alpha)}(t^{n+1}) \delta(v - z_\gamma^{(\alpha)}) \quad (3.41)$$

Contudo, a cada iteração no tempo é necessário expressar a função de distribuição expandida em M classes (malha inicial para o próximo passo de integração) e obter os momentos da distribuição usando a Eq. 3.28. Este procedimento resulta em um sistema de equações não lineares em N_α^{n+1} e $\xi_\alpha^{k,n+1}$. Os autores sugerem o método de GORDON [209] para resolver tal sistema devido à sua eficiência e robustez.

BOVE *et al.* [212] compararam simulações do PPDC com a solução analítica de MCCOY e MADRAS [160] e o resultado do método numérico de KUMAR e RAMKRISHNA [198] para três casos usando diferentes funções de agregação e quebra.

Os autores não relatam o passo de tempo adotado nas simulações. Os resultados obtidos pelo PPDC obtiveram ótima concordância com a solução analítica, mesmo utilizando um pequeno número de classes. No segundo caso analisado, foram realizadas comparações entre os resultados obtidos pelo PPDC (2 a 4 classes) e pelo método do pivô móvel [198] (90 classes). Neste caso, as simulações usando diferentes classes no PPDC convergiram entre si, porém com grande discordância dos resultados fornecidos pelo método do pivô móvel. Os autores não comentam esse fato apesar da simplicidade das funções de quebra e agregação deste caso. Uma possível solução seria aumentar o número de classes no método de KUMAR e RAM-KRISHNA [198] para avaliar a convergência da solução fornecida pelo mesmo. Uma das grandes desvantagens do PPDC está em sua formulação básica, que parte da integração numérica por Euler explícito.

Atualmente, o desenvolvimento dos métodos híbridos passou a focar os casos multivariados, em especial usando o QMOM e DQMOM. O primeiro estudo aplicando o QMOM em casos multivariados [213] utilizou o volume e a área superficial das partículas como variáveis internas para o caso de agregação pura. Nesta situação, os pesos e abscissas não podem ser obtidos pelo algoritmo PD e foram calculados pelo método do gradiente conjugado propiciando um maior custo computacional. Os resultados utilizando 12 pontos de quadratura foram comparados à simulação por Monte Carlo apresentando resultados concordantes. MCGRAW e WRIGHT [214] apresentaram uma nova metodologia derivada do QMOM para o fechamento dos momentos, chamada de Transformação da Matriz Jacobiana (TMJ), com o objetivo de tratar misturas multicomponentes em aerossóis. Contudo, os autores relacionaram os momentos das composições das partículas com os momentos da distribuição total de massa e, portanto, realizaram apenas simulações monovariadas para acompanhar a evolução da população.

Na sequência de trabalhos de YOON e MCGRAW [215, 216] foi introduzida uma formulação multivariada do QMOM acoplada ao algoritmo de análise dos principais componentes da população [217, 218] de modo a acompanhar os momentos combi-

dados para composição das partículas. O algoritmo de análise dos principais componentes fornece um técnica para extrair as combinações lineares não-correlacionadas que melhor caracterizam a função de distribuição. Neste caso, o algoritmo é usado para encontrar o número de pontos de quadratura e sua localização ótima para obter os momentos combinados. O método é aplicado a uma diversidade de exemplos como, a condensação e a coagulação multicomponente (efeitos separados) e a coagulação e sinterização bivariada em área e volume (efeitos combinados). Os resultados foram comparados com as simulações numéricas de WRIGHT *et al.* [213] apresentando boa concordância.

FOX [219] demonstrou a equivalência do DQMOM com o QMOM aplicado aos mesmos casos bivariados tratado por WRIGHT *et al.* [213]. Em casos sem dependência espacial, a solução pelo DQMOM é equivalente à TMJ [214] usado para obter os momentos. Em contraste, a dependência espacial produz termos fontes que consideram os efeitos referentes aos gradientes dos pesos e abscissas. Estes termos não podem ser desprezados na formulação bivariada do DQMOM, com pena de prever incorretamente os momentos.

Como discutido no trabalho de DIEMER e OLSON [220], o desenvolvimento de soluções eficientes baseadas no método dos momentos abrirá portas à modelagem de processos bivariados e espacialmente não-homogêneos já que o acoplamento CFD-EBP usando métodos seccionais ainda são computacionalmente intensivos. Contudo, um estudo mais aprofundado dos métodos multivariados foge do escopo deste trabalho.

3.3 Acoplamento CFD-EBP

Separadamente, as técnicas numéricas para CFD e EBP são campos de pesquisa muito ativos. Contudo, o acoplamento destas técnicas se torna necessário ao considerar de forma mais rigorosa os efeitos dos fenômenos relacionados às partículas (por

exemplo, quebra e agregação). Desta forma, a EBP deve ser resolvida junto com as equações da continuidade e de balanço de quantidade de movimento multifásicas. Esta seção tem o objetivo de mostrar a teoria e formulação das técnicas de acoplamento CFD-EBP e apresentar uma revisão da literatura referente às aplicações multifásicas acopladas ao balanço populacional.

3.3.1 Fundamentos do Acoplamento

A forma mais geral do acoplamento da EBP com o modelo Euleriano multifásico considera que todas as classes de partículas são tratadas como fases dispersas completamente independentes. Assim, cada fase é caracterizada por uma classe distinta com um diâmetro de partícula e possui campos de velocidade e fração volumétrica próprios. De forma natural, estas características fazem com que as abordagens discretas para a solução da EBP (método das classes, por exemplo) sejam as mais apropriadas ao acoplamento com escoamentos multifásicos. Contudo, a necessidade de considerar um número elevado de classes (20 – 30) para representar de forma apropriada a distribuição de partículas torna esta abordagem computacionalmente exaustiva. Dessa forma, outras formulações para o acoplamento CFD-EBP foram propostas.

A primeira tentativa em acoplar CFD e EBP foi o modelo MUSIG (*MUltiple SIze Group*) introduzida por LO [221], baseada na formulação do modelo de dois fluidos onde a fase dispersa inclui todas as classes da distribuição de partículas. A modelagem MUSIG assume que as velocidades de todas as partículas são iguais, sem dependência com seu tamanho. Desta forma, o conjunto de equações se reduz à abordagem de dois fluidos onde existe um campo de velocidade para a fase contínua e outro para a dispersa. As forças de interação entre as fases são calculadas usando o diâmetro médio de Sauter, d_s , de toda a fase dispersa. O diâmetro médio de Sauter usado no cálculo da força de arrasto pode ser aproximado pela Eq. 3.42, que usa a função de distribuição de tamanho, obtida pela equação de balanço populacional,

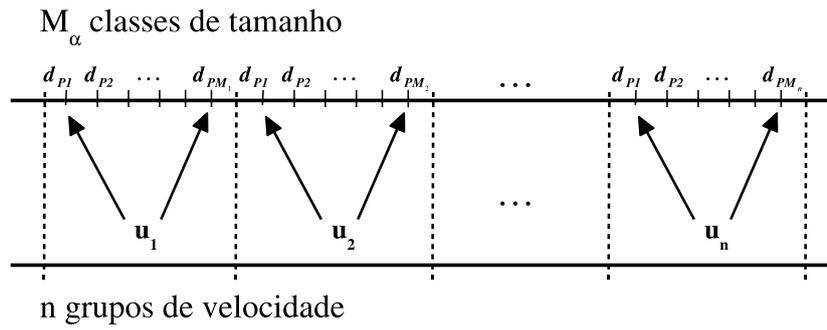


Figura 3.3: Representação dos n grupos de partícula separados em uma malha discreta em tamanho e suas respectivas velocidades.

integrada no volume

$$d_s \approx \frac{\int_0^\infty d^3(v)f(v) dv}{\int_0^\infty d^2(v)f(v) dv}, \quad (3.42)$$

onde $d(v)$ é o diâmetro de uma partícula com volume v .

A limitação do modelo MUSIG reside na sua incapacidade em prever a dinâmica de diferentes classes. Na tentativa de driblar esta limitação e seguindo a mesma idéia do modelo de KRISHNA *et al.* [123], KREPPER *et al.* [222] usou o modelo MUSIG modificado que considera duas fases gasosas para representar separadamente a dinâmica das bolhas pequenas e das bolhas grandes.

Em seguida, FRANK *et al.* [223, 224] desenvolveram o MUSIG não-homogêneo ao considerar que grupos de partículas se movem com diferentes velocidades. Neste caso, cada grupo representa uma fase dispersa do sistema polidisperso. Um esquema desta abordagem está colocado na Fig. 3.3 onde as partículas são agrupadas por tamanho em seções da malha e cada grupo possui uma velocidade distinta.

FRANK *et al.* [223, 224] sugerem que os grupos de velocidade, composto por grupos de classes, sejam caracterizados pelo comportamento das partículas devido às forças de troca de *momentum* e dispersão turbulência. Segundo os autores, 3 ou 4 grupos de velocidade (ou fases dispersas) são suficientes para caracterizar o escoamento homogêneo ou *slug* em colunas de borbulhamento. Os limites de diâmetros superior e inferior de cada grupo devem ser definidos e o diâmetro médio em cada grupo é usado para calcular os termos de troca de *momentum* entre as fases. Ape-

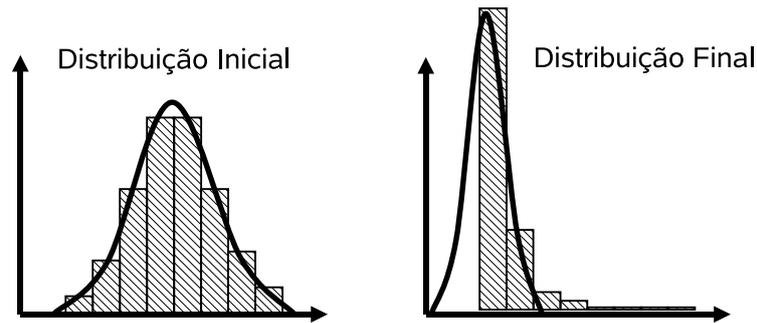


Figura 3.4: Exemplo da falta de acurácia em representar a função de distribuição usando o método das classes [166] em um caso com quebra dominante.

sar de usar todo o domínio discreto do tamanho de partícula (considerando todos os grupos) para calcular os efeitos de interação entre partículas, os autores não são claros em como o diâmetro médio de cada grupo é obtido.

O modelo MUSIG foi amplamente divulgado devido a sua implementação no pacote ANSYS CFX [225]. Este modelo ainda é utilizado até hoje, mas é bastante criticado pela sua ineficiência [18]. Recentemente foi descoberto um erro na implementação da discretização do termo de quebra [226], onde a massa é conservada mas não o número de partículas. Segundo DAMIAN [18], a ANSYS CFX já tomou conhecimento deste problema, mas até o momento não implementou a correção. Desta forma, acredita-se que todos os trabalhos que utilizaram o MUSIG até hoje tenham sido afetados em precisão por causa deste problema.

Note que as técnicas de acoplamento CFD apresentadas acima consideraram apenas o método das classes de KUMAR e RAMKRISHNA [184] na solução da EBP. Contudo, o uso de pivôs fixos limita esta técnica e pode levar a erros grosseiros na representação da função de distribuição. Isto ocorre na representação mostrada na Fig. 3.4, onde a função de distribuição inicial é alterada devido aos efeitos de quebra dominante.

No exemplo mostrado na Fig. 3.4, o método das classes não é capaz de representar corretamente a função de distribuição pois o limite inferior das classes foi ultrapassado e várias classes se tornaram não-representativas. Ao contrário, os mé-

todos híbridos se adaptam de forma ótima às mudanças na função de distribuição devido à aproximação pela quadratura ótima. Desta forma, os pesos e abscissas se ajustam para melhor representar a função de distribuição.

As metodologias híbridas usando quadratura também são atrativas ao acoplamento CFD-EBP por outras razões. A primeira delas é a alta acurácia dos métodos para casos de EBP monovariadas utilizando um pequeno número de momentos (4 – 6). Outra vantagem inerente ao DQMOM é que os pesos e abscissas podem ser vistos como pontos distintos no espaço das variáveis externas e o método pode ser usado para descrever sistemas multifásicos. Assim, cada fase dispersa seria caracterizada pelas variáveis de cada ponto de quadratura. O presente trabalho apresenta a comparação da acurácia e performance dos métodos híbridos QMOM, DQMOM e PPDC. Estes resultados estão discutidos no Cap. 6. Como conclusão, o DQMOM se apresenta como a técnica mais apropriada para ser acoplada a problemas de CFD.

3.3.2 Revisão de Aplicações usando CFD-EBP

O trabalho de OLMOS *et al.* [227] utilizou a formulação MUSIG desenvolvida por LO [221] e implementada no CFX 4.3 para combinar as equações de balanço populacional à simulação Euleriana-Euleriana de uma coluna de borbulhamento. A fase dispersa foi representada por 10 grupos de diferentes tamanhos, contudo apenas uma equação de balanço de *momentum* para a fase dispersa foi resolvida. Dessa forma, as 10 classes de partículas são convectadas com a mesma velocidade média. O diâmetro médio de Sauter foi usado para calcular os termos de arrasto e força de dispersão turbulenta [228]. Os autores usaram o modelo clássico $k - \epsilon$ para modelar a turbulência na fase contínua. A simulação foi estacionária e 2D axialmente simétrica e apenas os fenômenos de quebra e coalescência foram considerados utilizando os modelos de LUO e SEVENDSEN [155] e PRINCE e BLANCH [229], respectivamente. Os resultados foram comparados com dados experimentais e apresentaram uma boa concordância para o regime de borbulhamento homogêneo e a faixa de transição

de regime. Contudo, a restrição do modelo para as equações de transporte para cada classe de bolhas e a limitação da simulação bidimensional devem ter limitado a acurácia dos resultados de OLMOS *et al.* [227].

OLMOS *et al.* [230] se basearam na abordagem desenvolvida por KRISHNA *et al.* [123] para estender seu trabalho anterior [227]. Os autores consideraram a influência das bolhas na turbulência da fase contínua através dos modelos de SATO e SEKOGUCHI [76] que calcula um termo extra de viscosidade induzida pelas bolhas, e KATAOKA e SERIZAWA [82] que leva em conta a produção de turbulência nas esteiras das bolhas. Os autores ainda consideraram as interações entre bolhas através de uma correção do coeficiente de arrasto, mostrada na Eq. 3.43, que utiliza um parâmetro p em sua formulação [231] a ser obtido a partir de dados experimentais, a fração volumétrica da fase gás, α_g , e o coeficiente de arrasto sem fator de correção, C'_d .

$$C_d = C'_d(1 - \alpha_g)^p \quad (3.43)$$

A metodologia de solução proposta por OLMOS *et al.* [230] utiliza os resultados das simulações de seu trabalho anterior [227] para obter as classes dominantes (e portanto os diâmetros médios) para diferentes velocidades superficiais de gás e o ajuste do parâmetro p através de dados experimentais de retenção gasosa e velocidade superficial de gás. Após estas duas etapas, é resolvido o modelo Euleriano-Euleriano com as equações de conservação de *momentum* para a fase líquida e para as fases gasosas com os diâmetros médios das classes dominantes e desconsiderando a interação entre as bolhas. Os resultados mostraram uma boa concordância na transição do regime homogêneo ao heterogêneo. Ao aplicar os coeficientes de arrasto ajustados para cada fase gasosa, foi possível obter uma boa representação da transição entre os regimes de borbulhamento. OLMOS *et al.* [230] concluíram que estes resultados só foram possíveis devido à consideração da distribuição de tamanho de bolhas e as interações entre elas e ressaltaram a importância do uso de balanço populacional para representar o regime de transição de borbulhamento. Os autores também colocaram que o modelo de KATAOKA e SERIZAWA [82] apresentou resultados mais

acurados na descrição da turbulência no líquido.

Em trabalho recente, CHEN *et al.* [27] implementaram a EBP acoplada ao modelo Euleriano-Euleriano de dois fluidos na tentativa de representar o regime heterogêneo de borbulhamento. O modelo Euleriano bifásico descrito por PAN *et al.* [105] foi usado no trabalho considerando a abordagem ASMM e incluindo apenas a força de arrasto [232] no termo de troca de *momentum* entre fases e o diâmetro local da bolha é obtido pela solução da EBP. Isto implica que o modelo de mistura não requer a solução dos balanços de quantidade de movimento e massa para todas as fases, apenas para a mistura, e que todas as partículas fluidas se deslocam com uma única velocidade. A turbulência na fase líquida foi tratada com um modelo $k - \epsilon$ modificado para incluir transferência de momento turbulento entre as fases [107] e na fase dispersa o fechamento da turbulência é feito através de correlações baseadas na dispersão de partículas por turbulência homogênea [233]. Os fenômenos de quebra [155] e coalescência [234] foram considerados na EBP, contudo foi necessário ajustar a taxa de quebra predita pelo modelo de LUO e SVENDSEN [155] com um fator de 10 vezes ($b = 10 b^{\text{original}}$). CHEN *et al.* [27] sugerem que isto ocorre devido à falta de acurácia do modelo $k - \epsilon$ em fornecer valores realísticos de energia de dissipação turbulenta. Contudo, de acordo com ARAUJO [17], a formulação do modelo de LUO e SVENDSEN [155] é baseada em conceitos errados e pode ter sido a principal fonte de erro nas simulações. Os autores reportaram resultados próximos ao obtido experimentalmente [235] para a velocidade axial das bolhas e a retenção gasosa em simulações 2D e 3D.

CHEN *et al.* [236] estenderam seu trabalho avaliando o efeito de diferentes modelos de quebra e coalescência para obtenção da distribuição de tamanho de bolhas no processo de borbulhamento heterogêneo. A EBP foi resolvida pelo método do pivô fixo [184] considerando todas as partículas com uma única velocidade calculada pelo modelo Euleriano-Euleriano [27] e os modelos analisados para quebra foram de LUO e SVENDSEN [155] e MARTÍNEZ-BAZÁN *et al.* [237, 238] e para coalescência de PRINCE e BLANCH [229] e LUO [234]. Como no trabalho anterior, foi necessário

ajustar a taxa de quebra em um fator de 10 vezes, fato que tira a natureza preditiva da simulação numérica. Foram realizadas simulações bidimensionais discretizando a EBP com uma malha fixa no volume usando 16 classes com diâmetros de bolha entre 1 mm e 32 mm. Os resultados apontam que a solução da EBP permite uma melhor representação dos perfis de velocidade do líquido e de energia cinética turbulenta em relação às simulações usando um diâmetro médio fixo, apesar de ambas ainda apresentarem discrepâncias em relação ao experimento. Isto pode ser consequência da natureza da simulação 2D que não captura todos os detalhes do escoamento. Os autores também ressaltaram a concordância dos resultados simulados usando os diferentes modelos de quebra e coalescência, que apresentaram uma distribuição unimodal do tamanho de bolhas no regime heterogêneo de borbulhamento, apresentando grande discrepância com a curva bimodal experimental [239]. Apesar disso, os resultados de retenção gasosa não foram satisfatórios.

Em seqüência, CHEN *et al.* [240] realizaram simulações tridimensionais de colunas de borbulhamento com diferentes diâmetros e condições de pressão e velocidades superficiais de modo a operar em regime heterogêneo. Os resultados simulados foram comparados com dados experimentais da literatura [241, 242, 243]. Os autores utilizaram o mesmo modelo [27], mudando o modelo de frequência de colisão para o desenvolvido por SAFFMAN e TURNER [244]. Este trabalho avaliou três abordagens de solução do escoamento, sendo elas: (i) EBP com modelo de dois fluidos assumindo que a velocidade de todas as fases dispersas é localmente igual à velocidade média da fase gás; (ii) EBP com ASMM, sendo que a velocidade de todas as fases de bolhas são localmente iguais à velocidade da fase gasosa; (iii) EBP com ASMM com $n + 1$ fases, onde cada classe possui sua velocidade local calculada pelo ASMM. Contudo, o tipo de abordagem de solução não teve um efeito significativo nos resultados simulados para os valores médios temporais da velocidade axial e da retenção gasosa e os autores adotaram a metodologia (i) com 9 classes para gerar os resultados comentados a seguir. As flutuações de velocidade superficial e retenção gasosa conseguem ser capturadas, fato que não ocorre em simulações 2D. Apesar disso, os valores numéricos para a tensão turbulenta não são bem preditos e a curva

de distribuição de tamanho de bolhas ainda é unimodal.

O primeiro trabalho que testa a capacidade do método dos momentos com fechamento por quadratura para resolver problemas de escoamento acoplado à EBP foi de MARCHISIO *et al.* [245]. Os autores implementaram o QMOM no código CFD comercial FLUENT para simular o escoamento turbulento com partículas sólidas em um reator Taylor-Couette considerando os efeitos de quebra e coalescência. Devido ao pequeno tamanho das partículas e à sua baixa concentração no reator foi possível desconsiderar a influência da fase dispersa sobre a contínua e adotar um modelo de turbulência monofásico. As simulações foram comparadas com dados experimentais encontrados na literatura. Os resultados foram obtidos resolvendo o campo de escoamento monofásico até atingir o estado permanente e em seguida determinando a evolução dinâmica dos momentos μ_0 - μ_5 [210] em passos de 10 segundos. Diferentes modelos de agregação, quebra e distribuição de partículas filhas foram testados e foi possível notar claramente a dependência do resultado na escolha destes modelos. Em outras palavras, deve-se escolher corretamente estes modelos com base na física do processo analisado e nas escalas de tempo e dimensão para a qual estes foram desenvolvidos. Contudo, o custo computacional para resolver a EBP não apresentou dependência com as diferentes combinações de modelos de quebra e coalescência.

Em trabalho recente, PRAT e DUCOSTE [246] introduziram o QMOM no *software* de fluidodinâmica PHOENICS via subrotinas em FORTRAN, definidas pelo usuário para simular a evolução espacial da distribuição de tamanho de flocos em um reator agitado mecanicamente. As simulações utilizaram 6 momentos para caracterizar a população de partículas e foram usados os modelos de SAFFMAN e TURNER [244] e ADLER [247] para coalescência e KUSTERS [248] para quebra binária. A cada passo de tempo da simulação transiente, os momentos são atualizados para obter os pesos e abscissas utilizando as rotinas ORTHOG e GAUCOF [249] para estruturar o algoritmo de solução. A partir dos momentos normalizados, a rotina ORTHOG realiza o cômputo dos coeficientes da matriz jacobiana usando o algoritmo de Wheeler [250] e a rotina GAUCOF extrai os pesos e abscissas da

fórmula da quadratura.

O trabalho de FAN *et al.* [251] aplica o DQMOM acoplado a um modelo multi-fluido para simular a evolução da fase sólida polidispersa considerando a quebra e agregação em um reator de leito fluidizado. São consideradas as equações da continuidade e de *momentum* para uma fase contínua e n fases dispersas, incluindo apenas o arraste no termo de troca de *momentum* entre fases e a força de empuxo. Os autores adotaram modelos de quebra e agregação propícios para a dinâmica de partículas em leitos fluidizados baseados na teoria cinética de escoamento granular [231], porém alguns testes com funções constantes foram realizados para analisar melhor o comportamento da simulação. Os métodos SIMPLE (*Semi-Implicit Method for Pressure-Linked Equations* [252]) de acoplamento pressão velocidade e PEA (*Partial Elimination Algorithm* [253]) no tratamento do acoplamento entre fases foram usados. Foram realizadas simulações 2D com 4, 6 e 8 momentos apresentando resultados promissores na descrição dos fenômenos de crescimento, segregação, elutriação das partículas devido à quebra e agregação (apesar de não apresentarem comparações com dados experimentais). Como já esperado, os modelos baseados na teoria cinética apresentaram maior acurácia na descrição dos fenômenos. Os autores recomendam o uso de 6 momentos no cálculo, ponderando o tempo computacional e a acurácia obtida nos casos analisados.

Uma característica importante do trabalho de FAN *et al.* [251] é que a formulação básica do DQMOM acoplado ao modelo Euleriano-Euleriano não precisa ser alterada quando se deseja incluir outros fenômenos como transferência de massa e/ou calor, turbulência, reações químicas, etc. na simulação. Contudo, é necessário usar um número maior de variáveis internas. Recentemente, o modelo multi-fluido de FAN *et al.* [251] foi usado para avaliar a segregação das partículas em um leito fluidizado [254] e verificar sua validação ao comparar as simulações com dados experimentais [255] e dados simulados usando uma abordagem lagrangeana [256]. Em comparação aos dados experimentais, as simulações de FAN e FOX [254] conseguiram representar bem a segregação das partículas utilizando apenas 2 pontos de quadratura (caracte-

rizados pelas partículas grandes e pequenas). Contudo, diferenças intrínsecas entre os modelos e condições de contorno nas simulações não permitiu uma boa comparação entre os resultados usando as abordagens multi-fluido [251] e a lagrangeana [256].

ZUCCA *et al.* [257] aplicaram o DQMOM para resolver a EBP na predição da formação de fuligem em chamas turbulentas de etileno-ar. As equações da continuidade, conservação de quantidade de movimento e transporte de massa com reação passaram por um processo de média de modo a considerar a natureza turbulenta do processo. A descrição da modelagem dos termos de tensão de Reynolds ($k - \epsilon$) e as interações turbulentas das espécies químicas (modelo de densidade de probabilidade) decorrentes do processo de média podem ser encontradas em FOX [258]. Os efeitos de nucleação, crescimento molecular, oxidação e agregação foram incluídos na EBP, onde as partículas foram consideradas pequenas o suficiente para não afetar o campo de escoamento na chama. Os autores usaram o FLUENT para calcular os campos de velocidade, temperatura e concentração em simulações bidimensionais com o DQMOM implementado via função definida pelo usuário com 4 – 6 momentos. Foram realizadas comparações com dados experimentais [259] do perfil de temperatura e fração volumétrica de fuligem ao longo do queimador apresentando uma concordância satisfatória dos resultados simulados. Contudo, não existem dados suficientes para validar a evolução da distribuição de tamanho das partículas.

Em trabalho recente, DAMIAN [18, 260] implementou o acoplamento entre o modelo multifásico Euleriano e a EBP resolvida pelo DQMOM no ANSYS CFX através das rotinas de usuário em FORTRAN. Testes de verificação numérica da implementação foram realizados utilizando uma solução analítica [160, 158], que resolve a evolução temporal de um problema de balanço populacional monovariado com núcleos de quebra e agregação simples [160]. A precisão dos esquemas temporais e advectivos na solução da EBP usando o DQMOM foram explorados em casos transientes 0D e estacionários 1D, onde os métodos *Backward Euler* e *High Resolution* se destacaram. Também foram realizadas simulações bidimensionais para o

escoamento de uma emulsão usando o acoplamento CFD-EBP completo e a abordagem MUSIG. Apesar de obter resultados bastante satisfatórios, o próprio ANSYS CFX limita as implementações via rotinas de usuário em FORTRAN ao restringir a liberdade e a flexibilidade dos códigos. Além disso, a documentação sobre o uso das rotinas de usuário em FORTRAN é muito limitada e vaga, dificultando ainda mais o processo de implementação, modificação ou extensão destes códigos.

Capítulo 4

Pacote CFD — OpenFOAM

4.1 Introdução

Não há dúvidas que os pacotes CFD são ferramentas que fornecem grandes vantagens e uma maior facilidade na análise de diversos tipos de escoamentos. A importância dos pacotes CFD têm crescido muito nos últimos anos devido à confiabilidade e à rapidez com que os resultados são obtidos, além da economia frente a testes experimentais. Este fato se deve aos avanços no desenvolvimento da modelagem dos fenômenos físicos, das técnicas numéricas para solucionar o problema e do aumento do poderio computacional. Como consequência, além da facilidade de uso, o constante aprimoramento das técnicas e metodologias presentes nos códigos são considerados pontos cruciais em CFD. Desta forma, é interessante que o usuário possa interagir intimamente com o pacote, possibilitando a extensão do código e aprimorando a ferramenta ou mesmo implementando novas formas de solucionar um problema específico.

O pacote CFD OpenFOAM surge como uma ótima opção para a solução de uma grande variedade de problemas. Além disso, este pacote é distribuído gratuitamente e junto com seu código fonte e, portanto, permite uma interação mais profunda com

o usuário. Este capítulo apresenta as origens do desenvolvimento do OpenFOAM, assim como a metodologia numérica utilizada em sua implementação. Além disso, detalhes sobre a interpretação da linguagem de programação pelo pacote e seu uso para a solução de problemas CFD são apresentados na sequência. Por fim, um estudo dos modelos multifásicos já implementados no OpenFOAM é realizado, com uma análise mais aprofundada da abordagem bifásica Euleriana.

4.1.1 Origem do OpenFOAM

Quando se inicia um projeto de construção de um código CFD, é interessante estruturá-lo de forma que se torne geral o suficiente para que outras pessoas possam usá-lo no futuro como uma ferramenta confiável, eficiente e de fácil manipulação. Esta preocupação envolveu os pensamentos dos então alunos do *Imperial College*, Henry Weller e Hrvoje Jasak quando, em 1993, combinaram esforços para desenvolver o código do FOAM (*Field Operation and Manipulation*) para operar e manipular campos tensoriais visando sua aplicação à fluidodinâmica computacional. A tecnologia do FOAM é baseada em um conjunto eficiente e flexível de módulos escritos em C++ que são usados com o intuito de construir: (i) *solvers*¹ para resolver problemas específicos de engenharia que envolvam campos vetoriais, (ii) utilitários para realizar tarefas de pré e pós-processamento, que vão de uma simples manipulação de dados à visualização e construção e processamento de malhas, e (iii) bibliotecas de expansão usadas pelos *solvers* e utilitários, tal como uma biblioteca de modelos físicos.

Ao longo dos últimos anos, o FOAM vinha sendo desenvolvido em uma linha comercial pela empresa inglesa Nabla, com a idéia de fornecer ao usuário final uma ferramenta para simulações CFD de processos fisicamente complexos e oferecendo serviços de consultoria sobre o uso do *software* e implementação de novos *solvers* com

¹*Solvers* são arquivos executáveis construídos a partir do código fonte e das bibliotecas do OpenFOAM.

modelagem complexa. Contudo, os desenvolvedores chegaram a conclusão de que a abordagem comercial não era a ideal. Assim, em 10 de dezembro de 2004 o código do FOAM se tornou de domínio público sobre a GPL (*Gnu Public License*) e o programa passou a ser chamado de OpenFOAM (referente ao código ser aberto). Segundo os desenvolvedores, esta nova fase de desenvolvimento do código começa com o fim de sua abordagem comercial, reconhecendo que o OpenFOAM é essencialmente uma ferramenta de pesquisa e que o futuro do código só pode ser garantido através de colaborações e interações com o meio acadêmico.

Desde que seu código foi liberado, a comunidade científica têm tratado o OpenFOAM com muito interesse, testando-o para casos *benchmark*, utilizando em projetos de pesquisa e propondo inúmeras colaborações para o desenvolvimento do código. Foi criado um grupo de discussão na internet sobre o OpenFOAM para discutir aspectos de uso, implementação de código e análise numérica do *software* aberto. Um comentário do Prof. Jasak em especial reflete a situação atual do OpenFOAM: “É incrível a popularidade que o OpenFOAM está recebendo da comunidade científica. O código está se espalhando como um poderoso vírus na internet”. É certo que, se o objetivo de usar o código em um ambiente de colaborações for alcançado, o OpenFOAM terá grandes chances de crescer e se tornar uma das maiores plataformas de CFD em código aberto da próxima geração. De fato, grandes empresas multinacionais, como AUDI, Airbus, Bayer, Shell, Honda, entre outras, já visualizaram seu potencial e utilizam o OpenFOAM no desenvolvimento de projetos de alta tecnologia.

O trabalho aqui apresentado se iniciou junto com o lançamento da versão 1.0 do OpenFOAM e acompanhou sua evolução ao longo das novas versões. No momento da escrita deste documento, a última versão era a 1.4.1, sendo esta a utilizada no desenvolvimento final dos códigos.

4.2 Estrutura da Metodologia Numérica

O OpenFOAM possui implementações numéricas específicas para a solução de equações diferenciais parciais e entender seu funcionamento é vital para a implementação (ou extensão) dos códigos. Portanto, esta seção apresenta a metodologia da discretização e solução numérica das equações de transporte e sua interpretação pelo OpenFOAM.

4.2.1 Discretização por Volumes Finitos

A metodologia numérica aplicada na discretização das equações no OpenFOAM é baseada primariamente no método dos volumes finitos [261, 262, 182, 263, 264]. O termo “discretização” se refere à aproximação de um problema em um domínio contínuo por elementos discretos. O método dos volumes finitos, assim como o método das diferenças e elementos finitos, são aplicados na discretização do espaço físico e temporal. O primeiro tipo de discretização considera a divisão do domínio do espaço em um conjunto de volumes de controle interconectados entre si (contíguos) ou com o contorno do domínio. A discretização do espaço físico constrói a chamada malha computacional. De forma similar, a discretização temporal é aplicada a problemas transientes partindo de uma condição inicial e subdivide seu domínio em um número finito de intervalos no tempo. Uma representação de um domínio discretizado está colocada na Fig. 4.1.

Um volume de controle é composto por um conjunto de faces, que por sua vez é composta por um conjunto de arestas. As faces podem conectar dois volumes de controle ou um volume de controle ao contorno do domínio. A Fig. 4.2 apresenta as variáveis relevantes na interconexão de dois volumes de controle vizinhos, P e N .

Note que pela Fig. 4.2 o vetor \mathbf{S} é normal à face f e sua magnitude é igual à área da face. Este vetor aponta do volume de interesse P para o vizinho N . O vetor

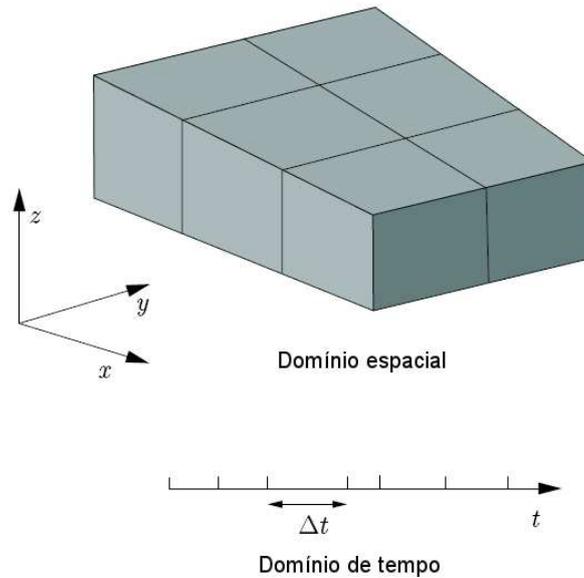


Figura 4.1: Discretização nos domínios de espaço e tempo (retirado de RUSCHE [52]).

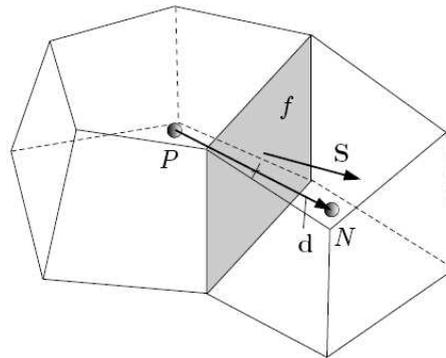


Figura 4.2: Parâmetros na discretização por volumes finitos (retirado de RUSCHE [52]).

unitário \mathbf{n} normal à face é definido como $\mathbf{n} = \frac{\mathbf{S}}{|\mathbf{S}|}$ enquanto que a distância \mathbf{d} entre o centro dos volumes P e N é definida como $\mathbf{d} = \mathbf{x}_N - \mathbf{x}_P$. Uma malha é considerada ortogonal quando \mathbf{d} é paralelo ao vetor \mathbf{S} em cada face da malha.

Um ponto importante na discretização é a alocação do campo de variáveis na malha. Usualmente, todas as variáveis dependentes (efetivamente solucionadas) são alocadas no centro dos volumes. Este tipo de alocação, denominado arranjo co-localizado, se torna a escolha mais óbvia pela sua simplicidade de controle dos índices da malha na implementação computacional. Contudo, problemas de oscilação e avaliação do gradiente de pressão podem aparecer no arranjo co-localizado [262]. Porém, com os avanços das técnicas numéricas [265, 266] estes problemas foram

resolvidos e, desde então, o arranjo co-localizado é usado na maioria dos códigos CFD, inclusive no OpenFOAM.

O processo de discretização por volumes finitos transforma as equações diferenciais de variáveis contínuas em um sistema de equações algébricas correspondente. Partindo de uma equação de transporte para uma variável φ tensorial genérica

$$\frac{\partial(\rho\varphi)}{\partial t} + \nabla \cdot (\rho\mathbf{u}\varphi) = \nabla \cdot (\Gamma\nabla\varphi) + S_\varphi(\varphi) \quad (4.1)$$

onde ρ é a massa específica, \mathbf{u} é o campo de velocidade, Γ é o coeficiente de difusão e $S_\varphi(\varphi)$ é o termo fonte, sua discretização por volumes finitos é obtida integrando-a sobre um volume de controle V_P e em um intervalo de tempo.

$$\int_t^{t+\Delta t} \left[\int_{V_P} \frac{\partial(\rho\varphi)}{\partial t} dV + \int_{V_P} \nabla \cdot (\rho\mathbf{u}\varphi) dV \right] dt = \int_t^{t+\Delta t} \left[\int_{V_P} \nabla \cdot (\Gamma\nabla\varphi) dV + \int_{V_P} S_\varphi(\varphi) dV \right] dt \quad (4.2)$$

O teorema de Gauss é usado para converter as integrais em volume dos termos derivativos no espaço em integrais sobre a superfície S no contorno dos volumes de controle.

$$\int_V (\nabla \cdot \varphi) dV = \int_{\partial V} d\mathbf{S} \cdot \varphi \quad (4.3)$$

O teorema de Gauss é utilizado na Eq. 4.2 para aproximar seus termos pelo somatório dos fluxos das propriedades pelas faces do volume de controle. Este procedimento é detalhado na sequência.

O termo convectivo $\nabla \cdot (\rho\mathbf{u}\varphi)$ da Eq. 4.2 é aproximado convertendo as integrais em volume em superfície usando o teorema de Gauss e somando-as em todo o volume de controle:

$$\begin{aligned} \int_V \nabla \cdot (\rho\mathbf{u}\varphi) dV &= \int_{\partial V} d\mathbf{S} \cdot (\rho\mathbf{u}\varphi) \approx \sum_f \mathbf{S} \cdot (\rho\mathbf{u})_f \varphi_{f(F,M,\gamma)} \\ &= \sum_f F \varphi_{f(F,M,\gamma)} \end{aligned} \quad (4.4)$$

onde F é o fluxo mássico através da face f definido como $F = \mathbf{S} \cdot (\rho\mathbf{u})_f$. O valor da propriedade na face $\varphi_{f(F,M,\gamma)}$ pode ser obtido usando uma função de interpolação

M pré-definida. Esta última usualmente utiliza informações do próprio volume de controle e seus vizinhos, além de requerer o fluxo F sobre a face f e um ou mais parâmetros γ para efetuar a interpolação. Existem vários métodos de interpolação, entre os quais pode-se citar as abordagens de interpolação linear (diferenças centrais), *upwind*, QUICK, MUSCL, TVD (*Total Variation Diminishing*) e NVD (*Normalised Variable Diagram*). A formulação destes métodos encontram-se em vários trabalhos [264, 263, 262, 267] e suas formulações só serão abordadas quando for necessário.

Da mesma forma, o termo difusivo $\nabla \cdot (\Gamma \nabla \varphi)$ da Eq. 4.2 é aproximado usando o teorema de Gauss:

$$\int_V \nabla \cdot (\Gamma \nabla \varphi) dV = \int_{\partial V} d\mathbf{S} \cdot (\Gamma \nabla \varphi) \approx \sum_f \Gamma_f (\mathbf{S} \cdot \nabla_f \varphi) \quad (4.5)$$

considerando Γ como uma variável escalar. Em malhas ortogonais, o gradiente $\nabla_f \varphi$ pode ser obtido pela expressão

$$\nabla_f \varphi = \frac{\varphi_N - \varphi_P}{|\mathbf{d}|}. \quad (4.6)$$

Usando a Eq. 4.6, o gradiente de φ avaliado na face é calculado usando os dois valores centrais localizados entre a face f . Como alternativa, a discretização do gradiente pode ser calculada usando o teorema de Gauss para a integral no volume:

$$\int_V \nabla \varphi dV = \int_{\partial S} d\mathbf{S} \varphi \approx \sum_f \mathbf{S} \varphi_f \quad (4.7)$$

onde o valor de φ_f é obtido através de uma função de interpolação.

Termos fontes, como $S_\varphi(\varphi)$ mostrado na Eq. 4.2, podem ser funções genéricas de φ . Assim, estes termos são linearizados antes da discretização

$$S_\varphi(\varphi) = S_I \varphi + S_E \quad (4.8)$$

sendo que S_I e S_E podem ser dependentes de φ . Este termo integrado no volume de controle resulta em:

$$\int_V S_\varphi(\varphi) dV = S_I \varphi_P V_P + S_E V_P \quad (4.9)$$

Ao assumir que os volumes de controle não variam com o tempo, a Eq. 4.2 pode ser reescrita usando as Eqs. 4.4, 4.5 e 4.9:

$$\int_t^{t+\Delta t} \left[\left(\frac{\partial(\rho\varphi)}{\partial t} \right)_P V_P + \sum_f F \varphi_{f(F,M,\gamma)} \right] dt = \int_t^{t+\Delta t} \left[\sum_f \Gamma_f (\mathbf{S} \cdot \nabla_f \varphi) + (S_I \varphi_P + S_E) V_P \right] dt \quad (4.10)$$

A expressão acima usualmente é dita como a forma semi-discretizada da equação de transporte [263, 66].

Assumindo uma variação linear de $\varphi(t)$ no tempo [263], a discretização da derivada no tempo mostrada na Eq. 4.10 e a integral no tempo podem ser calculadas diretamente como

$$\left(\frac{\partial(\rho\varphi)}{\partial t} \right)_P = \frac{\rho_P^n \varphi_P^n - \rho_P^0 \varphi_P^0}{\Delta t} \quad (4.11)$$

$$\int_t^{t+\Delta t} \varphi(t) dt = \frac{1}{2} (\varphi^0 + \varphi^n) \Delta t \quad (4.12)$$

onde $\varphi^n = \varphi(t+\Delta t)$ representa o novo valor de φ para passo de tempo a ser resolvido enquanto $\varphi^0 = \varphi(t)$ é o valor no passo anterior. Utilizando as Eqs. 4.10, 4.11 e 4.12, obtém-se a forma discretizada, temporal e espacial, da Eq. 4.1.

$$\begin{aligned} \rho_P \frac{\varphi_P^n - \varphi_P^0}{\Delta t} V_P &+ \frac{1}{2} \sum_f F^n \varphi_{f(F,M,\gamma)}^n - \frac{1}{2} \sum_f \Gamma_f^n \mathbf{S} \cdot (\nabla_f \varphi)^n \\ &+ \frac{1}{2} \sum_f F^0 \varphi_{f(F,M,\gamma)}^0 - \frac{1}{2} \sum_f \Gamma_f^0 \mathbf{S} \cdot (\nabla_f \varphi)^0 \\ &= \frac{1}{2} (S_E^n + S_E^0) V_P + \frac{1}{2} (S_I^n \varphi_P^n + S_I^0 \varphi_P^0) V_P \end{aligned} \quad (4.13)$$

A formulação da discretização temporal apresentada na Eq. 4.13 é o método de segunda ordem de Crank-Nicholson. Este método requer os valores de φ e $\nabla\varphi$ no volume de controle e nas suas faces no instante de tempo a ser resolvido e passado. Os valores nas faces são calculados a partir dos valores no centro dos volumes em cada lado da face e usando métodos adequados para a interpolação dos termos convectivos e difusivos.

4.2.2 Sistema Algébrico de Equações

De fato, o objetivo é obter o novo valor de φ_P . Como φ_f e $\nabla_f \varphi$ também dependem do valor de φ nos volumes vizinhos, a Eq. 4.13 pode ser representada como uma equação algébrica válida para o volume P .

$$a_P \varphi_P^n + \sum_N a_N \varphi_N^n = b_P \quad (4.14)$$

Ao considerar todos os volumes de controle da malha, a Eq. 4.14 pode ser escrita como um sistema algébrico de equações,

$$\mathcal{A}\varphi = \mathbf{b} \quad (4.15)$$

onde \mathcal{A} é uma matriz esparsa com coeficientes diagonais a_P e não-diagonais a_N , enquanto que φ e \mathbf{b} são respectivamente os vetores das variáveis φ e os termos fontes referentes ao centro dos volumes. A matriz \mathcal{A} pode ser decomposta em duas matrizes contendo apenas os coeficientes diagonais \mathcal{D} e os não-diagonais \mathcal{N} , tal que:

$$\mathcal{A} = \mathcal{D} + \mathcal{N} \quad (4.16)$$

Portanto, a estrutura da matriz de coeficientes \mathcal{A} pode variar dependendo da dimensão do problema, da forma de discretização das equações e da ordenação dos volumes de controle na malha. A estrutura da matriz é um parâmetro importante na escolha do método mais apropriado para resolver o sistema linear. Em aplicações CFD, cerca de 20% do tempo de computação necessário para resolver um dado problema está associado à solução de sistema linear. Os outros 80% estão associados à obtenção das equações discretizadas.

Os métodos numéricos para solução de sistemas lineares podem ser divididos em duas classes: diretos e iterativos. Os métodos diretos determinam a solução exata, a menos de erros de truncamento, em um número finito de operações matriciais. Eles são indicados para sistemas lineares com um número não muito de grande de equações (até algumas centenas) ou com uma matriz do sistema cuja estrutura permita

um método especial de solução. A limitação destes métodos a sistemas pequenos ocorre por dois motivos. Primeiro, o número de operações (matemáticas) cresce muito rapidamente com o tamanho do sistema, o que aumenta o custo computacional superlinearmente. Além disso, o alto número de operações leva ao aumento dos erros de truncamento, o que degrada a acurácia da solução.

Os métodos iterativos foram desenvolvidos para superar as deficiências dos métodos diretos no tratamento de grandes sistemas. Os métodos iterativos também ganharam espaço devido a sua facilidade de implementação em computadores de alto desempenho com memória compartilhada ou distribuída (supercomputadores e *clusters*), quando comparado aos métodos diretos. Alguns dos métodos iterativos, como o Gauss-Seidel, o Gradiente Conjugado (CG) e suas variantes e o *Algebraic Multigrid* (AMG) podem ser citados [268] e estão implementados no OpenFOAM.

4.2.3 Notação da Formulação Discreta

O processo de discretização pode ser efetuado de forma implícita ou explícita, onde a primeira é utilizada para construir o sistema linear através da discretização. Já segunda forma realiza os cálculos da operação de forma explícita, usando os valores de φ no instante atual. RUSCHE [66] *apud* WELLER [269] introduziu uma notação específica para volumes finitos que permite especificar o tratamento aplicado no processo de discretização.

Com isso, a discretização de um operador \mathcal{L} é representada por $[\mathcal{L}[\varphi]]$. O operador \mathcal{L} pode ser derivativo no espaço ou no tempo, ou mesmo um termo fonte que possui tratamento implícito. A variável dependente φ colocada entre colchetes deve ter tratamento implícito e ser idêntica em todos os termos da equação. Por outro lado, os termos discretizados de forma explícita não são representados entre colchetes. Assim, as operações implícitas e explícitas podem ser diferenciadas usando esta notação. Um resumo é apresentado na Tabela 4.1.

Tabela 4.1: Notação da discretização por volumes finitos.

Termo	Notação por volumes finitos	Termo discretizado
Derivada no tempo	$\left[\frac{\partial \rho[\varphi]}{\partial t} \right]$	$\frac{\rho_P^n \varphi_P^n - \rho_P^0 \varphi_P^0}{\Delta t} V_P$
Convectivo	$[\nabla \cdot (F[\varphi]_{f(F,M,\gamma)})]$	$\sum_f F \varphi_{f(F,M,\gamma)}$
Difusivo	$[\nabla \cdot (\Gamma \nabla[\varphi])]$	$\sum_f F \varphi_{f(F,M,\gamma)}$
Divergente	$\nabla \cdot \varphi$	$\sum_f \mathbf{S} \cdot \varphi_f^0$
Gradiente no volume	$\nabla \varphi$	$\sum_f \mathbf{S} \varphi_f^0$
Fonte Implícito	$[S_I[\varphi]]$	$S_I V_P \varphi^n$
Fonte Explícito	S_E	$S_E V_P$

Seguindo a notação mostrada na Tabela 4.1, o sistema linear obtido pela discretização da equação de transporte (Eq. 4.1) pode ser escrita como colocado abaixo.

$$\mathcal{A} := \left[\frac{\partial \rho[\varphi]}{\partial t} \right] + [\nabla \cdot (F[\varphi]_{f(F,M,\gamma)})] - [\nabla \cdot (\Gamma \nabla[\varphi])] - [S_I[\varphi]] - S_E \quad (4.17)$$

Note que a equação discretizada para cada volume de controle é idêntica àquela mostrada na Eq. 4.2.

Uma vez que o sistema linear é montado, pode-se definir operadores especiais para extrair os coeficientes da matriz e seus termos fonte usando respectivamente $\mathcal{A}_A \equiv \mathcal{A}$ e $\mathcal{A}_S \equiv \mathbf{b}$. Da mesma forma, matrizes contendo os coeficientes diagonais e não diagonais também podem ser obtidas a partir de $\mathcal{A}_D \equiv \mathcal{D}$ e $\mathcal{A}_N \equiv \mathcal{N}$ respectivamente.

O operador “H”, \mathcal{A}_H , definido na sequência, é usado exhaustivamente nos algoritmos numéricos do OpenFOAM. Sua definição parte do método iterativo de Jacobi para obtenção de uma solução aproximada das equações discretizadas como mostrado na Eq. 4.18.

$$\varphi \approx \mathcal{A}_D^{-1} \mathcal{A}_H \quad (4.18)$$

Logo, o operador “H” pode ser obtido utilizando as Eqs. 4.16 e 4.18.

$$\mathcal{A}_H = \mathcal{A}_S - \mathcal{A}_N \varphi \quad (4.19)$$

Definitivamente, o conhecimento das equações, modelos, métodos de discretização, solução e algoritmos é muito mais importante que os conceitos de programação. Com esse ponto de vista, recomendo fortemente que o leitor interessado em desenvolver códigos no OpenFOAM estude a fundo o trabalho de JASAK [263]. Este último apresenta 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 estruturadas, etc., implementados no OpenFOAM. Toda a implementação do código é baseada na teoria apresentada em JASAK [263].

4.3 Conceitos de Programação

Uma das dificuldades inerentes deste trabalho é passar ao leitor os conceitos e a sintaxe básica de programação no OpenFOAM, seja ele leigo em programação ou não. Existem várias fontes disponíveis para o estudo de C++, mas o autor destaca os livros de DEITEL e DEITEL [270] e YANG [271] pela didática clara e exemplos aplicados. Atualmente, as fontes sobre programação no OpenFOAM estão em seus manuais (*User's Guide* e *Programmer's Guide*) e em uma enciclopédia virtual gratuita e comunitária (página tipo *Wiki*²). Apesar de serem o ponto de partida, estas fontes ainda não são suficientes para que o usuário iniciante seja capaz de escrever seu próprio *solver*, sendo necessário um certo esforço para estudar os códigos existentes, a estrutura e o funcionamento dos algoritmos implementados.

²O termo *Wiki* é usado para designar uma coleção de documentos criados e editados de forma colaborativa em uma página da internet. Assim, os visitantes podem modificar os documentos, corrigindo erros, complementando idéias e novas informações. O *Wiki* do OpenFOAM está hospedado em: http://openfoamwiki.net/index.php/Main_Page

Esta seção descreve os conceitos básicos sobre orientação a objetos e sua aplicação na programação usando as bibliotecas do OpenFOAM.

4.3.1 Orientação a Objetos e C++

A maior vantagem na abordagem aplicada à orientação a objetos é remover algumas das falhas encontradas na abordagem seqüencial ou contínua. Na abordagem orientada a objetos, os dados são tratados como elementos críticos do programa, chamados objetos, e não é permitido alterá-los livremente. Os objetos são associados a funções que os acessam e operam, protegendo-os de modificações por uso de funções externas. Uma grande vantagem na abordagem de orientação a objetos é a reusabilidade do código.

Para melhor entendimento da linguagem orientada a objetos, os conceitos de objetos, classes, abstração de dados e encapsulamento, herança e polimorfismo estão colocados na seqüência.

Objetos são as entidades básicas de um sistema orientado a objeto. A programação é analisada em termos de objetos e na forma de comunicação entre eles. Quando um programa é executado, os objetos interagem uns com os outros por envio de mensagens, mesmo sem que estes tenham conhecimento sobre detalhes dos dados ou código. As classes formam uma coleção de objetos similares entre si.

A abstração se refere ao ato de representar aspectos essenciais do programa sem incluir detalhes ou explicações básicas de programação. Classes usam o conceito de abstração e são definidas como uma lista de atributos abstratos. O armazenamento de dados e funções em uma única unidade, intitulada como classe, é chamado encapsulamento. Com isso, os dados não podem ser acessados diretamente e somente as funções encapsuladas na classe podem acessá-los.

Herança é o processo no qual os objetos podem adquirir as propriedades de

objetos de outras classes. Esta característica proporciona a reusabilidade do código, como adicionar novas propriedades a uma classe existente sem modificá-la. Para tal, deriva-se uma nova classe a partir de uma já existente. A nova classe terá aspectos combinados das duas classes.

Por fim, o polimorfismo caracteriza a habilidade de realizar operações com diferentes comportamentos em situações diversas. O comportamento da operação depende do tipo de dado usado na operação. O polimorfismo é usado extensivamente na implementação da herança do código.

Tendo sido colocado essas informações, pode-se descrever algumas vantagens da programação orientada a objetos em relação às abordagens convencionais, como:

- Fornecer uma estrutura modular para programas, facilitando a definição de tipos de dados abstratos onde detalhes da implementação estão escondidos e a unidade possui uma interface claramente definida.
- Tornar mais fácil a manutenção e a modificação de códigos, assim como novos objetos podem ser criados com pequenas diferenças entre os existentes.
- Fornecer uma boa estrutura para bibliotecas de códigos onde os componentes de um software podem ser facilmente adaptados e modificados pelo programador.

Para entender o funcionamento das bibliotecas do OpenFOAM é necessário ter um conhecimento prévio de C++, a linguagem base do OpenFOAM. Esta é uma linguagem orientada a objetos e, portanto possui todas as características descritas acima. Por ser baseada na linguagem precursora C, C++ é uma linguagem de programação apropriada para trabalho científico, devido à rapidez com que os cálculos são efetuados. Contudo, as propriedades inerentes à orientação a objetos em C++ propiciam uma perda de cerca de 10% na velocidade de processamento em relação à linguagem C. Esta perda na eficiência pode variar dependendo da conscientização do programador em relação à efetividade computacional. MALAN e LEWIS [272]

realizaram um estudo sobre a aplicabilidade de códigos escritos em C++ na construção de algoritmos eficazes em cálculos de problemas CFD, aplicando algoritmos que reduzem o tráfego de dados e balanceando o polimorfismo do código com a eficiência computacional.

4.3.2 Interpretação da Linguagem pelo OpenFOAM

A grande vantagem do uso da linguagem matemática é a eficiência em expressar conceitos abstratos. Por exemplo, no escoamento de um fluido, o termo “campo de velocidade” possui um significado mesmo sem qualquer menção à natureza do escoamento ou qualquer dado específico de velocidade. O termo encapsula a idéia de movimento com direção e magnitude e a relação com outras propriedades físicas. Na matemática, pode-se representar o campo de velocidades por um único símbolo, por exemplo, \mathbf{u} , e expressar certos conceitos usando símbolos, por exemplo, o campo de magnitude de velocidade como $|\mathbf{u}|$. Assim, se torna possível expressar conceitos complexos com extrema clareza.

As equações da mecânica do contínuo são usualmente apresentadas como equações diferenciais parciais em 3 dimensões no espaço e com variação no tempo. Estas equações contêm conceitos de escalares, vetores, tensores e seus respectivos campos, e envolvem álgebra tensorial, cálculo tensorial e sistemas de unidades. A solução destas equações envolve procedimentos de discretização, representação de matrizes e implementação de algoritmos de solução de sistemas de equações lineares. A técnica de orientação a objetos usada pelo OpenFOAM permitiu criar de tipos de dados muito próximos aos usados na mecânica do contínuo, e a técnica de sobrecarregamento de operadores permitiu que a simbologia matemática usual fosse aplicada para operações básicas.

As classes implementadas no OpenFOAM declaram tipos e operações associadas que fazem parte da linguagem matemática utilizada na engenharia e no meio científico. O campo de velocidades apresentado anteriormente pode ser representado

no código de programação pelo símbolo \mathbf{U} e a magnitude do campo de velocidade pode ser $\text{mag}(\mathbf{U})$. A velocidade é um campo vetorial e, portanto, deve existir, em um código com orientação a objetos, uma classe `vectorField`. Então, o campo de velocidade pode ser visto como um objeto da classe `vectorField`.

A clareza no uso de objetos na programação para representar objetos físicos e entidades abstratas não deve ser subestimada. 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`. 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.

Assim, utilizando as classes do OpenFOAM, a sintaxe de escrita dos *solvers* se assemelha à solução de equações diferenciais parciais. Por exemplo, a Eq. 4.20

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\Phi \mathbf{u}) - \nabla^2 (\mu \mathbf{u}) = -\nabla p \quad (4.20)$$

é representada pelo código em C++

```

solve
(
    fvm::ddt(rho,U)
  + fvm::div(phi,U)
  - fvm::laplacian(mu,U)
  ==
  - fvc::grad(p)
);

```

onde os *templates* **fvm** e **fvc** referem-se respectivamente a operações de discretização implícita e explícita usando volumes finitos. O sistema linear resultante é resolvido com a função de classe **solve**.

Os códigos dos *solvers* são seqüenciais já que representam um algoritmo de solução e suas equações, que são seqüenciais por natureza. No ponto de vista do autor, os usuários não necessitam de um grande conhecimento de programação orientada a objetos e C++ para alterar códigos simples, mas devem conhecer os princípios por trás da orientação a objetos e ter um conhecimento básico da sintaxe de C++. Contudo, os conceitos de programação são necessários a medida que a complexidade da modelagem e seu algoritmo de solução aumenta. Fato comum em se tratando de problemas CFD.

4.3.3 Implementação de Códigos CFD

Como mencionado anteriormente, as classes e *templates*³ podem encapsular tipos e operações sobre variáveis. As classes devem ser declaradas no início do código através de bibliotecas para que seja possível usar e acessar os comandos presentes na mesma. A principal biblioteca do OpenFOAM está declarada em **fvcFD.H** pois agrega várias outras bibliotecas que são importantes para o pleno funcionamento do código. Em conjunto, estas bibliotecas são capazes de alocar, manipular e operar variáveis de campo interpretadas pelo OpenFOAM.

Na execução de um código CFD no OpenFOAM, primeiramente a geometria e a malha computacional são alocadas usando a classe **polyMesh**, que armazena informações topológicas e geométricas da malha. Estes dados podem ser atualizados durante a simulação, tendo liberdade para apagar informações sobre os volumes de controle (volume, área da faces, posição do centro do volume/face, etc.) e recalculá-las quando for necessário. Assim, é possível apagar ou modificar dados referentes

³A partir desse momento, para qualquer menção às classes deve-se incluir os *templates* implicitamente.

a mudanças topológicas (refinamento de malha) ou geométricas (malhas poliédricas móveis). A classe **fvMesh** é uma extensão da **polyMesh** que inclui dados adicionais necessários para a discretização por volumes finitos.

Em adição, as variáveis de campo e seus contornos são alocadas em posições específicas da malha como centro, faces ou vértices dos volumes de controle usando a classe **geometricField<Type>**. Esta última é construída a partir das classes referentes à malha (**fvMesh**) e ao campo alocado no interior e no contorno da geometria, respectivamente definidas pelas classes **Field<Type>** e **geometricBoundaryField<Type>**. Note que a classe **geometricField<Type>** herda as propriedades das classes com a qual é construída. De fato, esta classe é renomeada para distinguir a posição onde os campos são alocados na malha, definidos como:

- **volField<Type>** – o campo está alocado no centro dos volumes de controle.
- **surfaceField<Type>** – o campo está alocado nas superfícies dos volumes de controle.
- **pointField<Type>** – o campo está alocado nos vértices da malha.

Como citado anteriormente, a discretização das equações é realizada pelo método dos volumes finitos e as classes **fvn** e **fvb** são responsáveis pela aproximação dos termos derivativos das variáveis tensoriais calculadas. Apesar destas classes possuírem o mesmo propósito, suas aplicações são diferentes. A classe **fvn** reúne funções para realizar operações implícitas de discretização, armazenando os resultados em uma matriz esparsa definida pela classe **fvMatrix<Type>**. Em outras palavras, a classe **fvn** discretiza implicitamente os termos e constrói um sistema de equações lineares. Esta classe é capaz de realizar todas as operações que envolvem φ^n colocadas na Tabela 4.1.

