



Universidade Federal de Pernambuco
Centro de Tecnologia e Geociências
Pós-graduação em Engenharia Mecânica

Dissertação de Mestrado

Simulação de Reservatórios de Petróleo no Ambiente OpenFOAM

Rafael Cabral de Moura

Orientador: Ramiro Brito Willmersdorf, Ph. D.
Co-Orientador: Paulo Roberto Maciel Lyra, Ph. D.

Recife
Fevereiro de 2012

Universidade Federal de Pernambuco
Centro de Tecnologia e Geociências
Pós-graduação em Engenharia Mecânica

Dissertação de Mestrado

Simulação de Reservatórios de Petróleo no Ambiente OpenFOAM

Rafael Cabral de Moura

Trabalho apresentado ao Programa de Pós-Graduação em Mecânica Computacional do Departamento de Engenharia Mecânica da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Mestre em Engenharia Mecânica.

Orientador: Ramiro Brito Willmersdorf, Ph. D.
Co-Orientador: Paulo Roberto Maciel Lyra, Ph. D.

Recife
Fevereiro de 2012

Catálogo na fonte
Bibliotecária Margareth Malta, CRB-4 / 1198

M929s Moura, Rafael Cabral de.
Simulação de reservatórios de petróleo no ambiente OpenFOAM /
Rafael Cabral de Moura. - Recife: O Autor, 2012.
vii, 71 folhas, il., gráfs., tabs.

Orientador: Prof. Dr. Ramiro Brito Willmersdorf.
Co-Orientador: Prof. Dr. Paulo Roberto Maciel Lyra.
Dissertação (Mestrado) – Universidade Federal de Pernambuco. CTG.
Programa de Pós-Graduação em Engenharia Mecânica, 2012.
Inclui Referências Bibliográficas.

1. Engenharia Mecânica. 2. Meios porosos. 3. Escoamento multifásico.
4. IMPES. 5. OpenFOAM. 6. Volumes finitos. 7. Programação paralela. I.
Willmersdorf, Ramiro Brito. (Orientador). II. Lyra, Paulo Roberto Maciel.
(Co-Orientador). III. Título.

UFPE

621 CDD (22. ed.)

BCTG/2012-169

Simulação de Reservatórios de Petróleo no Ambiente OpenFOAM

Rafael Cabral de Moura

Trabalho apresentado ao Programa de Pós-Graduação em Mecânica Computacional do Departamento de Engenharia Mecânica da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Mestre em Engenharia Mecânica.

Aprovada por:

Ramiro Brito Willmersdorf Ph. D.
(Orientador)

Paulo Roberto Maciel Lyra Ph. D.
(Co-Orientador)

Darlan K. E. de Carvalho Dr.
(Examinador Interno)

Rogério Soares da Silva Dr.
(Examinador Externo: CAA-UFPE)

Bernardo Horowitz Ph. D.
(Examinador Externo: DECIV-UFPE)

Recife
Fevereiro de 2012

À Andelson Cabral de Moura.

“A simplicidade é o último degrau da sabedoria.”

Khalil Gibran

Agradecimentos

Agradeço primeiramente a Deus, que me deu a oportunidade de estar entre grandes pessoas e realizar muitos de meus sonhos. Agradeço ao professor Ramiro, por ter paciência comigo e sempre me mostrar, com muita calma, o caminho certo a seguir. A minha esposa por me ajudar muito e me dar apoio nos momentos mais decisivos de minha vida. Aos amigos do LabCom, que sempre juntos discutem e resolvem a maioria dos problemas impostos, contribuindo também com críticas ao trabalho. Agradeço ao meu amigo Bruno de Jesus que muito me ajudou durante os meus trabalhos. Por último, agradeço à minha família que acreditou em mim e sempre me encorajou a continuar com meus estudos.

Resumo

Neste trabalho foi implementado um programa para simulação computacional de reservatórios de petróleo, baseado no ambiente OpenFOAM®, que é um sistema de desenvolvimento para programação científica orientado a objetos, paralelo, de alto nível e com alto desempenho. Esse ambiente de desenvolvimento oferece grande flexibilidade quanto à escolha dos métodos de discretização, interpolação e solução dos sistemas resultantes. O OpenFOAM® é uma biblioteca da linguagem C++ que, através de suas várias ferramentas, pode ser usada para a solução de problemas envolvendo equações diferenciais parciais. Para a solução o ambiente emprega o método dos volumes finitos, usando malhas estruturadas ou não estruturadas. Neste trabalho foram usadas apenas malhas estruturadas. O simulador desenvolvido trata problemas descritos pelo modelo de escoamento bifásico imiscível água-óleo que é descrito por um sistema de equações diferenciais parciais altamente não-lineares, obtidas através da conservação da massa para cada fase, e do uso da Lei de Darcy para relacionar fluxos de massa com gradientes de potencial fluido. Para a integração temporal das equações é empregado o método IMPES, no qual o sistema composto pela equação de conservação de massa e pela Lei de Darcy é reescrito em termos de uma equação para a pressão de óleo, que é resolvida implicitamente, e uma equação para a saturação de água, que é resolvida explicitamente.

Palavras-chave: Meios Porosos, Escoamento Multifásico, IMPES, OpenFOAM, Volumes Finitos, Programação Paralela.

Abstract

In this work a program for computer simulation of oil reservoirs was implemented. The program was based on OpenFOAM® framework, which is a development system for scientific object-oriented programming, parallel, high-level and high performance computation. This development framework provides great flexibility in the choice of discretization and interpolation methods and for solving the resulting systems. The OpenFOAM® is a library of C++ language which, through its various tools, can be used to solve problems involving partial differential equations. The OpenFOAM® framework employs the finite volume method, using structured or unstructured meshes. In this work only structured meshes were used. The simulator was developed for problems described by the model of immiscible two-phase flow of water and oil is which described by a system of highly nonlinear partial differential equations, obtained by considering conservation of mass of each phase, and the use of Darcy's Law to relate mass flows with gradients of fluid potential. For the temporal integration of the equations the IMPES method is employed. The system composed of the mass conservation equation and Darcy's Law is rewritten in terms of an equation for the oil pressure, which is solved implicitly, and an equation for water saturation, which is solved explicitly.

Keywords: Porous Medium, Multiphase Flow, IMPES, OpenFOAM, Finite Volume, Parallel Programming.

Sumário

1	Introdução	3
1.1	Motivação	4
1.2	Objetivos	5
1.3	Organização da dissertação	6
2	OpenFOAM	7
2.1	Programação no OpenFOAM	8
2.2	Estrutura do OpenFOAM	9
2.3	Aplicações	10
2.3.1	Difusão de calor	12
2.3.2	Advecção linear	13
2.4	Casos	15
3	Equações Governantes	20
3.1	Propriedades da rocha e dos fluidos	21
3.2	Lei de Darcy	24
3.3	Lei da conservação de massa	25
3.4	Escoamento monofásico	27
3.5	Escoamento bifásico	28
4	Modelagem Computacional	36
4.1	Discretização do domínio	36
4.2	Discretização das equações	37
4.2.1	Discretização dos operadores de difusão	38
4.2.2	Discretização dos operadores de advecção	39
4.2.3	Discretização temporal	41
4.3	Método IMPES	42
4.3.1	Tratamento das não linearidades	43
5	Aplicação IMPES	45
5.1	Criação da aplicação	46

5.2	Criação do caso	50
6	Resultados	53
6.1	Resultados da aplicação de escoamento monofásico	53
6.2	Verificação da aplicação IMPES	54
6.2.1	Escoamento unidimensional	55
6.2.2	Escoamento bidimensional	57
6.2.3	Escoamento tridimensional	60
6.2.4	Caso heterogêneo bidimensional	63
6.2.5	Caso heterogêneo tridimensional	65
7	Conclusões e Trabalhos Futuros	66
7.1	Conclusões	66
7.2	Trabalhos futuros	66
	Bibliografia	68

Lista de Figuras

2.1	Estrutura do ambiente OpenFOAM [4].	9
2.2	Esquema do caso de difusão.	15
2.3	Estrutura do diretório de um caso.	16
2.4	Malha da discretização do problema de calor.	19
2.5	Solução do caso com o problema de calor.	19
3.1	Aplicação dos modelos de fluxo. Modificado de [20] <i>apud</i> [29].	21
3.2	Comportamento do fator volume de formação.	23
3.3	Comportamento da viscosidade.	24
3.4	Esquema de fluxo.	26
3.5	Curvas de permeabilidade relativa.	30
3.6	Curvas de pressão capilar [29].	31
3.7	Saturação na zona de transição [29].	32
4.1	Volume de controle.	37
4.2	Tipos de volume de controle do OpenFOAM.	38
4.3	Vetores \vec{S} e \vec{d} para uma malha não ortogonal [25].	39
4.4	Interpolação linear [25].	40
4.5	Representação de duas células vizinhas.	44
5.1	Caso que utiliza a aplicação IMPES.	50
5.2	Esquema <i>five-spot</i>	51
5.3	Ajuste da trajetória de um poço direcional.	52
6.1	Solução de pressão do caso monofásico.	54
6.2	Malha usada no problema unidimensional.	56
6.3	Comparação do perfil de pressão 1D.	56
6.4	Comparação do perfil de saturação 1D.	57
6.5	Malha usada no problema bidimensional.	58
6.6	Solução da pressão no problema bidimensional.	58
6.7	Comparação das soluções de pressão 2D na direção diagonal. .	59
6.8	Solução da saturação do problema bidimensional.	59

6.9	Comparação das soluções de saturação 2D na direção diagonal.	60
6.10	Malha usada no problema tridimensional.	60
6.11	Solução da saturação do problema tridimensional.	61
6.12	Comparação das soluções de saturação 3D na direção diagonal.	61
6.13	Solução da pressão do problema tridimensional no IMEX. . . .	62
6.14	Solução da pressão do problema tridimensional no OpenFOAM.	62
6.15	Comparação das soluções de pressão 3D na direção diagonal. .	63
6.16	Mapa de permeabilidades do caso bidimensional.	63
6.17	Solução do caso heterogêneo bidimensional.	64
6.18	Permeabilidades do caso tridimensional.	65
6.19	Permeabilidades do caso tridimensional.	65

Lista de Tabelas

2.1	Padrão de unidades no OpenFOAM.	13
3.1	Fatores de conversão de unidades.	25
5.1	Listagem dos campos criados na aplicação.	47
6.1	Propriedades do fluido (Caso monofásico).	53
6.2	Propriedades da rocha (Caso monofásico).	54
6.3	Propriedades da rocha (Caso bifásico).	55
6.4	Propriedades dos fluidos (Caso bifásico).	55
6.5	Curvas de permeabilidade relativa e pressão capilar.	55

Capítulo 1

Introdução

A modelagem computacional do escoamento multifásico em meios porosos é uma das principais ferramentas no gerenciamento de reservatórios de petróleo. Através do modelo de fluxo de um reservatório o engenheiro que o gerencia é capaz de prever o comportamento das pressões e fazer uma estimativa da produção futura de óleo. Com a previsão do comportamento da vazão de óleo ao longo do tempo e com uma estimativa do preço futuro do barril de óleo é feita uma análise econômica que determina por quanto tempo o reservatório continuará a produzir e a viabilidade, ou não, da implantação de projetos no campo. Assim como a modelagem computacional, também podem ser empregados outros métodos, como o balanço de materiais abordado em [17], [16], [24] e a análise por curvas de declínio de produção proposta inicialmente em [8] e posteriormente abordada por [31], [22] e [21]. Através das estimativas de produção feitas pelos engenheiros de reservatórios e das datas de abandono de cada campo tem-se o valor das reservas de uma empresa de petróleo. Através de simuladores de fluxo pode-se estimar o ganho que obtém-se com métodos de recuperação secundária de petróleo, como injeção de água [18] [15], vapor [23], polímeros [11] [32], bactérias [10], surfactantes [37] e métodos de combustão *in-situ*. O posicionamento dos poços e os volumes de água e vapor a serem injetados são alguns dos parâmetros operacionais buscados ao se fazer uma simulação de um reservatório de petróleo. Para cada método de recuperação têm-se uma modelagem associada e através de simuladores desenvolvidos para cada método é possível construir uma previsão do comportamento dos reservatórios.

Neste trabalho foi desenvolvido um simulador de escoamento bifásico em meios porosos no ambiente OpenFOAM. O simulador usa como base o ambiente OpenFOAM (Open Field Operator and Manipulator) que é uma biblioteca escrita em C++ com ferramentas para a solução de problemas de volumes finitos e elementos finitos. O desenvolvimento de um programa para

a simulação de um determinado fenômeno físico nesse sistema se dá por meio de aplicações que usam as classes existentes no ambiente. O ambiente possui aplicações prontas para a maioria dos fenômenos de dinâmica dos fluidos, inclusive para alguns casos envolvendo meios porosos. Antes da implementação do simulador foram feitas aplicações no sistema para a solução de problemas mais simples com o objetivo de facilitar o aprendizado. Uma das dificuldades no uso desse ambiente é a pouca documentação disponível para estudo, desta forma, a reconstrução de programas para a solução de problemas simples facilitou o aprendizado.

Inicialmente, foram feitas aplicações para a solução de problemas envolvendo equações parabólicas, hiperbólicas e elípticas. A partir destes programas que envolvem os tipos básicos de equações, pode-se desenvolver programas para solucionar casos que envolvam uma combinação desses fenômenos. Nesse ambiente OpenFOAM, uma vez desenvolvida uma aplicação tem-se um *solver* genérico, cujos esquemas de discretização, interpolação e métodos para a solução do sistema de equações resultante serão especificados em um diretório com as características do caso a ser simulado. Os diretórios contendo os casos que usam as aplicações devem ser organizados segundo o padrão do sistema. As aplicações desenvolvidas usando as classes do OpenFOAM modelam problemas tridimensionais mas podem ser usadas para resolver casos unidimensionais e bidimensionais. Desta forma, uma aplicação é escrita de uma forma genérica, podendo-se mudar o número de dimensões de acordo com o caso estudado. Em cada caso que use uma aplicação, pode-se mudar o número de dimensões do problema, a geometria, as condições de contorno e os métodos de discretização. Assim cada aplicação pode ser utilizada de diferentes modos o que facilita a verificação de qual combinação de esquemas resolve melhor um determinado problema. As primeiras aplicações que foram resolvidas tratam de problemas de condução de calor em regime permanente, em regime transiente e problemas de advecção. Como etapa inicial no desenvolvimento do simulador de reservatórios foi feita uma aplicação para a solução do caso de escoamento monofásico em meios porosos. Neste problema é resolvida apenas a equação de difusão que é similar a equação de pressão que é resolvida no simulador bifásico. A solução deste tipo de problema foi importante na verificação das técnicas empregadas no simulador bifásico.

1.1 Motivação

O OpenFOAM é uma biblioteca escrita na linguagem C++ que contém ferramentas para a solução de problemas da mecânica do contínuo, para modelagem de reações químicas, matemática financeira e qualquer problema que

possa ser descrito através de equações diferenciais parciais e ordinárias. Com alguma experiência no sistema pode-se desenvolver aplicações com rapidez e flexibilidade. O OpenFOAM é um sistema naturalmente paralelo e a implementação de suas aplicações é feita através de métodos e objetos de alto nível criados com classes que são escritas em paralelo, o que é mais uma grande vantagem no uso desse sistema, visto que, a implementação de programas paralelos é complexa e requer muito tempo dependendo do problema a ser resolvido. O modelo bifásico é bastante utilizado na modelagem problemas de injeção a fim de se determinar a eficiência de diferentes configurações de disposição dos poços injetores e produtores. A implementação de um simulador de reservatórios nesse ambiente possibilita a investigação da aplicação das diversas ferramentas numéricas do sistema na solução deste tipo de problema.

1.2 Objetivos

Este trabalho teve como objetivo geral a implementação de um programa para a simulação computacional de reservatórios de petróleo, baseado em um ambiente de desenvolvimento para programação científica orientado a objetos, paralelo, de alto nível e com alto desempenho. O simulador trata de problemas descritos pelo modelo de escoamento bifásico e imiscível óleo-água. Para alcançar o objetivo geral, as seguintes metas intermediárias foram definidas:

- Estudo da formulação matemática envolvida;
- Investigação e escolha do método numérico e das linearizações a serem usadas no problema;
- Estudo do funcionamento do ambiente OpenFOAM;
- Desenvolvimento de aplicações no ambiente para a solução de problemas modelo de advecção e difusão;
- Desenvolvimento de uma aplicação para solução do problema de escoamento monofásico em meios porosos;
- Desenvolvimento da aplicação para a solução do problema de escoamento bifásico em meios porosos.

1.3 Organização da dissertação

Este trabalho está dividido em sete capítulos e uma bibliografia incluindo-se a introdução. Os capítulos estão dispostos segundo a sequência de trabalho realizada para atingir o objetivo geral do trabalho. Os capítulos abordam os seguintes temas:

- *Capítulo 1: Introdução.* No presente capítulo é feita uma breve introdução ao objetivo de se usar um simulador de reservatórios seguida de um resumo da metodologia empregada na construção do simulador feito neste trabalho. São apresentadas as características do sistema que motivaram a sua escolha e os objetivos que se esperavam atingir ao término do trabalho.
- *Capítulo 2: OpenFOAM.* São apresentadas algumas das características do ambiente de desenvolvimento utilizado bem como alguns exemplos de programas desenvolvidos no sistema e algumas soluções encontradas.
- *Capítulo 3: Equações Governantes.* No terceiro capítulo se encontram as equações matemáticas dos modelos que regem o escoamento multifásico em reservatórios de petróleo e uma breve descrição das propriedades da rocha e dos fluidos presentes.
- *Capítulo 4: Modelagem Computacional.* Mostra a forma como o OpenFOAM faz a discretização dos principais operadores diferenciais abordando, de maneira simplificada, como o sistema faz a interpolação das propriedades através dos elementos da malha gerada.
- *Capítulo 5: Aplicação IMPES.* Apresenta a aplicação desenvolvida no sistema, com trechos do código explicando o seu funcionamento.
- *Capítulo 6: Resultados.* São descritos alguns resultados e é feita uma comparação entre o simulador desenvolvido e um programa comercial.
- *Capítulo 7: Conclusões e Trabalhos Futuros.* Traz uma breve conclusão do trabalho e algumas propostas para possíveis trabalhos futuros.

Capítulo 2

OpenFOAM

O OpenFOAM [26] [5] é uma biblioteca de código aberto escrita em C++ desenvolvida para utilização no sistema Linux mas que pode ser utilizada em outros sistemas operacionais. Essa biblioteca contém *namespaces* com classes que possuem métodos para a manipulação de campos escalares, vetoriais e tensoriais e para a discretização de operadores diferenciais como gradiente, divergente e laplaciano. Estas classes definem objetos como matrizes, malhas, arquivos de controle de simulação, campos dimensionais e adimensionais associados a malha ou não e muitos outros objetos relacionados com as operações envolvidas na programação científica. A criação de programas usando essa biblioteca é de alto nível visto que a maioria dos operadores aritméticos estão sobrecarregados para operações com os objetos que representam os campos, pode-se então criá-los com pouco uso da linguagem C++ devido ao uso das sobrecargas. Os objetos que representam os campos e operadores sobrecarregados estão todos escritos de forma paralela, desse modo, é possível criar programas paralelos programando-se serialmente.

Na implementação de programas para a solução numérica de equações diferenciais parciais é comum que se adote, antes, uma série de esquemas e técnicas para a solução do problema proposto. No final da implementação tem-se uma aplicação que resulta da combinação formada pelos métodos e técnicas numéricas escolhidas, qualquer variação dessa combinação de esquemas requer uma reprogramação no código fonte. A orientação a objetos tem como objetivo o encapsulamento dos dados, a modularização e o reaproveitamento do código. Os programas trabalham com a criação de objetos que pertencem a classes com características distintas. Pode-se pensar em um programa de solução numérica de equações diferenciais parciais (EDP's) como uma classe, na qual seus objetos derivados são combinações dos métodos numéricos escolhidos e com essas formas genéricas pode-se construir o aplicativo. Um programa de solução numérica de EDP's, que usa dife-

renças finitas, por exemplo, necessita de um método de discretização, uma formulação a ser usada e um esquema de discretização. Como método de discretização pode-se ter diferenças finitas, volumes finitos ou elementos finitos e a formulação pode ser implícita ou explícita ou ainda uma combinação entre ambas. O esquema de discretização pode ser, no caso das diferenças finitas uma combinação de derivadas para frente, para trás e centradas. A solução numérica do sistema de equações resultante, pode ser feita através de diferentes métodos como: gradientes conjugados, gradiente bi-conjugado preconditionado (Bi-CGStab) ou uma solução usando multigrid [3].

