



**UNIVERSIDADE FEDERAL DO CEARÁ**  
**CENTRO DE TECNOLOGIA**  
**DEPARTAMENTO DE ENGENHARIA QUÍMICA**  
**PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA QUÍMICA**

**IVENS DA COSTA MENEZES LIMA**

**SIMULAÇÃO DE RESERVATÓRIOS DE PETRÓLEO EM PARALELO**  
**UTILIZANDO MALHAS NÃO-ESTRUTURADAS 2D E 3D**

**FORTALEZA**

**2017**

IVENS DA COSTA MENEZES LIMA

SIMULAÇÃO DE RESERVATÓRIOS DE PETRÓLEO EM PARALELO UTILIZANDO  
MALHAS NÃO ESTRUTURADAS 2D E 3D

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Química da Universidade Federal do Ceará como requisito parcial à obtenção do título de Mestre em Engenharia Química. Área de concentração: Simulação de reservatórios de petróleo.

Orientador: Prof. Dr. Francisco Marcondes

FORTALEZA

2017

Dados Internacionais de Catalogação na Publicação  
Universidade Federal do Ceará  
Biblioteca Universitária

Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

---

- L698s Lima, Ivens da Costa Menezes.  
Simulação de reservatórios de petróleo em paralelo utilizando malhas não estruturadas 2D e 3D / Ivens da Costa Menezes Lima. – 2017.  
96 f. : il. color.
- Dissertação (mestrado) – Universidade Federal do Ceará, Centro de Tecnologia, Programa de Pós-Graduação em Engenharia Química, Fortaleza, 2017.  
Orientação: Prof. Dr. Francisco Marcondes.
1. Paralelização. 2. EbFVM. 3. Malhas não estruturadas. 4. Simulação. 5. UTCOMP. I. Título.  
CDD 660
-

IVENS DA COSTA MENEZES LIMA

SIMULAÇÃO DE RESERVATÓRIOS DE PETRÓLEO EM PARALELO UTILIZANDO  
MALHAS NÃO ESTRUTURADAS 2D E 3D

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Química da Universidade Federal do Ceará como requisito parcial à obtenção do título de Mestre em Engenharia Química. Área de concentração: Simulação de reservatórios de petróleo.

Aprovada em: 10 / 08 / 2017.

BANCA EXAMINADORA

---

Prof. Dr. Francisco Marcondes  
(Orientador)

Universidade Federal do Ceará (UFC)

---

Prof. Dr. Luis Glauber Rodrigues  
Universidade Federal do Ceará (UFC)

---

Prof. Dr. Joaquim Bento Cavalcante Neto  
Universidade Federal do Ceará (UFC)

---

Dr. Luiz Otávio Schmall dos Santos  
Petrobras

Dedico este trabalho a minha esposa Aline Barroso, aos meus pais Ivan Lima e Mônica Lima e a meu irmão Arthur Lima. Vocês sempre foram meus maiores incentivadores e sempre torceram pelo meu sucesso. Agradeço a Deus pela vida de vocês na minha vida.

## **AGRADECIMENTOS**

Ao Prof. Dr. Francisco Marcondes pela paciência e pela orientação durante o desenvolvimento deste trabalho.

Aos Profs. Drs. Joaquim Bento e Markos Feitas pelo apoio intelectual no esclarecimento de dúvidas e suporte computacional.

Ao Daniel Matos pelo suporte técnico com a instalação dos programas e a configuração e manutenção do ambiente de desenvolvimento.

Aos colegas Edílson Drumond, Paulo Vicente e José Renê pelas ajudas e pelo companheirismo durante a caminhada.

Aos bolsistas José Cláudio, João Pedro, João Henrique, Antônio Netto e Vanilson Nogueira pela ajuda com o desenvolvimento e a implementação do código.

A Petrobras e a Fundação Astef pelo apoio financeiro.

*“Confia no Senhor de todo o teu coração, e não te ensoberbeças no teu próprio entendimento; reconhece-O em todos os teus caminhos e Ele endireitará as tuas veredas. Não sejas sábio aos teus próprios olhos, teme ao Senhor e aparta-te do mal. Isto será saúde para o teu corpo e refrigério para os teus ossos. ”*

*Provérbios 3:5-8.*

## RESUMO

Simulações em grande escala envolvendo centenas de milhares de blocos de malha exigem a aplicação de técnicas de paralelização para atingir tempos computacionais práticos. Existem basicamente três tipos de paralelização: memória distribuída, memória compartilhada e uma combinação dos dois anteriores. Este trabalho baseia-se na primeira abordagem, onde o domínio é dividido entre processos e cada um é responsável apenas pela sua porção do reservatório. Esta abordagem tem duas vantagens principais: reduz a memória necessária por processo e permite que as simulações sejam realizadas usando *clusters* com grande número de processadores. Neste trabalho, bibliotecas de código aberto foram usadas para a partição do domínio computacional, gerenciamento das informações de malha entre os processos e resolução do sistema linear de equações gerado a partir da discretização das equações diferenciais parciais que modelam o fluxo no reservatório. O ParMetis (*Parallel Graph Partitioning and Fill-reducing Matrix Ordering*) é usado para particionar os domínios computacionais, o FMDB (*Flexible Distributed Mesh Database*) é responsável pelo gerenciamento das informações de malha entre os processos, e o PETSc (*Portable, Extensible Toolkit for Scientific Computation*) resolve o sistema linear de equações. A abordagem numérica baseia-se no Método de Volumes Finitos baseado em Elementos (EbFVM na sigla em inglês) em conjunto com malhas não estruturadas. O principal desafio deste trabalho foi gerenciar o conjunto de dados de malha, de fluidos e de propriedades de reservatório de modo que as comunicações entre os processos fossem reduzidas. Foi usado um simulador *in-house* composicional, multicomponente/multifásico chamado UTCOMP, o qual foi desenvolvido na Universidade do Texas em Austin, a fim de realizar essa implementação. É mostrado que o EbFVM é adequado para a modelagem de reservatórios com geometrias complexas, e é apresentado seu desempenho em modo paralelo tanto para malhas 2D quanto para malhas 3D. Os resultados são avaliados em termos de curvas de produção de óleo e de gás, curvas de *speedup* e tempos de CPU para vários estudos de caso.

**Palavras-chave:** Paralelização, EbFVM, malhas não estruturadas, simulador UTCOMP.



## ABSTRACT

Large-scale simulations involving hundreds of thousands of grid blocks require application of parallelization techniques in order to achieve practical computational times. There are essentially three types of parallelization: distributed memory, shared memory, and a combination of the two mentioned. This work is based on the first approach, where the domain is divided among processes and each one is responsible only for its portion of the reservoir. This approach has two main advantages: it reduces the required memory per process and allows the simulations to be carried out using clusters with large number of processors. In this work, open source libraries were used to partition the computational domain, manage the grid information between the processes, and solve the linear system of equations generated from the discretization of partial differential equation modeling fluid flow in the reservoir. ParMetis (Parallel Graph Partitioning and Fill-reducing Matrix Ordering) is used to partition the computational domains, FMDB (Flexible Distributed Mesh Database) is responsible to manage the grid information between the processes, and PETSc (Portable, Extensible Toolkit for Scientific Computation) solves the linear system of equations. The numerical approach is based on the Element based Finite Volume Method (EbFVM) in conjunction with unstructured meshes. The main challenge was to manage the grid, fluids, and reservoir data set in such a way that the communications between the processes were reduced. It was used an in-house compositional, multicomponent/multiphase simulator called UTCOMP, which was developed at The University of Texas at Austin, in order to perform this implementation. It is shown that the EbFVM is suited for modeling reservoirs with complex geometries, and its performance in parallel mode is presented. The results are evaluated in terms of oil and gas production curves, speedup curves and CPU times for various case studies.

**Keywords:** Parallelization, EbFVM, 3D unstructured grids, UTCOMP simulator.

## LISTA DE FIGURAS

Figura 1 – Poço em Oklahoma, Estado Unidos, em 1922 .....	15
Figura 2 – Malha não estruturada tipo PEBI.....	18
Figura 3 – Malha não estruturada com triângulos e quadriláteros .....	19
Figura 4 – Malha não estruturada com ênfase num volume de controle .....	37
Figura 5 – Elementos transformados: (a) quadrilátero e (b) triângulo.....	38
Figura 6 – Elementos 3D: (a) hexaedro, (b) tetraedro, (c) prisma e (d) pirâmide .....	39
Figura 7 – Exemplos de divisão de domínio.....	46
Figura 8 – Esquema de comunicação entre UTCOMP e FMDB .....	47
Figura 9 – Esquema de chamada das bibliotecas no simulador .....	48
Figura 10 – Malha dividida em 4 processos com ghost-layer .....	50
Figura 11 – Malha dividida em dois processos .....	50
Figura 12 – Malha com ghost layer em destaque .....	51
Figura 13 – Organização do vetor de indexação .....	53
Figura 14 – Topologia para uma malha com 8 processos .....	55
Figura 15 – Malha original do caso 1 .....	59
Figura 16 – Malha do caso 1 dividida em: (a) 2, (b) 4 e (c) 8 processos .....	60
Figura 17 – Curvas de produção de (a) óleo e (b) gás do caso 1 .....	61
Figura 18 – Curva de speedup do caso 1 .....	62
Figura 19 – Malha original do caso 2 .....	63
Figura 20 – Malha do caso 2 dividida em: (a) 2, (b) 4 e (c) 8 processos .....	64
Figura 21 – Curvas de produção de (a) óleo e de (b) gás do caso 2 .....	65
Figura 22 – Curvas de RGO e P méd. do caso 2 .....	66
Figura 23 – Curva de speedup do caso 2 .....	66
Figura 24 – Custos de comunicação do caso 2 com 16 processos .....	67
Figura 25 – Malha original do caso 3 .....	69
Figura 26 – Campos de (a) permeabilidade XY e (b) porosidade do caso 3.....	69
Figura 27 – Curvas de produção de (a) óleo e de (b) gás do caso 3 .....	70
Figura 28 – Curvas de RGO e P méd. do caso 3 .....	71
Figura 29 – Curva de speedup do caso 3 .....	71
Figura 30 – Malhas do caso 4: (a) 10201 vértices, (b) 20301 vértices e (c) 40401 vértices ....	72
Figura 31 – Curvas de produção de óleo e de gás do caso 4, malha (c).....	74
Figura 32 – Curvas de speedup do caso 4 .....	75

Figura 33 – Malha original do caso 5 .....	77
Figura 34 – Malha do caso 5 dividida em: (a) 2, (b) 4 e (c) 8 processos .....	77
Figura 35 – Curvas de produção de (a) óleo e de (b) gás do caso 5 .....	78
Figura 36 – Curva de speedup do caso 5 .....	79
Figura 37 – Malha original do caso 6 .....	80
Figura 38 – Malha do caso 6 dividida em: (a) 2, (b) 4 e (c) 8 processos .....	80
Figura 39 – Curvas de produção de (a) óleo e (b) gás do caso 6.....	81
Figura 40 – Curva de speedup do caso 6 .....	82
Figura 41 – Custos de comunicação do caso 6 com 16 processos .....	82
Figura 42 – Malha original do caso 7 .....	83
Figura 43 – Malha do caso 7 dividida em: (a) 2, (b) 4 e (c) 8 processos .....	83
Figura 44 – Curvas de produção de (a) óleo e de (b) gás do caso 7 .....	84
Figura 45 – Curva de speedup do caso 7 .....	85
Figura 46 – Custos de comunicação do caso 7 com 16 processos .....	86

## LISTA DE TABELAS

Tabela 1 – Propriedades de reservatório do caso 1 .....	58
Tabela 2 – Concentração de componentes do caso 1 .....	59
Tabela 3 – Restrição de poços do caso 1.....	59
Tabela 4 – Propriedades de reservatório do caso 2 .....	62
Tabela 5 – Concentração dos componentes do caso 2 .....	63
Tabela 6 – Restrição de poços do caso 2.....	63
Tabela 7 – Propriedades de reservatório do caso 3 .....	68
Tabela 8 – Concentração de componentes do caso 3 .....	68
Tabela 9 – Restrição de poços do caso 3.....	68
Tabela 10 – Comparativo dos valores de speedup do caso 4 .....	75

## LISTA DE ABREVIATURAS E SIGLAS

CP	Corner Point
EbFVM	Element based Finite Volume Method
EOS	Equation of State (Equação de Estado)
FI	Fully Implicit
FMDB	Flexible Distributed Mesh Database
IMPEC	Implicit Pressure Explicit Composition
IMPSAT	Implicit Pressure and Saturation
ip	integration point (ponto de integração)
nc	Número de componentes
np	Número de fases
ParMetis	Parallel Graph Partitioning and Fill-reducing Matrix Ordering
PETSc	Portable, Extensible Toolkit for Scientific computation
SVC	Sub-volume de controle
WAG	Water Alternating Gas

## LISTA DE SÍMBOLOS

$\xi$	Densidade molar
$x$	Fração molar do componente
$L$	Fração molar da fase
$f$	Fugacidade
$\lambda_r$	Mobilidade relativa
$N$	Número de moles
$k_r$	Permeabilidade relativa
$\gamma$	Peso específico
$\phi$	Porosidade
$P$	Pressão
$P_c$	Pressão capilar
$S$	Saturação
$\vec{K}$	Tensor dispersão
$\vec{k}$	Tensor permeabilidade absoluta de rocha
$q$	Vazão de injeção ou produção
$\vec{u}$	Velocidade da fase
$\mu$	Viscosidade
$V_b$	Volume do volume de controle

## SUBSCRITOS

$i$	Componente
$j$	Fase
$r$	De referência

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO .....</b>	<b>15</b>
<b>1.1</b>	<b>Discretização de domínio .....</b>	<b>17</b>
<i>1.1.1</i>	<i>Malhas não-estruturadas .....</i>	<i>17</i>
<b>1.2</b>	<b>Paralelização .....</b>	<b>19</b>
<b>1.3</b>	<b>UTCOMP .....</b>	<b>21</b>
<b>1.4</b>	<b>Objetivos.....</b>	<b>21</b>
<i>1.4.1</i>	<i>Geral .....</i>	<i>21</i>
<i>1.4.2</i>	<i>Específicos .....</i>	<i>22</i>
<b>1.5</b>	<b>Escopo do trabalho.....</b>	<b>22</b>
<b>2</b>	<b>EQUAÇÕES GOVERNANTES.....</b>	<b>23</b>
<b>2.1</b>	<b>Balanco de massa.....</b>	<b>23</b>
<b>2.2</b>	<b>Equação da pressão .....</b>	<b>25</b>
<b>2.3</b>	<b>Equações de restrição.....</b>	<b>27</b>
<b>2.4</b>	<b>Equações de propriedades .....</b>	<b>28</b>
<i>2.4.1</i>	<i>Viscosidade .....</i>	<i>28</i>
<i>2.4.2</i>	<i>Densidades molar e mássica .....</i>	<i>30</i>
<i>2.4.3</i>	<i>Saturação .....</i>	<i>30</i>
<i>2.4.4</i>	<i>Permeabilidade relativa .....</i>	<i>31</i>
<b>2.5</b>	<b>Equação de estado .....</b>	<b>32</b>
<b>3</b>	<b>FORMULAÇÕES MATEMÁTICAS .....</b>	<b>35</b>
<b>3.1</b>	<b>Equações aproximadas.....</b>	<b>35</b>
<b>3.2</b>	<b>Método dos Volumes Finitos baseado em Elementos.....</b>	<b>36</b>
<b>4</b>	<b>PARALELIZAÇÃO .....</b>	<b>44</b>
<b>4.1</b>	<b>Balaceamento de carga e divisão de domínio .....</b>	<b>45</b>
<i>4.1.1</i>	<i>Particionamento k-way.....</i>	<i>46</i>
<b>4.2</b>	<b>Gerenciamento de informações .....</b>	<b>46</b>

4.3	PETSc .....	48
4.4	Interface de Passagem de Mensagens .....	49
4.5	Ghost layers.....	49
4.6	Atributos do código .....	52
4.6.1	<i>Reordenação de vértices</i> .....	52
4.6.2	<i>Indexação de elementos</i> .....	53
4.6.3	<i>Topologia virtual de processos</i> .....	54
4.6.4	<i>Troca de informações</i> .....	55
4.7	Speedup .....	56
5	<b>RESULTADOS</b> .....	58
5.1	<b>Casos 2D</b> .....	58
5.1.1	<i>Caso 1</i> .....	58
5.1.2	<i>Caso 2</i> .....	62
5.1.3	<i>Caso 3</i> .....	67
5.1.4	<i>Caso 4</i> .....	72
5.2	<b>Casos 3D</b> .....	76
5.2.1	<i>Caso 5</i> .....	76
5.2.2	<i>Caso 6</i> .....	79
5.2.3	<i>Caso 7</i> .....	83
6	<b>CONCLUSÃO</b> .....	87
7	<b>SUGESTÕES DE TRABALHOS FUTUROS</b> .....	88
	<b>REFERÊNCIAS</b> .....	89
	<b>APÊNDICE A – CAMPOS DE SATURAÇÃO DE ÓLEO E DE GÁS</b> .....	93

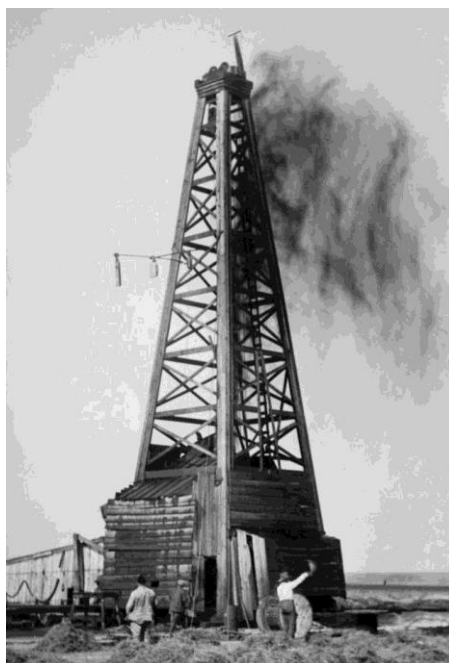


## 1 INTRODUÇÃO

O petróleo é a fonte de energia mais consumida no planeta. Sendo composto por vários tipos de hidrocarbonetos, de C1 a C20+, suas frações são utilizadas nas mais diversas áreas da indústria, desde a produção de combustíveis e lubrificantes até a fabricação de tintas, eletrônicos e remédios. Devido a sua importância econômica e energética, sua exploração, produção e processamento são buscados por todos os países.

Apesar dessa importância energética e industrial, sua produção não é uma tarefa fácil. Como a produção industrial de petróleo começou na segunda metade do século XIX, hoje é praticamente impossível encontrar reservatórios rasos, de fácil acesso. Enquanto em 1859, Edwin Drake encontrou petróleo em um poço de 21 metros de profundidade, atualmente no Brasil, o petróleo da camada pré-sal, por exemplo, pode chegar a 8.000 m de profundidade em alguns pontos, sendo que a lâmina d'água varia de 1.000 m a 2.000 m de espessura e a camada de sal possui aproximadamente 2.000 m de espessura (LIMA, 2008). Além da dificuldade de se chegar ao petróleo, há a dificuldade relacionada às técnicas de produção a partir da rocha reservatório.

Figura 1 – Poço em Oklahoma, Estado Unidos, em 1922



Fonte: [https://commons.wikimedia.org/wiki/File:Gusher\\_Okemah\\_OK\\_1922.jpg](https://commons.wikimedia.org/wiki/File:Gusher_Okemah_OK_1922.jpg)

Todos esses fatores são um enorme desafio para as empresas produtoras de petróleo. E é nesse ponto que a simulação se torna importante. Fazendo uso de *clusters*

de computadores e de técnicas modernas de programação, ela possibilita o entendimento do comportamento dos fluidos presentes no reservatório, permitindo que engenheiros elaborem a melhor estratégia de produção. Através da simulação, é possível estudar o posicionamento dos poços produtores e injetores, seu tempo de operação, o tipo de fluido injetado, além de realizar estudos de análise de histórico (*history matching*) e de comportamento de fases (*phase behavior*) (SILVA, 2008). Esses estudos são baseados em dados geológicos de propriedades de rocha, como porosidade e permeabilidade, em dados físicos, como pressão e temperatura do reservatório, e em informações de componentes presentes no óleo. Todas essas informações são obtidas de amostras recolhidas em campo.

A simulação busca, através de aproximações, analisar o comportamento do reservatório sob determinadas circunstâncias e prever o futuro da produção. Como consequência, há a diminuição das incertezas e riscos, e a maximização da produção. É tarefa do engenheiro de petróleo realizar essas análises e decidir a melhor forma de operação. Para tanto, é necessário que ele possua as ferramentas adequadas, ou seja, um simulador que gere resultados realistas no menor tempo possível.

Desde a década de 1960, com o advento dos computadores, a simulação de reservatórios é utilizada como ferramenta de otimização da produção de petróleo (COATS, 1982 *apud* FERNANDES, 2014). Nesse período, os únicos modelos resolvidos eram do tipo *Black-oil*. Ele considera apenas três pseudo-componentes presentes no reservatório: água, óleo e gás. Apesar da simplicidade do modelo e do baixo custo computacional, ele só é aplicável em reservatórios de óleo pesado (FERNANDES, 2014). À medida que mais técnicas de produção foram desenvolvidas, novos modelos também surgiram, de maneira que possibilitassem a simulação dessas técnicas. O modelo composicional é mais robusto, pois pode lidar com vários pseudo-componentes, além de realizar cálculos *flash* e modelar a transferência de massa entre fases. Simuladores que possuem esse modelo integrado permitem a simulação de técnicas de recuperação secundária como injeção miscível de gás e injeção tipo WAG (*Water Alternating Gas*). Existem também simuladores com modelos térmicos que possibilitam a simulação de outras técnicas de produção como combustão *in-situ* e injeção de vapor.

## 1.1 Discretização de domínio

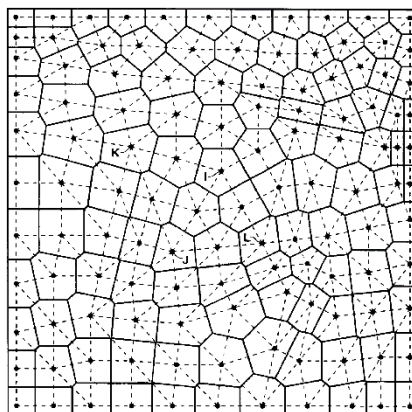
As equações que descrevem o escoamento multifásico em meios porosos são equações diferenciais parciais não lineares e, portanto, não possuem solução analítica, a não ser em casos bastante simplificados, com hipóteses que não condizem com a realidade. Para resolver problemas de reservatório de petróleo, são necessários métodos numéricos baseados na discretização do domínio em malhas geométricas. Existem vários tipos de malhas nas quais se pode dividir o domínio do reservatório, e é necessário que as equações sejam tratadas de acordo com a malha escolhida.

Dependendo da malha escolhida, a discretização das equações e a implementação do método podem ser mais fáceis, como é o caso da malha cartesiana. Por outro lado, ela não é indicada para modelar geometrias complexas, como as que possuem falhas e/ou fraturas. O uso de malhas não estruturadas é mais indicado quando se trata de um reservatório com irregularidades geométricas, contudo esse tipo de malha exige um esforço matemático maior para a discretização das equações.

### 1.1.1 Malhas não-estruturadas

O uso de malhas não-estruturadas na simulação de reservatórios de petróleo data da década de 1980 com o trabalho de Heinemann e Brand (1988), que usaram malhas do tipo PEBI (*PErpendicular BIsector*), também conhecidas como malhas de Voronoi (FERNANDES, 2014). Marcondes (1996) usou esse tipo de malha para resolver problemas de reservatório de petróleo usando métodos adaptativos. Esse tipo de malha também está presente em simuladores comerciais, como o desenvolvido pela ExxonMobil (BECKNER et al., 2015a; USADI et al., 2007). Malhas do tipo PEBI são construídas de tal forma que as fronteiras dos volumes de controle sejam sempre perpendiculares às arestas dos elementos da malha, como mostrado na Figura 2. Isso dá a elas a vantagem de utilizar apenas dois pontos no cálculo do fluxo de massa que entra ou sai de uma determinada aresta do volume de controle.

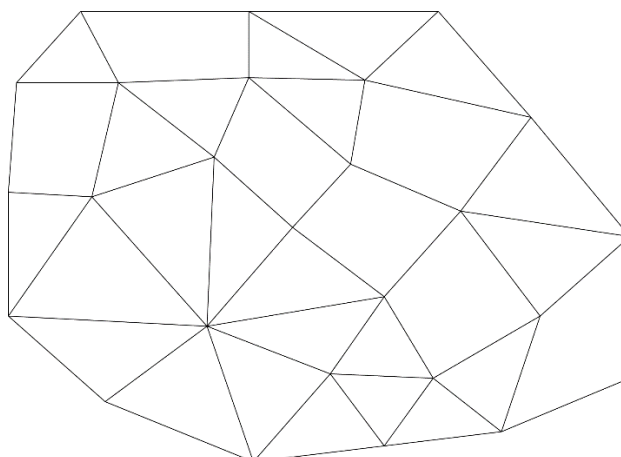
Figura 2 – Malha não estruturada tipo PEBI



Fonte: <http://www.google.com/patents/US6826520#backward-citations>

Baliga e Patankar (1980) criaram um novo método que alia a flexibilidade oferecida pelas malhas não estruturadas com a abordagem conservativa do Método dos Volumes Finitos (FVM, na sigla em inglês) chamado CVFEM (*Control Volume Finite Elements Method*). A utilização do CVFEM na simulação de reservatórios de petróleo data dos anos de 1980, quando Rozon (1989) usou esse método para resolver um fluxo monofásico usando elementos quadriláteros. Em 2004, Maliska sugeriu a substituição do nome CVFEM, argumentando que ele passa a ideia errada de que é um método de elementos finitos com balanço de material, quando na verdade é um método de volumes finitos que toma emprestado as funções de forma do método de elementos finitos. Na prática, é um método híbrido que busca agregar o melhor dos dois métodos. Por isso, Maliska (2004) propôs chamá-lo de Método de Volumes Finitos Baseado em Elementos (EbFVM, na sigla em inglês). Como este trabalho está baseado nesse método, e a visão do Maliska é compartilhada pelo autor, doravante ele será referido como EbFVM. A Figura 3 mostra um exemplo de malha não estruturada mista, com triângulos e quadriláteros. Ao contrário das malhas tipo PEBI, a malha não estruturada geral não cumpre o requisito de perpendicularidade entre a fronteira do volume de controle e a aresta do elemento. Entretanto ela apresenta a flexibilidade de poder utilizar triângulos e quadriláteros em 2D, e tetraedros, hexaedros, prismas e pirâmides em 3D. A seção 3.2 mostra os procedimentos matemáticos referentes ao método EbFVM.

Figura 3 – Malha não estruturada com triângulos e quadriláteros



Fonte: Elaborada pelo autor

Desde o surgimento dos métodos que utilizam malhas não estruturadas, vários autores publicaram diversos trabalhos na área. Prévost expandiu o método de Pollock de *streamline tracing* para malhas CP (*Corner Point*) e não estruturadas (PRÉVOST; EDWARDS; BLUNT, 2001). Cordazzo *et al.* (2005) mostraram a flexibilidade geométrica e a redução dos efeitos de orientação de malha proporcionados pelo método EbFVM em simulação 2D usando elementos triangulares e quadrangulares. Marcondes e Sepehnoori (2010) investigaram a viabilidade do método para resolver problemas heterogêneos e anisotrópicos e mostraram a necessidade de se utilizar uma quantidade muito menor de vértices para a obtenção de determinado resultado quando comparado com uma malha cartesiana. Fernandes *et al.* (2013) investigaram a utilização de várias funções de interpolação para malhas 2D. Continuando seu trabalho, Fernandes *et al.* (2015) implementaram a função de interpolação TVD (*Total Variation Diminishing*) para malhas 3D. Marcondes *et al.* (2013) expandiram o método EbFVM para malhas tridimensionais em um simulador com metodologia *fully implicit*. Araújo *et al.* (2016) aplicaram o método em malhas 3D em um simulador com metodologia IMPEC (*IMplicit Pressure Explicit Composition*).

## 1.2 Paralelização

A supercomputação através da vetorização surgiu por volta da década de 1970 (KILLOUGH, 1993), e a paralelização começou a ser utilizada na simulação de reservatórios de petróleo somente no final da década de 1980. O primeiro trabalho foi o de Scott *et al.* (1987) utilizando computadores MIMD (*Multiple Instruction, Multiple Data*) para otimizar as tarefas de geração da matriz de coeficientes e resolução do sistema

linear em modelos *black-oil* e composicional. Wheeler e Smith (1990) investigaram a paralelização em simulações 3D implícitas com escoamento bifásico água/óleo. Killough e Bhogeswara (1990, 1991) investigaram a simulação de modelos composicionais em paralelo para malhas cartesianas em computadores de estrutura hipercubo. Desde então, diversos trabalhos foram publicados nas diversas áreas da computação em paralelo.

Killough (1993) discutiu aspectos como o balanceamento de carga e a solução de sistemas lineares e suas influências na simulação em paralelo. Ghorri *et al.* (1995) investigaram uma versão cartesiana do simulador UTCOMP tanto em ambiente de memória compartilhada quanto de memória distribuída. Magras *et al.* (2001) investigaram a paralelização de *solvers* e pré-condicionadores através do esquema de memória compartilhada para diversos refinamentos de malha. Usadi *et al.* (2007) investigaram o comportamento de diversos pré-condicionadores usando o esquema de memória distribuída para simulação paralela utilizando malhas não estruturadas do tipo PEBI e modelos *black-oil* e composicional. Han *et al.* (2007) investigaram o comportamento de um simulador *fully implicit* químico em conjunto com malhas cartesianas e CP. Silva (2008) reescreveu o código de um simulador com formulação IMPES (*Implicit Pressure Explicit Saturation*) em conjunto com malhas não estruturadas (CVFEM baseado em arestas) da linguagem MATLAB para a linguagem C++ em paralelo e simulou o comportamento do fluxo bifásico óleo/água em meios heterogêneos e anisotrópicos.