Por outro lado, a classe **fvb** agrupa funções para calcular operações explícitas de discretização dos termos presentes nas equações. Assim, esta classe pode realizar cálculos explícitos diretos e seu resultado pode ser usado diretamente no código ou

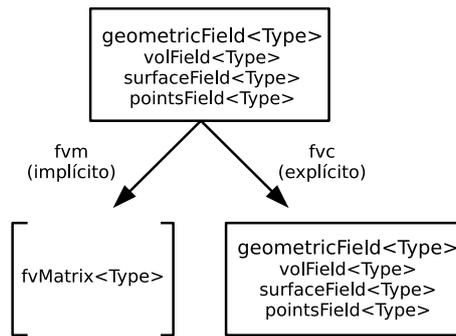


Figura 4.3: Operações da classe `geometricField<Type>` usando operações implícitas (`fvm`) e explícitas (`fvc`).

mesmo alocado no termo fonte do sistema linear. Estas operações de discretização envolvem as variáveis de campo no instante atual, φ^0 , e retornam um campo geométrico `geometricField<Type>`. O OpenFOAM fornece funções para realizar outras operações explícitas além daquelas colocadas na Tabela 4.1, como o rotacional de um campo e a média ponderada no volume de controle obtida a partir dos valores nas faces.

A Fig. 4.3 mostra as diferenças no uso e resultados das classes `fvm` e `fvc` operando sobre um `geometricField<Type>`.

Uma vez que o sistema linear definido pela classe `fvMatrix<Type>` é construído, este pode ser resolvido pela função de classe `solve`. Assim como os métodos e funções de interpolação usadas na discretização das equações, o método numérico para a solução do sistema linear é definido em tempo de execução. Assim, o usuário pode escolher livremente os métodos usados através dos arquivos de configuração do caso simulado.

Todos os códigos implementados no OpenFOAM utilizam os conceitos de programação apresentados nesta seção. Maiores detalhes sobre programação e estrutura de classes no OpenFOAM podem ser encontrados no *Programmer's Guide*.

4.4 Escoamento Multifásico no OpenFOAM

O desenvolvimento do escoamento multifásico no OpenFOAM teve início com as pesquisas realizadas no Imperial College [273, 274, 275, 276, 66, 277]. Nesta sequência de trabalhos foram tratados diversos assuntos, entre os quais a modelagem de escoamentos sólido-líquido [273, 277], líquido-líquido [276, 275] e gás-líquido [274, 66]. Desta forma, o OpenFOAM foi utilizado principalmente como ferramenta computacional [263] para a implementação de novos códigos e para a validação dos algoritmos numéricos multifásicos comparados a dados experimentais [66, 275, 277, 276].

A versão 1.4.1 do OpenFOAM possui códigos específicos para a simulação de escoamentos multifásicos incompressíveis. Uma breve revisão das diferentes abordagens para modelagem multifásica no OpenFOAM é apresentada nesta seção, com ênfase no modelo Euleriano bifásico de RUSCHE [66].

4.4.1 Abordagens Implementadas

Existem três abordagens de modelagem de escoamento multifásico implementadas no OpenFOAM e todas já foram comentadas no Cap. 2. São elas as abordagens Lagrangeana, Euleriana e VOF (*Volume Of Fluid*). A primeira abordagem é aplicada para a solução de escoamentos com pulverizadores (*sprays*), onde as partículas são tratadas de forma Lagrangeana. O *solver dieselFoam* e suas variantes são utilizados basicamente em problemas de combustão e por isso não estão enquadrados na categoria de modelagem multifásica pelos desenvolvedores do OpenFOAM.

A abordagem VOF é aplicada para o acompanhamento de superfície livre, considerando uma função coloração $F(x, y, z, t)$ para indicar a fração de fluido presente em dada posição (x, y, z) e instante t . Dessa forma, é possível acompanhar o deslocamento da interface, considerando que esta é advectada com a velocidade local do fluido. O *solver* básico usando VOF é o *interFoam*, porém este possui variantes

que incluem modelos de turbulência LES (*Large Eddy Simulation*) e RAS (*Reynolds Average Simulation*). Contudo, estes modelos consideram apenas escoamentos bifásicos. Em sua última versão, a versão multifásica do `interFoam`, chamada `multiphaseInterFoam`, foi lançada e pode tratar n fases diferentes.

A abordagem Euleriana-Euleriana considera que as fases são interpenetrantes entre si e introduz o conceito de fração volumétrica na modelagem. Os *solvers* implementados no OpenFOAM que usam esta abordagem são o `settlingFoam`, `bubbleFoam`, `twoLiquidMixingFoam` e `twoPhaseEulerFoam`. Os *solvers* `settlingFoam` e `twoLiquidMixingFoam` são indicados para simular respectivamente a deposição de uma fase dispersa em meio contínuo e a misturação de duas fases líquidas. Já os *solvers* `bubbleFoam` e `twoPhaseEulerFoam` são os mais apropriados para simular escoamentos dispersos, já que os termos de troca de quantidade de movimento entre as fases (arrasto, sustentação e massa virtual) estão incluídos em sua modelagem.

Na verdade, o desenvolvimento do `twoPhaseEulerFoam` basea-se na extensão do código do `bubbleFoam` e, portanto, retém as características básicas do último. Ambos consideram o modelo bifásico Euleriano descrito em RUSCHE [66] e utilizam o modelo $k - \epsilon$ para tratar a turbulência. No caso, o *solver* `twoPhaseEulerFoam` permite ainda que o usuário escolha diferentes modelos para o coeficiente de arrasto e inclui a modelagem da teoria cinética granular [231] utilizada em escoamentos gás-sólido. A modelagem deste último tópico não será revisada devido a sua aplicação específica e fora do contexto desse trabalho. Devido à sua implementação mais geral, a próxima seção se dedica a explicar com mais detalhes a modelagem do *solver* `twoPhaseEulerFoam`.

4.4.2 Modelo Euleriano Bifásico

O equacionamento da modelagem Euleriana implementada no `twoPhaseEulerFoam` está descrita no trabalho de RUSCHE [66]. A formulação para a modelagem mul-

tifásica é baseada no processo de média condicional proposta originalmente por DOPAZO [278] e estendido por WELLER [269] *apud* RUSCHE [66]. No processo de média condicional, as equações são multiplicadas por uma função indicadora (1 na fase dispersa α e 0 na contínua) antes de aplicar as técnicas usuais de média.

As Eqs. 4.21 e 4.22 apresentam as equações que compõem a modelagem Euleriana, obtidas pelo processo de média condicional, considerando fases incompressíveis e sem troca de massa entre fases:

$$\frac{\partial r_\alpha}{\partial t} + \nabla \cdot (r_\alpha \mathbf{u}_\alpha) = 0 \quad (4.21)$$

$$\begin{aligned} \frac{\partial (r_\alpha \mathbf{u}_\alpha)}{\partial t} + \nabla \cdot (r_\alpha \mathbf{u}_\alpha \mathbf{u}_\alpha) &= -\frac{r_\alpha}{\rho_\alpha} \nabla p - \nabla \cdot (r_\alpha \boldsymbol{\tau}_\alpha^{eff}) \\ &+ \frac{\mathbf{M}_\alpha}{\rho_\alpha} + r_\alpha \mathbf{g} \end{aligned} \quad (4.22)$$

onde os termos de troca de *momentum* \mathbf{M}_α e tensão $\boldsymbol{\tau}_\alpha^{eff}$, o qual combina os termos de tensão de Reynolds e viscosa, são originados do processo de média e necessitam de fechamento. De fato, as forças de arrasto, sustentação e massa virtual estão implementados no `twophaseEulerFoam`, que permite a escolha de diferentes modelos para coeficiente de arrasto [232, 279, 38]. Considerando apenas duas fases, a troca de quantidade de movimento entre as fases é $\mathbf{M}_d = -\mathbf{M}_c$, onde o subscrito d representa a fase dispersa e c a contínua.

A modelagem de turbulência descrita em RUSCHE [66] deriva do trabalho de GOSMAN *et al.* [113], que consiste do modelo $k - \epsilon$ com termos fontes adicionais que incorporam efeitos da fase dispersa na turbulência. As equações modificadas do modelo $k - \epsilon$ estão colocadas na seqüência.

$$\frac{\partial k_c}{\partial t} + (\mathbf{u}_c \cdot \nabla) k_c - \nabla \cdot \left(\frac{\nu_c^{eff}}{\sigma_k} \nabla k_c \right) = P_c - \epsilon_c + S_k \quad (4.23)$$

$$\frac{\partial \epsilon_c}{\partial t} + (\mathbf{u}_c \cdot \nabla) \epsilon_c - \nabla \cdot \left(\frac{\nu_c^{eff}}{\sigma_\epsilon} \nabla \epsilon_c \right) = \frac{\epsilon_c}{k_c} (C_1 P_c - C_2 \epsilon_c) + S_\epsilon \quad (4.24)$$

A produção de energia cinética turbulenta é representada por P_c , a viscosidade efetiva da fase contínua ν_c^{eff} é calculada por $\nu_c^{eff} = \nu_c + \nu^{turb}$, onde $\nu^{turb} = C_\mu \frac{k_c^2}{\epsilon_c}$. As

constantes C_1 e C_2 assim como os números de Schmidt σ_k e σ_ϵ permanecem inalterados em relação ao modelo monofásico. Na implementação do `twoPhaseEulerFoam`, RUSCHE [66] considera os termos fonte bifásicos adicionais S_k e S_ϵ nulos, devido a problemas de singularidade. Isto gera uma contradição em relação ao modelo de GOSMAN *et al.* [113], já que os termos fonte contribuem para a turbulência da fase dispersa. RUSCHE [66] destaca que ainda devem ser realizados testes numéricos para avaliar o modelo completo de GOSMAN *et al.* [113].

Por fim, a tensão de Reynolds $\boldsymbol{\tau}_\alpha^{eff}$ para uma fase genérica α é obtida pela hipótese de BOUSSINESQ [71], como mostra a Eq. 4.25.

$$\boldsymbol{\tau}_\alpha^{eff} = -\nu_\alpha^{eff} \left(\nabla \mathbf{u}_\alpha + \nabla \mathbf{u}_\alpha^t - \frac{2}{3} \mathbf{I} \nabla \cdot \mathbf{u}_\alpha \right) + \frac{2}{3} \mathbf{I} k_\alpha \quad (4.25)$$

Contudo, a implementação do modelo de dois fluidos aplicada sem as devidas manipulações pode gerar singularidades e instabilidades numéricas nas equações. De fato, a Eq. 4.22 se torna singular no limite em que $r_\alpha \rightarrow 0$. Desta forma, alguns truques numéricos devem ser usados para evitar este problema, como limitar r_α de modo que $r_\alpha \rightarrow \delta$, onde $0 < \delta \ll 1$. Estes detalhes estão descritos em RUSCHE [66].

A metodologia numérica de solução para o equacionamento descrito acima é baseado no algoritmo PISO [280] para o acoplamento pressão velocidade. Basicamente, o PISO consiste de uma etapa de predição de *momentum* e um laço de correção, no qual a equação da pressão baseada na equação da continuidade volumétrica é resolvida e a quantidade de movimento é corrigida com base na mudança de pressão. Para compor o algoritmo, equações de pressão, fração volumétrica e de correção de *momentum* devem ser derivadas, de modo similar ao descrito em JASAK [263].

De fato, esta formulação é mais apropriada para o desenvolvimento de novos códigos para a simulação de escoamentos polidispersos. Portanto, todos os detalhes da formulação numérica do `twophaseEulerFoam` são abordados apenas no Cap. 5, onde o desenvolvimento de novas metodologias para escoamento multifásico polidisperso são analisadas considerando a extensão do código bifásico de RUSCHE [66].

Capítulo 5

Desenvolvimento das Metodologias

5.1 Introdução

Neste capítulo estão descritas as etapas referentes ao desenvolvimento dos métodos para o acoplamento do modelo multifásico Euleriano com as técnicas de balanço populacional visando a simulação de escoamentos polidispersos. As metodologias foram implementadas usando a ferramenta CFD de código-livre OpenFOAM.

Todos os conceitos apresentados nos capítulos anteriores deste trabalho estão, enfim, agregados aqui. De forma natural, o desenvolvimento ocorre em etapas onde a complexidade de modelagem e sua implementação numérica aumentam de forma gradual. A elaboração das abordagens multifásicas são baseadas no trabalho de RUSCHE [66], que propôs e desenvolveu o modelo Euleriano bifásico implementado no OpenFOAM.

Portanto, a formulação numérica do modelo bifásico deve ser analisada primeiramente para entender e acompanhar as próximas etapas de desenvolvimento. Estas etapas estão colocadas na seção seguinte, onde são abordados os aspectos de formulação e implementação dos algoritmos no OpenFOAM.

A primeira etapa do desenvolvimento refere-se à solução da Equação de Balanço Populacional (EBP) usando o DQMOM [211], onde os detalhes da formulação numérica e sua implementação no OpenFOAM são apresentados.

Considerando que todas as fases dispersas possuem apenas um campo de velocidade, a solução da EBP desenvolvida na etapa anterior pôde, então, ser incorporada ao modelo bifásico de RUSCHE [66]. Esta é a estratégia MUSIG para o acoplamento CFD-EBP.

A implementação do modelo multi-fluido Euleriano se torna necessária uma vez que ainda não existe tal código no OpenFOAM. Sua elaboração parte da extensão do modelo de RUSCHE [66], considerando apenas uma fase contínua e n fases dispersas. Neste caso, cada fase possui seu próprio campo de velocidade, fração volumétrica e diâmetro característico.

Em sua etapa final de desenvolvimento, a solução da EBP pelo DQMOM é acoplada ao modelo multi-fluido elaborado na etapa anterior. Dessa forma, os efeitos de interação entre partículas, como a quebra e agregação, são capazes de afetar localmente as propriedades da população, como diâmetro e fração volumétrica.

5.2 Formulação Numérica do Modelo Bifásico

De modo geral, apenas os detalhes básicos da formulação numérica do modelo bifásico implementado no OpenFOAM estão colocados nesta seção. Sua modelagem já foi apresentada no Cap. 4 e maiores informações podem ser encontradas em RUSCHE [66].

A formulação numérica do modelo bifásico é aplicada para uma fase genérica α , válida para a fase dispersa ou contínua. Especificamente para descrever a modelagem bifásica, a fase dispersa é representada por $\alpha = d$ e $\alpha = c$ para a fase contínua.

De fato, o algoritmo implementado no *solver twoPhaseEulerFoam* inclui a formulação da teoria cinética de escoamento granular aplicada exclusivamente para a simulação de sistemas sólido-gás [231]. Devido ao foco deste trabalho, os detalhes desta formulação não são considerados.

5.2.1 Equações Multifásicas

RUSCHE [66] parte de uma versão incompressível da equação multi-fluido, obtida pela divisão da equação de conservação de quantidade de movimento multifásica, Eq. 4.22, pela massa específica ρ_α e pela fração volumétrica r_α , considerando $\alpha = d, c$.

$$\frac{\partial \mathbf{u}_\alpha}{\partial t} + \mathbf{u}_\alpha \cdot \nabla \mathbf{u}_\alpha + \nabla \cdot (\boldsymbol{\tau}_\alpha^{eff}) + \frac{\nabla r_\alpha}{r_\alpha} \cdot \boldsymbol{\tau}_\alpha^{eff} = -\frac{1}{\rho_\alpha} \nabla p + \frac{\mathbf{M}_\alpha}{r_\alpha \rho_\alpha} + \mathbf{g} \quad (5.1)$$

Utilizando a definição de $\boldsymbol{\tau}_\alpha^{eff}$, mostrada na Eq. 4.25, e após algumas manipulações, a Eq. 5.1 pode ser expressa como:

$$\begin{aligned} & \frac{\partial \mathbf{u}_\alpha}{\partial t} + \nabla \cdot (\mathbf{u}_\alpha \mathbf{u}_\alpha) - \mathbf{u}_\alpha (\nabla \cdot \mathbf{u}_\alpha) \\ & - \nabla \cdot \left(\nu_\alpha^{eff} \frac{\nabla r_\alpha}{r_\alpha} \mathbf{u}_\alpha \right) + \mathbf{u}_\alpha \nabla \cdot \left(\nu_\alpha^{eff} \frac{\nabla r_\alpha}{r_\alpha} \right) - \nu_\alpha^{eff} \nabla^2 \mathbf{u}_\alpha \\ & + \nabla \cdot (\boldsymbol{\tau}_\alpha^{effC}) + \frac{\nabla r_\alpha}{r_\alpha} \cdot \boldsymbol{\tau}_\alpha^{effC} = -\frac{1}{\rho_\alpha} \nabla p + \frac{\mathbf{M}_\alpha}{r_\alpha \rho_\alpha} + \mathbf{g} \end{aligned} \quad (5.2)$$

onde a correção do tensor de Reynolds $\boldsymbol{\tau}_\alpha^{effC}$ está descrita na Eq. 5.3.

$$\boldsymbol{\tau}_\alpha^{effC} = -\nu_\alpha^{eff} \left[(\nabla \mathbf{u}_\alpha)^t - \frac{2}{3} \mathbf{I} \nabla \cdot \mathbf{u}_\alpha \right] + \frac{2}{3} k_\alpha \mathbf{I} \quad (5.3)$$

Neste primeiro momento, apenas o lado esquerdo da Eq. 5.2 é discretizado, sendo que apenas os termos envolvendo \mathbf{u}_α são tratados implicitamente e os restantes por operações explícitas. Utilizando a notação de volumes finitos colocada na Tabela 4.1, a forma discretizada do lado esquerdo da Eq. 5.2, representada por $\boldsymbol{\Upsilon}_\alpha$, é descrita

como:

$$\begin{aligned}
\Upsilon_\alpha &= \left[\frac{\partial[\mathbf{u}_\alpha]}{\partial t} \right] + [\nabla \cdot (\phi_\alpha[\mathbf{u}_\alpha])] - [(\nabla \cdot \phi_\alpha)[\mathbf{u}_\alpha]] \\
&- \left[\nabla \cdot \left(\nu_{\alpha f}^{eff} \frac{\nabla_f^\perp r_\alpha}{r_{\alpha f} + \delta} [\mathbf{u}_\alpha] \right) \right] + \left[\nu_{\alpha f}^{eff} \frac{\nabla_f^\perp r_\alpha}{r_{\alpha f} + \delta} [\mathbf{u}_\alpha] \right] \\
&- \left[\nu_{\alpha f}^{eff} \nabla^2([\mathbf{u}_\alpha]) \right] + \nabla \cdot \boldsymbol{\tau}_\alpha^{effC} + \frac{\nabla r_\alpha}{\langle \bar{r}_\alpha \rangle + \delta} \cdot \boldsymbol{\tau}_\alpha^{effC}
\end{aligned} \tag{5.4}$$

onde ϕ_α é o fluxo volumétrico da fase α , definido como $\phi_\alpha = \mathbf{S} \cdot (\mathbf{u}_\alpha)_f$ e o subscrito f representa a interpolação da variável ao centro das faces. Já $\langle \bar{r}_\alpha \rangle$ representa a média na área da fração volumétrica avaliada nas faces do volume de controle e δ ($0 < \delta \ll 1$) é usado para limitar a fração volumétrica, $r_\alpha \rightarrow \delta$, com o intuito de evitar problemas numéricos quando a fração tende a zero.

A Eq. 5.2 pode ser reescrita em uma forma semi-discretizada para $\alpha = d, c$.

$$\Upsilon_\alpha = -\frac{\nabla p}{\rho_\alpha} + \frac{\boldsymbol{\Omega}_\alpha}{r_\alpha \rho_\alpha} + \mathbf{g} \tag{5.5}$$

Por sua vez, $\boldsymbol{\Omega}_\alpha$ representa a discretização dos termos de interação entre fases, como as forças de arrasto, sustentação e massa virtual. Um tratamento especial na discretização de cada destas forças é necessário para evitar problemas de convergência numérica. Dessa forma, as forças de arrasto e massa virtual recebem um tratamento semi-implícito, enquanto que a força de sustentação é discretizada por operações explícitas. A expressão com a formulação discretizada resultante para a fase dispersa, $\boldsymbol{\Omega}_d$, está colocada na sequência,

$$\begin{aligned}
\boldsymbol{\Omega}_d &= K_d^a(\mathbf{u}_c - [[\mathbf{u}_d]]) + K_d^s \mathbf{u}_r \times (\nabla \times \bar{\mathbf{u}}) \\
&+ K_d^{mv} \left(\frac{D_c \mathbf{u}_c}{Dt} - \left[\frac{D_d [\mathbf{u}_d]}{Dt} \right] \right)
\end{aligned} \tag{5.6}$$

onde $\bar{\mathbf{u}} = r_d \mathbf{u}_d + r_c \mathbf{u}_c$, $\mathbf{u}_r = \mathbf{u}_d - \mathbf{u}_c$ e os termos K_d^a , K_d^s e K_d^{mv} são fatores multiplicativos que envolvem respectivamente os coeficientes das forças de arrasto, sustentação e massa virtual. RUSCHE [66, 112] utiliza fórmulas para esses fatores de modo a considerar a inversão de fases. Contudo, a formulação descrita por RUSCHE [66] e a implementada no OpenFOAM são ligeiramente diferentes e não existe documentação sobre os motivos dessa mudança no código. As equações efetivamente

implementadas no código estão colocadas abaixo.

$$K_d^a = r_c r_d \frac{3}{4} \frac{\rho_c C_D}{d_d} |\mathbf{u}_r| \quad (5.7)$$

$$K_d^s = r_c r_d C_L (r_c \rho_c + r_d \rho_d) \quad (5.8)$$

$$K_d^{mv} = r_c r_d \rho_c C_{MV} \quad (5.9)$$

Os detalhes sobre a inversão de fase e seu efeito sobre as forças de interação podem ser encontrados na literatura [66, 112]. Por fim, o termo de troca de quantidade de movimento referente à fase contínua é expresso como $\mathbf{\Omega}_c = -\mathbf{\Omega}_d$.

Os termos discretizados implicitamente são incorporados à equação discretizada Υ_α , mostrada na Eq. 5.4, assim como as formas explícitas da força de sustentação e massa virtual. Neste processo, obtém-se a equação discretizada corrigida, Υ_α^C , definida abaixo para a fase dispersa $\alpha = d$,

$$\begin{aligned} \Upsilon_d^C &= \Upsilon_d + \left(\frac{K_d^a}{r_d \rho_d} \right)_f [[\mathbf{u}_d]] - \left(\frac{K_d^s}{r_d \rho_d} \right)_f \phi_r \times (\nabla \times \bar{\phi}) \\ &\quad - \left(\frac{K_d^{mv}}{r_d \rho_d} \right)_f \left(\left(\frac{D_c \mathbf{u}_c}{Dt} \right)_f - \left[\frac{D_d [\mathbf{u}_d]}{Dt} \right] \right) \end{aligned} \quad (5.10)$$

e para a fase contínua $\alpha = c$,

$$\begin{aligned} \Upsilon_c^C &= \Upsilon_c + \left(\frac{K_d^a}{r_c \rho_c} \right)_f [[\mathbf{u}_c]] + \left(\frac{K_d^s}{r_c \rho_c} \right)_f \phi_r \times (\nabla \times \bar{\phi}) \\ &\quad + \left(\frac{K_d^{mv}}{r_c \rho_c} \right)_f \left(\left[\frac{D_c [\mathbf{u}_c]}{Dt} \right] - \left(\frac{D_d \mathbf{u}_d}{Dt} \right)_f \right). \end{aligned} \quad (5.11)$$

Portanto, a equação semi-discretizada, mostrada na Eq. 5.5, pode ser reescrita usando as Eqs. 5.10 e 5.11 para a fase dispersa

$$\Upsilon_d^C = -\frac{\nabla p}{\rho_d} + \frac{K_d^a}{r_d \rho_d} \mathbf{u}_c + \mathbf{g} \quad (5.12)$$

e contínua

$$\Upsilon_c^C = -\frac{\nabla p}{\rho_c} + \frac{K_d^a}{r_c \rho_c} \mathbf{u}_d + \mathbf{g} \quad (5.13)$$

Contudo, a solução direta da Eq. 5.5 não garante a continuidade de massa no sistema. Tal objetivo pode ser alcançado através da dedução de uma equação para

pressão obtida de tal forma que a equação da continuidade seja obedecida. A convergência desta equação é atingida pela correção dos fluxos volumétricos nas faces dos volumes de controle. De forma iterativa, os fluxos corrigidos são usados para obter um campo de pressão atualizado até obter sua convergência.

As equações de correção são derivadas a partir da Eqs. 5.12 e 5.13, que podem ser expressas usando os operadores matriciais diagonal $(\cdot)_D$ e “H” $(\cdot)_H$ aplicados à matriz \mathcal{A}_α provinda da discretização de Υ_α^C . Tais operadores foram definidos previamente na Seção 4.2.3. Para a fase dispersa:

$$(\mathcal{A}_d)_D \mathbf{u}_d = (\mathcal{A}_d)_H - \frac{\nabla p}{\rho_d} + \frac{K_d^a}{r_d \rho_d} \mathbf{u}_c + \mathbf{g} \quad (5.14)$$

O rearranjo da Eq. 5.14 fornece uma equação de correção para a quantidade de movimento, que para a fase dispersa é:

$$\mathbf{u}_d = \frac{(\mathcal{A}_d)_H}{(\mathcal{A}_d)_D} - \frac{\nabla p}{\rho_d (\mathcal{A}_d)_D} + \frac{K_d^a}{r_d \rho_d (\mathcal{A}_d)_D} \mathbf{u}_c + \frac{1}{(\mathcal{A}_d)_D} \mathbf{g} \quad (5.15)$$

Seguindo o mesmo procedimento colocado acima, a dedução para a equação de correção para a fase contínua é trivial.

5.2.2 Equação da Pressão

A formulação da equação da pressão deve ser obtida considerando a continuidade das fases. De forma geral, utiliza-se a equação da continuidade da mistura, obtida pelo somatório da equação da continuidade, mostrada na Eq. 4.21, de todas as fases. Desta forma, é possível obter apenas uma única expressão para a pressão. Sua formulação parte, então, da equação da continuidade da mistura interpolada no centro da face f .

$$\nabla \cdot (r_{df} \phi_d + r_{cf} \phi_c) = 0 \quad (5.16)$$

Os fluxos volumétricos, obtidos pela interpolação da Eq. 5.15 ao centro da face

f , são definidos abaixo.

$$\phi_\alpha = \phi_\alpha^* - \left(\frac{1}{\rho_\alpha(\mathcal{A}_\alpha)_D} \right)_f |\mathbf{S}| \nabla_f p \quad (5.17)$$

onde ϕ_d^* é expresso para a fase dispersa ($\alpha = d$) como

$$\phi_d^* = \left(\frac{(\mathcal{A}_d)_H}{(\mathcal{A}_d)_D} \right)_f \cdot \mathbf{S} + \left(\frac{1}{(\mathcal{A}_d)_D} \right)_f \mathbf{g} \cdot \mathbf{S} + \left(\frac{K_d^a}{r_d \rho_d (\mathcal{A}_d)_D} \right)_f \phi_c \quad (5.18)$$

e ϕ_c^* para a fase contínua ($\alpha = c$) como

$$\phi_c^* = \left(\frac{(\mathcal{A}_c)_H}{(\mathcal{A}_c)_D} \right)_f \cdot \mathbf{S} + \left(\frac{1}{(\mathcal{A}_c)_D} \right)_f \mathbf{g} \cdot \mathbf{S} + \left(\frac{K_d^a}{r_c \rho_c (\mathcal{A}_c)_D} \right)_f \phi_d \quad (5.19)$$

Os fluxos volumétricos de ambas as fases são substituídos diretamente na Eq. 5.16 e resultam na equação da pressão colocada abaixo.

$$\nabla \cdot (D_p \nabla p) = \nabla \cdot (r_{df} \phi_d^* + r_{cf} \phi_c^*) \quad (5.20)$$

onde

$$D_p = r_{df} \left(\frac{1}{\rho_d (\mathbf{A}_d)_D} \right)_f + r_{cf} \left(\frac{1}{\rho_c (\mathcal{A}_c)_D} \right)_f \quad (5.21)$$

A equação da pressão, mostrada na Eq. 5.20, é discretizada como um termo difusivo implícito.

$$[\nabla \cdot (D_p \nabla [p])] = \nabla \cdot (r_{df} \phi_d^* + r_{cf} \phi_c^*) \quad (5.22)$$

Assim, o campo de pressão da mistura é obtido de forma a garantir a continuidade dos fluxos volumétricos da mistura. Esta forma de acoplamento entre os campos de pressão e velocidade das fases caracteriza o algoritmo PISO [280].

5.2.3 Equação da Fração Volumétrica

As frações volumétricas são obtidas através da equação da continuidade de cada fase, mostrada na Eq. 4.21. Sua discretização deve ser eficiente, levar a resultados conservativos e que respeitem os limites físicos da fração volumétrica ($0 \leq r_\alpha \leq 1$).

Ao longo dos anos, várias abordagens de solução da fração volumétrica foram propostas e, usualmente, estas apresentavam problemas de divergência numérica, na conservação da fração volumétrica e ao limitar seus valores. De modo a evitar esses problemas, RUSCHE [66] utilizou a formulação de WELLER [269] *apud* RUSCHE [66], que re-arranja a Eq. 4.21 de forma que seja conservativa para todos os termos e limite r_α em ambos contornos inferior e superior. A expressão resultante está colocada abaixo,

$$\frac{\partial r_d}{\partial t} + \nabla \cdot (\bar{\mathbf{u}}r_d) + \nabla \cdot [\mathbf{u}_r r_d r_c] = 0 \quad (5.23)$$

onde $\bar{\mathbf{u}} = r_d \mathbf{u}_d + r_c \mathbf{u}_c$ e $r_c = 1 - r_d$. De fato, esta abordagem fornece ainda um acoplamento implícito entre as fases através da velocidade relativa ($\mathbf{u}_r = \mathbf{u}_d - \mathbf{u}_c$).

A discretização da Eq. 5.23 é realizada de forma implícita em termos de r_d como:

$$\left[\frac{\partial [r_d]}{\partial t} \right] + [\nabla \cdot (\bar{\phi}[r_d]_{(\phi_r, M)})] + [\nabla \cdot (\phi_{rd}[r_d]_{(\phi_r, M)})] = 0 \quad (5.24)$$

onde $\phi_{rd} = r_{cf(-\phi_r, M)}\phi_r$ e $\phi_r = \phi_d - \phi_c$.

O segundo termo da Eq. 5.24 respeita os limites da fração uma vez que o fluxo volumétrico da mistura satisfaz a continuidade. No terceiro termo, os limites de r_d são respeitados usando ϕ_r na discretização do termo convectivo para interpolar r_d para a face e $-\phi_r$ na interpolação de r_c . Desta forma, a conservação da fração volumétrica em ambas as fases depende fortemente do método de interpolação usado. Portanto, o uso de métodos de interpolação de alta ordem com limitadores de fluxo, como TVD ou NVD [263, 267] são requeridos nesta abordagem.

A não linearidade de r_d , presente no terceiro termo da Eq. 5.24, é tratada através de um processo de solução iterativo. Dessa forma, a Eq. 5.24 é resolvida iterativamente até atingir convergência em r_d . RUSCHE [66] ressalta que este processo não é sempre convergente sendo necessário utilizar baixos números de *Courant* (Co). Isto não apresenta grandes limitações, pois baixos valores de passo de tempo já são necessários para a acurácia de simulações transientes.

5.2.4 Modelo $k - \epsilon$ para Turbulência

A turbulência na fase contínua é resolvida com o modelo $k - \epsilon$, cuja formulação foi mostrada nas Eqs. 4.23 e 4.24. Note que a turbulência só é resolvida se ativada pelo usuário.

$$\left[\frac{\partial [k_c]}{\partial t} \right] + [\nabla \cdot (\phi_c [k_c])] - [\nabla \cdot (\phi_c) [k_c]] - \left[\nabla \cdot \left(\frac{\nu_c^{eff}}{\sigma_k} \nabla [k_c] \right) \right] = P_c - \left[\frac{\epsilon_c}{k_c} [k_c] \right] \quad (5.25)$$

$$\left[\frac{\partial [\epsilon_c]}{\partial t} \right] + [\nabla \cdot (\phi_c [\epsilon_c])] - [\nabla \cdot (\phi_c) [\epsilon_c]] - \left[\nabla \cdot \left(\frac{\nu_c^{eff}}{\sigma_\epsilon} \nabla [\epsilon_c] \right) \right] = \left[C_1 P_c \frac{\epsilon_c}{k_c} \right] - \left[C_2 \frac{\epsilon_c}{k_c} [\epsilon_c] \right] \quad (5.26)$$

Por fim, a viscosidade efetiva é calculada para ambas as fases como $\nu_\alpha^{eff} = \nu_\alpha + \nu^{turb}$ com $\alpha = d, c$, onde $\nu^{turb} = C_\mu \frac{k_c^2}{\epsilon_c}$.

5.2.5 Implementação e Algoritmo de Solução

O modelo Euleriano bifásico de RUSCHE [66] está implementado no *solver* `twoPhaseEulerFoam`. Considerando as definições da linguagem de programação no OpenFOAM (ver Seção 4.3), os campos vetoriais, como a velocidade, e escalares, como as frações volumétricas, k e ϵ , são armazenados usando, respectivamente, os *templates* `volVectorField` e `volScalarField`. Assim, essas variáveis são armazenadas no centro dos volumes de controle. Enquanto isso, os fluxos volumétricos são variáveis escalares armazenadas no centro das faces dos volumes e, portanto, são declarados com o *template* `surfaceScalarField`. Por fim, o *template* `phaseModel` armazena as propriedades físicas, como viscosidade e massa específica de cada fase.

Uma vez tendo definido as variáveis de campo e suas propriedades físicas, as etapas para solução do modelo bifásico podem ser definidas. O Algoritmo 1 apresenta as etapas de solução até o final da simulação em t^{fim} , como implementadas no *solver* `twoPhaseEulerFoam`.

Algoritmo 1 Metodologia para o *solver twoPhaseEulerFoam*.

1. Início da simulação.
 2. Leitura de dados da malha, campos de velocidade, fração volumétrica e propriedades físicas.
 3. Laço do passo de tempo t^n .
 - 3.1 Laço para convergência da fração volumétrica com Na passos.
 - 3.1.1 Solução da fração volumétrica para a fase dispersa, Eq. 5.24, e contínua, $r_c = 1 - r_d$.
 - 3.2 Cálculo dos termos de interação entre fases, Ω_α .
 - 3.3 Discretização do lado esquerdo da equação de transporte de quantidade de movimento, Υ_α^C , pelas Eqs. 5.4, 5.10 e 5.11 para $\alpha = d, c$.
 - 3.4 Cálculo dos operadores matriciais $(\mathcal{A}_\alpha)_D$ e $(\mathcal{A}_\alpha)_H$.
 - 3.5 Laço PISO para o acoplamento pressão velocidade com Np passos.
 - 3.5.1 Solução da equação da pressão pela Eq. 5.22.
 - 3.5.2 Correção dos fluxos volumétricos, Eq. 5.17, e velocidades, Eq. 5.15.
 - 3.5.3 Cálculo das frações volumétricas com o procedimento descrito no item 3.1 (só efetuado se estiver ativado pelo usuário).
 - 3.6 Cálculo das Eqs. 5.25 e 5.26 para turbulência (só efetuado se estiver ativado pelo usuário).
 - 3.7 Volta ao item 3 para $n = n + 1$.
 4. Fim da simulação quando $t^n = t^{fim}$.
-

Por filosofia, o OpenFOAM não usa um critério de erro para verificar a convergência dos laços de correção. Assim, fica a cargo do usuário definir o número de iterações nos laços, sendo Na passos para a fração volumétrica e Np para o acoplamento PISO. Isso é um ato que pode comprometer a simulação quando mais interações seriam necessárias para atingir convergência. De fato, os erros associados a essa variável seriam propagados na malha e no tempo levando a resultados inacurados.

Na etapa 3.5.3 do Algoritmo 1, novos valores das frações volumétricas são obtidos com os campos atualizados de pressão e velocidade. Assim, esta etapa permite incluir os efeitos da correção das frações volumétricas no acoplamento PISO.

5.3 Balanço Populacional no OpenFOAM

Como apresentado no Cap. 3, o método numérico híbrido *Direct Quadrature Method Of Moments* [211], chamado DQMOM, foi escolhido como metodologia base para a solução da equação de balanço populacional neste trabalho. Nesta primeira etapa, a solução da EBP usando o DQMOM é implementada no OpenFOAM como um novo *solver* chamado `pbeFoam`.

5.3.1 Formulação do DQMOM

Uma das vantagens intrínsecas do DQMOM é que cada fase dispersa pode ser caracterizada por sua densidade numérica, chamada função peso w_α , e a variável interna considerada, chamada abscissa ξ_α . Em sua formulação, equações de transporte para os pesos e para as abscissas ponderadas, definidas como $\varsigma_\alpha = w_\alpha \xi_\alpha$, são derivadas para cada fase dispersa. A derivação e formulação do DQMOM foi discutida previamente na Seção 3.2.5 deste trabalho.

Neste trabalho, o transporte dos pesos e abscissas ponderadas é considerado pu-

ramente advectivo, devido à natureza do escoamento usualmente tratado. Portanto, suas equações de transporte ficam na forma colocada abaixo, onde \mathbf{u}_α representa a velocidade associada à cada ponto de quadratura.

$$\frac{\partial w_\alpha}{\partial t} + \nabla \cdot (\mathbf{u}_\alpha w_\alpha) = \theta_\alpha \quad (5.27)$$

$$\frac{\partial \zeta_\alpha}{\partial t} + \nabla \cdot (\mathbf{u}_\alpha \zeta_\alpha) = \kappa_\alpha \quad (5.28)$$

$$\alpha = 1, \dots, n$$

Os termos fonte das Eqs. 5.27 e 5.28 são obtidos através da solução do sistema linear colocado da Eq. 5.29.

$$(1 - k) \sum_{\alpha=1}^n \xi_\alpha^k \theta_\alpha + k \sum_{\alpha=1}^n \xi_\alpha^{k-1} \kappa_\alpha = \bar{H}_k^{(n)} \quad (5.29)$$

$$k = 0, \dots, n - 1$$

O termo fonte $\bar{H}_k^{(n)}$ do sistema linear acima agrega os efeitos de interação entre partículas. Considerando apenas os efeitos de quebra e agregação de partículas na formulação do DQMOM, este termo fonte é expresso como:

$$\begin{aligned} \bar{H}_k^{(n)} &= \frac{1}{2} \sum_{\alpha=1}^n \sum_{\beta=1}^n [(\xi_\alpha + \xi_\beta)^k - \xi_\alpha^k - \xi_\beta^k] a(\xi_\alpha, \xi_\beta) w_\alpha w_\beta \\ &+ \sum_{\alpha=1}^n b(\xi_\alpha) w_\alpha [\vartheta(\xi_\alpha) \pi_k(\xi_\alpha) - \xi_\alpha^k] \end{aligned} \quad (5.30)$$

Observe que o sistema linear colocado na Eq. 5.29 deve ser resolvido para cada volume da malha computacional. Após a etapa de solução do sistema linear, os termos fontes das Eqs. 5.27 e 5.28 são obtidos para todo o domínio e os campos escalares de pesos e abscissas ponderadas finalmente podem ser resolvidos.

5.3.2 Implementação Numérica

As abscissas e pesos são representados como campos escalares distribuídos no centro dos volumes de controle da malha computacional. Contudo, cada ponto de quadratura α possui seu próprio campo de abscissas e pesos. Na linguagem do OpenFOAM,

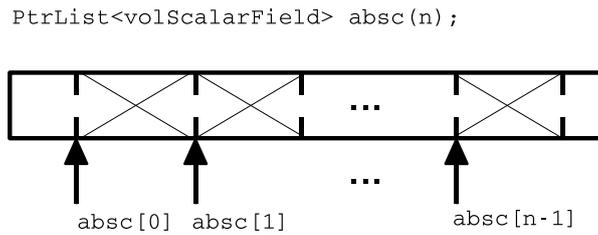


Figura 5.1: Representação de um *template* `PtrList<volScalarField>` com n elementos. Cada seta representa um ponteiro indicando o local onde cada `volScalarField` está alocado na memória.

cada campo é uma variável do tipo `volScalarField` e deve ser definida para cada α .

Em uma implementação mais geral, o número de pontos de quadratura deve ser definido em tempo de execução e, portanto, surge a necessidade de usar uma lista que aloque na memória n campos de variáveis do mesmo tipo. Para tal, o OpenFOAM utiliza o *template* `PtrList<T>` para construir uma lista de classes (ou *templates*) do tipo `T`. Na verdade, este *template* provê uma lista de ponteiros usados para localizar os endereços das classes `T` alocadas seqüencialmente na memória. Assim, o `PtrList<T>` fornece fácil acesso, armazenamento e manipulação de uma lista de classes do tipo `T`.

Assim, todas as variáveis do DQMOM, como pesos, abscissas, abscissas ponderadas e os termos fontes das Eqs. 5.27 e 5.28, foram programados como listas com n elementos usando o *template* `PtrList<T>`. Por exemplo, a definição de `PtrList<volScalarField>` para as abscissas declara uma lista de ponteiros que localiza o endereço na memória de n `volScalarField`. Neste caso, os ponteiros serão endereçados ao `volScalarField` referente às abscissas de cada ponto de quadratura.

A Fig. 5.1 mostra como o *template* `PtrList<T>` estrutura e localiza as classes `T` na memória.

As funções de quebra, $b(\xi_\alpha)$, e agregação, $a(\xi_\alpha, \xi_\beta)$, ambas presentes na Eq. 5.29, são implementadas usando um recurso de programação C++ chamado declaração

virtual. Este recurso permite que diferentes funções com o mesmo propósito sejam implementadas, mas apenas uma delas é usada no momento de execução do código. Ou seja, diferentes modelos podem ser programados e a escolha do desejado ocorre em tempo de execução. As contribuições dos efeitos de quebra e agregação são calculadas para cada volume de controle.

Uma vez tendo declarado as variáveis e funções pertinentes, o algoritmo do DQ-MOM pode ser efetuado. De fato, é necessário criar um laço em que as variáveis possam ser acessadas em cada um dos volumes de controle da malha de modo a montar e resolver o sistema linear colocado na Eq. 5.29. Este sistema é resolvido de forma direta por decomposição LU, método já implementado no OpenFOAM, e sua solução fornece os termos fontes das equações de transporte do DQMOM, θ_α e κ_α , em cada volume de controle da malha. Os termos fontes de cada ponto de quadratura são declarados armazenados acessando seus endereços de memória referentes à posição do volume na malha. Ao final do laço, os termos fonte são, enfim, reconstruídos para todo o domínio.

As Eqs. 5.27 e 5.28 são resolvidas diretamente, discretizando implicitamente todos os termos derivativos, enquanto que os termos fonte são tratados de forma explícita. A forma discretizada das Eqs. 5.27 e 5.28 estão colocadas na sequência:

$$\left[\frac{\partial[w_\alpha]}{\partial t} \right] + [\nabla \cdot (\phi_\alpha[w_\alpha]_{f(\phi, M, \gamma)})] = \theta_\alpha \quad (5.31)$$

$$\left[\frac{\partial[\varsigma_\alpha]}{\partial t} \right] + [\nabla \cdot (\phi_\alpha[\varsigma_\alpha]_{f(\phi, M, \gamma)})] = \kappa_\alpha \quad (5.32)$$

onde $\phi_\alpha = \mathbf{S} \cdot (\mathbf{u}_\alpha)_f$.

Por fim, as etapas de solução do `pbeFoam` estão descritas no Algoritmo 2.

5.4 Abordagem MUSIG

Esta abordagem acopla ao modelo bifásico [66] o uso de balanço populacional para calcular as propriedades da população. Esta se resume à abordagem MUSIG [221]

Algoritmo 2 Metodologia para o *solver* pbeFoam.

1. Início da simulação.
 2. Leitura de dados da malha, campos de pesos e abscissas ponderadas e propriedades físicas.
 3. Laço do passo de tempo t^n .
 - 3.1 Laço varrendo todos os volumes de controle da malha.
 - 3.1.1 Cálculo das funções de quebra e agregação. Cálculo do termo fonte $\bar{H}_k^{(n)}$ (Eq. 5.30).
 - 3.1.2 Solução do sistema linear (Eq. 5.29) usando decomposição LU.
 - 3.1.3 Armazenamento de θ_α e κ_α em cada volume de controle.
 - 3.2 Reconstrução dos termos fonte, θ_α e κ_α , das Eqs. 5.27 e 5.28 em todo o domínio como listas de ponteiros (**PtrList<volScalarField>**) para cada ponto de quadratura.
 - 3.3 Discretização e solução das equações de transporte do DQMOM, Eqs. 5.31 e 5.32, para t^n .
 - 3.4 Cálculo das abscissas $\xi_\alpha = \varsigma_\alpha/w_\alpha$.
 - 3.5 Volta ao item 3 para $n = n + 1$.
 4. Fim da simulação quando $t^n = t^{fim}$.
-

para acoplamento CFD-EBP, onde assume-se que existe apenas um campo de velocidade para todas as classes de partículas, independente de seu tamanho. Esta formulação foi analisada previamente na Seção 3.3.1 deste trabalho.

De fato, a fração volumétrica da fase, r_α , assim como o diâmetro médio de Sauter, d_s , são variáveis importantes na modelagem bifásica. A primeira variável é usada nas equações da continuidade e balanço de quantidade de movimento das fases. Por sua vez, o diâmetro de Sauter é usado no cálculo dos termos de troca de *momentum* entre as fases. Utilizando o volume da partícula como variável interna da EBP é possível obter expressões para r_d e d_s , como mostrado na sequência.

Considerando partículas esféricas, o diâmetro de Sauter é obtido aproximando a Eq. 3.42 por quadratura.

$$d_s = \frac{\int_0^\infty d^3 f(\mathbf{x}, v, t) dv}{\int_0^\infty d^2 f(\mathbf{x}, v, t) dv} \simeq \left(\frac{6}{\pi}\right)^{1/3} \frac{\sum_{\alpha=1}^n \xi_\alpha w_\alpha}{\sum_{\alpha=1}^n \xi_\alpha^{2/3} w_\alpha} \quad (5.33)$$

Ao aproximar a expressão para a fração global da fase dispersa, apresentada na Eq. 3.2 (Cap. 3), com a quadratura de Gordon, a Eq. 5.34 é obtida considerando n fases dispersas.

$$r_d = \sum_{\alpha=1}^n r_\alpha = \int_0^\infty v f(\mathbf{x}, v, t) dv \simeq \sum_{\alpha=1}^n \xi_\alpha w_\alpha = \sum_{\alpha=1}^n \varsigma_\alpha \quad (5.34)$$

Pela Eq. 5.34, nota-se que cada abscissa ponderada é, de fato, a própria fração volumétrica de cada fase dispersa. Assim, a equação de transporte de ς_α , mostrada na Eq. 5.28, pode ser reescrita usando a mesma abordagem colocada na Eq. 5.23 [66]. Sua formulação parte da velocidade da mistura, $\bar{\mathbf{u}} = \mathbf{u}_d \sum_{\alpha=1}^n \varsigma_\alpha + r_c \mathbf{u}_c$, e da velocidade relativa entre as fases, $\mathbf{u}_r = \mathbf{u}_d - \mathbf{u}_c$. Utilizando a velocidade da mistura para explicitar \mathbf{u}_d e a velocidade relativa para eliminar \mathbf{u}_c da expressão, chega-se à Eq. 5.35 após as devidas manipulações.

$$\mathbf{u}_d = \bar{\mathbf{u}} + \mathbf{u}_r - \mathbf{u}_r \sum_{\alpha=1}^n \varsigma_\alpha = \bar{\mathbf{u}} + r_c \mathbf{u}_r \quad (5.35)$$

Ao substituir a velocidade da fase dispersa, mostrada na Eq. 5.35, na Eq. 5.28, a equação de transporte para a abscissa ponderada usada na abordagem MUSIG é

obtida.

$$\begin{aligned} \frac{\partial \varsigma_\alpha}{\partial t} + \nabla \cdot (\mathbf{u}\varsigma_\alpha) + \nabla \cdot (\mathbf{u}_r r_c \varsigma_\alpha) &= \kappa_\alpha \\ \alpha &= 1, \dots, n \end{aligned} \quad (5.36)$$

A discretização da Eq. 5.36 é realizada de forma implícita em termos de ς_α e termo fonte explícito.

$$\begin{aligned} \left[\frac{\partial [\varsigma_\alpha]}{\partial t} \right] + \left[\nabla \cdot (\phi [\varsigma_\alpha]_{f(\phi_r, M, \gamma)}) \right] \\ + \left[\nabla \cdot (\phi_r r_{cf(-\phi_r, M)} [\varsigma_\alpha]_{f(\phi_r, M, \gamma)}) \right] &= \kappa_\alpha \end{aligned} \quad (5.37)$$

5.4.1 Implementação Numérica

Com a solução do DQMOM já implementada na Seção 5.3, a extensão do *solver* bifásico `twoPhaseEulerFoam` se torna simples. Esta abordagem é implementada como um novo *solver*, chamado `musigFoam`. As etapas de solução são as mesmas propostas originalmente por RUSCHE [66], mostradas no Algoritmo 1, substituindo o item 3.1, para solução da fração volumétrica, pelo Algoritmo 3.

Todos os termos dependentes do diâmetro, presentes no Algoritmo 1, são calculados agora usando o diâmetro de Sauter, calculado na etapa 3.1.5 do Algoritmo 3.

5.5 Modelagem Multi-fluido e EBP

Em muitos casos, considerar apenas duas fases e um diâmetro médio para todas as partículas não é adequado para caracterizar sistemas mais complexos. Além disso, com esta abordagem não é possível representar sistemas contendo fases dispersas com propriedades físicas diferentes entre si. Assim, o modelo Euleriano multi-fluido foi implementado tendo o modelo bifásico [66], apresentado na Seção 5.2, como base. Nesta abordagem, consideram-se n fases dispersas e apenas uma fase contínua, onde

Algoritmo 3 Cálculo das propriedades da população pelo DQMOM em substituição à etapa 3.1 do Algoritmo 1.

3.1.1 Laço varrendo todos os volumes de controle da malha.

3.1.1.1 Cálculo das funções de quebra e agregação. Cálculo do termo fonte $\bar{H}_k^{(n)}$ (Eq. 5.30).

3.1.1.2 Solução do sistema linear (Eq. 5.29) usando decomposição LU.

3.1.1.3 Armazenamento de θ_α e κ_α em cada volume.

3.1.2 Reconstrução dos termos fonte, θ_α e κ_α , das Eqs. 5.27 e 5.36 em todo o domínio como listas de ponteiros (**PtrList<volScalarField>**) para cada ponto de quadratura.

3.1.3 Discretização e solução das equações de transporte do DQMOM, Eqs. 5.31 e 5.37.

3.1.4 Cálculo das abscissas $\xi_\alpha = \varsigma_\alpha/w_\alpha$.

3.1.5 Cálculo das propriedades da população, r_d e d_s , usando respectivamente as Eqs. 5.34 e 5.33.

cada fase possui seu próprio campo de velocidade, fração volumétrica e diâmetro característico. Este último é constante no tempo e em todos os pontos do domínio espacial.

Com o intuito de incluir os efeitos de interação entre partículas no modelo multifásico, a implementação do seu acoplamento à solução da EBP usando o DQMOM foi realizada. Assim, as frações volumétricas e diâmetros característicos de cada uma das fases dispersas são obtidas através da solução da EBP. Neste caso, estas variáveis podem sofrer variações no espaço devido aos efeitos de quebra e agregação de partículas.

As próximas seções apresentam a modelagem multi-fluido e sua extensão para

a abordagem multifásica polidispersa. Em seguida, suas implementações no OpenFOAM são descritas.

5.5.1 Equação de Conservação Multifásica

A equação para conservação de quantidade de movimento multifásica, mostrada na Eq. 5.1, usada por RUSCHE [66] também se aplica ao modelo multi-fluido incompressível, considerando $\alpha = 0, \dots, n$. Sua forma semi-discretizada está mostrada na Eq. 5.4. Nas abordagens multifásicas, o subscrito $\alpha = 0$ refere-se à fase contínua enquanto que $\alpha = 1, \dots, n$ representa a fase dispersa.

Na modelagem usada neste trabalho, apenas a troca de *momentum* pela interface entre as fases contínua e dispersas foram consideradas. Desta forma, o termo de troca de quantidade de movimento para a fase contínua possui a contribuição das n fases dispersas, como

$$\mathbf{M}_0 = - \sum_{\alpha=1}^n \mathbf{M}_\alpha \quad (5.38)$$

onde \mathbf{M}_α é o termo de troca de momentum para a fase dispersa,

$$\mathbf{M}_\alpha = \mathbf{M}_\alpha^a + \mathbf{M}_\alpha^s + \mathbf{M}_\alpha^{mv} \quad (5.39)$$

considerando as forças de arrasto, sustentação e massa virtual.

A modelagem e discretização dos termos de interação entre fases requer um tratamento especial, mostrado na próxima seção.

Termos de Interação entre Fases

O termo de interação entre fases, considerando as forças colocadas na Eq. 5.39, é escrita para a fase dispersa como:

$$\begin{aligned} \mathbf{M}_\alpha &= r_\alpha K_\alpha^a [\mathbf{u}_0 - \mathbf{u}_\alpha] + r_\alpha K_\alpha^s \mathbf{u}_{r,\alpha} \times (\nabla \times \mathbf{u}_0) \\ &+ r_\alpha K_\alpha^{mv} \left[\frac{D_0 \mathbf{u}_0}{Dt} - \frac{D_\alpha \mathbf{u}_\alpha}{Dt} \right] \end{aligned} \quad (5.40)$$

onde $\mathbf{u}_{r,\alpha} = \mathbf{u}_0 - \mathbf{u}_\alpha$ e os termos multiplicadores estão definidos na Eq. 5.41, sob a hipótese de partículas esféricas.

$$K_\alpha^a = \frac{3}{4} \frac{\rho_0 C_{D,\alpha}}{d_\alpha} |\mathbf{u}_{r,\alpha}|, \quad K_\alpha^s = \rho_0 C_{L,\alpha}, \quad K_\alpha^{mv} = \rho_0 C_{MV,\alpha} \quad (5.41)$$

Os termos multiplicadores apresentados na Eq. 5.41 diferem da formulação usada por RUSCHE [66]. Desta forma, a formulação usual das forças de interação foi adotada neste trabalho [56, 51, 38], pois considera-se que as fases são bem definidas e não existe inversão de fases.

O termo de troca de quantidade de movimento para a fase contínua é expresso considerando sua interação com todas as fases dispersas, sendo derivado diretamente através das Eqs. 5.38, 5.39 e 5.40

$$\begin{aligned} \mathbf{M}_0 &= \sum_{\alpha=1}^n r_\alpha K_\alpha^a [\mathbf{u}_\alpha - \mathbf{u}_0] - \sum_{\alpha=1}^n r_\alpha K_\alpha^s \mathbf{u}_{r,\alpha} \times (\nabla \times \mathbf{u}_0) \\ &+ \sum_{\alpha=1}^n r_\alpha K_\alpha^{mv} \left[\frac{D_\alpha \mathbf{u}_\alpha}{Dt} - \frac{D_0 \mathbf{u}_0}{Dt} \right] \end{aligned} \quad (5.42)$$

Como no modelo bifásico de RUSCHE [66], os termos de troca de quantidade de movimento entre fases são discretizados de forma semi-implícita (ver Seção 5.2.1). Assim, a velocidade presente no termo de arrasto e na derivada substantiva da massa virtual recebe tratamento implícito para a fase considerada. Por sua vez, os termos de sustentação e a derivada substantiva na massa virtual referente à outra fase são discretizados de forma explícita. Seguindo a mesma metodologia mostrada na Seção 5.2.1, a forma discretizada do lado esquerdo da equação de conservação de quantidade de movimento multifásica, Eq. 5.4, é obtida para as fases contínua ($\alpha = 0$) e dispersas ($\alpha = 1, \dots, n$).

$$\begin{aligned} \Upsilon_0^C &= \Upsilon_0 + \left(\frac{1}{r_0 \rho_0} \right)_f [[\mathbf{u}_0]] \sum_{\alpha=1}^n r_{\alpha f} K_{\alpha f}^a + \left(\frac{1}{r_0 \rho_0} \right)_f \sum_{\alpha=1}^n r_{\alpha f} K_{\alpha f}^s \phi_{r,\alpha} \times (\nabla \times \bar{\phi}) \\ &+ \left(\frac{1}{r_0 \rho_0} \right)_f \sum_{\alpha=1}^n r_{\alpha f} K_{\alpha f}^{mv} \left(\left[\frac{D_0 [\mathbf{u}_0]}{Dt} \right] - \left(\frac{D_\alpha \mathbf{u}_\alpha}{Dt} \right)_f \right) \end{aligned} \quad (5.43)$$

$$\begin{aligned} \Upsilon_\alpha^C &= \Upsilon_\alpha + \left(\frac{K_\alpha^a}{\rho_\alpha} \right)_f [[\mathbf{u}_d]] - \left(\frac{K_\alpha^s}{\rho_\alpha} \right)_f \phi_{r,\alpha} \times (\nabla \times \bar{\phi}) \\ &\quad - \left(\frac{K_\alpha^{mv}}{\rho_\alpha} \right)_f \left(\left(\frac{D_0 \mathbf{u}_0}{Dt} \right)_f - \left[\frac{D_\alpha [\mathbf{u}_\alpha]}{Dt} \right] \right) \end{aligned} \quad (5.44)$$

A forma corrigida e semi-discretizada da equação de quantidade de movimento, Υ_α^C , obtida para as fases contínua e dispersas são escritas como mostrado respectivamente nas Eqs. 5.45 e 5.46.

$$\Upsilon_0^C = -\frac{\nabla p}{\rho_0} + \mathbf{g} + \frac{1}{r_0 \rho_0} \sum_{\alpha=1}^n r_\alpha K_\alpha^d \mathbf{u}_\alpha \quad (5.45)$$

$$\Upsilon_\alpha^C = -\frac{\nabla p}{\rho_\alpha} + \mathbf{g} + \frac{K_\alpha^d}{\rho_\alpha} \mathbf{u}_0 \quad (5.46)$$

As equações de correção para a quantidade de movimento das fases são obtidas usando o mesmo procedimento aplicado ao modelo Euleriano bifásico (Seção 5.2.1). As equações de correção para as fases dispersa e contínua estão colocadas respectivamente nas Eqs. 5.47 e 5.48.

$$\mathbf{u}_\alpha = \frac{(\mathcal{A}_\alpha)_H}{(\mathcal{A}_\alpha)_D} - \frac{\nabla p}{\rho_\alpha (\mathcal{A}_\alpha)_D} + \frac{1}{(\mathcal{A}_\alpha)_D} \mathbf{g} + \frac{K_\alpha^a}{\rho_\alpha (\mathcal{A}_\alpha)_D} \mathbf{u}_0 \quad (5.47)$$

$$\mathbf{u}_0 = \frac{(\mathbf{A}_0)_H}{(\mathbf{A}_0)_D} - \frac{\nabla p}{\rho_0 (\mathbf{A}_0)_D} + \frac{1}{(\mathbf{A}_0)_D} \mathbf{g} + \frac{1}{r_0 \rho_0 (\mathbf{A}_0)_D} \sum_{\alpha=1}^n r_\alpha K_\alpha^a \mathbf{u}_\alpha \quad (5.48)$$

Contudo, sabe-se que a solução das equações apresentadas acima não garante a continuidade. Desta forma, a equação multifásica da pressão deve ser formulada de forma a satisfazer a continuidade.