O OpenFOAM, oferece a possibilidade de criar, através de classes específicas, programas que possuem uma forma genérica nos quais pode-se facilmente mudar a combinação de métodos numéricos desejada, isso dá ao sistema uma grande flexibilidade para a implementação de programas de solução numérica de EDP's. Arquivos de entrada controlam quais métodos serão usados em cada parte da aplicação, sem a necessidade de reescrever o código fonte. O acoplamento de diferentes módulos aos programas, de forma independente do código já implementado, pode ser feito de maneira simples e rápida. O próprio ambiente oferece muitas ferramentas para serem incorporadas a aplicação desenvolvida. Ferramentas de geração adaptativa de malhas, movimentação de malhas e módulos de solução dos sistemas resultantes com o uso de GPU's [6] encontram-se disponíveis para o ambiente atualmente e alguns trabalhos já foram publicados com o uso dessa ferramenta [30]. A biblioteca possui muitos outros simuladores implementados que estão disponíveis para uso.

2.1 Programação no OpenFOAM

Existem três níveis de utilização do OpenFOAM [2]. A biblioteca contém programas com diferentes aplicações dos métodos, que implementam programas para a solução de problemas de dinâmica dos fluidos computacional, química, deformação dos sólidos e economia. Pode-se utilizar esses programas prontos ou desenvolver programas aplicando os objetos e classes da biblioteca e ainda pode-se escrever outras classes. Os programas que acompanham o sistema são chamados de aplicações do OpenFOAM, *OpenFOAM applications* [4], ou *solvers*, e os programas desenvolvidos pelo usuário são chamados de aplicações do usuário.

As aplicações são utilizadas para resolver problemas físicos que, segundo a nomenclatura do sistema, são chamados de casos. Por exemplo, na biblioteca temos um simulador de fluxo de calor, o qual é chamado de aplicação para problemas de calor, um problema de fluxo de calor ao longo de uma barra retangular constitui um caso que será resolvido com essa aplicação. Neste tra-

balho foi desenvolvida uma aplicação para problemas de escoamento bifásico imiscível óleo-água em meios porosos, os casos simulados são reservatórios sujeitos à injeção de água. Nas aplicações são resolvidas as equações que modelam o fenômeno físico relacionado, nos casos são descritas as condições iniciais e de contorno e é definida a geometria do problema estudado. Ainda nos casos estão contidos os arquivos que controlarão o comportamento das classes das aplicações modificando o funcionamento das mesmas. Esse fluxo de trabalho do sistema permite que com uma mesma aplicação seja possível resolver vários problemas com diferentes geometrias, parâmetros de simulação e esquemas numéricos através da mesma aplicação sem ser necessária a implementação de um programa diferente para cada caso.

2.2 Estrutura do OpenFOAM

O ambiente OpenFOAM está estruturado segundo o esquema apresentado na Figura 2.1, na qual as aplicações padrão representam os simuladores escritos no ambiente OpenFOAM e que acompanham o sistema. Cada aplicação pode fazer uso das utilidades do sistema que são rotinas escritas para a manipulação de dados, como por exemplo, leitura e escrita de arquivos, criação de gráficos e estatísticas de execução do programa.

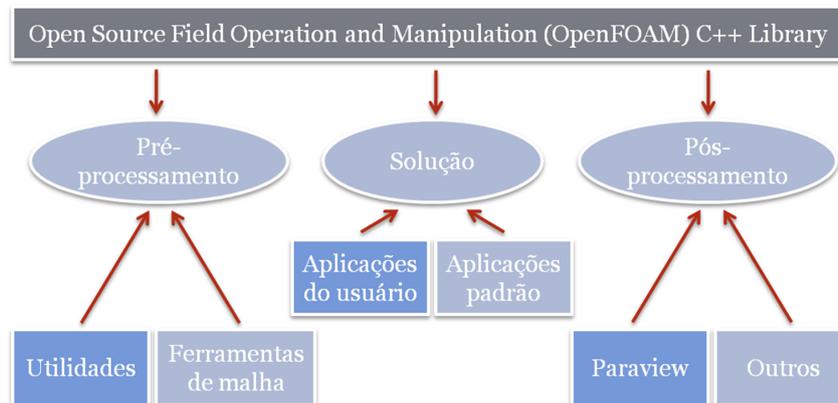


Figura 2.1: Estrutura do ambiente OpenFOAM [4].

No sistema existe um gerador de malhas chamado *blockmesh* restrito apenas a geração de malhas por blocos, no entanto, a geração de malhas pode ser feita em outro sistema, como o GMESH, e importada para o formato nativo através de utilidades desenvolvidas para a conversão entre os diversos formatos existentes. As utilidades em conjunto com as ferramentas de manipulação de malha compõem o módulo de pré-processamento do sistema.

Através das classes podem ser implementadas novas aplicações, as quais enquadram-se como aplicações do usuário, que podem fazer uso das utilidades do sistema assim como as aplicações padrão. No sistema existem aplicações padrão para a solução de problemas de difusão de calor, escoamento multifásicos, escoamento sujeito à reações químicas e escoamento supersônico, dentre outros. Os programas implementados pelo usuário e as aplicações padrão formam o módulo de solução do ambiente.

A visualização dos dados de simulação no sistema é obtida através do pacote de pós processamento, que basicamente é composto pelo programa ParaView que é distribuído junto à biblioteca. Outros visualizadores podem ser utilizados já que o OpenFOAM oferece ferramentas para exportar do formato nativo para a maioria dos formatos atualmente utilizados. Por ter código aberto, facilmente pode ser implementada uma rotina para converter os dados de simulação para um formato que o sistema não reconheça. Uma das maiores vantagens é a possibilidade de criar utilidades e aplicações no ambiente.

2.3 Aplicações

Aplicações são programas escritos em C++ que utilizam as classes definidas no OpenFOAM. Nelas são representadas as equações que modelam o fenômeno físico estudado. Essas equações geram uma matriz de volumes finitos que resulta da discretização de cada termo representado. Um operador diferencial no OpenFOAM é descrito por meio de um método, por exemplo para representar o operador de divergência usamos o método *div()*. Cada uma das equações tem seus operadores diferenciais discretizados de acordo com um esquema. Desta forma, na aplicação não se descreve como cada operador será discretizado. No OpenFOAM os volumes de controle são centrados nas células e alguns dos operadores de discretização realizam operações com os valores dos campos definidos nas faces dos elementos, desta forma é necessário interpolar os valores do centro das células para as faces. Essa interpolação é feita na aplicação sem especificar o método de interpolação usado, apenas pela chamada de um método que converte um campo dos centros para as faces. Uma vez desenvolvida uma aplicação tem-se um *solver* genérico, cujos esquemas de discretização, interpolação e métodos para a solução do sistema de equações resultante serão especificados em um diretório com as características do caso a ser simulado. Após a especificação dos esquemas de discretização têm-se então a formulação desejada para a solução da equação. Adiante, serão mostradas as representações de equações e como funciona a escolha dos métodos de discretização e interpolação.

O OpenFOAM é um ambiente de programação de alto nível orientado à objetos, os operadores matemáticos estão todos sobrecarregados no sistema, de modo que se pode realizar operações sobre estruturas de dados paralelas como se estivesse fazendo uma simples operação aritmética. As classes implementadas no sistema permitem que, em suas aplicações, sejam representadas de forma tensorial, as equações do fenômeno físico que se deseja modelar, como na Equação (2.1).

$$\frac{\partial(\phi)}{\partial t} + \nabla \cdot (U\phi) - \nabla \cdot (\Gamma_\phi \nabla \phi) = S_\phi \quad (2.1)$$

A equação (2.1) é típica de um problema de convecção-difusão que pode ser representada em uma aplicação do OpenFOAM da seguinte maneira:

```

1 fvScalarMatrix PhiEqn
2 (
3     fvm::ddt(phi)
4     + fvc::div(U, phi)
5     - fvc::laplacian(Gamma, phi)
6     ==
7     S
8 )

```

A representação dá-se através da chamada das classes como *fvScalarMatrix* que irá criar o objeto *phiEqn* que representa a equação. Ao definir cada termo da equação, através dos operadores diferenciais representados pelas classes, o sistema cria uma matriz de coeficientes da discretização resultante da ação de cada operador. Cada operador irá agir segundo o especificado pelo caso, ou seja, o método de discretização e interpolação das propriedades de cada um dos operadores pode mudar de acordo com o que for escolhido no caso. No caso dos operadores no *namespace fvm* serão feitas discretizações implícitas, caso sejam usados os operadores diferenciais do *namespace fvc* a discretização será feita de forma explícita. A possibilidade de mudar o esquema de discretização sem ser necessária a implementação de uma nova rotina, dá aos programas implementados nesse ambiente uma enorme flexibilidade, dessa forma é possível variar o método numérico facilmente e a investigação de diferentes alternativas torna-se uma tarefa simples. A solução do sistema de equações resultante é feita através da chamada de uma rotina de solução, que assim como os operadores diferenciais mudam o esquema de solução de acordo com o especificado no caso. A combinação de esquemas numéricos a serem utilizados será escolhida no caso a ser estudado, sendo a aplicação apenas uma matriz de alto nível, que forma o programa implementado ao ser utilizada pelo caso. A aplicação genérica sendo chamada pelo caso, segundo os métodos escolhidos, irá dar forma ao simulador desejado.

Para o aprendizado do sistema, foram implementadas aplicações com o OpenFOAM para problemas envolvendo separadamente as três equações diferenciais parciais básicas que estão presentes na maioria dos fenômenos estudados. A solução dessas equações separadamente ajuda a resolver os problemas de implementação do simulador a ser desenvolvido, que usa duas das equações presentes nesses modelos básicos. Os problemas estudados foram os fenômenos de difusão de calor em regime permanente, transiente e o problema de transporte advectivo. Como parte inicial do simulador foi desenvolvida uma aplicação para a solução do escoamento monofásico em meios porosos, que consiste na solução de uma equação elíptica, equação de pressão, modelando o fenômeno em regime permanente. Posteriormente foi implementado o modelo bifásico, que usa a equação de advecção para a saturação e uma equação elíptica para a solução da equação de pressão.

2.3.1 Difusão de calor

Para a solução da equação parabólica, que foi empregada na solução da equação de pressão do simulador monofásico, foram feitas duas aplicações no OpenFOAM para a solução do problema de difusão de calor em regime permanente e transiente. O fenômeno de difusão transiente é descrito pela equação (2.2),

$$\nabla^2 T + q = \frac{1}{\alpha} \frac{\partial T}{\partial t}, \quad (2.2)$$

onde T representa a temperatura, q representa um termo de geração pontual de calor e α a difusividade térmica do material. No OpenFoam o código em C++ referente a discretização é escrito da seguinte forma:

```

1  fvScalarMatrix Teqn
2  (
3      fvc::laplacian(T)
4      + q
5      ==
6      (1/alpha)*fvm::ddt(T)
7  )

```

Para a solução dos problemas em regime permanente é usada a equação 2.2 eliminando-se a derivada temporal. Pode-se facilmente mudar o método de discretização de uma formulação explícita para uma formulação implícita apenas mudando o termo $fvc::laplacian(T)$ para $fvm::laplacian(T)$.

Deve-se tomar cuidado ao implementar uma aplicação que utiliza equações adimensionalizadas nesse sistema, já que as variáveis, quando criadas, recebem uma unidade associada. A Tabela 2.1 mostra o padrão para representar as unidades no sistema.

Tabela 2.1: Padrão de unidades no OpenFOAM.

Posição	Propriedade	Unidade
1	Massa	Kilograma (kg)
2	Comprimento	metro (m)
3	Tempo	segundo (s)
4	Temperatura	Kelvin (K)
5	Quantidade	kilograma-mol (kg-mol)
6	Corrente	Ampere (A)
7	Intensidade luminosa	candela (cd)

A associação de cada campo a uma unidade no sistema é feita para evitar que sejam feitas operações que não estejam dimensionalmente corretas. Cada operador diferencial carrega consigo uma operação dimensional, por exemplo o resultado do operador `fv::div` resulta em uma divisão pela unidade de comprimento.

Por conveniência o sistema de unidades adotado na aplicação desenvolvida é o sistema internacional de unidades, SI, e na mudança de unidades é feito o uso de constantes em alguns termos das equações. Conforme mostrado na Tabela 2.1 ao criar-se um campo de pressões em $Pa = (kg\ m^{-1}\ s^{-2})$ as unidades da variável devem ser informadas ao método que cria o campo através de um vetor contendo números que representam as potências as quais serão elevadas cada unidade. Seguindo a ordem apresentada anteriormente na tabela 2.1 o vetor `[1 -1 -2 0 0 0 0]` representará a unidade da variável de pressão no sistema.

2.3.2 Advecção linear

Para a solução da equação hiperbólica, que será empregada na solução da equação de saturação do simulador, foi feita uma aplicação no OpenFOAM para a solução do problema de advecção. O fenômeno é descrito pela equação (2.3).

$$\frac{\partial f}{\partial t} + \nabla \cdot (\vec{u}f) = 0 \quad (2.3)$$

onde f representa a grandeza que se está sendo transportada e \vec{u} o campo vetorial de velocidades responsável pelo transporte advectivo de f . A aplicação usada para a solução dessa equação faz parte das aplicações que vem com a biblioteca, no entanto se optou por implementar uma aplicação similar para melhor aprendizado do funcionamento do sistema. No OpenFOAM o código

em C++ referente a aplicação é escrito da seguinte forma:

```

1  #include "fvCFD.H"
2
3  int main(int argc, char *argv[])
4  {
5  # include "setRootCase.H"
6  # include "createTime.H"
7  # include "createMesh.H"
8  # include "createFields.H"
9
10 surfaceScalarField phi=(fvc::interpolate(U) & mesh.Sf());
11
12 for (runTime++; !runTime.end(); runTime++)
13 {
14   f.storePrevIter();
15
16   solve(fvm::ddt(f) == - fvm::div(phi, f));
17   phi = (fvc::interpolate(U) & mesh.Sf());
18
19   runTime.write();
20 }
21 return 0;
22 }
```

A primeira inclusão descrita no código anterior torna acessível o *namespace* FOAM, que é o escopo onde estão os *namespaces* como *fvc* e *fvm* que por sua vez abrigam os templates para as classes como *div*, *grad* e *laplacian*. A inclusão *setRootCase.H* irá verificar uma lista de argumentos de entrada necessários à execução do programa. Em *createTime.H* são criados os parâmetros de controle da simulação, como instante inicial, final e incremento de tempo. A inclusão *createMesh.H* irá gerar os dados de malha necessários na discretização. A última inclusão *createFields.H* tem conteúdo definido pelo usuário que cria a aplicação. Neste arquivo são criados os campos que serão usados na aplicação, seus valores e condições de contorno que serão lidos do diretório do caso. Na representação descrita no código anterior o termo de divergência usa o campo vetorial das velocidades interpolado dos vértices para as faces. Para realizar a interpolação usa-se a seguinte linha de código em C++:

```
1 u_f = fvc::interpolate(u);
```

Com os valores do campo de velocidades interpolado para as faces é calculado o fluxo através de cada face, representado pelo campo escalar *phi*. A escolha do método de interpolação a ser usado é feita no diretório contendo o caso, desta forma pode-se usar diferentes métodos de interpolação sem fazer mudanças na implementação da aplicação desenvolvida. Assim como nos

problemas de calor descritos anteriormente pode-se mudar a formulação de explícita para implícita, desta forma para se implementar a técnica *upwind* implícita escolhemos o método de interpolação linear no caso a ser simulado e usamos o seguinte código na aplicação:

```

1 fvScalarMatrix Feqn(
2     fvm::div(phi, f)
3     + fvm::ddt(f)
4     == 0
5 )
6 Feqn.solve();

```

Para usar uma formulação explícita basta apenas usar o *template* de discretização de divergentes do *namespace* *fvc*, *fvc::div*.

2.4 Casos

Caso, ou *case*, é o nome dado ao diretório contendo os arquivos que servirão de entrada para a aplicação utilizada. Esses diretórios são organizados segundo o padrão do OpenFOAM. As aplicações desenvolvidas usando as classes desse sistema modelam problemas tridimensionais podendo ser usadas para resolver problemas unidimensionais e bidimensionais.

Como exemplo de um caso com geometria tridimensional, é mostrada a Figura 2.2 com o modelo de uma barra com difusividade térmica constante e uniforme ao longo de toda geometria da barra, submetida a um gradiente de temperatura imposto, aplicado às duas extremidades da barra.

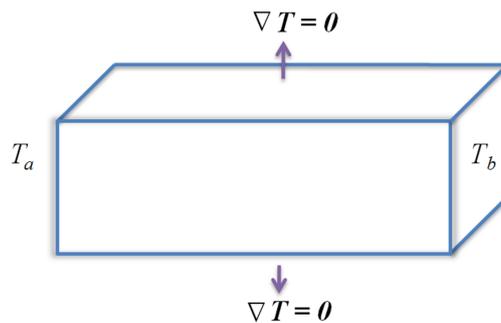


Figura 2.2: Esquema do caso de difusão.

É imposto um fluxo de calor nulo nas faces que circundam a barra. Essas condições de contorno com temperatura prescrita em duas extremidades e fluxo nulo em redor da barra fornecem uma solução conhecida, que representa uma distribuição linear de temperatura, possibilitando assim a verificação dos resultados. A Figura 2.3 mostra um exemplo da organização padrão de um

diretório com a estrutura de arquivos que forma um caso que usa a aplicação desenvolvida para o problema de difusão descrito anteriormente.

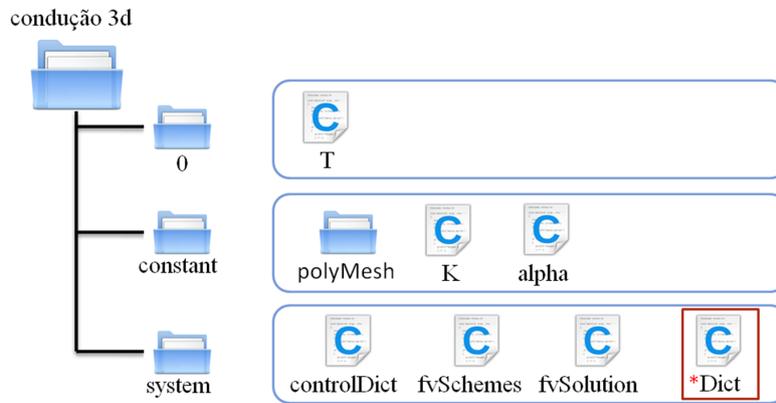


Figura 2.3: Estrutura do diretório de um caso.

Olhando-se para os arquivos no diretório do caso temos um diretório chamado de 0, que contém os arquivos que definem as condições iniciais dos campos e os valores das variáveis que dependem do tempo. No OpenFOAM existem campos escalares, vetoriais e tensoriais que, por definição, já são em três dimensões, para resolver um problema unidimensional deve-se criar uma fila de blocos e para um problema bidimensional um plano de blocos. No caso do problema de difusão aqui descrito o campo será tridimensional, escalar e uniforme com os valores iniciais de temperatura contidos no arquivo nomeado como T . Nesse arquivo, também encontra-se a definição das condições de contorno que são impostas segundo palavras chave e que são associadas a partes da geometria com nomes definidos no arquivo com a geometria do problema. No OpenFOAM são possíveis diferentes tipos de condições de contorno que podem ser impostas, como exemplo, pode-se citar as condições *fixedValue*, *cyclic*, *zeroGradient* e *symmetryPlane*.

O arquivo com a geometria do problema com o qual será gerada a malha encontra-se no diretório *polyMesh*, que neste caso está no diretório *constant*, se aqui estivesse sendo usada a ferramenta de geração adaptativa de malhas como a malha mudaria com o tempo este diretório deveria estar dentro da pasta 0 como o arquivo de temperatura. O trecho de código a seguir mostra a criação da geometria e a discretização deste problema com o *blockMesh*.

```

1 FoamFile
2 {
3     version      2.0;
4     format       ascii;
5     class        dictionary;

```

```

6     object      blockMeshDict;
7 }
8
9     convertToMeters 1;
10
11    vertices
12    (
13        (0 0 0) (1 0 0)
14        (1 5 0) (0 5 0)
15        (0 0 1) (1 0 1)
16        (1 5 1) (0 5 1)
17    );
18
19    blocks
20    (
21        hex (0 1 2 3 4 5 6 7) (5 25 5) simpleGrading (1 1 1)
22    );
23
24    edges();
25
26    patches
27    (
28        patch down ( (1 5 4 0) )
29        patch up   ( (3 7 6 2) )
30        patch left ( (0 4 7 3) )
31        patch right( (2 6 5 1) )
32        patch front( (4 5 6 7) )
33        patch back ( (0 3 2 1) )
34    );
35
36    mergePatchPairs();

```

No arquivo com a geometria são definidos os *patches* que, neste caso, são os planos que formam o contorno da barra. Através do nome associado a cada *patch* são definidas as condições de contorno de cada variável criada no problema. O trecho de código a seguir mostra o arquivo nomeado como T que cria o campo de temperaturas do problema descrito.

```

1 FoamFile
2 {
3     version      2.0;
4     format       ascii;
5     class        volScalarField;
6     location     "constant";
7     object       T;
8 }
9
10 dimensions      [1 -1 -2 0 0 0 0];
11

```

```

12 internalField    uniform 0;
13
14 boundaryField
15 {
16     down { type fixedValue; value uniform 0; }
17     up   { type fixedValue; value uniform 10;}
18     left { type zeroGradient; }
19     right { type zeroGradient; }
20     front { type zeroGradient; }
21     back { type zeroGradient; }
22 }

```

Ainda no diretório *constant* encontram-se os campos K , tensorial, e α , escalar. Outro arquivo pode ser adicionado a esse diretório a fim de definir variáveis com valor constante no problema. No diretório *system* encontram-se os arquivos com os parâmetros que irão controlar a aplicação. No arquivo *controlDict* estão parâmetros como tempo inicial e final de simulação, incremento de tempo e outras variáveis de controle como o número de Courant para problemas com advecção, podendo ainda o usuário utilizar como passo de tempo inicial o último passo de tempo simulado. No arquivo *fvSchemes* é especificado o método de discretização aplicado a cada operador diferencial e os esquemas de interpolação utilizados para cada variável. No arquivo *fvSolution* são definidos os *solvers* utilizados para a solução do sistema de equações resultante da discretização de cada equação. Outros arquivos de controle da simulação podem ser adicionados como mostrado na Figura 2.3 no arquivo **Dict* como por exemplo quando resolvemos o problema usando a aplicação em paralelo que é necessário criar o arquivo *decomposeParDict* que especifica em quantos subdomínios será dividido o domínio do problema, qual o método de divisão utilizado e quantos subdomínios irão ficar com cada processador.