Culham (1992 *apud* KILLOUGH, 1993) fez uma série de previsões sobre quais recursos os simuladores deveriam ter para atender os requisitos futuros do detalhamento de reservatórios de petróleo:

- Representação detalhada das heterogeneidades do reservatório advindas dos dados geoestatísticos;
- Inclusão de modelos físicos detalhados para a caracterização da fluidodinâmica;
- Uso de malhas refinadas para acurácia numérica;
- Uso de malhas não estruturadas e refinamento localizado;
- Gerenciamento de poços e instalações de superfícies complexas.

Hoje, os simuladores desenvolvidos pelas grandes empresas produtoras de petróleo apresentam todas essas características, e isso só é possível graças à simulação em paralelo, principalmente no que diz respeito ao detalhamento de heterogeneidades e refinamentos de malha localizados.

Empresas como a British Petroleum (SHIRALKAR et al., 2005), Exxon Mobil (BECKNER et al., 2015a; USADI et al., 2007), PetroChina (LIU et al., 2015) e Saudi Aramco (DOGRU et al., 2009; FUNG; DOGRU, 2008) vêm investindo massivamente em simulação paralela nos últimos 15 anos com o intuito de poder simular reservatórios com o máximo de acurácia possível, de preferência com o mesmo nível de refinamento dos dados geológicos de porosidade e permeabilidade. Esse nível de refino possibilita a elaboração de estratégias de produção específicas para cada campo, objetivando maximizar a produção de petróleo (DOGRU et al., 2009).

### 1.3 UTCOMP

Este trabalho foi desenvolvido no simulador UTCOMP. Ele é um simulador do tipo composicional, multifásico, multicomponente e isotérmico desenvolvido em *The University of Texas at Austin*. Ele vem sendo desenvolvido desde a década de 1990 e possui atributos importantes como: modelagem de uma segunda fase líquida de hidrocarbonetos, vários modelos de cálculo de permeabilidade relativa e viscosidade, formulações FI (*fully implicit*) e IMPSAT (*IMplicit Pressure and SATuration*), suporte a malhas não estruturadas e *corner point*, entre outros.

Neste trabalho, foi utilizada a formulação IMPEC (*IMplicit Pressure Explicit Composition*) e a equação de estado de Peng-Robinson (1976), além do EbFVM (*Element based Finite Volumes Method*) em conjunto com malhas não estruturadas.

### 1.4 Objetivos

#### 1.4.1 Geral

Este trabalho tem como objetivo a implementação da paralelização em um simulador composicional, com formulação IMPEC em conjunto com malhas não estruturadas, com o intuito principal de diminuir o tempo de computação e a quantidade de memória requerida por processo.

### 1.4.2 Específicos

Os objetivos específicos são:

- Aplicação de bibliotecas paralelas de código aberto;
- Desenvolvimento de rotinas de troca informações entre processos;
- Análise de tempo de simulação antes e depois da paralelização.

## 1.5 Escopo do trabalho

O Capítulo 1 apresenta uma descrição da simulação em reservatórios de petróleo, mostrando uma revisão de trabalhos publicados nas áreas de discretização de domínios e de paralelização. O Capítulo 2 mostra as equações governantes que descrevem o fluxo multifásico e multicomponente em meios porosos, bem como os modelos de cálculos de propriedades dos fluidos e componentes. No Capítulo 3 é mostrado o desenvolvimento matemático do EbFVM utilizado na discretização das equações da pressão e do balanço de massa. O Capítulo 4 descreve como a paralelização foi realizada, mostrando a aplicação das bibliotecas e seu impacto no código. Finalmente, os resultados são apresentados no Capítulo 5, onde são mostrados os tempos de CPU, as curvas de *speedup* e as curvas de produção de óleo e gás em diversos casos com diferentes refinamentos de malhas e configurações de poços.



## 2 EQUAÇÕES GOVERNANTES

As expressões matemáticas que descrevem o escoamento de fluidos em meios porosos são equações diferenciais parciais (EDP) não lineares difíceis de serem resolvidas. Elas envolvem derivadas espaciais nas três dimensões e derivadas temporais. Além dessas equações, é necessário que se resolvam também as relações termodinâmicas e os modelos que descrevem as propriedades dos fluidos.

Para o desenvolvimento dessas equações, são admitidas algumas hipóteses em relação ao comportamento do fluido na rocha reservatório (CHANG, 1990):

- Reservatório isotérmico.
- Reservatório cercado por zonas impermeáveis (sem fluxo nas fronteiras).
- Tensor permeabilidade é alinhado com os eixos coordenados X, Y e Z (para malhas estruturadas).
- Não ocorre precipitação, adsorção ou reação química.
- Fluxo descrito pela lei de Darcy para meios porosos.
- Formação rochosa ligeiramente compressível.
- Fluidos de injeção e produção são tratados como termos fonte (injeção é positivo, e produção é negativo).

Além dessas hipóteses, o simulador pode modelar a existência de quatro fases no reservatório: água, óleo, gás e segunda fase líquida de hidrocarbonetos. Essa fase geralmente aparece quando há mistura de CO<sub>2</sub> com hidrocarbonetos em baixas temperaturas (CHANG, 1990). Contudo, apesar dessa característica do simulador, não serão abordados estudos de caso onde ocorra a formação dessa quarta fase.

O modelo assume ainda equilíbrio termodinâmico local instantâneo, ou seja, a fugacidade dos componentes é a mesma em cada fase onde eles se encontram. É considerado também que não há mistura entre as fases água e óleo, e a água é considerada ligeiramente compressível.

### 2.1 Balanço de massa

A equação de balanço de massa é dada por:

$$\frac{1}{V_b} \frac{\partial N_i}{\partial t} + \vec{\nabla} \cdot \left( \sum_{j=1}^{n_p} \xi_j x_{ij} \vec{u}_j \right) + \vec{\nabla} \cdot \left( \sum_{j=1}^{n_p} \phi \xi_j S_j \vec{K}_{ij} \nabla x_{ij} \right) + \frac{q_i}{V_b} = 0, \quad (2.1)$$

onde:

$$N_i = \left( \phi \sum_{j=1}^{n_p} \xi_j S_j x_{ij} \right) \quad (2.2)$$

e  $\phi$  é a porosidade da rocha,  $\xi_j$  é a densidade molar da fase  $j$ ,  $S_j$  é a saturação da fase  $j$ ,  $x_{ij}$  é a fração molar do componente  $i$  na fase  $j$ ,  $\vec{u}_j$  é a velocidade da fase  $j$ ,  $\vec{K}_{ij}$  é o tensor dispersão,  $q_i$  é a vazão de injeção (ou produção),  $V_b$  é o volume do volume de controle e  $t$  é o tempo. Entretanto, devido à complexidade da equação, o termo de dispersão não será levado em consideração neste trabalho. A obtenção da equação (2.1), bem como o tratamento do termo de dispersão têm o seu desenvolvimento bem descrito no trabalho de Chang (1990).

A velocidade da fase é definida pela lei de Darcy para escoamento multifásico em meio poroso e pode ser escrita como:

$$\vec{u}_j = -\vec{k} \lambda_{rj} (\nabla P_j - \gamma_j \nabla D), \quad (2.3)$$

onde  $\vec{k}$  é o tensor permeabilidade da rocha,  $\lambda_{rj}$  é a mobilidade relativa da fase  $j$ ,  $P_j$  é a pressão da fase  $j$ ,  $\gamma_j$  é o peso específico da fase  $j$ , e  $D$  é a profundidade (positiva quando aponta para baixo).

O tensor permeabilidade da rocha pode ser escrito como:

$$\vec{k} = \begin{vmatrix} k_{xx} & k_{xy} & k_{xz} \\ k_{yx} & k_{yy} & k_{yz} \\ k_{zx} & k_{zy} & k_{zz} \end{vmatrix} \quad (2.4)$$

e representa a facilidade com a qual a rocha permite a passagem do fluido.

A mobilidade relativa é definida como:

$$\lambda_{rj} = \frac{k_{rj}}{\mu_j}, \quad (2.5)$$

onde  $k_{rj}$  é a permeabilidade relativa da fase  $j$ , e  $\mu_j$  é a viscosidade da fase  $j$ . A permeabilidade relativa da fase é função da saturação e seu cálculo depende do modelo escolhido.

A pressão é definida em relação a uma pressão de referência mais a pressão capilar da fase em questão em relação à fase de referência:

$$P_j = P_r + P_{crj}, \quad (2.6)$$

onde  $P_r$  é a pressão da fase de referência, e  $P_{crj}$  é a pressão capilar da fase  $j$  em relação à fase de referência. Nesse trabalho, a pressão de referência adotada foi a pressão da fase óleo.

Com todos os termos definidos e desconsiderando-se a dispersão, a equação de balanço de massa assume a seguinte forma para os hidrocarbonetos:

$$\frac{1}{V_b} \frac{\partial}{\partial t} \left( \phi \sum_{j=1}^{n_p} \xi_j S_j x_{ij} \right) + \vec{\nabla} \cdot \left[ \sum_{j=1}^{n_p} \xi_j x_{ij} \vec{k} \frac{k_{rj}}{\mu_j} (\nabla (P_r + P_{crj}) - \gamma_j \nabla D) \right] + \frac{q_i}{V_b} = 0, \quad (2.7)$$

e para a água:

$$\frac{1}{V_b} \frac{\partial}{\partial t} (\phi \xi_w S_w) + \vec{\nabla} \cdot \left[ \xi_w \vec{k} \frac{k_{rw}}{\mu_w} (\nabla (P_r + P_{crw}) - \gamma_w \nabla D) \right] + \frac{q_w}{V_b} = 0, \quad (2.8)$$

onde o subscrito  $w$  refere-se à fase água.

## 2.2 Equação da pressão

A equação da pressão adotada neste trabalho é oriunda do trabalho desenvolvido por Acs *et al.* (1985). Ela necessita de um esforço matemático maior para ser obtida quando comparada com a equação do balanço de massa. A expressão que define a pressão advém da restrição de que o volume de fluido ( $V_T$ ) deve ocupar todo o volume poroso da rocha ( $V_P$ ). Sendo que o volume de fluido é função da pressão e do número de moles de cada componente hidrocarboneto e da água, ao passo que o volume da rocha é função apenas da pressão, o qual resulta na seguinte equação:

$$V_T(P, N_1, N_2, \dots, N_{nc+1}) = V_P. \quad (2.9)$$

Similarmente:

$$dV_T = dV_P. \quad (2.10)$$

Trabalhemos com o lado esquerdo da Eq. (2.10). A derivada total do volume de fluido é dada por:

$$dV_T = \left( \frac{\partial V_T}{\partial P} \right)_N dP + \sum_{i=1}^{n_c+1} \left( \frac{\partial V_T}{\partial N_i} \right)_{P, N_{k(k \neq i)}} dN_i. \quad (2.11)$$

A primeira derivada no lado direito da Eq. (2.11) é a compressibilidade do fluido, e a segunda derivada é o volume molar parcial e será escrito, por simplicidade, como:

$$\left( \frac{\partial V_T}{\partial N_i} \right)_{P, N_{k(k \neq i)}} = \bar{V}_{Ti}. \quad (2.12)$$

Trabalhemos agora com o lado direito da Eq. (2.10). O volume poroso é função tanto da pressão de fluido quanto da pressão externa exercida pelas rochas ao redor do reservatório. Entretanto, neste trabalho a pressão externa não será levada em consideração, e o volume poroso é tido como função da pressão de fluido apenas. Dessa maneira, sua derivada total é:

$$dV_P = \frac{dV_P}{dP} dP, \quad (2.13)$$

onde:

$$V_P = \phi V_b. \quad (2.14)$$

O termo  $V_b$  é o volume do bloco (ou volume de controle) e a porosidade  $\phi$  é definida como:

$$\phi = \phi^\circ \left[ 1 + C_f (P - P_f) \right], \quad (2.15)$$

onde  $\phi^\circ$  é a porosidade da rocha avaliada numa pressão de referência  $P_f$ , e  $C_f$  é a compressibilidade da rocha.

Substituindo as Eqs. (2.14) e (2.15) em (2.13) e derivando, temos:

$$dV_p = V_b \phi^\circ C_f dP. \quad (2.16)$$

Substituindo as Eqs. (2.11), (2.12) e (2.16) em (2.10), temos:

$$\left( \frac{\partial V_T}{\partial P} \right)_N dP + \sum_{i=1}^{n_c+1} \bar{V}_{Ti} dN_i = V_b \phi^\circ C_f dP. \quad (2.17)$$

Diferenciando a Eq. (2.17) em relação ao tempo e rearranjando, temos:

$$\left( V_b \phi^\circ C_f - \frac{\partial V_T}{\partial P} \right) \frac{dP}{dt} = \sum_{i=1}^{n_c+1} \bar{V}_{Ti} \frac{dN_i}{dt}. \quad (2.18)$$

Dividindo todos os termos da Equação (2.18) por  $V_b$ , obtemos:

$$\left( \phi^\circ C_f - \frac{1}{V_b} \frac{\partial V_T}{\partial P} \right) \frac{dP}{dt} = \frac{1}{V_b} \sum_{i=1}^{n_c+1} \bar{V}_{Ti} \frac{dN_i}{dt}. \quad (2.19)$$

Analisando a Equação (2.19) é possível perceber que a derivada mais à direita é a variação do número de moles do componente  $i$  em relação ao tempo, ou seja, é o termo de acúmulo da equação do balanço de massa. Dessa maneira, podemos substituir a Eq. (2.7) na Eq. (2.19), a qual se torna:

$$\left( \phi^\circ C_f - \frac{1}{V_b} \frac{\partial V_T}{\partial P} \right) \frac{dP}{dt} = \sum_{i=1}^{n_c+1} \left\{ \bar{V}_{Ti} \vec{\nabla} \cdot \left[ \sum_{j=1}^{n_p} \xi_j x_{ij} \vec{k} \frac{k_{rj}}{\mu_j} \left( \nabla (P + P_{crj}) - \gamma_j \nabla D \right) \right] \right\} - \sum_{i=1}^{n_c+1} \bar{V}_{Ti} \frac{q_i}{V_b}. \quad (2.20)$$

A equação (2.20) é a forma final da equação da pressão, a qual é resolvida implicitamente no simulador. Vale salientar que o somatório do número de moles envolve também a fase água. A derivada do volume total de fluido em relação ao número de moles ( $\bar{V}_{Ti}$ ) depende da equação de estado utilizada. O desenvolvimento matemático para a obtenção dessa derivada e da derivada de volume em relação à pressão podem ser encontrados no trabalho de Chang (1990).

### 2.3 Equações de restrição

Essas equações são utilizadas como parâmetros para a verificação de cálculos e para descrever o comportamento termodinâmico dos fluidos.

A hipótese do equilíbrio de fases pode ser chamada também de critério de isofugacidade, ou seja, a fugacidade de um componente é a mesma em qualquer fase que ele esteja presente. Isso é traduzido matematicamente como:

$$f_{ij} = f_{im} \text{ e } j \neq m, \quad (2.21)$$

onde  $f$  representa a fugacidade do componente  $i$  nas fases  $j$  e  $m$ .

O somatório das saturações das fases é sempre igual a 1:

$$\sum_{j=1}^{n_p} S_j = 1. \quad (2.22)$$

As frações molares são definidas como:

$$x_{ij} = \frac{n_{ij}}{n_j}, \quad (2.23)$$

onde  $n_{ij}$  é o número de moles do componente  $i$  na fase  $j$ , e  $n_j$  é o número de moles total da fase  $j$ . O somatório de  $x_{ij}$  também é sempre igual a 1:

$$\sum_{i=1}^{n_c} x_{ij} = 1. \quad (2.24)$$

As equações das seções 2.1 a 2.3 são as principais equações do modelo, pois são utilizadas para calcular as variáveis primárias pressão ( $P$ ) e número de moles ( $N_i$ ). As demais propriedades são chamadas variáveis secundárias e são calculadas explicitamente a partir dos valores das variáveis primárias.

## 2.4 Equações de propriedades

Esta seção descreve como as propriedades são calculadas no simulador, entretanto serão mostradas apenas aquelas utilizadas nas simulações testadas no presente trabalho. Por exemplo, o simulador possui seis maneiras diferentes de calcular a permeabilidade relativa da fase, mas será mostrado apenas o modelo de Stone 2 modificado (STONE, 1973).

### 2.4.1 Viscosidade

O simulador UTCOMP possui quatro modelos de viscosidade implementados, contudo será mostrado apenas o modelo de Lohrenz *et al.* (1964). Esse

modelo é aplicado apenas nas fases de hidrocarbonetos, visto que a viscosidade da água é considerada constante. Primeiramente, são avaliadas as viscosidades dos componentes puros a baixa pressão através da correlação de Stiel e Thodos (1961):

$$\tilde{\mu}_i = \begin{cases} \frac{3,4 \cdot 10^{-4} T_{ri}^{0,94}}{\zeta_i} & \text{se } T_{ri} \leq 0,15 \\ \frac{1,776 \cdot 10^{-4} (4,58 T_{ri} - 1,67)^{0,625}}{\zeta_i} & \text{se } T_{ri} > 0,15 \end{cases}, \quad (2.25)$$

onde

$$\zeta_i = \frac{5,44 \cdot T_{ci}^{1/6}}{MM_i^{1/2} \cdot P_{ci}^{2/3}}. \quad (2.26)$$

Nas Eqs. (2.25) e (2.26),  $T_{ci}$  e  $P_{ci}$  referem-se à temperatura crítica e pressão crítica do componente  $i$  respectivamente;  $MM_i$  refere-se à massa molar do componente  $i$  e  $T_{ri}$  refere-se à temperatura reduzida do componente  $i$ , definida pela razão da temperatura atual do componente dividida por sua temperatura crítica.

A viscosidade da mistura a baixa pressão de cada fase é calculada através da correlação de Herning e Zipperer (1936):

$$\mu_j^* = \frac{\sum_{i=1}^{n_c} x_{ij} \tilde{\mu}_i \sqrt{MM_i}}{\sum_{i=1}^{n_c} x_{ij} \sqrt{MM_i}} \quad (2.27)$$

Finalmente, a viscosidade da fase a determinada pressão  $P$  é calculada pela correlação de Jossi *et al.* (1962):

$$\mu_j = \begin{cases} \mu_j^* + 2,05 \cdot 10^{-4} \cdot \frac{\xi_{jr}}{\eta_j} & \text{se } \xi_{jr} \leq 0,18 \\ \frac{\mu_j^* + (\chi_j^4 - 1)}{10^4 \eta_j} & \text{se } \xi_{jr} > 0,18 \end{cases}, \quad (2.28)$$

onde a densidade molar reduzida da fase  $\xi_{jr}$  é dada por:

$$\xi_{jr} = \xi_j \sum_{i=1}^{n_c} x_{ij} \nu_{ci}. \quad (2.29)$$

Na Eq. (2.29), o símbolo  $v_{ci}$  denota o volume molar crítico do componente  $i$ ; esse valor é dado no arquivo de *input* do programa, assim como as outras propriedades críticas dos componentes. Os parâmetros  $\chi$  e  $\eta$  são dados por:

$$\chi_j = 1,023 + 0,23364\xi_{jr} + 0,58533\xi_{jr}^2 - 0,40758\xi_{jr}^3 + 0,093324\xi_{jr}^4 \quad (2.30)$$

e

$$\eta_j = \frac{5,44 \left( \sum_{i=1}^{n_c} x_{ij} T_{ci} \right)^{1/6}}{\left( \sum_{i=1}^{n_c} x_{ij} MM_i \right)^{1/2} \left( \sum_{i=1}^{n_c} x_{ij} P_{ci} \right)^{2/3}}. \quad (2.31)$$

#### 2.4.2 Densidades molar e mássica

A densidade molar das fases de hidrocarbonetos é calculada através da equação (2.32):

$$\xi_j = \frac{P}{Z_j RT} \quad (2.32)$$

onde  $Z_j$  é o fator de compressibilidade da fase e é calculado através da equação de estado.

A água, por sua vez, é tida como ligeiramente compressível, e sua densidade molar é calculada como:

$$\xi_w = \xi_w^\circ \left[ 1 + C_w (P - P_w^\circ) \right], \quad (2.33)$$

onde  $\xi_w^\circ$  é a densidade molar da água calculada na pressão de referência  $P_w^\circ$ .

A densidade mássica, por sua vez, depende da densidade molar, tanto para a água quanto para os hidrocarbonetos:

$$\rho_w = MM_w \xi_w, \quad (2.34)$$

$$\rho_j = \xi_j \sum_{i=1}^{n_c} x_{ij} MM_i. \quad (2.35)$$

#### 2.4.3 Saturação

A saturação é calculada de acordo com as seguintes expressões:



$$S_w = \frac{N_w}{V_P \xi_w} \quad (2.36)$$

$$S_j = (1 - S_w) \frac{L_j / \xi_j}{\sum_{m=2}^{n_p} L_m / \xi_m} \quad (2.37)$$

onde a saturação da última fase é calculada através da restrição da saturação, Eq. (2.22), e  $L_j$  é a fração molar da fase  $j$  e é definida como:

$$L_j = \frac{n_j}{\sum_{m=1}^{n_p} n_m}. \quad (2.38)$$

#### 2.4.4 Permeabilidade relativa

Como mencionado anteriormente, o simulador UTCOMP possui seis maneiras de calcular a permeabilidade relativa, incluindo modelos de histerese para as fases água, óleo e gás. Entretanto, neste trabalho será apresentado apenas o modelo de Stone 2 modificado (STONE, 1973), e a histerese não será levada em consideração. Nesse modelo, as permeabilidades relativas da água e do gás são funções apenas da saturação da própria fase, enquanto a do óleo depende da saturação das três fases.

Para um sistema bifásico óleo/água, as permeabilidades relativas de cada fase são calculadas através das Equações (2.39) e (2.40):

$$k_{rw} = k_{rw}^\circ \left( \frac{S_w - S_{rw}}{1 - S_{rw} - S_{ro}} \right)^{e_w}, \quad (2.39)$$

$$k_{ro} = k_{ro}^\circ \left( \frac{S_o - S_{ro}}{1 - S_{rw} - S_{ro}} \right)^{e_w}, \quad (2.40)$$

onde  $k_r^\circ$  é a permeabilidade *end-point*,  $e$  é um expoente, e  $S_r$  é a saturação residual da fase. Esses parâmetros são obtidos experimentalmente, e o expoente é ajustável.

Para um sistema trifásico água/óleo/gás, temos:

$$k_{rw} = k_{rw}^{\circ} \left( \frac{S_w - S_{rw}}{1 - S_{rw} - S_{row}} \right)^{e_w}, \quad (2.41)$$

$$k_{rg} = k_{rg}^{\circ} \left( \frac{S_g - S_{rg}}{1 - S_{rw} - S_{rog} - S_{rg}} \right)^{e_g}, \quad (2.42)$$

$$k_{row} = k_{ro}^{\circ} \left( \frac{1 - S_w - S_{row}}{1 - S_{rw} - S_{row}} \right)^{e_{ow}}, \quad (2.43)$$

$$k_{rog} = k_{ro}^{\circ} \left( \frac{1 - S_g - S_{rw} - S_{row}}{1 - S_{rw} - S_{rog} - S_{rg}} \right)^{e_{og}}, \quad (2.44)$$

e

$$k_{ro} = k_{ro}^{\circ} \left[ \left( \frac{k_{row}}{k_{ro}^{\circ}} + k_{rw} \right) \left( \frac{k_{rog}}{k_{ro}^{\circ}} + k_{rg} \right) - (k_{rw} + k_{rg}) \right]. \quad (2.45)$$

## 2.5 Equação de estado

O simulador UTCOMP, sendo do tipo composicional, modela o comportamento de fases através de uma equação de estado (EOS na sigla em inglês). Nele estão implementadas duas EOS: Peng-Robinson (PENG; ROBINSON, 1976) e uma versão modificada da equação de Redlich-Kwong. A equação abordada neste trabalho é a de Peng-Robinson, descrita a seguir.

Primeiramente, para uma substância pura:

$$P = \frac{RT}{v-b} - \frac{a}{v(v+b)+b(v-b)}, \quad (2.46)$$

onde:

$$a = \Omega_a \frac{\alpha (RT_c)^2}{P_c}, \quad (2.47)$$

$$b = \Omega_b \frac{RT_c}{P_c}, \quad (2.48)$$

$$\Omega_a = 0,45724, \quad (2.49)$$

$$\Omega_b = 0,0778 \quad (2.50)$$

e

$$\alpha = \left\{ 1 + m \left[ 1 - \sqrt{\left( \frac{T}{T_c} \right)} \right] \right\}^2. \quad (2.51)$$

O parâmetro  $m$  é avaliado como sugerido por Peng-Robinson (1976):

$$m = \begin{cases} 0,37464 + 1,54226\omega - 0,26992\omega^2 & \text{se } \omega \leq 0,49 \\ 0,379642 + 1,48503\omega - 0,164423\omega^2 + 0,016666\omega^3 & \text{se } \omega > 0,49 \end{cases}, \quad (2.52)$$

onde  $\omega$  é o fator acêntrico.

A Eq. (2.46) pode ser escrita em função do fator de compressibilidade:

$$Z^3 - (1-B)Z^2 + (A-3B^2-2B)Z - (AB-B^2-B^3) = 0, \quad (2.53)$$

onde

$$A = \frac{aP}{(RT)^2} \quad (2.54)$$

e

$$B = \frac{bP}{RT}. \quad (2.55)$$

Para um sistema multifásico e multicomponente, as Eqs. (2.46) e (2.53) são escritas como:

$$P = \frac{RT}{v_j - b_j} - \frac{a_j}{v_j(v_j + b_j) + b_j(v_j - b_j)} \quad (2.56)$$

e

$$Z_j^3 - (1 - B_j)Z_j^2 + (A_j - 3B_j^2 - 2B_j)Z_j - (A_jB_j - B_j^2 - B_j^3) = 0. \quad (2.57)$$

Os parâmetros  $a_j$  e  $b_j$  são obtidos usando as seguintes regras de mistura:

$$a_j = \sum_{i=1}^{n_c} \sum_{k=1}^{n_c} x_{ij} x_{ik} a_{ik}, \quad (2.58)$$

onde

$$a_{ik} = (1 - \kappa_{ik})(a_i a_k)^{0.5} \quad (2.59)$$

e

$$b_j = \sum_{i=1}^{n_c} x_{ij} b_i, \quad (2.60)$$

onde  $\kappa_{ik}$  é o coeficiente de interação binária entre os componentes da mistura. Como a equação de Peng-Robinson é uma equação de estado cúbica, ela pode ter até três raízes reais como solução. Se for obtida mais de uma raiz real, é feito um laço para identificar e escolher a que apresentar a menor energia livre de Gibbs (FERNANDES, 2014; PERSCHKE, 1988).

Finalmente, o fator de compressibilidade é utilizado no cálculo do coeficiente de fugacidade  $\phi_{ij}$  do componente  $i$  na fase  $j$ . Fazendo uso da EOS de Peng-Robinson, esse coeficiente pode ser calculado como:

$$\ln \phi_{ij} = \frac{b_i}{b_j} (Z_j - 1) - \ln(Z_j - B_j) - \frac{A_j}{2\sqrt{2}B_j} \left[ 2 \sum_{k=1}^{n_c} \left( \frac{x_{kj} a_{ik}}{a_j} \right) - \frac{b_i}{b_j} \right] \ln \left( \frac{Z_j + (1 + \sqrt{2})B_j}{Z_j + (1 - \sqrt{2})B_j} \right). \quad (2.61)$$

### 3 FORMULAÇÕES MATEMÁTICAS

As equações da pressão e do balanço de massa apresentadas no capítulo 2 são equações diferenciais parciais, não-lineares e, portanto, não possuem solução analítica. É necessário, portanto, a utilização de um método numérico para resolvê-las. A formulação empregada neste trabalho é a IMPEC (*IMplicit Pressure Explicit Composition*), na qual as variáveis primárias são a pressão e o número de moles. Sendo que a pressão é resolvida implicitamente, e em seguida é resolvido o número de moles explicitamente. Numa etapa posterior, são resolvidas todas as outras variáveis (saturação, frações molares, densidades, etc.), chamadas variáveis secundárias.

A seção 3.1 apresenta as equações aproximadas obtidas pela integração das Eqs. (2.7) e (2.20). A seção 3.2 apresenta o Método de Volumes Finitos baseado em Elementos (EbFVM) e a discretização das equações da pressão e de balanço de massa utilizando esse método.

#### 3.1 Equações aproximadas

A equação (2.20), quando integrada no volume e no tempo, assume a forma:

$$\left( \phi^c C_f - \frac{1}{V_b} \frac{\partial V_T^n}{\partial P} \right) (P^{n+1} - P^n) = (V_T^n - V_P^n) + \Delta t \sum_{i=1}^{n_c+1} \bar{V}_{Ti} \int_{V,t} \frac{\partial N_i^n}{\partial t} \partial V \partial t \quad (3.1)$$

onde o primeiro termo entre parêntesis do lado direito da igualdade é um termo de correção volumétrica sugerido por Acs *et al.* (1985). Ele corrige a discrepância entre o volume total de fluido e o volume poroso calculados no passo de tempo anterior. Isso permite que seja realizado apenas um cálculo *flash* por passo de tempo. O leitor pode obter mais informações sobre o surgimento desse termo em Acs *et al.* (1985) e Watts (1986).