5.5.2 Equação Multifásica da Pressão

A solução da equação da pressão fornece correções para atualizar iterativamente os campos de pressão, fluxos volumétricos e velocidades, de forma que a continuidade seja respeitada [66]. Para tal, a equação incompressível da continuidade da mistura avaliada nas faces dos volumes de controle é usada na abordagem multifásica.

$$\nabla \cdot \left(\sum_{\alpha=0}^n r_{\alpha f} \phi_\alpha \right) = 0 \quad (5.49)$$

Os fluxos volumétricos, ϕ_α , são obtidos pela interpolação das equações de correção, Eqs. 5.47 e 5.48, para o centro das faces, i. e., $\phi_\alpha = \mathbf{S} \cdot (\mathbf{u}_\alpha)_f$. De forma geral, os fluxos volumétricos para $\alpha = 0, \dots, n$ podem ser escritos como mostrado na Eq. 5.50.

$$\phi_\alpha = \phi_\alpha^* - \left(\frac{1}{\rho_\alpha (\mathcal{A}_\alpha)_D} \right)_f |\mathbf{S}| \nabla_f p \quad (5.50)$$

onde ϕ_0^* refere-se à fase contínua ($\alpha = 0$), como

$$\phi_0^* = \left(\frac{(\mathbf{A}_0)_H}{(\mathbf{A}_0)_D} \right)_f \cdot \mathbf{S} + \left(\frac{1}{(\mathbf{A}_0)_D} \right)_f \mathbf{g} \cdot \mathbf{S} + \left(\frac{1}{r_0 \rho_0 (\mathbf{A}_0)_D} \right)_f \sum_{\alpha=1}^n r_{\alpha f} K_{\alpha f}^a \phi_\alpha \quad (5.51)$$

e ϕ_α^* é expresso para as fases dispersas ($\alpha = 1, \dots, n$)

$$\phi_\alpha^* = \left(\frac{(\mathcal{A}_\alpha)_H}{(\mathcal{A}_\alpha)_D} \right)_f \cdot \mathbf{S} + \left(\frac{1}{(\mathcal{A}_\alpha)_D} \right)_f \mathbf{g} \cdot \mathbf{S} + \left(\frac{K_\alpha^a}{\rho_\alpha (\mathcal{A}_\alpha)_D} \right)_f \phi_0. \quad (5.52)$$

A equação multifásica para pressão é construída ao substituir os fluxos volumétricos das fases contínua e dispersas, mostrados nas Eqs. 5.50, 5.51 e 5.52, na Eq. 5.49. A equação resultante é mostrada abaixo.

$$\nabla \cdot (D_p \nabla p) = \nabla \cdot \left(r_{0f} \phi_0^* + \sum_{\alpha=1}^n r_{\alpha f} \phi_\alpha^* \right) \quad (5.53)$$

onde

$$D_p = r_{0f} \left(\frac{1}{\rho_0 (\mathbf{A}_0)_D} \right)_f + \sum_{\alpha=1}^n r_{\alpha f} \left(\frac{1}{\rho_\alpha (\mathcal{A}_\alpha)_D} \right)_f \quad (5.54)$$

Assim como na abordagem bifásica, a equação multifásica para a pressão, mostrada na Eq. 5.53, é discretizada de forma implícita como um termo difusivo.

O campo de pressão para a mistura é determinado considerando a continuidade dos fluxos volumétricos. Estes fluxos, mostrados na Eq. 5.50, são corrigidos após a solução da pressão, usando a Eq. 5.53, onde um procedimento iterativo PISO é necessário para atingir convergência entre estes campos [263].

5.5.3 Equação para a Fração Volumétrica Multifásica

A equação para a fração volumétrica multifásica foi derivada de forma similar à proposta em RUSCHE [66]. A partir da velocidade da mistura, $\bar{\mathbf{u}} = \sum_{\alpha=0}^n r_\alpha \mathbf{u}_\alpha$,

uma expressão para \mathbf{u}_α foi obtida

$$\mathbf{u}_\alpha = \bar{\mathbf{u}} + r_0 \mathbf{u}_{r,0} + \sum_{\substack{i=1 \\ i \neq \alpha}}^n r_i \mathbf{u}_{r,i} \quad (5.55)$$

usando a velocidade relativa entre fases $\mathbf{u}_{r,0} = \mathbf{u}_\alpha - \mathbf{u}_0$ e $\mathbf{u}_{r,i} = \mathbf{u}_\alpha - \mathbf{u}_i$.

A expressão para \mathbf{u}_α , apresentada na Eq. 5.55, foi substituída na equação incompressível da continuidade da fase α , Eq. 4.21, resultando na equação multifásica para as frações volumétricas.

$$\frac{\partial r_\alpha}{\partial t} + \nabla \cdot (\bar{\mathbf{u}} r_\alpha) + \nabla \cdot (r_0 \mathbf{u}_{r,0} r_\alpha) + \nabla \cdot \left(\sum_{\substack{i=1 \\ i \neq \alpha}}^n r_i \mathbf{u}_{r,i} r_\alpha \right) = 0 \quad (5.56)$$

Note que, esta formulação propicia o acoplamento entre as fases contínua e dispersas através das velocidades relativas presentes na Eq. 5.56.

A discretização da Eq. 5.56 é totalmente implícita em relação à r_α . Note que a não-linearidade das frações volumétricas na Eq. 5.56 requer um procedimento iterativo para atingir convergência.

$$\left[\frac{\partial [r_\alpha]}{\partial t} \right] + \left[\nabla \cdot (\bar{\phi} [r_\alpha]_{(\phi_r, M)}) \right] + \left[\nabla \cdot (r_{0f} \phi_{r,0} [r_\alpha]_{(\phi_r, M)}) \right] + \left[\nabla \cdot \left(\sum_{\substack{i=1 \\ i \neq \alpha}}^n r_{if} \phi_{r,i} [r_\alpha]_{(\phi_r, M)} \right) \right] = 0 \quad (5.57)$$

onde o fluxo relativo entre as fases dispersa e contínua é $\phi_{r,0} = \phi_\alpha - \phi_0$ e entre as fases dispersas é $\phi_{r,i} = \phi_\alpha - \phi_i$.

5.5.4 Acoplamento CFD-EBP Multifásico

O acoplamento do modelo multi-fluido à solução do balanço populacional é realizado usando os pesos e abscissas do DQMOM para obter as informações da população de partículas.

Como visto na Eq. 5.34, cada abscissa ponderada, ς_α , é a própria fração volumétrica da fase dispersa para a classe α . Seguindo o mesmo procedimento aplicado à equação da fração volumétrica multifásica, a equação de transporte do DQMOM para a abscissa ponderada, Eq. 5.28, foi modificada como mostra a Eq. 5.58.

$$\frac{\partial \varsigma_\alpha}{\partial t} + \nabla \cdot (\bar{\mathbf{u}}_\alpha \varsigma_\alpha) + \nabla \cdot (\mathbf{u}_{r,0} \varsigma_\alpha) - \nabla \cdot \left(\sum_{i=1}^n \varsigma_i \mathbf{u}_{r,0} \varsigma_\alpha \right) + \nabla \cdot \left(\sum_{\substack{i=1 \\ i \neq \alpha}}^n \varsigma_i \mathbf{u}_{r,i} \varsigma_\alpha \right) = \kappa_\alpha \quad (5.58)$$

No modelo multifásico polidisperso, a solução do sistema linear do DQMOM, mostrado na Eq. 5.29, fornece os termos fonte para as equações de transporte dos pesos e das frações volumétricas das fases dispersas (abscissas ponderadas), mostradas respectivamente nas Eqs. 5.27 e 5.58. Uma vez que a solução da quadratura é obtida, os termos de interação entre fases são calculados em cada volume de controle da malha usando o diâmetro característico das fases dispersas, d_α , considerando partículas esféricas.

$$d_\alpha = \left(\frac{6\xi_\alpha}{\pi} \right)^{1/3} \quad (5.59)$$

Por fim, a Eq. 5.58 é discretizada de forma implícita em termos de ς_α , considerando κ_α como um termo fonte explícito.

$$\begin{aligned} & \left[\frac{\partial [\varsigma_\alpha]}{\partial t} \right] + \left[\nabla \cdot (\bar{\phi} [\varsigma_\alpha] f(\phi_r, M, \gamma)) \right] + \left[\nabla \cdot (\phi_{r,0} [\varsigma_\alpha] f(\phi_r, M, \gamma)) \right] \\ & - \left[\nabla \cdot \left(\sum_{i=1}^n \varsigma_i \phi_{r,0} [\varsigma_\alpha] f(\phi_r, M, \gamma) \right) \right] \\ & + \left[\nabla \cdot \left(\sum_{\substack{i=1 \\ i \neq \alpha}}^n \varsigma_i \phi_{r,i} [\varsigma_\alpha] f(\phi_r, M, \gamma) \right) \right] = \kappa_\alpha \end{aligned} \quad (5.60)$$

5.5.5 Implementação das Abordagens

Como nos desenvolvimentos anteriores, a classe **PtrList**<**T**> foi usada para implementar e alocar uma lista de ponteiros sobre a classe **T**. Neste caso, este *template* também foi usado para alocar o resultado da discretização das

n equações de conservação para as fases dispersas. Desta forma, os campos referentes às fases dispersas são definidos como `PtrList<volVectorField>` para as velocidades, `PtrList<volScalarField>` para as frações volumétricas e `PtrList<surfaceScalarField>` para os fluxos volumétricos.

A solução dos modelos multifásicos segue etapas similares às da metodologia proposta por RUSHE [66], mostrado no Algoritmo 1. Contudo, o número de iterações nos laços de convergência no algoritmo bifásico eram fixos e definidos pelo usuário. Em contraste, um critério misto entre as tolerâncias absoluta, ε_{abs} , e relativa, ε_{rel} , mostrado na Eq. 5.61 para uma variável genérica ϕ , foi implementado nos *solvers* multifásicos para controlar os laços de convergência. As tolerâncias são definidas pelo usuário.

$$\chi = \max \left[\frac{|\phi^{it} - \phi^{it-1}|}{\frac{\varepsilon_{abs}}{\varepsilon_{rel}} + |\phi^{it}|} \right] < \varepsilon_{rel} \quad (5.61)$$

A implementação destas abordagens no OpenFOAM gerou dois novos *solvers*, o `multiPhaseEulerFoam`, que é a extensão do modelo bifásico para a formulação multi-fluido com $n + 1$ fases, e o `multiPhasePbeFoam`, que acopla os efeitos entre partículas ao primeiro. O Algoritmo 4 apresenta as etapas de solução do *solver* `multiPhaseEulerFoam` até o final da simulação em t^{fim} .

O acoplamento da solução da EBP pelo DQMOM ao modelo multi-fluido ocorre na etapa 3.1 do Algoritmo 4, onde a fração volumétrica das fases é calculada. No acoplamento CFD-EBP, os efeitos de interação entre partículas são incluídos nesta etapa com a solução do DQMOM. Portanto, a solução do *solver* `multiPhasePbeFoam` é igual àquela apresentada no Algoritmo 4, com substituição da etapa 3.1 pela solução do DQMOM apresentada no Algoritmo 5.

O desenvolvimento do modelo multifásico e seu acoplamento com a EBP usando DQMOM foi apresentado na *OpenFOAM International Conference 2007*, que ocorreu na Inglaterra. Maiores detalhes de sua implementação no OpenFOAM estão descritos no artigo da conferência, exposto no Apêndice C.

Algoritmo 4 Metodologia para a solução do modelo multi-fluido no *solver* `multiPhaseEulerFoam`.

1. Início da simulação.
 2. Leitura de dados da malha, campos de velocidade, fração volumétrica e propriedades físicas.
 3. Laço do passo de tempo t^n .
 - 3.1 Laço para convergência da fração volumétrica até atingir a tolerância especificada para a fração volumétrica global da fase dispersa, Eq. 5.61, ou Na passos no máximo.
 - 3.1.1 Solução da fração volumétrica para a fase dispersa, Eq. 5.57, e contínua, $r_0 = 1 - \sum_{\alpha=1}^n r_\alpha$.
 - 3.2 Cálculo dos termos de interação entre fases para as fases dispersas, Eq. 5.40, e para a contínua, Eq. 5.42.
 - 3.3 Discretização do lado esquerdo da equação de transporte de quantidade de movimento, Υ_α^C , pelas Eqs. 5.43 e 5.44 para $\alpha = 0, \dots, n$.
 - 3.4 Cálculo dos operadores matriciais $(\mathcal{A}_\alpha)_D$ e $(\mathcal{A}_\alpha)_H$.
 - 3.5 Laço PISO para o acoplamento pressão velocidade até que a condição da Eq. 5.61 seja satisfeita para a pressão ou Np iterações no máximo.
 - 3.5.1 Solução da forma discretizada da equação multifásica da pressão, Eq. 5.53.
 - 3.5.2 Correção dos fluxos volumétricos, Eqs. 5.50, 5.51 e 5.52, e velocidades, Eqs. 5.47 e 5.48.
 - 3.5.3 Cálculo das frações volumétricas com o procedimento descrito no item 3.1 (só efetuado se estiver ativado pelo usuário).
 - 3.6 Cálculo das Eqs. 5.25 e 5.26 para turbulência (só efetuado se estiver ativado pelo usuário).
 - 3.7 Volta ao item 3 para $n = n + 1$.
 4. Fim da simulação quando $t^n = t^{fim}$.
-

Algoritmo 5 Cálculo das propriedades da população pelo DQMOM em substituição à etapa 3.1 do Algoritmo 4.

3.1.1 Laço varrendo todos os volumes de controle da malha.

3.1.1.1 Cálculo das funções de quebra e agregação. Cálculo do termo fonte $\bar{H}_k^{(n)}$ (Eq. 5.30).

3.1.1.2 Solução do sistema linear (Eq. 5.29) usando decomposição LU.

3.1.1.3 Armazenamento de θ_α e κ_α para cada volume de controle.

3.1.2 Reconstrução dos termos fonte, θ_α e κ_α , das Eqs. 5.27 e 5.58 em todo o domínio como listas de ponteiros (**PtrList**<volScalarField>) para cada ponto de quadratura.

3.1.3 Discretização e solução das equações de transporte do DQMOM, Eqs. 5.31 e 5.60.

3.1.4 Cálculo das abscissas $\xi_\alpha = \varsigma_\alpha/w_\alpha$.

3.1.5 Cálculo das propriedades características da fase dispersa, r_α e d_α , usando respectivamente as Eqs. 5.34 ($r_\alpha = \varsigma_\alpha$) e 5.59.

Capítulo 6

Resultados e Discussão

6.1 Introdução

O desenvolvimento das metodologias para escoamento polidisperso foram realizadas em etapas bem distintas. Em cada uma destas etapas, foram gerados resultados importantes para indicar os rumos deste trabalho.

Na sequência, são apresentados os resultados para a avaliação dos métodos híbridos para solução da EBP, indicando os fatores pelos quais o DQMOM se destacou. Em seguida, os resultados da implementação do DQMOM no OpenFOAM, avaliado em casos 0D transientes e 1D estacionários, e para o acoplamento MUSIG são apresentados. Por fim, são mostrados os resultados para a implementação multifásica no OpenFOAM e a abordagem com acoplamento CFD-EBP.

6.2 Comparação e Avaliação dos Métodos Híbridos

Uma avaliação dos métodos híbridos QMOM [208], DQMOM [211] e PPDC [212] foi realizada visando a escolha do melhor método numérico para a solução da EBP. A

acurácia destes métodos foi testada comparando seus resultados com soluções analíticas [158, 160] aplicadas a problemas transientes de agregação e quebra simultânea e equivalente, quebra dominante e agregação dominante sem dependência espacial. O método das classes de pivô fixo [184] também foi avaliado, com o intuito de verificar seu comportamento frente aos métodos híbridos. Um simulador de balanço populacional foi implementado em FORTRAN 90 para avaliar esses métodos.

O resultado desta comparação propiciou a elaboração de um artigo, submetido recentemente para publicação. A versão de submissão do trabalho se encontra no Apêndice A.

A análise dos resultados apresentados no Apêndice A mostrou a grande superioridade em acurácia dos métodos QMOM e DQMOM. De modo geral, estes dois métodos se comportam de forma similar, atingindo ótima acurácia ao avaliar os erros relativos dos momentos da distribuição, $\chi_{R,k}$.

$$\chi_{R,k} = \left| \frac{\mu_k(t) - \mu_k^{(a)}(t)}{\mu_k^{(a)}(t)} \right| \quad (6.1)$$

A Fig. 6.1 apresenta o comportamento dos erros relativos dos momentos para o caso de quebra dominante usando o QMOM, DQMOM, PPDC e o método das classes (MoC).

Contudo, verifica-se pela comparação dos resultados apresentados na Fig. 6.1(a) que, como o QMOM integra diretamente os momentos, seus erros relativos são inferiores aos obtidos pelo DQMOM para os dois primeiros momentos. O erro associado à aproximação da quadratura no cálculo destes momentos afeta a acurácia do DQMOM. Para os momentos de maior ordem, as acurácias do QMOM e DQMOM são bastante similares, como mostra a Fig. 6.1(b). De fato, a diferença de precisão não influencia a acurácia global de ambos métodos nos casos analisados.

Em contraste, verifica-se pelas Figs. 6.1(c) e (d) que os erros relativos dos momentos obtidos usando os métodos PPDC e MoC são maiores dos que aqueles obtidos pelo QMOM e DQMOM.

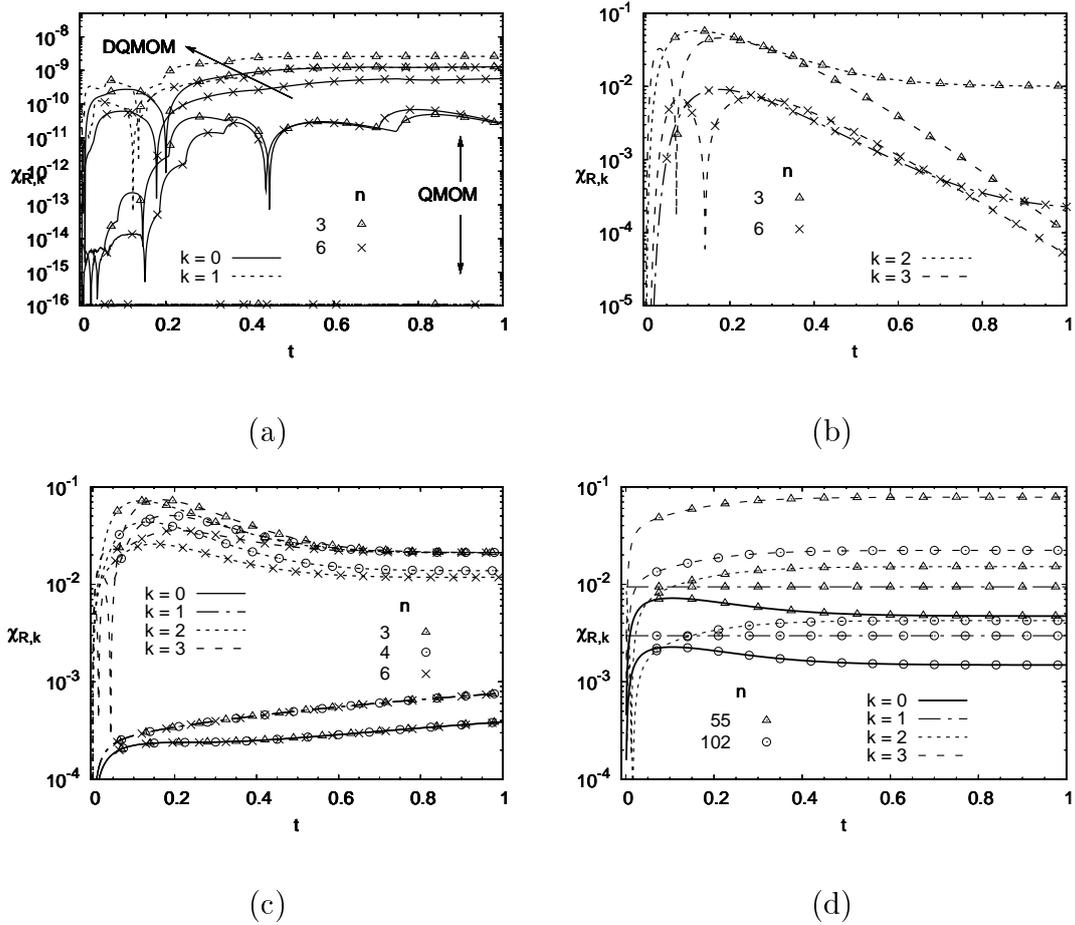


Figura 6.1: Erro relativo dos momentos no caso de quebra dominante para os momentos de menor ordem, $\mu_0 - \mu_1$, (a) e maior ordem, $\mu_2 - \mu_3$, (b) usando o QMOM e DQMOM, PPDC (c) e MoC (d).

Uma das limitações do PPDC está no fato deste apresentar falsa convergência em seus resultados. Neste caso, a solução numérica usando diferentes pontos de quadratura converge entre si, mas não com a solução analítica. Este comportamento pode ser verificado na Fig. 6.2, onde os resultados numéricos dos momentos 2 e 3 não convergem para a solução analítica ao usar diferentes valores de pontos de quadratura.

Para verificar o comportamento global dos métodos, definiu-se um erro médio envolvendo os seis primeiros momentos,

$$\chi = \sqrt{\frac{1}{6T} \sum_{l=1}^T \sum_{k=0}^5 \left| \frac{\mu_k(t_l) - \mu_k^{(a)}(t_l)}{\mu_k^{(a)}(t_l)} \right|^2} \quad (6.2)$$

onde o superscrito (a) refere-se ao valor analítico do momento [158, 160] e T é número

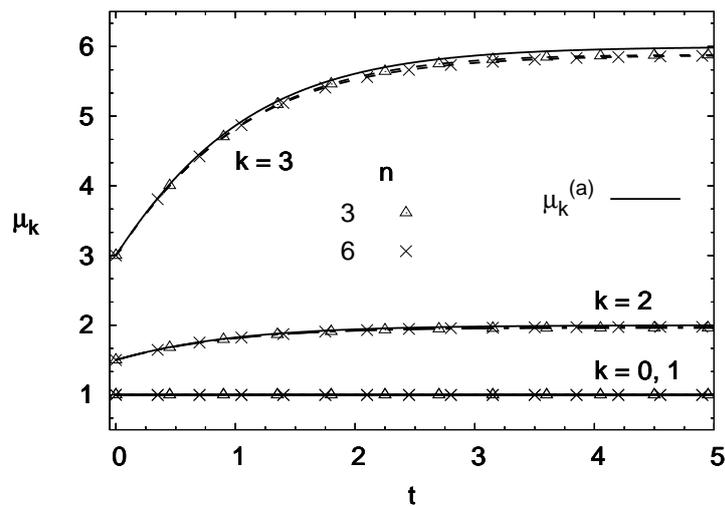


Figura 6.2: Comportamento de falsa convergência obtido pelo PPDC com 3 e 6 pontos de quadratura em um caso de quebra e agregação equivalentes.

total de amostragens de μ_k tendo o subscrito l referindo-se ao valor amostrado.

A Fig. 6.3 apresenta o erro médio obtido pelos métodos híbridos e pelo MoC aplicados a casos de quebra dominante e agregação dominante.

É possível verificar pela Fig. 6.3 que o comportamento global do QMOM e DQMOM são idênticos e sua acurácia aumenta rapidamente com o número de pontos de quadratura usados. Como o PPDC apresenta uma falsa convergência, sua acurácia pouco é afetada pelo número de pontos de quadratura. De fato, até o método das classes se torna mais acurado que o PPDC, quando cerca de 150-200 classes são usadas.

Por fim, os testes de tempo de computação, apresentados no Apêndice A, usando diferentes pontos de quadratura indicaram o DQMOM como o método mais eficiente usando até 8 pontos. Usando mais pontos de quadratura, o PPDC se torna mais eficiente que os outros métodos. Contudo, o DQMOM possui melhor acurácia que este último. Além disso, o uso 4-6 pontos de quadratura já é suficiente para representar bem as principais propriedades da distribuição.

Em adição, para casos com dependência espacial, cada peso e abscissa se com-

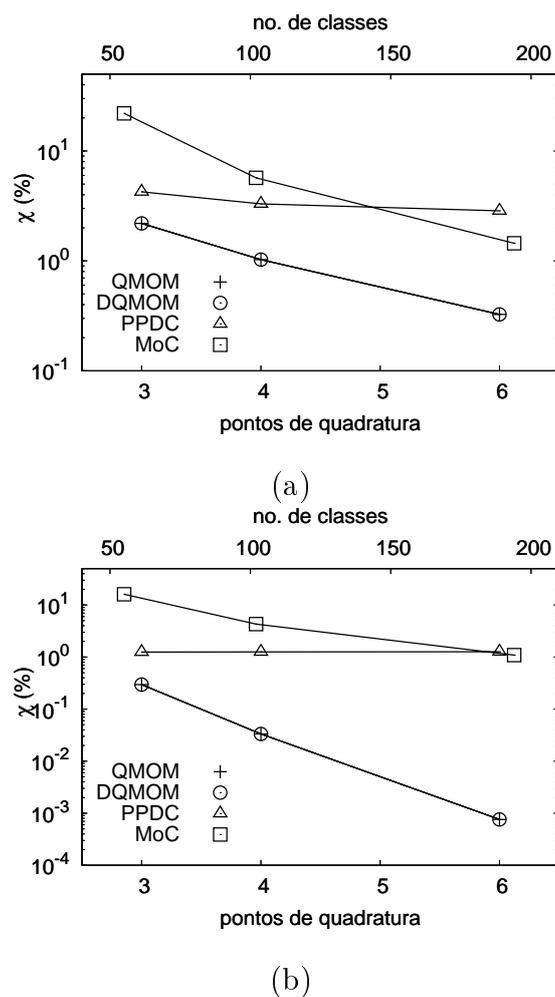


Figura 6.3: Erro global médio obtido para os casos de (a) quebra dominante e (b) agregação dominante.

porta como uma variável escalar que se propaga com seu próprio campo de velocidade. Essas variáveis podem ser associadas a cada fase dispersa em um escoamento multifásico, utilizando suas velocidades nas equações de transporte de pesos e abscissas. Portanto, o DQMOM se apresenta como o método mais indicado para a implementação do acoplamento CFD-EBP.

6.3 Avaliação do DQMOM e Acoplamento MUSIG

Uma vez tendo definido o método numérico a ser usado, iniciou-se a etapa de implementação no OpenFOAM. O DQMOM foi implementado (detalhes na Seção 5.3)

gerando o *solver* pbeFoam. Contudo, surge a necessidade de verificar se o DQMOM foi implementado corretamente no OpenFOAM. Para tal, foram realizadas simulações 0D transientes e 1D estacionários usando 4 pontos de quadratura e seus resultados foram comparados com a solução analítica de LAGE [158] para o caso de quebra e agregação equivalentes e MCCOY e MADRAS [160] para os casos de quebra e de agregação dominantes.

Nestas simulações, os processos de quebra e de agregação entre partículas são caracterizados por modelos irrealis [158, 160],

$$a(v, v') = 1, \quad b(v) = \frac{1}{2}\Phi(\infty)^2 v, \quad P(v | v') = \frac{1}{v'}, \quad \vartheta(v) = 2 \quad (6.3)$$

onde $\Phi(\infty) = N_T(\infty)/N_T(0)$ é o parâmetro que controla o efeito de quebra ou de agregação dominante. Por sua vez, $N_T(\infty)$ e $N_T(0)$ são respectivamente o número total de partículas no domínio no final e no início do processo. Portanto, $\Phi(\infty) > 1$ para os processos com quebra dominante e $\Phi(\infty) < 1$ para casos com agregação dominante.

De modo a verificar a aplicação do DQMOM em escoamentos, o acoplamento MUSIG foi testado para casos de quebra ou de agregação dominante em uma geometria de um canal com expansão em degrau (*backward facing step* - BFS) bidimensional. A mistura bifásica utilizada foi uma emulsão fictícia de água em óleo. Nestas simulações, os casos de quebra e de agregação dominantes foram simulados usando respectivamente $\Phi(\infty) = 6$ e $0, 1$.

Em sua dissertação de mestrado, DAMIAN [18] implementou a solução da EBP pelo DQMOM no ANSYS CFX via rotinas de usuário (em FORTRAN). Dessa forma, foi possível comparar os resultados obtidos pelo OpenFOAM com o ANSYS CFX. Os resultados desta comparação foram apresentados no *International Conference of Multiphase Flow 2007* [281], que ocorreu em Leipzig, Alemanha, cujo artigo se encontra no Apêndice B. Uma versão aprimorada deste trabalho foi submetido e aceito recentemente para publicação [260].

6.3.1 Solução da EBP no OpenFOAM

Estas simulações foram realizadas com o objetivo de realizar uma verificação rigorosa da acurácia da implementação do DQMOM no OpenFOAM. Em outras palavras, o interesse se concentra na acurácia da integração no tempo ou espaço das equações de transporte do DQMOM. Portanto, o erro da aproximação por quadratura foi excluído das comparações apresentadas nesta seção. Para tal, a EBP homogênea aplicada ao DQMOM fornece um sistema de equações diferenciais ordinárias para os pesos e abscissas

$$\frac{dw_\alpha}{dt} = \theta_\alpha, \quad \frac{d\xi_\alpha}{dt} = \kappa_\alpha, \quad \alpha = 1 \dots n \quad (6.4)$$

cuja solução foi implementada em um código numérico escrito em FORTRAN 90, usado para gerar os resultados apresentados na Seção 6.2, com a rotina DASSL [282] para integração no tempo.

O algoritmo adaptativo para o passo de tempo e ordem de integração numérica na DASSL em conjunto com um exigente critério para convergência das equações (tolerâncias absolutas e relativas de 10^{-10} e 10^{-12} , respectivamente) forneceram uma solução numérica virtualmente livre de erros de integração numérica. Neste caso, o estado estacionário para a solução numérica do caso de quebra e agregação equivalentes [160] é considerada a “solução analítica” para os pesos, $w_i^{(a)}$, e abscissas, $\xi_i^{(a)}$. Sem dúvida, esta solução numérica ainda contém os erros associados a aproximação por quadratura do DQMOM. Portanto, ao comparar os resultados provindos da implementação do DQMOM no OpenFOAM com a “solução analítica” da DASSL, apenas os erros associados à integração no tempo e espaço são mostrados [18].

De fato, as condições de simulação para o caso transiente foram as mesmas usadas na Seção 6.2. Usando os métodos de Euler explícito e Crank-Nicholson e um passo de tempo de 10^{-3} s, a acurácia das simulações foi mensurada através de um erro médio, mostrado na Eq. 6.5.

$$(\chi^t)^2 = \frac{1}{2nM} \sum_{j=1}^M \sum_{i=1}^n \left\{ \left[\frac{|w_{i,j}^{(a)} - w_{i,j}|}{w_{i,j}^{(a)}} \right]^2 + \left[\frac{|\xi_{i,j}^{(a)} - \xi_{i,j}|}{\xi_{i,j}^{(a)}} \right]^2 \right\} \quad (6.5)$$

Tabela 6.1: Erros médios nas simulações transientes para os casos de (a) quebra e agregação equivalentes, (b) quebra dominante e (c) agregação dominante.

Casos	$\chi^t(\%)$	
	Euler	Crank-Nicholson
a	0,00015	0,000045
b	0,58	6,26
c	0,0094	0,029

onde M é o número total de amostras retiradas no tempo e o subscrito j representa o valor amostrado.

Em todos os casos analisados, uma ótima acurácia foi obtida ao usar o método de Euler. Contudo, o método de Crank-Nicholson apresentou erros maiores do que os obtidos pelo Euler, devido ao seu comportamento oscilatório quando o passo de tempo usado não é pequeno o suficiente. Os resultados do erro médio estão colocados na Tabela 6.1.

A solução unidimensional obtida pelo OpenFOAM pôde ser comparada com a solução transiente de MCCOY e MADRAS [160] através de uma transformação de domínio,

$$\frac{d\phi}{dt} = \varphi \implies u_z \frac{d\phi}{dz} = \varphi \quad (6.6)$$

utilizando uma velocidade u_z de propagação das variáveis no domínio 1D com tamanho $L = 1 \text{ m}$. Desta forma, L/u_z representa o tempo advectivo para a propagação da solução da EBP pelo domínio 1D. Por consequência, u_z pode ser escolhido de forma que as variáveis do DQMOM apresentem perfis estacionários ao longo da coordenada z que sejam iguais aos perfis das simulações transientes.

Considerando que os valores estacionários para os pesos e abscissas nas simulações transientes são atingidos em 0.5 s para o caso de quebra dominante e 40 s para o caso de agregação dominante, é possível calcular as velocidades u_z para cada caso

Tabela 6.2: Erro médio (%) para as simulações 1D no casos de (b) quebra e (c) agregação dominante usando diferentes métodos de interpolação.

OpenFOAM Esquema advectivo	Caso	
	b	c
QUICK	0,11	0,47
UD	1,05	4,85
G 0.2	1,44	1,26
G 0.6	1,30	1,26
G 1.0	0,36	1,31

através dos tempos advectivos L/u_z . Assim, os campos de velocidade u_z , constante e definido para todo o domínio, foram calculados como 2 m/s e 0.025 m/s para os casos de quebra dominante e agregação dominante, respectivamente. Os detalhes desta operação estão colocados no Apêndice B.

Ao analisar a solução unidimensional, nota-se que variações na solução de apenas uma variável pode propagar erros para todas as outras variáveis do DQMOM. Isto ocorre pois o sistema linear do DQMOM agrega todas as informações das variáveis, afetando-as e podendo, inclusive, levar à divergência da simulação. O refino da malha pode ser essencial para atenuar estas variações e evitar a divergência da solução nestes casos. Outra solução a ser considerada é a análise da discretização do termo convectivo das equações de transporte do DQMOM. De fato, estas equações são puramente advectivas e, portanto, funções de interpolação de alta ordem e com limitadores de fluxo fornecem os melhores resultados. Os métodos *Upwind* (UD), QUICK e GAMMA (G Γ^1) foram testados para os casos de quebra e de agregação dominantes usando respectivamente malhas com 500 e 100 volumes.

Para o OpenFOAM, a função de interpolação QUICK apresentou os menores

¹O parâmetro Γ é um fator de *blending* entre os métodos UD e diferença central [263].

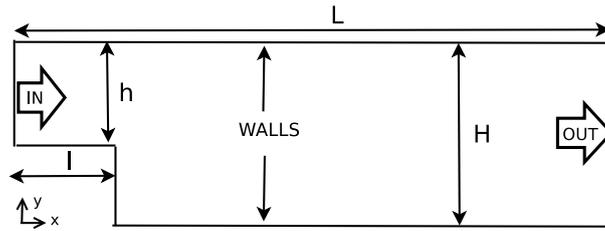


Figura 6.4: Dimensões paramétricas e nome dos contornos da geometria BFS, onde $L = 11H$, $l = H$ e $h = H/2$ sendo $H = 0,01 \text{ m}$ para o caso de quebra e $H = 0,1 \text{ m}$ para agregação dominante.

erros médios. Contudo, sabe-se que funções com limitadores de fluxo são capazes de minimizar a difusão numérica e oscilações. Levando isso em conta, a função GAMMA com fator $\Gamma = 1$ [263] se apresentou como a mais indicada para ser usada na discretização do DQMOM aplicado a escoamentos.

Com base nos testes apresentados acima e sua comparação com resultados analíticos, garante-se que o DQMOM foi implementado corretamente no OpenFOAM.

6.3.2 Escoamento em emulsão usando o modelo MUSIG

As simulações foram realizadas em uma geometria BFS, cujas dimensões estão mostradas na Fig. 6.4, discretizada com malhas hexaédricas e mista tetra-prismáticas com aproximadamente 4000, 8000 e 16000 volumes de controle.

Conforme foi discutido no Apêndice B, esta geometria foi utilizada devido à sua simplicidade e por possuir zonas de recirculação. Apenas a força de arrasto foi considerada nestas simulações e em condições de escoamento laminar (modelo de turbulência desativado nas simulações). Além disso, as mesmas funções de quebra e de agregação simples [160], mostradas na Eq. 6.3, foram usadas. As propriedades físicas e condições de entrada usadas nas simulações estão listados na Tabela 6.3.

A variável interna adotada neste caso é o volume da partícula, sendo que o domínio interno do BFS possui partículas com um diâmetro médio de Sauter de

Tabela 6.3: Propriedades físicas e condições de entrada para a mistura bifásica.

Prop. Físicas	óleo	água
ρ ($kg\ m^{-3}$)	900	1000
η ($kg\ m^{-1}\ s^{-1}$)	$1 \cdot 10^{-5}$	$1 \cdot 10^{-6}$
Variável	Condições de entrada	
d_s (μm)	-	50
r_α (-)	0,9	0,1

50 μm como a condição inicial. Nos primeiros testes de simulação com o MUSIG, apareceram problemas numéricos na precisão do cálculo do DQMOM. Deste modo, formas adimensionais dos pesos e abscissas foram usadas com sucesso para evitar estes problemas. Assim, todo o algoritmo do DQMOM utiliza variáveis adimensionais que devem ser convertidas à sua forma dimensional para calcular a fração volumétrica e o diâmetro de Sauter usados no modelo bifásico. As Eqs. 6.7 e 6.8 mostram respectivamente a adimensionalização dos pesos e abscissas,

$$w_\alpha^\phi = \frac{1}{N_T(0)} w_\alpha^f \quad (6.7)$$

$$\xi_\alpha^\phi = \frac{N_T(0)}{r_d(0)} \xi_\alpha^f \quad (6.8)$$

onde $N_T(0)$ é o número total de partículas no domínio (momento zero), $r_d(0)$ é a fração volumétrica global da fase dispersa (primeiro momento), ambas avaliadas no instante inicial ou na entrada do domínio. Por sua vez, os superscritos f e ϕ representam, respectivamente, as variáveis dimensionais e adimensionais. Maiores detalhes sobre o processo de adimensionalização se encontram no Apêndice B.

Os valores iniciais das variáveis do DQMOM foram obtidas pela condição estacionária dos pesos e abscissas obtidos pela solução da DASSL para o caso de quebra e agregação equivalentes [160]. Este procedimento foi discutido na Seção 6.3.1 do presente trabalho.

Todas as simulações foram realizadas usando o método implícito de Crank Ni-

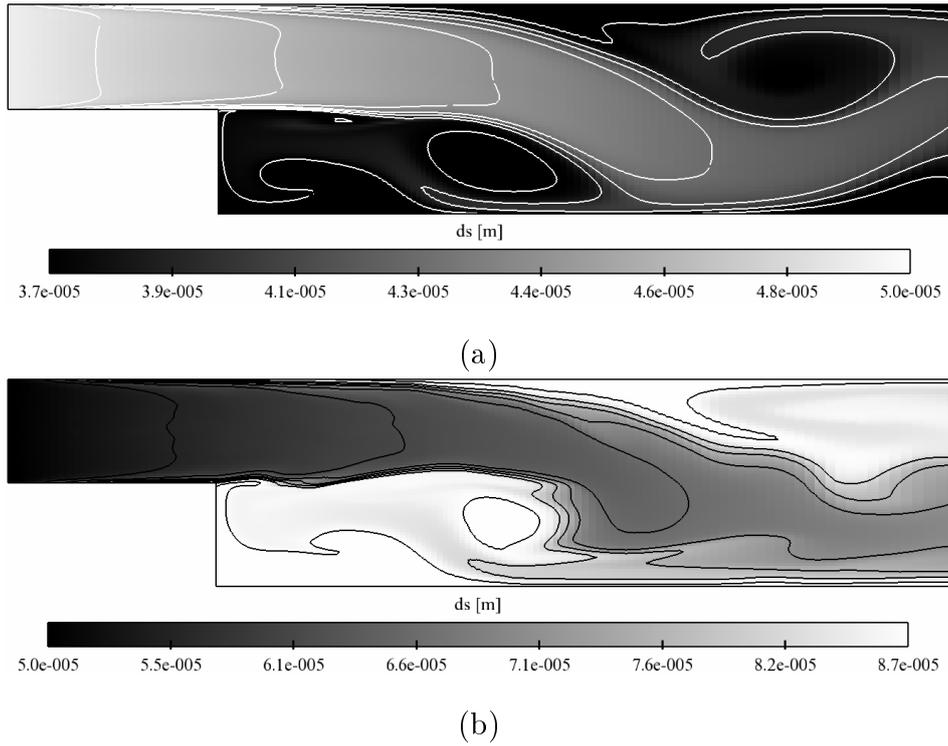
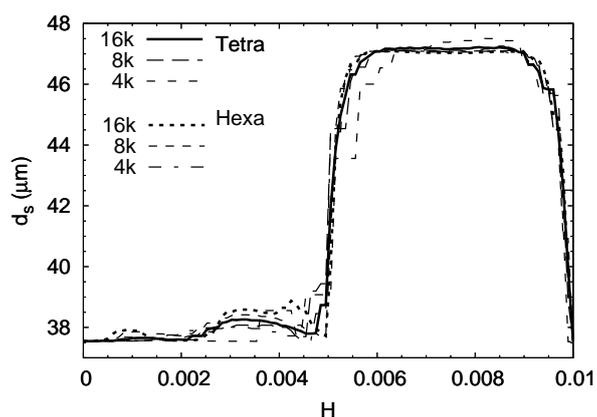


Figura 6.5: Contorno do diâmetro de Sauter, d_s , para os casos de (a) quebra e (b) agregação dominante.

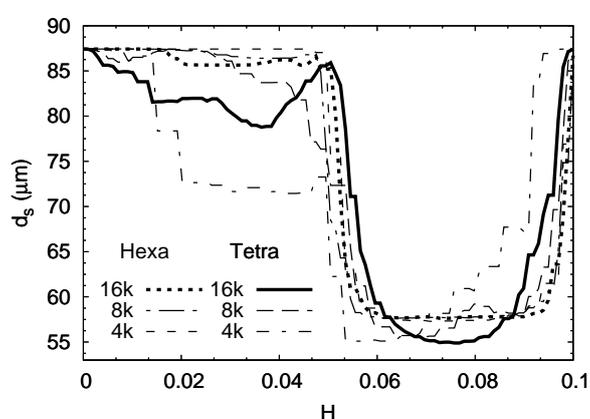
cholson para integração no tempo com passo de tempo adaptativo, controlado de tal forma que o número de Courant máximo na malha seja 0,3. Os termos advectivos das equações de transporte foram aproximados usando o método de interpolação Gamma, com $\Gamma = 1$ [263]. O algoritmo PISO para acoplamento pressão velocidade teve como critério de parada um erro absoluto na pressão de 0,01 e, para malhas tetraédricas, 3 iterações foram usadas para a correção de não-ortogonalidade da malha [263].

Os resultados do contorno do diâmetro de Sauter, para os casos de quebra e de agregação dominantes usando a malha hexaédrica mais refinada, estão colocados na Fig. 6.5.

A distribuição de tamanho de partículas é advectada com o fluido e suas mudanças são causadas pelos efeitos de quebra ou de agregação. Assim, o grau de mudança na distribuição é proporcional ao tempo de residência das partículas. Como observado nos resultados colocados na Fig. 6.5, as partículas que ficam presas no interior



(a)



(b)

Figura 6.6: Convergência em malha para os casos de (a) quebra e (b) agregação usando malhas hexaédricas e tetra-prismáticas.

das zonas de recirculação apresentam, como esperado, altas taxas de interação.

De modo a verificar a convergência da malha, os resultados do diâmetro de Sauter amostrados em uma linha vertical (alocada na região do primeiro vórtice, em $x = 1,25H$) usando as malhas hexa e tetra-prismáticas estão colocados na Fig. 6.6.

Como verificado na Fig. 6.6, os piores resultados foram obtidos usando a malha tetra-prismática. De fato, o formato da malha construída foi a principal razão deste comportamento ruim. Para que a formulação de volumes finitos implementada no OpenFOAM forneça um resultado mais acurado, o vetor que liga o centro do volume

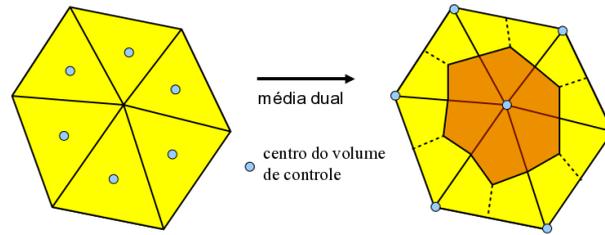


Figura 6.7: Ilustração da conversão de um grupo de elementos tetraédricos em um volume de controle poliédrico.

e da face deve ser preferencialmente normal à face. Se a não-ortogonalidade entre o vetor e o centro da face for alta, o OpenFOAM pode ter dificuldades em corrigir os fluxos e, conseqüentemente, levar a resultados inaccurados. A conversão da malha tetra-prisma para poliédrica (*dual mesh*), ilustrada na Fig. 6.7, é interessante pois alinha o fluxo normal a face com o centro dos volumes de controle.

Contudo, a ferramenta de conversão não pode ser aplicada a malhas bidimensionais no OpenFOAM e, portanto, o uso de malhas poliédricas não foi testado para este caso. Porém, deve-se ressaltar que um estudo mais aprofundado da solução do DQMOM usando malhas poliédricas deve ser efetuado no futuro.

Por fim, a implementação do DQMOM no OpenFOAM foi verificada com sucesso e sua extensão no acoplamento MUSIG pôde ser avaliada comparando resultados de dois pacotes CFD diferentes. Os resultados obtidos em ambos os pacotes apresentaram pequenas diferenças entre si devido aos detalhes de implementação específicos a cada código. De modo geral, não foram verificados problemas na solução numérica do DQMOM, principalmente pela escolha correta da função de interpolação. De fato, durante os testes iniciais das simulações com o acoplamento MUSIG usando a função QUICK nas variáveis do DQMOM, foram notados problemas de divergência numérica que levaram a erros de execução de ponto flutuante no código. Nestes casos, o uso de funções de interpolação com limitadores de fluxo foi crucial para o sucesso das simulações.

É importante ressaltar que as equações de transporte do DQMOM convergiram

mais rápido que as equações que regem o escoamento. Em cada passo de tempo, poucas iterações foram necessárias para que a tolerância na solução dos sistemas lineares provindos da discretização do DQMOM fosse atingida. Isto indica que os parâmetros numéricos, como refino de malha e passo de tempo adotado, pouco afetaram a solução do DQMOM e eram principalmente controlados pelas equações do escoamento bifásico.

Maiores detalhes sobre os resultados das simulações do MUSIG usando o OpenFOAM e sua comparação com o ANSYS CFX estão colocados no Apêndice B.

6.4 Escoamento Multifásico Polidisperso

As simulações multifásicas foram realizadas usando a mesma geometria BFS ($H = 0,01\text{ m}$) da simulação apresentada na Seção 6.3.2. Porém, apenas a malha hexaédrica mais refinada (com cerca de 16 mil volumes) foi utilizada neste caso. Novamente, apenas a força de arrasto entre as fases foi considerada. Em primeiro lugar, tem-se como objetivo verificar a implementação multifásica no OpenFOAM e suas capacidades. Em seguida, o acoplamento do *solver* multifásico com balanço populacional é analisado.

As simulações foram realizadas usando o método de Crank Nicholson para integração no tempo com passo de tempo adaptativo e controlado pelo número de Courant máximo menor que 0,3. Os termos convectivos das equações de transporte do DQMOM foram aproximadas usando o método Gamma [263] com coeficiente Γ igual a 1. A convergência no acoplamento pressão velocidade foi atingida assumindo uma tolerância de 10^{-8} no critério misto (Eq. 5.61) sobre a pressão.

Parte dos resultados contidos nesta seção foram apresentados na *OpenFOAM International Conference 2007*, cujo artigo se encontra no Apêndice C.

6.4.1 Simulações Multifásicas

Neste ponto, o principal objetivo é verificar a implementação do modelo Euleriano multifásico no OpenFOAM, como colocado na seção seguinte.

Verificação da Implementação

Como primeira avaliação, deve-se verificar se o *solver* multifásico foi implementado de forma correta. Para tal, pode-se comparar seus resultados usando $n = 2$ com aqueles obtidos com o *solver* bifásico de RUSCHE [66]. Em adição, simulações multifásicas usando várias fases dispersas com propriedades físicas iguais entre si também podem ser usadas para verificar o código. Neste caso, as frações volumétricas das fases dispersas devem ser iguais e definidas de tal forma que sua soma seja igual à fração volumétrica global definida no *solver* bifásico. Assim, a fração volumétrica das fases dispersas são calculadas dividindo a fração volumétrica global pelo número de fases dispersas utilizadas. Considerando a fração volumétrica global no `twoPhaseEulerFoam` como $r_a = 0,1$, definiu-se as frações no *solver* multifásico como $r_\alpha = 0,1, 0,05$ e $0,025$ para $n = 2, 3$ e 5 , respectivamente.

Dois casos foram analisados, sendo que o primeiro considera que as frações volumétricas e os diâmetros característicos das fases no interior do domínio possuem o mesmo valor definido na entrada da geometria. Já o segundo caso considera que estas variáveis são nulas no interior do domínio. Estes casos são referenciados respectivamente como Caso I e II. Os valores de fração volumétrica e pressão obtidos pelos *solvers* `twoPhaseEulerFoam` e `multiPhaseEulerFoam`, retirados de uma linha vertical posicionada na altura do primeiro vórtice do BFS ($x = 1,25H$), estão mostrados na Fig. 6.8.

É possível verificar uma excelente concordância entre os perfis de fração volumétrica global e pressão obtidos nas Figs. 6.8(a) e 6.8(b) usando ambos *solvers*. Neste caso, o domínio interno já está preenchido pela emulsão e, portanto, nota-se apenas

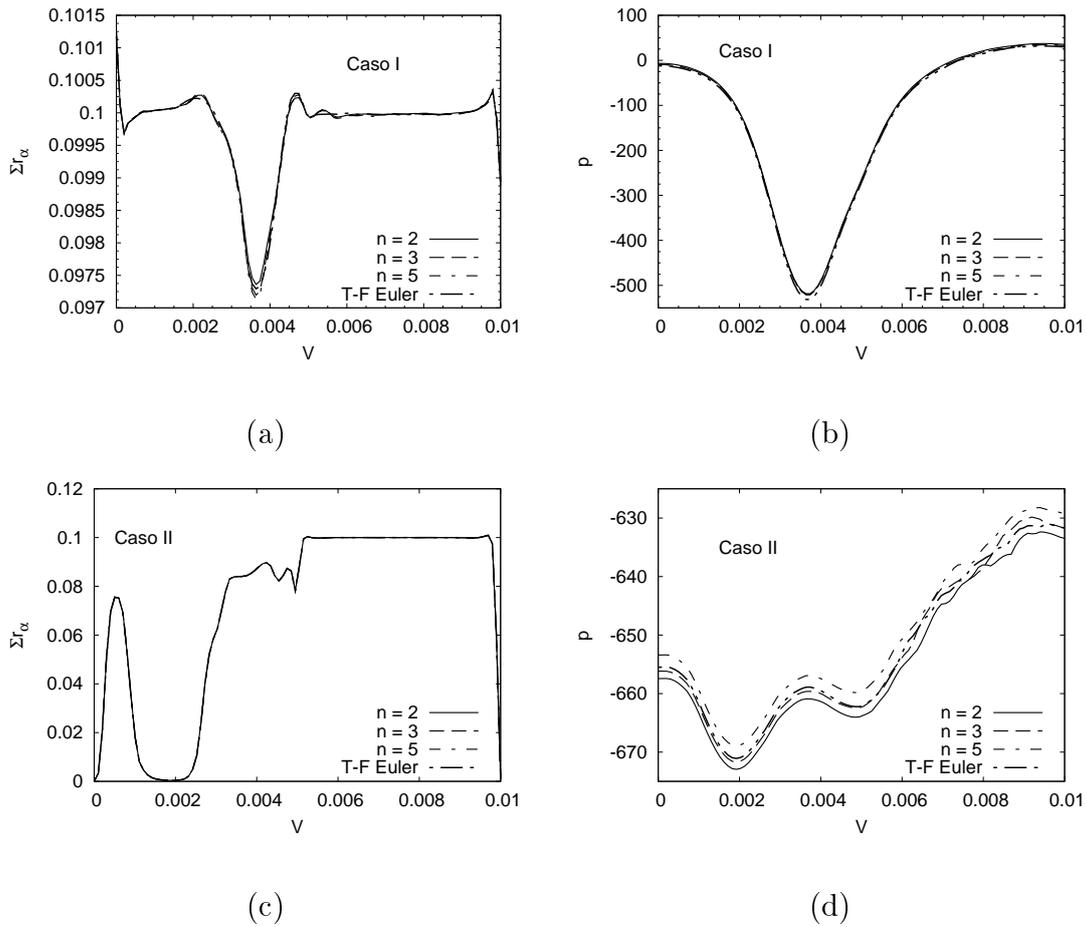


Figura 6.8: Perfis verticais de $\sum_{\alpha=1}^n r_\alpha$ e p para o Casos I, (a) e (b), e II, (c) e (d), usando o *solver* `twoPhaseEulerFoam` (T-F Euler) e o código multifásico com diferentes números n de fases.

pequenas variações na fração volumétrica global. O mesmo não ocorre no Caso II, uma vez que a emulsão preenche o domínio gerando zonas com alto gradiente de fração volumétrica. De fato, as variações de fração volumétrica foram calculadas de forma acurada em ambos os casos, como mostram as Figs. 6.8(a) e 6.8(c).

A solução para o perfil de pressão apresentou ótima concordância entre as abordagens no Caso I, como mostra a Fig. 6.8(b). Por outro lado, o perfil de pressão no Caso II apresentou pequenos desvios entre as soluções, como verificado na Fig. 6.8(d). A ordem de magnitude destes desvios é de quase 0,9% para $n = 5$, sendo que estes foram causados por falta de precisão na malha. Contudo, os desvios na pressão não afetaram o campo de velocidade das fases, como é mostrado nas Figs. 6.9(a) e 6.9(b), onde os componentes x e y da velocidade de mistura ($\bar{\mathbf{u}}_\alpha = \sum_{a=0}^n r_a \mathbf{u}_a$) são

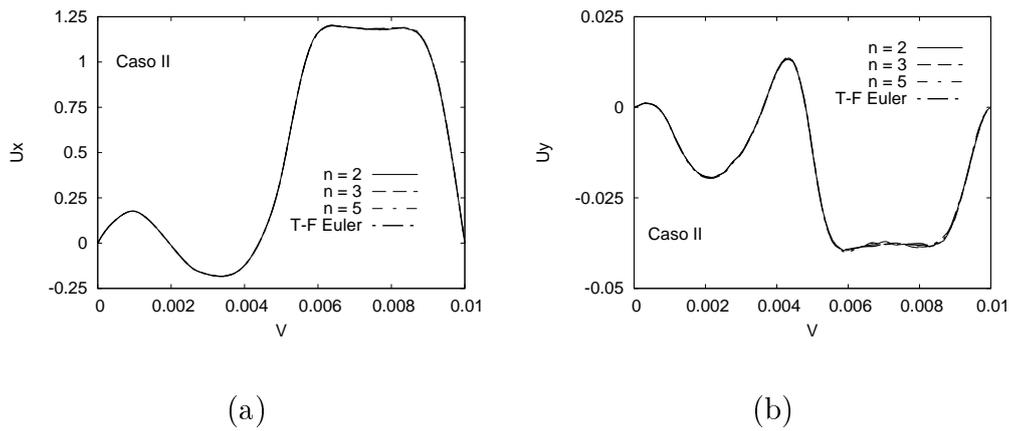


Figura 6.9: Perfis verticais dos componentes (a) x e (b) y da velocidade da mistura para o Caso II.

apresentados respectivamente.

Analisando os resultados apresentados acima, é possível concluir que a implementação do código multifásico (`multiPhaseEulerFoam`) foi realizada com sucesso.

Outros testes de simulação foram realizados para verificar o efeito da força de arrasto sobre o escoamento no BFS. Estes resultados se encontram no Apêndice C.

6.4.2 Simulações de Escoamentos Polidispersos

Nesta etapa, a implementação da extensão do código multifásico acoplado à solução da EBP usando o DQMOM é verificada analisando resultados de simulações. Em seguida, foram realizados testes para avaliar o custo computacional extra associado ao acoplamento CFD-EBP em relação ao *solver* multifásico.

Código com Acoplamento CFD-EBP

Assim como nas simulações usando MUSIG (Seção 6.3.2), as formas adimensionais dos pesos e abscissas, mostradas respectivamente nas Eqs. 6.7 and 6.8, foram usados nas simulações de modo a evitar problemas numéricos. Da mesma forma, a

condição inicial para o DQMOM foi obtida usando a “solução analítica” provinda da solução numérica da DASSL. Além disso, as funções de quebra e de agregação [160] mostradas na Eq. 6.3 foram usadas nestas simulações, com $\Phi(\infty) = 6$ e $0, 1$ respectivamente.

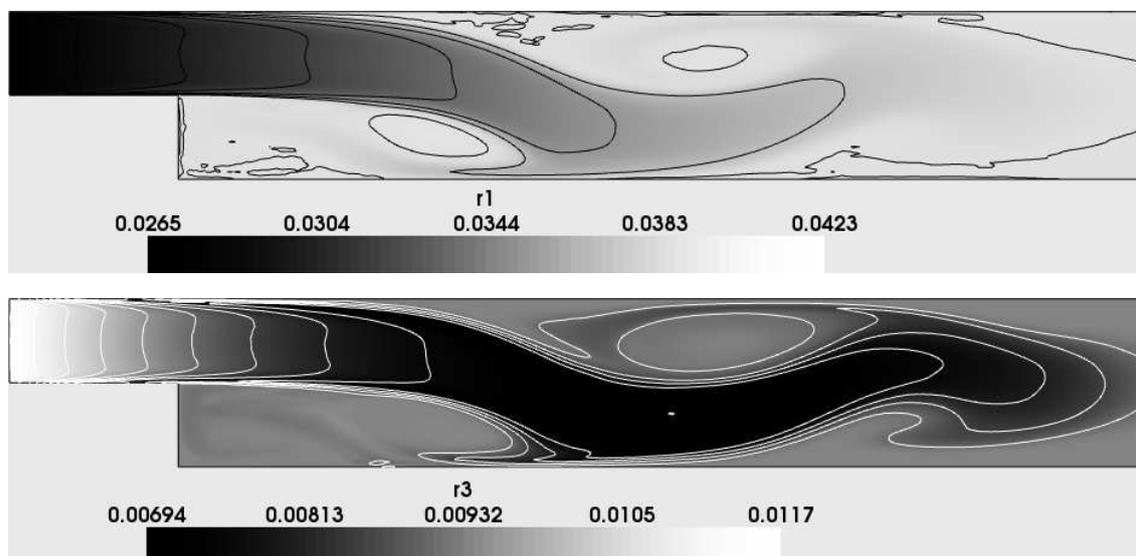
Após as devidas conversões dimensionais, os diâmetros característicos e as frações volumétricas podem ser obtidos através da aproximação de quadratura. Assim, a Eq. 5.59 fornece d_α sobre a hipótese de partículas esféricas e a Eq. 5.34 obtém a fração volumétrica global da fase dispersa $\sum_{\alpha=1}^n r_\alpha$, e a fração volumétrica de cada fase dispersa é ζ_α .

Neste caso, a densidade numérica total de partículas, $N_T(0)$, foi definida como $2,1 \cdot 10^{12} m^{-3}$ calculada através de uma expressão provinda do processo de adimensionalização e envolvendo o diâmetro médio da distribuição e a fração volumétrica global da fase dispersa (detalhes nos Apêndices B e C). As simulações foram realizadas considerando as mesmas propriedades físicas mostradas na Tabela 6.3 e as condições iniciais das variáveis no interior do domínio definidas como as mesmas na entrada do mesmo. Ou seja, as mesmas condições utilizadas no Caso I da Seção 6.4.1. O tempo total de simulação para os casos de quebra e de agregação foi de $0, 1$ s.

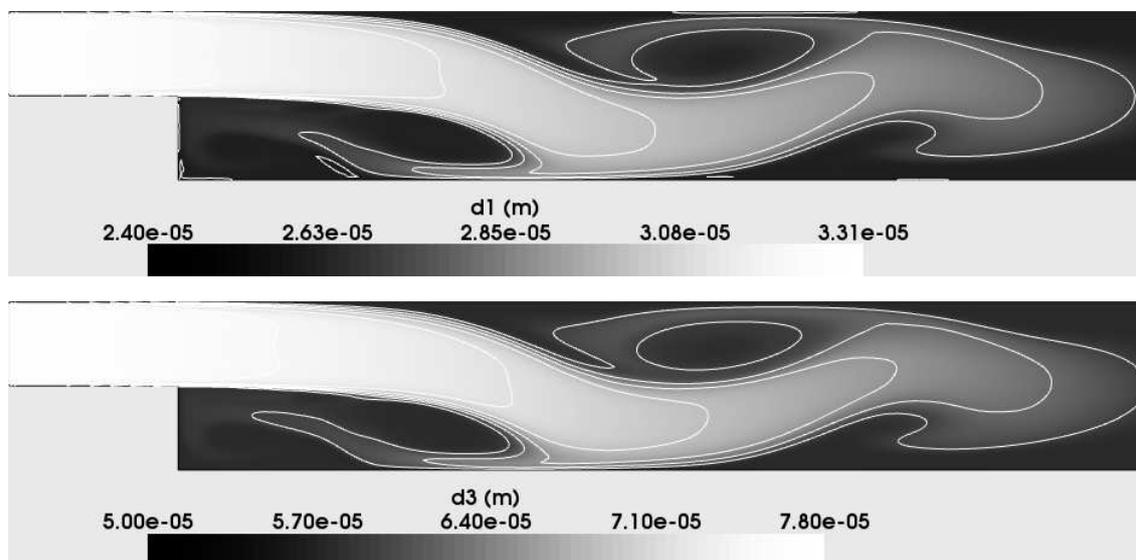
Usando as condições expostas acima e 4 pontos de quadratura, os valores iniciais dimensionais para os diâmetros característicos são $d_1 = 33, 125$, $d_2 = 57, 478$, $d_3 = 78, 018$ e $d_4 = 98, 392 \mu m$, e para as frações volumétricas são $r_1 = 0, 026904$, $r_2 = 0, 061219$, $r_3 = 0, 011652$ e $r_4 = 0, 000224$.

Os gráficos de contorno de diâmetro característico e de fração volumétrica das fases dispersas 1 e 3 obtidas para o caso de quebra dominante são mostrados na Fig. 6.10.