Após a implementação da aplicação, a mesma é compilada usando parâmetros de compilação definidos no arquivo de configuração da biblioteca OpenFOAM. Para compilar a aplicação basta digitar o comando *wmake* no diretório da aplicação. Após a compilação da aplicação no diretório do caso, são escolhidos os esquemas e as condições iniciais e de contorno do problema.

Para gerar a malha é executado o comando *blockMesh* no diretório do caso. Caso seja usado um gerador de malhas externo basta converter a malha para o formato nativo e copiar o arquivo para o diretório *polyMesh* dentro do diretório *constant*. A seguir é apresentado o resultado da geração da malha para o problema de difusão de calor mostrado anteriormente. A geometria do problema é uma barra com dimensões 1x5x1 discretizada através da malha mostrada na Figura 2.4, onde as dimensões em x , y e z foram discretizadas com 5, 25 e 5 blocos em cada direção respectivamente exposta a um diferencial

de temperatura de $10K$ para $0K$.

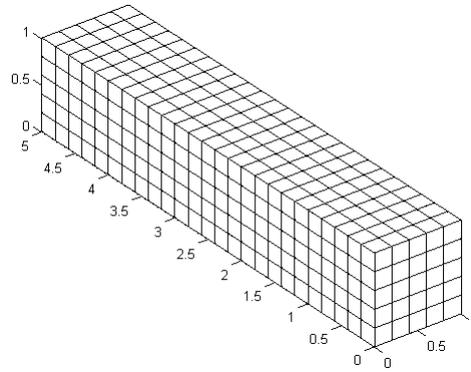


Figura 2.4: Malha da discretização do problema de calor.

Após a geração da malha com o *blockMesh* é executada a aplicação no diretório do caso digitando-se o comando com o nome referente a aplicação. A Figura 2.5 mostra a solução do problema descrito anteriormente no caso de difusão de calor.

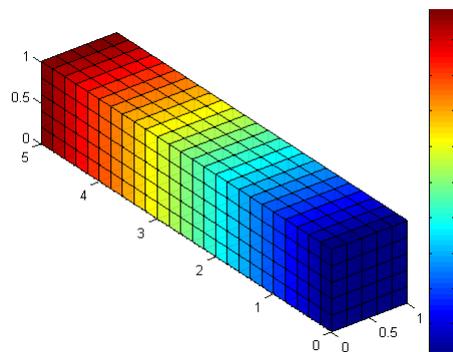


Figura 2.5: Solução do caso com o problema de calor.

A solução apresentada mostra o resultado quando o fluxo de calor na barra encontra-se em regime permanente.

Capítulo 3

Equações Governantes

A modelagem de reservatórios de petróleo consiste na simulação do escoamento dos fluidos tipicamente encontrados na rocha reservatório através de uma matriz permoporosa. O comportamento dos hidrocarbonetos nas condições de reservatório pode ser retratado através de modelos como o black-oil, API-tracking, composicional [14] [13] e pseudo composicional [28]. As diferenças básicas entre cada um desses modelos são o tratamento dado à interação entre os fluidos, o número de componentes e seu comportamento devido às variações de pressão, temperatura e composição. Cada modelo aplica-se melhor a um tipo de sistema petrolífero, onde para cada condição de pressão e temperatura do reservatório estudado tem-se um comportamento diferente dos fluidos contidos na rocha reservatório. As propriedades reológicas e termodinâmicas comportam-se de maneira diferente favorecendo, em cada caso, um modelo de fluxo em particular.

Através do diagrama de fases de uma determinada mistura de hidrocarbonetos pode-se, dependendo das condições de pressão e temperatura do reservatório, determinar as características dos fluidos que serão encontrados em seu interior. A Figura 3.1 mostra a aplicação comumente utilizada dos tipos de modelos mais usados na indústria do petróleo. O modelo utilizado neste trabalho considera que os fluidos que deslocam-se no reservatório são imiscíveis e incompressíveis, este modelo bifásico óleo água representa uma simplificação do modelo black oil. O modelo aqui apresentado considera que o escoamento desenvolve-se numa rocha com compressibilidade nula. A temperatura do escoamento é assumida constante e as propriedades dos fluidos não variam com o tempo.

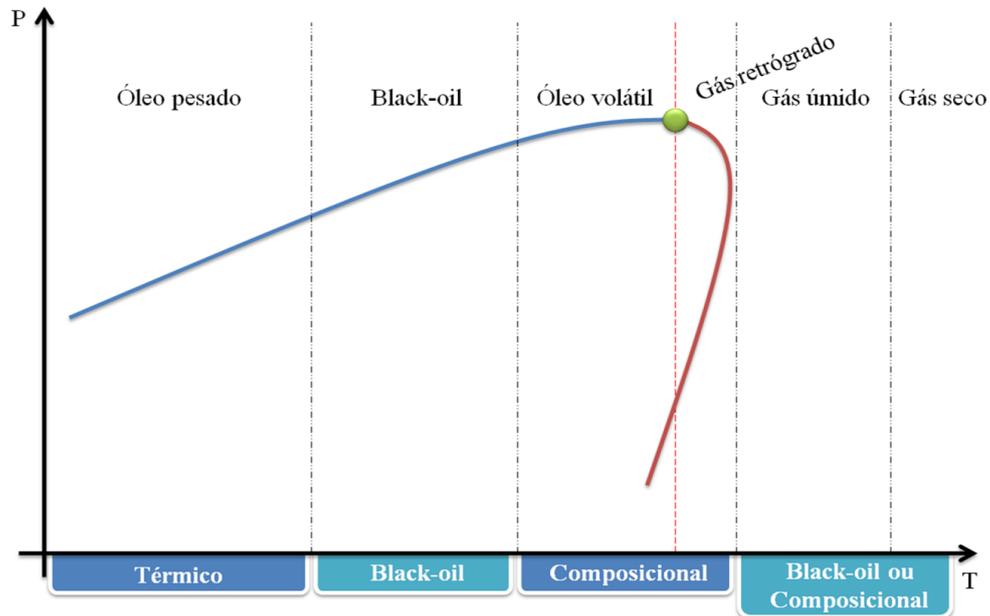


Figura 3.1: Aplicação dos modelos de fluxo. Modificado de [20] *apud* [29].

3.1 Propriedades da rocha e dos fluidos

A definição das propriedades da rocha reservatório e dos fluidos que nela escoam são o ponto de partida para a caracterização do problema abordado neste tipo de simulador. O simulador monofásico desenvolvido como etapa inicial considera fluido e rocha compressíveis. Desta forma, para a formulação do modelo monofásico, serão apresentadas as equações para as propriedades de fluido e rocha compressíveis. As equações do modelo bifásico serão apresentadas com os termos compressíveis e posteriormente serão simplificadas para o modelo bifásico incompressível. As propriedades da rocha usadas nos problemas de fluxo em reservatórios de petróleo são a porosidade, ϕ , e a permeabilidade absoluta, K , que, para o caso de um reservatório heterogêneo, possuem diferentes valores em cada célula do domínio. A porosidade, ou porosidade efetiva, em uma determinada porção da rocha, é definida como a razão entre o volume de vazios interconectados existentes e o volume total, podendo variar de acordo com a pressão do fluido confinado nos poros. A porosidade numa célula informa o quanto da célula está ocupada pelo fluido e sua variação com a pressão pode ser obtida pela seguinte expressão linearizada [20]:

$$\phi(p) = \phi_{ref} [1 + c_r(p - p_{ref})], \quad (3.1)$$

onde ϕ_{ref} é a porosidade obtida em relação a uma pressão de referência p_{ref} e c_r é a compressibilidade da rocha.

A permeabilidade absoluta é uma medida que representa a facilidade que o fluido tem de escoar pelos poros interconectados da rocha, essa variável aparece nos modelos matemáticos representada pelo tensor de segunda ordem de permeabilidades absolutas segundo as direções x , y e z , como mostrado na Equação (3.2):

$$K = \begin{pmatrix} k_{xx} & k_{xy} & k_{xz} \\ k_{yx} & k_{yy} & k_{yz} \\ k_{zx} & k_{zy} & k_{zz} \end{pmatrix}. \quad (3.2)$$

No equacionamento usado na simulação de reservatórios o tensor é frequentemente representado na forma diagonal, obtido quando as direções preferenciais de fluxo coincidem com as direções principais do tensor de permeabilidades absolutas. O tensor na forma diagonalizada é mostrado na Equação (3.3):

$$K = \begin{pmatrix} k_{xx} & 0 & 0 \\ 0 & k_{yy} & 0 \\ 0 & 0 & k_{zz} \end{pmatrix}. \quad (3.3)$$

Algumas propriedades dos fluidos são importantes para a modelagem computacional do fluxo em meios porosos nas condições específicas dos reservatórios. Dentre elas temos: compressibilidade, fator volume de formação, massa específica e viscosidade.

A compressibilidade de um fluido, c_l , pode ser entendida como uma medida da variação do volume do fluido quando o mesmo sofre uma variação de pressão. Admitindo-se que a temperatura ao longo de todo reservatório permanece constante podemos escrever as seguintes relações:

$$c_l = -\frac{1}{V} \frac{\partial V}{\partial p}, \quad (3.4)$$

ou ainda:

$$c_l = \frac{1}{\rho} \frac{\partial \rho}{\partial p}. \quad (3.5)$$

onde V e ρ são o volume e a massa específica do fluido respectivamente, e são medidos segundo as condições de reservatório. Uma outra propriedade bastante usada na simulação de reservatórios é o fator volume de formação, que representa uma relação entre o volume do fluido nas condições de reservatório, V , e em condições padrão V_{sc} . As condições padrão, *standard conditions*, adotadas são referentes a uma temperatura de $60^\circ F$ e a uma pressão de $14,7 \text{ psi}$. A Equação (3.6) mostra a relação:

$$B = \frac{V}{V_{sc}}, \quad (3.6)$$

ou:

$$B = \frac{\rho_{sc}}{\rho}. \quad (3.7)$$

Em escoamentos com fluidos levemente compressíveis podemos escrever uma relação entre a variação do fator volume de formação e a pressão. Integrando-se a Equação (3.4) admitindo-se temperaturas constantes ao longo do reservatório pode-se escrever:

$$B(p) = \frac{B_{ref}}{[1 + c_l (p - p_{ref})]}, \quad (3.8)$$

onde B_{ref} é o fator volume de formação em relação a pressão de referência que normalmente é a pressão de bolha, P_b , mas como a relação é linear pode-se utilizar qualquer valor maior que P_b . A Figura 3.2 mostra o comportamento do fator volume de formação para o óleo.

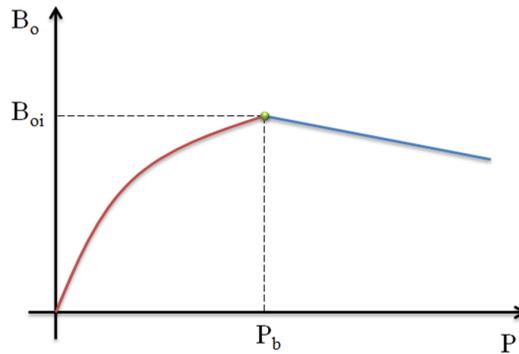


Figura 3.2: Comportamento do fator volume de formação.

Sabe-se que um óleo com uma maior quantidade de frações leves tem um B_o maior que um óleo pesado a uma dada pressão. Com a diminuição da pressão o valor de B_o aumenta até chegar no valor de pressão em que a fração mais leve do óleo sai de solução, a partir deste ponto o valor de B_o começa a cair novamente. A Equação (3.8) pode ser utilizada considerando-se que o reservatório encontra-se a uma profundidade tal que a pressão estática seja maior que a pressão de bolha, valendo assim a linearização feita segundo uma pressão de referência de modo que tenha-se $p_{ref} \geq P_b$.

Sob as condições de reservatório mencionadas, pode-se, a partir das Equações (3.7) e (3.8), escrever uma relação para a variação da massa específica em função da pressão para valores de pressão acima da pressão de bolha:

$$\rho(p) = \rho_{ref} [1 + c_l (p - p_{ref})], \quad (3.9)$$

A última propriedade a ser definida é a viscosidade do fluido, que varia em função da pressão assim como a porosidade e o fator volume de formação. A relação pode ser escrita como:

$$\mu(p) = \frac{\mu_{ref}}{[1 - c_\mu (p - p_{ref})]}, \quad (3.10)$$

onde μ_{ref} é o valor da viscosidade em relação à pressão de referência igual a p_{ref} . A constante c_μ é a mudança fracional da viscosidade por unidade de variação de pressão. Em sistemas trifásicos essa mudança fracional é em grande parte uma função da fração de gás dissolvido no óleo. A Figura 3.3 mostra a variação da viscosidade em função da pressão que assim como o fator volume da formação tem um comportamento linear para valores de pressão maiores que a pressão de bolha.

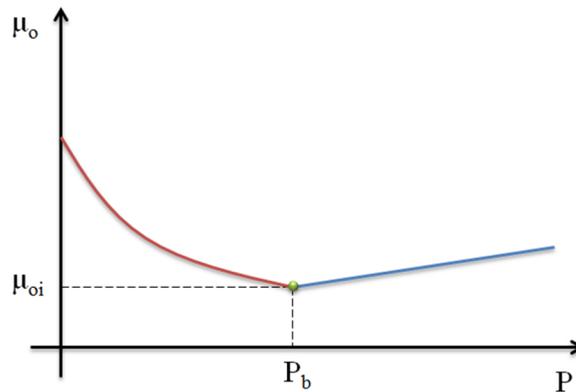


Figura 3.3: Comportamento da viscosidade.

Abaixo da pressão de bolha a medida que as frações mais leves saem de solução do óleo, com a queda de pressão, a viscosidade aumenta. Acima da pressão de bolha um aumento na pressão resulta em um respectivo aumento da viscosidade. Sob as mesmas condições de reservatório com pressão estática maior que a pressão de bolha é razoável a linearização.

3.2 Lei de Darcy

A lei de Darcy [20] é uma lei empírica que relaciona a vazão de um fluido que escoar em um meio poroso com o gradiente potencial aplicado ao mesmo. O potencial é definido pela Equação (3.11):

$$\Phi = p + \gamma Z, \quad (3.11)$$

onde p é a pressão do líquido a uma profundidade Z , calculada em relação a um referencial orientado para baixo, e γ é o peso específico do fluido definido como $\gamma = \gamma_c \rho g$, onde ρ e g são a massa específica do fluido e a gravidade local respectivamente.

Os termos α_c , β_c e γ_c que aparecem ao longo do texto são constantes usadas para compatibilizar as unidades usadas do sistema usual adotado na simulação de escoamento de petróleo [20] e o sistema internacional de unidades. A tabela 3.2 mostra o valor dos fatores para o sistema internacional de unidades (SI) e para o sistema de unidades usual utilizado na indústria do petróleo. Esses termos são introduzidos para facilitar a mudança de sistema de unidades, que no OpenFOAM é feita durante a criação das variáveis como foi discutido no Capítulo 2.

Tabela 3.1: Fatores de conversão de unidades.

Fator	SI	Usual
α_c	1.0	5.614583
β_c	$8.64 \cdot 10^{-5}$	1.127
γ_c	$1.0 \cdot 10^{-3}$	$2.1584 \cdot 10^{-4}$

Assumindo-se um fluxo monofásico unidimensional, a velocidade superficial, ou velocidade de Darcy pode ser escrita segundo a expressão:

$$u_x = -\beta_c \frac{k_{xx}}{\mu} \frac{\partial \Phi}{\partial x} \quad (3.12)$$

Na Equação (3.12) o termo k_{xx} representa a permeabilidade absoluta do meio poroso na direção x e μ a viscosidade do fluido. Pode-se ainda escrever a lei de Darcy para o caso multidimensional cuja equação de velocidades pode ser escrita da seguinte forma:

$$\vec{u} = -\beta_c \frac{K}{\mu} \nabla \Phi, \quad (3.13)$$

onde \vec{u} é o vetor de velocidades de Darcy e K é um tensor de permeabilidades absolutas do meio poroso.

3.3 Lei da conservação de massa

A lei da conservação de massa, também chamada de equação da continuidade, é uma expressão matemática que representa um balanço entre fluxos de massa. Pode-se obter a lei de conservação de massa considerando-se o

fluxo através de um volume de controle como mostrado na Figura 3.4. A variação temporal da massa que passa através de um volume de controle V pode ser expressa pela relação [20]:

$$(m_e - m_s) + m_g = m_{ac}, \quad (3.14)$$

onde m_e é a massa que entra no volume de controle, m_s a massa que sai do volume de controle, m_g a massa gerada no volume de controle durante um intervalo de tempo Δt , devido a termos de fonte, e m_{ac} é a massa acumulada no volume de controle durante o mesmo intervalo de tempo.

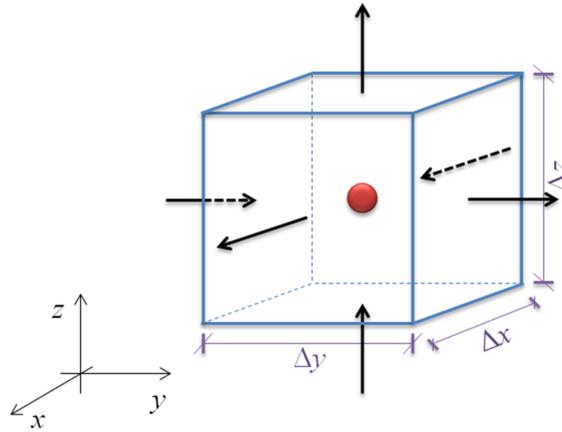


Figura 3.4: Esquema de fluxo.

Se considerarmos um volume de controle hexaédrico como o mostrado na Figura 3.1, podemos escrever:

$$m_e = \left[(\dot{m}_x A_x)_{x-\Delta x/2} + (\dot{m}_y A_y)_{y-\Delta y/2} + (\dot{m}_z A_z)_{z-\Delta z/2} \right] \Delta t \quad (3.15)$$

e

$$m_s = \left[(\dot{m}_x A_x)_{x+\Delta x/2} + (\dot{m}_y A_y)_{y+\Delta y/2} + (\dot{m}_z A_z)_{z+\Delta z/2} \right] \Delta t, \quad (3.16)$$

onde x , y e z são as coordenadas do centro do volume de controle, A_x , A_y e A_z são as áreas das faces segundo as direções x , y e z respectivamente e \dot{m} é o fluxo de massa por unidade de área que passa pelas faces do volume de controle, que, para a direção x , pode ser definido como:

$$\dot{m}_x = \alpha_c \rho u_x. \quad (3.17)$$

Definindo m_g como a massa gerada no interior do volume de controle durante um intervalo de tempo Δt devido a uma vazão q podemos escrever:

$$m_g = q_m \Delta t = \rho q \Delta t. \quad (3.18)$$

O termo de acumulação representa a quantidade de massa acumulada durante um intervalo de tempo e pode ser escrito como:

$$m_{ac} = m_V^{t+\Delta t} - m_V^t, \quad (3.19)$$

onde m_V^t é quantidade total de massa contida no volume de controle no instante de tempo t . Como trabalharemos com malhas estruturadas hexaédricas essa quantidade pode ser escrita em função do volume do bloco, V_b , e da porosidade do média no bloco, ϕ . Dessa forma podemos reescrever a Equação (3.19) como:

$$m_{ac} = (\rho\phi\Delta x\Delta y\Delta z)^{t+\Delta t} - (\rho\phi\Delta x\Delta y\Delta z)^t, \quad (3.20)$$

substituindo as equações (3.15), (3.16), (3.18) e (3.20) na Equação (3.14) e dividindo-se a equação resultante pelo volume do bloco hexaédrico e por Δt temos:

$$\begin{aligned} & \left[\frac{\dot{m}_{x+\Delta x/2} - \dot{m}_{x-\Delta x/2}}{\Delta x} \right] - \left[\frac{\dot{m}_{y+\Delta y/2} - \dot{m}_{y-\Delta y/2}}{\Delta y} \right] \\ & - \left[\frac{\dot{m}_{z+\Delta z/2} - \dot{m}_{z-\Delta z/2}}{\Delta z} \right] + \frac{q_m}{\alpha_c V_b} = \frac{\phi\rho^{t+\Delta t} - \phi\rho^t}{\Delta t}. \end{aligned} \quad (3.21)$$

Escrevendo-se a Equação (3.17) para demais direções cartesianas, pode-se avaliar a mesma no limite em que os valores de Δx , Δy , Δz e Δt tendem a zero, encontrando:

$$-\frac{\partial}{\partial x}(\rho u_x) - \frac{\partial}{\partial y}(\rho u_y) - \frac{\partial}{\partial z}(\rho u_z) + \frac{q_m}{\alpha_c V_b} = \frac{1}{\alpha_c} \frac{\partial}{\partial t}(\phi\rho), \quad (3.22)$$

ou ainda:

$$\frac{\partial}{\partial t}(\phi\rho) + \alpha_c \nabla \cdot (\rho \vec{u}) = \frac{q_m}{V_b}. \quad (3.23)$$

A Equação (3.23) representa a forma final da equação da lei de conservação de massa, ou equação da continuidade.