A equação (2.7), quando integrada no volume e no tempo assume a forma:

$$\frac{1}{V_b} \int_{V,t} \frac{\partial N_i}{\partial t} \partial V \partial t = - \int_{V,t} \vec{\nabla} \cdot \left[ \sum_{j=1}^{n_p} \xi_j x_{ij} \frac{k_{rj}}{\mu_j} (\nabla(P + P_{crj}) - \gamma_j \nabla D) \right] \partial V \partial t - \int_{V,t} \frac{q_i}{V_b} \partial V \partial t \quad (3.2)$$

Aplicando o teorema de Gauss e rearranjando:

$$\left( N_i^{n+1} - N_i^n \right) = -\Delta t \int_A \left[ \sum_{j=1}^{n_p} \xi_j^n x_j^n \bar{k} \frac{k_{rj}^n}{\mu_j^n} \left( \nabla \left( P^{n+1} + P_{crj}^n \right) - \gamma_j^n D \right) \right] dA - \Delta t q_i^n. \quad (3.3)$$

Entretanto, o número de moles é avaliado explicitamente no instante tempo  $n+1$ . Dessa maneira, a Eq. (3.3) é reescrita de forma que todos os termos avaliados no instante de tempo  $n$  estejam no lado direito:

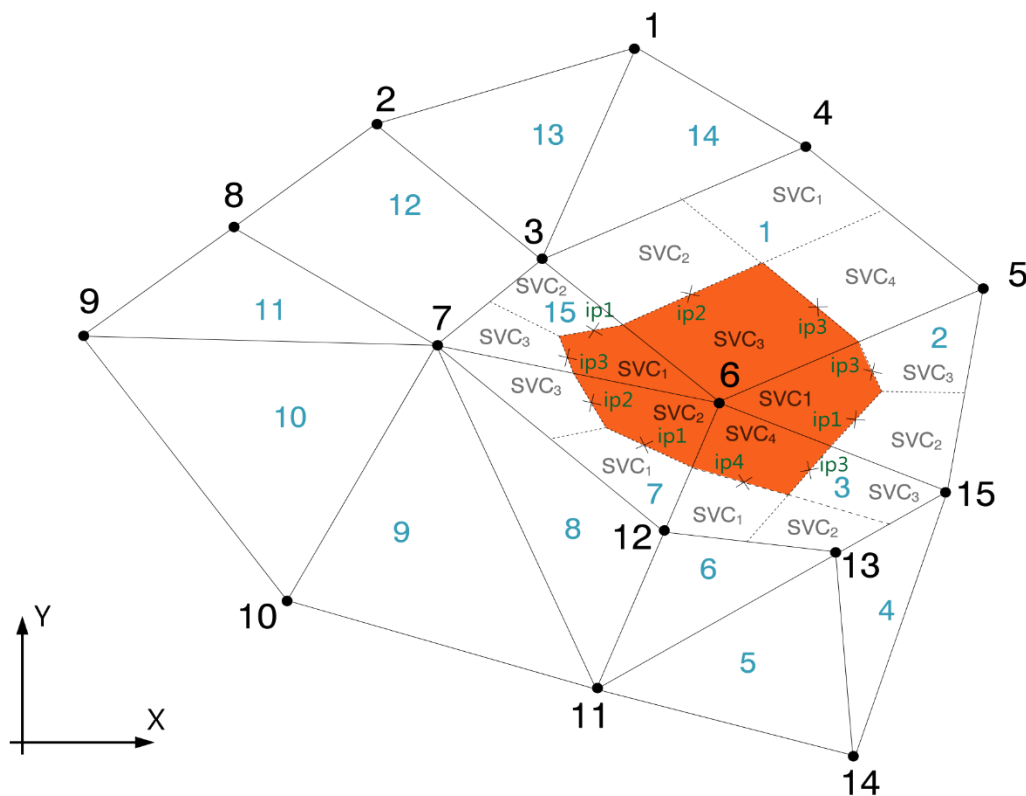
$$N_i^{n+1} = -\Delta t \int_A \left[ \sum_{j=1}^{n_p} \xi_j^n x_j^n \bar{k} \frac{k_{rj}^n}{\mu_j^n} \left( \nabla \left( P^n + P_{crj}^n \right) - \gamma_j^n D \right) \right] dA - \Delta t q_i^n + N_i^n. \quad (3.4)$$

Nas Eqs. (3.1) a (3.4), os sobrescritos  $n$  e  $n+1$  significam que a propriedade está sendo avaliada nos tempos anterior e atual respectivamente. A maneira como essas equações estão escritas é geral, ou seja, independem da malha. O próximo passo é especificar como se dá a integração para malhas não estruturadas, de maneira que a área utilizada no cálculo do fluxo de massa e o valor das propriedades dentro do elemento sejam calculados corretamente.

### 3.2 Método dos Volumes Finitos baseado em Elementos

No EbFVM, o domínio é dividido em elementos, que podem ser estruturados ou não, e esses elementos são divididos em subelementos. O volume do controle (VC) onde os balanços são avaliados é formado pelo ajuntamento dos subelementos que compartilham um mesmo vértice da malha. Por essa razão, os subelementos são também chamados sub-volumes do controle (SVC), que será o nome adotado neste trabalho. Um exemplo de malha não estruturada é mostrado na Figura 4. Nela, é destacado um volume de controle em laranja. Ele é formado pela união de todos os sub-volumes de controle que compartilham o vértice 6. Os números em preto identificam os vértices e os números em azul identificam os elementos. Os números em verde identificam os *integration points*, que serão explicados mais adiante. Dessa maneira, pode-se observar que o VC-6 é formado pelo SVC-3 do elemento 1, mais o SVC-1 do elemento 2, mais o SVC-4 do elemento 3, mais o SVC-2 do elemento 7, mais o SVC-1 do elemento 15. Essa operação é realizada em todos os vértices de maneira que abranja todo o domínio.

Figura 4 – Malha não estruturada com ênfase num volume de controle



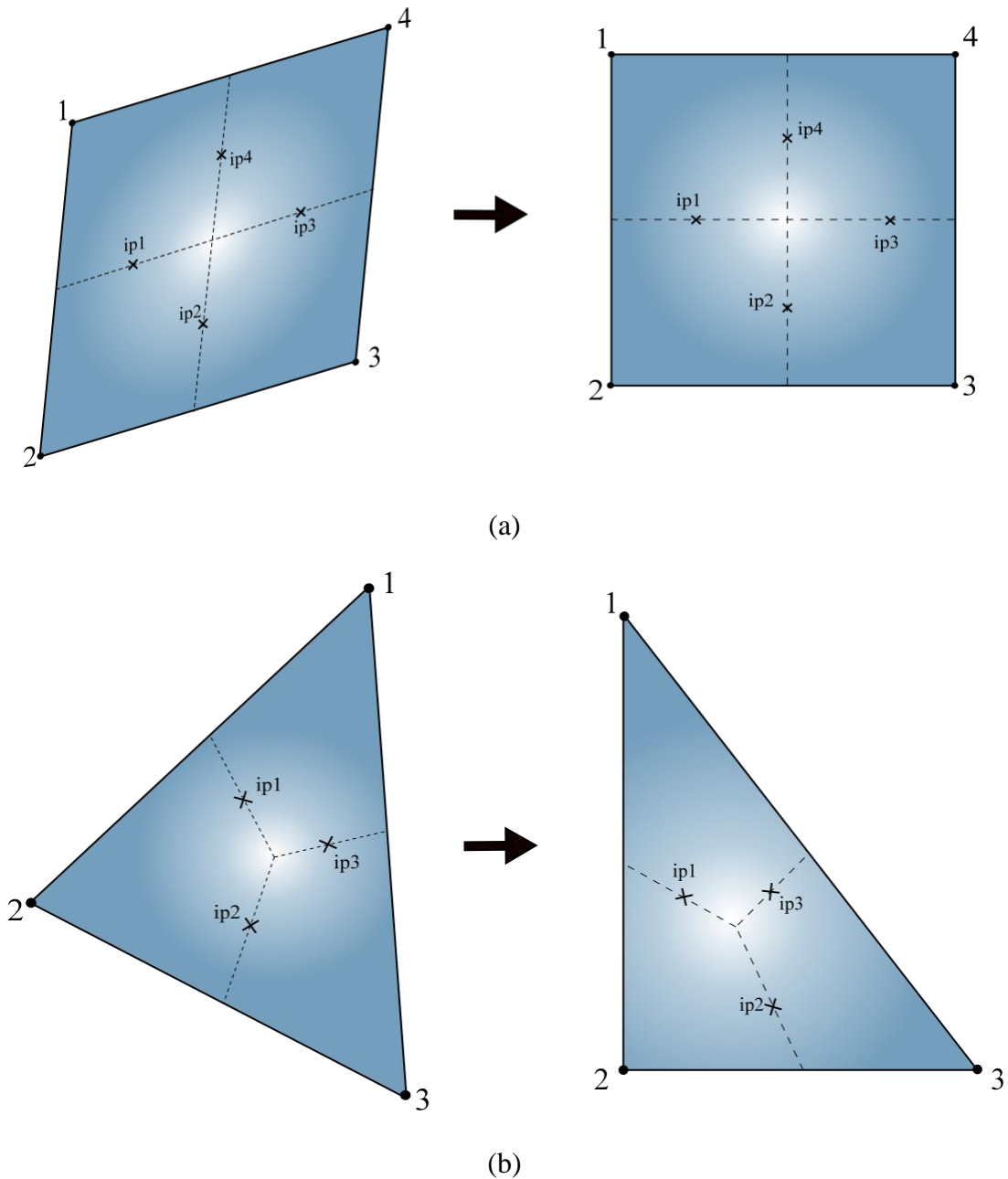
Fonte: Elaborada pelo autor.

As principais vantagens desse método são: ele permite o uso de malhas não estruturadas na discretização do domínio, as quais melhor representam geometrias complexas; ele adota uma abordagem conservativa das equações de balanço, mesmo para malhas não estruturadas; e todos os cálculos são feitos dentro do elemento de malha, o que é muito conveniente na hora de incluir o tensor permeabilidade de rocha e a porosidade variável nas equações. É importante salientar que o fluxo ocorre apenas nas fronteiras do SVC que são internas ao elemento de malha, ou seja, olhando para a Figura 4, o fluxo do SVC-3 do elemento 1 para o SVC-1 do elemento 2 e o fluxo do SVC-1 do elemento 2 para o SVC-3 do elemento 1 são iguais em módulo, mas com sinais trocados, de forma que esses fluxos são automaticamente cancelados.

O método em questão utiliza funções de forma para avaliar os termos de fluxo nas fronteiras do volume de controle. A Figura 5 mostra um exemplo de quadrilátero e triângulo transformados do plano físico para o computacional, onde as equações são resolvidas. Os ip's indicados representam os *integration points*, ou pontos de integração.

Eles são o local do elemento onde são avaliados os fluxos para as equações da pressão e do balanço de massa.

Figura 5 – Elementos transformados: (a) quadrilátero e (b) triângulo



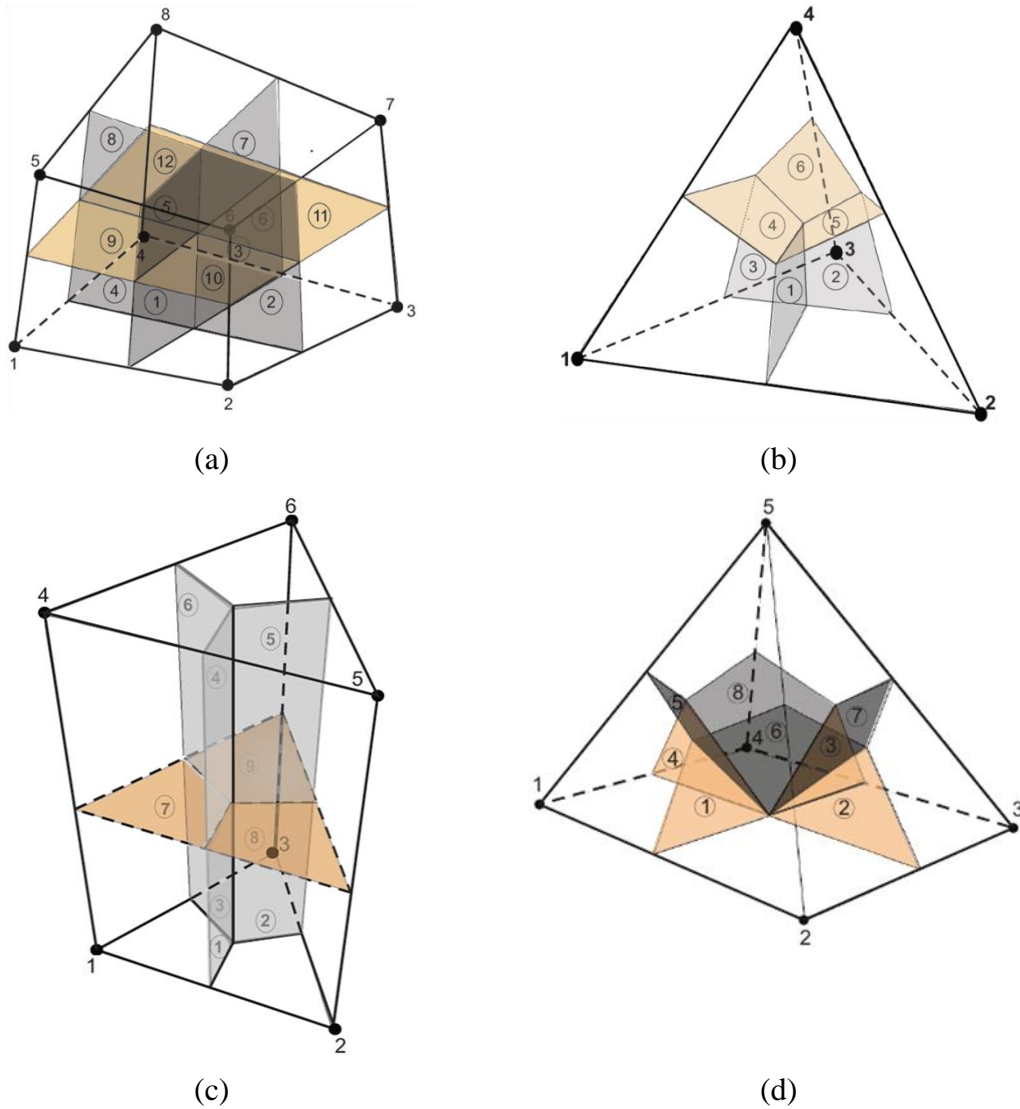
Fonte: Elaborada pelo autor.

Dessa maneira, não importa o quão distorcido está o elemento na malha, ele será sempre regular no plano computacional. Em geral, o formato do elemento não acarreta nenhuma instabilidade numérica, visto que as equações aproximadas para o escoamento multifásico são obtidas a partir das equações diferenciais para o escoamento multicomponente/multifásico. A Figura 5 mostra os dois elementos bidimensionais por



simplicidade. Vale ressaltar que a transformação ocorre também em elementos 3D, os quais são: tetraedro, pirâmide, prisma e hexaedro. A Figura 6 mostra esses elementos já transformados para o plano computacional e divididos em sub-volumes de controle.

Figura 6 – Elementos 3D: (a) hexaedro, (b) tetraedro, (c) prisma e (d) pirâmide



Fonte: FERNANDES, 2014.

Essa transformação do plano físico para o plano computacional é feita através das funções de forma. As Eqs. (3.5) e (3.6) mostram as funções de forma para triângulos e quadriláteros, enquanto as Eqs. (3.7) a (3.10) referem-se a tetraedros, pirâmides, prismas e hexaedros respectivamente.

$$\begin{aligned}
N_1(\xi, \eta) &= 1 - \xi - \eta; \\
N_2(\xi, \eta) &= \xi; \\
N_3(\xi, \eta) &= \eta.
\end{aligned} \tag{3.5}$$

$$\begin{aligned}
N_1(\xi, \eta) &= \frac{1}{4}(1 - \xi)(1 - \eta); \\
N_2(\xi, \eta) &= \frac{1}{4}(1 + \xi)(1 - \eta); \\
N_3(\xi, \eta) &= \frac{1}{4}(1 + \xi)(1 + \eta); \\
N_4(\xi, \eta) &= \frac{1}{4}(1 - \xi)(1 + \eta).
\end{aligned} \tag{3.6}$$

$$\begin{aligned}
N_1(\xi, \eta, \gamma) &= 1 - \xi - \eta - \gamma; \\
N_2(\xi, \eta, \gamma) &= \xi; \\
N_3(\xi, \eta, \gamma) &= \eta; \\
N_4(\xi, \eta, \gamma) &= \gamma.
\end{aligned} \tag{3.7}$$

$$\begin{aligned}
N_1(\xi, \eta, \gamma) &= \frac{1}{4} \left[ (1 - \xi)(1 - \eta) - \gamma + \frac{\xi\eta\gamma}{1 - \gamma} \right]; \\
N_2(\xi, \eta, \gamma) &= \frac{1}{4} \left[ (1 + \xi)(1 - \eta) - \gamma - \frac{\xi\eta\gamma}{1 - \gamma} \right]; \\
N_3(\xi, \eta, \gamma) &= \frac{1}{4} \left[ (1 + \xi)(1 + \eta) - \gamma - \frac{\xi\eta\gamma}{1 - \gamma} \right]; \\
N_4(\xi, \eta, \gamma) &= \frac{1}{4} \left[ (1 - \xi)(1 + \eta) - \gamma - \frac{\xi\eta\gamma}{1 - \gamma} \right]; \\
N_5(\xi, \eta, \gamma) &= \gamma.
\end{aligned} \tag{3.8}$$

$$\begin{aligned}
N_1(\xi, \eta, \gamma) &= (1 - \xi - \eta)(1 - \gamma); \\
N_2(\xi, \eta, \gamma) &= \xi(1 - \gamma); \\
N_3(\xi, \eta, \gamma) &= \eta(1 - \gamma); \\
N_4(\xi, \eta, \gamma) &= \gamma(1 - \xi - \eta); \\
N_5(\xi, \eta, \gamma) &= \xi\gamma; \\
N_6(\xi, \eta, \gamma) &= \eta\gamma.
\end{aligned} \tag{3.9}$$

$$\begin{aligned}
N_1(\xi, \eta, \gamma) &= \frac{1}{8}(1 + \xi)(1 - \eta)(1 + \gamma); \\
N_2(\xi, \eta, \gamma) &= \frac{1}{8}(1 + \xi)(1 - \eta)(1 - \gamma); \\
N_3(\xi, \eta, \gamma) &= \frac{1}{8}(1 - \xi)(1 - \eta)(1 - \gamma); \\
N_4(\xi, \eta, \gamma) &= \frac{1}{8}(1 - \xi)(1 - \eta)(1 + \gamma); \\
N_5(\xi, \eta, \gamma) &= \frac{1}{8}(1 + \xi)(1 + \eta)(1 + \gamma); \\
N_6(\xi, \eta, \gamma) &= \frac{1}{8}(1 + \xi)(1 + \eta)(1 - \gamma); \\
N_7(\xi, \eta, \gamma) &= \frac{1}{8}(1 - \xi)(1 + \eta)(1 - \gamma); \\
N_8(\xi, \eta, \gamma) &= \frac{1}{8}(1 - \xi)(1 + \eta)(1 + \gamma),
\end{aligned} \tag{3.10}$$

onde  $\xi$ ,  $\eta$  e  $\gamma$  são os eixos coordenados no plano computacional, ou plano transformado. Cada subscrito de  $N$  indica que a expressão em questão possui valor 1 no vértice do elemento indicado pelo subscrito, e valor 0 nos demais vértices.

Os gradientes dentro do elemento podem ser avaliados através das funções de forma de acordo com as seguintes expressões:

$$\frac{\partial \phi}{\partial x} = \sum_{i=1}^{n_y} \frac{\partial N_i}{\partial x} \Phi_i; \quad \frac{\partial \phi}{\partial y} = \sum_{i=1}^{n_y} \frac{\partial N_i}{\partial y} \Phi_i; \quad \frac{\partial \phi}{\partial z} = \sum_{i=1}^{n_y} \frac{\partial N_i}{\partial z} \Phi_i \tag{3.11}$$

onde  $n_v$  é o número de vértices do elemento e  $\Phi_i$  é o valor da propriedade no vértice.

A área das interfaces dos sub-volumes de controle para elementos 2D é dada por:

$$d\vec{A} = h(dy\hat{i} - dx\hat{j}), \quad (3.12)$$

onde  $h$  é a espessura do reservatório e  $dx$  e  $dy$  são avaliados no sentido anti-horário. Para elementos 3D, a expressão da área é dada por:

$$d\vec{A} = \left( \frac{\partial y}{\partial m} \frac{\partial z}{\partial n} - \frac{\partial y}{\partial n} \frac{\partial z}{\partial m} \right) dmdn\hat{i} - \left( \frac{\partial x}{\partial n} \frac{\partial z}{\partial m} - \frac{\partial x}{\partial m} \frac{\partial z}{\partial n} \right) dmdn\hat{j} + \left( \frac{\partial x}{\partial m} \frac{\partial y}{\partial n} - \frac{\partial x}{\partial n} \frac{\partial y}{\partial m} \right) dmdn\hat{k}, \quad (3.13)$$

onde  $m$  e  $n$  são quaisquer dois eixos coordenados  $\xi$ ,  $\eta$  ou  $\gamma$ . Vale salientar que  $d\vec{A}$  é definido como um vetor normal à superfície da interface do sub-volume de controle, seu módulo é igual à área dessa interface e ele sempre aponta para fora do volume de controle. Maiores detalhes sobre essa discretização podem ser obtidos nos trabalhos de Fernandes (2014), Maliska (2014) e Marcondes e Sepehrnoori (2010).

Apesar de a transmissibilidade ser definida apenas para a avaliação do fluxo entre dois vértices, pode-se definir uma pseudo-transmissibilidade para o EbFVM como:

$$T_{ihl} = \vec{k}_l \cdot \vec{\nabla} N_{h,l} \cdot d\vec{A}_l \quad (3.14)$$

onde  $l$  é o ponto de integração,  $i$  é o sub-volume de controle e  $h$  é o vértice, e todos são avaliados dentro de um mesmo elemento da malha. O termo  $\vec{\nabla} N_{h,l}$  representa a função de forma adequada ao elemento em questão, avaliada no ponto de integração  $l$  e referente ao vértice  $h$ .

Dessa forma, substituindo a Eq. (3.14) na (3.4), tem-se:

$$N_i^{n+1} = -\Delta t \sum_{j=1}^{n_p} \sum_{l=1}^{n_{ip}} \left[ \xi_j^n x_j^n \frac{k_{rj}^n}{\mu_j^n} \left( \sum_{h=1}^{n_v} T_{ihl} (P^n + P_{crj}^n) - \gamma_j^n D \right) \right] - \Delta t q_i^n + N_i^n. \quad (3.15)$$

A equação da pressão, Eq. (3.1), assume a seguinte forma quando discretizada no EbFVM:

$$\left( V_{svc,h} \phi^{\circ} C_f - \frac{V_{svc,h}}{V_b} \frac{\partial V_T^n}{\partial P} \right) (P^{n+1} - P^n) - \frac{V_{svc,h}}{V_b} (V_T^n - V_P^n) =$$

$$\Delta t \sum_{i=1}^{n_c+1} \bar{V}_{Ti} \left\{ \sum_{j=1}^{n_p} \sum_{l=1}^{n_{ip}} \left[ \xi_j^n x_j^n \frac{k_{rj}^n}{\mu_j^n} \left( \sum_{h=1}^{n_v} T_{ihl} (P^n + P_{crj}^n) - \gamma_j^n D \right) \right] - \Delta t q_i^n \right\} \quad (3.16)$$

As Eqs. (3.15) e (3.16) são a forma final das equações de balanço de massa e da pressão respectivamente. A maneira como elas estão escritas é válida tanto para 2D quanto 3D, onde devem ser aplicadas as funções de forma correspondentes. É importante enfatizar que tanto a pressão quanto o número de moles são resolvidos em cada vértice da malha. As funções de forma são utilizadas para avaliar o gradiente da pressão nos pontos de integração, que é onde ocorre o fluxo, contudo a pressão é resolvida no vértice. O termo de poço depende da condição de operação dele, se é injetor ou produtor com vazão ou pressão prescrita. Sua análise pode ser conferida em Fernandes (2014a) e Fernandes *et al.* (2014b).

A sequência de cálculo no simulador dá-se da seguinte maneira: a equação da pressão, Eq. (3.16), é resolvida implicitamente; em seguida é resolvido o balanço molar, Eq. (3.15), explicitamente; o passo seguinte é o cálculo de estabilidade de fases e o cálculo *flash*; finalmente, as demais propriedades são resolvidas baseadas na pressão e frações molares recém obtidas.

Foi utilizada a biblioteca PETSc para a resolução da equação da pressão em paralelo. O solver utilizado foi o Block-Jacobi/GMRES. Maiores detalhes na seção 4.3.

## 4 PARALELIZAÇÃO

A paralelização é um procedimento que consiste em reescrever as rotinas de forma que o código possa ser executado por mais de um processador ao mesmo tempo. O objetivo da paralelização é, principalmente, a redução do tempo de cálculo, visto que o programa será executado simultaneamente por mais de um processador. Em segundo lugar, espera-se uma redução do espaço de memória utilizado, pois cada processo será responsável por resolver apenas seu subdomínio da malha.

A maior vantagem da paralelização é, obviamente, poder resolver casos que envolvam malhas muito refinadas em tempo hábil. Quando se trabalha com casos desse tipo em serial, toda a memória requerida pelas propriedades geométricas da malha e pelas propriedades físicas dos fluidos é alocada no espaço de um processador, e ele é responsável por resolver todo o domínio. Seria necessário um computador com memória e capacidade de processamento consideráveis para resolver casos desse tipo. Em paralelo, a situação muda, pois o domínio é dividido em subdomínios, e cada processador é responsável apenas pela sua parte. Dessa maneira, é necessária uma quantidade menor de memória por processador, e os cálculos são processados em um tempo menor quando comparados com a situação em serial.

Outro benefício proporcionado pela paralelização está na etapa de otimização dos parâmetros de produção. Normalmente, os engenheiros precisam testar vários cenários diferentes até obter-se a melhor configuração, sempre objetivando maximizar a produção. Com a simulação em paralelo, é possível realizar mais testes no mesmo intervalo de tempo.

Existem três tipos básicos de paralelização: a de memória compartilhada, a de memória distribuída, e a paralelização híbrida, que é um modelo combinado dos outros dois. A paralelização de memória compartilhada consiste num computador com uma memória onde todos os processadores presentes têm acesso a qualquer espaço dela. Os processadores interagem modificando os dados presentes na memória, que pode ser local (exclusiva a um processador) ou global (comum a todos os processadores). Por outro lado, a paralelização de memória distribuída consiste em uma plataforma com  $p$  nós de processamento, cada um com seu endereço de memória exclusivo. Em tais plataformas, a interação entre os processos se dá através da passagem de mensagens, ou *message passing*. Essa troca de mensagens é utilizada para transferir dados, trabalho e para sincronizar as ações entre os processos (GRAMA et al., 2003). Este trabalho foi

desenvolvido baseado no modelo de memória distribuída, visto que assim ele se beneficia da divisão do domínio entre os processadores envolvidos.

A paralelização de um código envolve diversas etapas: balanceamento de carga, divisão de domínio, troca de informações entre processos e resolução do sistema linear em paralelo. Na elaboração deste trabalho, foram utilizadas bibliotecas de código aberto auxiliares em cada uma dessas etapas, a saber: FMDB, Zoltan, ParMetis, OpenMPI e PETSc, as quais são descritas a seguir.

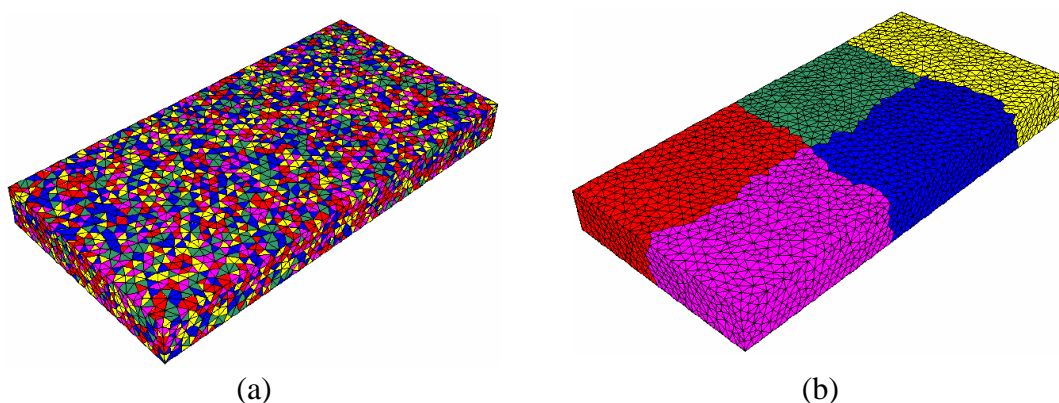
#### **4.1 Balanceamento de carga e divisão de domínio**

As etapas mais importantes na computação paralela são a divisão de domínio e o balanceamento de carga. São elas que determinam, de certa forma, a quantidade de trabalho que cada processador vai realizar durante a simulação. O balanceamento de carga significa quantos nós, ou vértices, devem ir para cada processo, e sua importância reside no fato de que cada processo deve possuir, aproximadamente, o mesmo número de nós, evitando assim que um processo fique mais sobrecarregado que outro. Esse equilíbrio faz com que os processos realizem, aproximadamente, a mesma quantidade de tarefas. Obviamente, a carga de cada processo raramente é exatamente a mesma, principalmente tratando-se de malhas não estruturadas, pois a divisão do número de nós pelo número de processos raramente é exata.