O caso de quebra dominante foi simulado com sucesso, onde o efeito do aprisionamento das partículas nos vórtices pode ser observado na Fig. 6.10. Nota-se



(a)



(b)

Figura 6.10: Gráficos de contorno para as (a) frações volumétricas e (b) diâmetros característicos das fases 1 e 3 no caso de quebra dominante.

que os gráficos de contorno dos diâmetros característicos são bastante parecidos, apesar de possuírem faixas de tamanho diferentes. Já no canal de entrada do BFS, as partículas começam a quebrar propiciando a queda da fração volumétrica r_3 ao longo das linhas de fluxo do escoamento. Por sua vez, a fase 1 incorpora as partículas originárias da quebra nas outras fases e, portanto, sua fração volumétrica se comporta de modo oposto à r_3 . Deve-se notar que a quebra das partículas na fase 3 é mais intensa que na primeira fase. Isto ocorre pois a função de quebra, mostrada na Eq. 6.3, é proporcional ao tamanho da partícula.

O contorno dos diâmetros característicos e as frações volumétricas das fases 1 e 3 para o caso de agregação dominante são mostrados na Fig. 6.11.

Nota-se pelos resultados na Fig. 6.11 que as partículas se agregam continuamente ao longo do escoamento devido ao uso de uma função de agregação constante (Eq. 6.3). Portanto, as maiores partículas no domínio são aquelas que ficam presas nos vórtices. Por sua vez, as frações volumétricas apresentam comportamento oposto ao verificado no caso de quebra dominante. Com a agregação das partículas, a fração volumétrica da fase 1 diminui conforme a da fase 3 aumenta. Isto ocorre pois os maiores diâmetros da distribuição se concentram nas fases 3 e 4.

Em conclusão, os testes de simulação de escoamento polidisperso foram realizados com sucesso para os casos de quebra e de agregação de partículas.

Eficiência de Paralelismo

A eficiência de paralelização dos códigos para o escoamento multifásico e do acoplamento CFD-EBP foram avaliados em uma máquina Intel Quad Core Q6600 2.4 GHz com o sistema operacional Linux 64 bits. As simulações foram realizadas usando $n = 5$, com condições e parâmetros idênticos ao Caso I (Seção 6.4.1) e usando uma malha refinada com 78000 volumes.

O algoritmo do gradiente biconjugado com preconditionador diagonal LU in-

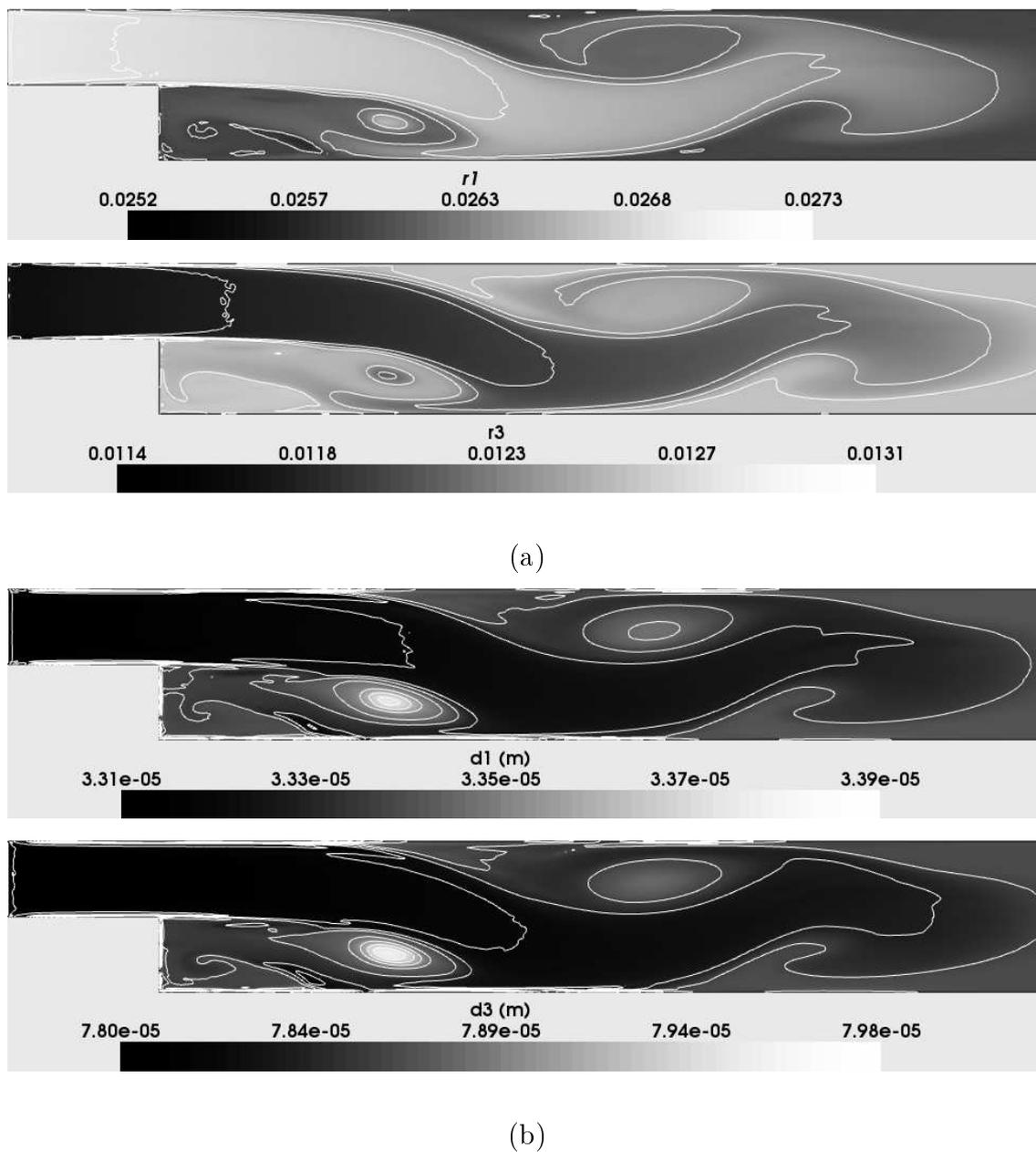


Figura 6.11: Gráficos de contorno para as (a) frações volumétricas e (b) diâmetros característicos das fases 1 e 3 no caso de agregação dominante.

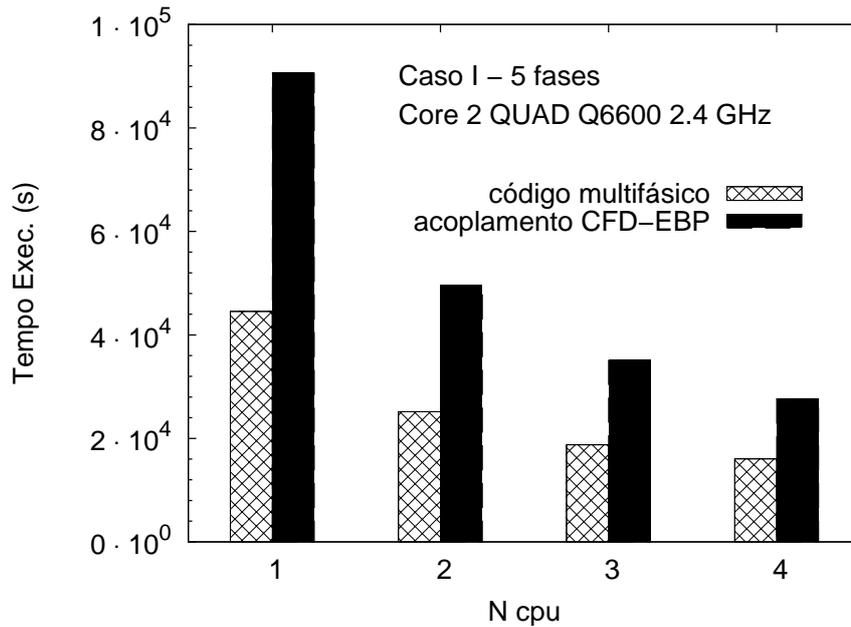


Figura 6.12: Tempo computacional de simulações seriais e paralelas (N_{cpu}) usando os códigos para escoamento multifásico com e sem o acoplamento com balanço populacional.

completo foi usado na solução de todos os sistemas lineares provindos de equações discretizadas, considerando uma tolerância de 10^{-10} . Apenas para a solução da equação da pressão utilizou o preconditionador diagonal Cholesky incompleto com tolerância de $5 \cdot 10^{-8}$. O algoritmo METIS [283] foi usado na decomposição da malha.

Os tempos computacionais de simulações seriais (1 processador) e paralelas usando os *solvers* `multiPhaseEulerFoam` e `multiPhasePbeFoam` com $n = 5$ estão colocados na Fig. 6.12.

A partir do tempo computacional mostrado na Fig. 6.12, verifica-se que a simulação em paralelo usando 2 processadores foi 1,77 e 1,82 mais rápida que as rodadas seriais nos códigos multifásico e CFD-EBP, respectivamente. Por sua vez, as simulações usando 4 processadores foram 2,77 e 3,27 vezes mais rápidas que as simulações seriais para estes mesmos *solvers*. Esta é uma clara evidência que a paralelização do `multiPhasePbeFoam` é mais efetiva do que o código multifásico. No algoritmo de CFD-EBP acoplado, a solução do sistema linear do DQMOM, mostrado na Eq. 5.29,

é computacionalmente custoso uma vez que o mesmo é avaliado em cada volume da malha usando um método LU de solução matricial. Contudo, a solução do sistema linear do DQMOM só é necessária para os volumes presentes respectivamente nos domínios decompostos da malha. Portanto, não existe comunicação entre os processadores na solução deste sistema linear, o que concede ao código de CFD-EBP melhor taxa de paralelização.

Mesmo assim, a diferença entre o custo computacional em simulações seriais usando os dois códigos é um fator de 2,03 em favor do multifásico. Este resultado difere daquele obtido por FAN *et al.* [251], que reportou que o custo extra da solução do sistema linear do DQMOM aumentou em apenas 18% o tempo computacional em relação ao seu código multifásico. Apesar disso, deve-se notar que FAN *et al.* [251] utilizaram funções de quebra e de agregação constantes e uma malha computacional muito grosseira (750 volumes), o que pode tornar seus resultados não equivalentes aos obtidos no presente trabalho. Por fim, a diferença entre os custos computacionais usando os dois códigos diminui devido à paralelização do sistema linear do DQMOM. Como foi verificado na Fig. 6.12, o código para escoamento multifásico é 1,97, 1,87 e 1,72 vezes mais rápido que o *solver multiPhasePbeFoam* ao usar respectivamente dois, três e quatro processadores.

Capítulo 7

Conclusões e Sugestões

7.1 Resumo do Trabalho e Conclusões

A presente tese atingiu com sucesso o objetivo de desenvolver métodos para a simulação de escoamentos polidispersos acoplando um modelo Euleriano multifásico à solução da equação de balanço populacional (EBP). A implementação destas metodologias foi realizada no pacote CFD OpenFOAM, escrito em C++ e com código fonte aberto. Partindo do modelo Euleriano bifásico [66] previamente implementado no OpenFOAM, o desenvolvimento foi realizado em etapas graduais. De fato, estas etapas foram a solução da EBP usando o método DQMOM e o desenvolvimento do modelo multifásico. Por fim, com a integralização destes chegou-se, enfim, a uma ferramenta computacional para a simulação de escoamentos multifásicos polidispersos considerando n fases dispersas.

Um estudo preliminar com base na revisão da literatura foi realizado para verificar os modelos multifásicos existentes e selecionar a formulação mais propícia para o desenvolvimento do trabalho. O modelo multifásico Euleriano se mostrou como o mais indicado, pois o uso de variáveis médias permite o acoplamento com a solução da EBP, e devido a sua generalidade de aplicações [30, 38, 39]. Neste

modelo, existe a necessidade de modelar a troca de quantidade de movimento entre as fases. Contudo, questões importantes para a reprodução de escoamentos multifásicos polidispersos realísticos como a modelagem da turbulência e formulação dos coeficientes presentes no termos de troca de quantidade de movimento não foram investigados. De fato, o foco do presente trabalho se concentra no desenvolvimento da metodologia de acoplamento CFD-EBP e sua implementação no OpenFOAM.

Portanto, a escolha de um método numérico eficiente para a solução da EBP eficiente é crucial para a metodologia de acoplamento. A partir da revisão bibliográfica, verifica-se claramente a superioridade das abordagens híbridas para a solução numérica da EBP. Os métodos híbridos incorporam propriedades do método das classes, como a discretização das variáveis internas (volume, massa, concentração, etc.), e o uso de momentos para caracterizar a distribuição de partículas [208]. Contudo, a grande vantagem dos métodos híbridos recai no uso da quadratura de Gordon [209]. Neste caso, os pesos e abscissas aproximam as integrais da função de distribuição da melhor forma possível e, em contraste com o método das classes, as abscissas podem se mover livremente por todo o domínio da variável interna sem limitações ou restrições. Além disso, poucos pontos de quadratura são necessários para representar os principais momentos da distribuição.

Dessa forma, foi realizado um estudo para determinar a acurácia numérica e sua eficiência computacional dos métodos híbridos existentes na literatura, o QMOM [208], DQMOM [211] e PPDC [212]. Os testes foram realizados para casos de quebra e agregação equivalentes, quebra dominante e agregação dominante, comparando os resultados numéricos obtidos com soluções analíticas [158, 160]. Verificou-se que a solução do PPDC apresenta erros numéricos severos causados, principalmente, por usar uma separação de operadores baseado no método de integração de Euler explícito. De modo geral, o QMOM e DQMOM apresentaram acurácia bastante similar. Contudo, verificou-se que a eficiência computacional do DQMOM é superior à do QMOM. Portanto, é possível concluir que o DQMOM se apresenta atualmente como o método numérico mais indicado para ser usado como base deste

trabalho.

A análise das técnicas de acoplamento CFD-EBP mostrou que o método das classes com pivô fixo é usado com frequência nos trabalhos da literatura. No acoplamento completo do método das classes com o modelo Euleriano multifásico, cada fase é representada por uma classe. Esta abordagem é computacionalmente intensiva devido ao grande número de classes necessários para bem representar a função de distribuição de partículas. Portanto, formas simplificadas de acoplamento CFD-EBP foram propostas, como o MUSIG homogêneo [221] e não-homogêneo [223, 224]. Porém, estas abordagens são bastante criticadas devido a sua ineficiência.

A abordagem Euleriana multifásica acoplada ao DQMOM é viável pois cada fase é representada por um ponto de quadratura e, portanto, seriam necessárias poucas fases para representar bem os principais momentos da distribuição de partículas. Além disso, o DQMOM adapta os pesos e abscissas para melhor representar os momentos da distribuição, evitando a perda de precisão causada pelo truncamento do domínio das variáveis internas que usualmente ocorre no método de pivô fixo. Devido as suas vantagens em relação as outras técnicas, a abordagem de acoplamento CFD-EBP usando o DQMOM foi utilizada no presente trabalho.

Visando implementar o acoplamento CFD-EBP no OpenFOAM, foi realizada uma revisão sobre as capacidades e limitações das metodologias numéricas reunidas neste pacote CFD. Em conclusão, a facilidade de uso, as formulações numéricas implementadas, como o método dos volumes finitos e solução de sistemas lineares, e, principalmente, a liberdade em manipular seu código fonte, tornaram o OpenFOAM propício para a execução do presente trabalho. Portanto, estudar a programação das classes e *templates* no OpenFOAM é, logicamente, vital para a implementação dos códigos. No início deste trabalho, a documentação sobre programação no OpenFOAM era escassa e provinda, principalmente, da lista de discussão dos usuários na internet. Dessa forma, o esforço para aprender e estudar o código fonte foi considerável devido à variedade de classes e à complexidade do problema a ser tratado.

Apesar disso, as fontes de informação sobre o OpenFOAM são hoje abundantes na internet, em trabalhos técnicos e científicos. De fato, as funções implementadas no código são poderosas e bastante variadas, sendo capazes de tratar quase todo problema que envolva campos tensoriais. Por fim, conclui-se que o OpenFOAM é uma ótima ferramenta para o desenvolvimento de códigos CFD, sem a limitação e falta de liberdade imposta por programas comerciais de código fechado.

Neste trabalho, a implementação do balanço populacional considerou um problema monovariado, empregando o volume das partículas para a descrição da distribuição de tamanhos, considerados apenas os efeitos de quebra e agregação.

Os modelos multifásicos Eulerianos foram desenvolvidos com base no trabalho de RUSCHE [66], que formulou a abordagem bifásica implementada no OpenFOAM, chamada `twoPhaseEulerFoam`, para fases incompressíveis. Neste caso, as forças de troca de *momentum* entre as fases são calculadas por um diâmetro médio, constante durante toda a simulação. Portanto, as metodologias aqui desenvolvidas estendem este modelo bifásico para considerar os efeitos de polidispersão usando $n + 1$ fases. O desenvolvimento ocorreu em etapas, onde a complexidade da modelagem aumentava gradualmente. Os detalhes sobre o uso das classes e *templates* em C++ para implementação dos códigos também foram apresentados neste trabalho de tese. Ao final do desenvolvimento e implementação dos códigos, quatro novos *solvers* foram gerados:

- `pbeFoam`, contendo a solução da EBP pelo DQMOM, que é utilizado em todos os outros *solvers*. Os termos fontes das equações de transporte dos pesos e abscissas ponderadas são obtidas através da solução do sistema linear do DQMOM em cada volume da malha. A decomposição LU foi usada na solução do sistema linear.
- `musigFoam`, onde o `pbeFoam` foi acoplado ao código bifásico [66]. Nesta abordagem, todas as partículas presentes na fase dispersa possuem o mesmo campo de velocidade. Em adição, o coeficiente da força de arrasto foi calculado usando

o diâmetro médio de Sauter obtido pela solução da EBP.

- `multiPhaseEulerFoam`, caracterizado pela extensão do modelo incompressível bifásico [66] de modo a considerar uma fase contínua e n fases dispersas. Neste caso, cada fase dispersa possui seu próprio campo de velocidade e fração volumétrica. Contudo, o diâmetro característico é constante em todo o domínio.
- `multiPhasePbeFoam`, construído visando o acoplamento dos *solvers* `pbeFoam` e `multiPhaseEulerFoam` para a simulação de escoamentos multifásicos polidispersos. Diferente do *solver* `multiPhaseEulerFoam`, os diâmetros das fases dispersas são caracterizados por campos escalares. Neste caso, os diâmetros característicos das fases dispersas são afetados diretamente pelos efeitos de interação entre partículas, como quebra e agregação.

A implementação do DQMOM no OpenFOAM, chamada `pbeFoam`, foi testada em casos transientes 0D e estacionários 1D. Neste último caso, a convergência dos resultados na malha e a acurácia de diferentes funções de interpolação para o termo convectivo das equações de transporte de DQMOM foram avaliadas. Os resultados numéricos foram comparados com a solução analítica de MCCOY e MADRAS [160] para casos de quebra e agregação dominante, apresentando ótima acurácia da solução numérica. Nos casos unidimensionais, o método de interpolação QUICK se destacou pela sua acurácia. Contudo, sabe-se que funções com limitadores de fluxo são capazes de minimizar a difusão numérica e as oscilações. Como as equações do DQMOM utilizadas neste trabalho possuem o comportamento hiperbólico, a função GAMMA com fator $\Gamma = 1$ [263] se apresentou como a mais indicada, entre as testadas, para ser usada na discretização do DQMOM em acoplamentos CFD-EBP. Considerando os resultados discutidos acima, concluiu-se que o `pbeFoam` foi implementado corretamente.

Simulações transientes com efeitos de quebra e agregação dominante foram realizadas para avaliar a abordagem MUSIG. Nestes casos foi utilizada uma emulsão

de água em óleo escoando em uma geometria bidimensional de um canal com expansão em degrau (*backward facing step* - BFS). Verificou-se que, para cada passo de tempo, as equações de transporte do DQMOM convergem mais rápido que as equações que regem o escoamento. Em comparação com o pacote comercial ANSYS CFX, houve concordância entre os resultados obtidos para os diâmetro de Sauter (ver Apêndice B). De fato, nos testes feitos com malhas hexaédricas e tetra-prismáticas mistas, as primeiras apresentaram melhores resultados. De fato, o uso de malhas tetraédricas no OpenFOAM deve ser considerado com cuidado.

Os códigos `multiPhaseEulerFoam` e `multiPhasePbeFoam` também foram avaliados usando a geometria BFS com a emulsão água-óleo. A implementação do código multifásico foi verificada simulando escoamentos bifásicos e comparando seus resultados com os obtidos para o *solver* `twoPhaseEulerFoam`. O *solver* multifásico foi simulado considerando 2, 3 e 5 fases dispersas com valores de fração volumétrica e propriedades físicas iguais. Além disso, a fração volumétrica global das fases dispersas é igual à fração volumétrica da fase dispersa na emulsão. A concordância dos resultados obtidos pelas duas abordagens verificam a programação correta do código multifásico. As simulações polidispersas apresentaram o comportamento esperado, onde as frações volumétricas das fases dispersas são alteradas de acordo com o efeito dominante da interação entre partículas.

Os tempos de computação dos *solvers* `multiPhaseEulerFoam` e `multiPhasePbeFoam` foram avaliados, conferindo ao primeiro maior eficiência. Isto se deve ao custo extra de computação do DQMOM no `multiPhasePbeFoam`. Porém, a paralelização do código CFD-EBP é bem mais eficiente do que a do `multiPhaseEulerFoam`, pois a solução do sistema linear do DQMOM é realizada volume a volume.

Note que o presente trabalho focou o desenvolvimento da metodologia e de sua implementação, visando obter ferramentas computacionais para a simulação de escoamentos multifásicos polidispersos. Com as ferramentas produzidas neste trabalho,

abre-se novas fronteiras para o tratamento dos problemas práticos em processos multifásicos.

7.2 Sugestões para Trabalhos Futuros

Tendo em vista os resultados obtidos, algumas sugestões para continuação do presente trabalho são propostas. Em relação aos avanços numéricos associados à implementação do acoplamento CFD-EBP, sugere-se que novos estudos enfoquem:

- Implementação de métodos para a EBP multivariada, sendo o próximo passo no desenvolvimento das técnicas para a simulação de escoamentos multifásicos polidispersos.
- Implementação de outras abordagens para o acoplamento pressão-velocidade em escoamentos multifásicos [284] nos códigos desenvolvidos, avaliando a performance destes algoritmos aplicada ao modelo multifásico polidisperso. Este é um ponto importante nas simulações CFD com algoritmos de acoplamento segregado.
- Implementação dos efeitos de crescimento e nucleação de partículas, de modo a caracterizar fenômenos de mudança de fase.
- Extensão da metodologia do PPDC, visando eliminar os erros associados à integração no tempo. Para tal, deve-se considerar o uso de um método de separação de operadores de mais alta ordem em sua formulação.

Visando a simulação de escoamentos polidispersos realísticos, a modelagem de alguns fenômenos devem ser investigadas.

- Modelos para a quebra e agregação de partículas são as principais fontes de erros nas simulações polidispersas. Segundo ARAUJO [17], muitos dos modelos presentes na literatura partem de conceitos de formulação errados. Desta

forma, mais estudos sobre o desenvolvimento dos modelos devem ser realizados. Apesar disso, simulações CFD tridimensionais podem fornecer dados para a estimação de parâmetros destes modelos.

- Termos de troca de quantidade de movimento entre as fases. Ainda há muita divergência na literatura sobre a formulação correta das forças e sua influência no escoamento.
- Estudo da turbulência em escoamentos multifásicos. Sabe-se da influência da turbulência na descrição do perfil de escoamento e da dinâmica dos fenômenos de interação entre partículas, porém ainda existem muitas dúvidas em sua formulação.

Referências Bibliográficas

- [1] KRISHNA, R., ELLENBERGER, J., MARETTO, C., “Flow Regime Transition in Bubble Columns”, *International Communications in Heat and Mass Transfer*, v. 26, n. 4, pp. 467–475, 1999.
- [2] RAMKRISHNA, D., MAHONEY, A. W., “Population balance modeling. Promise for the future”, *Chemical Engineering Science*, v. 57, pp. 595–606, 2002.
- [3] SCARLETT, B., “Particle Populations - to balance or not to balance, that is the question!” *Powder Technology*, v. 125, pp. 1–4, 2002.
- [4] LAGE, P. L. C., ESPÓSITO, R. O., “Experimental determination of bubble size distributions in bubble columns: Prediction of mean bubble diameter and gas hold up”, *Powder Technology*, v. 101, n. 2, pp. 142–150, 1999.
- [5] RIBEIRO JR., C. P., LAGE, P. L. C., “Experimental study on bubble size distributions in a direct-contact evaporator”, *Brazilian Journal of Chemical Engineering*, v. 21, n. 1, pp. 69–81, 2004.
- [6] RIBEIRO JR., C. P., LAGE, P. L. C., “Direct-contact evaporation in the homogeneous and heterogeneous bubbling regimes. Part I: experimental analysis”, *International Journal of Heat and Mass Transfer*, v. 47, n. 17-18, pp. 3825–3840, 2004.
- [7] SILVA, M. O., SILVA, M. V. L. R., LAGE, P. L. C., “On the Usage of Conductivity Sensors to Measure Gas Phase Fraction in Bubble Columns”. In:

- EBECEM 2008 1o Encontro Brasileiro sobre Ebulição, Condensação e Escoamento Multifásico Líquido-Gás*, v. CDROM, pp. 1–7, Florianópolis, 2008.
- [8] RIBEIRO JR., C. P., BORGES, C. P., LAGE, P. L. C., “Modelling of direct-contact evaporation using a simultaneous heat and multicomponent mass-transfer model for superheated bubbles”, *Chemical Engineering Science*, v. 60, n. 6, pp. 1761–1772, 2005.
- [9] RIBEIRO JR., C. P., LAGE, P. L. C., “Gas-Liquid Direct-Contact Evaporation: A Review”, *Chemical Engineering and Technology*, v. 28, n. 10, pp. 1081–1107, 2005.
- [10] LAGE, P. L. C., CAMPOS, F. B., “Advances in Direct Contact Evaporator Design”, *Chemical Engineering and Technology*, v. 27, n. 1, pp. 91–96, 2004.
- [11] SILVA, L. F. L. R., FONTES, C. E., LAGE, P. L. C., “Front tracking in recirculating flows: a comparison between the TVD and RCM methods in solving the VOF equation”, *Brazilian Journal of Chemical Engineering*, v. 22, n. 1, pp. 105–116, 2005.
- [12] MONTEAGUDO, J. E. P., LAGE, P. L. C., “Cross-Properties Relations in 3D Percolation Networks: I. Network Characteristic Length Determination”, *Transport in Porous Media*, v. 61, n. 2, pp. 143–156, 2005.
- [13] MONTEAGUDO, J. E. P., LAGE, P. L. C., “Cross-Properties Relations in 3D Percolation Networks: II. Network Permeability”, *Transport in Porous Media*, v. 61, n. 3, pp. 259–274, 2005.
- [14] MONTEAGUDO, J. E. P., SILVA, L. F. L. R., LAGE, P. L. C., “Scaling laws for network model permeability: application to wellbore oil flow simulation with solid deposition”, *Chemical Engineering Science*, v. 58, n. 9, pp. 1815–1829, 2003.

- [15] CAMPOS, F. B., LAGE, P. L. C., “A numerical method for solving the transient multidimensional population balance equation using an Euler-Lagrange formulation”, *Chemical Engineering Science*, v. 58, n. 12, pp. 2725–2744, 2003.
- [16] RIBEIRO JR., C. P., LAGE, P. L. C., “Population balance modeling of bubble size distributions in a direct-contact evaporator using a sparger model”, *Chemical Engineering Science*, v. 59, n. 12, pp. 2363–2377, 2004.
- [17] ARAUJO, J. F. M., *Estudo dos modelos de quebra e coalescência para escoamentos polidispersos*, Master’s Thesis, Universidade Federal do Rio de Janeiro, PEQ/COPPE, RJ, Brasil, 2006.
- [18] DAMIAN, R. B., *Simulação CFD de Escoamentos Multifásicos Polidispersos usando Balanço Populacional*, Master’s Thesis, Universidade Federal do Rio de Janeiro, PEQ/COPPE, RJ, Brasil, 2007.
- [19] DUDUKOVIC, M. P., LARACHI, F., MILLS, P. L., “Multiphase Reactors — revisited”, *Chemical Engineering Science*, v. 54, pp. 1975–1995, 1999.
- [20] JOSHI, J. B., “Computational flow modelling and design of bubble column reactors”, *Chemical Engineering Science*, v. 56, pp. 5893–5933, 2001.
- [21] KRISHNA, R., ELLENBERGER, J., SIE, S. T., “Reactor development for conversion of natural gas to liquid fuels: A scale-up strategy relying on the hydrodynamics analogies”, *Chemical Engineering Science*, v. 51, n. 10, pp. 2041–2050, 1996.
- [22] KRISHNA, R., URSEANU, M. I., VAN BATEN, J. M., et al., “Influence of scale on the hydrodynamics of bubble columns operating in the churn-turbulent regime: Experiments vs. Eulerian simulations”, *Chemical Engineering Science*, v. 54, pp. 4903–4911, 1999.
- [23] LAPIN, A., LÜBBERT, A., “Numerical simulations of the dynamics of two-phase gas-liquid flows in bubble columns”, *Chemical Engineering Science*, v. 49, pp. 3661–3674, 1994.

- [24] DELNOIJ, E., LAMMERS, F. A., KUIPERS, J. A. M., et al., “Dynamic simulation of dispersed gas-liquid two-phase flow using a discrete bubble model”, *Chemical Engineering Science*, v. 52, n. 9, pp. 1429–1458, 1997.
- [25] DREW, D. A., “Mathematical modeling of two-phase flow”, *Annual Reviews in Fluid Mechanics*, v. 15, pp. 261–291, 1983.
- [26] KRISHNA, R., VAN BATEN, J. M., “Eulerian simulations of bubble columns operating at elevated pressures in the churn turbulent flow regime”, *Chemical Engineering Science*, v. 56, pp. 6249 – 6258, 2001.
- [27] CHEN, P., SANYAL, J., DUDUKOVIC, M. P., “CFD modeling of bubble columns flows: implementation of population balance”, *Chemical Engineering Science*, v. 59, pp. 5201– 5207, 2004.
- [28] RAMKRISHNA, D., *Population Balances – Theory and Applications to Particulate Systems in Engineering*. Academic Press: Nova York, 2000.
- [29] BAUER, M., EIGENBERGER, G., “Multiscale modeling of hydrodynamics, mass transfer and reaction in bubble column reactors”, *Chemical Engineering Science*, v. 56, n. 3, pp. 1067 – 1074, 2001.
- [30] ISHII, M., *Thermo-fluid Dynamic Theory of Two-phase Flow*. Eyrolles: Paris, 1975.
- [31] VAN WACHEM, B. G. M., ALMSTEDT, A. E., “Methods for multiphase computational fluid dynamics”, *Chemical Engineering Journal*, v. 96, pp. 81–98, 2003.
- [32] DELNOIJ, E., KUIPERS, J. A. M., VAN SWAAIJ, W. P. M., “A three-dimensional CFD model for gas-liquid bubble columns”, *Chemical Engineering Science*, v. 54, n. 13-14, pp. 2217–2226, 1999.
- [33] CROWE, C., SOMMERFIELD, M., YUTAKA, T., *Multiphase Flows with Droplets and Particles*. CRC Press, 1998.

- [34] LI, C., MOSYAK, A., HETSRONI, G., “Direct numerical simulation of particle-turbulence interaction”, *International Journal of Multiphase Flow*, v. 25, n. 2, pp. 187–200, 1999.
- [35] DRUZHININ, O. A., ELGHOBASHI, S. E., “A Lagrangian-Eulerian Mapping Solver for Direct Numerical Simulation of Bubble-Laden Turbulent Shear Flows Using the Two-Fluid Formulation”, *Journal of Computational Physics*, v. 154, n. 1, pp. 174–196, 1999.
- [36] BOIVIN, M., SIMONIN, O., SQUIRES, K. D., “Direct numerical simulation of turbulence modulation by particles in isotropic turbulence”, *Journal of Fluid Mechanics*, v. 375, pp. 235–263, 1998.
- [37] SUSSMAN, M., PUCKETT, E. G., “A Coupled Level Set and Volume-of-Fluid Method for Computing 3D and Axisymmetric Incompressible Two-Phase Flows”, *Journal of Computational Physics*, v. 162, n. 2, pp. 301 – 337, 2000.
- [38] ENWALD, H., PEIRANO, E., ALMSTEDT, A.-E., “Eulerian two-phase flow theory applied to fluidization”, *International Journal of Multiphase Flow*, v. 22, n. Supp. 1, pp. 21–66, 1996.
- [39] DREW, D. A., PASSMAN, S. L., *Theory of Multicomponent Fluids*. 1st ed. Springer, 1999.
- [40] ISHII, M., HIBIKI, T., *Thermo-Fluid Dynamics of Two-Phase Flow*. 1st ed. Springer: Nova York, EUA, 2006.
- [41] MANNINEN, M., TAIVASSALO, V., KALLIO, S., *On the mixture model for multiphase flow*, Tech. Rep. 288, Technical Research Centre of Finland, 1996.
- [42] BIRD, R. B., STEWART, W. E., LIGHTFOOT, E. N., *Transport Phenomena*. John Wiley & Sons Inc.: Nova York, 1960.

- [43] MICHELE, V., *CFD modeling and measurement of liquid flow structure and phase holdup in two- and three-phase bubble columns*, Ph.D. Thesis, Technischen Universität Carolo-Wilhelmina zu Braunschweig, 2002.
- [44] BRENNEN, C. E., *Fundamentals of Multiphase Flow*. 1st ed. Cambridge University Press, 2005.
- [45] BOVE, S., *Computational fluid dynamics of gas-liquid flows including bubble population balances*, Ph.D. Thesis, Esbjerg Institute of Engineering, Dinamarca, 2005.
- [46] SANYAL, J., VÁSQUEZ, S., ROY, S., et al., “Numerical simulation of gas-liquid dynamics in cylindrical bubble column reactors”, *Chemical Engineering Science*, v. 54, pp. 5071–5083, 1999.
- [47] SOKOLICHIN, A., EIGENBERGER, G., “Applicability of the standard $k-\epsilon$ turbulence model to the dynamic simulation of bubble columns: Part I. Detailed numerical simulations”, *Chemical Engineering Science*, v. 54, pp. 2273–2284, 1999.
- [48] LAPIN, A., MAUL, C., JUNGHANS, K., et al., “Industrial-scale bubble column reactors: gas-liquid flow and chemical reaction”, *Chemical Engineering Science*, v. 56, n. 1, pp. 239 – 246, 2001.
- [49] CARTLAND GLOVER, G. M., GENERALIS, S. C., “The modelling of buoyancy driven flow in bubble columns”, *Chemical Engineering and Processing*, v. 43, n. 2, pp. 101 – 115, 2004.
- [50] FORD, B., LOTH, E., “Forces on ellipsoidal bubbles in a turbulent shear layer”, *Physics of Fluids*, v. 10, n. 1, pp. 178–188, 1998.
- [51] LOTH, E., “Numerical approaches for motion of dispersed particles, droplets and bubbles”, *Progress in Energy and Combustion Science*, v. 26, pp. 161 – 223, 2000.

- [52] NEWTON, I., *Philosophiae Naturalis: Principia Mathematica*. Societatis Regalis Sodali, 1686.
- [53] ALLEN, H. S., “The motion of a sphere in a viscous fluid”, In: *Philosophical Magazine*, v. 50, pp. 323–338, 1900.
- [54] STOKES, G. G., “On the Effect of the Internal Friction of Fluids on the Motion of Pendulums”, *Cambridge Philosophical Society Transactions*, v. 9, pp. 8–106, 1851.
- [55] SCHLICHTING, H., *Boundary-Layer Theory*. 7th ed. McGraw-Hill: Nova York, 1979.
- [56] CLIFT, R., GRACE, J. R., WEBER, M. E., *Bubbles, Drops and Particles*. Academic Press: Londres, 1978.
- [57] SIMONNET, M., GENTRIC, C., OLMOS, E., et al., “Experimental determination of the drag coefficient in a swarm of bubbles”, *Chemical Engineering Science*, v. 62, n. 3, pp. 858–866, 2007.
- [58] POISEUILLE, J. L. M., “Recherches experimentales sur le mouvement des liquides dans les tubes de très petits diametres”, In: *Comptes Rendus*, v. 12, pp. 112–115, Academie des Sciences, 1841.
- [59] SEGRÉ, G., SILBERBERG, A., “Behaviour of macroscopic rigid spheres in Poiseuille flow Part 1. Determination of local concentration by statistical analysis of particle passages through crossed light beams”, *Journal of Fluid Mechanics*, v. 14, pp. 115–135, 1962.
- [60] COX, R. G., MASON, S. G., “Suspended Particles in Fluid Flow Through Tubes”, *Annual Review of Fluid Mechanics*, v. 3, pp. 291–316, 1971.
- [61] TSUJI, Y., MORIKAWA, Y., SHIOMI, H., “LDV measurements of an air-solid two-phase flow in a vertical pipe”, *Journal of Fluid Mechanics*, v. 139, pp. 417–434, 1984.

- [62] AUTON, T. R., “The lift force on a spherical body in a rotational flow”, *Journal of Fluid Mechanics*, v. 183, pp. 199–218, 1987.
- [63] LEGENDRE, D., MAGNAUDET, J., “The lift force on a spherical bubble in a viscous linear shear flow”, *Journal of Fluid Mechanics*, v. 368, pp. 81–126, 1998.
- [64] MCLAUGHLIN, J. B., “Inertial migration of a small sphere in linear shear flows”, *Journal of Fluid Mechanics*, v. 224, pp. 261–274, 1991.
- [65] LEGENDRE, D., MAGNAUDET, J., “A note on the lift force on a spherical bubble or drop in a low-Reynolds-number shear flow”, *Physics of Fluids*, v. 9, n. 1, pp. 3572–3574, 1997.
- [66] RUSCHE, H., *Computational fluid dynamics of dispersed two-phase flows at high phase fractions*, Ph.D. Thesis, Imperial College of Science, Technology and Medicine, Londres, Reino Unido, 2002.
- [67] DREW, D. A., LAHEY, R. T., “Some supplemental analysis concerning the virtual mass and lift force on a sphere in a rotating and straining flow”, *International Journal of Multiphase Flow*, v. 16, pp. 1127–1130, 1990.
- [68] PFLEGER, D., GOMES, S., GILBERT, N., et al., “Hydrodynamic simulations of laboratory scale bubble columns fundamental studies of the Eulerian-Eulerian modelling approach”, *Chemical Engineering Science*, v. 54, pp. 5091–5099, 1999.
- [69] SUNDARESAN, S., “Modeling the Hydrodynamics of Multiphase Flow Reactors: Current Status and Challenges”, *AIChE Journal*, v. 46, n. 6, pp. 1102 – 1105, 2000.
- [70] DEEN, N. G., SOLBERG, T., HJERTAGER, B. H., “Large eddy simulation of the gas-liquid flow in a square cross-sectioned bubble column”, *Chemical Engineering Science*, v. 56, pp. 6341 – 6349, 2001.

- [71] BOUSSINESQ, J., “Théorie de l’Écoulement Tourbillant”, *Mem. Présentés par Divers Savants Acad. Sci. Inst. Fr.*, v. 23, pp. 46–50, 1877.
- [72] BRAUER, H., “Turbulenz in mehrphasigen Strömungen”, *Chem. Ing. Tech.*, v. 51, n. 10, pp. 934–948, 1979.
- [73] TSUCHIYA, K., OHSAKI, K., TAGUCHI, K., “Large and Small Bubble Interaction Patterns in a Bubble Column”, *International Journal of Multiphase Flow*, v. 22, n. 1, pp. 121–132, 1996.
- [74] LAUNDER, B., SPALDING, D., “The numerical computation of turbulent flows”, *Computer Methods in Applied Mechanics and Engineering*, v. 3, pp. 269–289, 1974.
- [75] WILCOX, D. C., *Turbulence Modeling for CFD*. DCW Industries: California, 1994.
- [76] SATO, Y., SEKOGUCHI, K., “Liquid velocity distribution in two-phase bubble flow”, *International Journal of Multiphase Flow*, v. 2, pp. 79–95, 1975.
- [77] SATO, Y., SADATOMI, M., SEKOGUCHI, K., “Momentum and heat transfer in two-phase bubble flow - I Theory”, *International Journal of Multiphase Flow*, v. 7, pp. 167–177, 1981.
- [78] LANCE, M., BARAILLE, J., “Turbulence in the liquid phase of a uniform bubbly air-water flow model for bubbly two-phase flow”, *Journal of Fluid Mechanics*, v. 222, pp. 95–118, 1991.
- [79] CUI, Z., FAN, L. S., “Turbulence energy distributions in bubbling gas-liquid and gas-liquid-solid flow systems”, *Chemical Engineering Science*, v. 59, pp. 1755 – 1766, 2004.
- [80] KAJISHIMA, T., “Influence of particle rotation on the interaction between particle clusters and particle-induced turbulence”, *International Journal of Heat and Fluid Flow*, v. 25, pp. 721–728, 2004.

- [81] LUCAS, D., KREPPER, E., PRASSER, H.-M., “Use of models for lift, wall and turbulent dispersion forces acting on bubbles for poly-disperse flows”, *Chemical Engineering Science*, v. 62, pp. 4146–4157, 2007.
- [82] KATAOKA, I., SERIZAWA, A., “Basic equations of turbulence in gas-liquid two-phase flow”, *International Journal of Multiphase Flow*, v. 15, n. 5, pp. 843–855, 1989.
- [83] LOPEZ DE BERTODANO, M., LAHEY, R. T., JONES, O. C., “Development of a k- ϵ model for bubbly two-phase flow”, *Journal of Fluids Engineering*, v. 116, pp. 128–134, 1994.
- [84] CHAHED, J., ROIG, V., MASBERNAT, L., “Eulerian-Eulerian two-fluid model for turbulent gas-liquid bubbly flows”, *International Journal of Multiphase Flow*, v. 29, pp. 23–49, 2003.
- [85] WILD, G., PONCIN, S., LI, H.-Z., et al., “Some Aspects of the Hydrodynamics of Bubble Columns”, *International Journal of Chemical Reactor Engineering*, v. 1, n. R7, pp. 1 – 36, 2003.
- [86] DECKWER, W. D., *Bubble Column Reactors*. 2nd ed. John Wiley & Sons LTD, 1992.
- [87] CAMARASA, E., VIALA, C., PONCIN, S., et al., “Influence of coalescence behaviour of the liquid and of gas sparging on hydrodynamics and bubble characteristics in a bubble column”, *Chemical Engineering and Processing*, v. 38, n. 4-6, pp. 329–344, 1999.
- [88] CHEN, R. C., REESE, J., FAN, L. S., “Flow structure in a three-dimensional bubble column and three-phase fluidized bed”, *AIChE Journal*, v. 40, n. 7, pp. 1093–1104, 1994.
- [89] ZAHRDNÍK, J., FIALOVÁ, M., RUZICKA, M., et al., “Duality of the gas-liquid flow regimes in bubble column reactors”, *Chemical Engineering Science*, v. 52, n. 21-22, pp. 3811–3826, 1997.

- [90] TORVIK, R., SVENDSEN, H. F., “Modelling of slurry reactors. A fundamental approach”, *Chemical Engineering Science*, v. 45, pp. 2325–2332, 1990.
- [91] SVENDSEN, H. F., JACKOBSEN, H. A., TORVIK, R., “Local flow structures in internal loop and bubble column reactors”, *Chemical Engineering Science*, v. 47, pp. 3297–3304, 1992.
- [92] JACKOBSEN, H. A., SVENDSEN, H. F., HJARBO, K. W., “On the prediction of local flow structures in internal loop and bubble column reactors using a two fluid model”, *Computers and Chemical Engineering*, v. 17S, pp. S531–S536, 1993.
- [93] GREVSKOTT, S., SANNÆS, B. H., DUDUKOVIC, M. P., et al., “Liquid circulation, bubble size distributions, and solid movement in two- and three-phase bubble columns”, *Chemical Engineering Science*, v. 51, pp. 1703–1713, 1996.
- [94] JAKOBSEN, H. A., SANNÆS, B. H., GREVSKOTT, S., et al., “Modeling of bubble driven vertical flows”, *Industrial and Engineering Chemical Research*, v. 36, pp. 4052–4074, 1997.
- [95] GRIENBERGER, J., HOFMANN, H., “Investigation and modelling of bubble columns”, *Chemical Engineering Science*, v. 42, pp. 2215–2220, 1992.
- [96] HILLMER, G., WEISMANTEL, L., HOFMANN, H., “Investigations and modelling of slurry bubble columns”, *Chemical Engineering Science*, v. 49, pp. 837–843, 1994.
- [97] RANADE, V. V., “Flow in bubble columns: some numerical experiments”, *Chemical Engineering Science*, v. 47, pp. 1857–1869, 1992.
- [98] CELIK, I., WANG, Y.-Z., “Numerical simulation of circulation in gas-liquid column reactors: isothermal, bubbly, laminar flow”, *International Journal of Multiphase Flow*, v. 20, n. 6, pp. 1053–1070, 1994.

- [99] SOKOLICHIN, A., EIGENBERGER, G., “Gas-liquid flow in bubble columns and loop reactors: Part I. Detailed modelling and numerical simulation”, *Chemical Engineering Science*, v. 49, n. 24B, pp. 5735–5746, 1994.
- [100] BECKER, S., SOKOLICHIN, A., EIGENBERGER, G., “Gas-liquid flow in bubble columns and loop reactors: Part II. Comparison of detailed experiments and flow simulations”, *Chemical Engineering Science*, v. 49, n. 24B, pp. 5747–5762, 1994.
- [101] DEVANATHAN, N., DUDUKOVIC, M. P., LAPIN, A., et al., “Chaotic flow in bubble column reactors”, *Chemical Engineering Science*, v. 50, pp. 2661–2667, 1995.
- [102] BORCHERS, O., BUSCH, C., SOKOLICHIN, A., et al., “Applicability of the standard $k-\epsilon$ turbulence model to the dynamic simulation of bubble columns. Part II. Comparison of detailed experiments and flow simulations”, *Chemical Engineering Science*, v. 54, pp. 5927–5935, 1999.
- [103] MUDDE, R. F., SIMONIN, O., “Two- and three-dimensional simulations of a bubble plume using a two-fluid model”, *Chemical Engineering Science*, v. 54, pp. 5061–5069, 1999.
- [104] DEEN, N. G., SOLBERG, T., HJERTAGER, B. H., “Numerical Simulation of the Gas-Liquid Flow in a Square Cross-sectioned Bubble Column”. In: *CHISA 14th International Congress of Chemical and Process Engineering*, Praga, Republica Tcheca, 2000.
- [105] PAN, Y., DUDUKOVIC, M. P., CHANG, M., “Dynamic simulation of bubbly flow in bubble columns”, *Chemical Engineering Science*, v. 54, pp. 2481–2489, 1999.
- [106] PFLEGER, D., BECKER, S., “Modelling and simulation of the dynamic flow behavior in a bubble column”, *Chemical Engineering Science*, v. 56, pp. 1737 – 1747, 2001.

- [107] ELGHOBASHI, S. E., ABOU-ARAB, T. W., “A two-equation turbulence model for two-phase flows”, *Physics of Fluids*, v. 26, n. 4, pp. 931–938, 1983.
- [108] SMAGORINSKY, J., “General circulation experiments with the primitive equations: I. the basic equations”, *Mon. Weather Review*, v. 91, pp. 99–164, 1963.
- [109] DEEN, N. G., HJERTAGER, B. H., SOLBERG, T., “Comparison of PIV and LDA Measurement Methods applied to the gas-liquid flow in bubble column”. In: *10th International Symposium on Applications of Laser Techniques to Fluid Mechanics*, Lisboa, Portugal, 2000.
- [110] BERTOLA, F., VANNI, M., BALDI, G., “Application of Computational Fluid Dynamics to Multiphase Flow in Bubble Columns”, *International Journal of Chemical Reactor Engineering*, v. 1, n. A3, pp. 1–14, 2003.
- [111] BOVE, S., SOLBERG, T., HJERTAGER, B. H., “Numerical aspects of bubble column simulations”, *International Journal of Chemical Reactor Engineering*, v. 2, n. A1, pp. 1–22, 2004.
- [112] BEHZADI, A., ISSA, R. I., RUSCHE, H., “Modelling of dispersed bubble and droplet flow at high phase fractions”, *Chemical Engineering Science*, v. 59, pp. 759–770, 2004.
- [113] GOSMAN, A. D., ISSA, R. I., LEKAKOU, C., et al., “Multidimensional modelling of turbulent two-phase flows in stirred vessels”, *A.I.Ch.E. Journal*, v. 38, n. 12, pp. 1946–1956, 1992.
- [114] BEL F'DHILA, R., MASBERNAT, L., SUZANNE, C., “Measurements in two-phase bubbly flows in complex geometries”, In: *Phase-Interface Phenomena in Multiphase Flow*, pp. 245–257, Hemisphere Publ. Corp.: Washington, 1991.
- [115] LANCE, M., MARIÉ, J. L., MOURSALI, E., et al., “Experimental study of turbulent bubbly shear flows”, *Chemical Engineering Communications*, v. 141-142, pp. 51–70, 1996.

- [116] BECH, K., “Dynamic simulation of a 2D bubble column”, *Chemical Engineering Science*, v. 60, pp. 5294–5304, 2005.
- [117] LUCAS, D., PRASSER, H.-M., MANERA, A., “Influence of the lift force on the stability of a bubble column”, *Chemical Engineering Science*, v. 60, pp. 3609–3619, 2005.
- [118] EKAMBARA, K., DHOTRE, M. T., JOSHI, J. B., “CFD simulations of bubble column reactors: 1D, 2D and 3D approach”, *Chemical Engineering Science*, v. 60, pp. 6733–6746, 2005.
- [119] MENZEL, T., WEIDE, T., STAUDACHER, O., et al., “Reynolds shear stress modeling of bubble column reactors”, *Industrial and Engineering Chemistry Research*, v. 29, pp. 988–994, 1990.
- [120] DEVANATHAN, N., MOSLEMIAN, D., DUDUKOVIC, M. P., “Flow mapping in bubble columns using CARPT”, *Chemical Engineering Science*, v. 45, pp. 2285–2291, 1990.
- [121] CHEN, J., KEMOUN, A., AL-DAHMAN, M. H., et al., “Comparative hydrodynamics study in a bubble column using computer-automated radioactive particle tracking (CARPT)/computed tomography (CT) and particle image velocimetry (PIV)”, *Chemical Engineering Science*, v. 54, n. 13-14, pp. 2199–2207, 1999.
- [122] KULKARNI, A. A., *Transport phenomena and non-linear dynamics in multiphase systems*, Ph.D. Thesis, University of Mumbai, India, 2003.
- [123] KRISHNA, R., URSEANU, M. I., VAN BATEN, J. M., et al., “Rise velocity of a swarm of large gas bubbles in liquids”, *Chemical Engineering Science*, v. 54, pp. 171–183, 1999.
- [124] KRISHNA, R., VAN BATEN, J. M., URSEANU, M. I., “Three-phase Eulerian simulations of bubble column reactors operating in the churn-turbulent regime: a scale up strategy”, *Chemical Engineering Science*, v. 55, pp. 3275 – 3286, 2000.

- [125] VAN BATEN, J. M., KRISHNA, R., “Eulerian simulations for determination of the axial dispersion of liquid and gas phases in bubble columns operating in the churn turbulent regime”, *Chemical Engineering Science*, v. 56, pp. 503 – 512, 2001.
- [126] KRISHNA, R., ELLENBERGER, J., HENNEPHOF, D. E., “Analogous description of gas-solid fluidized beds and bubble columns”, *Chemical Engineering Journal*, v. 53, pp. 89–101, 1993.
- [127] ELLENBERGER, J., KRISHNA, R., “A unified approach to the scaleup of gas solid fluidized and gas-liquid bubble column reactors”, *Chemical Engineering Science*, v. 49, pp. 5391–5411, 1994.
- [128] HARMATHY, T. J., “Velocity of large drops and bubbles in media of infinite or restricted extent”, *AIChE Journal*, v. 6, n. 2, pp. 281 – 288, 1960.
- [129] MENDELSON, H. D., “The prediction of bubble terminal velocities from wave theory”, *AIChE Journal*, v. 13, pp. 250–253, 1967.
- [130] DAVIES, R. M., TAYLOR, G. I., “The mechanics of large bubbles rising through extended liquids and through liquids in tubes”, *Proceedings of the Royal Society of London*, v. A200, pp. 375–390, 1950.
- [131] COLLINS, R., “The effect of a containing cylindrical boundary on the velocity of a large gas bubble in a liquid”, *Journal of Fluid Mechanics*, v. 28, pp. 97–112, 1967.
- [132] KRISHNA, R., ELLENBERGER, J., “Gas hold-up in bubble column reactors operating in the churn-turbulent flow regime”, *AIChE Journal*, v. 42, pp. 2627–2634, 1996.
- [133] KRISHNA, R., DE SWART, J. W. A., ELLENBERGER, J., et al., “Gas hold-up in slurry bubble columns”, *AIChE Journal*, v. 43, pp. 311–316, 1997.

- [134] HULBURT, H., KATZ, S., “Some problems in particle technology. A statistical mechanical formulation”, *Chemical Engineering Science*, v. 19, pp. 555–574, 1964.
- [135] VALENTAS, K. J., AMUNDSON, A. R., “Breakage and coalescence in dispersed phase systems”, *Industrial and Engineering Chemistry: Fundamentals*, v. 5, n. 4, pp. 533–542, 1966.
- [136] SMOLUCHOWSKI, M., “Drei Vorträge über Diffusion, Brownsche Molekularbewegung und Koagulation von Kolloidteilchen”, *Phys. Zeitchr.*, v. 17, pp. 585–599, 1916.
- [137] SMOLUCHOWSKI, “Versuch einer mathematischen Theorie der Koagulationskinetik kolloider Lösungen”, *Zeitschrift für Physikalische Chemie*, v. 92, pp. 129–168, 1917.
- [138] RANDOLPH, A. D., LARSON, M. A., *Theory of Particulate Processes*. 2nd ed. Academic Press: Nova York, EUA, 1988.
- [139] MANTZARIS, N. V., “A cell population balance model describing positive feedback loop expression dynamics”, *Computers and Chemical Engineering*, v. 29, n. 4, pp. 897–909, 2005.
- [140] SCHWARZERA, H.-C., SCHWERTFIRMB, F., MANHARTB, M., et al., “Predictive simulation of nanoparticle precipitation based on the population balance equation”, *Chemical Engineering Science*, v. 61, n. 1, pp. 167–181, 2006.
- [141] SMAGALAA, T. G., MCCOY, B. J., “Population balance modeling of polymer branching and hyperbranching”, *Chemical Engineering Science*, v. 61, n. 1, pp. 3–17, 2006.
- [142] NERE, N. K., RAMKRISHNA, D., “Solution of population balance equation with pure aggregation in a fully developed turbulent pipe flow”, *Chemical Engineering Science*, v. 61, n. 1, pp. 96–103, 2006.

- [143] HARPER, W. R., “An exact theory of the coagulation of spherical particles arising from thermal agitation”, *Transaction of Faraday Society*, v. 32, pp. 1139–1144, 1936.
- [144] HOWARTH, W. J., “Coalescence of drops in a turbulent flow field”, *Chemical Engineering Science*, v. 19, n. 1, pp. 33–38, 1964.
- [145] COULALOGLOU, C. A., TAVLARIDES, L. L., “Description of interaction processes in agitated liquid-liquid dispersions”, *Chemical engineering Science*, v. 32, pp. 1289–1297, 1977.
- [146] PARICHAY, K. D., KUMAR, R., RAMKRISHNA, D., “Coalescence of drops in stirred dispersion. A white noise model for coalescence”, *Chemical Engineering Science*, v. 42, n. 2, pp. 213–220, 1987.
- [147] LIU, L. X., LITSTER, J. D., IVESON, S. M., et al., “Coalescence of Deformable Granules in Wet Granulation Processes”, *AIChE Journal*, v. 46, n. 3, pp. 529 – 539, 2000.
- [148] CAMPBELL, G. M., WEBB, C., “On predicting roller milling performance: Part I: the breakage equation”, *Powder Technology*, v. 115, n. 3, pp. 234 – 242, 2001.
- [149] CAMPBELL, G. M., BUNN, P. J., WEBB, C., et al., “On predicting roller milling performance: Part II. The breakage function”, *Powder Technology*, v. 115, n. 3, pp. 243 – 255, 2001.
- [150] DAVIS, “On the statistics of randomly broken objects”, *Chemical Engineering Science*, v. 44, n. 9, pp. 1799–1805, 1989.
- [151] HINZE, J. O., “Fundamentals of the hydrodynamic mechanism of splitting in dispersions processes”, *AIChE Journal*, v. 1, n. 3, pp. 289–295, 1955.
- [152] SHINNAR, R., “On the behavior of liquid dispersions in mixing vessels”, *Journal of Fluid Mechanics*, v. 10, pp. 259–275, 1961.

- [153] NARSIMHAN, G., GUPTA, J. P., RAMKRISHNA. D., “A model for transitional breakage probability of droplets in agitated lean liquid-liquid dispersions”, *Chemical Engineering Science*, v. 34, n. 2, pp. 257–265, 1979.
- [154] TSOURIS, C., TAVLARIDES, L. L., “Breakage and coalescence models for drops in turbulent dispersions”, *AIChE Journal*, v. 40, pp. 395–406, 1994.
- [155] LUO, H., SVENDSEN, H. F., “Theoretical model for drop and bubble breakup in turbulent dispersions”, *AIChE Journal*, v. 42, n. 5, pp. 1225–1233, 1996.
- [156] VIGIL, R. D., ZIFF, R. M., “On the stability of coagulation-fragmentation population balances”, *Journal of Colloid and Interface Science*, v. 133, n. 1, pp. 257–264, 1989.
- [157] PATIL, D. P., ANDREWS, J. R. G., “An analytical solution to continuous population balance model describing floc coalescence and breakage - A special case”, *Chemical Engineering Science*, v. 53, n. 3, pp. 599–601, 1998.
- [158] LAGE, P. L. C., “Comments on the “An analytical solution to the population balance equation with coalescence and breakage – the special case with constant number of particles” by D.P. Patil and J.R.G. Andrews [Chemical Engineering Science 53(3) 599–601]”, *Chemical Engineering Science*, v. 57, n. 57, pp. 4253–4254, 2002.
- [159] DIEMER, R. B., OLSON, J. H., “A moment methodology for coagulation and breakage problems: Part 1 – analytical solution of the steady-state population balance”, *Chemical Engineering Science*, v. 57, pp. 2193–2209, 2002.
- [160] MCCOY, B. J., MADRAS, G., “Analytical solution for a population balance equation with aggregation and fragmentation”, *Chemical Engineering Science*, v. 58, pp. 3049–3051, 2003.

- [161] RAMKRISHNA, D., BORWANKER, J. D., “A puristic analysis of population balance – I”, *Chemical Engineering Science*, v. 28, n. 7, pp. 1423–1435, 1973.
- [162] RAMKRISHNA, D., BORWANKER, J. D., “A puristic analysis of population balance – II”, *Chemical Engineering Science*, v. 29, n. 8, pp. 1711 – 1721, 1974.
- [163] SHAH, B. H., RAMKRISHNA, D., BORWANKER, J., “Simulation of particulate systems using the concept of the interval of quiescence”, *AIChE Journal*, v. 23, pp. 897 – 904, 1977.
- [164] KENDALL, D., “Stochastic Processes and Population Growth”, *Journal of the Royal Statistical Society. Series B*, v. 11, pp. 230–282, 1949.
- [165] SWEET, I. R., GUSTAFSON, S. S., RAMKRISHNA, D., “Population Balance Modelling of Bubbling Fluidized Bed Reactors - I. Well-stirred dense phase”, *Chemical Engineering Science*, v. 42, n. 2, pp. 341–351, 1987.
- [166] DAS, P. K., “Monte Carlo simulation of drop breakage on the basis of drop volume”, *Computer and Chemical Engineering*, v. 20, n. 3, pp. 307–313, 1996.
- [167] GOODSON, M., KRAFT, M., “Simulation of coalescence and breakage: an assessment of two stochastic methods suitable for simulating liquid-liquid extraction”, *Chemical Engineering Science*, v. 59, pp. 3865 – 3881, 2004.
- [168] ZHAO, H., MAISELS, A., MATSOUKAS, T., et al., “Analysis of four Monte Carlo methods for the solution of population balances in dispersed systems”, *Powder Technology*, v. 173, pp. 38–50, 2007.
- [169] IRIZARRY, R., “Fast Monte Carlo methodology for multivariate particulate systems - I: Point ensemble Monte Carlo”, *Chemical Engineering Science*, v. 63, pp. 95–110, 2008.