3.4 Escoamento monofásico

No modelo monofásico em que admite-se que a temperatura é constante resta apenas a equação de pressão a ser resolvida e com os valores de pressão todas as outras propriedades do fluido são determinadas. A equação de pressão é obtida através das duas leis expostas anteriormente, da equação de

conservação de massa e da lei de Darcy. Substituindo a equação (3.12) na equação (3.23) encontramos:

$$\frac{\partial}{\partial t} (\phi\rho) - \alpha_c\beta_c\nabla \cdot \left(\rho \frac{K}{\mu} \nabla\Phi \right) = \frac{q_m}{V_b}. \quad (3.24)$$

introduzindo-se (3.11):

$$\nabla \cdot \left[\beta_c \rho \frac{K}{\mu} (\nabla p + \gamma \nabla Z) \right] + \frac{q_m}{\alpha_c V_b} = \frac{1}{\alpha_c} \frac{\partial (\rho\phi)}{\partial t} \quad (3.25)$$

dividindo por ρ_{sc} , usando-se (3.7) e sabendo que $q_m = \alpha_c q_{sc} \rho_{sc}$ obtém-se:

$$\nabla \cdot \left[\beta_c \frac{K}{B\mu} (\nabla p + \gamma \nabla Z) \right] + \frac{q_{sc}}{V_b} = \frac{1}{\alpha_c} \frac{\partial}{\partial t} \left(\frac{\phi}{B} \right) \quad (3.26)$$

O modelo implementado na aplicação para o escoamento monofásico assume que o fluido é levemente compressível, como água e óleo pesado, e que a porosidade da rocha varia segundo a Equação (3.1). As equações para a modelagem de um fluido levemente compressível são as Equações (3.8) e (3.10), substituindo essas equações na Equação (3.26) através de alguma manipulação pode-se obter:

$$\nabla \cdot \left[\beta_c \frac{K}{B\mu} (\nabla p + \gamma \nabla Z) \right] + \frac{q_{sc}}{V_b} = \frac{1}{\alpha_c} \left(\phi \frac{c_l}{B_{ref}} + \phi_{ref} \frac{c_r}{B} \right) \frac{\partial p}{\partial t} \quad (3.27)$$

A Equação (3.27), que descreve o escoamento monofásico em meios porosos para fluidos levemente compressíveis. Como os valores das propriedades dos fluidos e da rocha dependem da pressão, a Equação (3.27) é não linear. Para linearizá-la é usada uma formulação iterativa direta na qual os valores das propriedades do fluido e da rocha são calculados e a equação é resolvida implicitamente para a pressão, os valores das propriedades são recalculados e a equação é resolvida novamente, o processo continua até que os valores da pressão convirjam para a solução a cada passo de tempo.

3.5 Escoamento bifásico

O modelo aplicado para a modelagem do escoamento em reservatórios no simulador desenvolvido foi o de escoamento bifásico imiscível e incompressível em meios porosos. O modelo é baseado nos mesmos conceitos aplicados à obtenção das equações do escoamento monofásico. A partir da lei de conservação de massa para cada componente do escoamento é encontrada a equação

de saturação. Uma combinação das equações de saturação é feita para a obtenção da equação que calculará a distribuição de pressões no reservatório. A dedução das equações de pressão e saturação separadamente é necessária devido ao método empregado para a solução do problema. O método IMPES que será descrito no próximo capítulo calcula primeiro o campo de pressões e posteriormente é encontrada a distribuição de saturações no reservatório.

Equação de Saturação

A saturação de uma fase, em uma determinada porção de rocha, representa o quociente entre o volume da fase presente nos poros e o volume poroso, expressando uma medida percentual de cada fase presente. Para o modelo de escoamento bifásico totalmente saturado aqui estudado pode-se escrever:

$$s_w + s_o = 1 \quad (3.28)$$

onde s_o e s_w são as medidas em valores que variam entre 0 e 1 das saturações de óleo e água, respectivamente. No modelo de reservatórios de petróleo existem ainda duas variáveis, s_{ro} e s_{rw} , que representam as saturações residuais de óleo e água. As saturações residuais estão presentes no meio poroso e representam quantidades de fluido que não podem ser retiradas, permanecendo na matriz porosa durante o escoamento. A equação de saturação para cada fase é obtida através da Equação (3.23), escrita para cada uma das fases, donde encontramos:

$$\frac{1}{\alpha_c} \frac{\partial (\phi s_o \rho_o)}{\partial t} + \nabla \cdot (\rho_o \vec{u}_o) = \frac{\rho_{osc} q_{osc}}{V_b} \quad (3.29)$$

para a fase óleo e:

$$\frac{1}{\alpha_c} \frac{\partial (\phi s_w \rho_w)}{\partial t} + \nabla \cdot (\rho_w \vec{u}_w) = \frac{\rho_{wsc} q_{wsc}}{V_b} \quad (3.30)$$

para a fase água. Os termos q_o e q_w correspondem às vazões de óleo e água respectivamente, os vetores \vec{u}_o e \vec{u}_w representam as velocidades das fases e são calculados de acordo com a lei de Darcy e ponderada entre as fases de acordo com a permeabilidade relativa da mesma. A permeabilidade representa a medida da facilidade que o fluido tem de escoar no meio poroso. Um fluido confinado em um meio poroso possui como valor máximo de permeabilidade a permeabilidade absoluta. Em presença de outros fluidos a facilidade de escoamento no meio é reduzida a um valor menor que a permeabilidade absoluta. A permeabilidade de uma fase em um meio poroso na direção x pode ser descrita como:

$$k_l = k_x \left(\frac{k_l}{k_x} \right) = k_x k_{rl} \quad (3.31)$$

A permeabilidade da fase l , com $l = o$ ou $l = w$, é dada pelo produto entre a permeabilidade absoluta na direção x e a permeabilidade relativa. A permeabilidade relativa é uma grandeza que varia de acordo com a quantidade de fluido no poro, ou seja, em função da saturação do fluido. Existem modelos, obtidos através de experimentos, que relacionam os valores das permeabilidades relativas com as saturações [36]. Dependendo do meio poroso estudado, esses modelos podem mudar. Experimentalmente pode ser obtidas duas curvas de permeabilidade relativa, uma relativa ao processo de drenagem e outra ao processo de embebição. Nos simuladores de escoamento em meios porosos é comumente utilizada a curva de permeabilidade relativa referente ao processo que está ocorrendo. Em simulação de reservatórios o perfil das curvas de permeabilidade relativa é um dos parâmetros que são controlados para melhor representar o histórico de produção do reservatório variando-se o perfil das curvas de permeabilidade relativa é possível ajustar a resposta fornecida pelo simulador ao histórico de pressões do reservatório. A Figura 3.5 mostra um exemplo com curvas de permeabilidade relativa para um sistema bifásico, onde s_{wi} representa a saturação de água irreduzível e

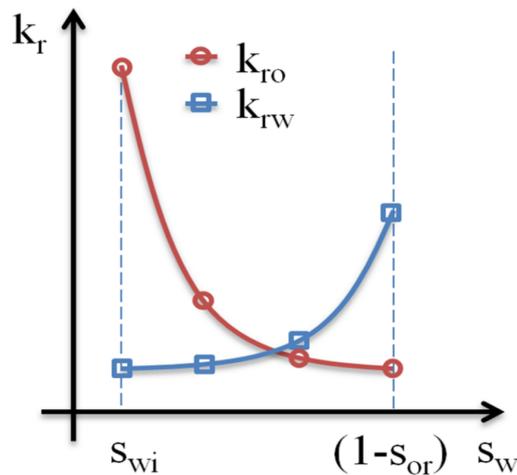


Figura 3.5: Curvas de permeabilidade relativa.

s_{or} a saturação de óleo residual. A saturação de água irreduzível representa a fração de água que fica aprisionada nos poros devido ao efeito Jamim [36] enquanto a saturação de óleo residual representa a fração de óleo que não pode ser extraída do reservatório.

A lei de Darcy descrita pela Equação (3.13) foi originalmente obtida para um escoamento monofásico, podendo ser modificada introduzindo-se o conceito de permeabilidade relativa. Desta forma podemos reescrever a equação

para a fase l como:

$$\vec{u}_l = -\beta_c \frac{Kk_{rl}}{\mu_l} (\nabla p_l + \gamma_l \nabla Z), \quad (3.32)$$

substituindo as velocidades na equação de conservação de massa para cada fase obtemos:

$$\frac{1}{\alpha_c} \frac{\partial (\phi s_l \rho_l)}{\partial t} + \nabla \cdot \left\{ \rho_l \left[-\beta_c \frac{Kk_{rl}}{\mu_l} (\nabla p_l + \gamma_l \nabla Z) \right] \right\} = \frac{\rho_{lsc} q_{lsc}}{V_b}. \quad (3.33)$$

Dividindo-se as equações resultantes pelas massas específicas de cada fase nas condições padrão obtemos:

$$\frac{1}{\alpha_c} \frac{\partial}{\partial t} \left(\frac{\phi s_l}{B_l} \right) - \nabla \cdot \left\{ \left[\beta_c \frac{Kk_{rl}}{B_l \mu_l} (\nabla p_l + \gamma_l \nabla Z) \right] \right\} = \frac{q_{lsc}}{V_b} \quad (3.34)$$

O modelo de escoamento aqui estudado supõe que os fluidos são incompressíveis e que o meio poroso é rígido. Essas duas condições implicam em um campo escalar de porosidades que não varia em função do tempo, um fator volume de formação igual à unidade para cada fase e valores de viscosidades constantes para cada fase.

Em cada poro, devido a diferença de molhabilidade preferencial da rocha, a um dos fluidos, tem-se um encurvamento da superfície de contato entre o óleo e a água, a esse fenômeno dá-se o nome de capilaridade. A Figura 3.6 mostra um exemplo dos diferentes tipos de curvas de pressão capilar.

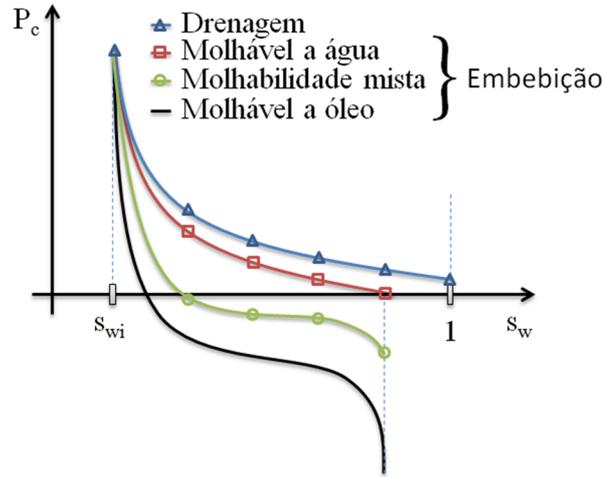


Figura 3.6: Curvas de pressão capilar [29].

A obtenção em laboratório das curvas de capilaridade é feita segundo os processos de drenagem e embebição [33], assim como na permeabilidade

relativa. Dependendo da molhabilidade da rocha tem-se diferentes tipos de curvas de pressão capilar para o processo de embebição.

Para a solução das equações a curva que deve ser usada é a curva referente ao processo de embebição enquanto que para se calcular as condições iniciais de saturação de água utiliza-se a curva de drenagem. A capilaridade afeta diretamente os valores das pressões de cada fase no contato e pode ser escrita como uma relação entre as pressões de cada fase como mostra a Equação (3.35):

$$p_c(s_w) = p_o - p_w \quad (3.35)$$

Com os campos de pressão dos componentes óleo e água são obtidos os valores de pressão capilar de cada ponto, e com a curva de pressão capilar referente ao processo de embebição são calculados os valores de saturação através da função inversa da curva de pressão capilar. Esses valores de saturação calculados são importantes para definir os valores de saturação na região de transição do componente óleo para o componente água como mostrado na Figura 3.7.

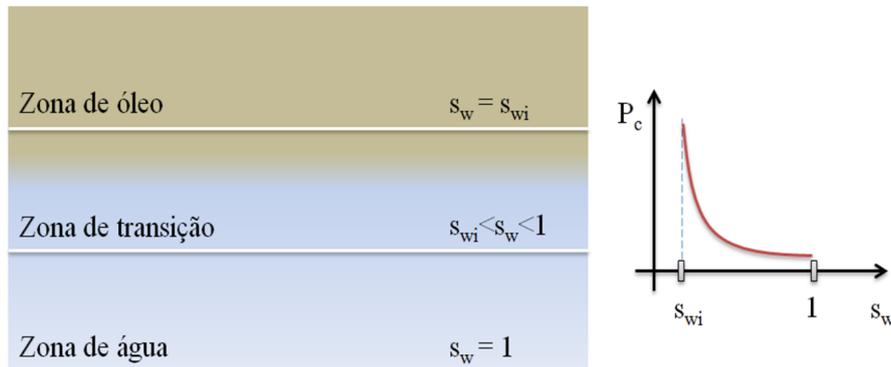


Figura 3.7: Saturação na zona de transição [29].

Uma aproximação válida que é comumente empregada é a de assumir que a formação rochosa possui poros largos e distribuição de porosidade uniforme, neste caso a região de transição seria muito pequena e pode ser desprezada devido às dimensões do problema.

Usando a Equação (3.35) pode-se reescrever as equações de saturação de cada fase em função apenas da pressão da fase óleo. Essa operação faz-se necessária para a obtenção posterior da equação de pressão que, de acordo com o método IMPES, deve ficar em função da pressão de apenas uma das

fases como mostrado nas Equações 3.36 e 3.37:

$$\frac{\phi}{\alpha_c} \frac{\partial s_o}{\partial t} = \nabla \cdot \left\{ \left[\beta_c \frac{K k_{ro}}{\mu_o} (\nabla p_o + \gamma_o \nabla Z) \right] \right\} + \frac{q_{osc}}{V_b} \quad (3.36)$$

para a fase óleo e:

$$\frac{\phi}{\alpha_c} \frac{\partial s_w}{\partial t} = \nabla \cdot \left\{ \left[\beta_c \frac{K k_{rw}(s_w)}{\mu_w} (\nabla p_o - \nabla p_c(s_w) + \gamma_w \nabla Z) \right] \right\} + \frac{q_{wsc}}{V_b} \quad (3.37)$$

para a fase água.

Equação de Pressão

A equação de pressão pode ser obtida somando-se as Equações 3.36 e 3.37. De acordo com a Equação (3.28) o termo da derivada temporal anula-se, dessa forma obtém-se:

$$\nabla \cdot \left\{ \beta_c K \left[\left(\frac{k_{ro}}{\mu_o} + \frac{k_{rw}}{\mu_w} \right) \nabla p_o - \frac{k_{rw}}{\mu_w} \nabla p_c + \left(\frac{k_{ro}}{\mu_o} \gamma_o + \frac{k_{rw}}{\mu_w} \gamma_w \right) \nabla Z \right] \right\} = -Q \quad (3.38)$$

ou:

$$\nabla \cdot (T_{po} \nabla p_o) = \nabla \cdot (T_{pc} \nabla p_c) - \nabla \cdot (T_z \nabla Z) - Q \quad (3.39)$$

onde:

$$Q = \frac{q_{wsc}}{V_b} + \frac{q_{osc}}{V_b} \quad (3.40)$$

$$T_{po} = \beta_c K \left(\frac{k_{ro}}{\mu_o} + \frac{k_{rw}}{\mu_w} \right) \quad (3.41)$$

$$T_{pc} = \beta_c K \left(\frac{k_{rw}}{\mu_w} \right) \quad (3.42)$$

$$T_z = \beta_c K \left(\frac{k_{ro}}{\mu_o} \gamma_o + \frac{k_{rw}}{\mu_w} \gamma_w \right) \quad (3.43)$$

essa representação através da separação de cada operador diferencial aplicado aos campos é feita para ilustrar a forma como são escritas as equações no OpenFOAM que serão mostradas no capítulo 5.

As equações do modelo de escoamento multifásico em meios porosos podem ser expressas através da formulação parabólica ou hiperbólica [9] [12]. As equações apresentadas neste capítulo são referentes à formulação parabólica. Nesta formulação o termo de difusão da saturação é representado pela derivada segunda da pressão capilar que é função da saturação. A formulação dita parabólica apresenta caráter parabólico e hiperbólico quando

consideram-se os efeitos da pressão capilar e hiperbólico quando é desprezado o termo de pressão capilar. Na formulação hiperbólica o equacionamento é feito através da velocidade das fases e de uma variável de velocidade total, tornando a equação de saturação uma equação hiperbólica.

Termos de fonte

Para o cálculo dos termos de fonte do problema faz-se necessário o uso de uma relação para o acoplamento das pressões no fundo do poço com as pressões do reservatório, o modelo adotado foi o modelo de Peaceman mostrado em [34] e [35]. Para poços produtores e injetores o tratamento faz-se de maneira distinta. Em poços injetores os valores de vazão ou pressão de fundo são especificados, em poços produtores, devido ao modelo incompressível aqui adotado, a vazão volumétrica total dos fluidos deve ser a mesma vazão volumétrica injetada sendo a pressão de fundo calculada a partir dos outros parâmetros. Para o acoplamento é utilizada a relação descrita pela Equação (3.44).

$$q_l = IP_l (p_e - p_{wf}) \quad (3.44)$$

onde IP representa o índice de produtividade, p_{wf} a pressão no fundo do poço na profundidade em que o poço encontra-se aberto para fluxo e p_e representa a pressão estática do reservatório na mesma profundidade. Para o caso aqui estudado será desprezada a capilaridade, sendo neste caso, a pressão estática igual à pressão das fases. No caso de um poço injetor a variável IP é chamada de II (Índice de Injetividade). Para o caso de um poço vertical o IP para a fase l é calculado segundo a Equação (3.45):

$$IP_l = \frac{-2\pi\beta_c k_{r_l} k_h h}{\mu_l B_l [\ln(r_e/r_w) + s]} \quad (3.45)$$

onde h é a espessura do intervalo aberto para fluxo no poço de raio r_w e k_h representa a permeabilidade areal calculada como uma média geométrica das permeabilidades nas direções x e y . O escalar s representa um valor que é ajustado para representar um dano causado a parte porosa aberta do intervalo em produção. O valor de r_e , raio de drenagem, é calculado em função da discretização através da seguinte expressão:

$$r_e = 0.28 \frac{\left\{ \left[(k_y/k_x)^{1/2} \Delta x^2 \right] + \left[(k_x/k_y)^{1/2} \Delta y^2 \right] \right\}^{1/2}}{\left((k_y/k_x)^{1/4} + (k_x/k_y)^{1/4} \right)} \quad (3.46)$$

A vazão de cada fase no poço produtor é calculada a partir da função de fluxo fracional f_l [9], calculada em função da velocidade das fases [36],

que desprezando-se a capilaridade e os efeitos gravitacionais pode ser escrito segundo a Equação (3.47):

$$f_l = \frac{\lambda_l}{\lambda_o + \lambda_w} \quad (3.47)$$

ou

$$f_l = \frac{IP_l}{IP_w + IP_o} = \frac{IP_l}{IP} \quad (3.48)$$

onde λ_l representa a mobilidade da fase l e é calculado segundo a Equação (3.49) e IP representa o índice de produtividade total do poço.

$$\lambda_l = \frac{kk_{rl}}{\mu_l B_l} \quad (3.49)$$

Para mostrar como é feito o acoplamento pode-se, como exemplo, considerar a situação com apenas um poço produtor e um injetor, escrevendo para o poço injetor a seguinte relação:

$$q_{inj} = II (p_e - p_{wf}) \quad (3.50)$$

essa relação considera que toda a mobilidade da fase água é igual a mobilidade total dos fluidos na área injetada. Assumindo que a permeabilidade areal do poço injetor é igual a do poço produtor, ambos com fator de *skin* nulo. Pode-se reescrever a Equação 3.50 como:

$$q_{inj} = IP (p_e - p_{wf}) \quad (3.51)$$

Para um poço produtor pode-se escrever a vazão volumétrica da seguinte forma:

$$q_p = q_w + q_o = IP_w (p_e - p_{wf}) + IP_o (p_e - p_{wf}) \quad (3.52)$$

multiplicando e dividindo a Equação (3.52) pelo IP total do poço e substituindo 3.51 na expressão resultante temos:

$$q_p = IP_w \frac{IP}{IP} (p_e - p_{wf}) + IP_o \frac{IP}{IP} (p_e - p_{wf}) = f_w q_{inj} + f_o q_{inj} \quad (3.53)$$

Assim pode-se calcular a vazão de cada fase no poço produtor como uma fração da vazão de injeção. Dado o valor da pressão de fundo do poço produtor, com o valor da vazão da fase óleo pode-se, através da Equação 3.44, encontrar a pressão estática p_e . Esta pressão será usada como pressão de referência para a solução da equação de pressão 3.39. A restrição no valor da pressão de fundo do poço produtor deve-se a condição pressão de separação ideal dos fluidos nas instalações de superfície.