A divisão de domínio define quais elementos devem ir para cada processo. A malha deve ser dividida de forma que a comunicação entre os processos seja minimizada. Adote-se como exemplo uma malha com 4000 elementos dividida em cinco processos. Uma maneira fácil, mas ineficiente, seria simplesmente designar os 800 primeiros nós ao processo 1, os 800 seguintes ao processo 2, e assim por diante, na ordem em que aparecessem no arquivo fornecido pelo gerador de malhas. Dessa maneira, seria gerada uma divisão onde o número de elementos na fronteira entre os processos poderia ser muito grande. Isso torna o simulador ineficiente, pois assim é perdido muito tempo com a troca de informações entre os processos. Por outro lado, quando se usa uma biblioteca desenvolvida especificamente para tal função, o número de elementos fronteiros é bastante reduzido, refletindo diretamente no tempo de comunicação. A Figura 7 mostra uma malha dividida em cinco processos, onde cada cor representa um processo. A Figura 7a mostra a malha dividida de maneira ineficiente, e a Figura 7b reflete um processo de divisão otimizado (SILVA, 2008). As bibliotecas Zoltan (DEVINE et al., 2002) e

ParMetis (*Parallel Graph Partitioning and Fill-reducing Matrix Ordering*) (LASALLE; KARYPIS, 2013; SCHLOEGEL; KARYPIS; KUMAR, 2002) atuam em conjunto para que as etapas de balanceamento de carga e divisão de domínio sejam executadas de maneira rápida e eficiente.

Figura 7 – Exemplos de divisão de domínio  
a) seguindo o gerador de malhas; b) divisão otimizada pela biblioteca.



Fonte: (SILVA, 2008).

#### 4.1.1 Particionamento *k*-way

O método de divisão da malha pela biblioteca ParMetis é chamado método de particionamento *k*-way, onde *k* é o número de subdomínios desejados. Nesse método, as malhas podem ser particionadas como grafos duais ou nodais. Esse modelo baseia-se na minimização do número de arestas do grafo que são cortadas pelo particionamento *edge-cut*. Na etapa de engrossamento (*coarsening phase*), alguns vértices são colapsados de maneira a reduzir o tamanho do grafo. Em seguida, é realizado o particionamento *edge-cut* no grafo grosseiro (*initital partitioning phase*). Posteriormente, o grafo particionado é refinado novamente (*refining phase*) de maneira que volte ao tamanho original. O resultado desse processo é a malha dividida. Obviamente, o número de etapas de engrossamento é igual ao número de etapas de refino, e esse número depende da quantidade de vértices da malha.

## 4.2 Gerenciamento de informações

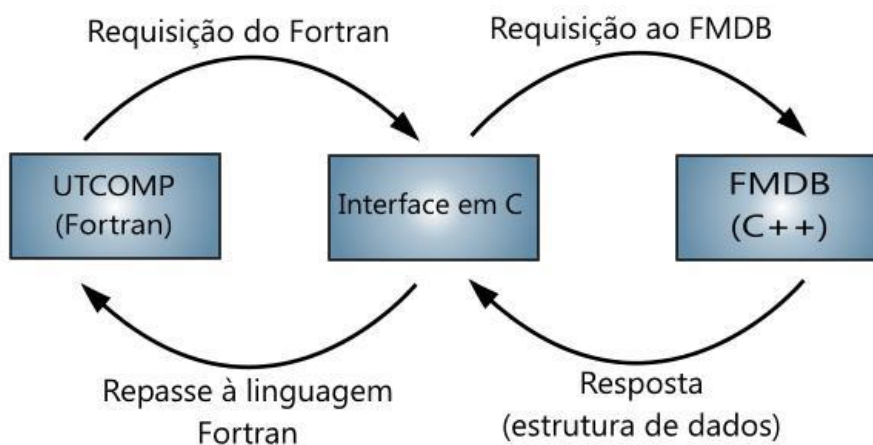
Para se definir uma malha são necessárias as seguintes informações: número de elementos, número de vértices, coordenadas (X, Y, Z) dos vértices, índices dos vértices e conectividades dos elementos (numeração dos vértices que formam cada elemento).



Todas essas informações precisam ser gerenciadas em paralelo de maneira que cada processo saiba com que parte da malha ele deve trabalhar. A biblioteca FMDB (*Flexible distributed Mesh DataBase*) (SEOL, 2005) fornece todas as ferramentas necessárias para realizar esse gerenciamento. Assim, cada processo chama uma função do FMDB que fornece a informação de malha desejada.

O FMDB é acoplado ao Zoltan e ao ParMetis, ou seja, possui uma interface de comunicação com essas bibliotecas. Assim, todas as informações geradas por elas são passadas ao FMDB, que, por sua vez, as transmite para o simulador. Pelo fato de o FMDB ser construído em C++, e o simulador UTCOMP em FORTRAN, foi necessário criar uma interface de comunicação entre essas duas linguagens. A linguagem FORTRAN já possui um conjunto de funções que permite sua comunicação com a linguagem C. Esta, por sua vez, é capaz de comunicar-se com a linguagem C++. Dessa maneira, a interface de comunicação entre o FMDB e o simulador foi desenvolvida em C, e é através dela que o simulador tem acesso às informações do FMDB. A Figura 8 descreve a interface de comunicação entre as linguagens:

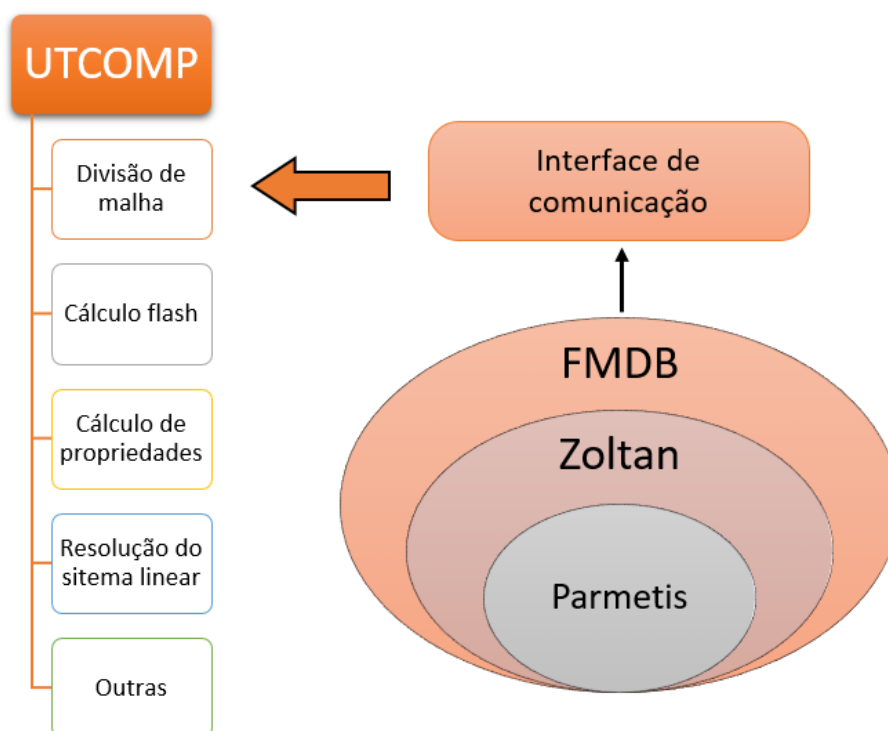
Figura 8 – Esquema de comunicação entre UTCOMP e FMDB



Fonte: Elaborada pelo autor.

O desenvolvimento dessa interface e a implementação e utilização das bibliotecas mencionadas acima foram feitos com a ajuda de integrantes do LIA (Laboratórios de Pesquisa em Computação). A Figura 9 mostra um esquema geral de como as bibliotecas se comunicam com o simulador e em que etapa da simulação elas atuam:

Figura 9 – Esquema de chamada das bibliotecas no simulador



Fonte: Elaborada pelo autor.

### 4.3 PETSc

Um dos maiores gargalos de simuladores em paralelo é o *solver* do sistema linear. Os tempos gastos no cálculo dos coeficientes da matriz jacobiana e na resolução do sistema linear somados podem chegar a 70% do tempo total de simulação dependendo do caso (KILLOUGH, 1993; LIU et al., 2015). Para que o simulador tenha um bom desempenho, é necessário que seja implementado um *solver* robusto e eficiente, capaz de resolver o sistema linear em paralelo. Claramente, o desenvolvimento de tal *solver* demanda muito tempo e foge ao escopo do que é proposto neste trabalho. Portanto, foi utilizada a biblioteca PETSc (*Portable Extensible Toolkit for Scientific Computation*) (BALAY et al., 2015) para tal propósito.

Ela dispõe de várias estruturas de dados que auxiliam no tratamento de matrizes e vetores e suas operações matemáticas. Além disso, ela possui vários *solvers* e pré-condicionadores embutidos, o que dá ao usuário a flexibilidade de escolher qual combinação *solver*/pré-condicionador é mais indicada para sua aplicação. Vale destacar que todos os *solvers* e pré-condicionadores são editáveis, por exemplo, parâmetros como número máximo de iterações, tolerância e grau de *fill-in* podem ser definidos pelo usuário,

ou lidos de um arquivo de entrada. Usando essa biblioteca, o usuário não precisa se preocupar em saber como o *solver* opera de fato, nem como ele realiza a transferência de mensagens. O solver padrão em paralelo é o Block-Jacobi para a decomposição do domínio e o GMRES para a resolução do sistema local. Todos esses fatores, além do bom desempenho apresentado pela biblioteca em outros trabalhos (HAN et al., 2007; SILVA, 2008), tornaram o PETSc a escolha natural para o desenvolvimento deste simulador.

#### 4.4 Interface de Passagem de Mensagens

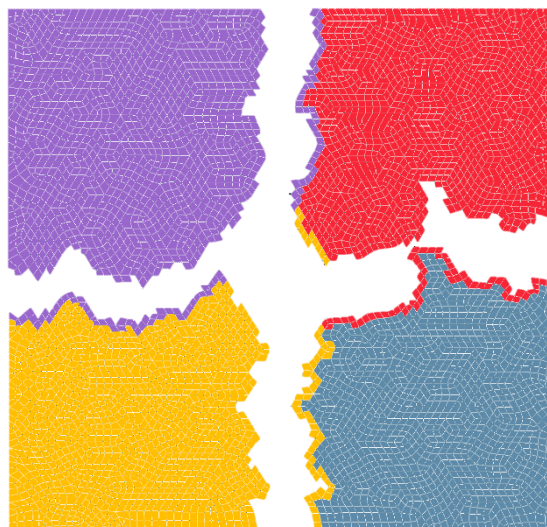
Todas as operações em paralelo, ou seja, tudo aquilo que envolve comunicação entre os processos, realizadas tanto nas bibliotecas externas quanto no simulador são desempenhadas pelas funções da biblioteca OpenMPI (GABRIEL et al., 2004), que é uma das muitas implementações de bibliotecas baseadas em MPI (*Message Passing Interface*). Bibliotecas baseadas em MPI são as mais utilizadas no desenvolvimento de programas em paralelo baseados em memória distribuída.

A OpenMPI compreende um conjunto de funções que permite a troca de informações entre os processos envolvidos na execução de um programa. Ela possui funções de envio e recebimento síncronos (MPI\_Send e MPI\_Recv) e assíncronos (MPI\_Isend e MPI\_Irecv), comunicação global (MPI\_Bcast, MPI\_Gather, etc.), sincronização (MPI\_Barrier), entre outras. Ela é uma biblioteca extremamente flexível e otimizada para obter o melhor desempenho de comunicação independentemente do ambiente onde é utilizada.

Como existe uma implementação de MPI específica para cada uma das linguagens FORTRAN, C e C++, sua aplicação é direta, tanto nas bibliotecas, quanto no simulador.

#### 4.5 Ghost layers

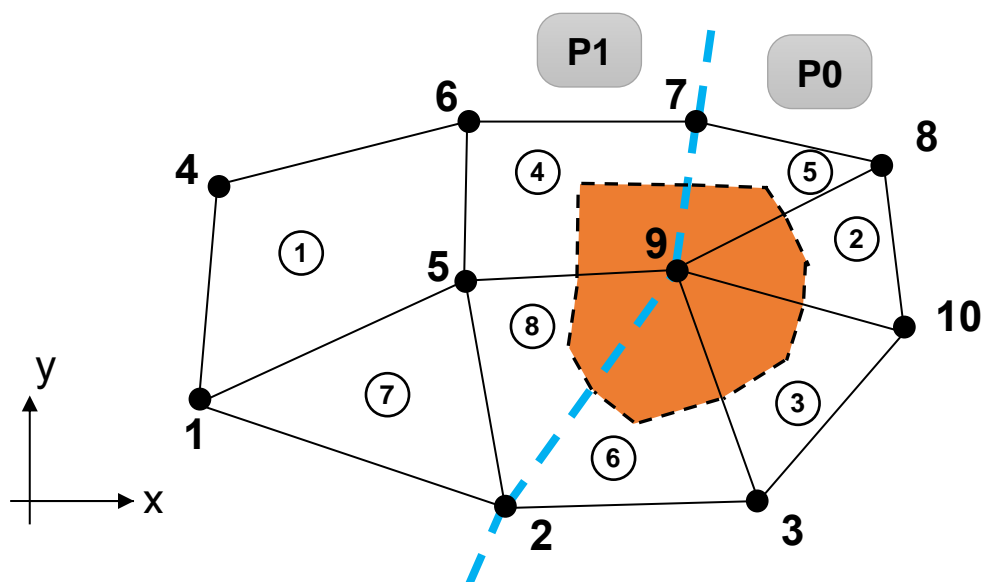
Um conceito fundamental para o desenvolvimento de um bom simulador, não só de reservatórios de petróleo, mas de fluidodinâmica em geral, são as *ghost layers*. A *ghost layer* pode ser definida como sendo a camada de elementos fronteiros de um processo, onde tais elementos pertencem a processos vizinhos. Isso é exemplificado na Figura 10.

Figura 10 – Malha dividida em 4 processos com *ghost-layer*

Fonte: Elaborada pelo autor

A pergunta que se faz é: por quê um processo precisa de um elemento que não é seu, visto que os cálculos são realizados apenas em elementos e vértices próprios? Para responder essa questão, é necessário analisar como o EbFVM calcula o termo de fluxo nas equações de balanço. Suponha que uma malha foi dividida em dois processos, P0 e P1, assim como apresentada na Figura 11, e suponha também que os vértices 2, 7 e 9 pertençam ao processo 0.

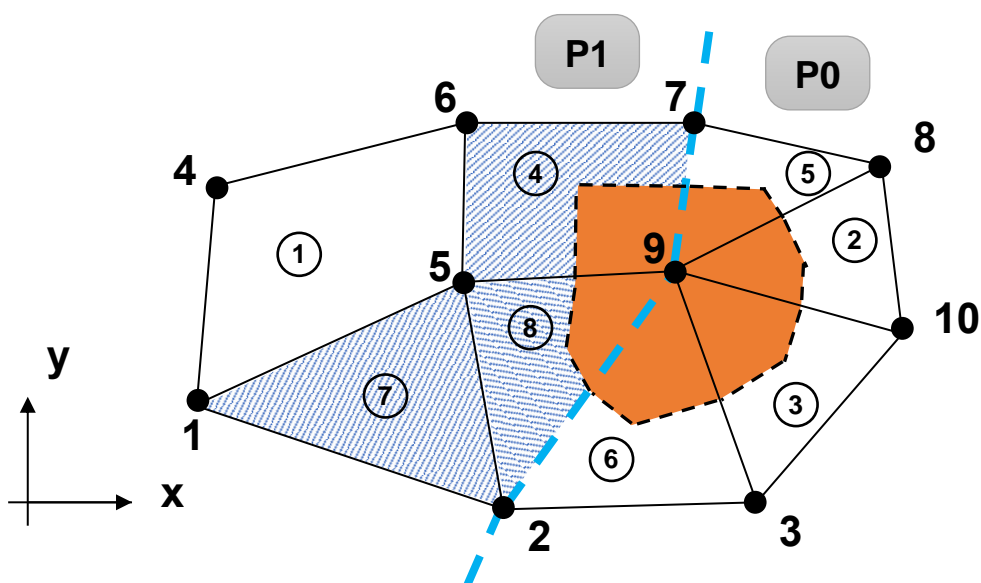
Figura 11 – Malha dividida em dois processos



Fonte: Elaborada pelo autor

Para que o processo 0 possa montar a equação do vértice 9 corretamente, é preciso que o processo 1 envie as informações dos elementos 4 e 8 para o processo 0. Então, os elementos 4 e 8, que pertencem ao processo 1, são copiados para o processo 0, de maneira que passam a compor sua *ghost layer*. Aplicando a mesma ideia ao vértice 2, vê-se que são necessárias as informações dos elementos 7 e 8. Dessa maneira, a *ghost layer* do processo 0 é formada pelos elementos 4, 7 e 8 do processo 1. A Figura 12 mostra a *ghost layer* destacada com o padrão de linhas diagonais.

Figura 12 – Malha com *ghost layer* em destaque



Fonte: Elaborada pelo autor

Como as bibliotecas utilizadas para divisão de malha e gerenciamento de informações não possuem funções próprias para a criação das *ghost layers*, foi desenvolvida uma sub-rotina com tal propósito. Ela utiliza os objetos de malha do FMDB e as funções de passagem de mensagens do MPI para identificar quais elementos precisam ser enviados a quais processos. A ideia por trás do algoritmo segue os seguintes passos:

1. Identificação dos vértices fronteiros;
2. Identificação dos processos donos dos vértices fronteiros;
3. Envio das informações de elemento aos quais esses vértices pertencem, incluindo: conectividades, id's globais dos vértices e coordenadas (x,y,z) dos vértices.

Assim, todos os processos sabem quais são os elementos e vértices próprios e não próprios, e podem identificar os locais corretos onde os cálculos devem ser realizados.

Como mostrado na seção 4.1, as bibliotecas Zoltan e ParMetis se encarregam de dividir a malha de maneira otimizada, reduzindo ao máximo o número de elementos fronteiros e, conseqüentemente, o tamanho das *ghost layers*. Esse é um atributo importante que um simulador deve ter, pois são as informações das *ghost layers* que precisam ser trocadas entre os processos, de maneira que determinado processo saiba o que está acontecendo em um processo vizinho e possa calcular seus gradientes corretamente. Assim, quanto menor o tamanho das *ghost layers*, menor é a quantidade de informações trocadas, e menor é o tempo gasto nessa operação. É importante enfatizar que a rotina de montagem das *ghost layers* não influencia na divisão da malha, mas sim toma como parâmetros de entrada as informações da malha dividida pelas bibliotecas.

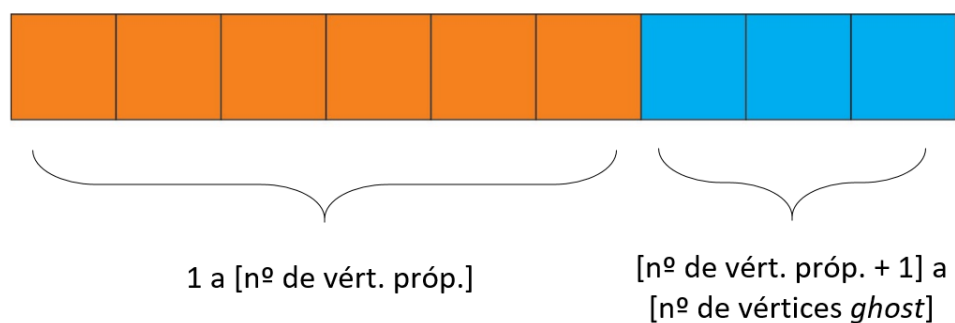
## 4.6 Atributos do código

Esta seção os descreve atributos mais importantes implementados com o objetivo de otimizar a comunicação entre processos, diminuindo o número de chamadas de funções e reduzindo tempos de busca e de troca de informações.

### 4.6.1 Reordenação de vértices

Após a divisão da malha, é feito um processo de rearranjo dos vértices de maneira que, dentro de um processo, todos os vértices próprios estejam localizados nas primeiras posições do vetor de indexação, e os vértices das *ghost layers* estejam no final. Isso faz com que as rotinas específicas para vértices próprios tenham seus laços executados de maneira direta, sem a necessidade de um teste condicional para verificar a qual processo o vértice pertence. A Figura 13 ilustra o resultado dessa reorganização.

Figura 13 – Organização do vetor de indexação



Fonte: Elaborada pelo autor

Juntamente com a função de rearranjo dos vértices, foi desenvolvida uma função de indexação local. Essa função cria vetores auxiliares que permitem que se obtenha os índices local (dentro de um processo) e global (baseado na malha original) de cada vértice. A criação desses vetores auxiliares é feita através de uma busca direta dos vértices locais na lista de vértices globais. Isso evita que funções de busca sejam executadas várias vezes ao longo da simulação, pois como a posição dos vértices dentro do um processo é constante, esse procedimento só é realizado uma única vez.

#### 4.6.2 Indexação de elementos

Uma limitação das bibliotecas de divisão de malha utilizadas neste trabalho é que elas não geram a informação de índice global do elemento. Essa informação é importante quando se trabalha com campos heterogêneos de porosidade e permeabilidade, pois no EbFVM esses dados são armazenados nos elementos, diferentemente da pressão e das propriedades de fluidos, as quais são armazenadas nos vértices. A informação de índice global do elemento possibilita que as porosidades e permeabilidades obtidas do arquivo de entrada sejam distribuídas corretamente entre os processos. Por essa razão, foi desenvolvida uma rotina de indexação global dos elementos, com o intuito de suprir a limitação das bibliotecas auxiliares. Essa rotina faz a indexação baseada na malha original e segue os seguintes passos:

1. Leitura da malha original;
2. Correspondência da conectividade de um elemento com os valores do FMDB;
3. Correspondência das novas conectividades na malha original;
4. Indexação dos elementos.

Ao final da rotina, cada processo sabe quais elementos pertencem a ele, baseado na indexação da malha original. Como os elementos de um processo são constantes ao longo da simulação, essa indexação é feita apenas uma vez.

#### 4.6.3 Topologia virtual de processos

É sabido que o MPI tem suporte para a criação automática de topologias virtuais de processos quando as malhas são cartesianas, e que com uma pequena sequência de chamadas de funções pode-se determinar a divisão e organização virtual dos processos e quantos e quais são os processos vizinhos. Mais informações sobre topologias virtuais e funções do MPI relacionadas a isso podem ser encontradas em Gropp *et al.* (1999). Contudo, essas funções não existem para malhas não estruturadas. Por isso, após a divisão da malha e criação das *ghost layers*, foi construída uma rotina que realizasse esse trabalho de construção da topologia.

Essa rotina determina o número de vértices, a identidade dos vértices e para quais processos esses vértices devem ser enviados (ou de quais processos devem ser recebidos). Para isso, ela toma como base o processo dono de cada vértice presente nas *ghost layers* e segue os passos baixo:

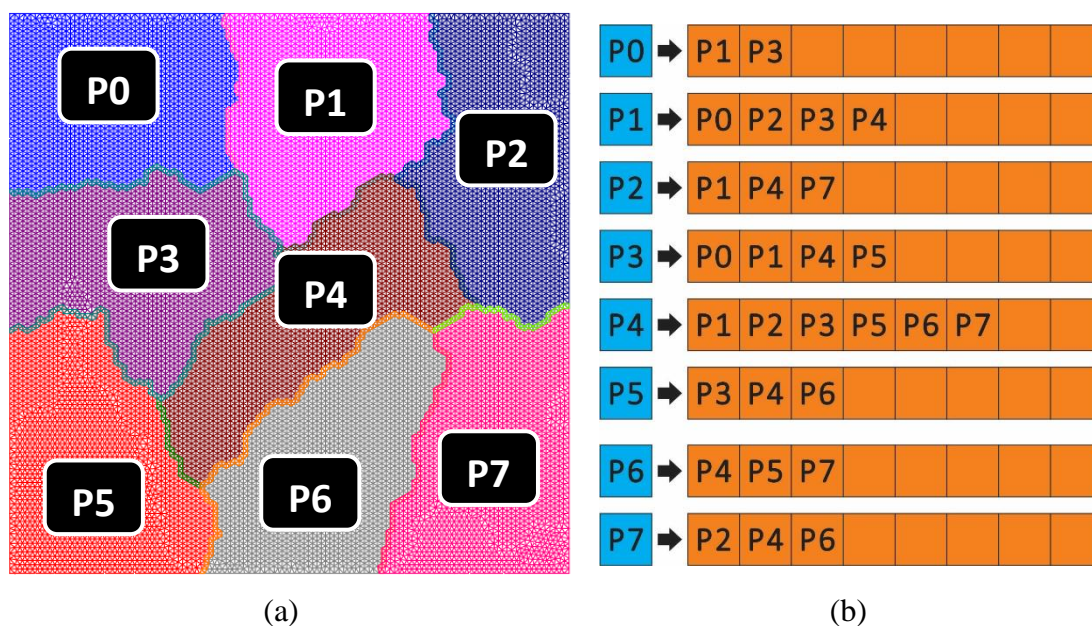
1. Identificação dos processos donos dos vértices das *ghost layers*;
2. Identificação das id's globais dos vértices das *ghost layers*;
3. Envio da lista de vértices para o processo dono;
4. Recebimento da lista de vértices dos processos vizinhos.

Ao final da rotina, cada processo sabe quem são seus vizinhos, ou seja, com quem deve se comunicar.

Isso possibilita também que já seja estabelecido o tamanho máximo das mensagens trocadas; tamanho igual ao número de vértices da maior *ghost layer*. Com isso, os vetores utilizados para troca de informações entre processos são alocados todos de uma vez no início da execução do programa. Como essas informações são constantes ao longo da simulação, basta que a topologia seja criada uma única vez. Assim, cada processo tem a informação de com qual processo vai se comunicar e quais vértices estão envolvidos nessa comunicação. A Figura 14a exemplifica uma malha dividida em 8 processos, e a Figura 14b é uma representação visual da topologia de comunicação após sua construção.



Figura 14 – Topologia para uma malha com 8 processos



Fonte: Elaborada pelo autor

#### 4.6.4 Troca de informações

Um dos fatores que degrada o desempenho de um programa paralelo é a quantidade excessiva de troca de informações entre os processos, principalmente quando se utilizam funções síncronas do MPI. Em cada par de chamada `MPI_Send/MPI_Recv`, é necessário que o MPI aloque a memória necessária à comunicação e sincronize os processos, ou seja, faça com que um processo espere o outro para que a comunicação ocorra. Dessa maneira, um número muito grande de chamadas de funções de troca de mensagens degrada o desempenho na medida em que é gasto tempo na sincronização e na latência natural da operação, ou seja, o tempo gasto para construir o canal de comunicação entre os processos envolvidos.

Buscando diminuir a influência desses problemas dentro do código, foram abordadas duas alternativas. Em primeiro lugar, só existem quatro pontos de troca de informações, e eles são suficientes para que todas as informações necessárias à realização dos cálculos sejam passadas entre os processos. Em segundo lugar, três dos quatro pontos de troca utilizam comunicação assíncrona, ou seja, um processo pode enviar a informação e continuar executando outras operações, sem ter que esperar que o outro processo tenha recebido a informação. Da mesma maneira, um processo pode preparar o espaço de memória para receber uma informação, executar outras operações e utilizar a informação quando ela tiver chegado. Essas operações assíncronas são realizadas através do par de

funções MPI\_Isend/MPI\_Irecv. O outro ponto que continua síncrono é onde as pressões calculadas são redistribuídas entre os processos. Ele foi mantido assim por questão de simplicidade do código, visto que é gasto um tempo irrisório para completar a comunicação, dada a natureza da informação envolvida.

#### 4.7 Speedup

Existem várias maneiras de se medir a eficiência da paralelização de um programa, contudo a maneira mais comum é através do cálculo do *speedup*. Ele é avaliado como sendo a razão entre o tempo de CPU em serial em relação ao tempo de CPU em paralelo. E pode ser entendido como sendo uma medida de quantas vezes mais rápida é a execução de uma rodagem em paralelo em relação à serial:

$$Speedup = \frac{Tempo\ de\ CPU_{(serial)}}{Tempo\ de\ CPU_{(paralelo)}}. \quad (4.1)$$

Suponha que determinada simulação é executada em 3600 segundos em serial. Quando com quatro processos, espera-se que o tempo de CPU seja em torno de 900 segundos. Isso representaria um *speedup* de 4,00, ou seja, a simulação em paralelo com quatro processos seria quatro vezes mais rápida que a simulação em serial. Infelizmente, isso só acontece em um cenário ideal. Na prática, existe um tempo de comunicação associado à paralelização com memória distribuída. Esse tempo inclui a chamada da função de transferência, preparo, envio e recebimento da mensagem entre os processos envolvidos. É muito difícil que esse *overhead* seja zerado devido a questões de *hardware*, entretanto ele pode ser bastante diminuído. Para tanto, é necessário que o código seja bem estruturado, que a quantidade de chamadas de funções de transferência seja diminuída para o mínimo necessário, e que se utilize comunicação assíncrona e/ou coletiva quando possível.

De uma maneira geral, pode-se afirmar que quanto maior o número de processos, maior é o tempo gasto com comunicação, e o *speedup* tende a diminuir. É preciso que se faça uma análise da curva de *speedup* para ver até quando é vantajoso aumentar o número de processos, pois haverá um ponto onde esse aumento acarretará num tempo de simulação maior, mesmo em relação à simulação em serial.