- [170] IRIZARRY, R., “Fast Monte Carlo methodology for multivariate particulate systems - II: τ -PEMC”, *Chemical Engineering Science*, v. 63, pp. 111–121, 2008.
- [171] BHATIA, S. K., CHAKRABORTY, D., “Modified MWR approach: Application to agglomerative precipitation”, *AIChE Journal*, v. 38, n. 6, pp. 868–878, 1992.
- [172] GELBARD, F., SEINFELD, J. H., “Numerical solution of the dynamic equation for particulate systems”, *Journal of Computational Physics*, v. 28, pp. 357–375, 1978.
- [173] EYRE, D., WRIGHT, C. J., REUTER, G., “Spline-collocation with adaptive mesh grading for solving the stochastic collection equation”, *Journal of Computational Physics*, v. 78, pp. 288–304, 1988.
- [174] NICMANIS, M., HOUNSLOW, M. J., “Finite-element methods for steady-state population balance equations”, *AIChE Journal*, v. 44, pp. 2258–2272, 1998.
- [175] LIU, Y., CAMERON, I. T., “A new wavelet-based method for the solution of the population balance equation”, *Chemical Engineering Science*, v. 56, pp. 5283 – 5294, 2001.
- [176] MAHONEY, A. W., RAMKRISHNA, D., “Efficient solution of population balance equation with discontinuities by finite elements”, *Chemical Engineering Science*, v. 57, pp. 1107–1119, 2002.
- [177] RIGOPOULOS, S., JONES, A. G., “Finite-Element Scheme for Solution of the Dynamic Population Balance Equation”, *AIChE Journal*, v. 49, n. 5, pp. 1127 – 1139, 2003.
- [178] DORAO, C. A., JAKOBSEN, H. A., “Application of the least square method to population balance problems”, *Computers and Chemical Engineering*, v. 30, n. 3, pp. 535–547, 2006.

- [179] DORAO, C. A., JAKOBSEN, H. A., “Application of the least square method for solving population balance problems in R^{d+1} ”, *Chemical Engineering Science*, v. 61, n. 15, pp. 5070–5081, 2006.
- [180] DORAO, C. A., JAKOBSEN, H. A., “Least-squares spectral method for solving advective population balance problems”, *Journal of Computational and Applied Mathematics*, v. 201, n. 1, pp. 247–257, 2007.
- [181] DORAO, C. A., JAKOBSEN, H. A., “Time-space-property least squares spectral method for population balance problems”, *Chemical Engineering Science*, v. 62, n. 5, pp. 1323–1333, 2007.
- [182] PINTO, J. C., LAGE, P. L. C., *Métodos Numéricos em Problemas de Engenharia Química*. E-papers: Brasil, RJ, 2001.
- [183] KOSTOGLU, M., KARABELAS, A. J., “Evaluation of zero order methods for simulating particle coagulation”, *Journal of Colloid and Interface Science*, v. 163, pp. 420–431, 1994.
- [184] KUMAR, S., RAMKRISHNA, D., “On the solution of population balance equations by discretization - I. a fixed pivot technique”, *Chemical Engineering Science*, v. 51, n. 8, pp. 1311–1332, 1996.
- [185] HOUNSLOW, M. J., RYALL, R. L., MARSHALL, V. R., “A discretized population balance for nucleation, growth and aggregation”, *AIChE Journal*, v. 34, n. 11, pp. 1821–1832, 1988.
- [186] LISTER, J. D., SMITH, D. J., HOUNSLOW, M. J., “Adjustable discretized population balance for growth and aggregation”, *AIChE Journal*, v. 41, n. 3, pp. 591–603, 1995.
- [187] HILL, P., NG, K., “New discretisation procedure for the breakage equation”, *AIChE Journal*, v. 41, n. 5, pp. 1204–1216, 1995.

- [188] WYNN, E. J. W., “Simulating Aggregation and Reaction: New Hounslow DPB and Four-Parameter Summary”, *AIChE Journal*, v. 50, n. 3, pp. 578 – 588, 2004.
- [189] ALEXOPOULOS, A. H., ROUSSOS, A. I., KIPARISSIDES, C., “Part I: dynamic evolution of the particle size distribution in particulate processes undergoing combined particle growth and aggregation”, *Chemical Engineering Science*, v. 59, pp. 5751– 5769, 2004.
- [190] ALEXOPOULOS, A. H., ROUSSOS, A. I., KIPARISSIDES, C., “Part II: dynamic evolution of the particle size distribution in particulate processes undergoing particle nucleation, growth and aggregation”, *Chemical Engineering Science*, v. 60, pp. 4157 – 4169, 2005.
- [191] KUMAR, J., PEGLOW, M., WARNECKE, G., et al., “Improved accuracy and convergence of discretized population balance for aggregation: The cell average technique”, *Chemical Engineering Science*, v. 61, pp. 3327– 3342, 2006.
- [192] KUMAR, J., PEGLOW, M., WARNECKE, G., et al., “The cell average technique for solving multi-dimensional aggregation population balance equations”, No prelo doi: 10.1016/j.compchemeng.2007.10.001, 2007.
- [193] VANNI, M., “Approximate population balance equations for aggregation-breakage processes”, *Journal of Colloid and Interface Science*, v. 221, pp. 143 – 160, 2000.
- [194] BATTERHAM, R. J., HALL, J. S., BARTON, G., “Pelletizing kinetics and simulation of full-scale balling circuits”. In: *3rd International Symposium on Agglomeration*, v. A136, Nürnberg, Alemanha, 1981.
- [195] GELBARD, F., TAMBOUR, Y., SEINFELD, J. H., “Sectional representation of simulating aerosol dynamics”, *Journal of Colloids and Interface Science*, v. 76, pp. 541–556, 1980.

- [196] MARCHAL, P., DAVID, R., KLEIN, J. P., et al., “Crystallization and Precipitation Engineering – I. An Efficient Method for Solving Population Balance in Crystallization and Agglomeration”, *Chemical Engineering Science*, v. 43, n. 1, pp. 59 – 67, 1988.
- [197] VANNI, M., “Discretization procedure for the breakage equation”, *AIChE Journal*, v. 45, pp. 916–919, 1999.
- [198] KUMAR, S., RAMKRISHNA, D., “On the solution of population balance equations by discretization - II. a moving pivot technique”, *Chemical Engineering Science*, v. 51, n. 8, pp. 1333–1342, 1996.
- [199] ATTARAKIH, M. M., BART, H. J., FAQIR, N. M., “Optimal moving and fixed grids for the solution of discretized population balance in batch and continuous systems: droplet breakage”, *Chemical Engineering Science*, v. 58, pp. 1251 – 1269, 2003.
- [200] NOPENS, I., BEHEYDT, D., VANROLLEGHEM, P. A., “Comparison and pitfalls of different discretised solution methods for population balance models: a simulation study”, *Computers and Chemical Engineering*, v. 29, pp. 367–377, 2005.
- [201] KUMAR, S., RAMKRISHNA, D., “On the solution of population balance equations by discretization — III. Nucleation, growth and aggregation of particles”, *Chemical Engineering Science*, v. 52, pp. 4659–4679, 1997.
- [202] LEE, G., YOON, E. S., LIM, Y. I., et al., “Adaptative mesh method for the simulation of crystallization processes including agglomeration and breakage: the potassium sulfate system”, *Industrial and Engineering Chemistry Research*, v. 40, pp. 6228 – 6235, 2001.
- [203] HUANG, W., REN, Y., RUSSELL, R. D., “Moving Mesh Methods Based on Moving Mesh Partial Differential Equations”, *Journal of Computational Physics*, v. 113, n. 2, pp. 279–290, 1994.

- [204] HU, Q., ROHANI, S., JUTAN, A., “New numerical method for solving the dynamic population balance equations”, *AIChE Journal*, v. 51, n. 11, pp. 3000–3006, 2005.
- [205] FRIEDLANDER, S. K., *Smoke, dust and haze*. Wiley: Nova York, 1977.
- [206] MCGRAW, R., SAUNDERS, J. H., “A condensation feedback mechanism for oscillatory nucleation and growth”, *Aerosol Science and Technology*, v. 3, pp. 367–380, 1984.
- [207] BARRETT, J. C., JHEETA, J. S., “Improving the accuracy of the moments method for solving the aerosol general dynamic equation”, *Journal of Aerosol Science*, v. 27, n. 8, pp. 1135–1142, 1996.
- [208] MCGRAW, R., “Description of the aerosol dynamics by the quadrature method of moments”, *Aerosol Science and Technology*, v. 27, pp. 255–265, 1997.
- [209] GORDON, R. G., “Error bounds in equilibrium statistical mechanics”, *A.I.Ch.E. Journal*, v. 9, n. 5, pp. 655–663, 1968.
- [210] MARCHISIO, D. L., VIGIL, R. D., FOX, R. O., “Quadrature method of moments for aggregation-breakage processes”, *Journal of Colloid and Interface Science*, v. 258, pp. 322–334, 2003.
- [211] MARCHISIO, D. L., FOX, R. O., “Solution of the population balance equation using the direct quadrature method of moments”, *Journal of Aerosol Science*, v. 36, pp. 43–73, 2005.
- [212] BOVE, S., SOLBERG, T., HJERTAGER, B. H., “A novel algorithm for solving population balance equations: The parallel parent and daughter classes. Derivation, analysis and testing”, *Chemical Engineering Science*, v. 60, pp. 1449 – 1464, 2005.
- [213] WRIGHT, D. L., MCGRAW, R., ROSNER, D. E., “Bivariant extension of the quadrature method of moments for modeling simultaneous coagulation

- and sintering of particle populations”, *Journal of Colloid and Interface Science*, v. 236, pp. 242 – 251, 2001.
- [214] MCGRAW, R., WRIGHT, D. L., “Chemically resolved aerosol dynamics for internal mixtures by the quadrature method of moments”, *Journal of Aerosol Science*, v. 34, pp. 189 – 209, 2003.
- [215] YOON, C., MCGRAW, R., “Representation of generally mixed multivariate aerosols by the quadrature method of moments: I. Statistical foundation”, *Journal of Aerosol Science*, v. 35, pp. 561–576, 2004.
- [216] YOON, C., MCGRAW, R., “Representation of generally mixed multivariate aerosols by the quadrature method of moments: II. Aerosol dynamics”, *Journal of Aerosol Science*, v. 35, pp. 577–598, 2004.
- [217] JOHNSON, R. A., WICHERN, D. W., *Applied multivariate statistical analysis*. Prentice-Hall: Upper Saddle River, NJ, 1992.
- [218] DIAMANTARAS, K. I., KUNG, S. Y., *Principal component neural networks: Theory and applications*. Wiley: Nova York, 1996.
- [219] FOX, R. O., “Bivariate direct quadrature method of moments for coagulation and sintering of particle populations”, *Journal of Aerosol Science*, v. 37, n. 11, pp. 1562–1580, 2006.
- [220] DIEMER JR., R. B., OLSON, J. H., “Bivariate moment methods for simultaneous coagulation, coalescence and breakup”, *Journal of Aerosol Science*, v. 37, n. 3, pp. 363–385, 2006.
- [221] LO, S., *Application of the musig model to bubbly flows*, Tech. Rep. AEAT-1096, AEA Technology, 1996.
- [222] KREPPER, E., LUCAS, D., PRASSER, H.-M., “On the modelling of bubbly flow in vertical pipes”, *Nuclear Engineering and Design*, v. 235, pp. 597–611, 2005.

- [223] FRANK, T., ZWART, P. J., SHI, J.-M., et al., “Inhomogeneous MUSIG Model - a Population Balance Approach for Polydispersed Bubbly Flows”. pp. 067.1–067.14, Bled, Eslovênia, 2005.
- [224] FRANK, T., ZWART, P. J., KREPPER, E., et al., “Validation of CFD models for mono- and polydisperse air-water two-phase flows in pipes”, *Nuclear Engineering and Design*, v. 238, pp. 647–659, 2008.
- [225] ANSYS INC., *ANSYS CFX-11.0 User Manual*, 2007.
- [226] LAGE, P. L. C., “The quadrature method of moments for continuous thermodynamics”, *Computers and Chemical Engineering*, v. 31, n. 7, pp. 782–799, 2006.
- [227] OLMOS, E., GENTRIC, C., VIAL, C., et al., “Numerical simulation of multiphase flow in bubble column reactors. Influence of bubble coalescence and break-up”, *Chemical Engineering Science*, v. 56, pp. 6359 – 6365, 2001.
- [228] KURUL, N., PODOWSKI, M. Z., “Multi-dimensional effects in sub-cooled boiling”. In: *9th heat transfer conference*, 1990.
- [229] PRINCE, M. J., BLANCH, H. W., “Bubble coalescence and breakup in air-sparged bubble columns”, *AIChE Journal*, v. 36, pp. 1485–1499, 1990.
- [230] OLMOS, E., GENTRIC, C., MIDOUX, N., “Numerical description of flow regime transitions in bubble column reactors by a multiple gas phase model”, *Chemical Engineering Science*, v. 58, pp. 2113 – 2121, 2003.
- [231] GIDASPOW, D., *Multiphase Flow and Fluidization*. Academic Press: San Diego, 1994.
- [232] SCHILLER, L., NAUMANN, A., “Über die grundlegenden berechnungen bei der schwerkraftbereitung”, *Z. Vereins deutscher Ing.*, v. 77, n. 12, pp. 318–320, 1933.

- [233] TCHEN, C. M., *Mean value and correlation problems connected with the motion of small particles suspended in a turbulent fluid*, Ph.D. Thesis, TU Delft, Holanda, 1947.
- [234] LUO, H., *Coalescence, breakup and liquid circulation in bubble column reactors*, Ph.D. Thesis, University of Trondheim, Trondheim, Noruega, 1993.
- [235] KUMAR, S. B., MOSLEMIAN, D., DUDUKOVIC, M. P., “Gas-holdup measurements in bubble columns using computed tomography”, *AIChE Journal*, v. 43, n. 6, pp. 1414–1425, 1997.
- [236] CHEN, P., SANYAL, J., DUDUKOVIC, M. P., “Numerical simulation of bubble columns flows: effect of different breakup and coalescence closures”, *Chemical Engineering Science*, v. 60, pp. 1085–1101, 2005.
- [237] MARTÍNEZ-BAZÁN, C., MONTAÑÉS, J. L., LASHERAS, J. C., “On the breakup of an air bubble injected into a fully developed turbulent flow. Part 1. Breakup frequency”, *Journal of Fluid Mechanics*, v. 401, pp. 157–182, 1999.
- [238] MARTÍNEZ-BAZÁN, C., MONTAÑÉS, J. L., LASHERAS, J. C., “On the breakup of an air bubble injected into a fully developed turbulent flow. Part 2. Size PDF of the resulting daughter bubbles”, *Journal of Fluid Mechanics*, v. 401, pp. 183–207, 1999.
- [239] RIBEIRO JR., C. P., *Desenvolvimento de um Processo Combinado de Evaporação por Contato Direto e Permeação de Vapor para Tratamento de Sucos*, Ph.D. Thesis, Universidade Federal do Rio de Janeiro, PEQ/COPPE, RJ, Brasil, 2005.
- [240] CHEN, P., DUDUKOVIC, M. P., SANYAL, J., “Three-Dimensional Simulation of Bubble Column Flows with Bubble Coalescence and Breakup”, *AIChE Journal*, v. 51, n. 3, pp. 696–712, 2005.

- [241] CHEN, J., LI, F., DEGALEESAN, S., et al., “Fluid dynamic parameters in bubble columns with internals”, *Chemical Engineering Science*, v. 54, pp. 2187–2197, 1999.
- [242] ONG, B., *Experimental investigation of bubble column hydrodynamics – Effect of elevated pressure and superficial gas velocity*, Ph.D. Thesis, Washington University, St. Louis, MO, 2003.
- [243] SHAIKH, A., RADOS, N., AL-DAHMAN, M. H., “Phase Distribution in a High Pressure Slurry Bubble Column via Computed Tomography”. In: *4th Middle East Refining and Petrochemical Exhibition and Conference*, Manama, Bahrain, 2003.
- [244] SAFFMAN, P. G., TURNER, J. S., “On the collisions of drops in turbulent clouds”, *Journal of Fluid Mechanics*, v. 1, pp. 16–30, 1956.
- [245] MARCHISIO, D. L., VIGIL, R. D., FOX, R. O., “Implementation of the quadrature method of moments in CFD codes for aggregation-breakage problems”, *Chemical Engineering Science*, v. 58, pp. 3337–3351, 2003.
- [246] PRAT, O. P., DUCOSTE, J. J., “Modeling spatial distribution of floc size in turbulent processes using the quadrature method of moment and computational fluid dynamics”, *Chemical Engineering Science*, v. 61, pp. 75–86, 2006.
- [247] ADLER, P., “Heterocoagulation in shear flow”, *Journal of Colloids and Interface Science*, v. 83, n. 1, pp. 106–115, 1981.
- [248] KUSTERS, K. A., *The influence of turbulence on aggregation of small particles in agitated vessels*, Ph.D. Thesis, University of Eindhoven, Holanda, 1991.
- [249] PRESS, W. H., FLANNERY, B. P., TEUKOLSKY, S. A., et al., *Numerical Recipes – The Art of Scientific Computing (FORTRAN version)*. Cambridge University Press: Cambridge, 1990.

- [250] PRESS, W. H., TEUKOLSKY, S. A., “Orthogonal polynomials and gaussian quadrature with nonclassical weights”, *Computers in Physics*, v. 4, pp. 423–426, 1990.
- [251] FAN, R., MARCHISIO, D. L., FOX, R. O., “Application of the direct quadrature method of moments to polydisperse gas-solid fluidised beds”, *Powder Technology*, v. 139, pp. 7 – 20, 2004.
- [252] PATANKAR, S. V., SPALDING, D. B., “A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows”, *International Journal of Heat and Mass Transfer*, v. 15, n. 10, pp. 1787–1806, 1972.
- [253] SPALDING, D. B., “Numerical computation of multi-phase fluid flow and heat transfer”, In: *Recent Advances in Numerical Methods in Fluids*, v. 1, pp. 39–68, Pineridge Press, 1980.
- [254] FAN, R., FOX, R. O., “Segregation in polydisperse fluidized beds: Validation of a multi-fluid model”, *Chemical Engineering Science*, v. 63, n. 1, pp. 272–285, 2008.
- [255] GOLDSCHMIDT, M. J. V., LINK, J. M., MELLEMA, S., et al., “Digital image analysis measurements of bed expansion and segregation dynamics in dense gas-fluidized beds”, *Powder Technology*, v. 138, pp. 135–159, 2003.
- [256] DAHL, S. R., HRENYA, C. M., “Size segregation in gas-solid fluidized beds with continuous size distributions”, *Chemical Engineering Science*, v. 60, pp. 6658–6673, 2005.
- [257] ZUCCAA, A., MARCHISIO, D. L., BARRESIA, A. A., et al., “Implementation of the population balance equation in CFD codes for modelling soot formation in turbulent flames”, *Chemical Engineering Science*, v. 61, n. 1, pp. 87–95, 2006.

- [258] FOX, R., *Computational models for turbulent reacting flows*. Cambridge University Press, 2003.
- [259] KENT, J. H., HONNERY, D., “Soot and mixture fraction in turbulent diffusion flames”, *Combustion Science and Technology*, v. 54, pp. 383–397, 1987.
- [260] SILVA, L. F. L. R., DAMIAN, R. B., LAGE, P. L. C., “Implementation and analysis of numerical solution of the population balance equation in CFD packages”, *Computers and Chemical Engineering*, v. doi:10.1016/j.compchemeng.2008.03.007, 2008.
- [261] PATANKAR, S. V., *Numerical Heat Transfer and Fluid Flow*. Taylor & Francis Group: Nova York, 1980.
- [262] MALISKA, C. R., *Transferência de Calor e Mecânica dos Fluidos Computacional*. 2nd ed. LTC: Rio de Janeiro, 2004.
- [263] JASAK, H., *Error analysis and estimation for the finite volume method with applications to fluid flows*, Ph.D. Thesis, Imperial College of Science, Technology and Medicine, Londres, Reino Unido, 1996.
- [264] VERSTEEG, H. K., MALALASEKERA, W., *An introduction to computational fluid dynamics - The finite volume method*. Longman Scientific and Technical: Reino Unido, 1995.
- [265] RHIE, C. M., CHOW, W. L., “A numerical study of the turbulent flow past an isolated airfoil with trailing edge separation”, *American Institute of Aeronautics and Astronautics*, v. 21, n. 11, pp. 1525–1532, 1983.
- [266] PERIC, M., KESSLER, R., SCHEUERER, G., “Comparison of Finite Volume Numerical Methods with Staggered and Colocated Grids”, *Computers and Fluids*, v. 16, n. 4, pp. 389–403, 1988.
- [267] TORO, E. F., *Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction*. 2nd ed. Springer: Nova York, 1999.

- [268] SAAD, Y., *Iterative Methods for Sparse Linear Systems*. 2nd ed. SIAM, 2003.
- [269] WELLER, H. G., *Derivation, modelling and solution of the conditionally averaged two-phase flow equations*, Tech. Rep. TR/HGW/02, Nabla Ltd., 2002.
- [270] DEITEL, P. J., DEITEL, H. M., *C++ How to Program*. 6th ed. Prentice Hall, 2007.
- [271] YANG, D., *C++ and Object-Oriented Numeric Computing for Scientists and Engineers*. Springer: Nova York, 2001.
- [272] MALAN, A. G., LEWIS, R. W., “On the development of high-performance C++ object-oriented code with application to an explicit edge-based fluid dynamics scheme”, *Computers and Fluids*, v. 33, n. 10, pp. 1291 – 1304, 2004.
- [273] POLITIS, S., *Prediction of Two-Phase Solid-Liquid Turbulent Flow Stirred Vessels*, Ph.D. Thesis, University of London, Imperial College, 1989.
- [274] OLIVEIRA, P. J., *Computer Modelling of Multidimensional Multiphase Flow and Application to T-Junctions*, Ph.D. Thesis, University of London, Imperial College, 1992.
- [275] HILL, D. P., *The Computer Simulation of Dispersed Two-Phase Flows*, Ph.D. Thesis, Imperial College of Science, Technology and Medicine, Londres, Reino Unido, 1998.
- [276] UBBINK, O., *Numerical Prediction of Two Fluid Systems with Sharp Interfaces*, Ph.D. Thesis, Imperial College of Science, Technology and Medicine, Londres, Reino Unido, 1997.
- [277] BRENNAN, D., *The Numerical Simulation of Two-Phase Flows in Settling Tanks*, Ph.D. Thesis, Imperial College of Science, Technology and Medicine, Londres, Reino Unido, 2001.

- [278] DOPAZO, C., “On conditional averages for intermittent turbulent flows”, *Journal of Fluid Mechanics*, v. 81, n. 3, pp. 433 – 438, 1977.
- [279] WEN, C. Y., YU, Y. H., “Mechanics of fluidization”, In: *Chem. Eng. Prog. Symp. Ser.*, v. 62, pp. 100–111, 1966.
- [280] ISSA, R. I., “Solution of the implicitly discretised fluid flow equations by operator-splitting”, *Journal of Computational Physics*, v. 62, n. 1, pp. 40–65, 1986.
- [281] SILVA, L. F. L. R., DAMIAN, R. B., LAGE, P. L. C., “Implementation and analysis of numerical solution of the population balance equation in CFD packages”. In: *available on CD*, Leipzig, Alemanha, 2007.
- [282] PETZOLD, L. R., *A description of DASSL: A differential algebraic system solver*, Sand82-8637, Sandia National Laboratories, 1982.
- [283] KARYPIS, G., KUMAR, V., “A fast and high quality multilevel scheme for partitioning irregular graphs”, *SIAM Journal on Scientific Computing*, v. 20, n. 1, pp. 359 – 392, 1999.
- [284] KAREMA, H., LO, S., “Efficiency of interphase coupling algorithms in fluidized bed conditions”, *Computers and Fluids*, v. 28, pp. 323–360, 1999.

Apêndice A

Comparação dos Métodos de Solução para EBP

A.1 Informações sobre o Apêndice

Este trabalho foi submetido recentemente para publicação em um periódico.

Comparison of the accuracy and performance of quadrature-based methods for population balance problems with simultaneous breakage and aggregation

L.F.L.R. Silva, R.C. Rodrigues, J.F. Mitre and P.L.C. Lage*

Programa de Engenharia Química — COPPE, Universidade Federal do Rio de Janeiro, PO Box 68502, Rio de Janeiro, RJ, 21941-972 Brazil

Abstract

Simulations of polydisperse multiphase flows must include the effects of particle breakage and aggregation, which requires the solution of the population balance equation (PBE). Therefore, the analysis of the existing numerical techniques to solve the PBE regarding their efficiency and accuracy is paramount to their implementation in CFD codes. This work focused on analyzing the three quadrature-based methods available in the literature (QMOM, DQMOM and PPDC) in terms of efficiency and accuracy and against the classical method of classes. Analytical solutions were used to derive test cases from dominant breakage to dominant aggregation. The methods were evaluated in terms of moment accuracy and convergence. The computational costs were evaluated for all cases. It was verified that PPDC has poor convergence and is not adequate. For all cases, the QMOM and DQMOM presented similar accuracy which was the best one achieved, but the DQMOM was always the most efficient method.

Key words: population balance, aggregation, breakage, QMOM, DQMOM, PPDC

1 Introduction

Disperse multiphase flow occurs in several equipment used in the modern chemical industry, as bubble columns, slurry reactors, liquid-liquid extractors

* Corresponding author. Tel: +55-21-2562-8346; Fax: 55-21-2562-8300.
Email address: paulo@peq.coppe.ufrj.br (P.L.C. Lage).

and spray towers. The complex interactions among the particles play an important role in such flows by changing the dynamical behavior of the particle population properties and the flow. The proper modelling of these systems involves the solution of the population balance equation (PBE). The PBE is the conservation equation for the mean number density distribution function of particles, $f(\mathbf{v}, t)$, whose dimensions depend on the particle properties, \mathbf{v} , considered as distribution variables. For an unique particle property, the problem is called monovariated, as described by Hulburt and Katz (1964) and Ramkrishna (2000), and the PBE is given by:

$$\frac{\partial f(v, t)}{\partial t} + \frac{\partial}{\partial v} [\dot{V}(v, t)f(v, t)] = H(v, t) \quad (1)$$

where v is the particle property, $\dot{V}(v, t)$ is the time variation of the particle property for particles with property v and $H(v, t)$ is the net number rate of particles generated with property v by unit volume of the particle space. For a problem without physical space variations where v is the particle volume with L^3 units, as in the present work, the units of $f(v, t)$, $\dot{V}(v, t)$ and $H(v, t)$ are L^{-3} , $(L^3 T^{-1})$ and $(TL^3)^{-1}$, respectively.

The last term of Eq. (1) usually involves integral functionals of $f(v, t)$ due to the breakage and aggregation processes, turning it into an integro-differential equation. Therefore, analytical solutions can only be obtained in very simple cases, making numerical methods essential to solve the PBE for most practical applications.

1.1 Numerical Methods for the PBE

There are several numerical techniques available to solve Eq.(1), such as the Monte Carlo method (Ramkrishna and Borwanker, 1973, 1974), weighted residuals based methods (Gelbard and Seinfeld, 1978), the method of classes (MoC) (Hounslow et al., 1988; Kumar and Ramkrishna, 1996a,b; Vanni, 1999) and the method of moments (MOM) (Hulburt and Katz, 1964; McGraw and Saunders, 1984). These numerical techniques are summarized in Ramkrishna (2000), but all of them are too time-consuming for coupling to CFD simulations.

From these techniques, the MoC is considered an usually acceptable balance between accuracy and computational effort for monovariate population balance problems. In the MoC, the continuous range of the internal coordinate, v , is partitioned into a finite series of contiguous intervals defined by v_i and v_{i+1} , $i = 0, \dots, n$. The $[v_i, v_{i+1}]$ interval is the i class, in which a single property value, the pivot or abscissa ξ_i , is used to represent all particles belonging

to this interval. This discretization process transforms the integro-differential PBE into a set of differential equations that can be solved by standard solvers. There are several MoC variants, which mainly differ on the choice of the discretization grid and on the properties that are assumed to be conserved during the discretization. The classic formulation of MoC (Ramkrishna, 2000) utilizes fixed pivots to discretize the particle domain, generating a set of differential equations to be solved for the particle number density in each interval, N_i .

In the MOM, a solution is sought for the lower-order moments of the number density distribution function which are sometimes sufficient to provide a statistical description of the particulate system (Friedlander, 1977). The k^{th} moment of a distribution is defined as:

$$\mu_k = \int_0^{\infty} v^k f(v, t) dv, \quad k = 0, \dots, \infty \quad (2)$$

The method is based on the PBE integration with the operator $\int_0^{\infty} v^k \cdot dv$, giving differential equations for the moments. At this point the MOM weakness appears, because, except for very special cases, high-order moments are present in the differential equations for the lower-order moments. This closure problem has to be dealt with closure laws for some of the largest moments, which are approximated relations that allows the calculation of the largest moments ($\mu_k, k \geq n$) that appears in the set of differential equations for the first n moments ($\mu_k, k = 0, \dots, n - 1$). This lack of closure is a severe drawback to the MOM, because it introduces an unknown error in the solution. This was the main reason that make the MOM unpopular in the academic community for several decades.

McGraw (1997) proposed the Quadrature Method of Moments (QMOM) that solves the closure problem by approximating the terms in the integrated PBE by a Gaussian quadrature that can be calculated using only those lower-order moments of the distribution function which are been solved for. The essence of this approach lies in determining this Gaussian quadrature whose weight function is the unknown distribution function. This can be done representing the distribution function with a set of delta functions with weight functions w_i and abscissas ξ_i (Lanczos, 1988),

$$f(v, t) = \sum_{i=1}^{\infty} w_i \delta[v - \xi_i] \quad (3)$$

which, when substituted on Eq. (2) for the first $2n$ moments, leads to $2n$

equations for w_i and ξ_i :

$$\mu_k = \sum_{i=1}^n \xi_i^k w_i, \quad k = 0, \dots, 2n - 1 \quad (4)$$

Thus, if the PBE is solved for the $2n$ first moments, they can be used to calculate the n weights and n abscissas for the quadrature closure. However, the numerical solution of Eq. (4) is computationally expensive. Fortunately, there exists the Product Difference Algorithm (PDA), proposed by Gordon (1968), that is an efficient approach to evaluate the quadrature in which the moments are used to build a tri-diagonal matrix whose eigenvalue problem solution calculates the weights and abscissas.

Using the same idea of a quadrature closure, Marchisio and Fox (2005) developed the Direct Quadrature Method of Moments (DQMOM). The DQMOM assumes the existence of a Gaussian quadrature, shown in Eq. (3), which is thus substituted into the PBE giving, after convenient integration using the operator $\int_0^\infty v^k \cdot dv$, differential equations for the weights and abscissas. Solving directly for the quadrature rule, the DQMOM avoids the extra load of the PDA, which is necessary for the QMOM. The PDA is necessary only for initialization. During integration, the weights and abscissas evolve as the particle distribution function changes.

Recently, Bove et al. (2005) formulated the Parallel Parent and Daughter Classes (PPDC) method to solve the PBE that also uses the Gaussian quadrature given by Gordon (1968). The PPDC method uses an operator splitting of the explicit Euler time marching method and several grids for the particle property space. Particle death events occurs in the parent particle grid whereas birth events due to breakage and aggregation of particles are represented in an one-grid-per-event-type basis. Thus, the distribution functions for the parent and daughters particles are represented by series of Dirac delta functions, as given by Eq. (3), each one evolving in a different grid. The parent particle distribution function is assumed to be the Gaussian-Gordon quadrature. Thus, there is one grid for the parent particles with n classes, n grids for the daughters formed by breakage, one for each i -class parent particle, $i = 1, \dots, n$, and $n(n - 1)/2$ grids for the daughters generated by aggregation of two parent particles, one for each combination of i and j parent classes, $i, j = 1, \dots, n$. These grids are called parallel because death and corresponding birth events occur simultaneously in them. A characteristic of the PPDC method, which is analyzed in the present work, is the need to define the grids associated to the birth by breakage in terms of number of classes and their pivot positions. The PPDC method regenerates the Gordon-Gaussian quadrature of the overall particle distribution at the end of the time step by summing up the moments of the parent and daughter particle distribution functions and then the PDA.

From the above, it is clear that QMOM, DQMOM and PPDC share one main characteristic: the closure of each method is given by a Gaussian quadrature whose weight is the number density distribution function of particles. When this quadrature rule is needed, all methods use the PDA algorithm of Gordon (1968) that needs the moments of the distribution function. Due to the quadrature rule, all of these methods has a discrete representation of the particle distribution function given by Eq. (3), which make them similar to the MoC. Bove et al. (2005) themselves considered their method as a “non-standard” MoC. In the authors point of view, these three methods can be seen as *hybrid methods*, as they give a discrete representation of the number density distribution function but with the accuracy of a Gaussian quadrature.

1.2 Scope and aim

The convergence of these hybrid methods have been analyzed in the original contributions (McGraw, 1997; Marchisio and Fox, 2005; Bove et al., 2005) but their numerical efficiencies have never been compared quantitatively. The objective of the present work is to quantitatively compare the computational efficiency of the QMOM, DQMOM and PPDC method among themselves and against the classical MoC of Kumar and Ramkrishna (1996a).

2 Comparison framework

The best way to compare numerical methods is against analytical solutions, when the true numerical error can be easily evaluated, because there is no point in comparing numerical methods using their solutions for a given problem with different accuracies.

Although with some limitations, analytical solutions for the PBE considering simultaneous breakage and coalescence are indeed available from the literature (Patil and Andrews, 1998; Lage, 2002; McCoy and Madras, 2003). Patil and Andrews (1998) obtained an analytical solution for a special case where the total number of particles is constant. Latter, Lage (2002) corrected a simplification error in the deduction of Patil and Andrews (1998), obtaining a simpler expression for the analytical solution. McCoy and Madras (2003) derived an analytical solution for a more general case, where the total number of particles is not constant, but using a different initial condition.

The present work uses the PBE problems whose analytical solutions are given by Lage (2002) and McCoy and Madras (2003) to verify the convergence, accuracy and computational efficiency of the MoC, QMOM, DQMOM and PPDC

method. Cases with dominant breakage, dominant aggregation and constant number of particles (invariant) were used. Firstly, the convergence of the numerical moments using different number of classes and quadrature points was evaluated for these cases. Then, the transient behavior of the relative errors of the moments was analyzed. As the PPDC method is based on the explicit Euler method, the other methods were also solved using this time scheme for comparison. Finally, the computational efficiency of the numerical solutions were compared by determining the computational time for solutions with similar accuracy.

2.1 The PBE problems

Patil and Andrews (1998) analyzed the simultaneous breakage and aggregation PBE for a mono-variate distribution function in particle volume, $f(v, t)$, that is given by:

$$\begin{aligned} \frac{df(v, t)}{dt} = & \frac{1}{2} \int_0^v a(v - v', v') f(v - v', t) f(v', t) dv' \\ & - \int_0^\infty a(v, v') f(v, t) f(v', t) dv' \\ & + \int_v^\infty \vartheta(v') b(v') P(v | v') f(v', t) dv' - b(v) f(v, t) \end{aligned} \quad (5)$$

where $a(v, v')$ is the aggregation frequency, $\vartheta(v)$ is the mean number of particles formed by breakage, $b(v)$ is the breakage frequency and $P(v | v')$ is the probability density function of a particle of volume v generated by breakup of a particle of volume v' .

The PBE shown in Eq. (5) is subjected to the following initial conditions:

$$f(v, 0) = \mu_0(0) \left(\frac{\mu_0(0)}{\mu_1(0)} \right) e^{-\frac{\mu_0(0)}{\mu_1(0)} v} \quad (6)$$

or

$$f(v, 0) = \mu_0(0) \left[2 \frac{\mu_0(0)}{\mu_1(0)} \right]^2 v e^{-2 \frac{\mu_0(0)}{\mu_1(0)} v} \quad (7)$$

where $\mu_0(0)$ and $\mu_1(0)$ are the initial zero and first-order moments. Due to mass conservation, μ_1 is constant for the considered problems.

The breakage and coalescence frequencies, the breakage mean number and the probability function adopted by Patil and Andrews (1998) are shown below:

$$\vartheta(v) = 2, \quad b(v) = Sv, \quad S = \text{constant} \quad (8)$$

$$a(v - v', v') = C, \quad C = \text{constant} \quad (9)$$

$$P(v | v') = \frac{1}{v'} \quad (10)$$

Patil and Andrews (1998) introduced the following dimensionless variables

$$\Phi(\tau) = \frac{\mu_0(t)}{\mu_0(0)}, \quad \eta = \frac{v\mu_0(0)}{\mu_1}, \quad \tau = \mu_0(0)Ct, \quad \phi(\eta, \tau) = \frac{f(v, t)\mu_1}{[\mu_0(0)]^2} \quad (11)$$

which turns Eq. (5) into:

$$\begin{aligned} \frac{\partial \phi}{\partial \tau} = & \frac{1}{2} \int_0^\eta \phi(\eta - x, \tau) \phi(x, \tau) dx - \phi(\eta, \tau) \Phi(\tau) \\ & + [\Phi(\infty)]^2 \int_\eta^\infty \phi(x, \tau) dx - \frac{\eta}{2} [\Phi(\infty)]^2 \phi(\eta, \tau) \end{aligned} \quad (12)$$

with initial conditions

$$\phi(\eta, 0) = e^{-\eta} \quad (13)$$

or

$$\phi(\eta, 0) = 4\eta e^{-2\eta}, \quad (14)$$

where

$$\Phi(\infty) = \sqrt{2S\mu_1/C}/\mu_0(0) \quad (15)$$

is the steady-state solution of the dimensionless particle number density.

Thus, when $\Phi(\infty) = 1$, the number density and the total number of particles are constant. The dimensionless formulation presented in Eqs. (12), (13) and (14) were used by Patil and Andrews (1998), Lage (2002) and McCoy and Madras (2003) to obtain their analytical solutions.

If the initial distribution is normalized, $\mu_0(0) = 1$, and with $\mu_1 = 1$, the dimensionless variables are identical to its dimensional form when $C = 1.0$. In

this case, it is possible to relate S directly to $\Phi(\infty)$ by simplifying Eq. (15) to:

$$S = \frac{1}{2}[\Phi(\infty)]^2 \quad (16)$$

Assuming these conditions, the dimensionless and dimensional solutions can be compared directly. Therefore, the following description of the numerical methods was carried out in the dimensional form.

2.2 Analytical solutions

2.2.1 Solution of Lage (2002)

The analytical solutions proposed by Patil and Andrews (1998) and Lage (2002) assumed no variation on the total number of particles, that is, $\Phi(\tau) = \Phi(\infty) = 1, \forall \tau$. They managed to derive analytical solutions for the initial conditions given by Eq. (13) and (14). The former was shown to be a steady-state solution. The latter initial condition gives a transient solution whose simplified form given by Lage (2002) is:

$$\phi^{(a)}(\eta, \tau) = \sum_{i=1}^2 \frac{K_1(\tau) + p_i(\tau)K_2(\tau)}{L_2(\tau) + 4p_i(\tau)} e^{p_i(\tau)\eta}, \forall \tau > 0 \quad (17)$$

where

$$\begin{aligned} K_1(\tau) &= 7 + \tau + e^{-\tau} \\ K_2(\tau) &= 2 - 2e^{-\tau} \\ L_2(\tau) &= 9 + \tau - e^{-\tau} \\ p_{1,2}(\tau) &= \frac{1}{4}(e^{-\tau} - \tau - 9) \pm \sqrt{d(\tau)} \\ d(\tau) &= \tau^2 + (10 - 2e^{-\tau})\tau + 25 - 26e^{-\tau} + e^{-2\tau} \end{aligned} \quad (18)$$

The analytical moments are obtained through their definition, Eq. (2), as:

$$\mu_k^{(a)}(\tau) = \begin{cases} \frac{\Gamma(2+k)}{2^k} & \text{for } \tau = 0 \\ \sum_{i=1}^2 [-p_i(\tau)]^{-k-1} \frac{K_1(\tau) + p_i(\tau)K_2(\tau)}{L_2(\tau) + 4p_i(\tau)} \Gamma(k+1) & \text{for } \tau > 0 \end{cases} \quad (19)$$

2.2.2 Solution of McCoy and Madras (2003)

McCoy and Madras (2003) treated the general case where the total number of particles is not constant, $\Phi(0) \neq \Phi(\infty)$. Thus, $\Phi(\infty)$ can assume arbitrary values, that represents systems with predominant breakage ($\Phi(\infty) > 1$) or aggregation ($\Phi(\infty) < 1$). For the initial condition given by Eq. (14), they could not find a closed-form solution. For the initial condition given by Eq. (13), they managed to derive the following solution:

$$\phi^{(a)}(\eta, \tau) = \Phi^2(\tau)e^{-\eta\Phi(\tau)} \quad (20)$$

where

$$\Phi(\tau) = \Phi(\infty) \left[\frac{1 + \Phi(\infty) \tanh(\Phi(\infty)\tau/2)}{\Phi(\infty) + \tanh(\Phi(\infty)\tau/2)} \right] \quad (21)$$

whose moments, for any finite $\Phi(\infty)$, are given by:

$$\mu_k^{(a)}(\tau) = \left[\frac{\Phi(\infty) + \tanh(\frac{1}{2}\Phi(\infty)\tau)}{\Phi(\infty)(1 + \Phi(\infty) \tanh(\frac{1}{2}\Phi(\infty)\tau))} \right]^{k-1} \Gamma(k+1), \text{ for } \tau \geq 0 \quad (22)$$

3 Numerical techniques

This section describes the MoC, QMOM, DQMOM and PPDC numerical formulations applied to Eq. (5).

3.1 The method of classes

In the MoC of Kumar and Ramkrishna (1996a), only two population properties can be conserved which are usually taken to be the zero and first moments. The zero order sectional moment, N_i , represents the number of particles in class i while the first moment conserves the chosen internal variable, which is the volume, v , in the present case. In fact, N_i includes all the particles with properties between v_i and v_{i+1} ($v_i < \xi_i < v_{i+1}$, ξ_i is the pivot) and is given by

$$N_i = \int_{v_i}^{v_{i+1}} f(v, t) dv, \quad i = 1 \dots n \quad (23)$$

Eq. (5) is integrated in the $[\xi_{i-1}, \xi_{i+1}]$ interval and, after some manipulations (Ramkrishna, 2000), the following system of equations is obtained:

$$\begin{aligned}
\frac{dN_i}{dt} = & \sum_{\substack{k \geq j \\ k, j=1 \\ \xi_{i-1} \leq (\xi_k + \xi_j) \leq \xi_{i+1}}} (1 - 0.5\delta_{kj}) \Psi_{kji} a(\xi_k, \xi_j) N_k N_j - N_i \sum_{k=1}^n N_k a(\xi_k, \xi_i) \\
& + \sum_{k=i}^n \varphi_{i,k} b(\xi_k) N_k - N_i b(\xi_i)
\end{aligned} \tag{24}$$

where Ψ and φ are grid dependent functions of the variables which, in the case of number and volume conservation, are given by:

$$\Psi_{kji} = \begin{cases} \frac{(\xi_j + \xi_k) - \xi_{i-1}}{\xi_i - \xi_{i-1}} & \text{for } \xi_{i-1} \leq (\xi_j + \xi_k) \leq \xi_i \\ \frac{\xi_{i+1} - (\xi_j + \xi_k)}{\xi_{i+1} - \xi_i} & \text{for } \xi_i \leq (\xi_j + \xi_k) \leq \xi_{i+1} \end{cases} \tag{25}$$

$$\varphi_{ik} = \int_{\xi_i}^{\xi_{i+1}} \frac{\xi_{i+1} - v}{\xi_{i+1} - \xi_i} P(v | \xi_k) dv + \int_{\xi_{i-1}}^{\xi_i} \frac{v - \xi_{i-1}}{\xi_i - \xi_{i-1}} P(v | \xi_k) dv \tag{26}$$

In order to provide better evaluations of the cases with dominant breakage or dominant aggregation, geometric grids were used. The grid ends were defined in order to avoid excessive truncation error of the initial distribution function, keeping the relation $\xi_n/\xi_1 = 10^8$. The following geometric rule was used to construct the pivots:

$$\xi_{i+1} = s\xi_i \quad \text{for } i = 0, \dots, n \tag{27}$$

where s is the geometric parameter used to control the number of classes, whose value was chosen between 1.05 and 2. Only the ξ_i , $i = 1, \dots, n$ were active pivots. ξ_0 and ξ_{n+1} were calculated to evaluate the limits of each class by:

$$v_i = \frac{\xi_{i-1} + \xi_i}{2}, \quad i = 1, \dots, n+1 \tag{28}$$

Due to the choice of the conserved properties, it is not possible to conserve the moments μ_k , $k > 1$, but it is still possible to calculate them from the solution:

$$\mu_k = \sum_{i=1}^n \xi_i^k N_i \tag{29}$$

Further details about this method and its formulation can be found in Ramkrishna (2000) and Campos and Lage (2003).

3.2 Quadrature method of moments

The PBE, given by Eq. (5), is operated with $\int_0^\infty v^k \cdot dv$, and the right-hand sides of the resulting equations are approximated by using the Gordon-Gaussian quadrature, giving the following QMOM moment equations:

$$\frac{d\mu_k(t)}{dt} = \bar{H}_k^{(n)}, \quad k = 0, \dots, 2n - 1 \quad (30)$$

where

$$\begin{aligned} \bar{H}_k^{(n)} = & \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n [(\xi_i + \xi_j)^k - \xi_i^k - \xi_j^k] a(\xi_i, \xi_j) w_i w_j \\ & + \sum_{i=1}^n b(\xi_i) w_i [\vartheta(\xi_i) \pi_k(\xi_i) - \xi_i^k] \end{aligned} \quad (31)$$

and

$$\pi_k(\xi_i) = \int_0^{\xi_i} v^k P(v | \xi_i) dv \quad (32)$$

For the present PBE problems, Eq. (10) leads to $\pi_k(\xi_i) = \frac{\xi_i^k}{k+1}$. The PDA implementation used in this work was the one given by Lage (2007) that gains numerical precision by representing real numbers in critical computations in the algorithm by using different variables for their mantissa and exponents.

3.3 Direct quadrature method of moments

The substitution of Eq. (3) into the PBE, Eq. (5), followed by operation with $\int_0^\infty v^k \cdot dv$ results in the differential equations for the weights and weighted-abscissas, $\varsigma_i = w_i \xi_i$, given by:

$$\frac{dw_i}{dt} = \theta_i, \quad i = 1, \dots, n \quad (33)$$

$$\frac{d\varsigma_i}{dt} = \varrho_i, \quad i = 1, \dots, n \quad (34)$$

together with the following system of $2n$ linear equations:

$$(1 - k) \sum_{i=1}^n \xi_i^k \theta_i + k \sum_{i=1}^n \xi_i^{k-1} \varrho_i = \bar{H}_k^{(n)}, \quad k = 0, \dots, 2n - 1 \quad (35)$$

which is solved for the source terms of Eqs. (33) and (34), θ_i and ϱ_i , $i = 1, \dots, n$. It should be noted that the independent vector of the linear system, $\vec{H}_k^{(n)}$, is the same as in QMOM, given by Eq. (31) for simultaneous aggregation and breakage.

In the DQMOM solution, the linear system given by Eq. (35) is solved and coupled to the solution of the differential equations given by Eqs. (33) and (34). From a given distribution function, the PDA is used to generate the initial conditions of w_i and ς_i , $i = 1, \dots, n$.

3.4 Parallel parent and daughter classes method

The time discretization used in PPDC is the explicit Euler method which implies that the source terms in the PBE due to aggregation and breakage processes, referred here as superscript a and b , respectively, are evaluated using the distribution function at the beginning of the time step. Denoting the terms of birth and death of particles by the subscripts B and D , respectively, the time-discretized PBE can be written as:

$$\begin{aligned} \frac{f(v, t^{n+1}) - f(v, t^n)}{\Delta t} = & H_B^a(v, t^n) + H_D^a(v, t^n) \\ & + H_B^b(v, t^n) + H_D^b(v, t^n) \end{aligned} \quad (36)$$

In order to complete the PBE splitting, the distribution function is decomposed into several series of Dirac delta functions for the several parallel grids:

$$\begin{aligned} f_e(v, t) = & \sum_{i=1}^n N_i(t) \delta[v - \xi_i] + \sum_{i=1}^n \sum_{j=1}^n A_{ij}(t) \delta[v - y_{ij}] \\ & + \sum_{i=1}^n \sum_{k=1}^{NB(i)} B_k^{(i)}(t) \delta[v - z_k^{(i)}] \end{aligned} \quad (37)$$

N_i and ξ_i are, respectively, the number densities and pivots for the classes in the parent grid. For each i and j , $i, j = 1, \dots, n$, A_{ij} is the number density at pivot $y_{ij} = \xi_i + \xi_j$, that defines the daughter grid due to aggregation of particles belonging to the i and j parent classes. One daughter grid is defined for the breakage of each parent class i , with $B_k^{(i)}$ being the number densities at the pivots $z_k^{(i)} \in [0, \xi_i)$. $NB(i)$ are the number of classes in such grids.

Mass (or volume) conservation imposes a restriction upon the grids for the daughter particles due to breakage. For binary breakage ($\vartheta = 2$), the death of one parent particle with volume ξ_i results in the birth of two breakage daughter

particles with volumes $\lambda\xi_i$ and $(1-\lambda)\xi_i$ ($0 \leq \lambda \leq 1$). Thus, a symmetric grid in the form

$$z_k^{(i)} = \lambda_k^{(i)} \xi_i, \quad 0 \leq \lambda_k^{(i)} \leq 1 \quad \text{and} \quad \lambda_k^{(i)} = 1 - \lambda_{NB(i)+1-k}^{(i)} \quad (38)$$

suffices for mass conservation. However, it is still necessary to define the value of $NB(i)$. The original work of Bove et al. (2005) used an uniform rule to construct the daughter breakage grids, that is, they set:

$$NB(i) = c, \quad c = \text{constant} \quad (39)$$

Since the daughter classes must be empty at the beginning of the time interval $[t^n, t^{n+1}]$, then

$$A_{ij}(t^n) = B_k^{(i)}(t^n) = 0, \quad \forall i, j, k \quad (40)$$

Combining Eqs. (36) and (37) and using Eq. (40), the numerical densities at t^{n+1} can be obtained by:

$$N_i(t^{n+1}) = N_i(t^n) + \left[-N_i(t^n) \sum_{j=1}^n a(\xi_i, \xi_j) N_j(t^n) - b(\xi_i) N_i(t^n) \right] \Delta t \quad (41)$$

$i = 1, \dots, n$

$$A_{ij}(t^{n+1}) = \left[\left(1 - \frac{1}{2} \delta_{ij} \right) a(\xi_i, \xi_j) N_i(t^n) \right] \Delta t, \quad i, j = 1, \dots, n, j \geq i \quad (42)$$

$$B_k^{(i)}(t^{n+1}) = [\vartheta_i b(\xi_i) N_i(t^n) \Pi(\xi_i)] \Delta t, \quad i = 1, \dots, n, k = 1, \dots, NB(i) \quad (43)$$

where

$$\Pi(\xi_i) = \int_{v_k^{(i)}}^{v_{k+1}^{(i)}} P(v | \xi_i) dv \quad (44)$$

which can be evaluated using Eq. (10) as $\Pi(\xi_i) = \int_{v_k^{(i)}}^{v_{k+1}^{(i)}} \frac{1}{\xi_i} dv = \frac{v_{k+1}^{(i)} - v_k^{(i)}}{\xi_i}$.

From Eq. (37), the moments at t^{n+1} can be calculated by:

$$\mu_k(t^{n+1}) = \sum_{i=1}^n \xi_i^k N_i(t^{n+1}) + \sum_{i=1}^n \sum_{j=1}^n y_{ij}^k A_{ij}(t^{n+1})$$

$$+ \sum_{i=1}^n \sum_{k=1}^{NB(i)} [z_k^{(i)}]^k B_k^{(i)}(t^{n+1}), \quad k = 0, \dots, 2n - 1 \quad (45)$$

which are then used in the PDA to determine N_i and ξ_i at t^{n+1} . Thus, the parent classes are rebuilt from the moments calculated in Eq. (45) at every time step.

4 Numerical Procedure

A PBE solver was programmed in FORTRAN 90 and the source code was compiled with gfortran-4.1 using -O3 high optimization flag and double precision variables. The Γ function used in the analytical solutions and its initial conditions, shown in Eqs. (19) and (22), were calculated with the `gammln` subroutine described in the *Numerical Recipes* book (Press et al., 1990).

The eigenvalues and eigenvectors of the tri-diagonal matrix in PDA were obtained through the `imtq12` subroutine (Smith et al., 1976), downloaded from the NetLib website. The DQMOM linear system, shown in Eq. (35), was solved through Gaussian elimination using the `gausslin` subroutine (Pinto and Lage, 2001). The DASSL routine (Petzold, 1982) was also used for time integration of the systems of differential equations. In all cases simulated using DASSL, the values of absolute and relative tolerances in the code were set to 10^{-10} and 10^{-12} , respectively.

5 Numerical Results

Three PBE cases were selected for the method comparison: (i) the one for each Lage (2002) gave the analytical solution, Eq. (19), in which breakage and aggregation are equally important ($\Phi(\infty) = 1$), (ii) the problem solved by McCoy and Madras (2003), as given by Eq. (22), for dominant breakage ($\Phi(\infty) = 10$) and (iii) this same problem but for dominant aggregation ($\Phi(\infty) = 10^{-1}$). In the following, these cases are referred as cases I, II and III. It should be noted that μ_1 is constant in all cases due to mass conservation. Besides, number conservation occurs in case I, that leads to a constant μ_0 . Only the first 4 to 6 moments of the distribution were considered in comparing the numerical methods. This was considered enough for comparing the lower-order approximations that need to be used in PBE-CFD coupled solution.

The MoC was tested with 3 different grids using $s = 1.4, 1.2$ and 1.1 in Eq. (27) which resulted in geometric meshes with 55, 102 and 194 classes, respectively. Grid endpoints were somewhat different for each case, being ξ_n

equal to 100, 10 and 140 for cases I, II and III, respectively. In addition, the daughter breakage grid in PPDC was discretized using Eq. (39) with $c = 5$.

First, the moments obtained by the numerical solutions with the MoC, QMOM, DQMOM and PPDC method were compared to the analytical solutions in order to evaluate the convergence of the moments using different number of quadrature points or classes. Since a good evaluation of the methods accuracies is essential, several error definitions were used.

The relative error of the k^{th} moment, $\chi_{R,k}$, defined by Eq. (46), provides a good description of the transient numerical accuracy of the methods.

$$\chi_{R,k}(t) = \left| \frac{\mu_k(t) - \mu_k^{(a)}(t)}{\mu_k^{(a)}(t)} \right| \quad (46)$$

As defined in Eq. (47), χ_P is the mean error of the first N moments of the distribution at each time instant.

$$\chi_P(t) = \sqrt{\frac{1}{N} \sum_{k=0}^{N-1} \chi_{R,k}^2(t)} \quad (47)$$

The relative mean error (RME), given by Eq. (48), was used as a representative way to evaluate the mean transient behavior of the first N moments. In Eq. (48), T is the number of solution samples considered during the transient simulation.

$$\chi = \sqrt{\frac{1}{NT} \sum_{l=1}^T \sum_{k=0}^{N-1} \chi_{R,k}^2(t_l)} = \sqrt{\frac{1}{T} \sum_{l=1}^T \chi_P(t_l)} \quad (48)$$

Finally, all the simulations were carried out until all moments reach steady-state values.

5.1 Time step convergence

The first results shown in this work aim to evaluate the time step convergence for the quadrature-based methods. Since PPDC is derived with an explicit Euler discretization in time, an independent time-step solution is required for the proper evaluation and comparison of the methods. Thus, a mean error formulation involving only the first four moments of the distribution ($N = 4$), shown in Eq. (47), was used to evaluate the Euler time step integration error for the QMOM, DQMOM and PPDC simulations. All cases were integrated

using the hybrid methods with the explicit Euler scheme using different time steps and 3 quadrature points.

The QMOM and DQMOM simulations were performed using the DASSL subroutine as well. The latter uses a backward differentiation formula with an adaptive time step to achieve a prescribed tolerance. Since the DASSL relative and absolute tolerances used in this work are very small, DASSL results can be seen as a reference solution for comparing the solutions obtained with the Euler method. As long as PPDC can not be integrated with DASSL, the influence of the Euler time step over the PPDC precision was tested until the time step convergence was achieved.

The mean instantaneous errors for case I simulations with all hybrid methods are shown in Figure 1. The results shown in Fig. 1(a) show different behaviors between the QMOM and DQMOM only for large time steps and specially at the beginning of the simulation, when the DQMOM error is larger. The behaviors of the QMOM and DQMOM errors tend to become similar as the time step is reduced until convergence. Their numerical solution using the Euler method converges to the DASSL results for $\Delta t = 10^{-4}$ s. It should be noted that the DASSL results have no appreciable time integration error, being the simulations results with QMOM and DQMOM almost identical. Therefore, the error seen in Fig. 1(a) for DASSL simulations are solely caused by the quadrature approximation. Therefore, although using the same quadrature approximation, the DQMOM does not solve the problem in terms of the moments, existing a non-linear interaction that increases the DQMOM error for large time steps in the integration using the Euler method. On the other hand, the PPDC method accuracy depends heavily on the time step, as shown in Fig. 1(b). Its solution became independent of the time step for $\Delta t = 10^{-6}$ s. This clearly indicates that the main source of error is the operator splitting. At the steady state is approached, $t \geq 10$ s, the error in the converged PPDC method is about twice larger than the QMOM and DQMOM errors.

Figures 2 and 3 shows the results for the simulations of cases II and III, respectively. Comparing the error magnitudes to those show in Fig. 1, it can be seen that the errors are the largest for case II and the smallest for case III. This shows that the numerical solution of the breakage phenomena is the largest source of error in these methods. The convergence behavior of the results obtained by all methods using Euler integration shows the same trends. The QMOM and DQMOM simulation results obtained by Euler integration for case III also converged to their DASSL simulation results for $\Delta t = 10^{-4}$ s, as in case I, whereas this happened only for $\Delta t = 10^{-5}$ s for case II. The convergence of the PPDC method occurred for $\Delta t = 10^{-5}$ s for case III but only for $\Delta t \lesssim 10^{-7}$ s for case III.

It is not shown here, but the same convergence analysis was evaluated using 4

quadrature points. Basically the same behaviors for the numerical errors and the Δt convergence were obtained, showing that the present results are not appreciably affected by the chosen number of quadrature points.

5.2 Evaluation of the numerical errors

The comparison between the relative errors of the first four moments of the distribution, obtained by Eq. 46, using different number of classes (MoC) and quadrature points (QMOM, DQMOM and PPDC) is shown in this section. In order to minimize the time integration errors, the DASSL subroutine was used to integrate QMOM and DQMOM while the PPDC method was integrated with the explicit Euler scheme using $\Delta t = 10^{-6}$ s. The moments were obtained using 3, 4 and 6 quadrature points for QMOM, DQMOM and PPDC. Since the MoC results are also evaluated in this section, it is possible to compare the accuracy between a fixed pivot method and the hybrid methods with their adaptive pivot positioning. The MoC was employed using geometric grids as described in Section 5.

The comparison of the moment relative errors for the solution of case I is presented in Fig. 4, where it is clear that QMOM and DQMOM errors are much lower than those obtained with PPDC and MoC. A better comparison between QMOM and DQMOM accuracies can be made from Figs. 4(a) and 4(b). Since the number, μ_0 , and the volume, μ_1 , of particles are constant for case I, these lower moments present a very different behavior for the QMOM and DQMOM solutions. While QMOM provides relative errors in the order of machine accuracy ($\sim 10^{-15}$) for these two moments, DQMOM shows relative errors in the order of the tolerances imposed in DASSL integration ($10^{-10} - 10^{-12}$). This comes from the fact that DQMOM solves for the weights and abscissas that are not constant along time even though μ_0 and μ_1 are. On the other hand, the errors of the higher moments, μ_2 and μ_3 , obtained by QMOM and DQMOM are almost identical. In fact, the curves of μ_2 and μ_3 shown in Figs. 4(a) and 4(b) coincide if put together in the same graph. Furthermore, the accuracy of these moments increased as more quadrature points were used.

Differently from the QMOM and DQMOM results, Figure 4(c) shows that the accuracy of the moments obtained using the PPDC method are hardly affected by the number of quadrature points. In contrast, when the number of classes in the MoC was increased, the relative errors of the moments dropped by an almost constant factor, as shown in Fig. 4(d). In MoC, the error of μ_1 is constant as the moment itself and its accuracy is increased with grid refinement. Compared with PPDC method results, MoC was able to achieve lower relative errors for μ_2 and μ_3 when using 102 classes.

The moment relative errors using the PPDC method should decrease as the number of quadrature points was increased. Instead of this expected behavior, the moment errors remain nearly the same independently of n , which indicates a false convergence behavior. In this case, the numerical moments achieve convergence for increasing n but they are different from the analytical solution. This is confirmed in Figure 5, where the analytical moments are compared with those obtained by PPDC with 3 and 6 quadrature points. From this figure, it can be noted that the numerical results of μ_3 achieve the same accuracy independently of n and both quadrature approximations do not follow the analytical results. Thus, the PPDC method was not able to achieve convergence of the moments in case I.