Capítulo 4

Modelagem Computacional

O OpenFOAM usa o método dos volumes finitos para discretizar os operadores diferenciais. O método dos volumes finitos é um método para representar e avaliar equações diferenciais parciais na forma de equações algébricas [27] [38]. São feitas integrais de volume nos termos de divergência e, através do teorema da divergência, esses termos são convertidos para integrais de superfície. Neste método o valor da diferença entre vazão mássica que entra em um volume de controle e a vazão mássica que deixa o mesmo é idêntico ao valor acumulado no volume, garantindo assim a conservação numérica do método. Esta propriedade é importante em um método numérico pois garante que nenhuma quantidade de fluxo integral surja ou desapareça do domínio devido ao método. Para a solução das equações estudadas foi empregado o método dos volumes finitos usando malhas estruturadas, que é equivalente ao método das diferenças finitas baseadas em volumes de controle que foi o método proposto para este trabalho. Para uma melhor compreensão dos métodos utilizados neste ambiente são abordadas resumidamente a seguir a forma como são discretizados o domínio e as equações resolvidas usando o OpenFOAM. Para uma descrição mais detalhada é recomendado a leitura do trabalho feito por Hrvoje Jasak [25] que foi o criador do sistema.

4.1 Discretização do domínio

No OpenFOAM os volumes de controle podem assumir diferentes geometrias, de modo que ao escrever uma aplicação no OpenFOAM pode-se usar diferentes geometrias para a discretização do domínio sem que seja necessário mudar a aplicação. Uma mesma aplicação pode simular um caso que usa uma malha estruturada ou uma malha não estruturada. O processo de discretização pode ser dividido em dois passos: a discretização do domínio

da solução e a discretização da equação. A discretização do domínio produz uma relação entre a geometria do contínuo que se quer estudar e os pontos onde a equação discretizada será resolvida. No OpenFOAM para cada face são armazenadas as coordenadas dos vértices e do centroide, o volume e os vetores que orientam a direção normal de cada face. A Figura 4.1 mostra um exemplo de volume de controle.

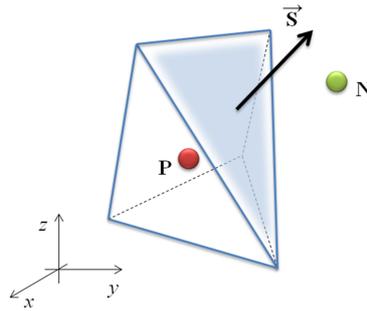


Figura 4.1: Volume de controle.

O ponto P encontra-se no centroide do volume de controle de modo que:

$$\int_V (x - x_p) dV = 0 \quad (4.1)$$

O vetor \vec{S} , normal a face destacada, encontra-se no centroide da mesma e é construído em cada face do volume de controle de modo que sua norma euclidiana equivale a área da face que o contém. O ponto N representa o centroide de um volume de controle de uma célula vizinha. A discretização do domínio no OpenFOAM pode ser feita usando-se diferentes tipos de células, sem que precise tratar explicitamente os tipos células nas aplicações desenvolvidas. Há ainda a possibilidade da definição de um novo tipo de célula que pode ser especificada pelo usuário, fornecendo-se a sua geometria. A Figura 4.2 mostra a geometria dos diferentes tipos de células que podem ser usados para discretizar o domínio no OpenFOAM. Além dos tipos de elementos mostrados é possível ainda que o usuário crie um elemento diferente e informe as coordenadas dos vértices em um arquivo de configuração do sistema.

4.2 Discretização das equações

No OpenFOAM a discretização dos termos de uma dada equação é feita separadamente, podendo-se escolher que tipo de esquema será aplicado a cada

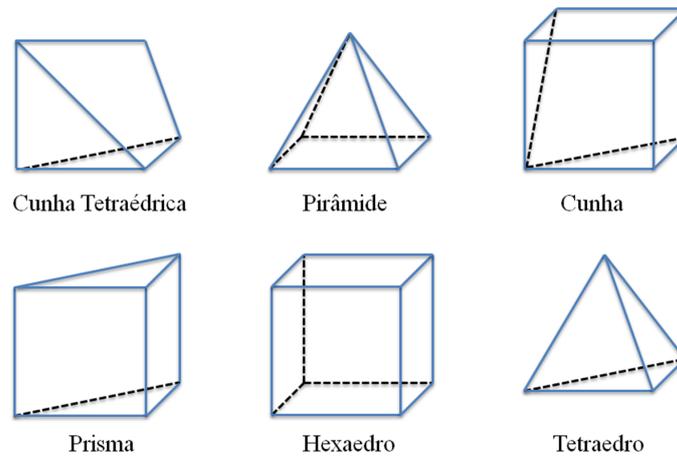


Figura 4.2: Tipos de volume de controle do OpenFOAM.

operador [2]. Uma das vantagens de utilizar este ambiente é a praticidade obtida ao tratar isoladamente a discretização de cada termo de uma dada equação. Por exemplo, para a discretização da Equação (4.2):

$$\frac{\partial \phi}{\partial t} + \nabla \cdot (U\phi) - \nabla \cdot (\Gamma_\phi \nabla \phi) = S_\phi(\phi) \quad (4.2)$$

podemos usar a seguinte representação escrita em C++ usando o OpenFOAM:

```

1 fvScalarMatrix PhiEqn
2 (
3     fvm::ddt(phi)
4     + fvc::div(U, phi)
5     - fvc::laplacian(Gamma, phi)
6     ==
7     S
8 )

```

No trecho de código acima cada operador diferencial é discretizado separadamente e a ação conjunta destes operadores discretos, compõem a matriz de discretização da equação a ser solucionada. Em cada um dos operadores pode-se usar uma formulação explícita ou implícita, usando-se os métodos das classes *fvc* (finite volume calculus) e *fvm* (finite volume method) respectivamente.

4.2.1 Discretização dos operadores de difusão

Os operadores de difusão são relacionados aos termos de derivada segunda no espaço, como o operador mostrado na Equação (4.3) que atua sobre a

grandeza primitiva ϕ ,

$$\nabla \cdot (\Gamma_\phi \nabla \phi) \quad (4.3)$$

onde Γ_ϕ é um campo tensorial com tensores de segunda ordem, associados aos centros dos volumes de controle cujo efeito de sua interpolação para as faces será discutida adiante. A discretização usando o método dos volumes finitos é feita integrando-se a Equação (4.3) e assumindo-se um volume de controle com geometria poliédrica, desta forma temos que a integral sobre o volume de controle pode ser escrita, fazendo uso do teorema da divergência, como:

$$\int_V \nabla \cdot (\Gamma_\phi \nabla \phi) dV = \sum_f \vec{S} \cdot (\Gamma_\phi \nabla \phi)_f = \sum_f (\Gamma_\phi)_f \vec{S} \cdot (\nabla \phi)_f \quad (4.4)$$

A malha utilizada no simulador desenvolvido é estruturada composta de volumes de controle com geometria hexaédrica, neste caso, se a malha for ortogonal, os vetores \vec{S} e \vec{d} mostrados na Figura 4.3 são paralelos, então podemos escrever:

$$\vec{S} \cdot (\nabla \phi)_f = |\vec{S}| \frac{\phi_N - \phi_P}{|\vec{d}|} \quad (4.5)$$

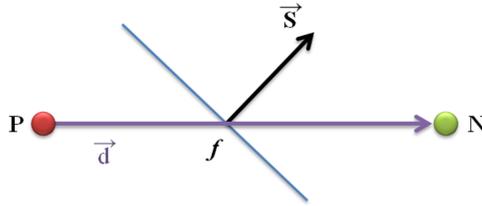


Figura 4.3: Vetores \vec{S} e \vec{d} para uma malha não ortogonal [25].

No caso de malhas não ortogonais, o OpenFOAM dispõe de funções para a correção dos gradientes. A aplicação implementada atualmente funciona com malhas estruturadas mas, com a inclusão dos corretores não ortogonais do sistema, podem ser usados diferentes tipos de células.

4.2.2 Discretização dos operadores de advecção

Os operadores de advecção são relacionados aos termos de derivada primeira no espaço, como o operador mostrado na Equação (4.6) que atua sobre a grandeza primitiva ϕ ,

$$\nabla \cdot (U\phi), \quad (4.6)$$

onde ϕ é um campo escalar definido no centro dos volumes de controle e U é um campo vetorial. A discretização é feita integrando-se a Equação (4.6) e

assumindo-se um volume de controle com geometria poliédrica, assim através do teorema da divergência podemos escrever:

$$\int_V \nabla \cdot (U\phi) dV = \sum_f \vec{S} \cdot (U\phi)_f = \sum_f (\vec{S} \cdot (U))_f \phi_f = \sum_f F\phi_f \quad (4.7)$$

onde F na Equação (4.7) representa o fluxo através de cada face do volume de controle e seus valores são interpolados para cada face. O modo como interpola-se os valores de ϕ_f em cada face irá definir o esquema de convecção usado. A Figura 4.4 mostra um exemplo de interpolação linear, que pode ser descrita como:

$$(\phi_f)_{DC} = \alpha\phi_P + (1 - \alpha)\phi_N \quad (4.8)$$

Usando-se a interpolação linear temos que o fator de interpolação α é dado por:

$$\alpha = \frac{fN}{|\vec{d}|} \quad (4.9)$$

que representa o esquema de diferenças centradas de segunda ordem, o valor

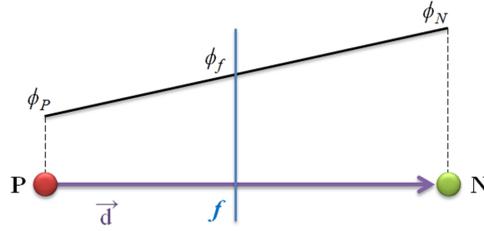


Figura 4.4: Interpolação linear [25].

de ϕ_f pode também ser avaliado nas faces de acordo com a direção do fluxo que passa pela face, F , segundo a relação:

$$(\phi_f)_{UP} = \begin{cases} \phi_P, & F \geq 0 \\ \phi_N, & F < 0. \end{cases} \quad (4.10)$$

que representa o esquema de diferenças *Upwind* de primeira ordem. Há ainda um terceiro caso onde podemos combinar os dois esquemas anteriores e obter um novo esquema chamado de *Blended Differencing*, que pode ser obtido pela relação:

$$\phi_f = \gamma (\phi_f)_{UP} + (1 - \gamma) (\phi_f)_{DC} \quad (4.11)$$

O fator γ , $0 \leq \gamma \leq 1$, na Equação (4.11) determina a quantidade de difusão numérica que será introduzida no método. Além dos esquemas de interpolação mostrados o OpenFOAM disponibiliza 70 esquemas de interpolação na versão 2.0.0 que podem ser usados pela aplicação desenvolvida.

4.2.3 Discretização temporal

Os operadores de derivada temporal, em problemas transientes, são discretizados no OpenFOAM segundo diferentes técnicas, que, incorporadas a formulação escolhida na discretização dos termos de derivadas espaciais, formarão o método de discretização da equação. Dependendo do método de discretização do operador de derivada temporal escolhido serão armazenados os valores dos campos em diferentes instantes de tempo. A discretização pode ser feita de duas formas:

Método Euler Implícito, que armazena os valores dos campos no novo passo de tempo, $n + 1$, e no passo anterior n . A discretização é feita segundo a Equação (4.12):

$$\frac{\partial}{\partial t} \int_V \rho \phi dV = \frac{(\rho_P \phi_P)^{n+1} - (\rho_P \phi_P)^n}{\Delta t} \quad (4.12)$$

Método de Diferenciação para Trás, que armazena os valores do novo passo de tempo, $n + 1$, e nos dois passos anteriores n e $n - 1$. A discretização é feita segundo a Equação (4.13):

$$\frac{\partial}{\partial t} \int_V \rho \phi dV = \frac{3(\rho_P \phi_P)^{n+1} - 4(\rho_P \phi_P)^n + (\rho_P \phi_P)^{n-1}}{2\Delta t} \quad (4.13)$$

Seja A a matriz que representa a ação conjunta dos operadores diferenciais como difusão e advecção. Aplicando o método dos volumes finitos tem-se:

$$\int_t^{t+\Delta t} \left[\frac{\partial}{\partial t} \int_V (\rho \phi) dV + \int_V (A\phi) dV \right] dt = 0 \quad (4.14)$$

Usando-se o método Euler implícito, Equação (4.12), para discretizar o termo da derivada temporal tem-se:

$$\begin{aligned} \int_t^{t+\Delta t} \left[\frac{\partial}{\partial t} \int_V (\rho \phi) dV \right] dt &= \int_t^{t+\Delta t} \frac{(\rho_P \phi_P)^{n+1} - (\rho_P \phi_P)^n}{\Delta t} dt \\ \int_t^{t+\Delta t} \left[\frac{\partial}{\partial t} \int_V (\rho \phi) dV \right] dt &= (\rho_P \phi_P)^{n+1} - (\rho_P \phi_P)^n \end{aligned} \quad (4.15)$$

O segundo termo da Equação (4.14) pode ser escrito como:

$$\int_t^{t+\Delta t} \left[\int_V (A\phi) dV \right] dt = \int_t^{t+\Delta t} A^d \phi dt \quad (4.16)$$

onde A^d representa o operador diferencial discreto resultante da discretização do operador A feita usando o método dos volumes finitos. As diferentes formas de avaliar a integral na Equação (4.16) em combinação com os métodos

de discretização da derivada temporal forma as diferentes formulações para a discretização da equação diferencial. Na formulação explícita o operador discreto é aplicado aos valores da função no passo de tempo atual, este método de discretização é feito usando-se a classe *fvc* e é descrito pela Equação (4.17).

$$\int_t^{t+\Delta t} \left[\int_V (A\phi) dV \right] dt = A^d \phi^n \Delta t \quad (4.17)$$

Na formulação implícita o operador discreto é aplicado aos valores da função no passo de tempo posterior, este método de discretização é feito usando-se a classe *fvm* e é descrito pela Equação (4.18).

$$\int_t^{t+\Delta t} \left[\int_V (A\phi) dV \right] dt = A^d \phi^{n+1} \Delta t \quad (4.18)$$

Há ainda uma terceira formulação, Crank Nicholson, que pode ser obtida usando-se os dois esquemas anteriores descrito pela Equação 4.19.

$$\int_t^{t+\Delta t} \left[\int_V (A\phi) dV \right] dt = A^d \left(\frac{\phi^{n+1} + \phi^n}{2} \right) \Delta t \quad (4.19)$$

Na sequência temos a representação em C++ para a solução da equação de difusão de calor 2.2 usando-se o método Crank Nicholson com o OpenFOAM.

```

1 fvScalarMatrix Teqn
2 (
3     0.5*fvc::laplacian(T)
4     + 0.5*fvm::laplacian(T)
5     + q
6     ==
7     (1/alpha)*fvm::ddt(T)
8 )

```

O método de discretização do termo da derivada temporal é escolhido no caso a ser simulado. Pode-se ainda obter diferentes técnicas combinando os operadores descritos anteriormente.

4.3 Método IMPES

O método IMPES (*Implicit Pressure Explicit Saturation*) é uma técnica empregada na solução numérica do problema de escoamento multifásico. No esquema IMPES é feito um desacoplamento na equação de conservação de massa entre a pressão e a saturação, obtendo-se uma equação para cada variável. É obtida uma equação de pressão que pode ser em função da pressão

de uma das fases ou em função da pressão média entre as fases. A equação de pressão é resolvida implicitamente seguida da solução explícita da equação de saturação. Para obter a equação de pressão são somadas as equações de conservação de massa para cada fase e através da pressão capilar a equação é escrita em função de apenas da pressão de uma das fases. Para a solução da equação de pressão obtida são necessários os valores das variáveis da equação no passo de tempo $n+1$, como esses valores ainda não são conhecidos eles são calculados no passo de tempo n e a equação de pressão é resolvida implicitamente. Após a solução da equação de pressão, é resolvida a equação de saturação para uma das fases usando o campo de pressões obtido.

4.3.1 Tratamento das não linearidades

A equação de pressão possui termos de permeabilidade relativa e pressão capilar que dependem do valor da saturação no novo passo de tempo e termos que variam com a pressão. Para um escoamento incompressível os termos que variam com a pressão permanecem constantes ao longo do tempo, mas os valores que dependem da saturação precisam ser obtidos para o cálculo da distribuição de pressões. No método IMPES a equação de pressão (3.39) é linearizada explicitamente como mostra a Equação (4.20):

$$\nabla \cdot (T_{po}^n \nabla p_o^{n+1}) = \nabla \cdot (T_{pc}^n \nabla p_c^n) - \nabla \cdot (T_Z^n \nabla Z) - Q^n \quad (4.20)$$

Após a solução da equação de pressão obtemos a saturação, Equação (3.37), a partir da Equação (4.21):

$$\frac{\phi}{\alpha_c} \frac{\partial s_w}{\partial t} = \nabla \cdot \left\{ \left[\beta_c \frac{K k_{rw}(s_w^n)}{\mu_w} (\nabla p_o^{n+1} - \nabla p_c(s_w^n) + \gamma_w \nabla Z) \right] \right\} + \frac{q_{wsc}^n}{V_b} \quad (4.21)$$

que é resolvida explicitamente e pode ser reescrita como:

$$\frac{\phi}{\alpha_c} \frac{\partial s_w}{\partial t} = \nabla \cdot (T_{pc}^n \nabla p_o^{n+1}) - \nabla \cdot (T_{pc}^n \nabla p_c^n) + \nabla \cdot (T_{zw}^n \nabla Z) + \frac{q_{wsc}^n}{V_b} \quad (4.22)$$

onde T_{pc} é dado pela Equação (3.42).

Para a solução das equações é utilizado o método dos volumes finitos, que para discretizar os operadores necessita dos valores das transmissibilidades nas faces dos volumes de controle como mostrado na Equação (4.23). O campo de permeabilidades absolutas é interpolado para as faces usando-se uma média harmônica entre os valores dos centros das células vizinhas como mostrado na malha estruturada composta por hexaedros da a Figura 4.5.

Na Figura 4.5 k_f representa a permeabilidade média na face entre os pontos N e P . Essa média pode ser: aritmética, geométrica ou harmônica.

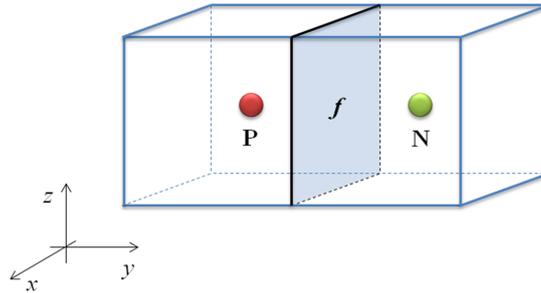


Figura 4.5: Representação de duas células vizinhas.

A interpolação mais utilizada é a média harmônica que pode ser calculada segundo a expressão:

$$\frac{1}{k_f} = \frac{1}{2} \left(\frac{1}{k_P} + \frac{1}{k_N} \right) \quad (4.23)$$

O OpenFOAM dispõe de muitas técnicas de interpolação, para decidir qual a melhor técnica a ser empregada posteriormente serão avaliadas as diferentes propostas de interpolação, e o efeito de cada uma delas na solução dos fenômenos de escoamento em meios porosos. No capítulo 5 será discutido como foi feita a interpolação do campo de permeabilidades absolutas já que na versão atual do OpenFOAM não existe um esquema de interpolação harmônica implementado. No caso do campo de permeabilidades relativas a interpolação é feita de acordo com o esquema escolhido no diretório do caso. Para todos os problemas resolvidos na verificação do simulador foi utilizado o esquema *upwind*. A verificação e a comparação entre diferentes esquemas de interpolação para a permeabilidade relativa não serão feitas neste texto mas estarão presentes em trabalhos futuros.