Não há como saber com precisão qual será o *speedup* de um programa antes que ele seja executado. No caso da simulação de reservatórios de petróleo, vários fatores como tamanho da malha, número de componentes, e *solver* influenciam no tempo de

simulação e, conseqüentemente, no *speedup*. Um ajuste dos parâmetros de simulação é essencial para que um caso tenha o melhor desempenho possível.

## 5 RESULTADOS

Nesta seção serão apresentados os resultados obtidos para diversas situações. Serão mostrados resultados para malhas 2D e 3D, com diferentes tipos e números de elementos, além de casos com diferentes números de componentes e poços. Os resultados são avaliados em termos das curvas de produção e dos tempos de simulação, onde são apresentados os *speedups* de cada caso.

As malhas 2D foram geradas com o *software* GiD<sup>®</sup>, e as malhas 3D com o Ansys<sup>®</sup> ICEM. Esses geradores de malha fornecem um arquivo de saída segundo a formatação ASCII, o qual é lido pelo simulador UTCOMP.

### 5.1 Casos 2D

A seguir, serão apresentados os resultados obtidos para casos bidimensionais em diversas configurações de poços, e tipos de malhas diferentes.

#### 5.1.1 Caso 1

Esse caso consiste na injeção de gás em um reservatório inicialmente bifásico, com presença de água e óleo, sendo que a água permanece imóvel ao longo da simulação. À medida em que o CO<sub>2</sub> é injetado, a fase gasosa aparece, e o reservatório torna-se trifásico. Os poços estão dispostos em ¼ de *five-spot*, sendo um produtor e outro injetor. As Tabelas 1 a 3 mostram as propriedades do reservatório e dos poços.

Tabela 1 – Propriedades de reservatório do caso 1

Propriedade	Valor
Comprimento, largura e espessura	170,7 m; 170,7 m e 30,48 m
Porosidade	0,30
Saturação inicial de água	0,25
Pressão inicial	20,68 MPa
Permeabilidade em X e Y	10 <sup>-13</sup> e 10 <sup>-13</sup> m <sup>2</sup>
Temperatura da formação	299,82 K

Tabela 2 – Concentração de componentes do caso 1

Componente	Composição inicial do reservatório	Composição do fluido de injeção
CO <sub>2</sub>	0,010	0,950
C <sub>1</sub>	0,190	0,050
nC <sub>16</sub>	0,800	-

Tabela 3 – Restrição de poços do caso 1

Poço	Tipo de operação	Valor
Injetor	Vazão constante	283.170 m <sup>3</sup> /dia
Produtor	Pressão constante	20,68 MPa

A malha usada nesse caso tem o formato quadrado, com 14540 vértices e 28630 elementos triangulares. A malha é mostrada na Figura 15, onde o ponto azul é o poço injetor, e o ponto vermelho é o poço produtor. A Figura 16 mostra a malha dividida em 2, 4 e 8 processos.

Figura 15 – Malha original do caso 1

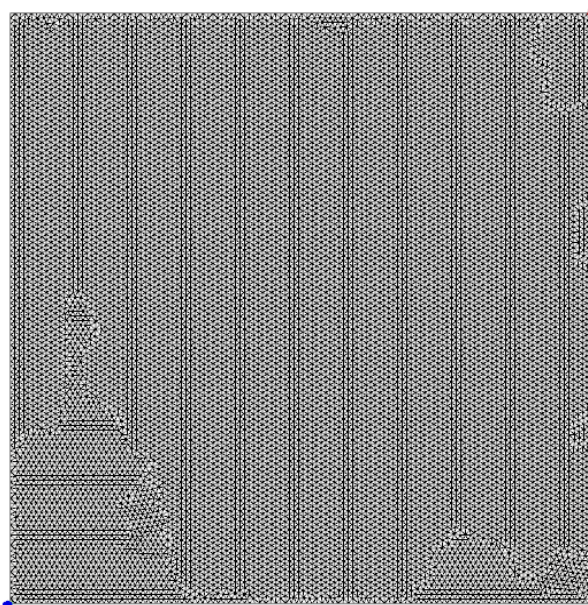
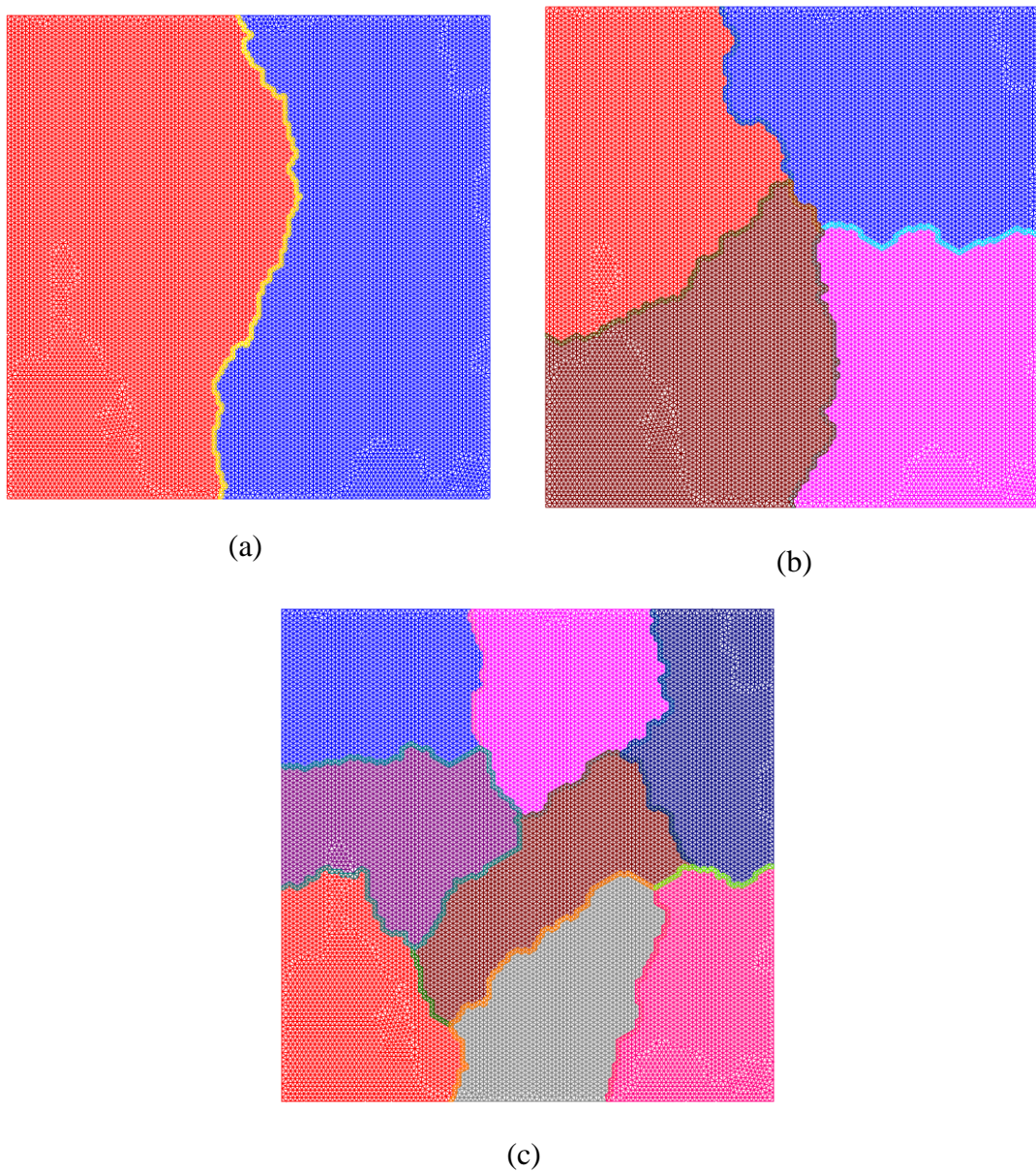
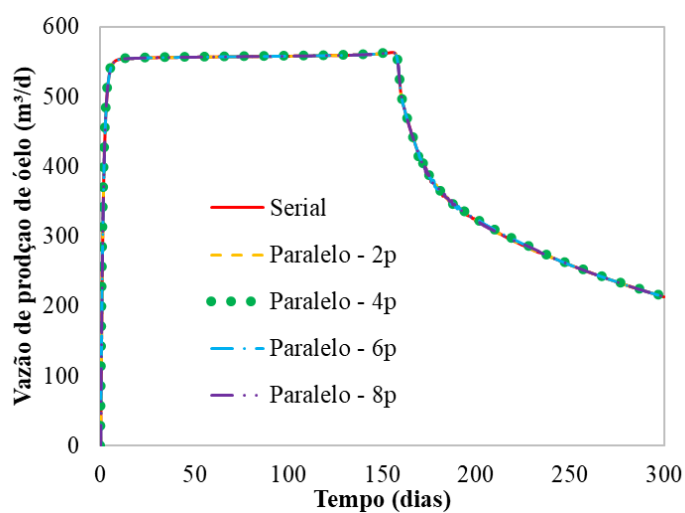


Figura 16 – Malha do caso 1 dividida em: (a) 2, (b) 4 e (c) 8 processos

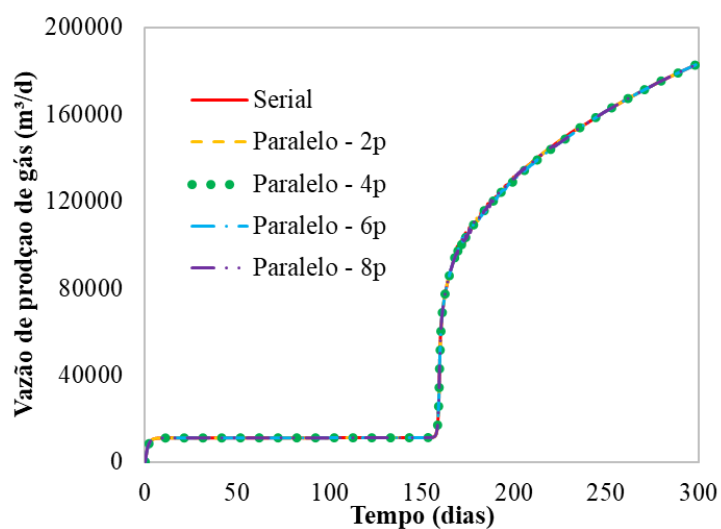


A Figura 17 apresenta as curvas de produção de óleo e de gás. O intuito é comprovar que, independentemente do número de processos, a resposta é a mesma. É importante salientar que essa concordância das curvas mostra que a física do problema é respeitada com qualquer número de processos. Esse fato é comprovado em todos os casos apresentados neste trabalho, tanto 2D quanto 3D.

Figura 17 – Curvas de produção de (a) óleo e (b) gás do caso 1

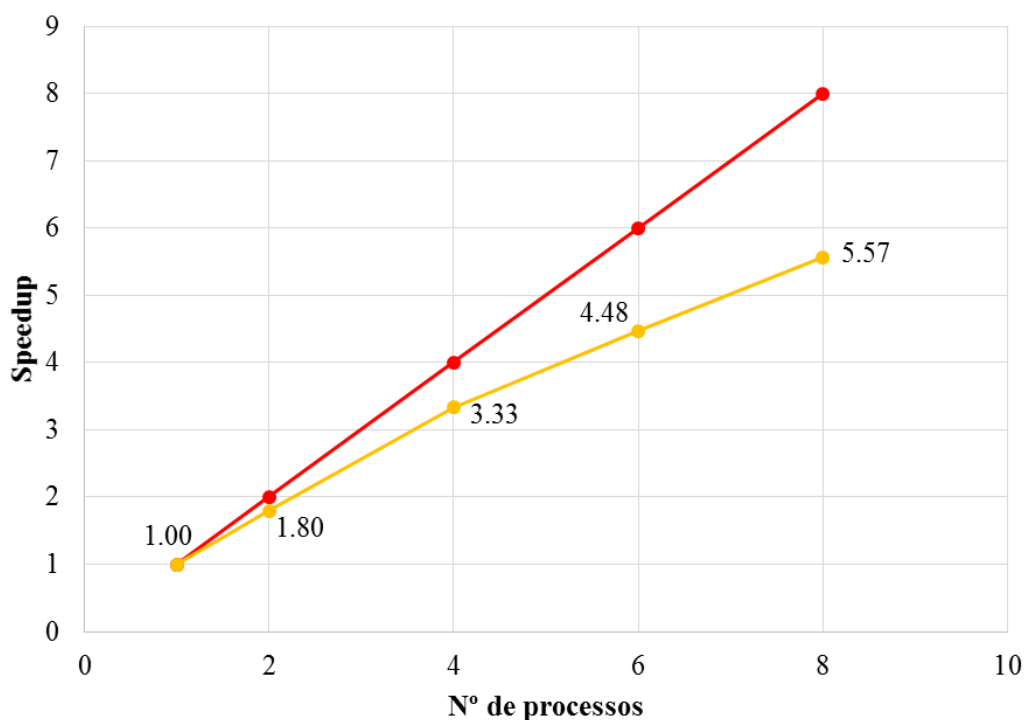


(a)



(b)

A curva de *speedup* é mostrada na Figura 18. Nela, é possível observar um crescimento do *speedup* com o aumento do número de processos, como esperado. O distanciamento da curva real para a ideal deve-se ao aumento dos custos de comunicação à medida que se aumenta o número de processos.

Figura 18 – Curva de *speedup* do caso 1

### 5.1.2 Caso 2

O segundo caso consiste em um processo de injeção de gás num reservatório irregular. O fluido é caracterizado por seis componentes, e o reservatório já possui três fases desde o início da simulação (água, óleo e gás), sendo que a água permanece imóvel durante o processo. Esse caso possui oito poços, sendo dois poços injetores e seis poços produtores. Todos os produtores operam da mesma maneira do início ao fim da simulação, assim como os injetores. As Tabelas 4 a 6 mostram as características do reservatório, as concentrações molares dos fluidos e as restrições de operação dos poços.

Tabela 4 – Propriedades de reservatório do caso 2

Propriedade	Valor
Espessura	30,48 m
Porosidade	0,35
Saturação inicial de água	0,17
Pressão inicial	10,34 MPa
Permeabilidade em X e Y	$10^{-13}$ e $10^{-13}$ m <sup>2</sup>
Temperatura da formação	344,26 K



Tabela 5 – Concentração dos componentes do caso 2

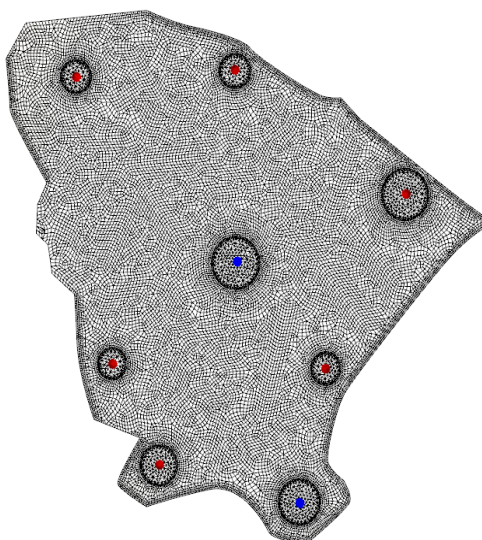
Componente	Composição inicial do reservatório	Composição do fluido de injeção
C <sub>1</sub>	0,500	0,770
C <sub>3</sub>	0,030	0,200
C <sub>6</sub>	0,070	0,010
C <sub>10</sub>	0,200	0,010
C <sub>15</sub>	0,150	0,005
C <sub>20</sub>	0,050	0,005

Tabela 6 – Restrição de poços do caso 2

Poço	Tipo de operação	Valor
Injetor	Vazão constante	$2,83 \times 10^5$ m <sup>3</sup> /dia
Produtor	Pressão constante	10,34 MPa

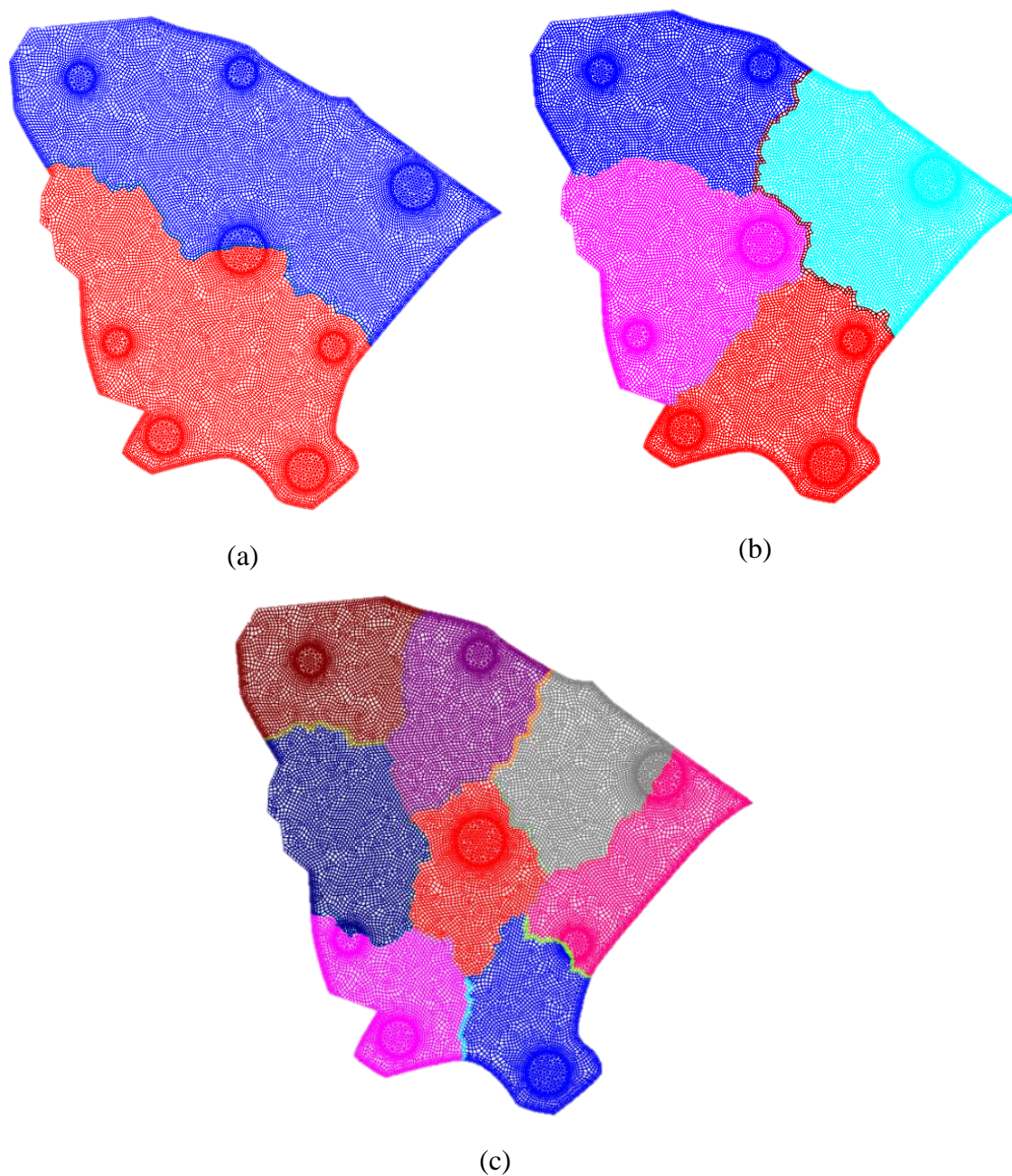
O reservatório obviamente é fictício e possui um formato parecido com o do estado do Ceará. Isso tem o propósito de mostrar como o EbFVM pode mapear com precisão uma geometria irregular. A Figura 19 mostra o reservatório original com a localização dos poços, sendo os pontos azuis os poços injetores e os pontos vermelhos os produtores. Essa malha possui 20205 vértices e 21372 elementos, e, desses elementos, 3254 são triângulos e 18118 são quadriláteros.

Figura 19 – Malha original do caso 2



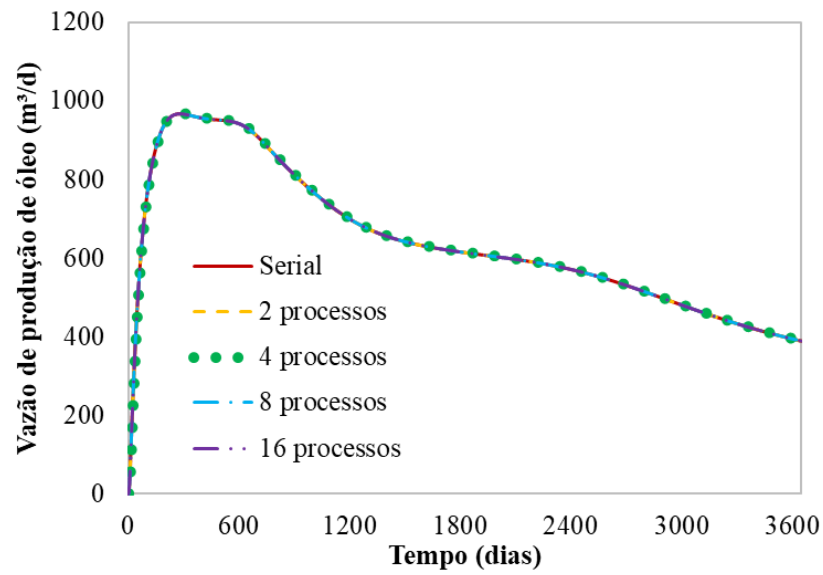
A Figura 20 mostra a malha dividida em 2, 4 e 8 processos.

Figura 20 – Malha do caso 2 dividida em: (a) 2, (b) 4 e (c) 8 processos

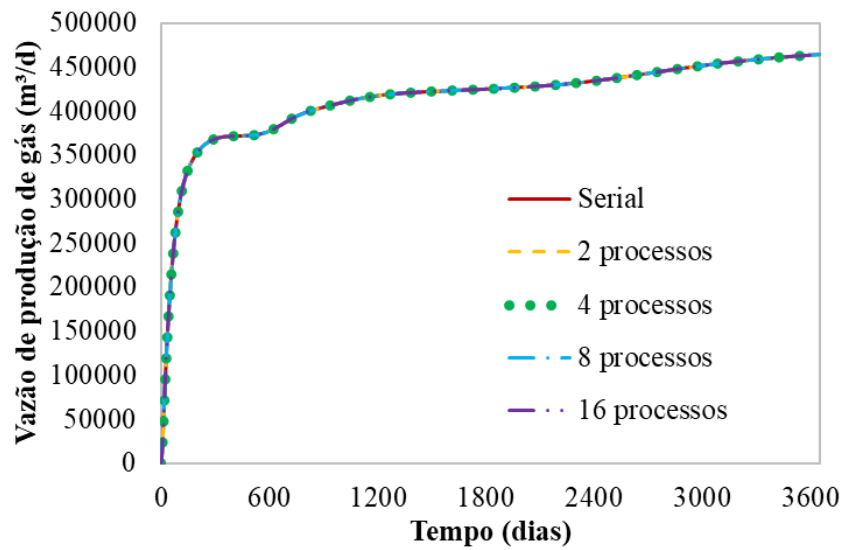


As curvas de produção de óleo e de gás são mostradas na Figura 21. As curvas sobrepostas confirmam que a resposta não depende do número de processos, como esperado.

Figura 21 – Curvas de produção de (a) óleo e de (b) gás do caso 2



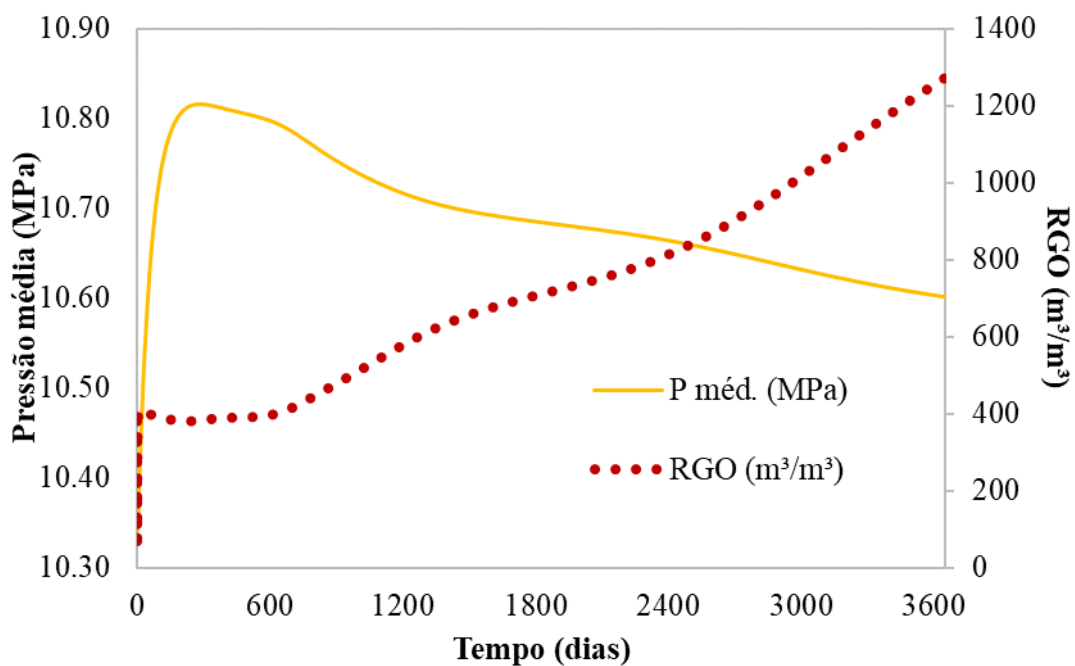
(a)



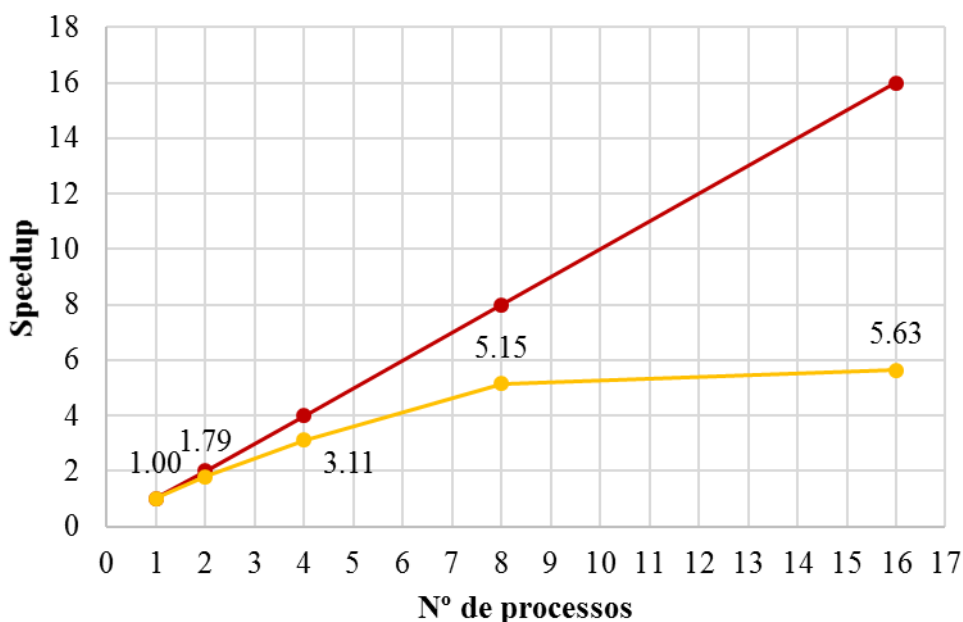
(b)

A Figura 22 mostra as curvas de RGO e Pressão média do reservatório.

Figura 22 – Curvas de RGO e P méd. do caso 2



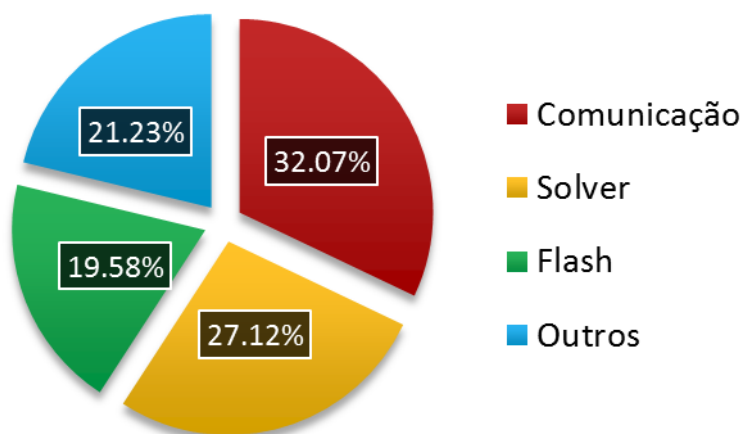
A curva de *speedup* é mostrada na Figura 23.

Figura 23 – Curva de *speedup* do caso 2

O comportamento assintótico do *speedup* obtido é perceptível a partir de 4 processos. Isso se deve ao aumento dos custos de comunicação devido ao aumento do número de processos. Uma análise dos tempos de comunicação com 16 processos é

mostrada na Figura 24. Foi observado que 32% do tempo de simulação foi gasto com a troca de informações entre os processos, e o *solver* foi responsável por 27% do tempo de simulação, onde o tempo total de simulação foi de 705 segundos.

Figura 24 – Custos de comunicação do caso 2 com 16 processos



Na prática, existe um ponto onde os custos de comunicação se tornam muito elevados, e aumentar o número de processos passa ser uma desvantagem. Essa análise mostra que é necessário avaliar-se até quando é vantajoso aumentar o número de processos em uma determinada simulação.