In case II, the number of particles, μ_0 , is not constant due the dominant breakage. The relative errors of the simulations for case II are shown in Fig. 6. Again, QMOM and DQMOM achieved very small errors for the first two moments and the errors for the higher moments, μ_2 and μ_3 , were basically the same for both methods. Thus, for a better readability, Fig. 6(a) only shows the lower moments, μ_0 and μ_1 , obtained by using QMOM and DQMOM. Since the latter methods obtained the same values for the higher moments, Fig. 6(b) shows their results only for DQMOM with $n = 3$ and 6. Finally, Figs. 6(c) and 6(d) present respectively the numerical errors for the PPDC and MoC methods.

As in case I, QMOM also evaluates the constant μ_1 with errors near the machine precision but the μ_0 relative errors are in the order of the DASSL tolerances, varying negligibly with n , as seen in Fig. 6(a). On the other hand, the μ_0 and μ_1 relative errors obtained with DQMOM are two orders of magnitude larger than the DASSL tolerances, indicating that there is error accumulation due to the fast varying distribution in this dominant breakage case. In contrast, the accuracy of μ_2 and μ_3 greatly increases with n , as can be seen in Fig. 6(b).

The moment relative errors obtained by the PPDC method, shown in Fig. 6(c), are almost independent on the number of quadrature points. There is an increase in accuracy by increasing n only for μ_2 . In addition, μ_0 and μ_1 relative errors are accumulating in time, indicating that the chosen time step is still somewhat large. The relative errors for the higher moments have similar values for the QMOM and DQMOM errors for $n = 3$ but are much larger for $n = 6$. This lack of proper convergence indicates that, as in case I, the PPDC method leads to a false convergence behavior. Figure 6(d) shows that MoC results show a better performance than PPDC in this case. As in case I, moment accuracy in MoC is improved by a constant factor as the grid is refined.

Fig. 7 presents the relative errors of the moments for case III using QMOM, DQMOM, PPDC and MoC. As observed in Fig. 7(a) and (b), the behaviors of

the errors of the lower moments, μ_0 and μ_1 , obtained by QMOM and DQMOM are similar to those of case II. The DQMOM errors for μ_0 and μ_1 and the QMOM error for μ_0 stay around 10^{-8} , being controlled by DASSL tolerances, whereas the QMOM error for μ_1 is near to machine precision. Both QMOM and DQMOM obtained the same accuracy for the higher moments although some differences could be noticed for $n = 6$.

From Fig. 7(c), the relative moment errors obtained by the PPDC method also show lack of proper convergence as n is increased. Besides, error accumulation is present for all moments. Figure 7(d) shows that MoC gives the worst solution among all analyzed methods for case III, showing its difficulty in representing fast aggregation processes.

The analysis of the RME, defined in Eq. (48), can provide an overall perspective regarding the convergence rate of the methods. Figures 8, 9 and 10 presents the RME obtained for all methods using the first 6 moments and $T = 1000$ time samples for cases I, II and III, respectively. For the hybrid methods, the number of quadrature points was chosen to be 3, 4 and 6, and for the MoC, three geometrical grids were chosen that basically double the number of pivots between successive grids.

As noticed in Figs. 8-10, the global behavior of QMOM and DQMOM are very similar. They basically reached the same RME values for all cases. Thus, the small differences in their accuracies are not relevant to the overall behavior. Moreover, QMOM/DQMOM presented the largest convergence rate for all cases, reaching very low values for the RME. For all cases, PPDC have the smallest convergence rate, even when compared to the MoC. In fact, MoC using the most refined grid was able to achieve better RME values than PPDC in all cases. The accuracy of MoC was almost the same for all simulated cases since the grid was properly adjusted for each case.

5.3 Computational Time

An evaluation of the computational time of QMOM, DQMOM and PPDC is shown in this section. The gfortran 4.0 intrinsic subroutine CPU_TIME was used to return the elapsed CPU time in seconds, accurate to the millisecond. The tests were realized using the explicit Euler method with $\Delta t = 10^{-5}$ s which was chosen in order to provide a good balance between accuracy and the proper comparison between all methods. It should be noted that in PBE-CFD coupled simulations of polydisperse multiphase flows, low order methods are usually employed what makes our choice of integration method quite adequate.

The computational effort of each hybrid method was evaluated for $n = 2$ to 10. The tests were evaluated for case I with 5 s of simulation time and using

an Intel Core 2 Duo 6600 2.4 MHz processor with Linux operational system. The computational time of the methods is shown in Fig. 11.

Indeed, DQMOM is much faster than the other methods in most of the simulations, because the extra effort of the PDA increases the computational time of the PPDC method and QMOM. Figure 11 shows that the computation efforts of the QMOM and PPDC method were very similar for $n \leq 6$. For higher values of n , the increase of the computational effort was not so severe for PPDC as for the other methods. As a matter of fact, PPDC presented the best computational time for $n \geq 9$. However, as seen in the previous section, the numerical solution of PPDC is extremely dependent on the time step and lacks proper convergence with the number of quadrature points. Therefore, its usage is not recommended.

Both QMOM and DQMOM can provide an accurate solution using higher values of time steps which results in faster simulations. Moreover, QMOM and DQMOM can use other time integration approaches than the explicit Euler method, such as a BDF method. The computational time of QMOM and DQMOM for case I using the DASSL routine are shown in Fig. 12. As expected, the simulations integrated with DASSL are much faster than those using the explicit Euler method, reported in Fig. 11. However, when using $n = 10$ the DASSL routine presented an error during the integration of DQMOM whereas the QMOM solution was interrupted due to loss of precision in the PDA. As noticed in Fig. 12, the DASSL routine had difficulties to integrate the DQMOM solution using $n = 9$, which led to a substantial increase in the computational time.

The MoC was not considered in the previous tests since its computational effort to solve the PBE is huge when compared to the other methods. For instance, MoC solution with 102 classes for case I takes more than 3 seconds using the DASSL routine and nearly 490 seconds using the explicit Euler with $\Delta t = 10^{-5}$ s.

6 Conclusions

All three quadrature-based methods available in the literature, QMOM, DQMOM and PPDC, and the method of classes, MoC, were successfully compared for population balance problems with simultaneous breakage and aggregation. Three cases were setup with equivalent breakage and aggregation, dominant breakage and dominant aggregation, referred as cases I, II and III, respectively. The numerical results were compared against the Lage (2002) and McCoy and Madras (2003) analytical solutions. The QMOM, DQMOM and MoC were integrated in time using the backward difference formula (BDF) implemented in

the DASSL routine (Petzold, 1982) and also using the explicit Euler method, needed for the PPDC method.

The convergence of all quadrature-based methods regarding the time step used in time integration was evaluated. Both QMOM and DQMOM could easily achieved converged solutions. On the other hand, very small time steps were necessary to achieve an time-step independent solution with the PPDC method, indicating that the error is mainly due to the existing operator splitting.

All methods were analyzed regarding their convergence regarding the number of particle classes, n . Although QMOM and DQMOM presented some differences in the convergence behaviors of the distribution moments, their overall behavior is basically the same, showing the largest convergence rate with n . However, the PPDC method showed poor convergence, showing only minor improvements in the solutions when n was increased. The convergence rate is so small that a false convergence can easily be inferred if an analytical solution is not available. The MoC was able to achieve a better accuracy than the PPDC method with a good convergence rate.

Finally, the computational effort of each quadrature-based method was evaluated for different n values, using the explicit Euler method for the time integration. The DQMOM was shown to be the most efficient for $n \leq 8$ and the PPDC presented the best efficiency for $n > 8$. However, the PPDC solution for the same n value is much less accurate than the DQMOM solution. In addition, the computational effort of both QMOM and DQMOM was evaluated using the DASSL routine for integration, and DQMOM was also the most efficient method.

From the above comments, it can be conclude that the simulations using the PPDC method strongly suffer from numerical errors. The error in the operator splitting requires a very small timestep in the explicit Euler integration method which leads to accumulation of truncation error. This makes the present version of PPDC inadequate. It can also be concluded that the DQMOM is the more efficient method for lower order moment approximations and it is recommended for coupling population balance solutions to CFD simulations.

Acknowledgments

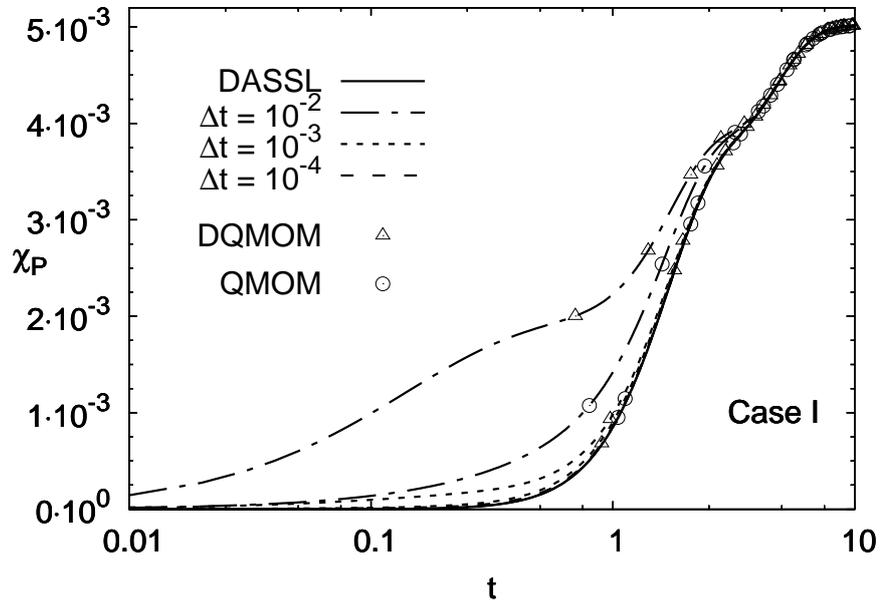
The authors would like to thank CNPq (grant no. 301548/2005-6) and FAPERJ for partially funding this research. L.F.L.R. Silva would also like to acknowledge the financial support given by Chemtech.

References

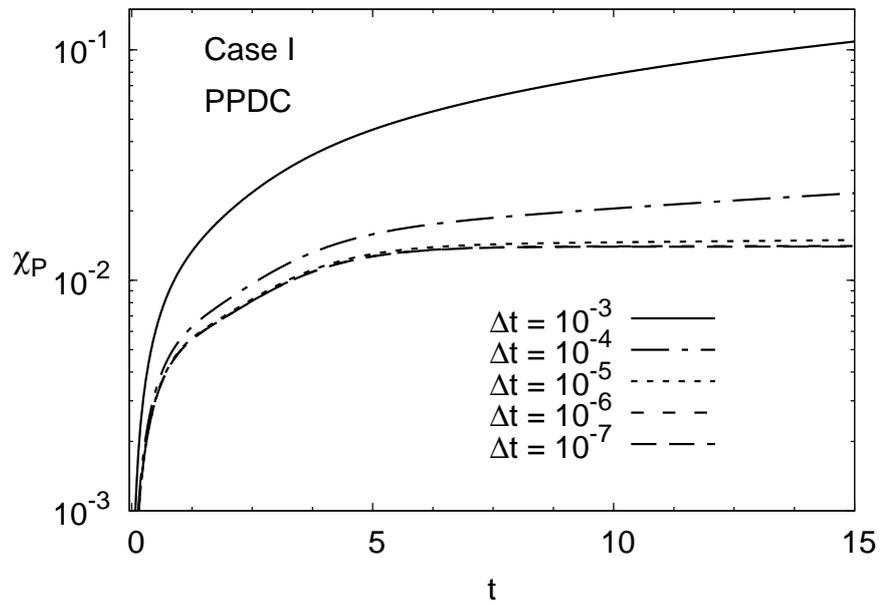
- Bove, S., Solberg, T., Hjertager, B. H., 2005. A novel algorithm for solving population balance equations: the parallel parent and daughter classes. Derivation, analysis and testing. *Chemical Engineering Science* 60, 1449.
- Campos, F., Lage, P., 2003. A numerical method for solving the transient multidimensional population balance equation using an euler-lagrange formulation. *Chemical Engineering Science* 58, 2725.
- Friedlander, S., 1977. *Smoke, Dust and Haze*. John Wiley & Sons Inc.
- Gelbard, F., Seinfeld, J., 1978. Numerical solution of the dynamic equation for particulate systems. *Journal of Computational Physics* 28, 357.
- Gordon, R., 1968. Error bounds in equilibrium statistical mechanics. *Journal of Mathematical Physics* 9 (5), 655.
- Hounslow, M., Ryall, R., Marshall, V., 1988. A discrete population balance for nucleation, growth and aggregation. *A.I.Ch.E. Journal* 34 (11), 1821.
- Hulburt, H., Katz, S., 1964. Some problems in particle technology. A statistical mechanical formulation. *Chemical Engineering Science* 19, 555.
- Kumar, S., Ramkrishna, D., 1996a. On the solution of population balance equations by discretization - I. A fixed pivot technique. *Chemical Engineering Science* 51 (8), 1311.
- Kumar, S., Ramkrishna, D., 1996b. On the solution of population balance equations by discretization - II. A moving pivot technique. *Chemical Engineering Science* 51 (8), 1333.
- Lage, P., 2002. Comments on the “An analytical solution to the population balance equation with coalescence and breakage – the special case with constant number of particles” by D.P. Patil and J.R.G. Andrews [*Chemical Engineering Science* 53(3) 599–601]. *Chemical Engineering Science* 57, 4253.
- Lage, P., 2007. The quadrature method of moments for continuous thermodynamics. *Computers & Chemical Engineering* 31 (7), 782–799.
- Lanczos, C., 1988. *Applied analysis*. Dover.
- Marchisio, D., Fox, R., 2005. Solution of population balance equations using the direct method of moments. *Journal of Aerosol Science* 36, 43.
- McCoy, B., Madras, G., 2003. Analytical solution for a population balance equation with aggregation and fragmentation. *Chemical Engineering Science* 58, 3049.
- McGraw, R., 1997. Description of aerosol dynamics by the quadrature method of moments. *Aerosol Science and Technology* 27, 255.
- McGraw, R., Saunders, J., 1984. A condensation feedback mechanism for oscillatory nucleation and growth. *Aerosol Science and Technology* 3, 367.
- Patil, D., Andrews, J., 1998. An analytical solution to the population balance equation with coalescence and breakage - the special case with constant number of particles. *Chemical Engineering Science* 53 (3), 599.
- Petzold, L., 1982. A description of DASSL: A differential/algebraic system solver. Tech. rep., Sandia National Laboratories.
- Pinto, J., Lage, P., 2001. *Métodos Numéricos em Problemas de Engenharia*

- Química. E-papers.
- Press, W., B.P., F., Teukolsky, S., Vetterling, W., 1990. Numerical Recipes — The Art of Scientific Computing (FORTRAN version). Cambridge University Press.
- Ramkrishna, D., 2000. Population Balances — Theory and Applications to Particulate Systems in Engineering. Academic Press, San Diego.
- Ramkrishna, D., Borwanker, J., 1973. A puristic analysis of population balance - I. Chemical Engineering Science 28, 1423.
- Ramkrishna, D., Borwanker, J., 1974. A puristic analysis of population balance - II. Chemical Engineering Science 29, 1711.
- Smith, B. T., Boyle, J. M., Dongarra, J. J., Garbow, B. S., Ikebe, Y., Klema, V. C., Moler, C. B., 1976. Matrix Eigen-system Routines – EISPACK Guide. Springer-Verlag.
- Vanni, M., 1999. Discretization procedure for the breakage equation. A.I.Ch.E. Journal 45 (4), 916.

Figure Captions

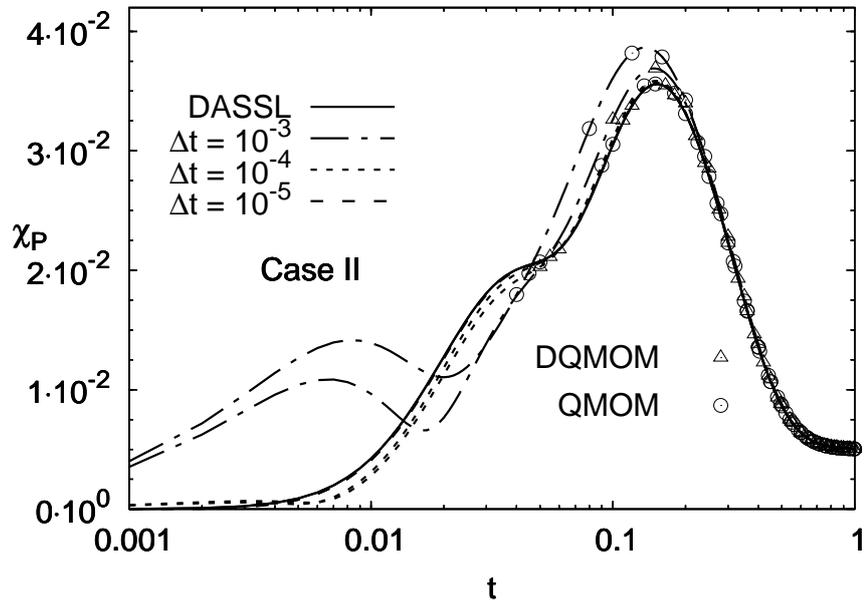


(a)

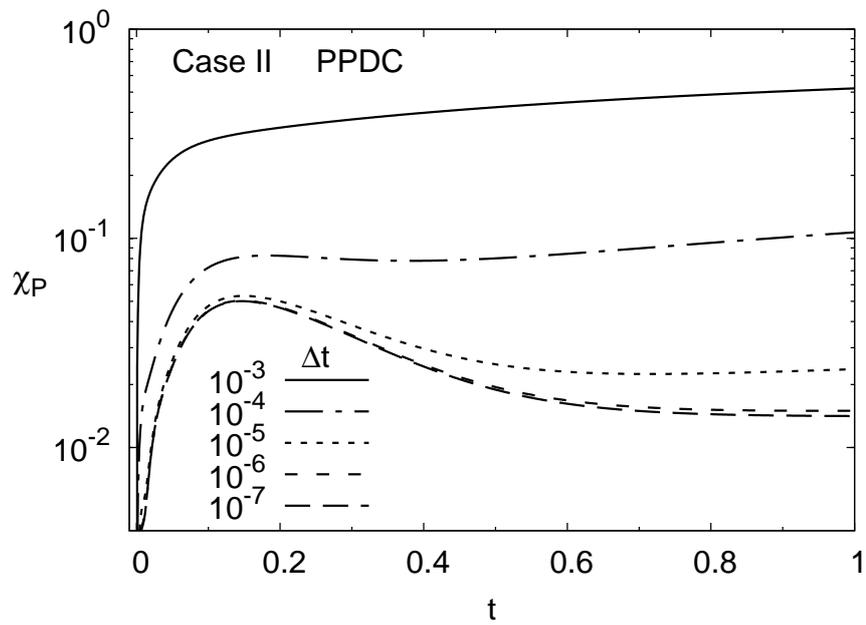


(b)

Fig. 1. Moment mean errors for case I using (a) QMOM, DQMOM (b) and PPDC.

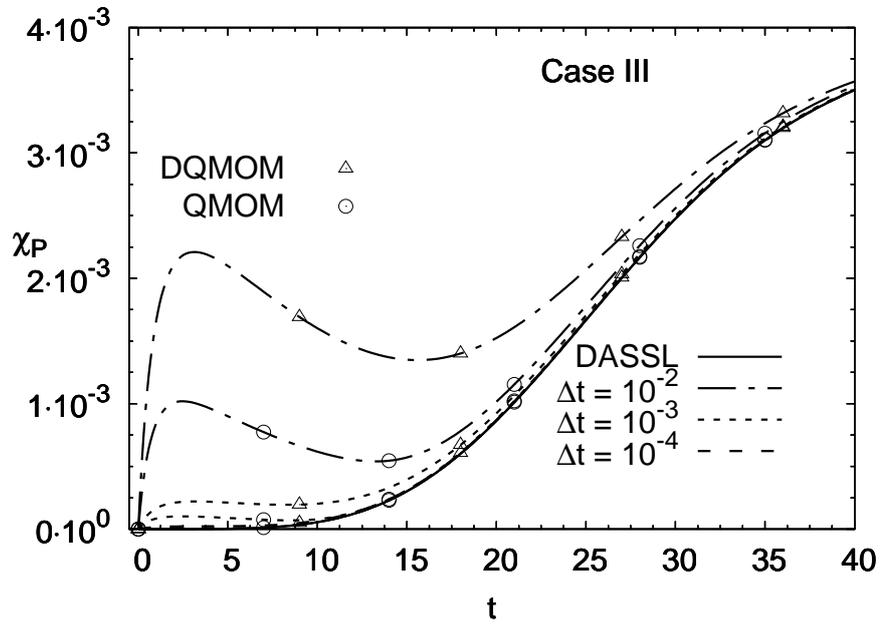


(a)

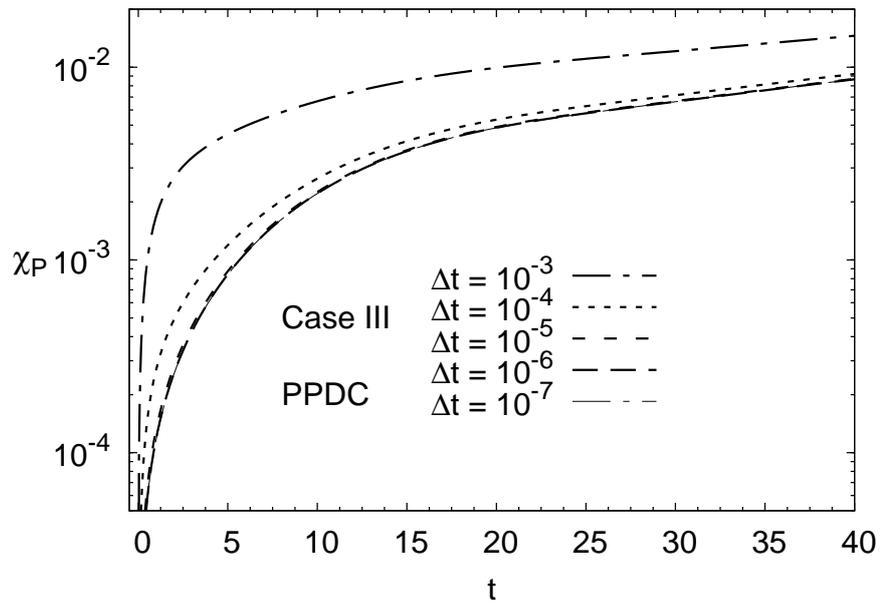


(b)

Fig. 2. Moment mean errors for case II using (a) QMOM, DQMOM and (b) PPDC.



(a)



(b)

Fig. 3. Moment mean errors for case III using (a) QMOM, DQMOM and (b) PPDC.

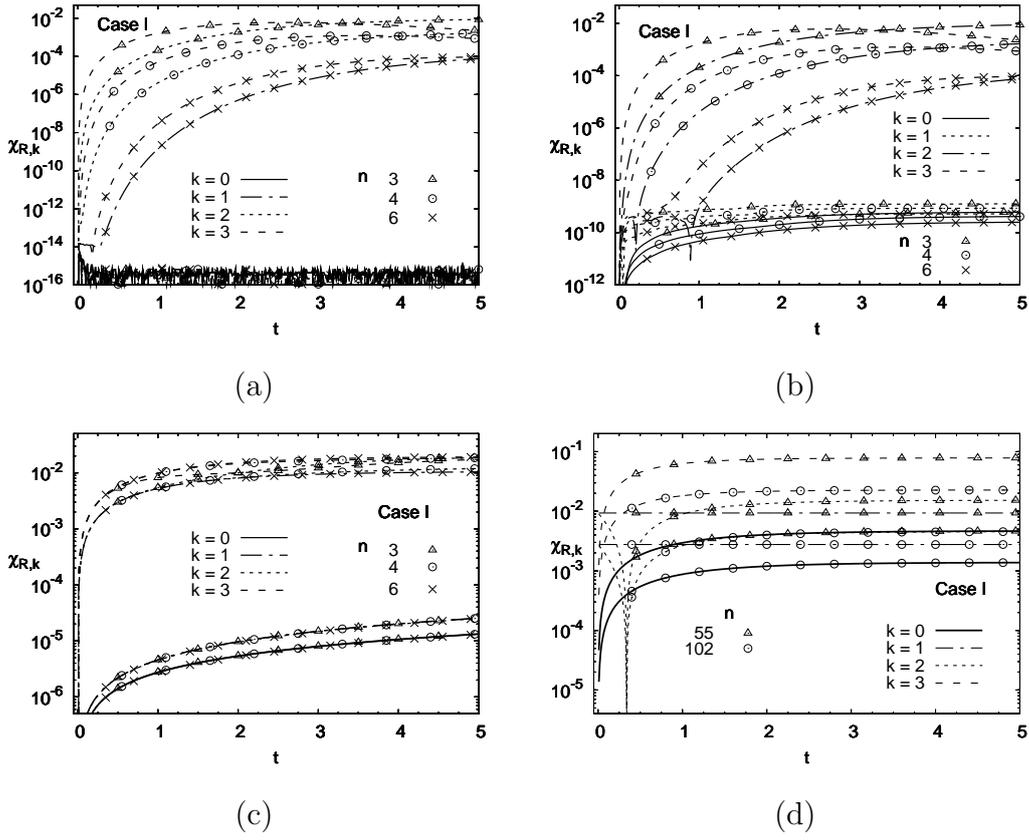


Fig. 4. Relative errors of the moments for case I using (a) QMOM, (b) DQMOM, (c) PPDC and (d) MoC.

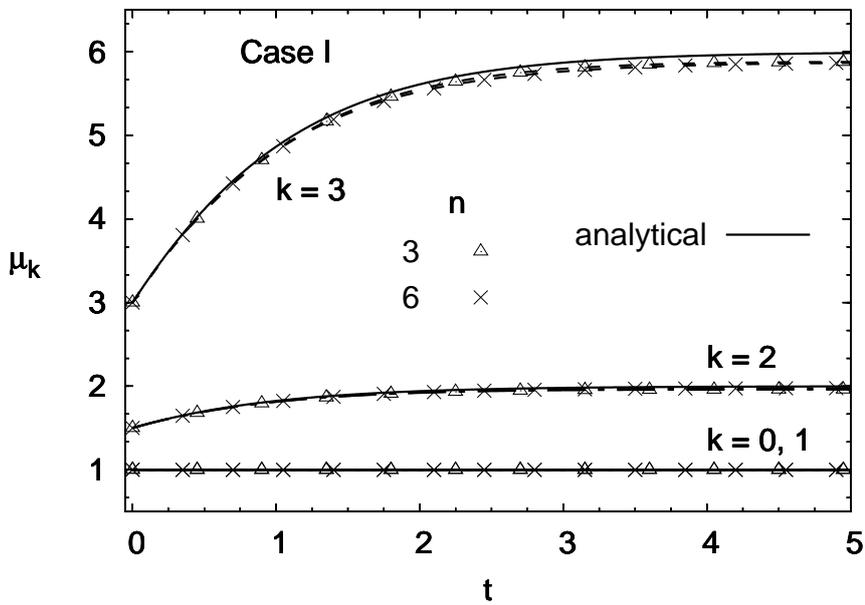
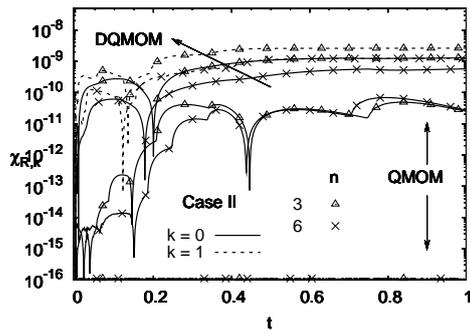
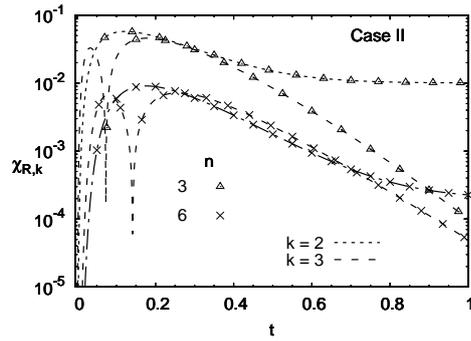


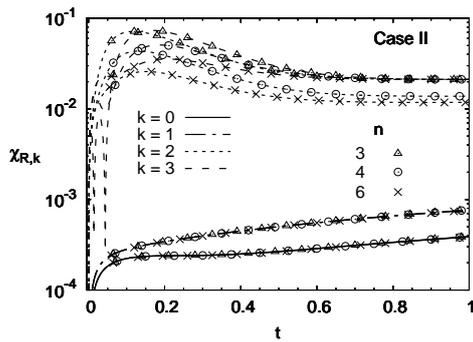
Fig. 5. Comparison between the analytical and numerical moments using PPDC with 3 and 4 quadrature points.



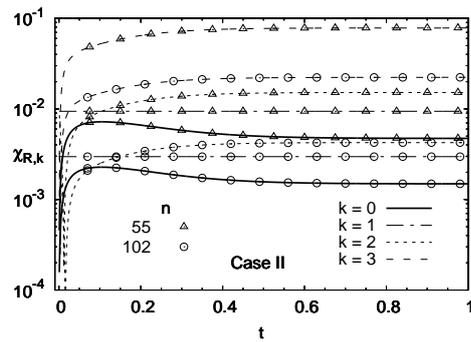
(a)



(b)



(c)



(d)

Fig. 6. Relative errors of moments for case II using QMOM (a), DQMOM (b), PPDC (c) and MoC (d).

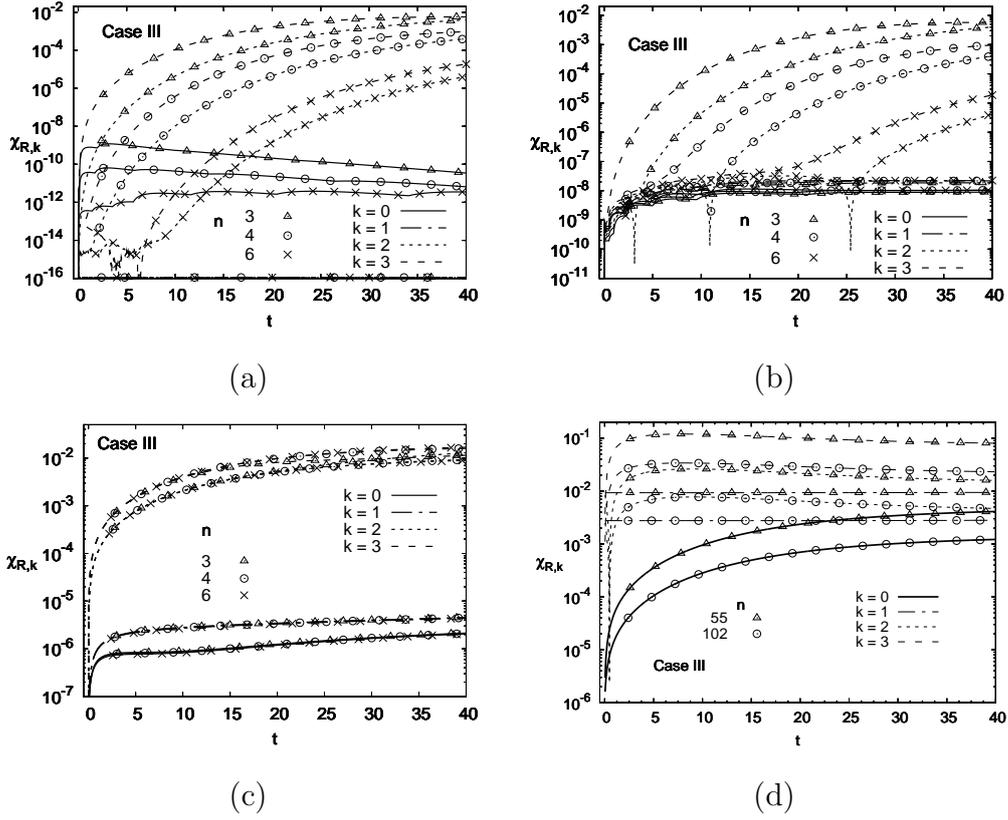


Fig. 7. Relative errors of moments for case III using QMOM (a), DQMOM (b), PPDC (c) and MoC (d).

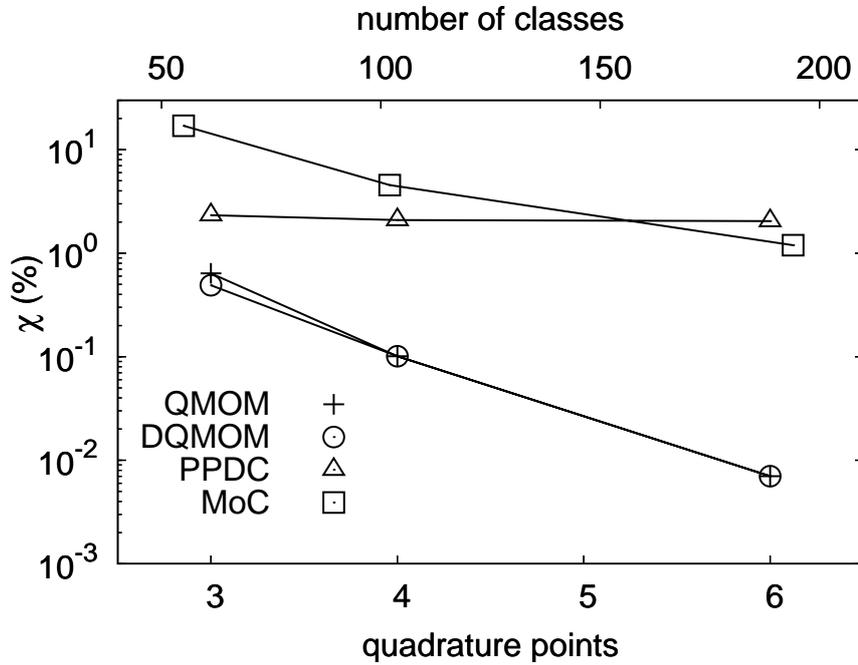


Fig. 8. Results for the RME (%) using QMOM, DQMOM, PPDC and MoC simulated for case I using different number of classes/quadrature points.

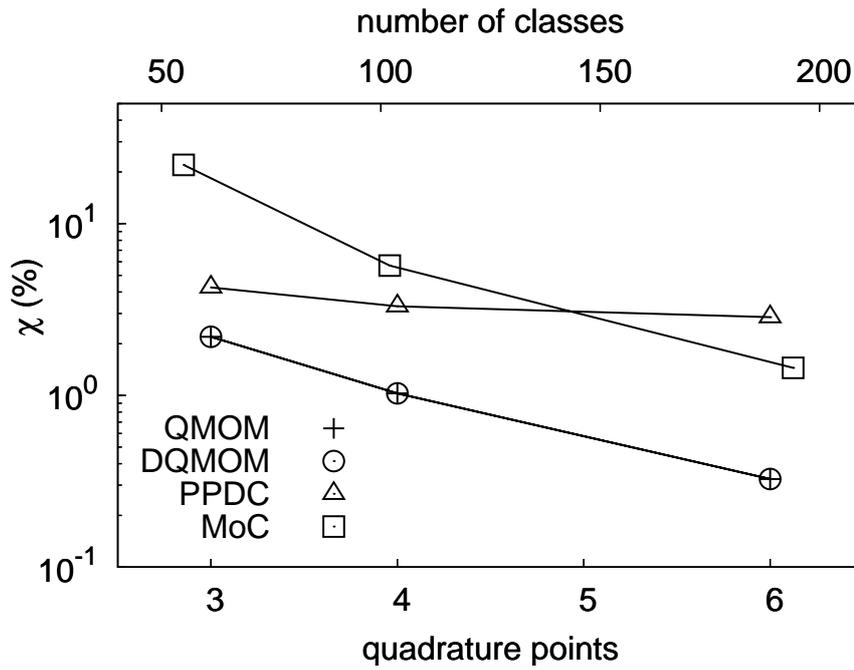


Fig. 9. Results for the RME (%) in case II using QMOM, DQMOM, PPDC and MoC simulated using different number of classes/quadrature points.

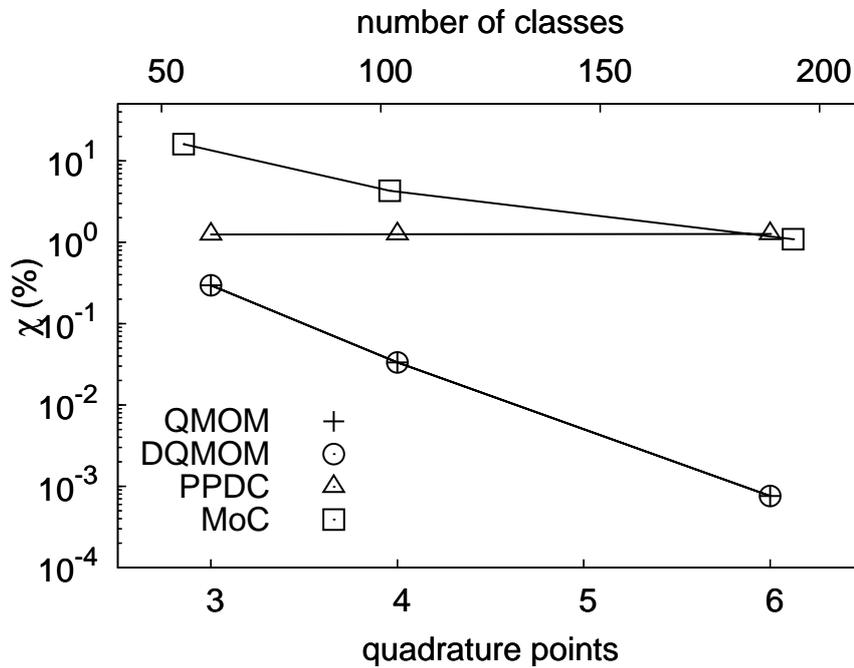


Fig. 10. RME results (%) for case III using QMOM, DQMOM, PPDC and MoC simulated using different number of classes/quadrature points.

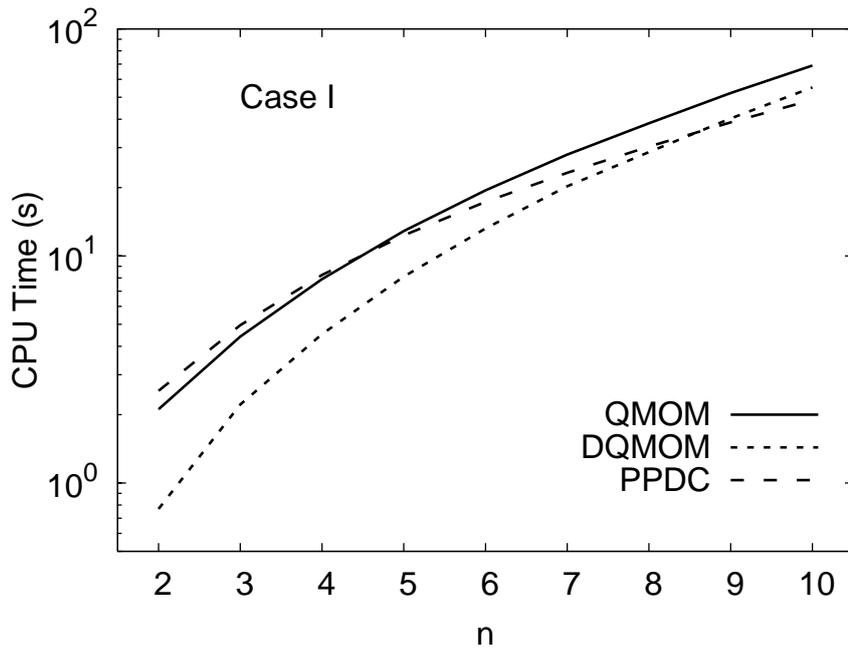


Fig. 11. Computational time of the simulations using QMOM, DQMOM and PPDC for case I using the explicit Euler method with $\Delta t = 10^{-5}$ s.

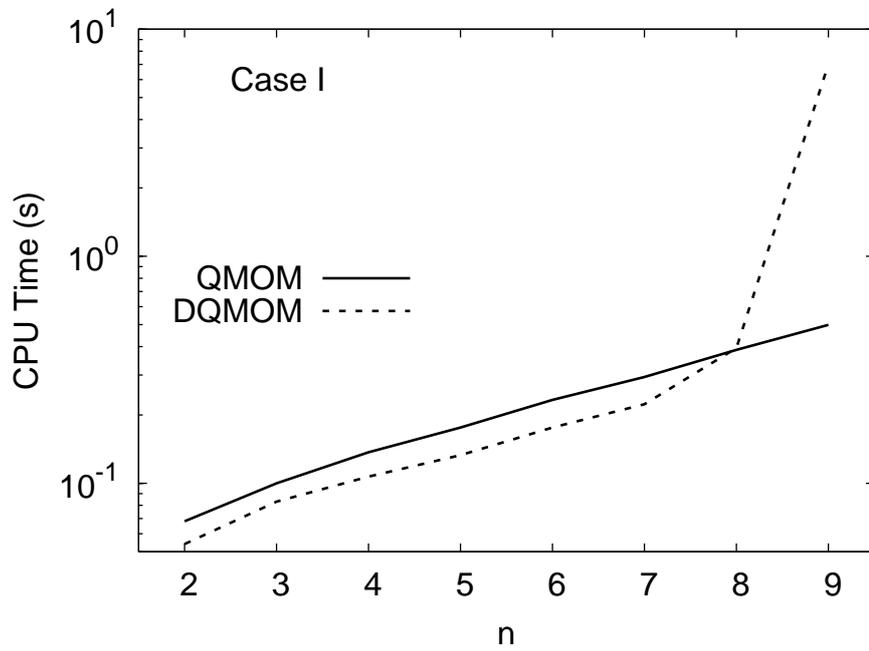


Fig. 12. Computational time of QMOM and DQMOM simulated with the DASSL routine.

Apêndice B

Avaliação e Implementação da EBP e MUSIG

B.1 Informações sobre o Apêndice

O trabalho apresentado neste anexo foi apresentado na *International Conference of Multiphase Flow 2007* [281], que ocorreu em Leipzig, Alemanha. Uma versão aprimorada deste trabalho foi aceita recentemente para publicação [260].

Implementation and analysis of numerical solution of the population balance equation in CFD packages

Silva L.F.L.R., Damian R.B. and Lage P.L.C.

Programa de Engenharia Química, COPPE, Universidade Federal do Rio de Janeiro

P. O. Box 68502, 21941-972, Rio de Janeiro, RJ, Brazil

paulo@peq.coppe.ufrj.br

Keywords: Population balance, DQMOM, ANSYS CFX, OpenFOAM

Abstract

Simulation of polydisperse flows must include the effects of particle-particle interaction, as breakage and aggregation, coupling the population balance equation (PBE) with the multiphase modelling. In fact, the implementation of efficient and accurate new numerical techniques to solve the PBE is necessary. The Direct Quadrature Method of Moments (Marchisio & Fox 2005), known as DQMOM, came into view as a promising choice for this implementation. DQMOM is a further development from QMOM (McGraw 1997), a moment-based method that uses an optimal adaptive quadrature closure. In the present work, DQMOM was implemented in two CFD packages: the commercial ANSYS CFX code, through FORTRAN subroutines, and the open-source OpenFOAM code, by directly coding the PBE solution. Transient zero-dimensional and steady one-dimensional simulations were performed in order to explore the PBE solution accuracy using several interpolation schemes. Simulation cases with dominant breakage, dominant aggregation and invariant solution (equivalent breakage and aggregation) were simulated and validated against an analytical solution (McCoy & Madras 2003). The solution of the population balance equation was then coupled to the two-fluid model, considering that all particles classes share the same velocity field. Momentum exchange terms were evaluated using the local instantaneous Sauter mean diameter of the size distribution function. The two-dimensional tests were performed in a backward facing step geometry where the vortex zones traps the particles and provides high rates of breakage and aggregation.

Introduction

The Computational Fluid Dynamic (CFD) simulations have been used with great success in chemical engineering applications. Nowadays, CFD analysis allows the reproduction of local details in three-dimensional domains in equipments and the treatment of processes with complex physics as multiphase, turbulent and reactional flows. However, the Direct Numeric Simulation (DNS) of these flows in industrial scale are still intractable. Therefore, statistical averaging modelling approaches are necessary to reduce the computational effort.

One of these modelling approaches for multiphase flows is the multifluid model, also called the Eulerian multiphase flow model, in which conservation equations are derived for each relevant mean variable field of each phase (momentum, energy, turbulence energy, mass concentration, etc.) (Ishii 1975; Drew & Prassman 1999). In the statistical averaging process, interphase exchange terms are produced, which need to be modeled considering a characteristic length scale of the interface. Dispersed multiphase flows are those in which one phase is a connect domain and all the others phases are dispersed in the form of small elements, the particles, used here for drops, bubbles or solid particles. In these cases, particle diameter characterises the interphase length scale and it is

commonly used in modelling all interphase exchange terms which give the interactions between the particles and the continuous phase. However, they do not consider the particle-particle interactions.

In dispersed flows, the particles interact among themselves. They collide and may aggregate. They can be broken due to collisions or due to interaction with the continuous phase flow. Therefore, the occurrence of particles with different sizes are more a rule than an exception in a dispersed flow. In actual applications, the particles in a dispersed multiphase flow are different due to several properties: size, shape, composition, energy content, etc. When the particles cannot be assumed to be equal, the flow is called a polydisperse multiphase flow.

There are two challenges in modelling dispersed flows: the exchange of momenta between particles due to collisions, which is important for dense solid particle flows (granular flows), and the changes of all other particle properties due to particle breakage and aggregation, which is paramount in fluid particles flows. The former problem is analysed by modelling each collision in the discrete element method (Bertrand *et al.* 2005) or by including their statistical averaged influence in the multifluid model as in the theory of granular flows (Gidaspow 1994). The latter problem is analysed by population balance (PB) (Ramkrishna 2000), which

includes the statistically averaged effect of breakage and aggregation on the particle population.

Population balance has shown to be a powerful tool to describe the evolution of particle distributions in different pieces of equipment for decades. Besides the modelling of the phenomena which change the number of particles (breakage, aggregation and nucleation), it can also account for any variation of a particle property due to its interaction to the continuous phase. Recently, much research has been carried out on the coupling of the solution of the population balance equation (PBE) to CFD software. The main limitation of this coupling is the extra computational effort for PBE solution. Classical numerical techniques to solve the PBE, as Monte Carlo or sectional methods, are very computationally expensive. Therefore, efficient and accurate new numerical methods are necessary.

The Direct Quadrature Method of Moments (DQMOM) (Marchisio & Fox 2005) came into view as a promising choice for the PBE-CFD coupling. DQMOM was inspired by QMOM (McGraw 1997), a moment-based method that uses an optimal adaptive quadrature closure. DQMOM avoids the moment balance equations of QMOM. The main advantage of QMOM/DQMOM is that few abscissas are necessary to accurately describe a particle distribution due to the optimal quadrature closure. Since this leads to moving pivots, when the DQMOM solution of the PBE is coupled to CFD simulations, each point in the physical domain has its own quadrature abscissas and weights to describe the particle distribution. The accuracy gain thus obtained enables that DQMOM can be used in PBE-CFD simulations without increasing too much the computational effort as previous techniques do. Marchisio & Fox (2005) presented the DQMOM technique, showing its efficiency against previous techniques. DQMOM was used to the PBE-CFD simulations of a polydisperse gas-solid fluidised bed using the FLUENT package (Fan *et al.* 2004) and was recently applied to simulate flows with bivariate particle distributions (Fox 2006).

Besides the PBE-CFD coupling, another great challenge on the simulation of polydisperse flows are the breakage and aggregation modelling. Araújo (2006) showed that the current available breakup and coalescence models for gas-liquid flows are not completely adequate. For instance, a simulation of a bubble column flow must consider several difficult modelling issues as turbulence in two-phase flow, momentum interphase exchange forces (drag, lift, virtual mass, wall lubrication and turbulent dispersion forces) and the PBE solution with adequate breakage and coalescence models. Note that experimental validations of this kind of flow are very difficult due to the fact that too many modelling uncertainties are present.

The present work is focused on the investigation of the numerical behaviour of the coupled DQMOM solution in CFD codes. For the present time, no attention will be given to realistic breakage and aggregation models and neither to the adequacy of the modelling of the momentum interphase exchange forces. Rigorous numerical verification steps were followed to learn how this coupling behaves in order to guarantee numerical accuracy of the PBE solution in CFD codes before moving ahead to the other modelling stages.

The numerical verification were performed using the analytical solution of McCoy & Madras (2003), which solves the monovariate PBE for a transient zero-dimensional problem considering simple breakup and aggregation models. Cases with dominant breakage, dominant aggregation and the invariant solution (equivalent breakage and aggregation) were considered. First, transient zero-dimensional simulations were performed in order to check the time integration accuracy of the CFD codes. Next, the time dependent zero-dimensional solution was transformed to a steady one-dimensional solution considering a uniform velocity field. This allowed the analysis of several advection schemes and mesh convergence. Finally, in order to verify the multi-dimensional behaviour of the implementation, a fictitious two-dimensional emulsion flow through a Backward Facing Step (BFS) geometry was simulated using the same simple models for breakage and aggregation.

Nomenclature

A_α	cross section area divided by the volume of the particle (m^{-1})
$a(v, v', t)$	aggregation frequency ($m^{-3} s^{-1}$)
$b(v, t)$	breakage frequency (s^{-1})
C_D	coefficient of interphase drag force
\mathbf{D}_α	deformation tensor of phase α ($kg m^{-1} s^{-2}$)
d_s	Sauter mean diameter (m)
$f(\mathbf{x}, v, t)$	particle distribution function (m^{-6})
\mathbf{g}	gravitational field ($m s^{-2}$)
$H(\mathbf{x}, v, t)$	PBE source term ($m^{-6} s^{-1}$)
k	turbulent kinetic energy ($m s^{-2}$)
\mathbf{M}_α	interphase force acting on phase α ($kg m^{-2} s^{-2}$)
N	number of phases
n	number of quadrature points
$P(v v')$	conditional probability for generating a daughter particle of volume v once a particle of volume v' has broken (m^{-3})
p	pressure ($kg m^{-1} s^{-2}$)
r_α	fraction of phase α
\mathbf{T}_α	tension tensor of phase α ($kg m^{-1} s^{-2}$)
t	time (s)
\mathbf{u}_α	velocity field of phase α ($m s^{-1}$)
v	volume of particle (m^3)
\mathbf{x}	external variables, spatial coordinates (m)
w_i	quadrature weight (m^{-3})
<i>Greek letters</i>	
χ	mean relative error
Γ	Gamma function
μ_k	moment k of distribution
ς_i	weighted abscissa ($\varsigma_i = \xi_i w_i$)
ν_α	dynamic viscosity of phase α ($kg m^{-1} s^{-1}$)
η_α	kinematic viscosity of phase α ($m^2 s^{-1}$)
ρ_α	density of phase α ($kg m^{-3}$)
τ_α	viscous tension tensor of phase α ($kg m^{-1} s^{-2}$)
θ_i, κ_i	DQMOM source terms
$\vartheta(v)$	mean number of particles formed due to the

ξ_i breakage of a particle with volume v
quadrature abscissa (m^3)

Subscripts

c continuous phase
 d dispersed phase
 r relative

Superscripts

(a) analytical solution
 D drag
 eff effective
 f dimensional variable
 ϕ dimensionless variable
 lam laminar
 $turb$ turbulent

Two-fluid Model

The two-fluid model is based on the mean mass and *momentum* conservation equations to describe the dynamic behaviour of the two-phase flow. These equations are obtained through average procedures, which usually are the time average or the ensemble average. The average procedure introduces the average occurrence of phase α , r_α , which is known as the phase fraction. Due the average procedure, additional terms appears in the mean conservation equations which need to be modeled. These terms represent the phenomena that occurs on scales smaller than the averaging scale. Details about the theory and formulation of the two-fluid model can be found in Ishii (1975), Drew & Prassman (1999) and Bove (2005).

Assuming that the interfacial tension effects are not important, it is possible to consider that all phase share the same pressure field. In this case, the average two-fluid equations are given by:

$$\frac{\partial(r_\alpha \rho_\alpha)}{\partial t} + \nabla \cdot (r_\alpha \rho_\alpha \mathbf{u}_\alpha) = 0 \quad (1)$$

$$\frac{\partial(r_\alpha \rho_\alpha \mathbf{u}_\alpha)}{\partial t} + \nabla \cdot (r_\alpha \rho_\alpha \mathbf{u}_\alpha \mathbf{u}_\alpha) = \nabla \cdot (r_\alpha \mathbf{T}_\alpha^{eff}) + \mathbf{M}_\alpha + r_\alpha \rho_\alpha \mathbf{g} \quad (2)$$

where \mathbf{T}_α^{eff} represents the effective tensor composed by the mean viscous and turbulent tensors of phase α , which is usually modelled using the Newtonian functional form:

$$\mathbf{T}_\alpha^{eff} = -p_\alpha \mathbf{I} + \tau_\alpha^{eff} \quad (3)$$

$$\tau_\alpha^{eff} = \nu_\alpha^{eff} \left[2\mathbf{D}_\alpha - \frac{2}{3} (\nabla \cdot \mathbf{u}_\alpha) \mathbf{I} \right] - \frac{2}{3} \rho_\alpha k_\alpha \mathbf{I} \quad (4)$$

$$\mathbf{D}_\alpha = \frac{1}{2} [\nabla \mathbf{u}_\alpha + (\nabla \mathbf{u}_\alpha)^t] \quad (5)$$

In Eq. 4, k_α stands for the turbulent kinetic energy of phase α . For turbulent flows, Eqs. 3 and 4 can be written in a different form by the definition of the modified pressure, \hat{p} :

$$\mathbf{T}_\alpha^{eff} = -\hat{p}_\alpha \mathbf{I} + \hat{\tau}_\alpha^{eff} \quad (6)$$

$$\hat{\tau}_\alpha^{eff} = 2\nu_\alpha^{eff} \mathbf{D}_\alpha \quad (7)$$

$$\hat{p}_\alpha = p_\alpha + \frac{2}{3} \rho_\alpha k_\alpha + \frac{2}{3} \nu_\alpha^{eff} (\nabla \cdot \mathbf{u}_\alpha) \quad (8)$$

The formulation of Eqs. 3 and 4 is used in the twoPhaseEuler-Foam solver of OpenFOAM and that given by Eqs. 6 and 7 is used in CFX.

In Eq. 2, \mathbf{M}_α represents the momentum exchange through the interface or a force per unit volume acting on phase α . This force is usually decomposed as interaction forces responsible for drag, lift and virtual mass, among others. As commented before, there is no general agreement on the literature about the modeling of \mathbf{M}_α . In order to avoid additional modelling uncertainty, it was decided to include only the drag force in \mathbf{M}_α because it is usually the dominant force and both ANSYS CFX and OpenFOAM use exactly the same correlation.

Considering a two-phase flow in which $\alpha = c$ represents the continuous phase and $\alpha = d$ the dispersed phase, the drag force is modelled as shown in Eq. 9.

$$\mathbf{M}_c^D = \frac{1}{2} r_d A_d \rho_c C_D |\mathbf{u}_r| \mathbf{u}_r \quad (9)$$

where $\mathbf{u}_r = \mathbf{u}_d - \mathbf{u}_c$ is the relative disperse phase velocity, A_d is the particle projected area normal to the relative velocity divided by the particle volume and, for spherical particles, A_d simplifies as:

$$A_d = \frac{\pi d_d^2/4}{\pi d_d^3/6} = \frac{3}{2d_d} \quad (10)$$

The drag coefficient, C_D , is usually obtained through correlations which are dependent on the particle size. The Schiller & Naumann (1933) correlation, shown in Eq. 11, is used in this work.

$$C_D = \begin{cases} \frac{24(1+0.15Re_d^{0.687})}{Re_d} & \text{if } Re_d < 1000 \\ 0.44 & \text{if } Re_d > 1000 \end{cases} \quad (11)$$

where the particle Reynolds number is shown below.

$$Re_d = \frac{\rho_c r_c |\mathbf{u}_d - \mathbf{u}_c| d_d}{\mu_d} \quad (12)$$

For the approximation that all particles have the same velocity field, the interaction force is determined using the Sauter mean diameter, d_s , of the particle size distribution (PSD).

Population Balance and DQMOM

A proper way of modelling the evolution of the properties of a relatively large number of particles in a two-phase flow, including the phenomena of aggregation and breakage, is by using population balance. This implies the application of the principle of conservation to the number of particles, deriving an equation in terms of the particle distribution function (PDF), f , which is called the population balance equation (PBE).

The PDF depends on time, external variables, which are the physical space coordinates, and internal variables, which can be any particle property (mass, volume, area, species concentration, energy, etc.). In order to couple to CFD simulations, the monovariate case of interest is the one in which the particles are distributed by their mass or volume. The

general form of the monovariated PBE (in volume, v) including particle breakage and aggregation is given by Eq. 13 (Ramkrishna 2000).

$$\frac{\partial f(\mathbf{x}, v, t)}{\partial t} + \nabla_{\mathbf{x}} \cdot [\mathbf{u}_d f(\mathbf{x}, v, t)] = H(\mathbf{x}, v, t) \quad (13)$$

In Eq. 13, the source term $H(\mathbf{x}, v, t)$ includes the birth and death rates due aggregation and breakage processes, as defined below

$$\begin{aligned} H(\mathbf{x}, v, t) &= \frac{1}{2} \int_0^v a(v-v', v') f(v-v', t) f(v', t) dv' \\ &- \int_0^\infty a(v, v') f(v, t) f(v', t) dv' \\ &+ \int_v^\infty \vartheta(v') b(v') P(v | v') f(v', t) dv' \\ &- b(v) f(v, t), \end{aligned} \quad (14)$$

where $a(v, v')$ is the aggregation frequency, $\vartheta(v)$ is the mean number of particles formed by breakage, $b(v)$ is the breakage frequency and $P(v | v')$ is the conditional probability of generating a particle of volume v once a particle of volume v' has broken. As seen in Eqs. 13 and 14, the PBE forms an integro-differential equation and a proper method is necessary to its solution.

The Direct Quadrature Method of Moments (DQMOM) (Marchisio & Fox 2005) considers a quadrature closure approximation for the integrals of the distribution function in the space of internal variables in terms of Dirac delta functions. For a monovariate problem, the representation of the distribution function is shown in Eq. 15, where ξ_i and w_i are, respectively, the quadrature abscissas and weights, being scalar fields in the physical space.

$$f(\mathbf{x}, v, t) = \sum_{i=1}^n w_i(\mathbf{x}, t) \delta[v - \xi_i(\mathbf{x}, t)] \quad (15)$$

The moments of the distribution function can be approximated using Eq. 15 as shown below.

$$\mu_k = \int_0^\infty v^k f(v, t) dv \simeq \sum_{i=1}^n \xi_i^k w_i, \quad k = 0, \dots, 2n-1 \quad (16)$$

The quadrature approximation given by Eq. 15 can be substituted in the PBE (Eq. 13). The resulting equation involves the derivatives of the Dirac delta distribution, but it can be integrated to give a relation between ordinary functions. If it is operated with $\int v^k \cdot dv$, $k = 0, \dots, 2n-1$, the linear system of equations given by Eq. 17 is obtained after some manipulation (details is Marchisio & Fox 2005), together with transport equations for the weights and weighted abscissas ($\varsigma_i = w_i \xi_i$), which are given in Eqs. 18 and 19.

$$(1-k) \sum_{i=1}^n \xi_i^k \theta_i + k \sum_{i=1}^n \xi_i^{k-1} \kappa_i = \bar{H}_k^{(n)}, \quad k = 0, \dots, 2n-1 \quad (17)$$

$$\frac{\partial w_i(\mathbf{x}, t)}{\partial t} + \nabla \cdot [\mathbf{u}_{d,i} w_i(\mathbf{x}, t)] = \theta_i, \quad i = 0, \dots, n \quad (18)$$

$$\frac{\partial \varsigma_i(\mathbf{x}, t)}{\partial t} + \nabla \cdot [\mathbf{u}_{d,i} \varsigma_i(\mathbf{x}, t)] = \kappa_i, \quad i = 0, \dots, n \quad (19)$$

The solution of Eqs. 17, 18 and 19 are fully coupled. In order to solve the partial differential equations, Eqs. 18 and 19, in the (t, \mathbf{x}) domain, the linear system, Eq. 17, must be solved at every point of this domain to give the source terms θ_i and κ_i .

The term $\bar{H}_k^{(n)}$ in Eq. 17 is the k moment of the source term given by Eq. 14 approximated by the n -point quadrature. This term incorporates the aggregation and breakage effects and is given by:

$$\begin{aligned} \bar{H}_k^{(n)} &= \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n [(\xi_i + \xi_j)^k - \xi_i^k - \xi_j^k] a(\xi_i, \xi_j) w_i w_j \\ &+ \sum_{i=1}^n b(\xi_i) w_i [\vartheta(\xi_i) \pi_k(\xi_i) - \xi_i^k], \end{aligned} \quad (20)$$

where $\pi_k(\xi_i)$ is defined as

$$\pi_k(\xi_i) = \int_0^{\xi_i} v^k P(v | \xi_i) dv \quad (21)$$

Application of DQMOM to more general cases can be found in Marchisio & Fox (2005).