Capítulo 5

Aplicação IMPES

Conforme mencionado anteriormente, para a implementação da aplicação foi escolhido o método IMPES, que é um método numérico de solução aplicado ao modelo Black-oil. Na modelagem do escoamento em meios porosos existem dois modelos de fluxo que são mais usados, o modelo Beta e o Composicional. No modelo Beta admite-se que o fluido é composto de dois pseudocomponentes, líquido e vapor, o balanço de massa é aplicado para cada pseudocomponente e assume-se que as propriedades das fases são funções da pressão e da temperatura. O modelo chamado Beta é comumente chamado de modelo Black-oil em que o gás pode estar presente em ambas as fases enquanto o componente óleo fica restrito à fase líquida, neste modelo são assumidos três componentes: água, óleo e gás [9]. Os componentes água e óleo são assumidos ser imiscíveis e o gás é considerado ser solúvel no óleo. O comportamento PVT do sistema formado pelos componentes água, óleo e gás é definido pelas variáveis B_o , B_w e B_g sendo a transferência de massa entre óleo e gás descrita pela razão de solubilidade R_g . O programa implementado neste trabalho utiliza o método IMPES para resolver um problema que representa uma simplificação do modelo Black-oil onde apenas a fase líquida está presente. A escolha de um modelo simplificado não restringe a expansão do simulador a um modelo mais complexo, já que a incorporação de outras físicas no programa é facilitada devido a orientação a objetos que é característica do ambiente.

Após a implementação do modelo monofásico, mencionado no Capítulo 2, foi desenvolvida a aplicação que simula o escoamento bifásico em meios porosos. Basicamente o modelo monofásico consiste na solução de uma equação elíptica, a equação de pressão, equivalente a resolvida nos problemas de difusão de calor. A implementação do simulador bifásico usa como base o modelo monofásico, pois a equação de pressão resolvida no modelo bifásico é similar a do modelo monofásico. De maneira simplificada, neste modelo será

resolvida apenas uma equação adicional com relação ao modelo monofásico, a equação de saturação, que é uma equação de advecção-difusão.

Neste capítulo será apresentada a aplicação desenvolvida no OpenFOAM. Cada trecho de código apresentado no capítulo foi retirado da aplicação com o objetivo de explicar o funcionamento do sistema de forma prática. A apresentação da aplicação é feita segundo a ordem de execução do programa, para um melhor entendimento de como é feito um programa no ambiente. A fim de se ressaltar melhores práticas de programação com o ambiente são apresentadas algumas características do sistema que geraram dificuldades na implementação e o modo como essas dificuldades foram contornadas.

5.1 Criação da aplicação

A implementação de aplicações no OpenFoam, como já foi exposto anteriormente, se dá através do uso da representação tensorial das equações através das classes que representam os operadores diferenciais. O IMPES foi a técnica escolhida para a solução das equações resultantes do problema de escoamento bifásico. A discretização e os métodos de interpolação escolhidos seguem o esquema apresentado no principal livro que orientou este trabalho [20]. Como a maioria dos artifícios numéricos utilizados para a linearização das equações e os esquemas de interpolação usados é feita na forma discretizada das equações, é necessária uma construção de equações diferenciais na forma tensorial com esquemas de interpolação para cada termo, de cada operador diferencial, que seja equivalente a apresentada após a aplicação dos métodos de linearização aplicados nas equações discretizadas.

A criação dos campos de variáveis envolvidas nas equações presentes no modelo implementado é feita através de classes específicas do ambiente. As variáveis são definidas na aplicação e arquivos contendo suas condições de contorno, seus valores iniciais e as unidades são criadas no diretório do caso.

O trecho de código a seguir mostra a definição de um campo tensorial na aplicação. A classe *volTensorField* cria um campo tensorial tridimensional que associa um tensor a cada volume de controle da malha criada. A criação é feita com a estrutura *IObject* na qual são especificados o nome da variável, em qual diretório será armazenado ou lido o campo, a malha que estará associada ao campo de tensores e dois parâmetros de leitura e escrita.

```

1  volTensorField K
2  (
3      IObject
4      (
5          "K",
6          runtime.constant(),

```

```

7         mesh ,
8         IObject :: MUST_READ ,
9         IObject :: NO_WRITE
10    ),
11    mesh
12 );
```

Para a criação de uma variável dependente do tempo deve ser especificado o diretório como *runTime.timeName()*, que representa o diretório com o nome do passo de tempo atual no caso e os parâmetros de escrita devem ser mudados a fim de que sejam gravados os valores a cada iteração. Na biblioteca existem ainda outras variáveis pré-definidas como constantes matemáticas e físicas. A tabela 5.1 mostra os campos criados na aplicação.

Tabela 5.1: Listagem dos campos criados na aplicação.

Variável	Tipo de campo
K	<i>volTensorField</i>
ac, so, sw, kro, krw, po, pw e pc	<i>volScalarField</i>
Kf, Tpc, Tpo, Tz e Tzw	<i>surfaceTensorField</i>
krof, krwf	<i>surfaceScalarField</i>

No OpenFOAM um campo do tipo *volField* é um campo definido no centro das células resultantes da discretização do domínio e um do tipo *surfaceField* é um campo definido nas faces das células. Devido à grande quantidade de dados gerados pela criação de cada campo é importante que seja criado o menor número de campos possível, a fim de reduzir a memória necessária à execução do programa.

A criação dos parâmetros escalares constantes da aplicação é feita através da classe *IOdictionary*. A classe utiliza a mesma estrutura para variáveis de entrada e saída do sistema e assim como no exemplo anterior, o diretório do arquivo contendo os parâmetros é especificado na estrutura. No código a seguir encontra-se a criação do objeto que contém os parâmetros.

```

1  IOdictionary params
2  (
3      IObject
4      (
5          "params",
6          runTime.constant(),
7          mesh,
8          IObject :: MUST_READ,
```

```

9             IObject::NO_WRITE
10         )
11 );

```

Quando a aplicação é executada no diretório do caso o programa irá reconhecer a estrutura de arquivos característica do mesmo e irá atribuir a estrutura *runTime*, os caminhos de cada diretório no caso. Após a definição da variável que aponta para o arquivo com os parâmetros, a criação de um escalar é feita como mostrado a seguir:

```
1 dimensionedScalar ALPHA_C (params.lookup("ALPHA_C"));
```

para que a variável possa ser criada é necessário que ela esteja no arquivo *params* criado no diretório do caso.

Para a obtenção dos valores das permeabilidades relativas no programa desenvolvido é usada uma interpolação feita a partir de valores fornecidos em uma tabela de pontos com os valores experimentais de permeabilidade relativa para cada saturação dada. Através desses pontos é obtida uma curva interpolada com *splines*, a qual é usada para o cálculo do campo de permeabilidades. Assim como a permeabilidade relativa, a capilaridade é uma função da saturação e na implementação do simulador foi obtida também através de uma interpolação com uso de *splines* entre valores experimentais fornecidos ao programa.

Após a definição das variáveis são lidos dois arquivos no diretório do caso, o arquivo de permeabilidades relativas e o arquivo de pressões capilares. Estes arquivos contêm os pontos que serão interpolados usando *splines* que irão gerar as curvas utilizadas pela aplicação. Como foi comentado no capítulo 3, a curva fornecida para a permeabilidade relativa deve ser a referente ao processo de embebição. Para o cálculo das condições iniciais será assumido que os poros da formação são largos e que o campo de porosidades é uniforme, podendo-se, assim, desprezar a zona de transição de saturações. A curva de pressão capilar utilizada para a solução das equações será a referente ao processo de embebição.

Para o cálculo dos termos de fluxo, os valores de permeabilidade absoluta devem ser calculados nas faces, já que o campo primeiramente criado está associado apenas ao centro dos volumes de controle. Como foi mostrado no capítulo 4 o cálculo dos valores nas faces deve ser feito através de uma média harmônica dos valores das células que compartilham a face. O OpenFOAM oferece 41 diferentes esquemas de interpolação para os valores das faces na versão 2.0.0, incluído um esquema para média harmônica. Entretanto esse esquema não se aplica a campos tensoriais. Dessa forma, foi necessária a implementação do esquema de interpolação na aplicação. Para não se perder o paralelismo característico do sistema existe uma sintaxe própria do Open-

FOAM para laços que sejam executados em paralelo utilizando as estruturas e variáveis do sistema. O trecho de código a seguir mostra um laço feito para calcular o valor das permeabilidades absolutas nas faces.

```

1  const unallocLabelList& own = mesh.owner();
2  const unallocLabelList& nei = mesh.neighbour();
3
4  forAll(Kf,i)
5  {
6    Kf[i].xx() = 2*K[own[i]].xx()*K[nei[i]].xx()/
7                (K[own[i]].xx()+K[nei[i]].xx());
8    Kf[i].yy() = 2*K[own[i]].yy()*K[nei[i]].yy()/
9                (K[own[i]].yy()+K[nei[i]].yy());
10   Kf[i].zz() = 2*K[own[i]].zz()*K[nei[i]].zz()/
11                (K[own[i]].zz()+K[nei[i]].zz());
12 }

```

Através dos métodos do objeto *mesh*, *mesh.owner()* e *mesh.neighbour()*, são obtidos os *labels* das células que compartilham cada face, com esses valores são calculadas as médias das permeabilidades nas faces. Para utilizar a sintaxe *forAll* é necessário que seja especificado o campo no qual será realizado o *loop*, como o desejado é calcular os valores nas faces como média harmônica dos centros, foi feito um laço para cada campo. O melhor modo seria utilizar *fvc::interpolate(K, "harmonic")*, que não está disponível para um campo *volTensorField*. Os campos resultantes descritos pelas equações (3.40), (3.41), (3.42) e (3.43) são calculados e em seguida é feita a solução das equações de fluxo. A representação das equações de fluxo do método IMPES descrito no capítulo 4, 4.20 e 4.22, em C++ é feita da seguinte forma:

```

1  fvScalarMatrix poEqn
2  (
3      fvm::laplacian(Tpo,po)
4      ==
5      fvc::laplacian(Tz,Z)
6      + fvc::laplacian(Tpc,pc)
7      - Q
8  );
9  poEqn.solve();

1  fvScalarMatrix swEqn
2  (
3      ac*fvm::ddt(sw) ==
4      fvc::laplacian(Tpc,po)
5      - fvc::laplacian(Tpc,pc)
6      - fvc::laplacian(Tzw,Z)
7      + Qw
8  );
9  swEqn.solve();

```

após a solução dessas equações todas as propriedades que variam com o tempo são recalculadas e um novo passo de tempo se inicia. Deve-se destacar que o sinal do termo referente ao efeito gravitacional está com o sinal diferente das Equações 4.20 e 4.22. Essa mudança de sinal se deve à orientação de eixos adotada na dedução das equações, com o eixo z apontando para baixo. Como o operador do OpenFOAM utiliza o eixo para cima é necessário trocar o sinal na equação.

5.2 Criação do caso

Como em qualquer caso que usa uma aplicação desenvolvida no OpenFOAM, deve-se seguir a estrutura de um diretório padrão de caso do sistema. A Figura 5.1 mostra a estrutura de arquivos criada. No diretório 0 são armazenadas as variáveis que tem seu valor alterado a cada passo de tempo, propriedades como porosidade e permeabilidade absoluta neste problema não variam com o tempo sendo necessária a sua definição no diretório *constant*. Ainda no diretório *constant* fica armazenado o arquivo *params*.

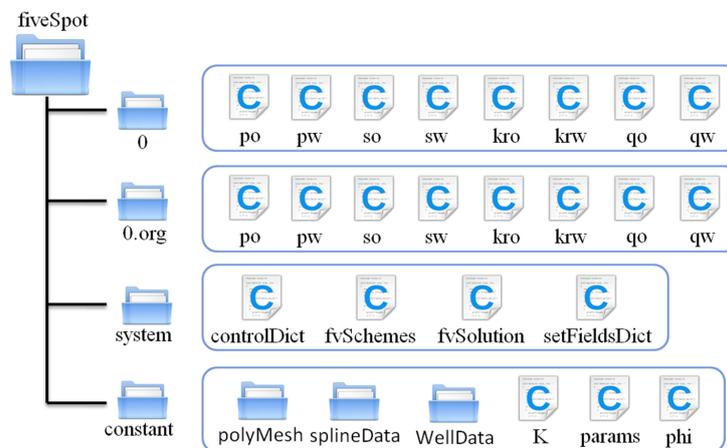


Figura 5.1: Caso que utiliza a aplicação IMPES.

O diretório *0.org* foi criado pois cada vez que executa-se o programa, os arquivos no diretório *0* são alterados e para executar novamente o programa é necessário que os arquivos estejam com as mesmas condições dos arquivos *0.org*, deste modo para facilitar a execução, os arquivos são copiados do diretório *0.org* para o diretório *0*. Não é necessária a criação do campo s_o no diretório *0* pois o mesmo pode ser calculado como $1 - s_w$, o campo foi colocado no diretório de variáveis de tempo por conveniência.

A geometria do problema de *benchmark* mais usado neste tipo de simulador é a do modelo de malha de injeção *five-spot*. Devido a simetria do problema é apenas resolvido um quarto de um elemento da malha de injeção. A Figura 5.2 ilustra o modelo. Para a criação da geometria e geração da malha foi utilizado o gerador de malhas de blocos, o *blockMesh*, nativo do sistema. Devido ao fato do OpenFOAM não possuir médias harmônicas, para campos tensoriais, a alternativa implementada apenas poder ser usada para malhas estruturadas e com espaçamento constante.

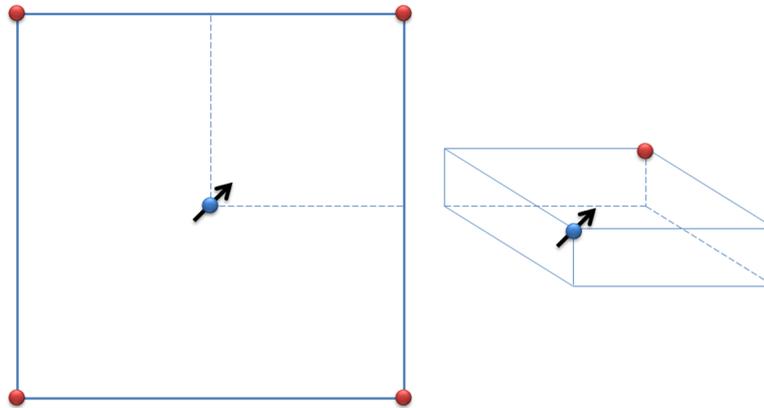


Figura 5.2: Esquema *five-spot*.

Para possibilitar a solução de problemas em que poços direcionais são usados foi implementada uma rotina para calcular a trajetória dos poços. Esse modelo apesar de ainda não estar sendo usado de fato na aplicação foi implementado com o intuito de poder calcular o perfil de pressões ao longo da coluna de um poço direcional. O cálculo das perdas de carga nos poços ainda não foi implementado, mas poderá ser feito em trabalhos futuros. A partir de pontos definidos em um arquivo no caso são interpoladas duas curvas, utilizando *splines*, uma com a projeção xz da curva e outra com a projeção yz . Com as curvas pode-se determinar quais blocos da malha são cortados pela trajetória do poço e variáveis booleanas determinam em que intervalos o poço estará aberto para o fluxo. Dependendo do modelo geológico é possível a utilização desse modelo de trajetórias para colocar termos de fonte que representam um poço com completação dupla, ou seja, que injeta em um intervalo e produz de outro. Muitas configurações são possíveis com este tipo de modelo e a ocorrência de poços de petróleo com essa configuração é bastante comum em campos de terra. A Figura 5.3 mostra um esquema do ajuste feito na trajetória de um poço injetor aberto ao fluxo em duas zonas.

Na construção do modelo de poço inclinado é necessário o acoplamento

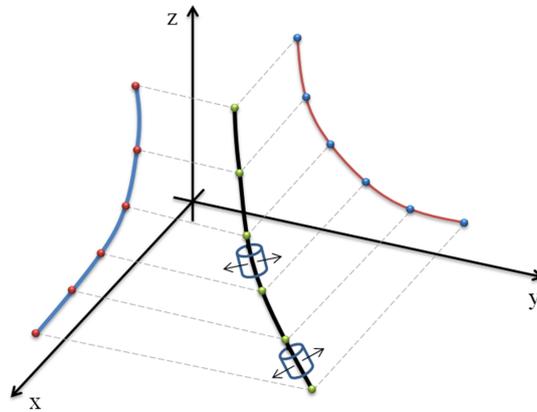


Figura 5.3: Ajuste da trajetória de um poço direcional.

das pressões especificadas no fundo do poço, com as pressões do reservatório. Para o acoplamento é utilizado o modelo de Peaceman [35] no qual a Equação (3.44) é reescrita para o bloco (i, j, k) que contém o poço como:

$$q_{i,j,k} = IP_l (p_{i,j,k} - p_{wf}) \quad (5.1)$$

Para poços verticais IP pode ser calculado pela Equação (3.45), no entanto, para poços inclinados essa equação só é válida no caso em que o campo de permeabilidades absolutas é isotrópico. Existem modelos na literatura que descrevem métodos diferentes para o cálculo do IP para o caso de poços inclinados [7] [39] [19]. A implementação desse tipo de modelo de acoplamento será abordada em trabalhos futuros.

Capítulo 6

Resultados

Para a verificação do simulador implementado foram resolvidos problemas no simulador comercial IMEX [1]. Para a comparação foi usado o modelo *oilwater* do IMEX com modelo similar ao implementado na aplicação. Foram especificados valores nulos de compressibilidades dos fluidos e da rocha com ponto de bolha fixo e as mesmas condições iniciais e de contorno. Foram resolvidos exemplos de escoamento unidimensional, com geometria plana e com geometria tridimensional. Para demonstração foi resolvido um exemplo que usa a aplicação feita com o modelo monofásico levemente compressível.

6.1 Resultados da aplicação de escoamento monofásico

As Tabelas 6.1 e 6.2 mostram as propriedades da rocha e do fluido utilizadas na solução do caso monofásico. A aplicação não foi verificada com o IMEX como a aplicação bifásica. Os resultados aqui apresentados apenas ilustram um exemplo de um caso resolvido com esta aplicação.

Tabela 6.1: Propriedades do fluido (Caso monofásico).

Propriedade	Símbolo	Valor
Fator volume de formação à P_{ref}	B_{ref}	1
Massa específica à P_{ref}	ρ_{ref}	30 kg/m^3
Viscosidade à P_{ref}	μ_{ref}	8 cp
Compressibilidade do fluido	c_f	$1.0 \times 10^{-5} \text{ psi}^{-1}$

Tabela 6.2: Propriedades da rocha (Caso monofásico).

Propriedade	Símbolo	Valor
Porosidade	ϕ	0.2
Compressibilidade da rocha	c_r	$1.0 \times 10^{-5} \text{ psi}^{-1}$
Permeabilidade absoluta	k	200 mD

Para a solução foi utilizada uma geometria hexaédrica medindo 1000 ft nas direções x e y com 100 ft na direção z . A geometria foi discretizada com blocos cúbicos com aresta medindo 20 ft e foi colocado um sumidouro nas células de um dos vértices da geometria com vazão total das células de 16 bbl/d . Foi usado um passo de tempo de 1 dia, a Figura 6.1 mostra a solução de pressão após 100 dias.

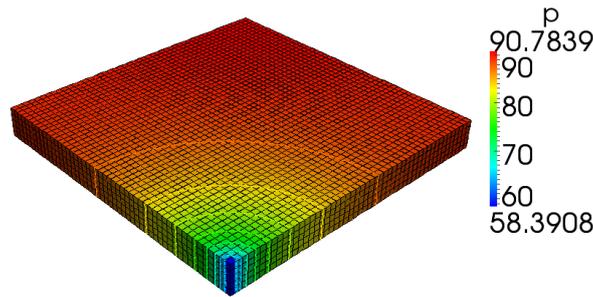


Figura 6.1: Solução de pressão do caso monofásico.

6.2 Verificação da aplicação IMPES

As propriedades da rocha utilizada em todos os exemplos a seguir estão listadas na Tabela 6.3. Os valores de compressibilidade foram atribuídos nulos apenas no IMEX já que a aplicação feita no OpenFOAM já utiliza as equações do modelo incompressível. O campo de permeabilidades relativas utilizado é homogêneo e isotrópico com o valor dado na tabela 6.3. As propriedades dos fluidos estão descritas na Tabela 6.4, assim como a compressibilidade da rocha a compressibilidade do fluido foi atribuída apenas no IMEX. As curvas de permeabilidade relativa e pressão capilar utilizadas nos exemplos estão descritas na tabela 6.5. Foram resolvidos problemas com e sem pressão capilar. Os valores na tabela de permeabilidades e pressões capilares foram utilizados pela aplicação do OpenFOAM para a construção de

Tabela 6.3: Propriedades da rocha (Caso bifásico).

Propriedade	Símbolo	Valor
Porosidade	ϕ	0.2
Compressibilidade	c_r	0.0 psi^{-1}
Permeabilidade absoluta	k	200 mD

Tabela 6.4: Propriedades dos fluidos (Caso bifásico).