### 5.1.3 Caso 3

No caso 3, o reservatório contém as fases água, óleo e gás desde o início. Ele é mais complexo que os dois anteriores pois é um reservatório com porosidade e permeabilidade heterogêneas isotrópicas. Além disso, há um total de 60 poços, sendo 29 produtores e 31 injetores, e estes operam numa agenda WAG (*Water Alternating Gas*). As propriedades de reservatório, concentração de fluidos e operação dos poços são listadas nas Tabelas 7 a 9. Devido à complexidade do cronograma de operação dos poços, não é viável mostrá-lo neste trabalho, todavia as restrições de operação são mostradas na Tabela 9.

Tabela 7 – Propriedades de reservatório do caso 3

Propriedade	Valor
Espessura	30 m
Porosidade	0,008 – 0,15
Saturação inicial de água	0,25
Pressão inicial	50 Mpa
Permeabilidade em X e Y (isotrópico)	$0,85 \times 10^{-15}$ a $390 \times 10^{-15}$ m <sup>2</sup>
Temperatura da formação	353,15 K

Tabela 8 – Concentração de componentes do caso 3

Componente	Composição inicial do reservatório	Composição do fluido de injeção
PC1	0,10	0,950
PC2	0,50	0,050
PC3	0,12	-
PC4	0,06	-
PC5	0,07	-
PC6	0,06	-
PC7	0,06	-
PC8	0,03	-

Tabela 9 – Restrição de poços do caso 3

Poço	Tipo de operação	Valor
Injetor de gás	Vazão constante	$1,98 \times 10^5$ m <sup>3</sup> /dia
Injetor de água	Vazão constante	794,94 m <sup>3</sup> /dia
Produtor	Pressão constante	47,57 MPa

O reservatório empregado nesse caso é mostrado na Figura 25, onde as dimensões são mostradas em pés (ft). A malha possui 29241 vértices e 28900 quadriláteros. As permeabilidades nas direções X e Y são iguais e estão representadas na Figura 26a, enquanto a porosidade é representada na Figura 26b.

Figura 25 – Malha original do caso 3

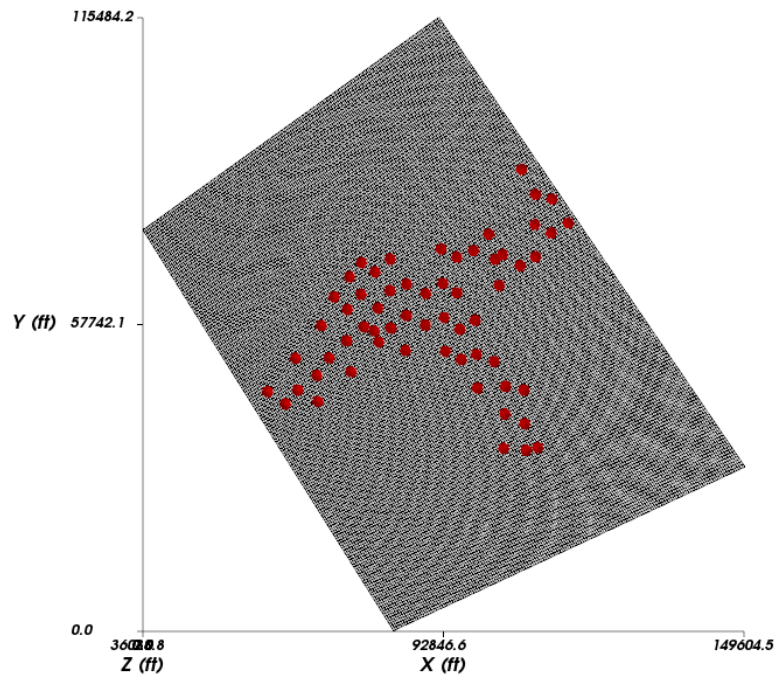
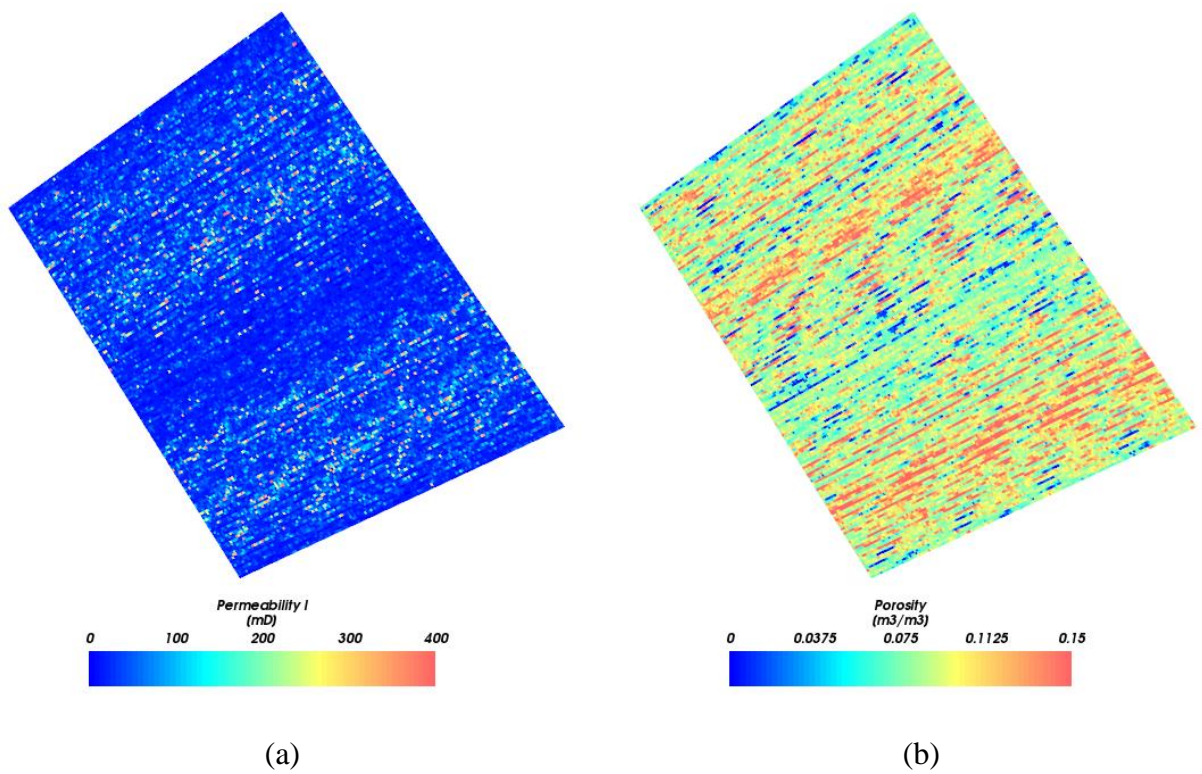
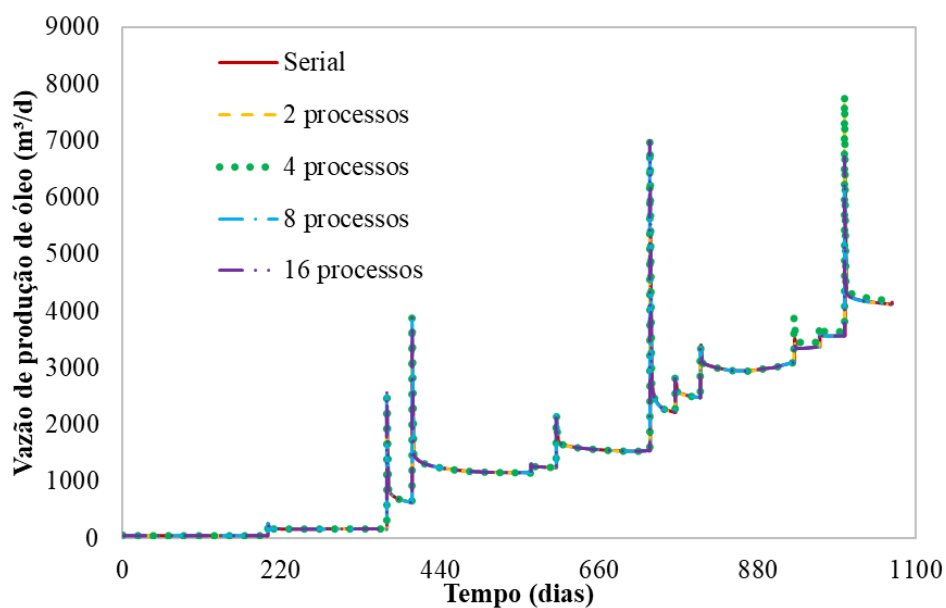


Figura 26 – Campos de (a) permeabilidade XY e (b) porosidade do caso 3

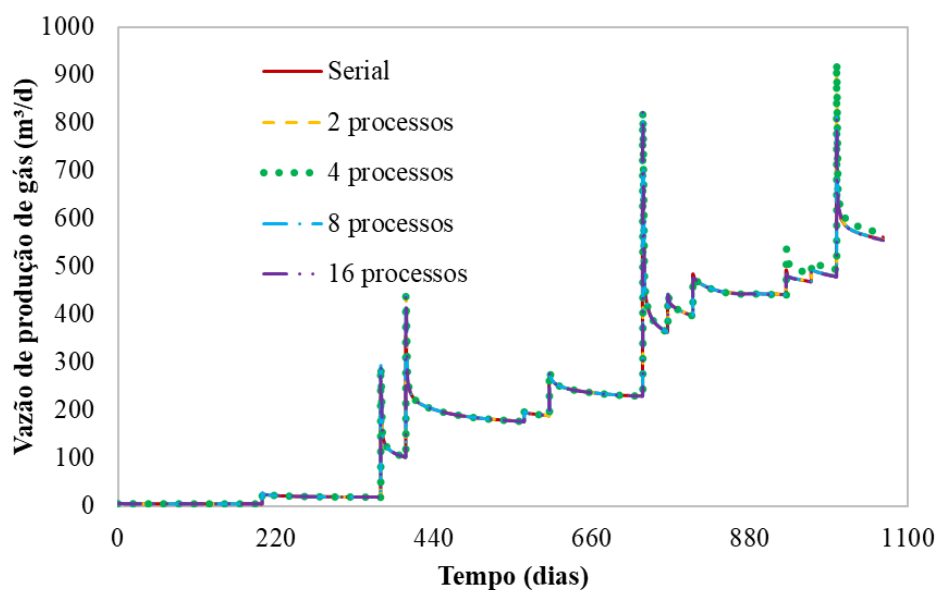


As curvas de produção de óleo e de gás são apresentadas na Figura 27. Os picos denotam ou um início de ciclo de injeção de gás ou a abertura de um poço produtor. Mais uma vez salienta-se a concordância das curvas de produção. Esse caso é heterogêneo na permeabilidade e na porosidade e tem uma agenda de operação de poços bem mais complexa em relação aos casos 1 e 2. Ainda assim, a física do problema é respeitada e é independente do número de processos.

Figura 27 – Curvas de produção de (a) óleo e de (b) gás do caso 3



(a)

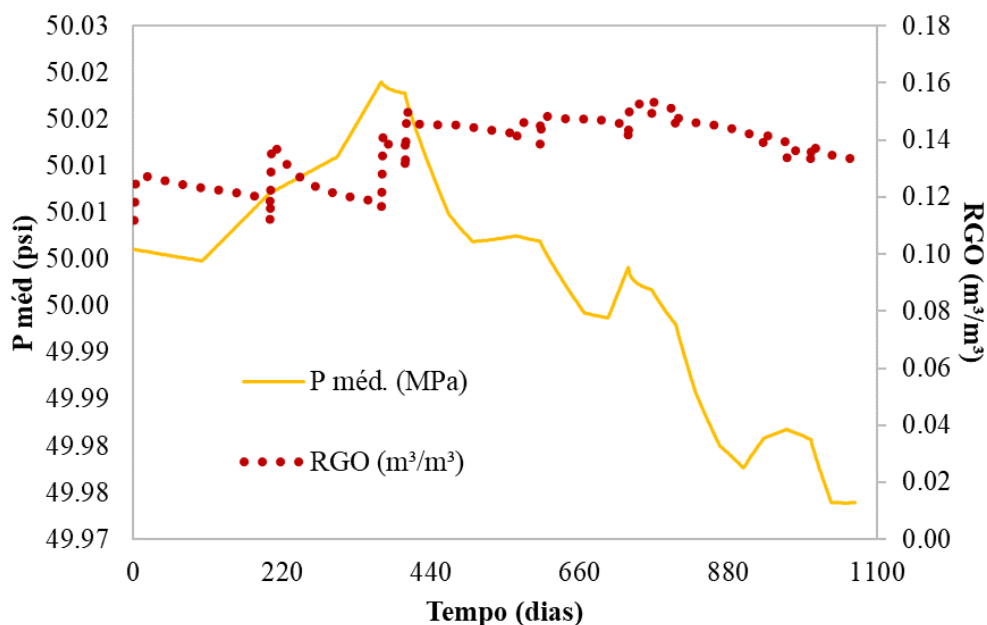


(b)

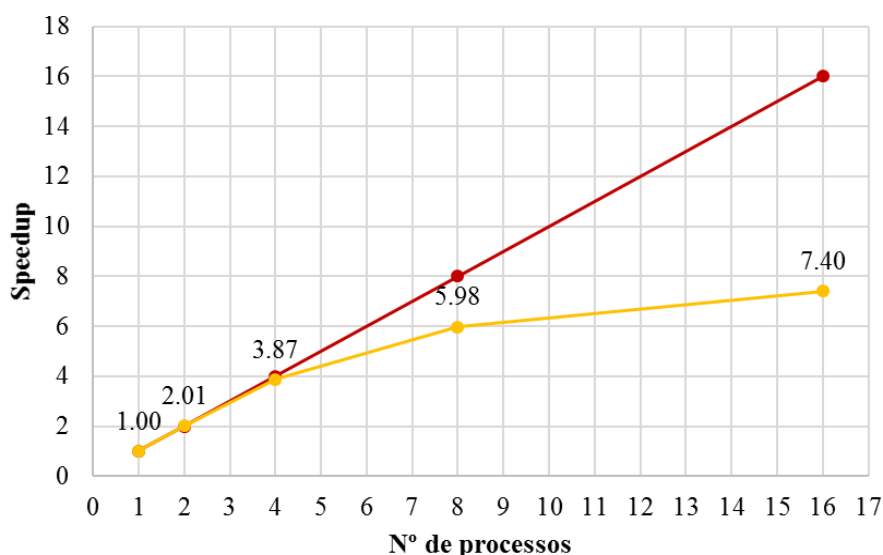
A Figura 28 mostra as curvas de RGO e Pressão média do reservatório.



Figura 28 – Curvas de RGO e P méd. do caso 3



A curva de *speedup* é mostrada na Figura 29. Dela é possível observar que o *speedup* para 2 e 4 processos foi linear, o que representa um resultado excelente. Apesar dos bons resultados com 2 e 4 processos, a inclinação da curva real foi reduzida com 8 e 16 processos. Esse comportamento assintótico é devido ao aumento dos custos de comunicação, decorrente do aumento do número de processos.

Figura 29 – Curva de *speedup* do caso 3

Apesar da distância em relação ao *speedup* linear, os valores apresentados nesse caso são maiores que os dos casos 1 e 2, mesmo que o número de vértices das malhas dos três casos seja parecido. Isso corrobora com o que foi dito na seção 4.7, que

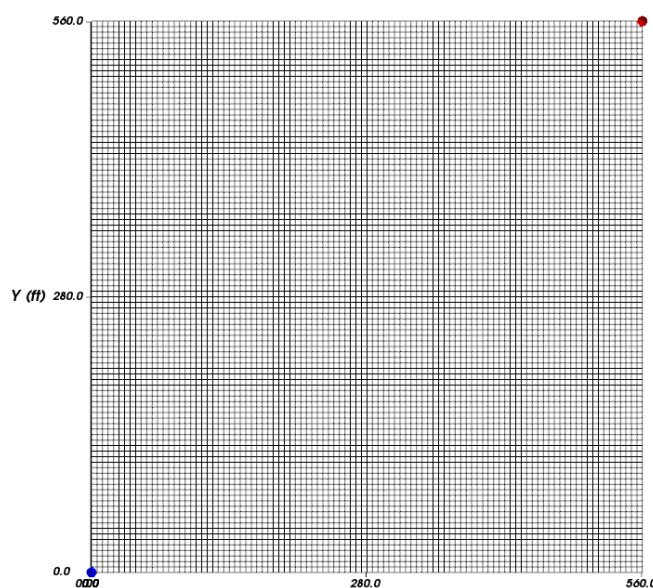
cada caso possui sua curva própria de *speedup*, influenciada por fatores como número de poços e de componentes, características de rocha e de fluidos, etc.

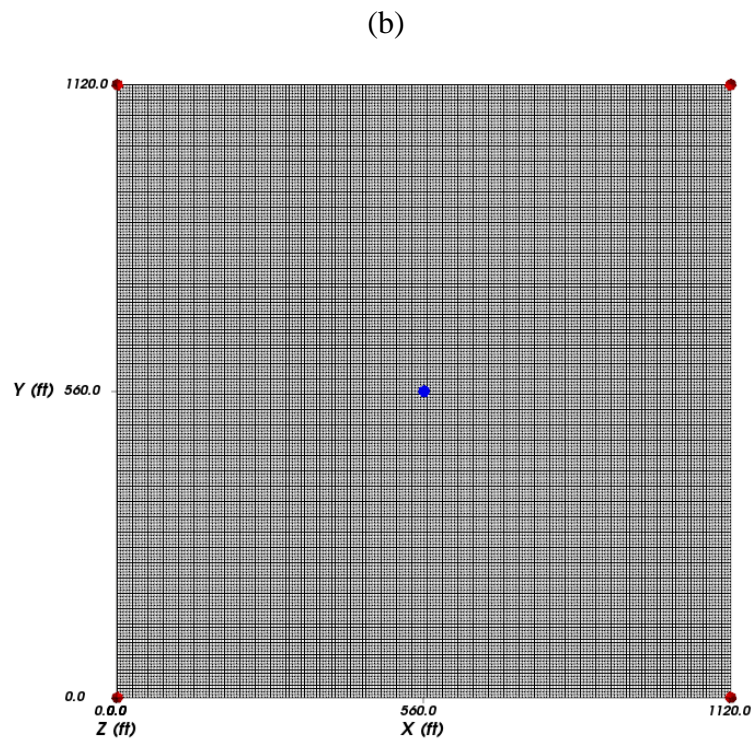
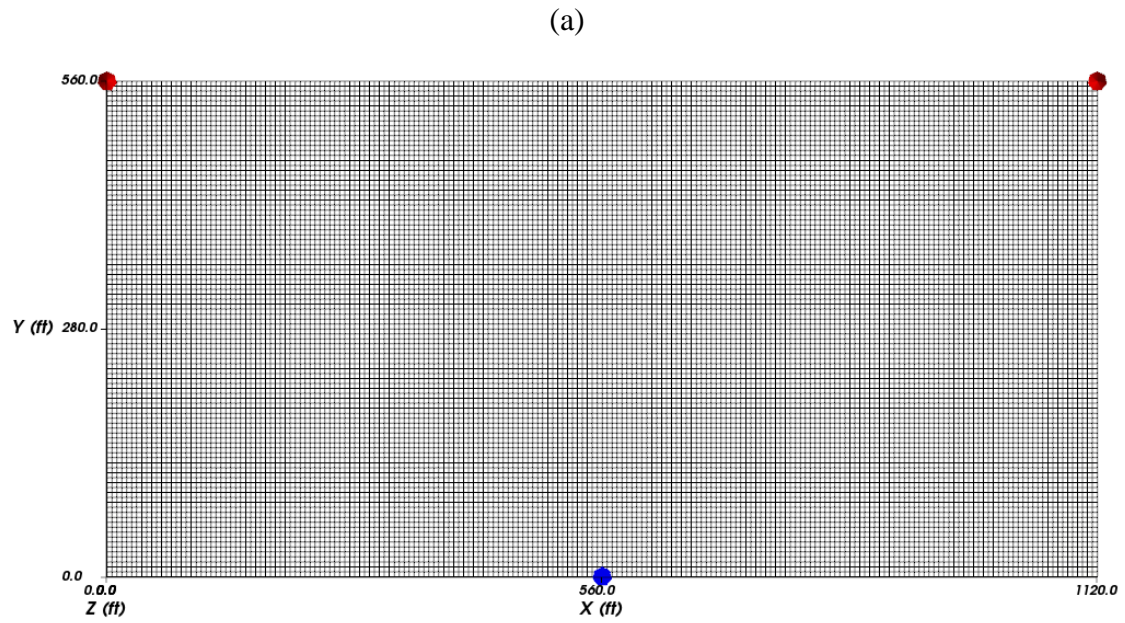
#### 5.1.4 Caso 4

Este caso possui exatamente as mesmas propriedades de reservatório e de poços que o caso 1 (ver seção 5.1.1), além dos mesmos componentes. A diferença reside nas malhas testadas. O caso 4 é caracterizado pelas mesmas condições de operação aplicadas a três malhas diferentes. Apesar de as malhas serem regulares, elas são tratadas pelo EbFVM, assim como as outras apresentadas. Este caso tem por objetivo a investigação da influência do aumento do número de vértices nos resultados.

As malhas testadas são mostradas na Figura 30. A Figura 30a representa uma configuração de  $\frac{1}{4}$  de *five-spot* com 10201 vértices, a Figura 30b uma de  $\frac{1}{2}$  de *five-spot* com 20301 vértices e a Figura 30c uma configuração de um *five-spot* completo com 40401 vértices. Foram escolhidas essas configurações de poços para que a malha pudesse ser aumentada sem que a fluidodinâmica do caso fosse muito alterada. À medida que a malha dobrou de número de vértices, a vazão de injeção também foi dobrada, mas as pressões de produção permaneceram iguais, de modo que o  $\Delta P$  entre os produtores e o injetor se manteve o mesmo com as três malhas. As vazões de injeção de gás foram de  $2,83 \times 10^5$ ;  $5,66 \times 10^5$  e  $11,32 \times 10^5$  m<sup>3</sup>/dia para cada malha respectivamente. E as pressões de produção se mantiveram constantes em 20,68 MPa.

Figura 30 – Malhas do caso 4: (a) 10201 vértices, (b) 20301 vértices e (c) 40401 vértices

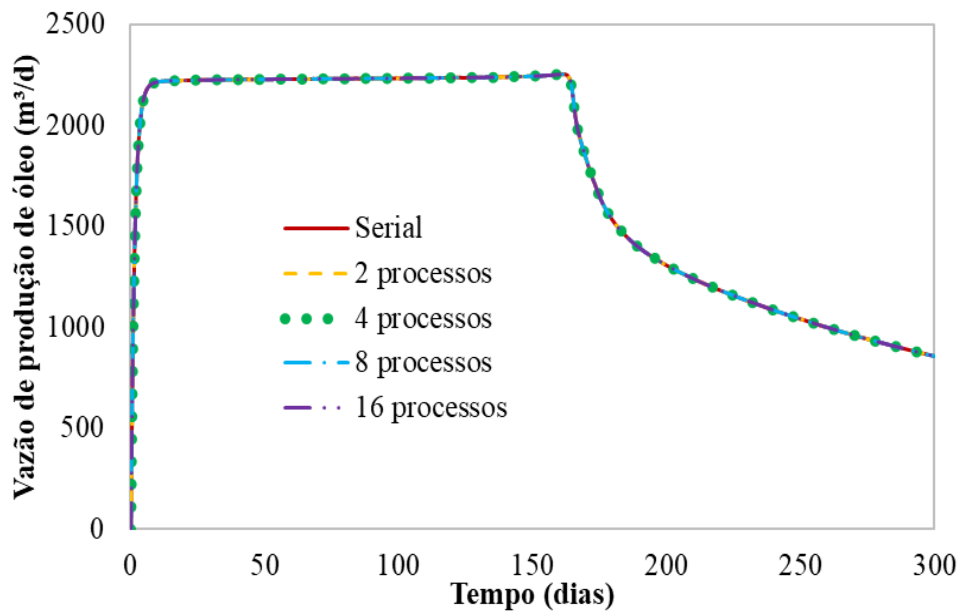




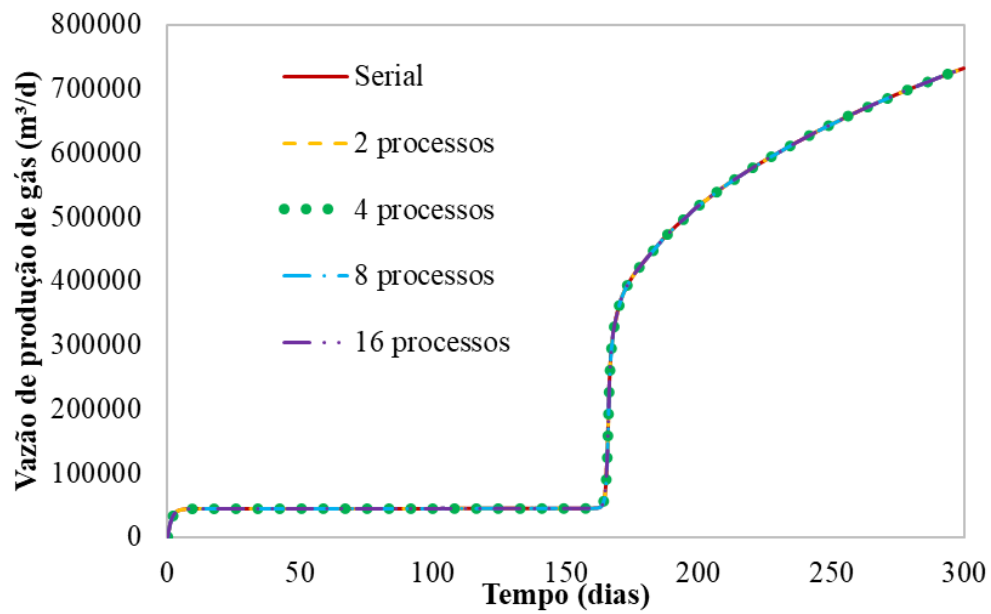
(c)

As curvas de produção de óleo e de gás são mostradas na Figura 31. Elas são o resultado da simulação com a malha (c). As curvas dos outros casos possuem o mesmo formato e foram omitidas para evitar repetitividade.

Figura 31 – Curvas de produção de óleo e de gás do caso 4, malha (c)

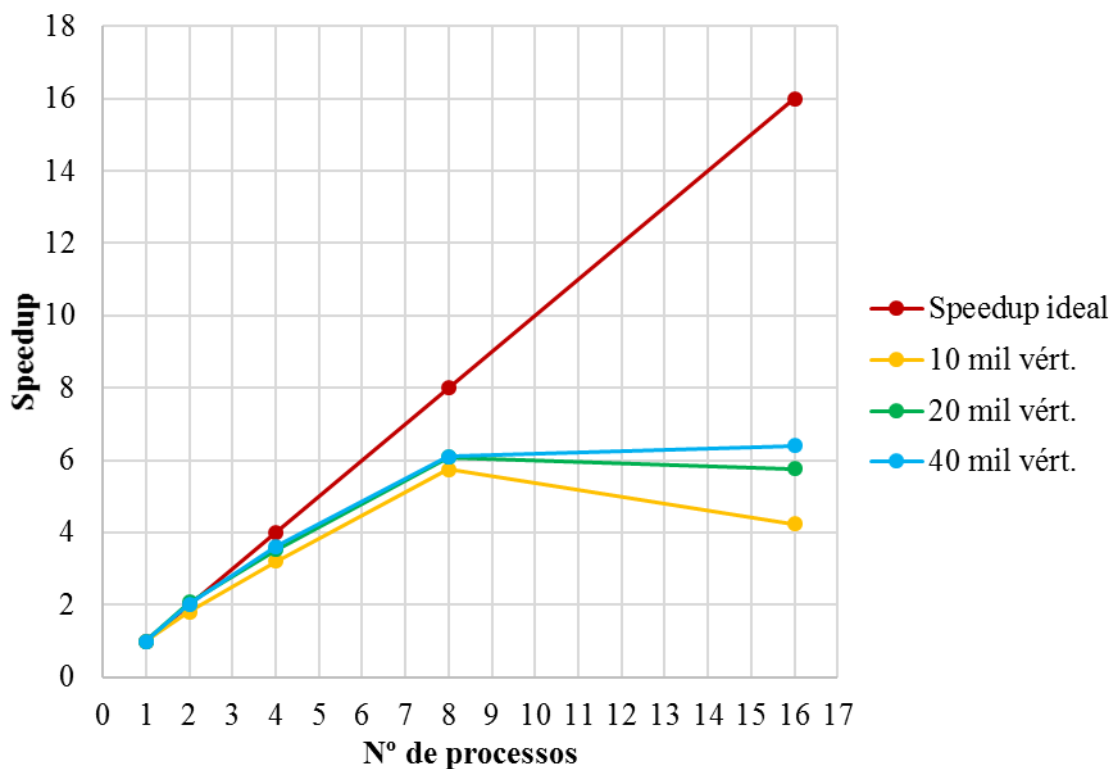


(a)



(b)

A Figura 32 mostra as curvas de *speedup* das três malhas.