Once the solution is known, any population property can be calculated. The disperse phase fraction, r_d , is of interest as well as the Sauter mean diameter, d_s . Due to the assumption that $\mathbf{u}_{d,i} = \mathbf{u}_d, \forall i$, the latter was used to evaluate the inter-phase forces. Using the quadrature approximation, Eqs. 22 gives r_d , whereas Eq. 23 gives d_s under the assumption of spherical particles.

$$r_d = \int_0^\infty v f(\mathbf{x}, v, t) dv \simeq \sum_{i=1}^n \xi_i w_i = \sum_{i=1}^n \varsigma_i \quad (22)$$

$$d_s = \frac{\int_0^\infty d^3 f(\mathbf{x}, v, t) dv}{\int_0^\infty d^2 f(\mathbf{x}, v, t) dv} \simeq \left(\frac{6}{\pi}\right)^{1/3} \frac{\sum_{i=1}^n \xi_i w_i}{\sum_{i=1}^n \xi_i^{2/3} w_i} \quad (23)$$

CFD Packages

The DQMOM algorithm was implemented in both ANSYS CFX and OpenFOAM frameworks and then used to simulate the zero and one-dimensional verification cases. Further, the coupling of PBE with a two-phase flow considering a single velocity field for the disperse phase was also simulated. This section briefly presents the CFD packages and also explains the CFD-PBE coupling implementation.

ANSYS CFX is a commercial multipurpose CFD code currently developed by ANSYS Inc. The CFX numerical kernel uses the Element Based Finite Volume (EbFVM) method to treat generalised unstructured meshes in Cartesian coordinates. The discrete system of linearised equations is solved using the Algebraic Multigrid (AMG) method accelerated by the Incomplete Lower Upper (ILU) factorisation technique. The Pressure-Velocity coupling is carried out in a single cell

of the collocated grid using a Rhie and Chow like formulation. This solution approach uses a fully implicit discretisation of the equations. In steady state solutions, the false time step technique is applied to the solution relaxation (Raw 1994).

In ANSYS CFX, due to the EbFVm, the control volume is generated around each mesh node by connecting all the neighbour element centres and corresponding edge centres by planes, forming a polyhedron. This approach leads to a higher numerical accuracy, because it has many more integration points per control volume (24 for a hexahedral volume and 60 for a tetrahedral volume) than the classical finite volume method (Raw 1994).

ANSYS CFX has the full Eulerian-Eulerian multiphase formulation, which consist on separated variable fields (velocity, energy, turbulence, mass concentration, ...) for each phase, but restricted to the fact that all phases have to share the same pressure field. The preliminary DQMOM implementation in ANSYS CFX presented in this work considers that all particles share the same velocity field. Thus, only two phases were considered: one continuous and one disperse phases. The disperse phase contains all particles, independently of their sizes.

The transport equations for weights and weighted abscissas for the n -point quadrature, Eqs. 18 and 19, are implemented in ANSYS CFX as additional transported variables in the dispersed phase, which have only temporal, advective and source terms. No diffusive term was present, leading to equations with hyperbolic behaviour. Therefore, special care was taken to verify the accuracy of the advection schemes used in this work.

Using FORTRAN subroutines, θ_i and κ_i , which are the source terms of the DQMOM additional variables transport equations, are calculated for each mesh node from the solution of the linear system given by Eq. 17. The solution of these additional transport equation tracks the DQMOM weights and abscissas through the space and time. However, initial fields for the weights and abscissas of the particle size distribution must be given. One can specify them directly or resort to the Product-Difference Algorithm (PDA) proposed by Gordon (1968), which can calculate the weights and abscissas from the first $2n$ moments of the particle distribution function which must be given. These can be easily calculated if the initial PDF is known.

OpenFOAM (Field Operation And Manipulation) is a free source CFD package written in C++ which uses classes and templates to manipulate and operate scalar, vectorial and tensorial fields (Weller *et al.* 1998). Thus, OpenFOAM can interpretate the true meaning of a field, encapsulating the idea of magnitude and direction of a vector, for instance. Combined with implementations of adequate numerical methods to the discretisation of partial differential equations and to the solution of the resulting linear systems, OpenFOAM is as a good choice to handle CFD problems. Besides, its open-source characteristics is an advantage in the implementation of any addition or modification in the code.

The OpenFOAM finite volume approach in each control volume is performed using the Gauss theorem to transform the volume integrals of flux divergence into surface integrals

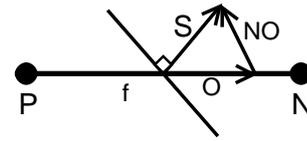


Figure 1: OpenFOAM non-orthogonality treatment that uses the decomposition of \mathbf{S} in the \mathbf{O} and \mathbf{NO} vectors ($\mathbf{S} = \mathbf{O} + \mathbf{NO}$).

of normal fluxes. This leads to a flux discretisation which is the sum of normal fluxes evaluated at each volume face centres by interpolation from the neighbour volume centres. This is quite suitable for orthogonal meshes, but needs correction for non-orthogonal grids. Jasak (1996) implemented an iterative procedure to correct the face fluxes due to the mesh non-orthogonality. Thus, OpenFOAM separate the flux in two parts known as the orthogonal and non-orthogonal contributions. For instance, the volume integral of $\nabla^2 \phi$ leads to the surface integral of $\hat{\mathbf{S}} \cdot \nabla \phi$ which has to be evaluated at all volume faces. For a given face with area S , the integrated term is $\mathbf{S} \cdot (\nabla \phi)_f$, where $\mathbf{S} = S \hat{\mathbf{S}}$. Eq. 25 shows its vector decomposition in the orthogonal contribution, easily calculated, and the non-orthogonal correction.

$$\mathbf{S} \cdot (\nabla \phi)_f = \underbrace{\mathbf{O} \cdot (\nabla \phi)_f}_{\text{orthogonal part}} + \underbrace{\mathbf{NO} \cdot (\nabla \phi)_f}_{\text{non-orthogonal part}} \quad (24)$$

Fig. 1 shows the vectors \mathbf{S} , \mathbf{O} and \mathbf{NO} for the flux evaluation at face f whose neighbour volumes have centres given by points P e N . The non-orthogonal contribution is explicitly evaluated and the number of iterative flux corrections may be set by the user.

Several numerical flux interpolation schemes are available in OpenFOAM, including upwind, central difference, QUICK and flux limiters schemes. Also, many numerical methods for solving linear systems are available, as Gauss-Siedel, AMG and variants of the conjugated gradient methods.

The PBE was implemented in OpenFOAM as a new solver called `pbeFoam` following exactly the DQMOM algorithm (Eqs. 17, 18 and 19) explained previously in this work. Thus, the coupled CFD-PBE solver was implemented adapting an existing two-phase flow solver (`twoPhaseEulerFoam`) using the `pbeFoam` solver.

The `twoPhaseEulerFoam` solver handles incompressible two-phase flows (Rusche 2002). This solver is implemented considering the drag, turbulent drag, lift and virtual mass interphase forces and it may utilise the modified $k - \epsilon$ turbulent model of Gosman (1992). It uses the PISO algorithm for the velocity-pressure coupling proposed by Issa (1986). The interphase drag terms are calculated using a particle diameter which is constant through the simulation. For more details on the numerical algorithms and approximations for the two-phase modelling in OpenFOAM, the reader is encouraged to read Jasak (1996) and Rusche (2002).

This work improves the actual `twoPhaseEulerFoam` solver including the solution of the PBE using DQMOM. For

an incompressible dispersed phase, Eq. 1 implies to:

$$\frac{\partial r_d}{\partial t} + \nabla \cdot (r_d \mathbf{u}_d) = 0 \quad (25)$$

Since there is mass conservation in the breakage and aggregation processes, $\sum_{i=1}^n \kappa_i = 0$, and the sum for all i of Eq. 19 gives Eq. 25.

Rusche (2002) utilises Eq. 25 as re-arranged by Weller (2002) in a conservative and bounded form for all r_d ($0 \leq r_d \leq 1$) given by Eq. 26.

$$\frac{\partial r_d}{\partial t} + \nabla \cdot (\bar{\mathbf{u}} r_d) + \nabla \cdot [\mathbf{u}_r r_d (1 - r_d)] = 0 \quad (26)$$

where $\bar{\mathbf{u}} = r_c \mathbf{u}_c + r_d \mathbf{u}_d$. In fact, this approach implicitly couples the phases through the relative velocity ($\mathbf{u}_r = \mathbf{u}_d - \mathbf{u}_c$).

As seen in Eq. 22, each weighted abscissa ς_i is actually the fraction of the disperse phase represented by class i . Therefore, following Rusche (2002), Eq. 19 was implemented as expressed in the form given by Eq. 27.

$$\frac{\partial \varsigma_i}{\partial t} + \nabla \cdot (\bar{\mathbf{u}} \varsigma_i) + \nabla \cdot [\mathbf{u}_r (1 - r_d) \varsigma_i] = \kappa_i \quad (27)$$

Therefore, in the coupled CFD-PBE solver, the disperse phase fraction is obtained summing up the n weighted abscissas as given by Eq. 22. Once the quadrature is known, the interphase forces are calculated using the Sauter mean diameter evaluated by Eq. 23.

McCoy & Madras (2003) Solution

McCoy & Madras (2003) developed an analytical solution for the monovariated homogeneous PBE given by

$$\begin{aligned} \frac{df(v, t)}{dt} &= \frac{1}{2} \int_0^v a(v - v', v') f(v - v', t) f(v', t) dv' \\ &- \int_0^\infty a(v, v') f(v, t) f(v', t) dv' \\ &+ \int_v^\infty \vartheta(v') b(v') P(v | v') f(v', t) dv' \\ &- b(v) f(v, t) \end{aligned} \quad (28)$$

for the following initial condition and breakage and aggregation kernels:

$$f(v, 0) = \frac{\mu_0(0)^2}{\mu_1(0)} e^{-\frac{\mu_0(0)}{\mu_1(0)} v} \quad (29)$$

$$\vartheta(v) = 2, \quad b(v) = Sv, \quad S = \text{constant} \quad (30)$$

$$a(v - v', v') = C, \quad C = \text{constant} \quad (31)$$

$$P(v | v') = \frac{1}{v'} \quad (32)$$

McCoy & Madras (2003) used the initial number of particles, $\mu_0(0) = \int_0^\infty f(\mathbf{x}, v, t) dv$, and the total particle volume, $\mu_1 = \int_0^\infty v f(\mathbf{x}, v, t) dv$, which is constant by volume

conservation upon breakage and aggregation, to define the following dimensionless variables:

$$\begin{aligned} \Phi(\tau) &= \frac{\mu_0(t)}{\mu_0(0)}, \quad \eta = \frac{v \mu_0(0)}{\mu_1}, \\ \tau &= \mu_0(0) C t, \quad \phi(\eta, \tau) = \frac{f(v, t) \mu_1}{[\mu_0(0)]^2} \end{aligned} \quad (33)$$

Using these variables, the PBE became

$$\begin{aligned} \frac{\partial \phi(\eta, \tau)}{\partial \tau} &= \frac{1}{2} \int_0^\eta \phi(\eta - x, \tau) \phi(x, \tau) dx \\ &- \phi(\eta, \tau) \Phi(\tau) \\ &+ [\Phi(\infty)]^2 \int_\eta^\infty \phi(x, \tau) dx \\ &- \frac{\eta}{2} [\Phi(\infty)]^2 \phi(\eta, \tau) \end{aligned} \quad (34)$$

with the following initial condition:

$$\phi(\eta, 0) = e^{-\eta} \quad (35)$$

whose analytical solution is (McCoy & Madras 2003):

$$\begin{aligned} \phi^{(a)}(\eta, \tau) &= \Phi^2(\tau) e^{-\eta \Phi(\tau)}, \quad (36) \\ \Phi(\tau) &= \Phi(\infty) \left[\frac{1 + \Phi(\infty) \tanh(\Phi(\infty) \tau / 2)}{\Phi(\infty) + \tanh(\Phi(\infty) \tau / 2)} \right] \end{aligned}$$

where $\Phi(\infty) = \sqrt{2S\mu_1/C}/\mu_0(0)$ is the steady-state solution of the dimensionless particle number density.

From Eq. 36, the moments of the dimensionless distribution function can be evaluated as:

$$\mu_k^{(a)}(\tau) = \left[\frac{\Phi(\infty) + \tanh(\frac{\Phi(\infty)\tau}{2})}{\Phi(\infty)(1 + \Phi(\infty) \tanh(\frac{\Phi(\infty)\tau}{2}))} \right]^{k-1} \Gamma(k+1) \quad (37)$$

whereas the initial moments of the dimensionless distribution are given by

$$\mu_k^{(a)}(0) = \Gamma(1+k) \quad (38)$$

Thus, when $\Phi(\infty) = 1$, the solution is invariant with time. This happens when the expression $\mu_0(0) = \sqrt{2S\mu_1/C}$ is satisfied, leading to a process with same intensity for breakage and aggregation. Thus, since $\Phi(\infty)$ can assume arbitrary values, this solution may represent systems with dominant breakage ($\Phi(\infty) > 1$) or dominant aggregation ($\Phi(\infty) < 1$).

Simulation Results

In order to evaluate the numerical code implemented into the CFD packages, different cases were simulated and compared against the analytical solution of McCoy & Madras (2003).

For the numerical tests, a f distribution with $\mu_0(0) = \mu_1 = 1$ can be used. In this case, if $C = 1$ is also set, then $S = \Phi(\infty)^2/2$. Therefore, the dimensional PBE problem solved for this distribution with:

$$a(v, v') = 1, b(v) = \frac{1}{2} \Phi(\infty)^2 v, P(v | v') = \frac{1}{v'}, \vartheta(v) = 2 \quad (39)$$

has a dimensional solution which is numerically equal to the dimensionless analytical solution of McCoy & Madras

(2003). By choosing the values of $\Phi(\infty)$ cases with different behaviour can be obtained. Three cases were considered: (a) the invariant solution ($\Phi(\infty) = 1$), (b) the breakage dominant solution ($\Phi(\infty) = 10$) and (c) the aggregation dominant solution ($\Phi(\infty) = 0.1$).

The McCoy & Madras (2003) analytical solution gives the moments of the distribution, μ_k , in Eq. 37. The DQMOM solution provides the n weights and abscissas, which can be used to approximate the $2n$ first moments of the distribution used in the DQMOM algorithm. When comparing these two sets of moments, two different kinds of error emerge. The first kind of error is related to the quadrature approximation (Eq. 16) of the DQMOM method, which is reduced dramatically with the increase of the number of quadrature points (Marchisio & Fox 2005). The second kind of error is related to the time-integration of the differential equation.

In previous works (Marchisio & Fox 2005; Fan *et al.* 2004), the convergence of DQMOM with the quadrature order has already been proved, showing that 3 – 4 quadrature points are usually sufficient to obtain an accurate solution for CFD simulations. In the present work, our goal is a rigorous verification of the accuracy of DQMOM implementations in CFD codes. That is, our focus was in the accuracy of the time or space integration of the DQMOM equation. Therefore, we excluded the quadrature error from the present comparison and carried out all the simulations using $n = 4$. In order to perform this exclusion, the DQMOM method was applied to Eq. 13, generating a set of ordinary differential equations for weights and abscissas:

$$\frac{dw_i}{dt} = \theta_i, \quad \frac{d\xi_i}{dt} = \kappa_i, \quad i = 1 \dots n \quad (40)$$

which were implemented in a separated numerical code, which uses the DASSL routine (Petzold 1982) for the time integration. The adaptive time step and integration order procedure of DASSL together with a very strict criteria for convergence (absolute and relative tolerances of 10^{-10} and 10^{-12} , respectively) led to a numerical solution for the 4-point quadrature case virtually free of numerical integration error. This was called the “analytical” solution, $w_i^{(a)}$ and $\xi_i^{(a)}$. Of course, this numerical solution still has the DQMOM quadrature error. Therefore, when the results from a DQMOM implementation in a CFD code are compared to this DASSL solution, only integration errors in time or space are shown.

Transient zero-dimensional simulations

The transient zero-dimensional simulations were carried out integrating in time Eqs. 18 and 19 for a null velocity field. Thus, the DQMOM transport equations were numerically solved as simple ordinary differential equations using the explicit Euler method for OpenFOAM and the 2nd order Backward Differencing Scheme (BDS) for ANSYS CFX. The chosen transient schemes were the default ones for each CFD package. An invariant numerical solution was reached when using a constant time step of 10^{-3} s in both codes. Figs. 2 and 3 show the comparison between the analytical abscissas and weights (DASSL solution) and their values obtained

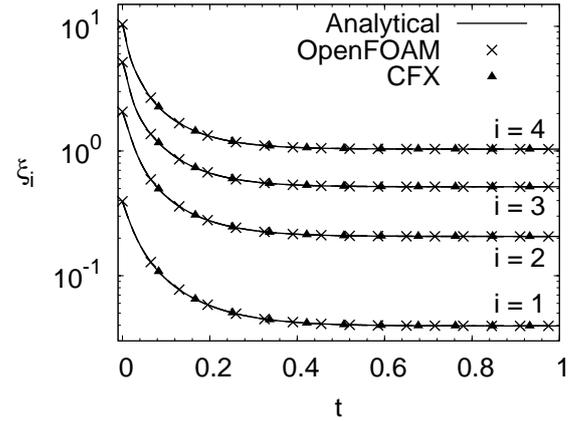


Figure 2: Comparison of the abscissas obtained from ANSYS CFX and OpenFOAM in the 0D simulations for case (b).

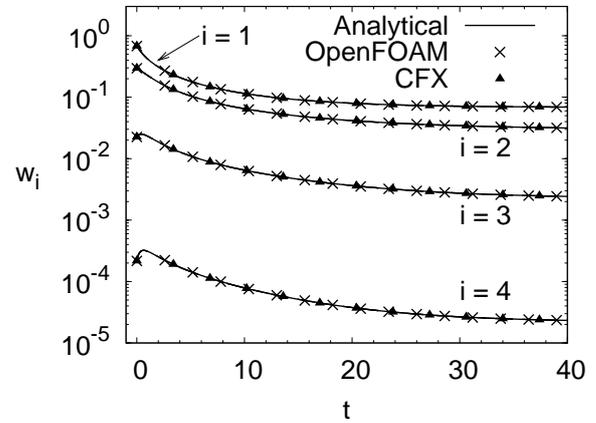


Figure 3: Comparison of the weights obtained from ANSYS CFX and OpenFOAM in the 0D simulations for case (c).

through ANSYS CFX and OpenFOAM for cases (b) and (c), respectively. From these figures, the numerical results using both codes are in excellent agreement to the analytical solution. A more detailed analysis of the solution errors can be done by defining the overall mean relative error (OMRE), χ^t :

$$(\chi^t)^2 = \frac{1}{2nM} \sum_{j=1}^M \sum_{i=1}^n \left\{ \left[\frac{|w_{i,j}^{(a)} - w_{i,j}|}{w_{i,j}^{(a)}} \right]^2 + \left[\frac{|\xi_{i,j}^{(a)} - \xi_{i,j}|}{\xi_{i,j}^{(a)}} \right]^2 \right\} \quad (41)$$

where M is the number of samples regularly taken from the transient simulation and the subscript j represents the time sample.

Considering $M = 1000$, the OMRE results for cases (a), (b) and (c) using ANSYS CFX and OpenFOAM are shown in Table 1.

Analysing the OMRE results shown in Table 1, the OpenFOAM performance was very good, with its OMRE results

Table 1: ANSYS CFX and OpenFOAM OMRE for zero-dimensional simulations of cases (a), (b) and (c).

Cases	χ^t (%)	
	CFX	OpenFOAM
a	$1.52 \cdot 10^{-4}$	$5.12 \cdot 10^{-5}$
b	0.21	0.58
c	$9.55 \cdot 10^{-3}$	$3.63 \cdot 10^{-6}$

reaching 10^{-5} – 10^{-6} accuracy. On the other hand, ANSYS CFX performance was not good. Even though the software claims to use the second-order BDS integration scheme, its OMRE results are not accurate, being in the 10^{-3} – 10^{-4} range for cases (a) and (c) and reaching the surprisingly large value of 0.56 for case (b), the breakage dominant solution. As this code is not open-source, it was not possible to verify the source of this large error.

Steady-state one-dimensional simulations

Although the analytical solution was given for the transient zero-dimensional PBE, it can also be used to verify one-dimensional simulations by transforming the time dependent PBE into a steady uniform-velocity one-dimensional PBE, as shown below for a general variable ϕ .

$$\frac{d\phi}{dt} = \varphi \implies u_z \frac{d\phi}{dz} = \varphi \quad (42)$$

For a domain with length L in the z direction, L/u_z represents the advective time in the propagation of the PBE solution into the domain. From Figs. 2 and 3, it is possible to estimate the time necessary for the abscissas and weights to reach their steady-state values. Therefore, u_z can be chosen in order to these variables present profiles along the z coordinate in the steady-state one-dimensional simulation that are equal to their transient profiles in the transient zero-dimensional simulation.

This work used the one-dimensional geometry shown in Fig. 4, 1 m long in z direction and with edges of 0.1 m in the normal directions. ANSYS CFX simulations needed a 3-D domain but OpenFOAM simulations were carried out in a 2-D domain (0.1×1 m). Of course, due to the 1-D nature of the problem, the mesh had only one volume in the normal directions.

According to Fig. 4, the boundary conditions for all DQMOM variables consists of specified value at the inlet and zero gradient for the others patches. The u_z velocity field is constant and specified for the whole domain in z direction. In order to avoid numerical inconsistencies during DQMOM calculations, the initial fields of the weights and abscissas

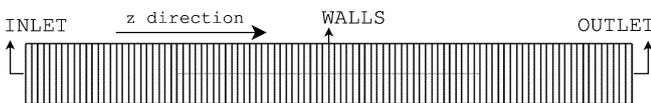


Figure 4: The mesh with 100 volume used in the one-dimensional steady simulations. Boundary patches and z orientation are also shown.

were set to be uniform and equal to the corresponding inlet condition.

From Figs. 2 and 3 the steady-state solution is approximately reached in 0.5 s for case (b) and 40 s for case (c). Thus, equating those values to the advective times L/u_z , u_z values were calculated to be 2 m/s for case (b) and 0.025 m/s for case (c).

The steady-state solution approach was nearly the same for each code. ANSYS CFX uses, by default, a false transient relaxation method for the steady-state simulations. In OpenFOAM, a transient run were adopted until the steady-state pattern was reached. Thus, the total simulation time in OpenFOAM were set as 0.5 s for case (b) and 40 s for case (c), which guarantees the DQMOM variables advection through the whole 1D domain. Besides of that, a maximum residual criterion was used to ensure the steady-state convergence in both codes. Usually, 1000-10000 time steps were necessary to obtain convergence in cases (b) and (c).

A mesh convergence analysis was made for the results of both packages using the OMRE, as defined in Eq. 41, but now M is the number of volumes in the mesh. It was assumed that the mesh convergence was achieved when the resultant OMRE was lower than 1%.

The advection terms of the DQMOM transport equations were discretised using high resolution numerical schemes with flux limiters. The numerical schemes used to evaluate the mesh convergence are based on a blending between first order upwind (UD) and second order approaches, as explained below for both packages.

Advective terms are interpolated in ANSYS CFX using advection schemes based on the blend factor, β , and can be cast in the form:

$$\phi_{ip} = \phi_{UD} + \beta \nabla \phi \cdot \mathbf{r} \quad (43)$$

where ϕ_{UD} is the value at the upwind node, \mathbf{r} is the vector from the upwind node to the integration point ip and $\nabla \phi$ is the numerical approximation of the gradient at the upwind node. When β is zero, the advection scheme yields the first order upwind scheme. When β is one, the solution is the second order accurate upwind scheme. The quantity $\beta \nabla \phi \cdot \mathbf{r}$, called the Numerical Advection Correction (NAC), may be viewed as an anti-diffusive correction applied to the upwind scheme.

The ANSYS CFX user can specify a constant value for β or use the High Resolution (HR) advection scheme. This scheme chooses automatically a β based on flux limiters in order to avoid oscillation in the solution in regions with large gradients. In other regions, the HR scheme keeps β equal to 1 (Barth & Jespersen 1989), to maintain second-order accuracy. This scheme is a quasi-second-order-accurate advection scheme which avoids numeric oscillations.

For the OpenFOAM simulations, the Gamma scheme developed by Jasak (1996), based on the Normalised Variable Diagram (NVD), was used. This scheme uses a smoothing coefficient γ ($0.1 < \gamma < 0.5$) to blend the first order upwind (UD) and the central difference (CD) schemes. The larger the value of γ , the larger the blending introduced between these schemes and the smoother the transition between them. Its implementation in OpenFOAM used the factor $\Gamma = 2\gamma$

Table 2: OMRE results (%) using uniform meshes for simulations with ANSYS CFX and OpenFOAM for case (b).

Software (advective scheme)	Number of volumes		
	250	500	750
CFX (HR)	1.76	0.10	0.05
OF (G1)	4.39	0.36	0.27

as input. The simulations using the γ advection scheme were referred in the present work by a G followed by the Γ ($0.2 < \Gamma < 1$) value employed.

Due its invariant behaviour, the one-dimensional simulations for case (a) presented mesh independent results, differing from the analytical solution only due to machine precision errors. As expected for constant fields, the use of different interpolation functions presented minimum effects. Therefore, the analysis of mesh convergence and advective interpolation functions are presented only for cases (b) and (c).

The first convergence tests were performed with very coarse uniform meshes (25-50 volumes). Due to the fast breakage dynamics originated by its linear kernel, case (b) did not converge in these coarse meshes, presenting large oscillations rates. The w_3 and w_4 variables showed a large gradient near the inlet boundary which might have contributed for the divergence problem. In fact, the coupling introduced by the DQMOM linear system, Eq. 17, leads to divergence even if only one of the DQMOM variables is not well approximated.

More refined meshes were then used to achieve convergence. At least 250 volumes were necessary in both CFD packages to reach a convergent steady-state in case (b). The OMRE results for case (b) using uniform hexaedrical (orthogonal) meshes with 250, 500 and 750 volumes are shown in Table 2.

Table 2 shows that overall convergence was achieved for case (b) in the meshes with 500-750 volumes for the simulations using both softwares. However, the simulations with different packages present convergence differences in

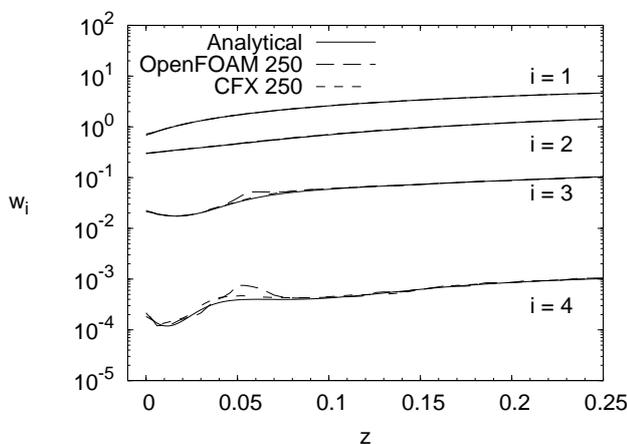


Figure 5: ANSYS CFX and OpenFOAM simulations in the 250-volume mesh for case (b): numerical oscillations in the weights solutions near the inlet.

Table 3: OMRE results (%) using uniform meshes for simulations with ANSYS CFX and OpenFOAM for case (c).

Software (advective scheme)	Number of volumes		
	25	50	100
CFX (HR)	2.31	1.06	0.37
OF (G1)	2.87	1.31	0.33

the OMRE values. Fig. 5 shows the solutions obtained by ANSYS CFX and OpenFOAM for the 250-volume mesh. It is clear that the region with the oscillatory behaviour near the inlet is largely responsible for the OMRE values. Therefore, the differences in the convergence patterns for the results obtained using the two softwares are related to the different advective schemes used by each one. The usage of local β values based on flux limiters by the HR scheme in ANSYS CFX is surely the responsible of its slightly better convergence behaviour than the constant γ coefficient simulations of OpenFOAM. The lack of convergence for case (b) in coarse meshes is also explained by this oscillatory behaviour of the numerical solution near the inlet.

Similar convergence tests were performed for case (c), for which convergence was achieved in coarse meshes. Table 3 shows the convergence of OMRE results for case (c). It is clear that convergence was achieved for the uniform meshes with 50-100 volumes. Although not shown here, differently from case (b), the weight and abscissa profiles did not show a region with large gradients near the inlet. Thus, convergence was much easily achieved for case (c) than for case (b).

Analysis of different advection schemes

Considering the above convergence analysis, the 50 and 500 volume meshes were chosen for cases (b) and (c), respectively, to evaluate different advective schemes implemented in OpenFOAM and ANSYS CFX. For the latter, three schemes with different fixed values of the β blend factor were tested, being referred by BF followed by the β value. In ANSYS CFX, the High Resolution and QUICK interpolations schemes were also compared.

The schemes tested in OpenFOAM included the Gamma scheme, with different γ coefficients, and the QUICK and UD schemes. Details of these interpolation schemes can be found elsewhere (Barth & Jespersen 1989; Jasak 1996; Versteeg & Malalasekera 1995). Tables 4 and 5 present the OMRE results for ANSYS CFX and OpenFOAM, respectively.

Table 4: OMRE (%) results for ANSYS CFX simulations of cases (b) and (c) using different advection schemes.

ANSYS CFX Advection scheme	Case	
	b	c
QUICK	0.04	0.36
HR	0.10	1.06
BF 0.0 (UD)	1.06	4.55
BF 0.75	0.25	1.65
BF 1.0	0.18	0.61

Table 5: OMRE (%) results for OpenFOAM simulations of cases (b) and (c) using different advection schemes.

OpenFOAM Advection scheme	Case	
	b	c
QUICK	0.11	0.47
UD	1.05	4.85
G 0.2	1.44	1.26
G 0.6	1.30	1.26
G 1.0	0.36	1.31

Tables 4 and 5 shows expected behaviours for simulations in sufficiently refined meshes. Second-order accurate methods, QUICK and BF 1.0, gave the smallest values of OMRE irrespectively of case and software, except for the HR scheme of ANSYS CFX in the simulation of case (b), for which the HR scheme is slightly better than BF 1.0. This is caused by the existence of numerical oscillations for this case, as shown before. The first order accurate UD method gave the worst results for the ANSYS CFX simulations. Curiously, the Gamma scheme in OpenFOAM simulation of case (b) was worse than the UD method for two values of Γ . This should also be an effect of the numerical oscillations still present in this mesh. It is also interesting to note that the advection schemes that are similarly implemented in both softwares, QUICK and UD, gave basically the same value of OMRE for both cases.

In general, a blending interpolation scheme with flux limiter is preferred, minimising the occurrence of numerical diffusion and oscillations. Considering the balance between numerical robustness and accuracy, the better choices for the advective scheme are the HR scheme for ANSYS CFX and the Gamma scheme with $\Gamma = 1$ for OpenFOAM.

Emulsion Flow in 2D Backward Facing Step

This section presents the results of a two-dimensional test case that was proposed to evaluate the performance of the implemented PBE solution in a multi-dimensional flow field with strong gradients. The well-known flow through a two-dimensional backward facing step (BFS) was chosen due to its simplicity and the presence of circulation zones with steep gradients in laminar flow.

The considered two-phase mixture consists of a water in oil emulsion. The breakup and coalescence between the water droplets was considered using the same non-physical models of the McCoy & Madras (2003) analytical solution. It should be emphasised that this was not a validation case as

Table 6: Physical properties and inlet conditions of the two-phase mixture.

Physical Properties	oil	water
ρ ($kg\ m^{-3}$)	900	1000
η ($kg\ m^{-1}\ s^{-1}$)	$1 \cdot 10^{-5}$	$1 \cdot 10^{-6}$
Variable	Inlet condition	
d_s (μm)	-	50
r_α (-)	0.9	0.1

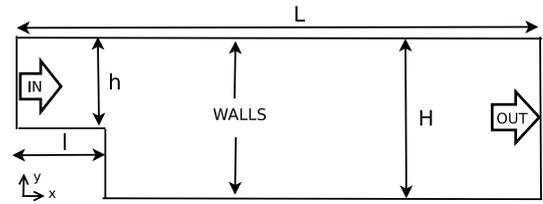


Figure 6: The 2D BFS with parametric dimensions and boundary patches.

no experimental data was available for comparison. Therefore, the breakage and coalescence models are immaterial in this numerical convergence test. Besides, the McCoy & Madras (2003) simple breakup and aggregation kernels do not depend on flow variables, as the turbulent energy dissipation or shear stress, which make them adequate to compare the results of different softwares in laminar flow, without the uncertainties in the implementation of turbulent models.

In these simulations, all particles share the same velocity field. This assumption is reasonable for small droplets in a liquid-liquid flow with small density difference. As the particle size distribution is advected by the flow, droplet breakage and coalescence cause its evolution. Therefore, the degree of PSD change is proportional to the disperse phase residence time. The interphase forces between the continuous and the disperse phases were calculated using the Sauter mean diameter of the particle size distribution obtained through the population balance calculation. Therefore, the Sauter mean diameter is the only influence of the PBE solution in the CFD simulation of the two phase flow.

The physical properties of the liquid-liquid dispersion and the inlet conditions for the Sauter mean diameter and the disperse phase fraction are shown in Table 6, being based on actual water-in-oil emulsions.

The BFS geometry, dimensions and boundary patches are shown in Fig. 6. In the chosen BFS geometry, $L = 11H$, $l = H$ and $h = H/2$. H values are shown in Table 7.

The simulations were performed in quadrilateral and triangular-quadrilateral meshes as shown in Fig. 7, which are called the hexahedral (hexa) and tetrahedral (tetra) meshes, respectively, due to the common practise. The mesh conver-

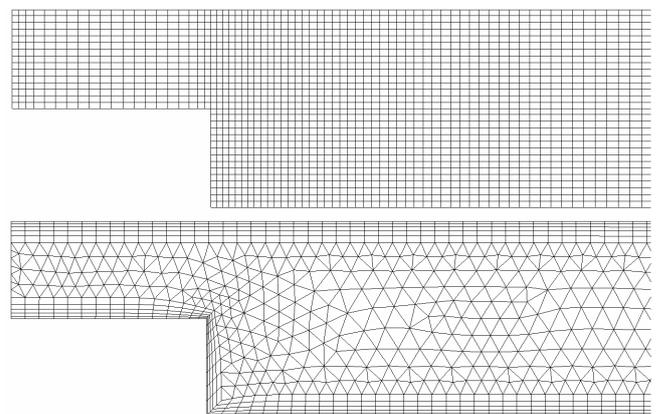


Figure 7: Hexahedral and tetrahedral meshes applied to the BFS geometry.

Table 7: Simulation Parameters for dominant breakup and aggregation situations.

Simulation Parameters	$\Phi(\infty) = 6$	$\Phi(\infty) = 0.1$
Channel Height, H [m]	0.01	0.1
Inlet Velocity [m/s]	0.1	0.01
Simulation Time [s]	0.1	10

gence test on each mesh type was performed using meshes with 4000, 8000 and 16000 elements. Cases with dominant breakage and dominant aggregation were investigated. All case assumed an inlet PDF given by Eq. 29. The flow was laminar, with a Reynolds number of 100 for the continuous phase, in order to avoid possible implementation differences between the turbulence models in the two codes. Also, in order to avoid different implementations of the interphase momentum forces and to get sharper profiles, the drag force, with the Schiller & Naumann (1933) correlation for C_D , was the only interphase force considered in all simulations.

On the previous zero and one-dimensional simulations, the dimensional DQMOM results could be directly compared with the dimensionless analytical solution due the choice of the PBE parameters values in Eq. 33. For the general implementation of the PBE solution, it was verified that a dimensionless DQMOM formulation was beneficial to avoid numerical truncation problems due to the very small values assumed by the dimensional weights. The relations between the dimensional and dimensionless abscissas and weights are based on the dimensionless relations shown in Eq. 33. For the abscissas, Eq. Eq. 33 gives directly:

$$\eta = \frac{\mu_0 v}{\mu_1} \rightarrow \xi_i^\phi = \frac{\mu_0 \xi_i^f}{\mu_1} \quad (44)$$

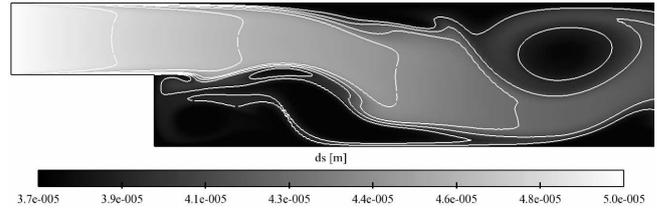
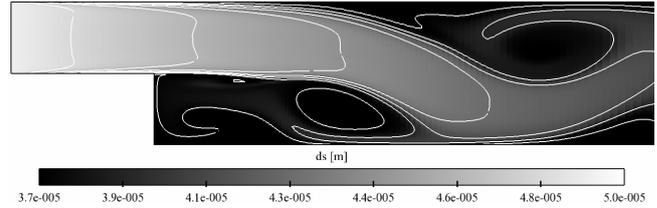
For the weights, the following derivation establishes the required relation.

$$\int_0^\infty \phi(\eta, \tau) d\eta = \frac{1}{\mu_0} \int_0^\infty f(v, t) dv \quad \text{or} \quad \sum_{i=1}^n w_i^\phi = \frac{1}{\mu_0} \sum_{i=1}^n w_i^f \Rightarrow w_i^\phi = \frac{1}{\mu_0} w_i^f \quad (45)$$

In these equations, μ_0 and μ_1 are conveniently chosen constants, usually associated to the dimensional PDF at a boundary or initial condition. In fact, the first moment μ_1 is the disperse phase fraction, r_d , and the inlet value given in Table 6 was used. The zero moment μ_0 can be related to the Sauter mean diameter of the inlet distribution, whose value is also given in Table 6. Manipulating Eq. 23 with Eq. 33 definitions and using the PDF of the inlet condition in dimensionless form, Eq. 35, the value of μ_0 can be calculated from d_s as follows:

$$d_s = \left[\frac{6\mu_1}{\pi\mu_0} \right]^{\frac{1}{3}} \frac{\int_0^\infty \eta e^{-\eta} d\eta}{\int_0^\infty \eta^{2/3} e^{-\eta} d\eta} \Rightarrow \mu_0 = \left[\frac{3}{2d_s \Gamma(\frac{2}{3})} \right]^3 \frac{6\mu_1}{\pi} \quad (46)$$

Using the dimensionless abscissas and weights in the DQMOM algorithm, it is convenient to calculate the disperse phase fraction and the Sauter mean diameter directly from

Figure 8: Sauter mean diameter contour plot in the **finest hexahedral** mesh for the case with dominant **breakage** using **ANSYS CFX**.Figure 9: Sauter mean diameter contour plot in the **finest hexahedral** mesh for the case with dominant **breakage** using **OpenFOAM**.

them for the CFD-PBE coupling. These relations are easily derived from Eqs. 33 and 46 and they are given in Eqs. 47 and 48.

$$r_d = \mu_1 \sum_{i=1}^n \xi_i^\phi w_i^\phi \quad (47)$$

$$d_s = \left[\frac{6\mu_1}{\pi\mu_0} \right]^{\frac{1}{3}} \frac{\sum_{i=1}^n \xi_i^\phi w_i^\phi}{\sum_{i=1}^n (\xi_i^\phi)^{\frac{2}{3}} w_i^\phi} \quad (48)$$

As shown in the zero and one-dimensional simulations, the PDF evolves in the dominant breakage case much faster than in the aggregation case. Thus, in order to use adequate advective times for the BFS flow and the same Reynolds number for both cases, different geometry and velocity scales had to be used, as shown in Table 7. The one-dimensional simulations showed that the simulation of the dominant breakup case (b) required a much finer mesh that the simulation of the dominant aggregation case (c). It was then decided to use a smaller breakage rate ($\Phi(\infty) = 6$) for the two-dimensional flow simulations in the BFS geometry in order to use the same meshes for the cases with dominant breakup and dominant aggregation, although in different scales.

Transient flow simulations were performed, starting from a stagnant flow with the same properties of the inlet. The simulation results at the chosen total simulation time were compared. As the BFS flow has large gradients caused by its recirculation zones, numerical weakness in the solution using different meshes in both software could be visualised.

In ANSYS CFX, the transient simulations were performed using the BFS with a total of 1000 time steps using a coefficient loop convergence criterion of 10^{-4} in the maximum residual of the discretized equations. The HR scheme was used for the advection terms.

In OpenFOAM, the simulations were carried out using the implicit Crank Nicholson scheme for time integration with

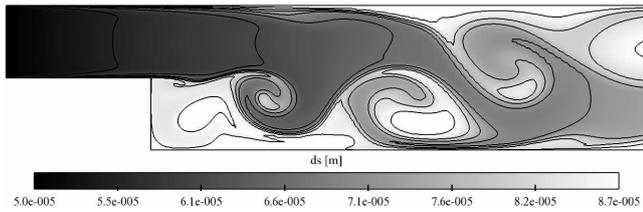


Figure 10: Sauter mean diameter contour plot in the **finest hexahedral** mesh for the case with dominant **aggregation** using ANSYS CFX.

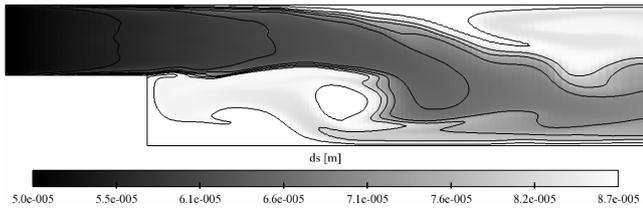


Figure 11: Sauter mean diameter contour plot in the **finest hexahedral** mesh for the case with dominant **aggregation** using OpenFOAM.

adaptive time step, which was controlled to keep the maximum mesh Courant number below 0.3. The advective terms were interpolated with the Gamma scheme with $\Gamma = 1$. The PISO algorithm was performed until the absolute error in the pressure over the iterations was lower than 0.01 and, for tetrahedral meshes, 13 iterations were used in the non-orthogonal face flux correction loop.

The following Figures 8, 9, 10 and 11 show contour plots of the Sauter mean diameter at the final simulation time using the finest hexahedral mesh. It is possible to observe that both codes calculated solutions for each of the cases that are different. As commented before, the PSD evolution mainly depends on the residence time of the disperse phase. Therefore, the Sauter mean diameter field is directly dependent on the disperse flow pathlines. Moreover, as both solvers gave the same solution for the one-dimensional steady-state cases, it could be inferred that the differences in the results obtained by both codes are caused by differences in their velocity field solutions caused by time integration error or incomplete mesh convergence.

Tetrahedral meshes were also tested in order to investigate their behaviour. In the d_s contour plots given in Figures 12 and 13, it can be observed that the solution is poorer for tetrahedral meshes. The gradients and flow structures are not well captured as in the hexahedral meshes, specially for the OpenFOAM solution.

In order to investigate the mesh convergence more carefully, d_s profiles along straight lines were analysed, which are shown in Figure 14. Figures 15 and 17 compare the Sauter mean diameter results extracted along the horizontal line in the middle of the channel inlet in the direction of the main stream using the finest meshes. Figures 16 and 18 show similar results extracted along the vertical line normal to the main stream that crosses the main recirculation zone.

Analysing Figures 15, 16, 17 and 18, it can be observed that the tetrahedral solutions for both cases have lower qual-

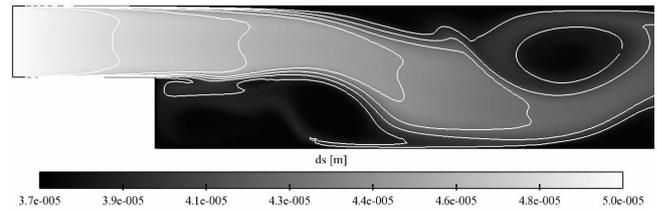


Figure 12: Sauter mean diameter contour plot in the **finest tetrahedral** mesh for the case with dominant **breakage** using ANSYS CFX.

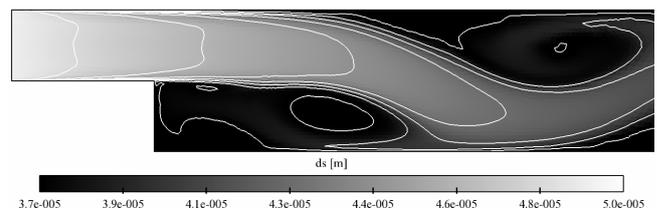


Figure 13: Sauter mean diameter contour plot in the **finest tetrahedral** mesh for the case with dominant **breakage** using OpenFOAM.

ity. The convergence of the results along the horizontal line have a good agreement because they represent the particle distribution evolution along the main flow direction in a region with small velocity gradients. However, the convergence along the vertical line, which crosses the main recirculation zone and the shear layer, is worse, showing different convergence behaviours for each code.

Regardless of the fact that each code presented a different flow solution, when analysing the results individually, it is possible to observe that each code tends to converge with mesh refinement. The charts given in Figures 19 and 20 show the mesh convergence in the vertical line for ANSYS CFX whereas Figures 21 and 22 present the mesh convergence results for OpenFOAM. The ANSYS CFX results showed a clear mesh convergence behaviour, which was slightly better for hexahedral meshes. In OpenFOAM, hexahedral meshes also presented a good mesh convergence, but tetrahedral meshes performed much worse. This different behaviour is even more pronounced for the dominant aggregation case.

In ANSYS CFX, as commented before, the control volume is constructed around each mesh node. Therefore, tetrahedral meshes have fewer nodes than elements, whereas in hexahedral meshes, both quantities are nearly the same. For the tetrahedral meshes used in this work, the number of nodes was almost half of the number of elements. Thus, these tetrahedral meshes have almost half of the control volumes of a hexahedral mesh with the same number of elements

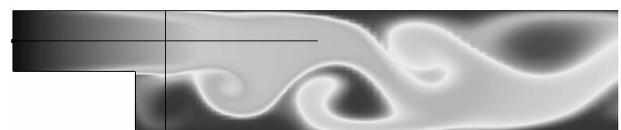


Figure 14: Position of the vertical and horizontal lines used in the mesh convergence studies.

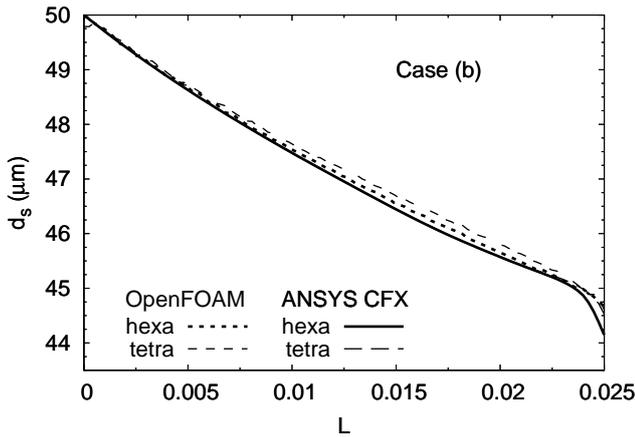


Figure 15: Comparison between ANSYS CFX and OpenFOAM results using the **finest** meshes along **horizontal** line for the case with dominant **breakage**.

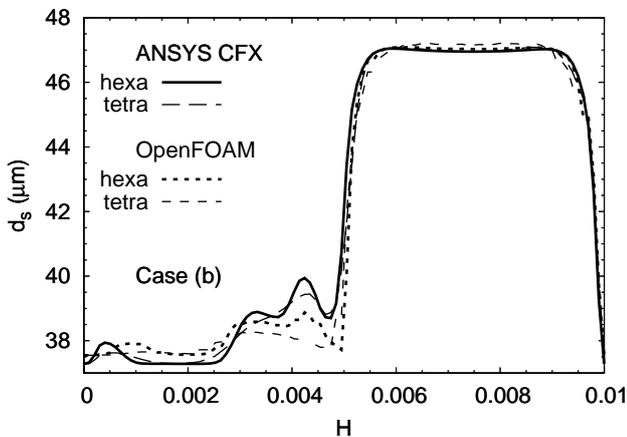


Figure 16: Comparison between ANSYS CFX and OpenFOAM results using the **finest** meshes along **vertical** line for the case with dominant **breakage**.

in ANSYS CFX discretization. Despite the higher numerical accuracy of integration for a polyhedral control volume around a tetrahedral node due to the larger number of integration points, the fewer number of control volumes made the accuracy of the results for the 16K-elements tetrahedral mesh be a little worse than those for the 16K-elements hexahedral mesh. In OpenFOAM, the treatment of the non-orthogonalities present on tetrahedral meshes could be considered the reason to the worse performance of these meshes. For tetrahedral meshes, OpenFOAM would require more refined meshes to achieve convergence.

As commented before, the simulation results for the Sauter mean diameter obtained using both codes have some differences mainly caused by the flow solution, which depends on specific details of the numerical implementation of each code. Moreover, no strong numerical weakness was found in the solution of the DQMOM transport equations, because, in general, the DQMOM variables converged easier than the fluid flow solution. The correct choice of advection schemes based on flux limiters can be pointed out as the key for the success of these simulations.

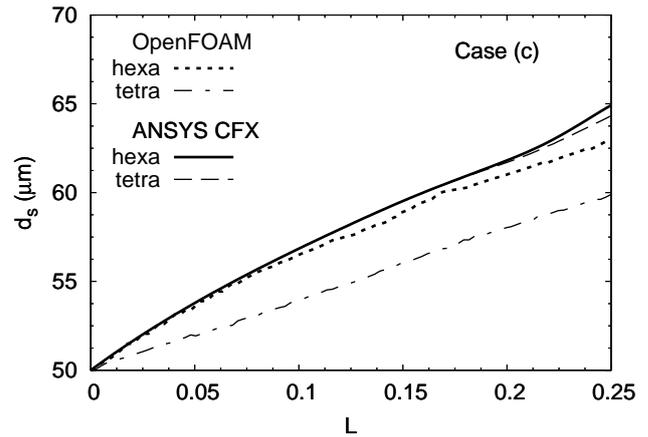


Figure 17: Comparison between ANSYS CFX and OpenFOAM results using the **finest** meshes along **horizontal** line for the case with dominant **aggregation**.

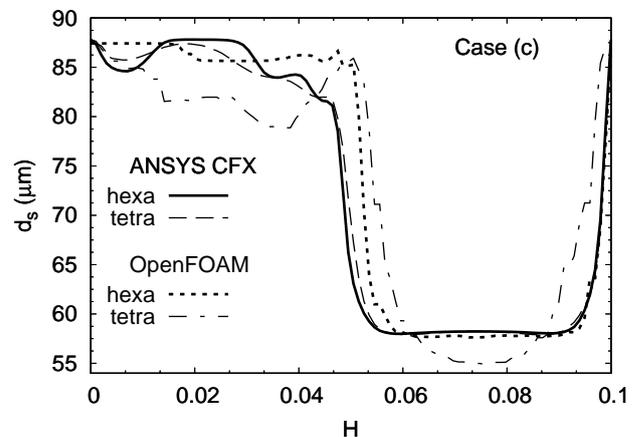


Figure 18: Comparison between ANSYS CFX and OpenFOAM results using the **finest** meshes along **vertical** line for the case with dominant **aggregation**.

Conclusions

The present work carried out the implementation of the DQMOM method in two different CFD codes. An extensive work on the numerical verification of the coupled solution behaviour was accomplished. An analytical solution for a homogeneous time dependent population balance problem with simple models for breakage and aggregation was used for the comparisons. Important issues as mesh refinement, transient and advective schemes specific of each CFD code were also analysed. In order to investigate the errors associated to the solution of the DQMOM transport equations, the quadrature error was suppressed. The error analysis utilised the RMS average of the relative error over all simulated results.

The zero-dimensional analysis, designed to evaluate the time integration accuracy, concluded that the explicit Euler method of OpenFOAM performed very well. However, the second order implicit Euler method, default method in ANSYS CFX, showed significant integration errors.

As the DQMOM transport equations have hyperbolic behaviour, a special attention was dedicated to the advective

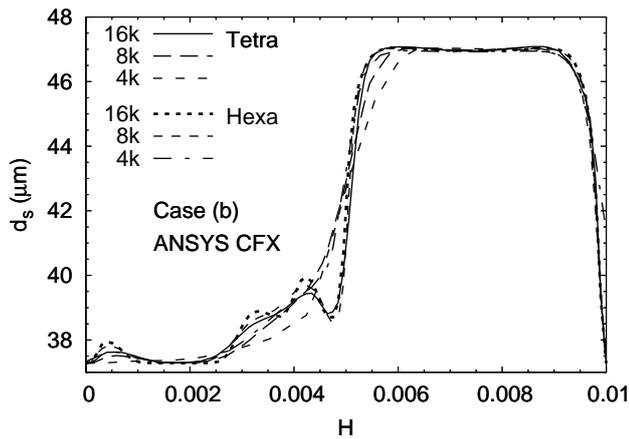


Figure 19: Mesh convergence test for **ANSYS CFX** along the **vertical** line for the case with dominant **breakage**.

schemes used in the solution of the one-dimensional steady-state PBE. After a successful mesh refinement study, the analysis of the advection schemes were accomplished using the intermediate meshes. The quasi-second-order-accurate advection schemes based on flux limiters (High Resolution in ANSYS CFX and Gamma with $\Gamma = 1$ in OpenFOAM) and the second-order accurate QUICK scheme performed very well. In fact, QUICK performed slight better than the HR or G1. ANSYS CFX also showed slightly better results than OpenFOAM. Since the flux-limited advection schemes are more robust, they were chosen for performing some two-dimensional two-phase flow simulations.

Simulations of transient two-dimensional laminar flows of a water-in-oil emulsion through a backward facing step were performed. The convergence of the DQMOM solution was faster than the equations of motion, showing that the time step and the mesh refinement are still controlled by the fluid flow. Each software showed visibly different results in the meshes analysed. Individually, each software showed fairly good mesh convergence.

For both codes, hexahedral meshes produced better results than the tetrahedral ones. In ANSYS CFX, this was ex-

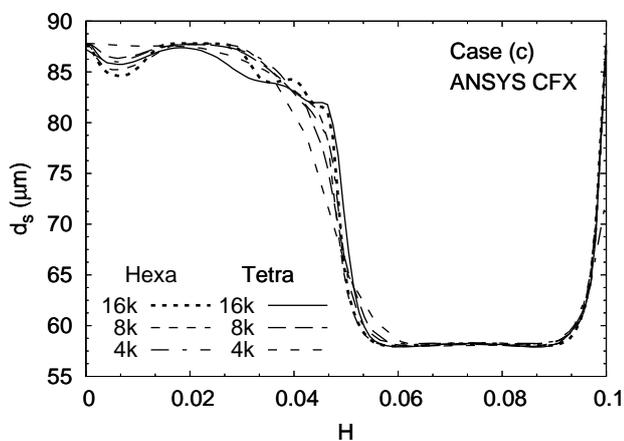


Figure 20: Mesh convergence test for **ANSYS CFX** along the **vertical** line for the case with dominant **aggregation**.

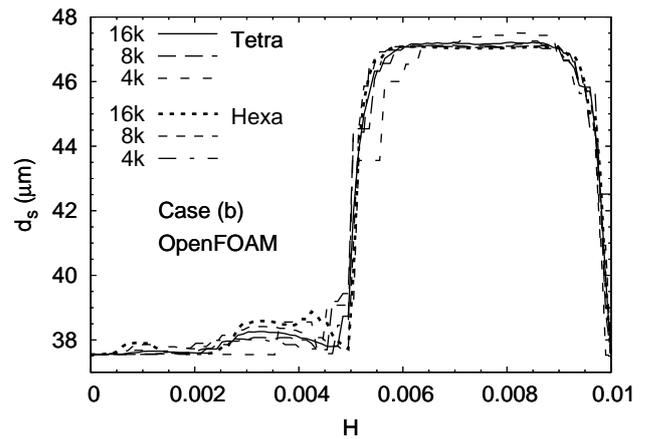


Figure 21: Mesh convergence test for **OpenFOAM** along the **vertical** line for the case with dominant **breakage**.

plained due to the lower number (about half) of control volumes found in tetrahedral mesh with the number of elements kept fixed. In OpenFOAM, the worse results for the tetrahedral meshes could be explained by some weakness of its iterative explicit flux-correction method for non-orthogonal meshes.

The implementations of the DQMOM algorithm in the two CFD codes were verified. No strong difficult were found in solving the hyperbolic DQMOM transport equations. These implementations are reliable enough to be applied in actual flow simulations with realistic breakup and aggregation models.

Acknowledgements

The authors would like to thank CNPq (grant no. 301548/2005-6). L.F.L.R. Silva would also like to acknowledge the financial support given by Chemtech.

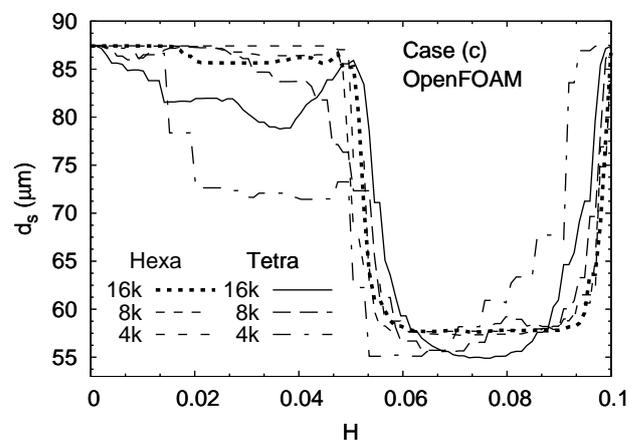


Figure 22: Mesh convergence test for **OpenFOAM** along the **vertical** line for the case with dominant **aggregation**.

References

- Araújo, J.F.M. Study of break-up and coalescence models for dispersed flows. Master Thesis (in Portuguese), PEQ/COPPE/UFRJ, Brazil (2006)
- Barth, T.J. & Jespersen, D.C. The design and application of upwind schemes on unstructured meshes. Technical Report AIAA Paper 89-0366 (1989)
- Bertrand, F., Leclaire, L.-A. & Levecque, G. DEM-based models for the mixing of granular materials. *Chemical Engineering Science*, Volume 60, 2517 – 2531 (2005)
- Bove, S. Computational fluid dynamics of gas-liquid flows including bubble population balances. PHD Thesis, Esbjerg Institute of Engineering, Denmark (2005)
- Drew, D.A. & Passman, S.L. *Theory of Multicomponent Fluids*. Springer, 1st Ed. (1998)
- Enwald, H., Peirano, E. & Almstedt, A.E. Eulerian two-phase flow theory applied to fluidization. *International Journal of Multiphase Flow*, Volume 22, Suppl. 21 – 66 (1996)
- Fan, R. & Marchisio, D.L. & Fox, R.O. Application of the direct quadrature method of moments to polydisperse gas-solid fluidised beds. *Powder Technology*, Volume 139, 7 – 20 (2004)
- Fox, R.O. Bivariate direct quadrature method of moments for coagulation and sintering of particle populations. *Journal of Aerosol Science*, Volume 37, 1562 – 1580 (2005)
- Issa, R.I. Solution of the implicitly discretised fluid flow equations by operator-splitting. *Journal of Computational Physics*, Volume 62, Number 1, 40 – 65 (1986)
- Ishii, M. *Thermo-fluid Dynamic Theory of Two-phase Flow*. Eyrolles, Paris (1975)
- Jasak, H. Error analysis and estimation for the finite volume method with applications to fluid flows. PHD Thesis, Imperial College of Science, Technology and Medicine, UK (1996)
- Gidaspow, D. *Multiphase Flow and Fluidization*. Academic Press, San Diego (1994)
- Gordon, R.G. Error bounds in equilibrium statistical mechanics. *A.I.Ch.E. Journal*, Volume 9, Number 5, 655 – 663 (1968)
- Gosman, A.D., Issa, R.I., Lekakou, C., Looney, M.K. & Politis, S. Multidimensional modelling of turbulent two-phase flows in stirred vessels. *A.I.Ch.E. Journal*, Volume 38, Number 12, 1946 – 1956 (1992)
- Marchisio, D.L. & Fox, R.O. Solution of the population balance equation using the direct quadrature method of moments. *Journal of Aerosol Science*, Volume 36, 43 – 73 (2005)
- McCoy, B.J. & Madras, G. Analytical solution for a population balance equation with aggregation and fragmentation. *Chemical Engineering Science*, Volume 58, 3049 – 3051 (2003)
- McGraw, R. Description of the aerosol dynamics by the quadrature method of moments. *Aerosol Science and Technology*, Volume 27, 255 – 265 (1997)
- Petzold, L.R. A description of DASSL: A differential/algebraic system solver. Technical Report SAND82-8637, Sandia National Laboratories (1982)
- Ramkrishna, D. *Population Balances – Theory and Applications to Particulate Systems in Engineering*. Academic Press, New York (2000)
- Raw, M.J. A coupled algebraic multigrid method for the 3D Navier-Stokes equations. 10th GAMM Seminar, Kiel (1994)
- Rusche, H. Computational fluid dynamics of dispersed two-phase flows at high phase fractions. PHD Thesis, Imperial College of Science, Technology and Medicine, UK (2002)
- Schiller, L. & Naumann, A. Über die grundlegenden berechnungen bei der schwerkraftbereitung. *Z. Vereins deutscher Ing.*, Volume 77, Number 12, 318 – 320 (1933)
- Versteeg, H.K. & Malalasekera, W. *An introduction to computational fluid dynamics - The finite volume method*. Longman Scientific & Technical, UK (1995)
- Weller, H.G., Tabor, G., Jasak, H. & Fureby, C. A tensorial approach to continuum mechanics using object-oriented techniques. *Computers in Physics*, Volume 12, Number 6, 620 – 631 (1998)
- Weller, H.G. Derivation, modelling and solution of the conditionally averaged two-phase flow equations. Technical Report TR/HGW/02, Nabla Ltd. (2002)

Apêndice C

Modelagem e Implementação do Modelo Polidisperso

C.1 Informações sobre o Apêndice

O artigo colocado na sequência foi apresentado na *OpenFOAM International Conference 2007*, que ocorreu em Old Windsor, Reino Unido. Contudo, alguns pequenos erros foram observados no manuscrito e permaneceram na versão aceita e divulgada no congresso. Desta forma, uma errata com referência ao artigo é colocada abaixo.