Propriedade	Símbolo	Valor
Densidade do óleo	ρ_{osc}	46.2 kg/m^3
Densidade da água	ρ_{wsc}	62.2 kg/m^3
Viscosidade do óleo	μ_o	0.90 cp
Viscosidade da água	μ_w	0.30 cp
Compressibilidade do óleo	c_o	0.00
Compressibilidade da água	c_w	0.00

Tabela 6.5: Curvas de permeabilidade relativa e pressão capilar.

sw	kro	krw	pc
0.0	1.0000	0.0000	10.00
0.2	0.5625	0.0000	1.949
0.4	0.2500	0.0625	0.280
0.6	0.0625	0.2500	0.059
0.8	0.0000	0.5625	0.026
1.0	0.0000	1.0000	0.000

splines, sendo os valores calculados durante a simulação obtidos da interpolação desses dados. Como não houve diferença significativa entre as soluções com e sem pressão capilar, os exemplos a seguir mostrarão apenas as soluções sem pressão capilar.

6.2.1 Escoamento unidimensional

Para a comparação dos resultados obtidos no caso unidimensional foi utilizada a geometria mostrada na Figura 6.2 na qual foram usados 40 elementos

cúbicos com aresta de 500 *m* cada, totalizando um comprimento de 20000 *m*. Foram colocados dois poços, um injetor no centro do primeiro bloco e um produtor no centro do último bloco. A pressão no poço produtor foi fixada em 1000 *psi* e a vazão em ambos os poços em 2000 *bbl/d*. Tanto no IMEX como na aplicação foi usado um incremento de tempo de 10 dias. A Figura 6.3 mostra o perfil do campo de pressões para o instante de tempo 28105*d* que equivale a 77 anos.

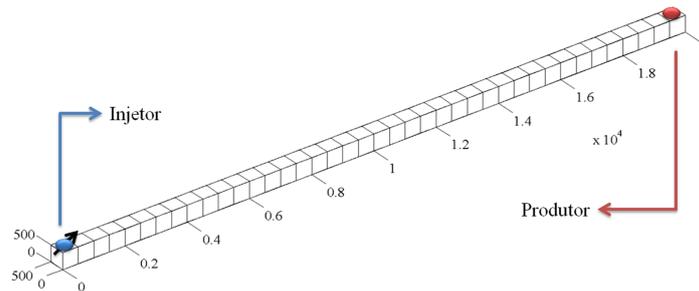


Figura 6.2: Malha usada no problema unidimensional.

Pode-se observar na Figura 6.3 que o perfil da curva obtida com o OpenFOAM é similar ao obtido no IMEX a não ser pela pressão no poço injetor que não foi honrada, com uma diferença de pouco menos de 0.99%.

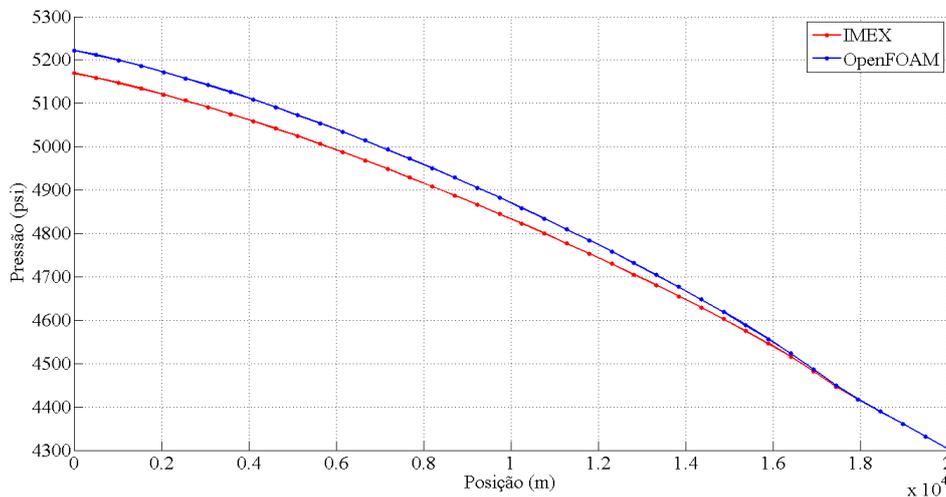


Figura 6.3: Comparação do perfil de pressão 1D.

O modelo de acoplamento de poço utilizado fixa a pressão apenas no poço produtor deixando a célula do poço injetor apenas com o termo de fonte correspondente à vazão local. Esse problema está sendo analisado e

algumas alternativas estão sendo estudadas afim de melhor representar as condições de injeção. A Figura 6.4 mostra o perfil do campo de saturações para o instante de tempo de 77 anos. Apesar da solução da equação de

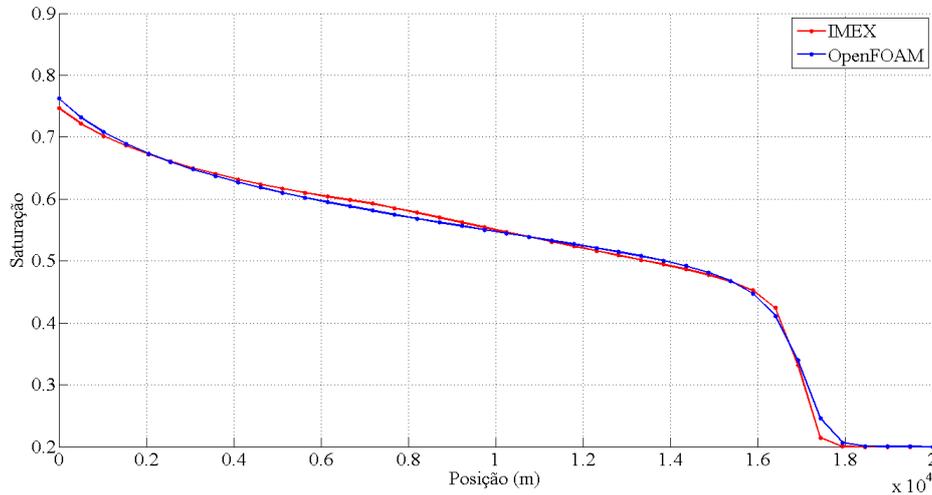


Figura 6.4: Comparação do perfil de saturação 1D.

saturação no OpenFOAM representar de maneira similar a solução obtida com o IMEX observa-se a grande difusão numérica. Esta grande difusão se deve ao esquema *upwind*, adotado na aplicação do OpenFOAM. Alternativas ao esquema de interpolação que é usado na advecção da saturação estão sendo investigadas e a implementação de outra formulação para a equação de saturação será tratada em trabalhos futuros.

6.2.2 Escoamento bidimensional

Para a comparação dos resultados em duas dimensões foi resolvido um problema equivalente ao unidimensional, com o mesmo passo de tempo, mudando-se apenas a geometria do problema para uma caso bidimensional e aumentando-se a vazão de cada poço para 20000 *bbl/d*. Nos problemas de duas ou mais dimensões usados para a comparação com o IMEX foi utilizada a condição de contorno de simetria do OpenFOAM, *symmetryPlane*, devido a característica simétrica dos problemas resolvidos. O sistema de equações resultante da discretização da equação de pressão resulta em um problema mal posto. Neste tipo de problema é necessário que se especifique um valor de pressão de referência para se resolver o sistema de equações. A pressão de referência que foi utilizada é a pressão obtida através do modelo de Peaceman, que foi apresentado no Capítulo 3. A Figura 6.5 mostra a malha

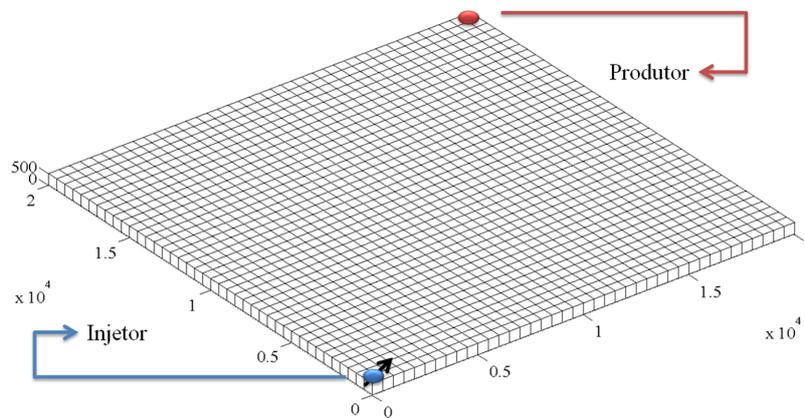


Figura 6.5: Malha usada no problema bidimensional.

utilizada na solução do problema e o posicionamento dos poços. A figura 6.6 mostra a solução da pressão no instante de tempo igual a $91250d$ que equivale à 250 anos.

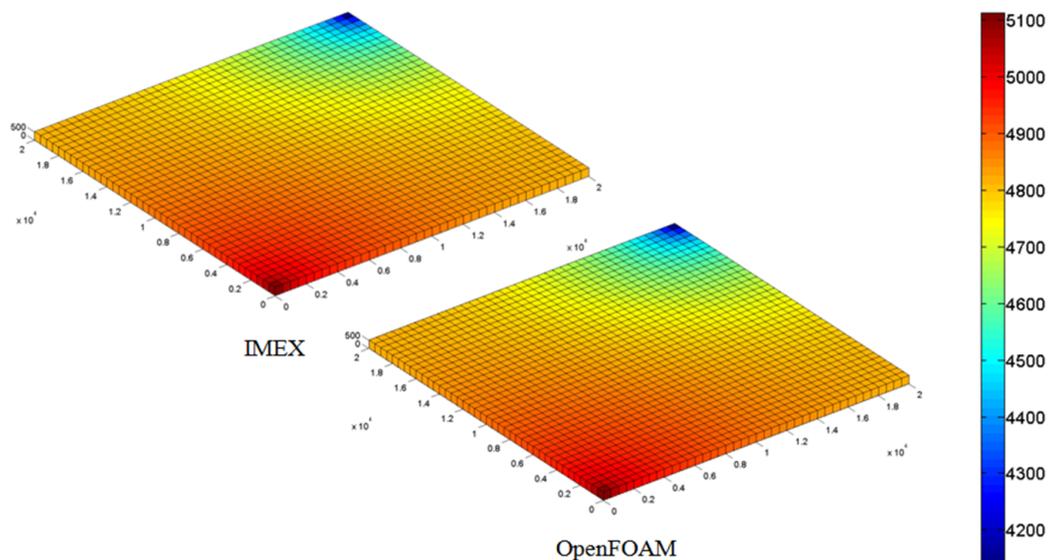


Figura 6.6: Solução da pressão no problema bidimensional.

Para uma melhor visualização da diferença entre os dois simuladores foi feita uma secção na diagonal que liga os poços. A figura 6.7 mostra as curvas de pressão na diagonal mencionada. Novamente pode-se observar que a solução com a aplicação desenvolvida é similar à obtida no IMEX.

A figura 6.8 mostra a solução da saturação para o mesmo instante de tempo. Assim como na solução de pressão uma comparação entre as duas

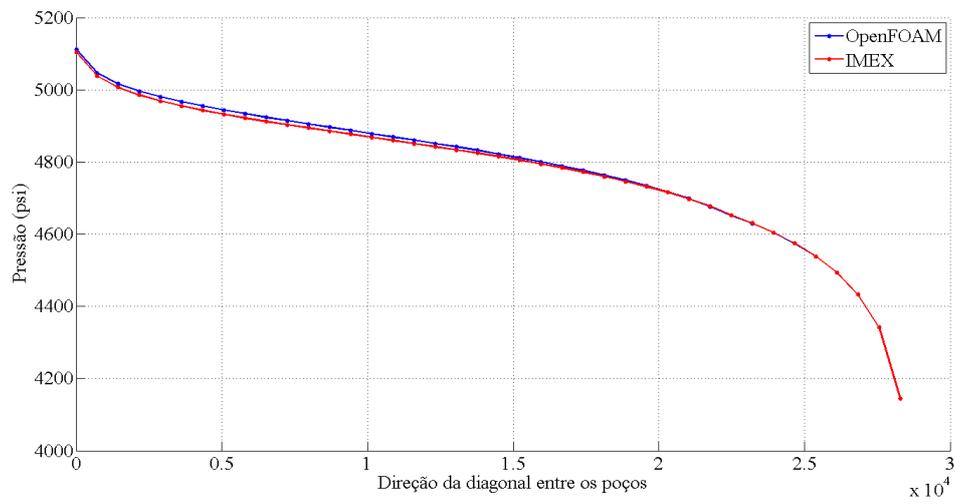


Figura 6.7: Comparação das soluções de pressão 2D na direção diagonal.

soluções foi feita ao longo da diagonal que liga os poços. A figura 6.9 mostra a comparação entre as soluções de saturação na diagonal. A solução das equações para o caso bidimensional retrata bem a solução obtida com o IMEX.

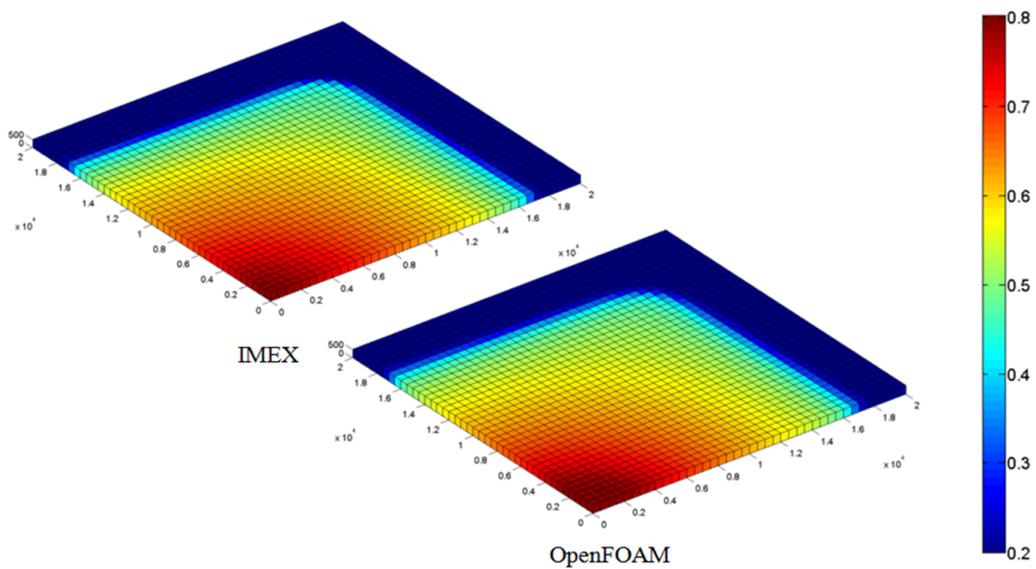


Figura 6.8: Solução da saturação do problema bidimensional.

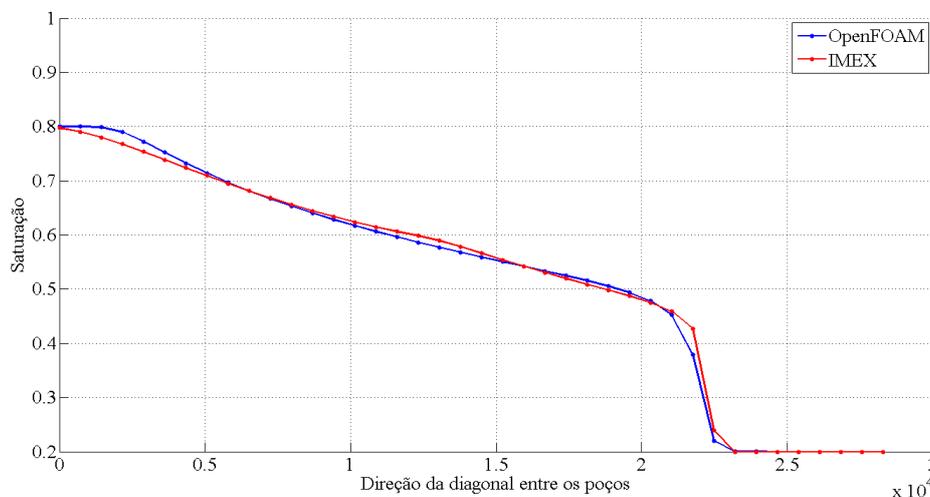


Figura 6.9: Comparação das soluções de saturação 2D na direção diagonal.

6.2.3 Escoamento tridimensional

Para a comparação no caso tridimensional foi criada uma geometria hexaédrica com 40 blocos na direção x , 40 na direção y e 5 blocos na direção z . O mesmo incremento de tempo dos casos anteriores foi utilizado em ambos os simuladores. A vazão dos poços foi fixada em $200000 \text{ bbl}/d$. Os poços injetor e produtor se encontram na primeira e na última célula do plano superior, como mostrado na figura 6.10.

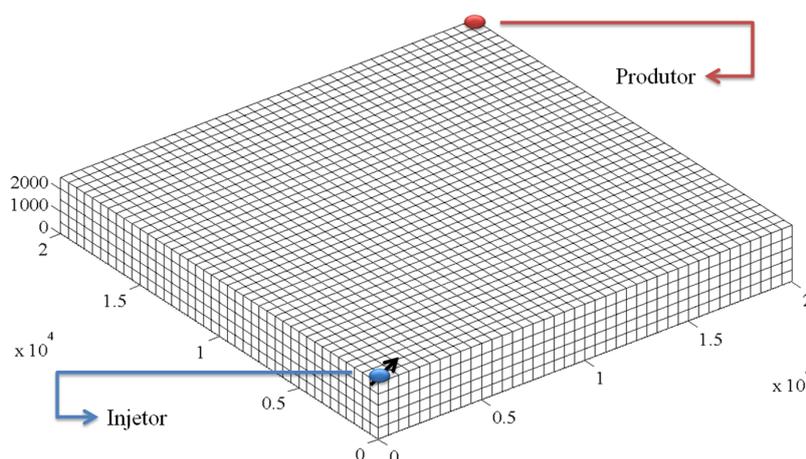


Figura 6.10: Malha usada no problema tridimensional.

A figura 6.11 mostra a solução da equação de saturação, nos dois simuladores, no instante de tempo igual a $28105d$ que equivale à 77 anos.

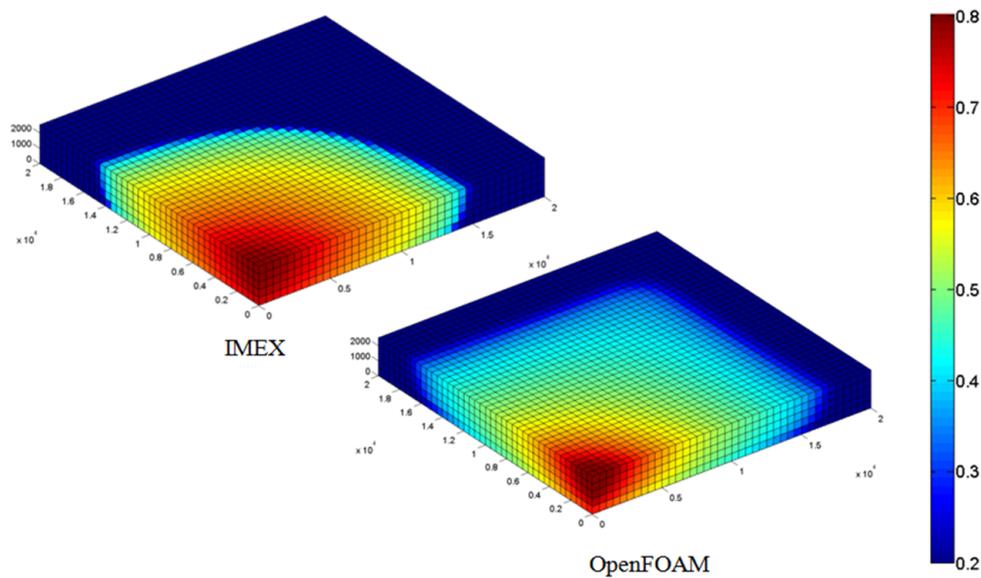


Figura 6.11: Solução da saturação do problema tridimensional.

Uma secção foi feita na diagonal do plano superior para uma melhor visualização dos resultados. A comparação entre as soluções na diagonal é mostrada na figura 6.13

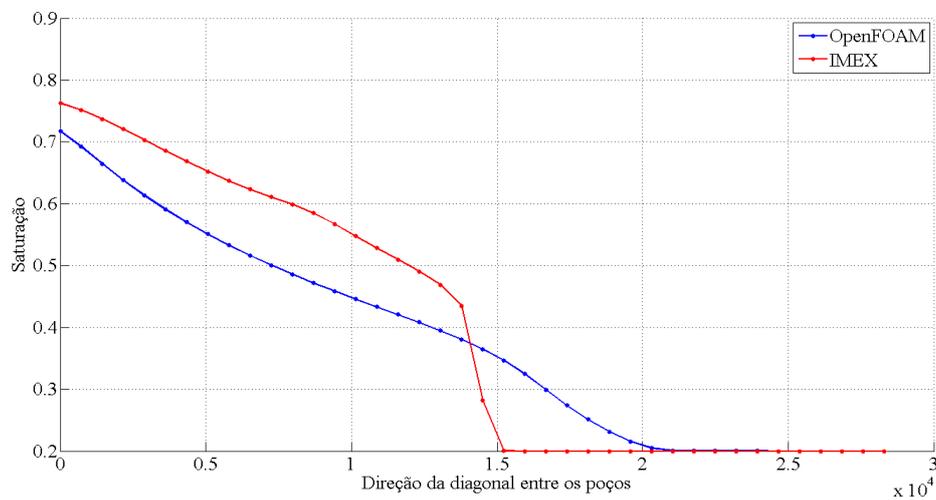


Figura 6.12: Comparação das soluções de saturação 3D na direção diagonal.