Figura 32 – Curvas de *speedup* do caso 4Tabela 10 – Comparativo dos valores de *speedup* do caso 4

	1 processo	2 processos	4 processos	8 processos	16 processos
<b>10 mil vért.</b>	1,00	1,81	3,22 ↓	5,75 ↓	4,24 ↓
<b>20 mil vért.</b>	1,00	2,07	3,52 ↓	6,09 ↓	5,76 ↓
<b>40 mil vért.</b>	1,00	2,02	3,62 ↓	6,11 ↓	6,40 ↓

Apesar de os *speedups* com 16 processos não estarem bons, lembre-se de que o objetivo deste caso é a verificação do aumento do *speedup* com o aumento do número de vértices. Mesmo não estando muito claro na Figura 32, a Tabela 10 mostra que os valores de *speedup* aumentam a medida em que a malha é refinada. Esse resultado é esperado e demonstra o seguinte fato: o aumento do número de vértices contribui para o aumento do *speedup*. Com malhas mais refinadas, cada processo acaba tendo que executar mais cálculos e isso obviamente demanda mais tempo. Dessa maneira, o tempo de transferência de informações entre processos torna-se relativamente menor em relação ao tempo de cálculo, e o *speedup* aumenta. Observe que praticamente todos os parâmetros do caso se mantiveram os mesmos, de maneira que o número de vértices das malhas foi

a única condição alterada. Cada curva de *speedup* é única para cada situação estudada, mas esse resultado mostra que a tendência é que o *speedup* aumente à medida que o número de vértices da malha aumente também.

## 5.2 Casos 3D

A seguir, serão apresentados os resultados obtidos para casos tridimensionais em diversas configurações de poços e tipos de malhas diferentes. Apesar do prisma ter sido mencionado nesse texto (ver seção 3.2) e ter sido implementado no UTCOMP, não serão apresentados resultados envolvendo esse elemento devido à dificuldade de geração de malhas utilizando esse elemento de transição.

### 5.2.1 Caso 5

O caso em questão possui as mesmas propriedades de fluido e de reservatório que o caso 2 (ver seção 5.1.2). Entretanto, os poços injetores operam a uma vazão constante de 28.317 m<sup>3</sup>/dia, e os produtores a uma pressão constante de 8,96 MPa. O reservatório é bastante irregular, e possui uma região modelada como um buraco na malha. A maioria dos simuladores comerciais trabalha com o conceito de células inativas, ou blocos inativos, e modela essa região para, em seguida, excluir dos cálculos os elementos contidos nela. Essa ferramenta de células inativas tem como função a modelagem geométrica das superfícies externas do reservatório. Contudo, mesmo que os blocos inativos sejam retirados da simulação, ainda é necessário que se construa o arquivo de malha com todos os blocos, e que haja outro arquivo informando quais são os ativos e quais são os inativos.

Outra vantagem do EbFVM é que com ele essa região de blocos inativos não precisa ser modelada geometricamente, nem excluída. Basta gerar a malha para as regiões do reservatório que serão simuladas de fato.

Esse reservatório possui sete poços, 4 injetores (topo azul) e três produtores (topo vermelho), e todos operam em todas as camadas em Z. A Figura 33 mostra a malha original com o posicionamento dos poços. Ela possui 41.392 vértices e 36.975 hexaedros. A divisão da malha em 2, 4 e 8 processos é mostrada na Figura 34.

Figura 33 – Malha original do caso 5

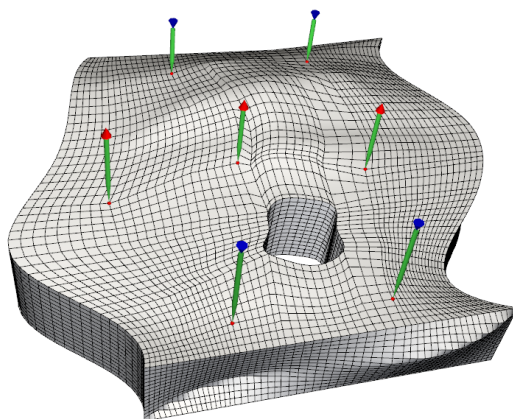
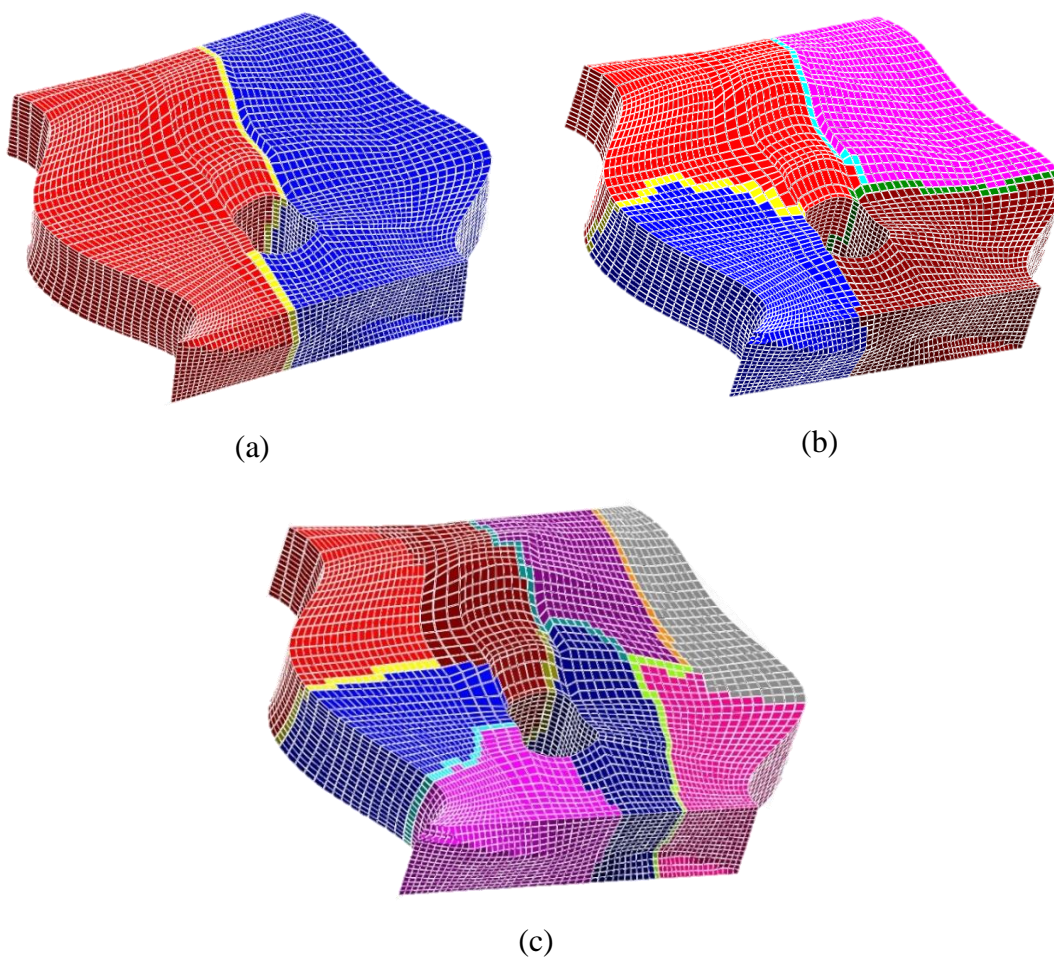
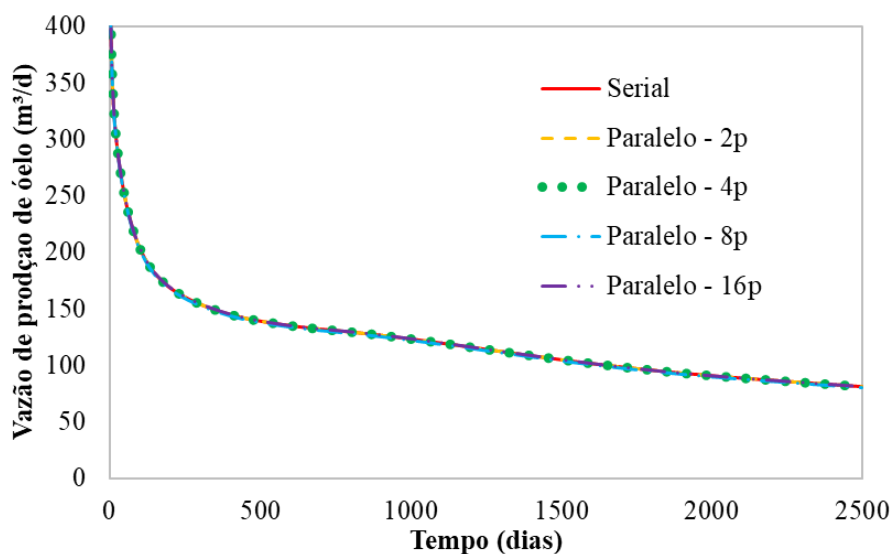


Figura 34 – Malha do caso 5 dividida em: (a) 2, (b) 4 e (c) 8 processos

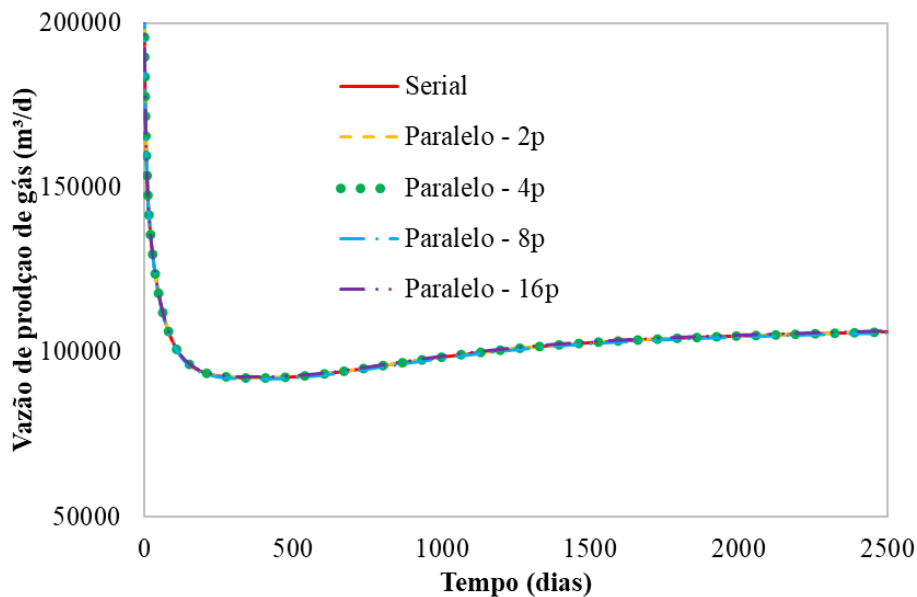


As curvas de produção de óleo e de gás estão representadas na Figura 35.

Figura 35 – Curvas de produção de (a) óleo e de (b) gás do caso 5



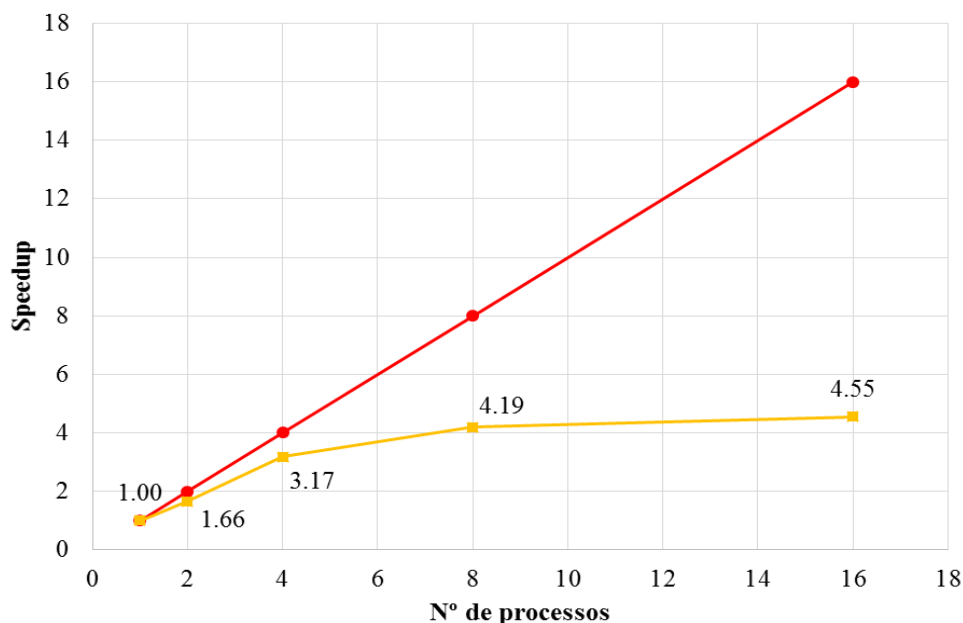
(a)



(b)

A curva de *speedup* obtida para esse caso é mostrada na Figura 36. De todos os casos testados neste trabalho, este apresentou o pior resultado. A paralelização ainda faz com que o caso seja rodado aproximadamente 4,5 vezes mais rápido que sua versão serial, mas, apesar disso, ainda está muito longe da idealidade. Assim como mencionado no caso 2 (seção 5.1.2), é preciso avaliar até quando aumentar o número de processos é uma vantagem.

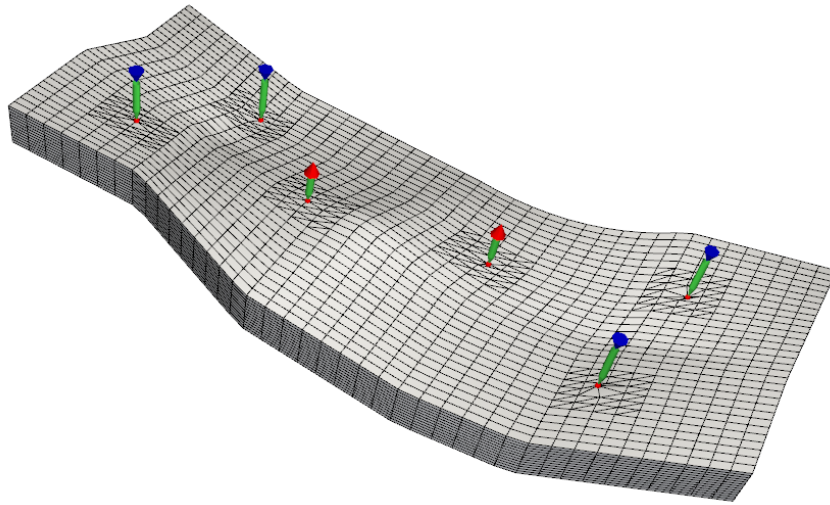


Figura 36 – Curva de *speedup* do caso 5

### 5.2.2 Caso 6

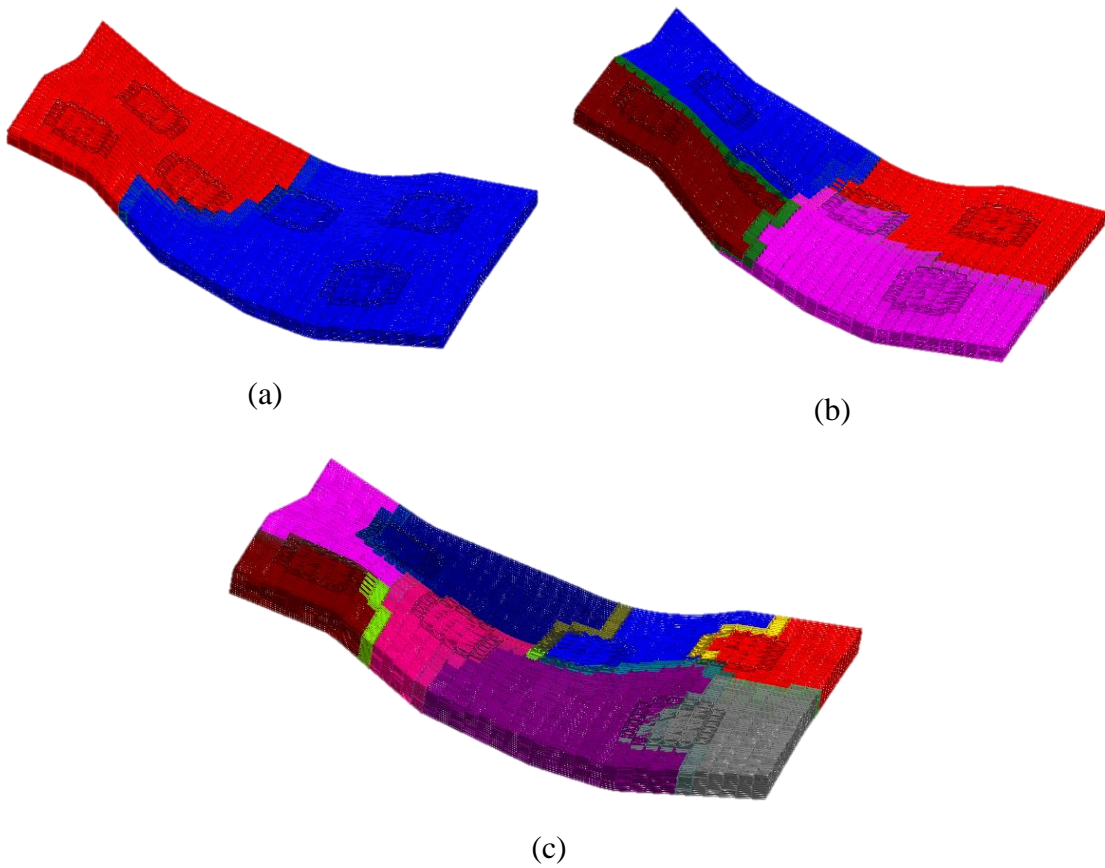
O caso 6 possui as mesmas propriedades de fluido e reservatório apresentadas no caso 2 (ver seção 5.1.2). As propriedades de poços também são as mesmas, ou seja, injeção de gás em reservatório inicialmente trifásico (água, óleo e gás), porém com água imóvel durante toda a simulação. O reservatório é irregular e possui seis poços, sendo quatro injetores (topo azul) e dois produtores (topo vermelho). Essa malha possui 14 camadas em Z, onde a injeção ocorre nas 5 camadas superiores, e a produção nas 7 camadas inferiores. Este caso é um exemplo com malha mista e mostra a flexibilidade do EbFVM para trabalhar com vários tipos de elementos. A malha é representada na Figura 37. Ela contém 19928 vértices e 34840 elementos, sendo 14352 tetraedros, 12688 hexaedros e 7800 pirâmides. As regiões de tetraedros se concentram ao redor dos poços. Elas são regiões mais refinadas, com seis vezes mais elementos que a região de hexaedros. As pirâmides servem como elementos de transição, onde suas faces triangulares se encaixam com os tetraedros, e sua face quadrangular (base da pirâmide) se encaixa com os hexaedros. O propósito desse refino localizado é captar com maior precisão as regiões de maior gradiente de pressão, as quais são ao redor dos poços. Isso auxilia também na redução das oscilações numéricas associadas à formulação IMPEC. Outras formulações com maior grau de implicitude são mais tolerantes à ausência de refino localizado e permitem a utilização de passos de tempo maiores (FERNANDES, 2014).

Figura 37 – Malha original do caso 6



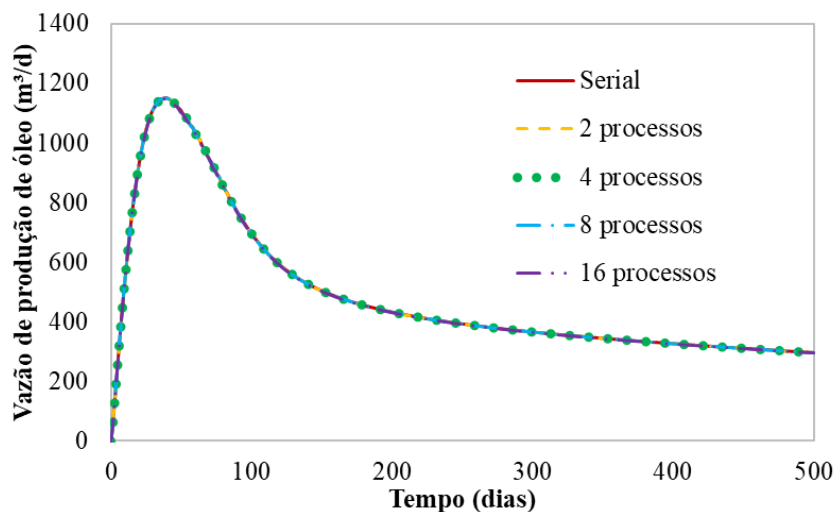
A Figura 38 mostra a malha dividida em 2, 4, e 8 processos.

Figura 38 – Malha do caso 6 dividida em: (a) 2, (b) 4 e (c) 8 processos

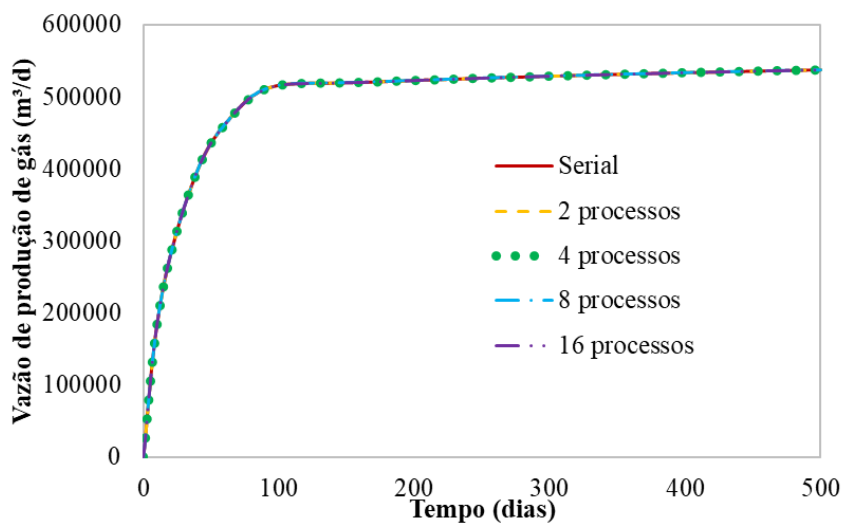


As curvas de produção de óleo e de gás são mostradas na Figura 39. Mais uma vez, as curvas demonstram que o resultado independe do número de processos utilizados na resolução do problema.

Figura 39 – Curvas de produção de (a) óleo e (b) gás do caso 6

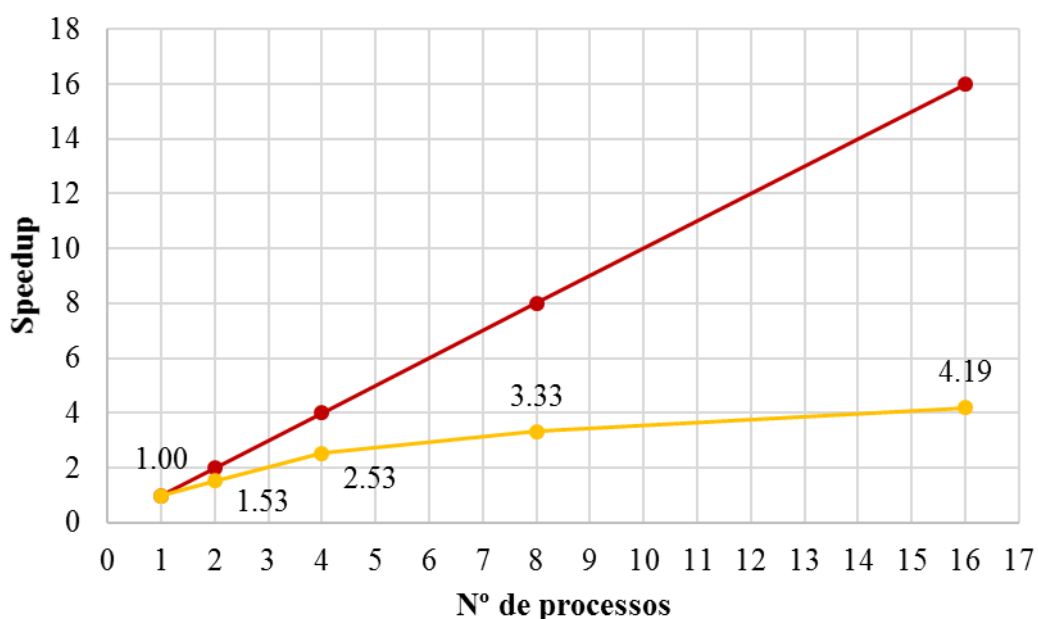


(a)



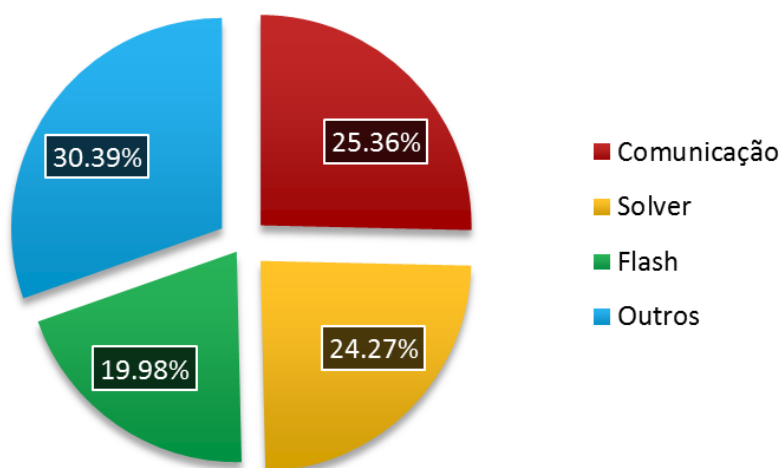
(b)

A curva de *speedup* desse caso é mostrada na Figura 40. Apesar dos resultados satisfatórios obtidos com 2 e 4 processos, o mesmo não pôde ser observado com 8 e 16 processos.

Figura 40 – Curva de *speedup* do caso 6

Um estudo dos tempos de comunicação, mostrado na Figura 41, revela que os custos com transferência de informações entre os processos foram muito altos quando se trabalhou com 16 processos. Mesmo não sendo tão alto quanto o apresentado no caso 2, o valor de 25% ainda é bastante considerável. Outro fator a ser levado em conta é que, neste caso, o cálculo *flash* foi responsável por 20% do tempo de simulação. Por ser um caso 3D, é necessário computar também a influência da gravidade na segregação de fases. Por conta desse fator, acredita-se que o cálculo *flash* tenha demandado mais tempo neste caso. No caso 2, esse cálculo demandou menos de 10% do tempo total de CPU.

Figura 41 – Custos de comunicação do caso 6 com 16 processos

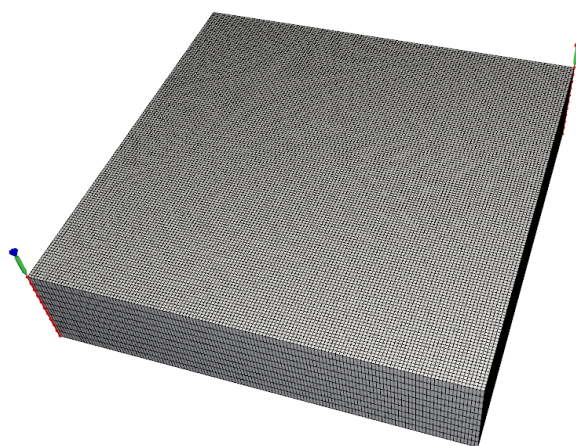


### 5.2.3 Caso 7

Este caso possui as mesmas propriedades de reservatório apresentadas no caso 1 (ver seção 5.1.1), com exceção das dimensões do reservatório. Elas foram alteradas para as dimensões de 397,0 x 397,0 x 91,5 metros. A vazão de injeção também foi alterada para 566.340 m<sup>3</sup>/dia. Esse aumento da vazão teve o propósito de adequar o esquema de produção ao reservatório maior. Ambos os poços estão completados em todas as camadas.

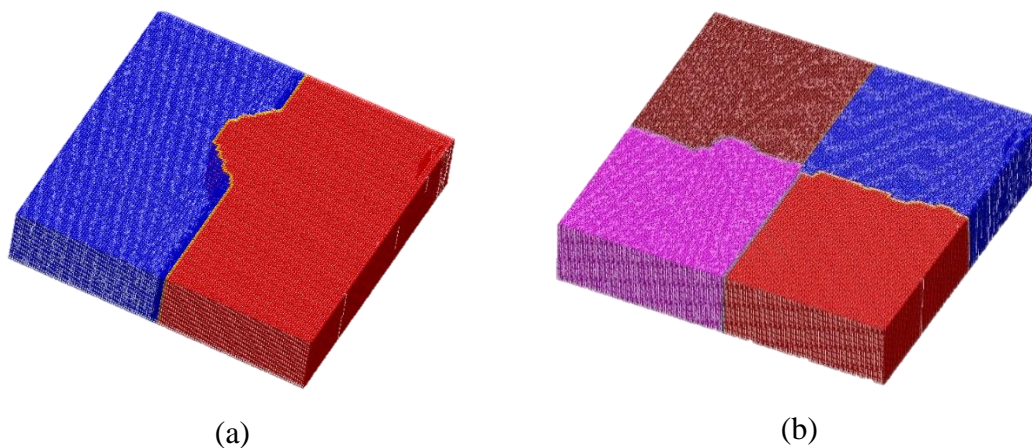
A malha utilizada nesse caso possui 203456 vértices e 182250 elementos hexaédricos regulares. Ela é mostrada na Figura 42.

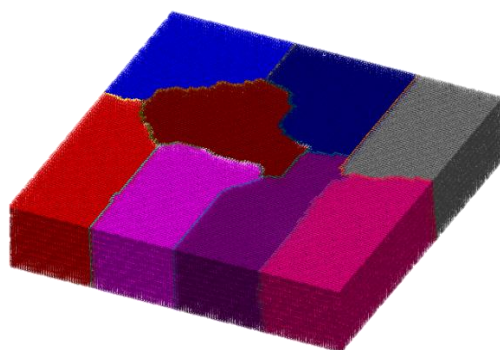
Figura 42 – Malha original do caso 7



A malha dividida em 2, 4 e 8 processos é mostrada na Figura 43.