C.1.1 Errata

Os erros estão organizados por ítems referenciando o local onde estes ocorrem no manuscrito.

1. Leia-se “OpenFOAM is a good choice to...” na seção 4, primeiro parágrafo, sétima linha.
2. Leia-se “...construction of the resulting linear system.” na seção 4, segundo

parágrafo, sexta linha.

3. Leia-se “The following equation is used as an example...” na seção 4, segundo parágrafo, sétima linha.
4. Leia-se “ $\Upsilon_\alpha^c = \dots$ ” e “ $\Upsilon_0^c = \dots$ ” repectivamente nas Eqs. 24 e 25. Neste caso, estas são as formas complementares de Υ_α , onde

$$\begin{aligned}\Upsilon_\alpha^c &= \Upsilon_\alpha + \frac{K_{\alpha f}}{\rho_\alpha}[\mathbf{u}_\alpha] \quad \text{e} \\ \Upsilon_0^c &= \Upsilon_0 + \frac{1}{r_0 \rho_0} \sum_{\alpha=1}^n r_{\alpha f} K_{\alpha f}[\mathbf{u}_0]\end{aligned}$$

5. Leia-se “..., back to step 4.” na etapa 4.5 do algoritmo colocado na página 9.
6. Leia-se “...different approaches validated...” na seção 4, segundo parágrafo, sexta linha.

No presente momento, este trabalho está sendo estendido para ser submetido para publicação em uma revista científica.

Implementation of an Eulerian Multi-phase Model in OpenFOAM and its Application to Polydisperse Two-Phase Flows

Silva L.F.L.R. and Lage P.L.C.*

Programa de Engenharia Química, COPPE, Universidade Federal do Rio de Janeiro
Rio de Janeiro/RJ, P.O. Box 68502, 21941-972, Brazil

*paulo@peq.coppe.ufrj.br

Keywords: Multiphase modeling, Population balance, DQMOM

Abstract

Simulation of polydisperse flows must include the effects of particle-particle interaction, as breakage and aggregation, coupling the population balance equation (PBE) with the multi-phase modelling. In fact, the implementation of efficient and accurate new numerical techniques to solve the PBE is necessary. The Direct Quadrature of Moments (Marchisio & Fox 2005), known as DQMOM, came into view as a promising choice for this implementation. DQMOM is a moment-based method that uses an optimal adaptive quadrature closure whereas only a few quadrature points are usually necessary to obtain an accurate solution.

Recently, Silva *et al.* (2007) extended the current OpenFOAM two-phase Eulerian model (Rusche 2002) using the PBE solved with DQMOM. In this case, all particles classes shared the same velocity field and the *momentum* exchange terms were evaluated using the local instantaneous Sauter mean diameter of the size distribution function. Finally, transient simulations of a water-in-oil emulsion in a backward facing step geometry were performed considering simplified breakage and coalescence kernels.

In the present work, the Rusche (2002) two-phase formulation was extended to a multi-phase approach (n dispersed and 1 continuous phases) and then coupled with the PBE solution by DQMOM. Each one of the disperse phases has its own velocity field. In the present implementation, only the interfacial *momentum* exchange between the continuous and the disperse phases were considered.

This work intends to focus on the multi-phase formulation and the issues regarding the PBE solution by DQMOM. In addition, details of the multi-phase and CFD-PBE coupling algorithms and OpenFOAM programming are provided. Moreover, simulations evaluating the multi-phase code were performed where the same simple breakage and aggregation kernels used by Silva *et al.* (2007) were used in the CFD-PBE simulations.

1 Introduction

The Computational Fluid Dynamic (CFD) simulations have been used with great success in many fields of engineering. The simulation of polydisperse flows, which must include the effects of particle-particle interactions, is an intensive field of CFD research. One of the modeling approaches of polydisperse flows include the coupling of the Eulerian multi-phase model and the population balance equation.

Recently, Silva *et al.* (2007) investigated the numerical behavior of the coupled solution of DQMOM (Marchisio & Fox 2005) using the commercial ANSYS CFX and open-source OpenFOAM CFD packages. The authors first evaluated the solution of DQMOM in transient 0D and steady-state 1D simulations, comparing the results with the McCoy & Madras (2003) analytical solution. Equivalent breakage and aggregation, dominant breakage and dominant aggregation simulations were performed. Next, the PBE solution was coupled with the two-phase flow using the simplified breakage and aggregation kernels of McCoy & Madras (2003). In this approach, all the particles shared the same velocity field and the interphase forces were calculated using the mean Sauter diameter obtained using the PBE variables. Dominant breakage and aggregation simulations were conducted in a backward facing-step (BFS) geometry presenting a good agreement among the CFD packages.

In the present work, the incompressible two-phase code (Rusche 2002) currently implemented in OpenFOAM-1.4 was extended to handle $n + 1$ phases, considering one continuous and n disperse phases. In addition, the CFD-PBE coupling using DQMOM was implemented for this multi-phase approach. In the present approach, each phase has its own velocity and diameter field. The formulation and details of code programming are presented in this work. As in Silva *et al.* (2007), numerical tests of the multi-phase code were performed in a BFS geometry using a two-phase mixture. These results were compared with the simulation results of the `twoPhaseEulerFoam` solver. Numerical tests using three-phase systems and parallel simulations were performed as well. Finally, the CFD-PBE coupling was simulated using the same simple models for breakage and aggregation of Silva *et al.* (2007).

2 Eulerian multi-phase model

The multi-phase model is based on the mean mass and *momentum* conservation equations to describe the dynamic behaviour of the multi-phase flow. These equations are obtained through average procedures, which introduces the average occurrence of phase α , r_α , known as the phase fraction. Due to the average procedure, additional terms appears in the mean conservation equations which need to be modeled. These terms represent the phenomena that occurs on scales smaller than the averaging scale. Details about the theory and formulation of the multi-phase model can be found in Ishii (1975), Drew & Prassman (1999) and Bove (2005).

Assuming that the interfacial tension effects are not important, it is possible to consider that all phases share the same pressure field. In this case, the average multi-phase equations are given by:

$$\frac{\partial(r_\alpha\rho_\alpha)}{\partial t} + \nabla \cdot (r_\alpha\rho_\alpha\mathbf{u}_\alpha) = 0 \quad (1)$$

$$\frac{\partial(r_\alpha \rho_\alpha \mathbf{u}_\alpha)}{\partial t} + \nabla \cdot (r_\alpha \rho_\alpha \mathbf{u}_\alpha \mathbf{u}_\alpha) = \nabla \cdot (r_\alpha \mathbf{T}_\alpha^{eff}) + \mathbf{M}_\alpha + r_\alpha \rho_\alpha \mathbf{g} \quad (2)$$

where \mathbf{T}_α^{eff} represents the effective tensor composed by the mean viscous and turbulent tensors of phase α , which is usually modelled using the Newtonian functional form:

$$\mathbf{T}_\alpha^{eff} = -p_\alpha \mathbf{I} + \tau_\alpha^{eff} \quad (3)$$

$$\tau_\alpha^{eff} = \nu_\alpha^{eff} \left[2\mathbf{D}_\alpha - \frac{2}{3} (\nabla \cdot \mathbf{u}_\alpha) \mathbf{I} \right] - \frac{2}{3} \rho_\alpha k_\alpha \mathbf{I} \quad (4)$$

$$\mathbf{D}_\alpha = \frac{1}{2} [\nabla \mathbf{u}_\alpha + (\nabla \mathbf{u}_\alpha)^t] \quad (5)$$

where k_α , in Eq. 4, stands for the turbulent kinetic energy of phase α . In this work, $n+1$ phases are considered where there is one continuous phase, referred with subscript 0, and n disperse phases.

In Eq. 2, \mathbf{M}_α represents the momentum exchange through the interface or a force per unit volume acting on phase α . This force is usually decomposed as interaction forces responsible for drag, lift and virtual mass, among others (Rusche 2002). In this work, only the drag force is considered since it is the dominant force in the analyzed cases of this work. The drag force for the disperse phase α is modelled as shown in Eq. 6.

$$\mathbf{M}_\alpha = \frac{1}{2} r_\alpha A_\alpha \rho_0 C_{D,\alpha} |\mathbf{u}_{r,\alpha}| \mathbf{u}_{r,\alpha} \quad (6)$$

where $\mathbf{u}_{r,\alpha} = \mathbf{u}_0 - \mathbf{u}_\alpha$ is the relative disperse phase velocity, A_α is the particle projected area normal to the relative velocity divided by the particle volume and, for spherical particles, A_α simplifies as:

$$A_\alpha = \frac{\pi d_\alpha^2 / 4}{\pi d_\alpha^3 / 6} = \frac{3}{2d_\alpha} \quad (7)$$

The drag coefficient, $C_{D,\alpha}$, is usually obtained through correlations which are dependent on the particle size, d_α . The Schiller & Naumann (1933) correlation is used in this work.

Finally, only the interfacial *momentum* exchange between the continuous and the disperse phases were considered, i.e., $\mathbf{M}_0 = -\sum_{\alpha=1}^n \mathbf{M}_\alpha$.

3 Population balance and DQMOM

The general form of the monovariated PBE (in volume, v) including particle breakage and aggregation is given by Eq. 8 (Ramkrishna 2000).

$$\frac{\partial f(\mathbf{x}, v, t)}{\partial t} + \nabla_{\mathbf{x}} \cdot [\bar{\mathbf{u}}_\alpha f(\mathbf{x}, v, t)] = H(\mathbf{x}, v, t) \quad (8)$$

In Eq. 8, the source term $H(\mathbf{x}, v, t)$ includes the birth and death rates due aggregation and breakage processes, as defined below

$$\begin{aligned} H(\mathbf{x}, v, t) = & \frac{1}{2} \int_0^v a(v-v', v') f(v-v', t) f(v', t) dv' - \int_0^\infty a(v, v') f(v, t) f(v', t) dv' \\ & + \int_v^\infty \vartheta(v') b(v') P(v | v') f(v', t) dv' - b(v) f(v, t), \end{aligned} \quad (9)$$

where $a(v, v')$ is the aggregation frequency, $\vartheta(v)$ is the mean number of particles formed by breakage, $b(v)$ is the breakage frequency and $P(v | v')$ is the conditional probability of generating a particle of volume v once a particle of volume v' has broken. As seen in Eqs. 8 and 9, the PBE forms an integro-differential equation and a proper method is necessary to its solution.

The Direct Quadrature Method of Moments (DQMOM) (Marchisio & Fox 2005) considers a quadrature closure approximation for the integrals of the distribution function in the space of internal variables in terms of Dirac delta functions. For a monovariate problem, the representation of the distribution function is shown in Eq. 10, where ξ_α and w_α are, respectively, the quadrature abscissas and weights, being scalar fields in the physical space.

$$f(\mathbf{x}, v, t) = \sum_{\alpha=1}^n w_\alpha(\mathbf{x}, t) \delta[v - \xi_\alpha(\mathbf{x}, t)] \quad (10)$$

The quadrature approximation given by Eq. 10 can be substituted in the PBE (Eq. 8). The resulting equation involves the derivatives of the Dirac delta distribution, but it can be integrated to give a relation between ordinary functions. If it is operated with $\int v^k \cdot dv$, $k = 0, \dots, 2n - 1$, the linear system of equations given by Eq. 11 is obtained after some manipulation (details in Marchisio & Fox 2005), together with transport equations for the weights and weighted abscissas ($\varsigma_\alpha = w_\alpha \xi_\alpha$), which are given in Eqs. 12 and 13.

$$(1 - k) \sum_{\alpha=1}^n \xi_\alpha^k \theta_\alpha + k \sum_{\alpha=1}^n \xi_\alpha^{k-1} \kappa_\alpha = \bar{H}_k^{(n)}, \quad k = 0, \dots, 2n - 1 \quad (11)$$

$$\frac{\partial w_\alpha(\mathbf{x}, t)}{\partial t} + \nabla \cdot [\mathbf{u}_\alpha w_\alpha(\mathbf{x}, t)] = \theta_\alpha, \quad \alpha = 1, \dots, n \quad (12)$$

$$\frac{\partial \varsigma_\alpha(\mathbf{x}, t)}{\partial t} + \nabla \cdot [\mathbf{u}_\alpha \varsigma_\alpha(\mathbf{x}, t)] = \kappa_\alpha, \quad \alpha = 1, \dots, n \quad (13)$$

The solution of Eqs. 11, 12 and 13 are fully coupled. In order to solve the partial differential equations, Eqs. 12 and 13, in the (t, \mathbf{x}) domain, the linear system, Eq. 11, must be solved at every point of this domain to give the source terms θ_α and κ_α .

The term $\bar{H}_k^{(n)}$ in Eq. 11 is the k moment of the source term given by Eq. 9 approximated by the n -point quadrature. This term incorporates the aggregation and breakage effects and is given by:

$$\bar{H}_k^{(n)} = \frac{1}{2} \sum_{\alpha=1}^n \sum_{\beta=1}^n [(\xi_\alpha + \xi_\beta)^k - \xi_\alpha^k - \xi_\beta^k] a(\xi_\alpha, \xi_\beta) w_\alpha w_\beta + \sum_{\alpha=1}^n b(\xi_\alpha) w_\alpha [\vartheta(\xi_\alpha) \pi_k(\xi_\alpha) - \xi_\alpha^k], \quad (14)$$

where $\pi_k(\xi_\alpha)$ is defined as

$$\pi_k(\xi_\alpha) = \int_0^{\xi_\alpha} v^k P(v | \xi_\alpha) dv \quad (15)$$

Once the solution is known, any population property can be calculated. The disperse phase fraction, r_α , is of interest as well as the phase diameter, d_α . Using the quadrature approximation, Eq. 16 gives d_α under the assumption of spherical particles

$$d_\alpha = \left(\frac{6\xi_\alpha}{\pi} \right)^{1/3} \quad (16)$$

whereas Eq. 17 gives the global volumetric fraction of the disperse phase $\sum_{\alpha=1}^n r_{\alpha}$,

$$\sum_{\alpha=1}^n r_{\alpha} = \int_0^{\infty} v f(\mathbf{x}, v, t) dv \simeq \sum_{\alpha=1}^n \xi_{\alpha} w_{\alpha} = \sum_{\alpha=1}^n \varsigma_{\alpha}, \quad (17)$$

and the volumetric fraction for the α disperse phase is ς_{α} .

4 The OpenFOAM CFD package

OpenFOAM (Field Operation And Manipulation) is a free source CFD package written in C++ which uses classes and templates to manipulate and operate scalar, vectorial and tensorial fields (Weller *et al.* 1998). Thus, OpenFOAM can interpretate the true meaning of a field, encapsulating the idea of magnitude and direction of a vector, for instance. Combined with implementations of adequate numerical methods to the discretisation of partial differential equations and to the solution of the resulting linear systems, OpenFOAM is as a good choice to handle CFD problems. Besides, its open-source characteristics is an advantage in the implementation of any addition or modification in the code.

OpenFOAM provides two static functions, **fvm** and **fvc**, to discretise the differential operators of the field, e.g., ∇^2 , $\nabla \cdot$ and $\partial/\partial t$. The first function is used to discretise implicit derivatives resulting in a linear system to be solved. In fact, the implicit source terms of the linear system can be defined using the **fvm** function as well. Therefore, the **fvm** function provides the PDE discretisation using the finite volume method and the construction of resulting linear system. On the other hand, the **fvc** function calculates explicit derivatives and can be used anywhere in the code. The following equation as an example and to show the notation of these functions in this work.

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\phi \mathbf{u}) - \nabla^2 (\mu \mathbf{u}) = -\nabla p \quad (18)$$

Eq. 18 must be discretised in terms of \mathbf{u} using the **fvm** function and ∇p as an explicit source term with the **fvc** function. For instance, the notation for an implicit discretisation is $[\bullet[\mathbf{u}]]$ where \bullet represents the differential operator discretised in terms of the $[\mathbf{u}]$ variable. The explicit operation is referred with the underlined notation over the operator. Thus, the discretisation of the Eq. 18 is:

$$\left[\frac{\partial \rho[\mathbf{u}]}{\partial t} \right] + [\nabla \cdot (\phi[\mathbf{u}])] - [\nabla^2 (\mu[\mathbf{u}])] = -\underline{\nabla} p \quad (19)$$

For further informations about these discretisation procedures, the reader should read the Programmer's Guide available with OpenFOAM.

5 Implementation of the multi-phase code

The formulation of the multi-phase model presented in this work is based on Rusche (2002). In fact, Rusche (2002) describes the solution procedure using a multi-phase approach for the momentum equations and uses the interphase forces and the pressure correction algorithm specifically for two-phase flows. Thus, only the interphase forces and the pressure correction were modified for the multi-phase approach. The momentum

equations are briefly presented in this work, but further details about them can be found on Rusche (2002)¹.

5.1 Multi-phase equations

Rusche (2002) utilises the “phase intensive” version of the incompressible momentum equation which is obtained dividing Eq. 2 by ρ_α and r_α considering $\alpha = 0, \dots, n$.

$$\frac{\partial \mathbf{u}_\alpha}{\partial t} + \mathbf{u}_\alpha \cdot \nabla \mathbf{u}_\alpha + \nabla \cdot (\tau_\alpha^{eff}) + \frac{\nabla r_\alpha}{r_\alpha} \cdot \tau_\alpha^{eff} = -\frac{1}{\rho_\alpha} \nabla p + \frac{\mathbf{M}_\alpha}{r_\alpha \rho_\alpha} + \mathbf{g} \quad (20)$$

Rusche (2002) describes in details the manipulation and discretisation procedure of the l.h.s. of Eq. 20 (see section 3.2.2, pg. 109 of Rusche 2002) which results in semi-discretised equation shown below.

$$\mathbf{\Upsilon}_\alpha = -\frac{\nabla p}{\rho_\alpha} + \frac{\mathbf{\Omega}_\alpha}{r_\alpha \rho_\alpha} + \mathbf{g} \quad (21)$$

where $\mathbf{\Upsilon}_\alpha$ and $\mathbf{\Omega}_\alpha$ refer respectively to the discretised forms of the l.h.s. of Eq. 20 and the interphase term. The interphase term is discretised semi-implicitly as source terms (see pg. 114 of Rusche 2002). Thus, the drag force of the disperse phases is

$$\mathbf{\Omega}_\alpha = [r_{\alpha f} K_{\alpha f} (\mathbf{u}_0 - [\mathbf{u}_\alpha])] \quad (22)$$

where $K_\alpha = \frac{1}{2} \rho_0 A_\alpha C_{D,\alpha} |\mathbf{u}_{r,\alpha}|$ and the subscript f represents the interpolation to the face centres. Similarly, the interphase term for the continuous phase is

$$\mathbf{\Omega}_0 = -\sum_{\alpha=1}^n [r_{\alpha f} K_{\alpha f} ([\mathbf{u}_0] - \mathbf{u}_\alpha)] \quad (23)$$

The implicit source terms of Eqs. 22 and 23 are included in the discretised l.h.s., $\mathbf{\Upsilon}_\alpha$, for the continuous ($\alpha = 0$) and disperse ($\alpha = 1, \dots, n$) phases resulting in the semi-discretised forms shown below.

$$\mathbf{\Upsilon}_\alpha = -\frac{\nabla p}{\rho_\alpha} + \frac{K_\alpha}{\rho_\alpha} \mathbf{u}_0 + \mathbf{g} \quad (24)$$

$$\mathbf{\Upsilon}_0 = -\frac{\nabla p}{\rho_0} + \frac{1}{r_0 \rho_0} \sum_{\alpha=1}^n r_\alpha K_\alpha \mathbf{u}_\alpha + \mathbf{g} \quad (25)$$

The solution of the above equations does not guarantee the continuity. It can be achieved by correcting the velocities using an update pressure field, which is chosen such that continuity is satisfied.

5.2 Phase momentum correction equation

The pressure equation is derived from the semi-discretised form of the momentum equations shown in Eq. 21,

$$(\mathbf{A}_\alpha)_D \mathbf{u}_\alpha = (\mathbf{A}_\alpha)_H - \frac{\nabla p}{\rho_\alpha} + \frac{K_\alpha}{\rho_\alpha} \mathbf{u}_0 + \mathbf{g} \quad (26)$$

¹Several thesis related with OpenFOAM can be found at <http://foamcfd.org/resources/theses.html>

where \mathbf{A}_α denotes the system of linear equations arising from the discretisation of the momentum equation whereas the $(\)_D$ operator represents the diagonal coefficients of the matrix and $(\)_H$ is the ‘‘H’’ operator. The ‘‘H’’ operator is an approximated solution of the linear system obtained from the discretised equations that includes only the off-diagonal terms of the matrix. Further details about this operations can be found in Jasak (1996) and Rusche (2002).

Eq. 26 can be re-arranged to provide the phase momentum correction equation for the disperse phases.

$$\mathbf{u}_\alpha = \frac{(\mathbf{A}_\alpha)_H}{(\mathbf{A}_\alpha)_D} - \frac{\nabla p}{\rho_\alpha(\mathbf{A}_\alpha)_D} + \frac{K_\alpha}{\rho_\alpha(\mathbf{A}_\alpha)_D} \mathbf{u}_0 + \frac{1}{(\mathbf{A}_\alpha)_D} \mathbf{g} \quad (27)$$

Consequently, the momentum correction equation for the continuous phase can be derived using the same procedure and it is shown in Eq. 28.

$$\mathbf{u}_0 = \frac{(\mathbf{A}_0)_H}{(\mathbf{A}_0)_D} - \frac{\nabla p}{\rho_0(\mathbf{A}_0)_D} + \frac{1}{r_0\rho_0(\mathbf{A}_0)_D} \sum_{\alpha=1}^n r_\alpha K_\alpha \mathbf{u}_\alpha + \frac{1}{(\mathbf{A}_0)_D} \mathbf{g} \quad (28)$$

5.3 The pressure equation

The solution of the pressure equation provides corrections for updating the pressure, fluxes and velocities so that continuity is obeyed. Rusche (2002) combined the volumetric continuity equation, shown in Eq. 1, into one to obtain an expression for the mixture pressure. In the multi-phase approach, the mixture continuity equation evaluated at the cell faces is shown below.

$$\nabla \cdot \left(r_{0f} \phi_0 + \sum_{\alpha=1}^n r_{\alpha f} \phi_\alpha \right) = 0 \quad (29)$$

The volumetric phase fluxes, ϕ_α , are obtained interpolating the momentum correction equations, Eqs. 27 and 28, to the face centres. In a general form, the volumetric flux for $\alpha = 0, \dots, n$ is shown in Eq. 30.

$$\phi_\alpha = \phi_\alpha^* - \left(\frac{1}{\rho_\alpha(\mathbf{A}_\alpha)_D} \right)_f |\mathbf{S}| \nabla_f^\perp p \quad (30)$$

where ϕ_α^* is expressed for the dispersed phases ($\alpha = 1, \dots, n$) as

$$\phi_\alpha^* = \left(\frac{(\mathbf{A}_\alpha)_H}{(\mathbf{A}_\alpha)_D} \right)_f \cdot \mathbf{S} + \left(\frac{K_\alpha}{\rho_\alpha(\mathbf{A}_\alpha)_D} \right)_f \phi_0 + \left(\frac{1}{(\mathbf{A}_\alpha)_D} \right)_f \mathbf{g} \cdot \mathbf{S} \quad (31)$$

and ϕ_0^* for the continuous phase ($\alpha = 0$) as

$$\phi_0^* = \left(\frac{(\mathbf{A}_0)_H}{(\mathbf{A}_0)_D} \right)_f \cdot \mathbf{S} + \left(\frac{1}{r_0\rho_0(\mathbf{A}_0)_D} \right)_f \sum_{\alpha=1}^n r_{\alpha f} K_{\alpha f} \phi_\alpha + \left(\frac{1}{(\mathbf{A}_0)_D} \right)_f \mathbf{g} \cdot \mathbf{S} \quad (32)$$

The pressure equation is constructed by substituting the volumetric fluxes of the continuous and disperse phases, shown in Eqs. 30, 31 and 32, into Eq. 29. The resulting equation is discretised implicitly as a diffusion term.

$$\left[\nabla \cdot \left[r_{0f} \left(\frac{1}{\rho_0(\mathbf{A}_0)_D} \right)_f + \sum_{\alpha=1}^n r_{\alpha f} \left(\frac{1}{\rho_\alpha(\mathbf{A}_\alpha)_D} \right)_f \right] \nabla [p] \right] = \nabla \cdot \left(r_{0f} \phi_0^* + \sum_{\alpha=1}^n r_{\alpha f} \phi_\alpha^* \right) \quad (33)$$

The mixture pressure field is determined considering the validity of the continuity of the volumetric fluxes. The phase fluxes, shown in Eq. 30, are corrected after the solution of the pressure through Eq. 33 where an iterative procedure is necessary to achieve convergence.

5.4 The volumetric fraction equation

Rusche (2002) utilises Eq. 1 as re-arranged by Weller (2002) in a conservative and bounded form (see section 3.2.6 of Rusche 2002). The multi-phase equation for the volumetric fraction was derived similarly, using the mixture velocity, defined as $\bar{\mathbf{u}} = \sum_{\alpha=0}^n r_{\alpha} \mathbf{u}_{\alpha}$, to obtain an expression for \mathbf{u}_{α} .

$$\mathbf{u}_{\alpha} = \bar{\mathbf{u}} + r_0 \mathbf{u}_{r,0} + \sum_{\substack{i=1 \\ i \neq \alpha}}^n r_i \mathbf{u}_{r,i} \quad (34)$$

using the phase relative velocities $\mathbf{u}_{r,0} = \mathbf{u}_{\alpha} - \mathbf{u}_0$ and $\mathbf{u}_{r,i} = \mathbf{u}_{\alpha} - \mathbf{u}_i$.

The expression for \mathbf{u}_{α} presented by Eq. 34 is substituted into the incompressible form of Eq. 1 resulting the multi-phase volumetric fraction equations.

$$\frac{\partial r_{\alpha}}{\partial t} + \nabla \cdot (\bar{\mathbf{u}} r_{\alpha}) + \nabla \cdot (r_0 \mathbf{u}_{r,0} r_{\alpha}) + \nabla \cdot \left(\sum_{\substack{i=1 \\ i \neq \alpha}}^n r_i \mathbf{u}_{r,i} r_{\alpha} \right) = 0 \quad (35)$$

The non-linear characteristic of the multi-phase volumetric fraction equations requires iteration to achieve convergence. The discretised form of Eq. 35 is shown below.

$$\left[\frac{\partial [r_{\alpha}]}{\partial t} \right] + \left[\nabla \cdot (\bar{\phi} [r_{\alpha}]) \right] + \left[\nabla \cdot (r_{0f} \phi_{r,0} [r_{\alpha}]) \right] + \left[\nabla \cdot \left(\sum_{\substack{i=1 \\ i \neq \alpha}}^n r_{if} \phi_{r,i} [r_{\alpha}] \right) \right] = 0 \quad (36)$$

5.4.1 The multi-phase CFD-PBE coupling

The coupling of the multi-phase code and the population balance is accomplished using the weights and abscissas to obtain important informations of the disperse phase population properties.

As seen in Eq. 17, each weighted abscissa ς_{α} is actually the fraction of the disperse phase represented by class α . Therefore, following the same procedure applied to the multi-phase volumetric fraction equation, Eq. 13 was implemented as expressed in the form given by Eq. 37.

$$\frac{\partial \varsigma_{\alpha}}{\partial t} + \nabla \cdot (\bar{\mathbf{u}} \varsigma_{\alpha}) + \nabla \cdot (\mathbf{u}_{r,0} \varsigma_{\alpha}) - \nabla \cdot \left(\sum_{i=1}^n \varsigma_i \mathbf{u}_{r,0} \varsigma_{\alpha} \right) + \nabla \cdot \left(\sum_{\substack{i=1 \\ i \neq \alpha}}^n \varsigma_i \mathbf{u}_{r,i} \varsigma_{\alpha} \right) = \kappa_{\alpha} \quad (37)$$

Therefore, in the coupled CFD-PBE solver, the solution of the DQMOM linear system, shown in Eq. 11, provides the source terms for the weights and disperse phase fractions (weighted abscissas) transport equations, shown respectively in Eqs. 12 and 37. Once the quadrature is known, the interphase forces are calculated in each grid cell using the disperse phase characteristic diameter evaluated by Eq. 16.

5.5 Algorithm of solution

The sequence of solution of the multi-phase code implemented in OpenFOAM is summarised next.

1. Solve the volumetric fraction for the disperse phases.
 - (a) Using the multi-phase code, Eq. 36.
 - (b) Using the CFD-PBE coupling, Eqs. 11, 12 and 37.
 - (b.1) Calculate the volumetric fraction and the characteristic diameter, Eqs. 17 and 16.
 - 1.1 Convergence loop for the volumetric fractions, back to step 3.
 - 1.2 Calculate the continuous phase fraction, $r_0 = 1 - \sum_{\alpha=1}^n r_{\alpha}$.
2. Evaluation of the drag coefficient using Schiller & Naumann (1933).
3. Discretisation of the phase momentum equations, l.h.s. of Eqs. 24 and 25.
4. PISO-loop.
 - 4.1 Calculate the $(\mathbf{A}_{\alpha})_D$ and $(\mathbf{A}_{\alpha})_H$ operators.
 - 4.2 Construction and solution of the pressure equation, Eq. 33.
 - 4.3 Update the volumetric phase fluxes, Eqs. 30 and 31 or 32.
 - 4.4 Update the phase velocities, Eqs. 27 and 28.
 - 4.5 Convergence loop for the pressure, back to step 6.

Usually, the user must define how many steps are necessary for the convergence of the volumetric fractions (step 1) and the coupling of pressure-velocity (step 4) loops. In contrast to this approach, a mixed absolute and relative tolerance, shown in Eq. 38, was implemented in this work to control and stop the convergence loop where the stop criteria must be set by the user.

$$\chi = \max \left[\frac{|\phi^{it} - \phi^{it-1}|}{1 + |\phi^{it}|} \right] \quad (38)$$

The `twoPhaseEulerFoam` programming was extended to handle $n + 1$ phases using the `PtrList<T>` C++ template to construct an array of classes or templates of type `T`. This template is a list of pointers used to locate the `T` classes allocated sequentially in the computer memory. Thus, the `PtrList<T>` template provides an easy access, storage and manipulation of the `T` classes array.

All the disperse phase variables, such as transport properties (density, kinematic viscosity and diameter), PBE variables and volumetric fractions, velocities and fluxes fields, were programmed as arrays with n elements using the `PtrList<T>` template. For instance, a single volumetric fraction is defined using a `volScalarField` template whereas the `PtrList<volScalarField>` declares a list of pointers which locates the address in memory of n `volScalarField` templates. In this case, each pointer will be addressed to each allocated volumetric phase `volScalarField`.

On the other hand, the continuous phase was not included in the `PtrList<T>` template array only to provide a better distinction of the continuous and n disperse phases in the code.

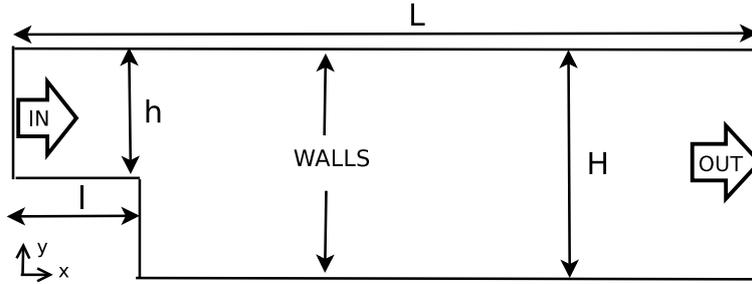


Figure 1: The 2D BFS with parametric dimensions and boundary patches.

Table 1: Physical properties and inlet conditions of the two-phase mixture.

Physical Properties	oil	water
ρ ($kg\ m^{-3}$)	900	1000
ν ($m\ s^{-2}$)	$1 \cdot 10^{-5}$	$1 \cdot 10^{-6}$
Variable	Inlet condition	
\mathbf{u}_{IN} ($m\ s^{-1}$)	1	1
d_α (μm)	-	50
r (-)	0.9	0.1

6 Numerical simulations

This section presents the results of a two-dimensional test case that was proposed to evaluate the performance of the implemented multi-phase code and the CFD-PBE solution in a multi-dimensional flow field with strong gradients. The well-known flow through a two-dimensional backward facing step (BFS) was chosen due to its simplicity and the presence of circulation zones with steep gradients in laminar flow. The BFS geometry, dimensions and boundary patches, shown in Fig. 1, was used by Silva *et al.* (2007). In the BFS geometry, $L = 11H$, $l = H$ and $h = H/2$, considering $H = 0.01\ m$ for all simulations.

A two-phase mixture consisting of a water in oil emulsion was used in the simulations (Silva *et al.* 2007). The physical properties of the liquid-liquid dispersion and the inlet conditions for the Sauter mean diameter and the global disperse phase fraction are shown in Table 1, being based on actual water-in-oil emulsions.

Silva *et al.* (2007) performed mesh convergence tests using meshes with 4000, 8000 and 16000 elements for hexahedral and tetrahedral meshes. In this work, only the finest hexaedral mesh was used in the simulations.

All the simulations were carried on using the implicit Crank Nicholson scheme for time integration with adaptive time step, which was controlled to keep the maximum mesh Courant number below 0.3. The advective terms were interpolated with the Gamma scheme with coefficient Γ equal 0.5 (Jasak 1996). The iterative convergence procedures were performed until the mixed error, shown in Eq. 38, achieved values lower than the specified tolerances of 10^{-8} and 10^{-7} for the pressure and volumetric fraction respectively. Since the main flow was laminar, no turbulence model was used.

The first simulations were performed to verify the implementation of the multi-phase code in OpenFOAM considering the emulsion flow and comparing its results with the two-phase solver. Next, simulations including one more disperse phase, denser and more viscous than the others, were tested using different characteristic diameters in order to

increase the drag force. Finally, the multi-phase CFD-PBE coupling was simulated for dominant breakage and dominant aggregation cases using simplified kernels (Silva *et al.* 2007).

6.1 Verification of the multi-phase code

The simpler way to verify the multi-phase code is comparing its simulation results using $n = 2$ with those obtained by the two-phase solver implemented in OpenFOAM. In addition, simulations using more phases can also be used to verify the multi-phase code when using the same transport properties for the n disperse phases and the phase volumetric fraction are equally set with the value obtained by the division of the global volumetric fraction by n . In this case, the fluidynamics and the global volumetric fraction of the disperse phase, $\sum_{\alpha=1}^n r_{\alpha}$, should present the same simulated results when using different number of phases.

The simulation of two different situations were analysed to verify the multi-phase code. In the first situation, the initial conditions of r_{α} and d_{α} into the domain were set as the same as the inlet, as shown in Table 1. On the contrary, these variables were set as zero into the domain for the second instance. These situations are respectively referred as cases I and II with 0.01 s and 0.1 s of simulation time.

Thus, simulations using $n = 2, 3$ and 5 were performed using respectively $r_{\alpha} = 0.1, 0.05$ and 0.025, and using the water transport properties shown in Table 1 for the n disperse phases. In addition, the gravity force was not considered in the simulation tests. The results shown in this section were extracted from a vertical line positioned at $x = 0.0125 m$ into the channel (crossing the BFS recirculation zone). The following results presents the comparison of the global volumetric fraction of the disperse phase and the pressure fields obtained using the `twoPhaseEulerFoam` code and the multi-phase code.

An excellent agreement on the solutions of the multi-phase code and the `twoPhaseEulerFoam` solver can be observed in Figs. 2(a) and (b). It is a clear evidence of a successful modeling and programming of the multi-phase approach. Since the whole domain is already filled with the emulsion mixture in case I, only minor variations of the global volumetric fraction caused by the recirculation flow should be noticed. As seen in Fig. 2(a), these small variations were accurately solved when using the multi-phase code. For case II, the two-phase emulsion is filling the domain and gradients on the volumetric fraction should appear. Similarly as case I, the variations of the global volumetric fraction were simulated accurately in case II.

The solution of the pressure vertical profiles in case II presented small deviations, as seen in Fig. 2(d). Compared with the `twoPhaseEulerFoam` solution, the order of magnitude of these deviations is almost 0.9% for $n = 5$ and, as observed in Fig. 2(d), it does not follow a pattern when increasing n . These differences are probably caused due to minor numerical errors during the solution of the pressure equation despite of the small convergence errors used in the simulation. Although, it does not affect the velocity field which magnitude is shown in Fig. 3 for case II.

6.2 Effects of increasing the drag force

In order to evaluate the capabilities of the multi-phase solver, a three-phase emulsion was considered including in the oil-water mixture a fictitious denser and more viscous liquid,

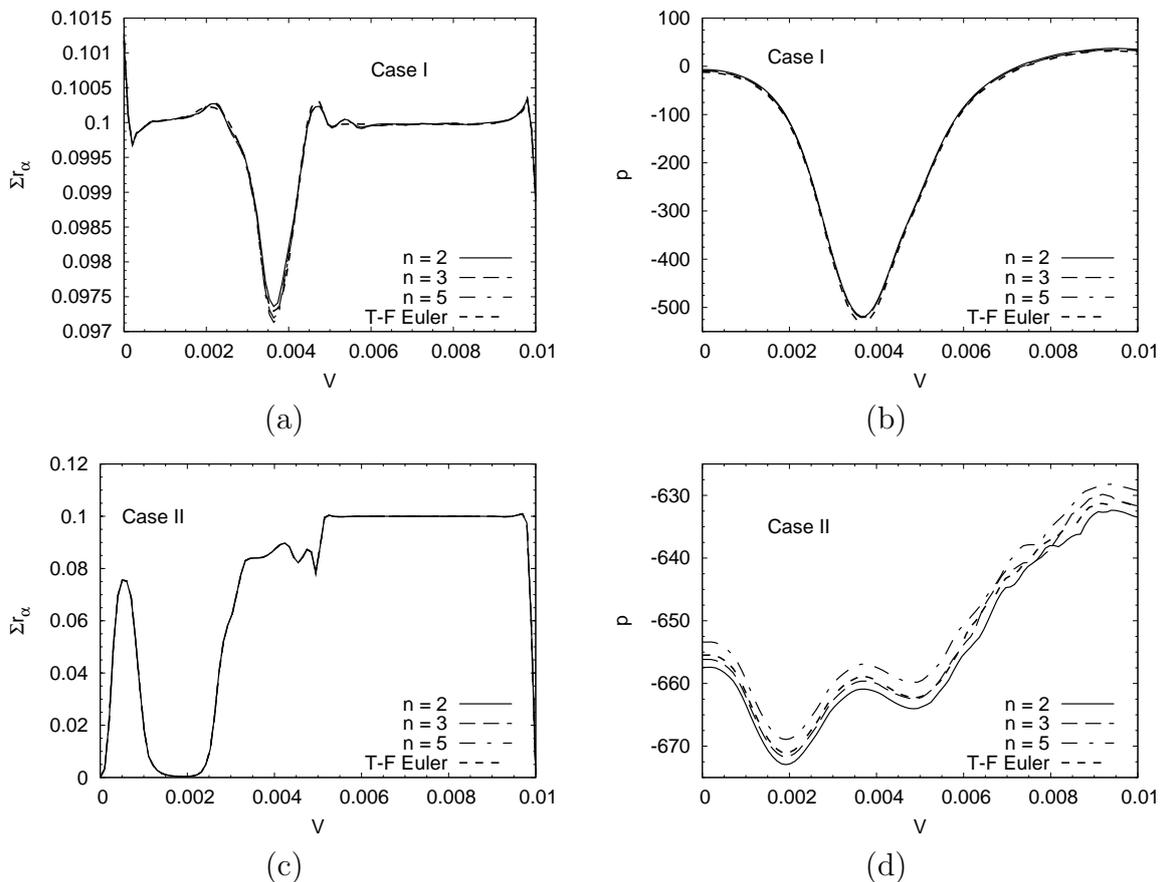


Figure 2: Vertical profiles of $\sum_{\alpha=1}^n r_\alpha$ and p for cases I, (a) and (b), and II, (c) and (d), using the `twoPhaseEulerFoam` code and the multi-phase code with different number of phases.

named phase L , which transport properties are $\rho = 3000 \text{ kg/m}^3$ and $\nu = 10^{-5} \text{ m/s}^2$. The simulations were conducted using $n = 5$ with the multi-phase solver and considering the oil as the continuous phase, which transport properties are in the Table 1, and the other as disperse phases. As in section 6.1, the volumetric fractions were set with the same value, $r_\alpha = 0.025$, for all the disperse phases. At this time, the subscripts 1 and 2 refer to the water and 3 and 4 refer to the L phases. In addition, each disperse phase has different characteristic diameter with values $d_1 = 40$, $d_2 = 80$, $d_3 = 120$ and $d_4 = 160 \text{ }\mu\text{m}$.

As in case II of section 6.1, the domain was empty in the simulations and the total simulation time was 0.1 s . In addition, the gravity force was now included in these simulations with -9.8 m/s^2 in the y axis direction. The contour plots of the volumetric fractions in a section of the domain are shown in Fig. 4.

As observed in Fig. 4, the water volumetric fractions suffer only minor changes when comparing the r_1 and r_2 contour plots. This is mainly caused due to the drag force against the continuous phase. On the other hand, the drag force on the L phase is more intense due to the higher values of the diameters and it affected the solutions of r_3 and r_4 differently. In fact, a settling of the denser phase occurs on the lower wall just before the expansion of the channel and below the greater recirculation zone. Moreover, due to its high viscosity the L phase is more resistant to be carried with the flow, and as consequence, it is not entrapped into the vortices as the water is.

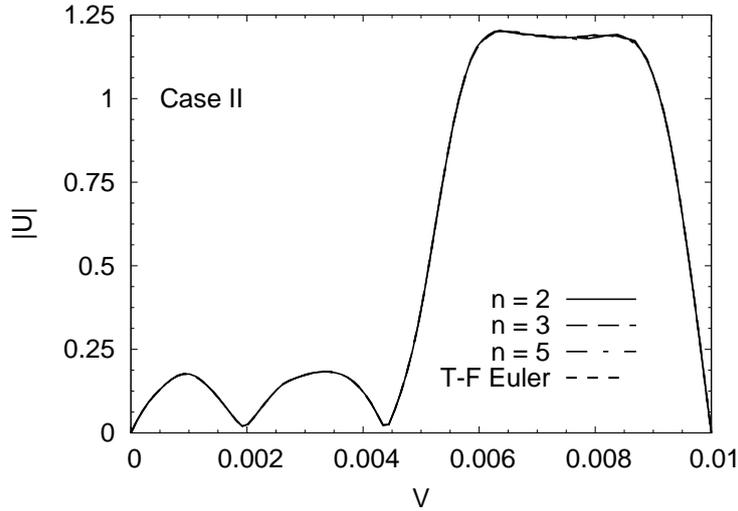


Figure 3: Vertical profiles of the velocity magnitude in case II using the `twoPhaseEuler-Foam` code and the multi-phase code with different number of phases.

6.3 CFD-PBE simulations

Finally, the two-phase emulsion was tested using the multi-phase with CFD-PBE coupling approach. The simulations performed in this work used the same conditions used by Silva *et al.* (2007). In this simulations, the breakup and coalescence between the water droplets was considered using the same non-physical models of the McCoy & Madras (2003),

$$a(v, v') = 1, b(v) = \frac{1}{2}\Phi(\infty)^2v, P(v | v') = \frac{1}{v'}, \vartheta(v) = 2 \quad (39)$$

where $\Phi(\infty)$ is the parameter which controls the breakage or aggregation dominance effect. As in Silva *et al.* (2007), the breakage and aggregation cases were simulated using respectively $\Phi(\infty) = 6, 0.1$. The multi-phase CFD-PBE cases were simulated with 5 phases and, as a result, the DQMOM formulation used 4 quadrature points.

The dimensionless weights and weighted abscissas were used in the simulations (Silva *et al.* 2007). Although, the volumetric fraction and the characteristic diameter obtained through the DQMOM variables were converted to the dimensional form to be used in the multi-phase momentum equations as shown in Silva *et al.* (2007). As in case I (section 6.1), the initial condition of the variables into the domain are the same as the inlet. The initial condition for the DQMOM variables were set by McCoy & Madras (2003) where the characteristic diameters were calculated using the dimensional abscissas (see Eq. 16), obtaining $d_1 = 33.125$, $d_2 = 57.478$, $d_3 = 78.018$ and $d_4 = 98.392 \mu m$.

The total simulation time for the breakage case was 0.1 s. The contour plot of the characteristic diameter and the volumetric fraction of phases 1 and 3 simulated with dominant breakage are shown in Fig. 5.

As seen in Fig. 5, the breakage case were successfully simulated where the particles are entrapped into the recirculation zones. In addition, the distribution of the particles are quite similar among phases 1 and 3 despite of the different phase sizes. The breakage in phase 3 is more intense than phase 1 since the breakage kernel, shown in Eq. 39, is linearly dependent with the particle size. On the other hand, the occurrence of phase 1 is higher than phase 3.

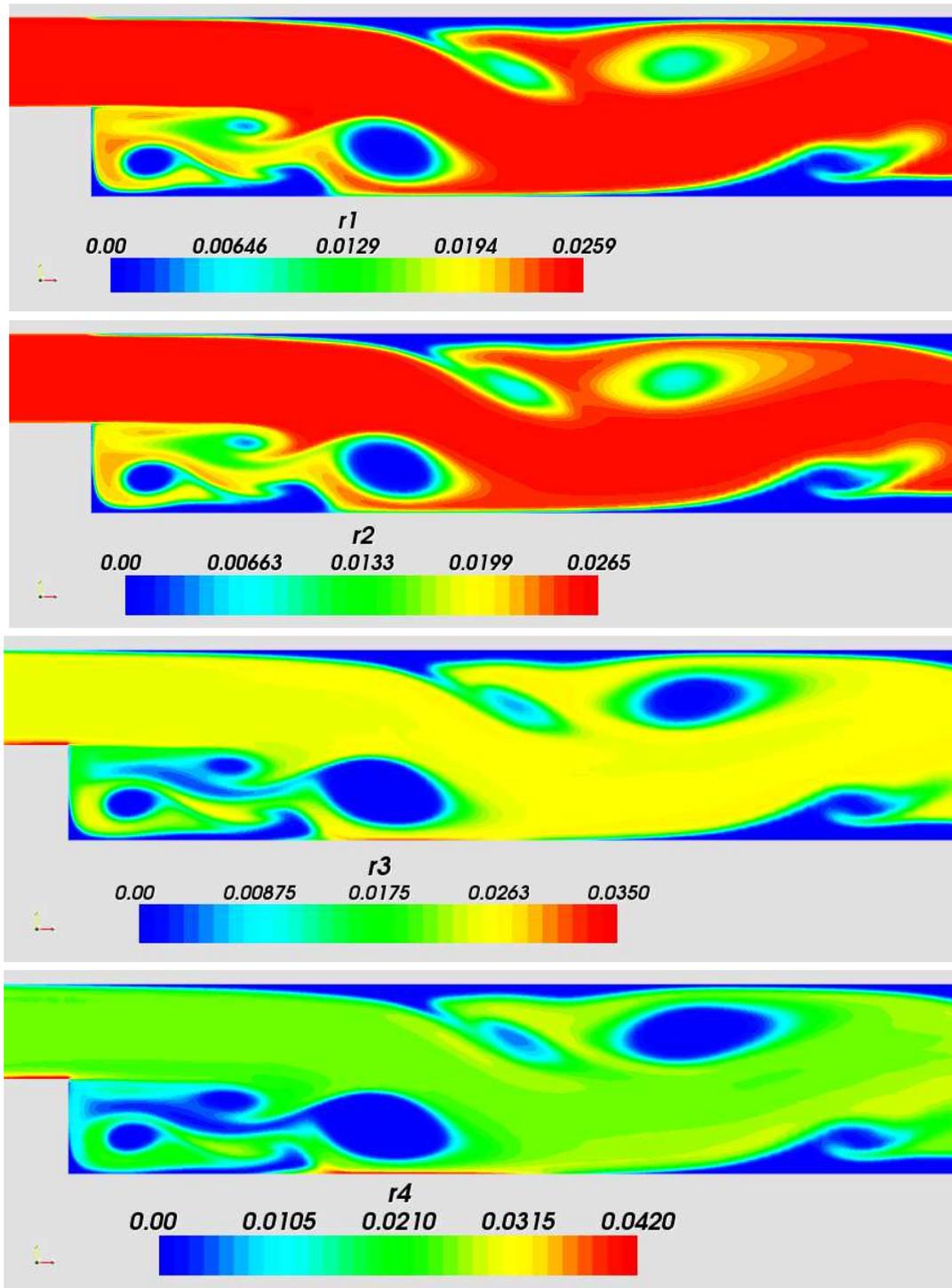


Figure 4: Countour plots of the disperse phases volumetric fractions.

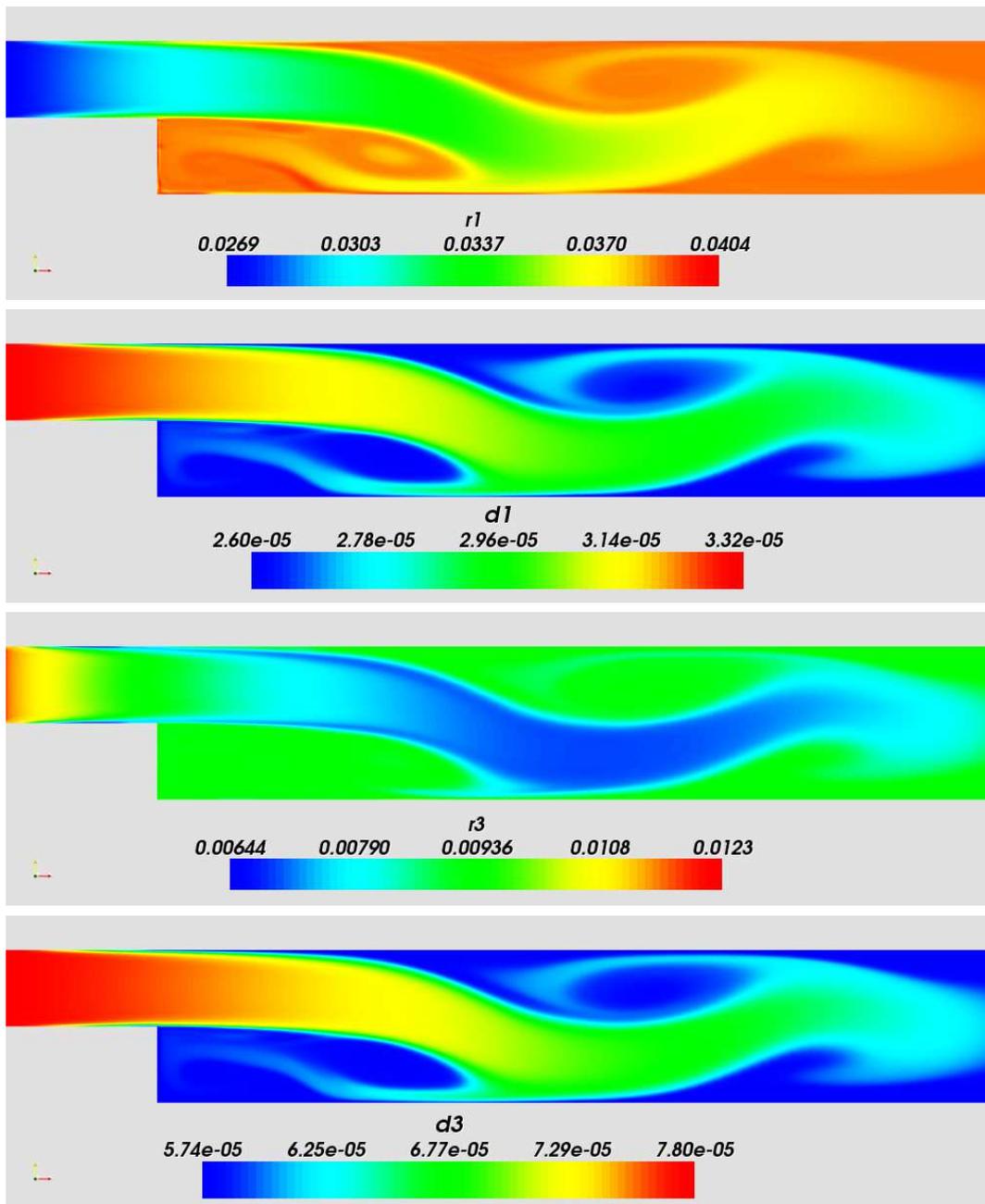


Figure 5: Contour plots of the volumetric fractions and characteristic diameters for phases 1 and 3 for the breakage case.

The aggregation rate is much slower than the breakage, thus a longer simulation time of 1 s was used. The diameter and volumetric fraction of phases 1 and 3 are shown in Fig. 6.

In this case, the flow reached the steady state and the recirculation zones are well defined, as seen in Fig. 6. The particles aggregates continually into the domain where the particle size distribution mainly depends on the residence time of the disperse phase.

In conclusion, the first simulation tests of the multi-phase CFD-PBE coupling code performed very well for the breakage and aggregation cases.

7 Conclusions

The formulation of the multi-phase model based on Rusche (2002) and the CFD-PBE coupling using DQMOM was presented in this work. The discretisation procedure and the pressure velocity coupling was formulated for the multi-phase model. As well, details of the algorithm and the programming of the multi-phase code in OpenFOAM are provided in this work.

The multi-phase code was tested simulating two-phase flows and comparing the its simulated results with the OpenFOAM two-phase solver. The multi-phase code was simulated using 2, 3 and 5 equal disperse phases. The agreement of the solutions using the different approachs validated the multi-phase programming.

The effect of the drag force was evaluated considering a three-phase liquid system with different diameters. The density and viscosity differences were noticed in the simulations where the denser phase settled in the lower wall of the geometry.

The first simulation tests using the coupled CFD-PBE multi-phase approach implemented in OpenFOAM were successfully done. These implementations are reliable enough to be applied in actual flow simulations with realistic breakup and aggregation models. As noticed by Silva *et al.* (2007), no strong difficult were found in solving the hyperbolic DQMOM transport equations when using different velocity fields.

Acknowledgements

The authors would like to thank CNPq (grant no. 301548/2005-6). L.F.L.R. Silva would also like to acknowledge the financial support given by Chemtech.

References

- Bove, S. Computational fluid dynamics of gas-liquid flows including bubble population balances. PHD Thesis, Esbjerg Institute of Engineering, Denmark (2005)
- Drew, D.A. & Passman, S.L. Theory of Multicomponent Fluids. Springer, 1st Ed. (1998)
- Ishii, M. Thermo-fluid Dynamic Theory of Two-phase Flow. Eyrolles, Paris (1975)
- Jasak, H. Error analysis and estimation for the finite volume method with applications to fluid flows. PHD Thesis, Imperial College of Science, Technology and Medicine, UK (1996)

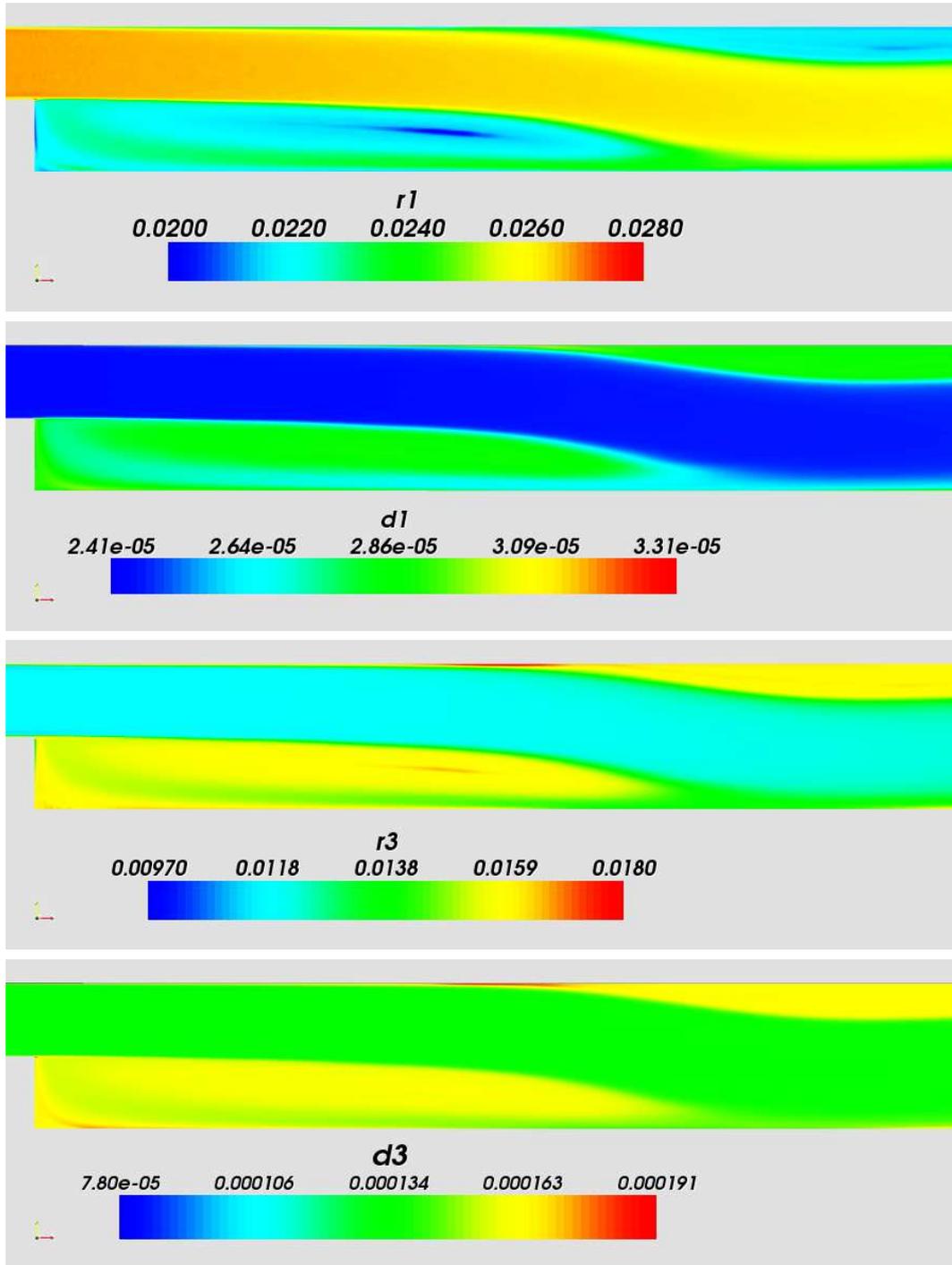


Figure 6: Countour plots of the volumetric fractions and characteristic diameters for phases 1 and 3 for the dominant aggregation case.

- Marchicio, D.L. & Fox, R.O. Solution of the population balance equation using the direct quadrature method of moments. *Journal of Aerosol Science*, Volume 36, 43 – 73 (2005)
- McCoy, B.J. & Madras, G. Analytical solution for a population balance equation with aggregation and fragmentation. *Chemical Engineering Science*, Volume 58, 3049 – 3051 (2003)
- Ramkrishna, D. *Population Balances – Theory and Applications to Particulate Systems in Engineering*. Academic Press, New York (2000)
- Rusche, H. Computational fluid dynamics of dispersed two-phase flows at high phase fractions. PHD Thesis, Imperial College of Science, Technology and Medicine, UK (2002)
- Schiller, L. & Naumann, A. Über die grundlegenden berechnungen bei der schwerkraftbereitung. *Z. Vereins deutscher Ing.*, Volume 77, Number 12, 318 – 320 (1933)
- Silva, L.F.R., Damian, R.B. & Lage, P.L.C. Implementation and analysis of numerical solution of the population balance equation in CFD packages. *International Conference on Multiphase Flow*, available on CD (2007)
- Weller, H.G., Tabor, G., Jasak, H. & Fureby, C. A tensorial approach to continuum mechanics using object-oriented techniques. *Computers in Physics*, Volume 12, Number 6, 620 – 631 (1998)
- Weller, H.G. Derivation, modelling and solution of the conditionally averaged two-phase flow equations. Technical Report TR/HGW/02, Nabla Ltd. (2002)