A comparação entre as soluções na diagonal mostra claramente uma grande difusão presente na aplicação implementada. Ainda que a solução do esquema *upwind* seja bastante difusiva a quantidade de difusão apresen-

tada mostra que há um problema na solução para o caso 3D. A figura 6.14 mostra a solução da equação de pressão no IMEX para o mesmo problema no mesmo instante de tempo.

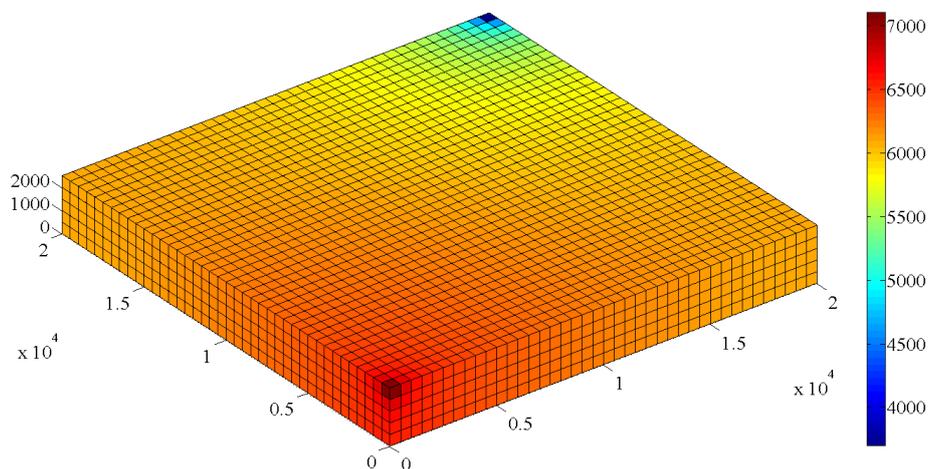


Figura 6.13: Solução da pressão do problema tridimensional no IMEX.

A figura 6.14 mostra a solução encontrada com a aplicação desenvolvida no OpenFOAM.

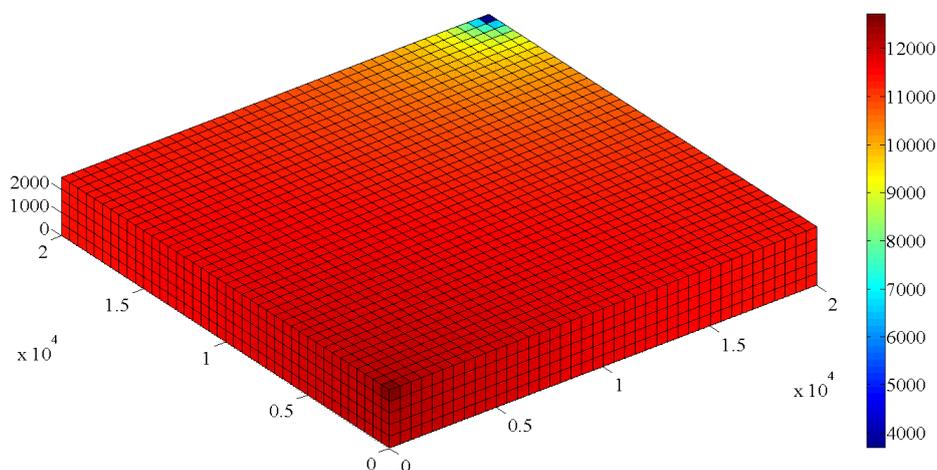


Figura 6.14: Solução da pressão do problema tridimensional no OpenFOAM.

Uma mesma comparação foi feita para as soluções de pressão na direção da diagonal do plano superior. A figura 6.15 mostra a comparação. Assim como na solução de saturação uma grande diferença foi encontrada entre as soluções de pressão. O maior nível de pressão é apresentado na solução da

aplicação desenvolvida. O motivo da do erro na solução da aplicação no caso tridimensional ainda está sendo investigado.

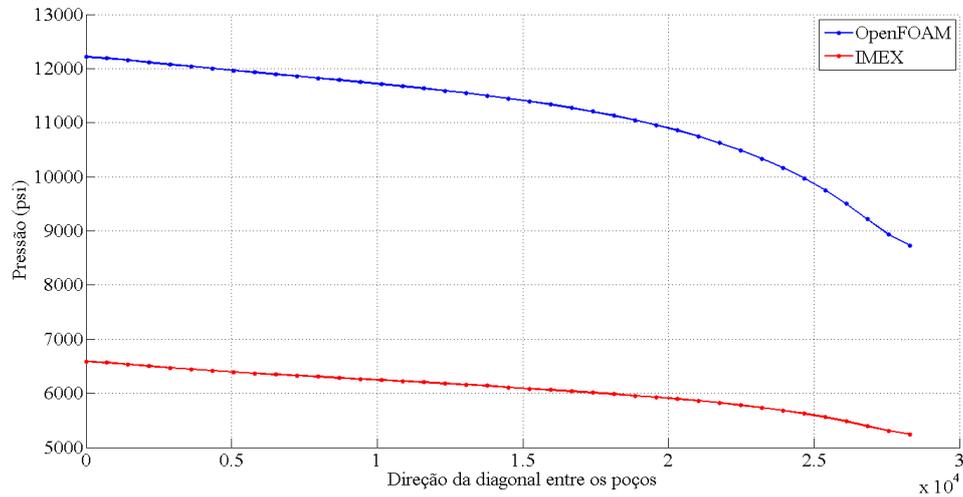


Figura 6.15: Comparação das soluções de pressão 3D na direção diagonal.

6.2.4 Caso heterogêneo bidimensional

Como exemplo de solução de um problema heterogêneo foi montado um mapa de permeabilidades através da intensidade de cor do logotipo da universidade. A figura 6.16 mostra o mapa de permeabilidades montado.

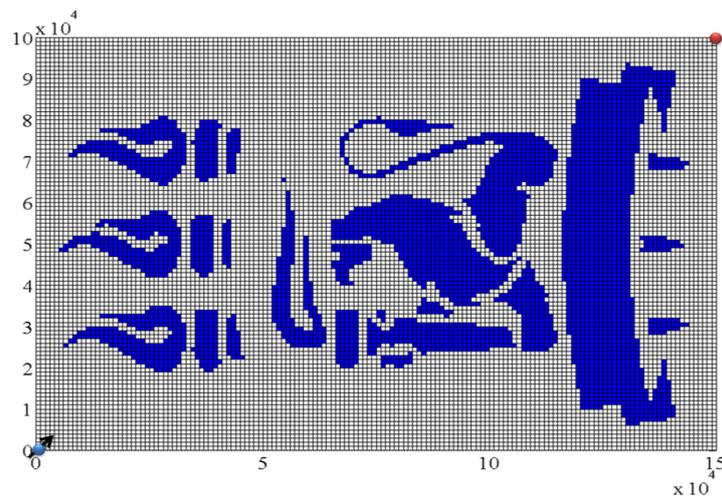


Figura 6.16: Mapa de permeabilidades do caso bidimensional.

A intensidade de cor foi normalizada e transformada em um fator que foi multiplicado por 200 mD , deste modo a menor permeabilidade do mapa é de 2 mD . Como mostrado na figura 6.16 foram colocados um poço injetor no primeiro bloco e um poço produtor no último bloco. O mapa de permeabilidades foi usado em uma geometria plana com 150 blocos na direção x e 100 blocos na direção y . Cada bloco cúbico possui 100 m de aresta e as mesmas propriedades utilizadas nos problemas de comparação foram utilizadas. A figura 6.17 mostra as soluções da equação de saturação com isolinhas de pressão para os instantes de tempo 33, 77, 110 e 176 anos.

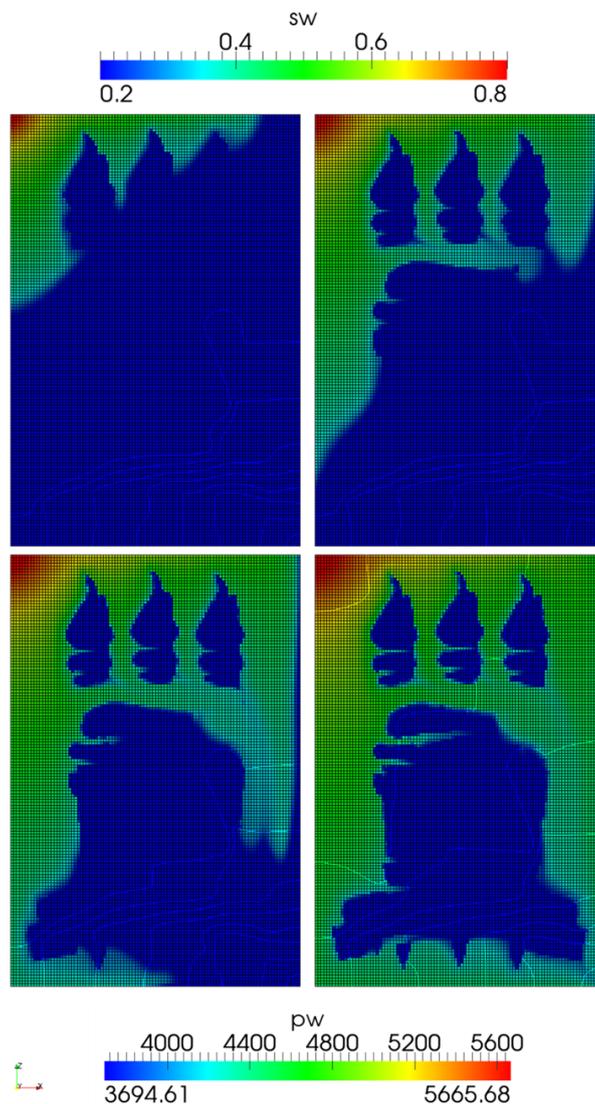


Figura 6.17: Solução do caso heterogêneo bidimensional.

6.2.5 Caso heterogêneo tridimensional

O segundo exemplo de problema heterogêneo apresentado utiliza a mesma configuração do problema tridimensional *five-spot*. Uma barreira de permeabilidades com o valor de 0.002 mD foi criada como mostra a figura 6.18.

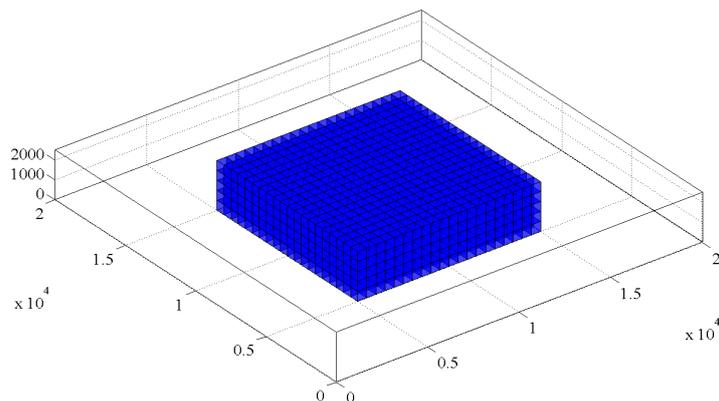


Figura 6.18: Permeabilidades do caso tridimensional.

A figura 6.19 mostra a solução com isosuperfícies de pressão e saturação para os instantes de tempo 22, 55, 66 e 88 anos.

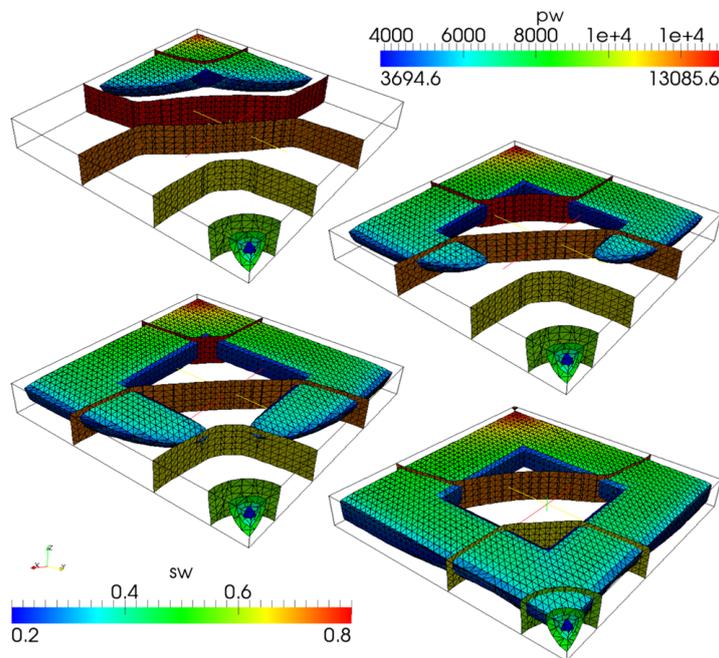


Figura 6.19: Permeabilidades do caso tridimensional.

Capítulo 7

Conclusões e Trabalhos Futuros

7.1 Conclusões

No OpenFOAM é possível se desenvolver aplicações para a solução de problemas complexos em um curto intervalo de tempo. O sistema dispõe de muitas ferramentas que possibilitam o reuso de código. Após a instalação do sistema em um cluster a execução das aplicações em paralelo é aparentemente simples e nenhuma modificação é necessária na aplicação, apenas alguns arquivos de controle adicionais são criados no caso. É possível a criação de classes *template* nos *namespaces fvm* e *fv*, podendo o usuário escrever seu próprio esquema de gradiente ou divergente, isso também é possível para os esquemas de interpolação e qualquer ferramenta adicional que se queira implementar.

A pouca documentação disponível do sistema foi a maior dificuldade encontrada. Para se aprender como são feitas as aplicações é necessário que se estudar com o código fonte das aplicações nativas do sistema. Ler o código fonte das aplicações nativas serve como orientação para adotar as melhores práticas de programação no sistema.

7.2 Trabalhos futuros

As soluções dos problemas expostos para a verificação do simulador implementado mostraram-se razoáveis quando comparadas com o simulador IMEX. O simulador desenvolvido neste trabalho é de escopo acadêmico, que adota uma modelagem bastante simplificada quando comparado com simuladores comerciais como o IMEX. Como propostas de ferramentas que poderão ser incorporadas à aplicação implementada em trabalhos futuros tem-se:

- Cálculo do índice de produtividade para poços inclinados com malhas não estruturadas;
- Implementação de um modelo de escoamento nos poços;
- Investigação de alternativas à solução da equação de saturação na forma hiperbólica;
- Incorporação da ferramenta de geração adaptativa de malhas presente no sistema;
- Implementação do modelo black-oil.
- Implementação do método sequencial implícito e totalmente implícito.
- Implementação do método IMPES adaptativo.
- Verificação de desempenho na solução de problemas em paralelo.

Referências Bibliográficas

- [1] *Cmg web site*, <http://www.cmgl.ca/software/imex.htm>.
- [2] *Openfoam programers guide*, <http://foam.sourceforge.net/docs/Guides-a4/ProgrammersGuide.pdf>.
- [3] *Openfoam solvers*, http://www.tfd.chalmers.se/~hani/kurser/OS_CFD_2008/TimBehrens/tibeh-report-fin.pdf.
- [4] *Openfoam user's guide*, <http://foam.sourceforge.net/docs/Guides-a4/UserGuide.pdf>.
- [5] *Openfoam web site*, <http://www.openfoam.com/>.
- [6] *Speedit web site*, <http://speedit.vratis.com/>.
- [7] I. Aavatsmark and RA Klausen, *Well index in reservoir simulation for slanted and slightly curved wells in 3d grids*, SPE Journal **8** (2003), no. 1, 41–48.
- [8] JJ Arps, *Analysis of decline curves*, Transactions of the American Institute of Mining, Metallurgical and Petroleum Engineers **160** (1945), 228.
- [9] K. Aziz and A. Settari, *Petroleum reservoir simulation*, vol. 476, Applied Science Publishers London, 1979.
- [10] L.R. Brown, *Microbial enhanced oil recovery (meor)*, Current Opinion in Microbiology **13** (2010), no. 3, 316–320.
- [11] H. L. Chang, *Polymer flooding technology - yesterday, today, and tomorrow*, Journal of Petroleum Technology (1978), 1113–1128.
- [12] Z. Chen, G. Huan, and Y. Ma, *Computational methods for multiphase flows in porous media*, vol. 2, Society for Industrial Mathematics, 2006.

- [13] Z. Chen, G. Huan, and H. Wang, *Simulation of a compositional model for multiphase flow in porous media*, Numerical Methods for Partial Differential Equations **21** (2005), no. 4, 726–741.
- [14] Z. Chen, G. Qin, and R.E. Ewing, *Analysis of a compositional model for fluid flow in porous media*, SIAM Journal on Applied Mathematics (2000), 747–777.
- [15] R.S. da Silva, *Simulação de escoamento bifásico óleo-água em reservatórios de petróleo usando computadores paralelos de memória distribuída*, 2008.
- [16] L.P. Dake, *Fundamentals of reservoir engineering*, vol. 8, Elsevier Science, 1978.
- [17] _____, *The practice of reservoir engineering*, vol. 36, Elsevier Science, 2001.
- [18] D.K.E. de Carvalho, *Uma formulação do método dos volumes finitos com estrutura de dados por aresta para a simulação de escoamentos em meios porosos*, 2005.
- [19] Y. Ding, *A generalized 3d well model for reservoir simulation*, SPE Journal **1** (1996), no. 4, 437–450.
- [20] T. Ertekin, J.H. Abou-Kassem, and G.R. King, *Basic applied reservoir simulation*, Society of Petroleum Engineers, 2001.
- [21] J.R. Fanchi, *Principles of applied reservoir simulation*, Gulf Professional Publishing, 2005.
- [22] MJ Fetkovich, *Decline curve analysis using type curves*, Journal of Petroleum Technology **32** (1980), no. 6, 1065–1077.
- [23] B. Hascakir and A. Kavscek, *Reservoir simulation of cyclic steam injection including the effects of temperature induced wettability alteration*, SPE Western Regional Meeting, 2010.
- [24] D. Havlena and AS Odeh, *The material balance as an equation of a straight line*, Journal of Petroleum Technology **15** (1963), no. 8, 896–900.
- [25] H. Jasak, *Error analysis and estimation for the finite volume method with applications to fluid flows*, Direct **1000** (1996), no. June.

- [26] H. Jasak, A. Jemcov, and Z. Tukovic, *Openfoam: A c++ library for complex physics simulations*, International Workshop on Coupled Methods in Numerical Dynamics, IUC, Dubrovnik, Croatia, 2007.
- [27] R.J. LeVeque, *Finite volume methods for hyperbolic problems*, vol. 31, Cambridge Univ Pr, 2002.
- [28] Y.K. Li, L. Nghiem, and A. Siu, *Phase behaviour computations for reservoir fluids: Effect of pseudo-components on phase diagrams and simulation results*, Journal of Canadian Petroleum Technology **24** (1985), no. 6.
- [29] M.V.B. Machado, *Modelagem e simulação numérica de reservatórios de petróleo: principais aspectos envolvidos.*, Petrobras: Salvador, 2010. Livro em preparação.
- [30] Z. Malecha, Ł. Mirosław, T. Tomczak, Z. Koza, M. Matyka, W. Tarnawski, and D. Szczerba, *Gpu-based simulation of 3d blood flow in abdominal aorta using openfoam*, Archives of Mechanics **63** (2011), no. 2, 137–161.
- [31] R. Mannon, *Oil production forecasting by decline curve analysis*, Fall Meeting of the Society of Petroleum Engineers of AIME, 1965.
- [32] P. Morelato, L. Rodrigues, and O. Romero, *Effect of polymer injection on the mobility ratio and oil recovery*, SPE Heavy Oil Conference and Exhibition, 2011.
- [33] N. Morrow, *Capillary pressure correlations for uniformly wetted porous media*, Journal of Canadian Petroleum Technology **15** (1976), no. 4.
- [34] D.W. Peaceman, *Fundamentals of numerical reservoir simulation*, vol. 6, Elsevier, 1977.
- [35] ———, *Interpretation of well-block pressures in numerical reservoir simulation (includes associated paper 6988)*, Old SPE Journal **18** (1978), no. 3, 183–194.
- [36] A.J. Rosa, R. de Souza Carvalho, and J.A.D. Xavier, *Engenharia de reservatórios de petróleo*, Interciência, 2006.
- [37] VC Santanna, FDS Curbelo, TN Castro Dantas, AA Dantas Neto, HS Albuquerque, and AIC Garnica, *Microemulsion flooding for enhanced oil recovery*, Journal of Petroleum Science and Engineering **66** (2009), no. 3-4, 117–120.

- [38] E.F. Toro, *Riemann solvers and numerical methods for fluid dynamics. 1999*, Springer, Berlin.
- [39] C. Wolfsteiner, L.J. Durlofsky, and K. Aziz, *Calculation of well index for nonconventional wells on arbitrary grids*, Computational Geosciences **7** (2003), no. 1, 61–82.