Figura 43 – Malha do caso 7 dividida em: (a) 2, (b) 4 e (c) 8 processos

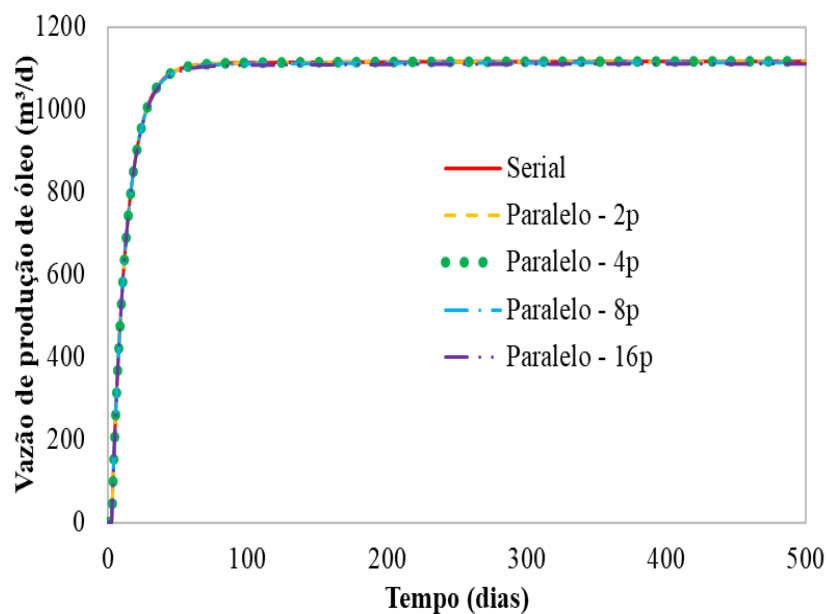




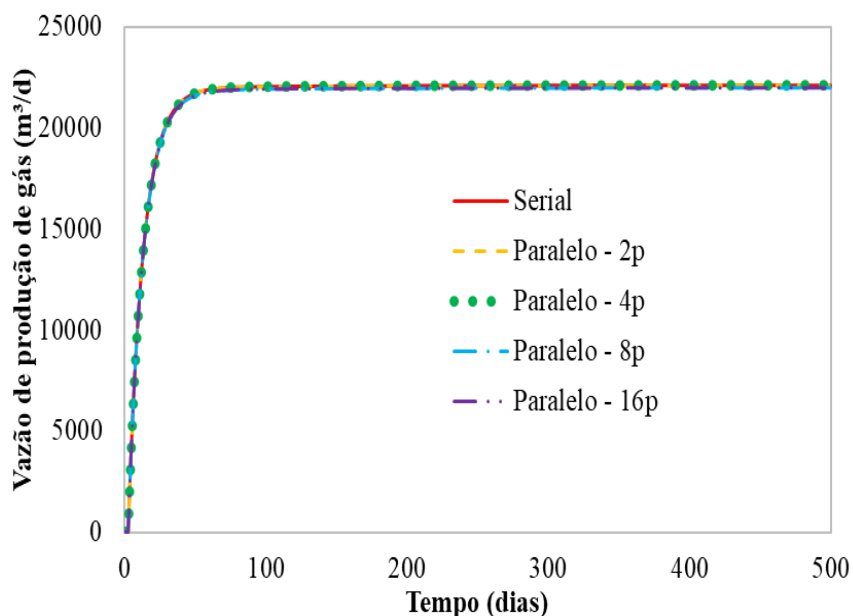
(c)

As curvas de produção de óleo e de gás são apresentadas na Figura 44. O comportamento da curva é o mesmo do caso 1, contudo aqui o *breakthrough* não foi atingido devido ao longo tempo de CPU requerido para este caso. Ele só ocorre por volta de 1300 dias de simulação.

Figura 44 – Curvas de produção de (a) óleo e de (b) gás do caso 7

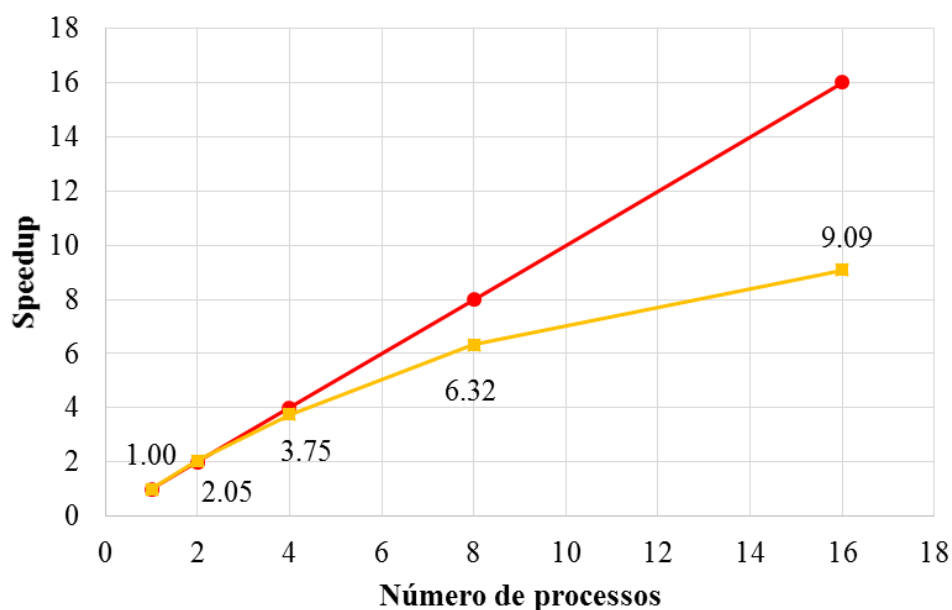


(a)



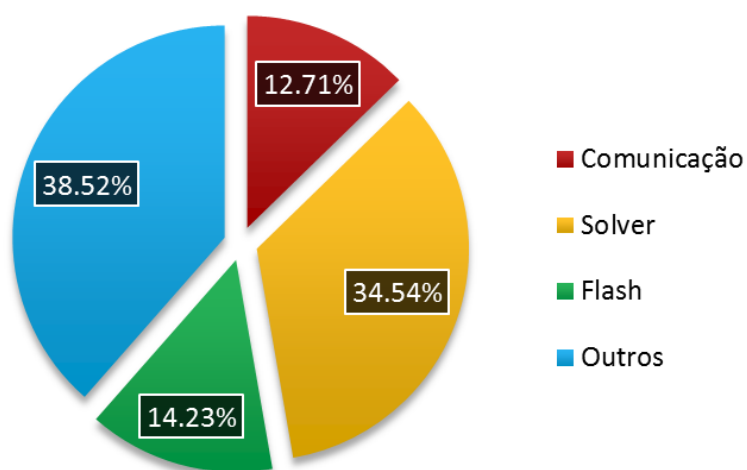
(b)

A curva de *speedup* mostrada na Figura 45 apresenta um resultado superlinear, assim como no caso 3. Mais uma vez, isso se deve a oscilações no tempo de CPU medido. Da Figura 45, pode-se observar que os *speedups* obtidos são os melhores, comparando-se com os outros resultados deste trabalho. Um fator que contribui para isso é o grau de refino da malha. Quanto mais refinada a malha, mais tempo será gasto para que um processo resolva sua porção da malha. Com isso, os custos de comunicação passam a representar uma parcela menor do tempo de execução do programa, e o *speedup* aumenta.

Figura 45 – Curva de *speedup* do caso 7

Em contraste com o que foi apresentado no caso 6, os tempos de comunicação neste caso foram responsáveis por apenas 12,71% do tempo total de CPU. Isso fez com que o *speedup* deste caso chegasse a 9,09 com 16 processos, o maior conseguido neste trabalho. Acredita-se que o principal fator responsável por isso foi o número de vértices da malha, aproximadamente 10 vezes maior que a malha do caso 6.

Figura 46 – Custos de comunicação do caso 7 com 16 processos





## 6 CONCLUSÃO

A implementação da paralelização em um simulador de reservatórios de petróleo é crucial para que se tenham resultados de casos mais complexos, com grande número de elementos, componentes e poços, em tempo hábil. Este trabalho mostrou que com a paralelização foi possível reduzir o tempo de simulação em até 9 vezes em relação ao tempo original do caso em serial.

Destaca-se que a física do problema foi respeitada durante o desenvolvimento deste trabalho. Pode-se observar que as curvas de produção de óleo e de gás das simulações em paralelo são as mesmas obtidas com as simulações em serial, ou seja, são independentes do número de processos. Isso mostra que as rotinas do código responsáveis pelo cálculo de propriedades físicas, cálculo *flash*, etc., foram paralelizadas corretamente.

Foram utilizadas bibliotecas de código aberto com o intuito de reduzir o custo de implementação da paralelização. Essas bibliotecas apresentaram um excelente desempenho e auxiliaram com ferramentas que, de outra maneira, seriam muito mais difíceis de serem implementadas.

O código desenvolvido neste trabalho possui vantagens importantes:

- A comunicação entre processos ocorre de maneira ou assíncrona ou coletiva onde é possível, reduzindo os custos de comunicação.
- O Método dos Volumes Finitos baseado em Elementos suporta triângulos e quadriláteros em 2D, e tetraedros, hexaedros, prismas e pirâmides em 3D. Essa funcionalidade não foi encontrada em nenhum trabalho pesquisado pelo autor.
- O fato de o poço ser dividido por conta da divisão da malha, nos casos em 3D, não afeta o desempenho do código, visto que cada processo é responsável apenas por sua parte da malha, incluindo o poço. Alguns autores relataram essa “quebra” do poço como um fator negativo em seus trabalhos (BECKNER et al., 2015b; USADI et al., 2007).
- As bibliotecas de código aberto reduziram o custo de implementação da paralelização no simulador.

Diante do que foi apresentado, e baseando-se nos resultados obtidos, pode-se concluir que os objetivos propostos neste trabalho foram atingidos com sucesso. O código chegou a rodar até 9 vezes mais rápido que sua versão serial em alguns casos.

## 7 SUGESTÕES DE TRABALHOS FUTUROS

Alguns resultados mostraram uma curva de *speedup* longe da curva ideal. É claro que o comportamento do *speedup* é assintótico. Porém, deseja-se que essa assíntota seja percebida o mais longe possível da origem do gráfico. Por isso, ainda é necessário que sejam feitos estudos de melhoria das rotinas de comunicação e ajustes do *solver*. É importante destacar que esses dois tópicos são os principais fatores de degradação do *speedup*. O PETSc, por exemplo, possui vários *solvers* e pré-condicionadores disponíveis. Neste trabalho utilizou-se a configuração padrão do PETSc: GMRES(30) com ILU(0) em serial, e ILU(0) com Block-Jacobi em paralelo. Avaliar outros pré-condicionadores, como ASM (*Additive Schwarz Method*), DD (Domain Decomposition) e AMG (*Algebraic Multi Grid*), bem como outros métodos de solução dos sistemas lineares, como o BICG (*Biconjugate Gradient*) ou o BICGSTAB (*Biconjugate Gradient Stabilized*), são vitais para a otimização do tempo de CPU gasto com a solução dos sistemas lineares, em paralelo e serial.

Um ponto importante diz respeito às malhas utilizadas no simulador. Malhas cartesianas e *corner-point* ainda são bastante utilizadas na simulação de reservatórios. Apesar de o UTCOMP possuir uma versão paralelizada para malhas cartesianas e não-estruturadas, ainda é preciso acrescentar as malhas *corner-point* e juntar essas três abordagens sob um único *framework*. Isso daria mais flexibilidade ao simulador e ao usuário, que poderia escolher a abordagem conveniente.

Outro fator de melhoria é a paralelização das formulações FI (*fully implicit*) e AIM (*Adaptive Implicit Method*). Essas formulações existem no código e são numericamente mais vantajosas que a formulação IMPEC (*Implicit Pressure and Composition*) utilizada neste trabalho. Elas são numericamente mais estáveis, e possibilitam a utilização de passos de tempo maiores. Poder trabalhar com as formulações FI e AIM em paralelo representaria um grande avanço para o simulador.

Acredita-se que a adição dessas funcionalidades ao código o tornariam apto a rodar uma gama maior de casos, com parâmetros de operação mais conectados com prática industrial, além de proporcionar simulações mais rápidas e eficientes.

## REFERÊNCIAS

- ACS, G.; DOLESCHALL, S.; FARKAS, E. General Purpose Compositional Model. **Society of Petroleum Engineers Journal**, v. 25, n. 4, p. 543–553, 1985.
- ARAÚJO, A. L. S.; FERNANDES, B. R. B.; DRUMOND FILHO, E. P.; ARAUJO, R. M.; LIMA, I. C. M.; GONÇALVES, A. D. R.; MARCONDES, F.; SEPEHRNOORI, K. 3D Compositional Reservoir Simulation In Conjunction With Unstructured Grids. **Brazilian Journal of Chemical Engineering**, v. 33, n. 2, p. 347–360, 2016.
- BALAY, S.; ABHYANKAR, S.; ADAMS, M.; BROWN, J.; BRUNE, P.; BUSCHELMAN, K.; GROPP, W. **PETSc Users ManualWork**. Condado de DuPageArgonne National Laboratory, 2015.
- BALIGA, B. R.; PATANKAR, S. V. A New Finite-Element Formulation For Convection-Diffusion Problems. **Numerical Heat Transfer**, v. 3, n. 4, p. 393–409, 1980.
- BECKNER, B. L.; HAUGEN, K. B.; MALIASSOV, S.; DYADECHKO, V.; WIEGAND, K. D. **General Parallel Reservoir Simulation**. SPE Abu Dhabi International Petroleum Exhibition and Conference. Abu Dhabi: Society of Petroleum Engineers, 2015a.
- BECKNER, B. L.; HAUGEN, K. B.; MALIASSOV, S.; DYADECHKO, V.; WIEGAND, K. D. General Parallel Reservoir Simulation. **SPE Abu Dhabi International Petroleum Exhibition and Conference**, p. 10, 2015b.
- CHANG, Y.-B. **Development and Application of an Equation of State Compositional Simulator**. Tese (Doutorado em Engenharia de Petróleo) – The University of Texas at Austin, Austin, 1990.
- CORDAZZO, J.; MALISKA, C. R.; SILVA, A. F. C.; HURTADO, F. S. V. **An Element Based Conservative Scheme using Unstructured Grids for Reservoir Simulation**. SPE Youth Forum. 2005.
- DEVINE, K.; BOMAN, E.; HEAPHY, R.; HENDRICKSON, B.; VAUGHAN, C. Zoltan data management services for parallel dynamic applications. **Computing in Science & Engineering**, v. 4, n. 2, p. 90–96, 2002.
- DOGRU, A. H.; FUNG, L. S. K.; MIDDYA, U.; AL-SHAALAN, T.; PITA, J. A. **A Next-Generation Parallel Reservoir Simulator for Giant Reservoirs**. SPE Reservoir Simulation Symposium. The Woodlands: Society of Petroleum Engineers, 2009.  
Disponível em:  
<<http://www.earthdoc.org/publication/publicationdetails/?publication=42008>>
- FERNANDES, B. R. B. **Implicit And Semi-Implicit Techniques For The Compositional Petroleum Reservoir Simulation Based On Volume Balance**. Dissertação (Mestrado em Engenharia Química) – Universidade Federal do Ceará, Fortaleza, 2014.

FERNANDES, B. R. B.; DRUMOND FILHO, E. P.; GONÇALVES, A. D. R.; SEPEHRNOORI, K.; MARCONDES, F. **Well Indices For Complex Well Configurations In Unstructured Grids**. XXXV Ibero-Latin American Congress on Computational Methods in Engineering. Fortaleza, 2014.

FERNANDES, B. R. B.; GONÇALVES, A. D. R.; FILHO, E. P. D.; LIMA, I. C. M.; MARCONDES, F.; SEPEHRNOORI, K. A 3D Total Variation Diminishing Scheme for Compositional Reservoir Simulation Using the Element-Based Finite-Volume Method. **Numerical Heat Transfer, Part A: Applications**, v. 67, n. 8, p. 839–856, 2015.

FERNANDES, B. R. B.; MARCONDES, F.; SEPEHRNOORI, K. Investigation of Several Interpolation Functions for Unstructured Meshes in Conjunction with Compositional Reservoir Simulation. **Numerical Heat Transfer, Part A: Applications**, v. 64, n. 12, p. 974–993, 2013.

FUNG, L. S. K.; DOGRU, A. H. **Distributed Unstructured Grid Infrastructure for Complex Reservoir Simulation**. Europec/EAGE Conference and Exhibition. Rome: Society of Petroleum Engineers, 2008. Disponível em: <<http://www.onepetro.org/doi/10.2118/113906-MS>>

GABRIEL, E.; FAGG, G. E.; BOSILCA, G.; ANGSKUN, T.; DONGARRA, J. J.; SQUYRES, J. M.; SAHAY, V.; KAMBADUR, P.; BARRETT, B.; LUMSDAINE, A.; CASTAIN, R. H.; DANIEL, D. J.; GRAHAM, R. L.; WOODALL, T. S. Open MPI: Goals, Concept, and Design of a Next Generation MPI Implementation. **11th European PVM/MPI Users' Group Meeting**. p. 97–104, 2004.

GHORI, S. G.; WANG, C. H.; LIM, M. T.; POPE, G. A.; SEPEHRNOORI, K.; WHEELER, M. F. **Compositional Reservoir Simulation on CM-5 and KSR-1 Parallel Machines**. 13th SPE Symposium on Reservoir Simulation. San Antonio: Society of Petroleum Engineers, 1995. Disponível em: <<http://www.onepetro.org/doi/10.2118/29140-MS>>

GRAMA, A.; GUPTA, A.; KARYPIS, G.; KUMAR, V. **Introduction to Parallel Computing**. 2<sup>a</sup> ed. Harlow: Pearson Education Limited, 2003.

GROPP, W.; LUSK, E.; SKJELLUM, A. **Using MPI**. 2 ed ed. Cambridge: The MIT Press, 1999.

HAN, C.; DELSHAD, M.; SEPEHRNOORI, K.; POPE, G. A. **A Fully Implicit, Parallel, Compositional Chemical Flooding Simulator**. SPE Annual Technical Conference and Exhibition. Dallas: Society of Petroleum Engineers, 2007. Disponível em: <<http://www.onepetro.org/doi/10.2118/97217-PA>>

HEINEMANN, Z. E.; BRAND, C. W. **Gridding Techniques in Reservoir Simulation**. Proceedings of the First and Second International Forum on Reservoir Simulation. Alpbach, 1988.

HERNING, F.; ZIPPERER, L. Calculation of the Viscosity of Technical Gas Mixtures from the Viscosity of Individuals Gases. **Gas and Wasserfach**, v. 79, p. 49–54, 1936.

JOSSI, J. A.; STIEL, L. I.; THODOS, G. The viscosity of pure substances in the dense gaseous and liquid phases. **AIChE Journal**, v. 8, n. 1, p. 59–63, 1962.

KILLOUGH, J. E. **Will Parallel Computing Ever Be Practical?** Middle East Oil Show. Manama: Society of Petroleum Engineers, 1993. Disponível em: <<http://www.onepetro.org/doi/10.2118/25556-MS>>

KILLOUGH, J. E.; BHOGESWARA, R. **Simulation of Compositional Reservoir Phenomena on a Hypercube**. First IMA/SPE European Conference on the Mathematics of Oil Recovery. Oxford University Press, 1990.

KILLOUGH, J. E.; BHOGESWARA, R. Simulation of Compositional Reservoir Phenomena on a Distributed-Memory Parallel Computer. **Journal of Petroleum Technology**, v. 43, n. 11, p. 1368–1374, 1991.

LASALLE, D.; KARYPIS, G. **Multi-threaded graph partitioning**. IEEE 27th International Parallel and Distributed Processing Symposium, IPDPS, 2013.

LIMA, P. C. R. **Os Desafios , os Impactos o a Gestão da Exploração do Pré-Sal**. Câmara dos Deputados, Brasília, 2008.

LIU, H.; WANG, K.; CHEN, Z.; LUO, J.; WU, S.; WANG, B. **Development of Parallel Reservoir Simulators on Distributed-memory Supercomputers**. SPE Reservoir Characterisation and Simulation Conference and Exhibition. Abu Dhabi: Society of Petroleum Engineers, 2015. Disponível em: <<http://www.onepetro.org/doi/10.2118/175573-MS>>

LOHRENZ, J.; BRAY, B. G.; CLARK, C. R. Calculating Viscosities of Reservoir Fluids From Their Compositions. **Journal of Petroleum Technology**, v. 16, n. 10, p. 1171–1176, 1964.

MAGRAS, J.-F.; QUANDALLE, P.; BIA, P. **High-Performance Reservoir Simulation With Parallel ATHOS**. SPE Reservoir Simulation Symposium. Houston: Society of Petroleum Engineers, 2001. Disponível em: <<http://www.onepetro.org/doi/10.2118/66342-MS>>

MALISKA, C. R. **Trânsferência de Calor e Mecânica dos Fluidos Computacional**. 2ª ed. Rio de Janeiro: LTC, 2014.

MARCONDES, F. **Solução Numérica Usando Métodos Adaptativos-Implicitos E Malha De Voronoi De Problemas De Reservatórios De Petróleo**. Tese (Doutorado em Engenharia Mecânica) – Universidade Federal de Santa Catarina, Florianópolis, 1996.

MARCONDES, F.; SEPEHRNOORI, K. An element-based finite-volume method approach for heterogeneous and anisotropic compositional reservoir simulation. **Journal of Petroleum Science and Engineering**, v. 73, n. 1–2, p. 99–106, 2010.

PENG, D.-Y.; ROBINSON, D. B. A New Two-Constant Equation of State. **Industrial & Engineering Chemistry Fundamentals**, v. 15, n. 1, p. 59–64, 1976.

PERSCHKE, D. R. **Equation of State Phase Behavior Modeling for Compositional Simulation**. Tese (Doutorado em Engenharia de Petróleo) – The University of Texas at Austin, Austin, 1988.

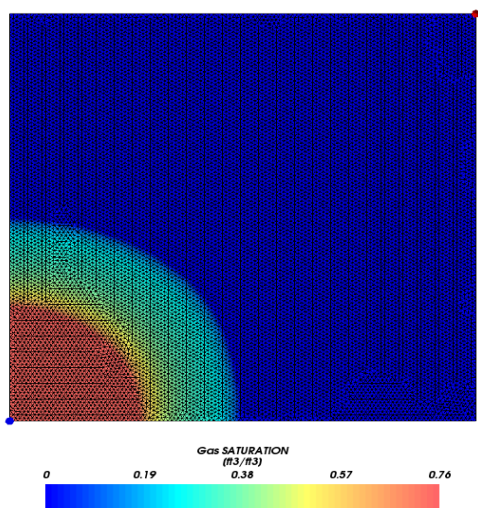
- PRÉVOST, M.; EDWARDS, M. G.; BLUNT, M. J. **Streamline Tracing on Curvilinear and Unstructured Grids**. SPE Reservoir Simulation Symposium. Houston, 2001.
- ROZON, B. J. **A Generalized Finite Volume Discretization Method for Reservoir Simulation**. Reservoir Simulation Symposium. Houston, 1989.
- SCHLOEGEL, K.; KARYPIS, G.; KUMAR, V. Parallel static and dynamic multi-constraint graph partitioning. **Concurrency and Computation: Practice and Experience**, v. 14, n. 3, p. 219–240, 2002.
- SCOTT, S. L.; WAINWRIGHT, R. L.; RAGHAVAN, R.; DEMUTH, H. **Application of Parallel (MIMD) Computers to Reservoir Simulation**. SPE Symposium on Reservoir Simulation. San Antonio: Society of Petroleum Engineers, 1987. Disponível em: <<http://www.onepetro.org/doi/10.2118/16020-MS>>
- SEOL, E. S. **FMDB: Flexible Distributed Mesh Database For Parallel Automated Adaptive Analysis**. Tese (Doutorado em Ciências da Computação) – Rensselaer Polytechnic Institute, Troy, 2005.
- SHIRALKAR, G. S.; FLEMING, G. C.; WATTS, J. W.; WONG, T. W.; COATS, B. K.; MOSSBARGER, R.; ROBBANA, E.; BATTEN, A. H. **Development and Field Application of a High Performance, Unstructured Simulator with Parallel Capability**. SPE Reservoir Simulation Symposium. Houston: Society of Petroleum Engineers, 2005. Disponível em: <<http://www.spe.org/elibrary/servlet/spepreview?id=00093080>>
- SILVA, R. S. DA. **Simulação de Escoamento Bifásico Oléo- Água em Reservatórios de Petróleo Usando Computadores Paralelos de Memória Distribuída**. Tese (Doutorado em Engenharia Civil) – Universidade Federal de Pernambuco, Recife, 2008.
- STIEL, L. I.; THODOS, G. The viscosity of nonpolar gases at normal pressures. **AIChE Journal**, v. 7, n. 4, p. 611–615, 1961.
- STONE, H. L. Estimation of Three-Phase Relative Permeability And Residual Oil Data. **Journal of Canadian Petroleum Technology**, v. 12, n. 4, p. 1171–1176, 1973.
- USADI, A. K.; MISHEV, I. D.; SHAW, J. S.; WIEGAND, K. D. **Parallelization on Unstructured Grids**. SPE Reservoir Simulation Symposium. Houston: Society of Petroleum Engineers, 2007. Disponível em: <<http://www.onepetro.org/doi/10.2118/106063-MS>>
- WATTS, J. W. A Compositional Formulation of the Pressure and Saturation Equations. **SPE Reservoir Engineering**, v. 1, n. 3, p. 243–252, 1986.
- WHEELER, J. A.; SMITH, R. A. Reservoir Simulation on a Hypercube. **SPE Reservoir Engineering**, v. 5, n. 4, p. 544–548, 1990.

## APÊNDICE A – CAMPOS DE SATURAÇÃO DE ÓLEO E DE GÁS

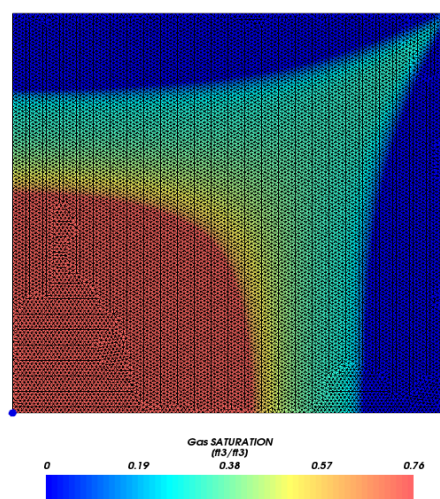
A seguir apresentam-se as figuras dos campos de saturação de óleo e de gás obtidas para os casos apresentados. Elas visam fornecer melhor compreensão sobre o comportamento dos fluidos no reservatório durante a simulação.

- Caso 1

Campos de saturação de gás em (a) 50 e (b) 200 dias:



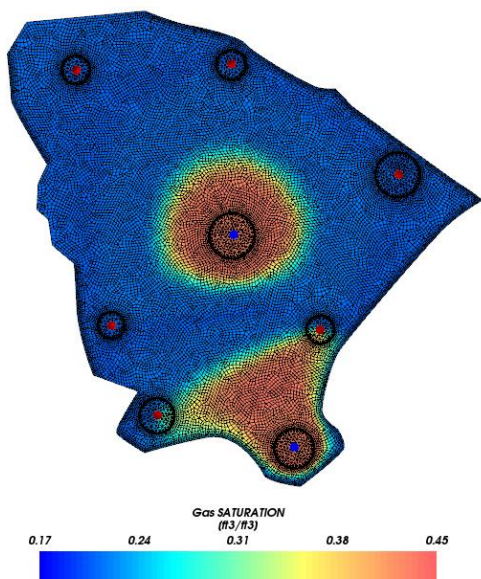
(a)



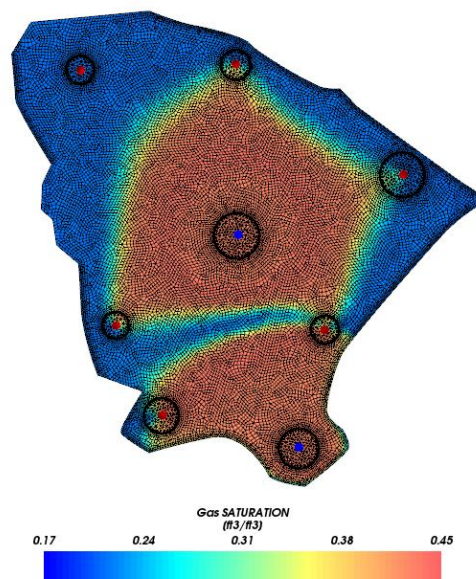
(b)

- Caso 2

Campos de saturação de gás em 1000 e 3500 dias:



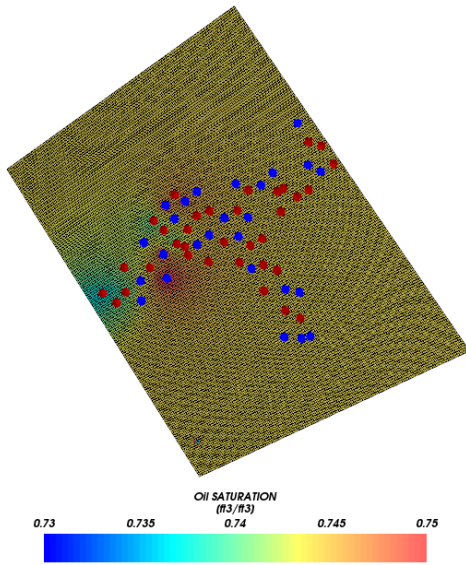
(a)



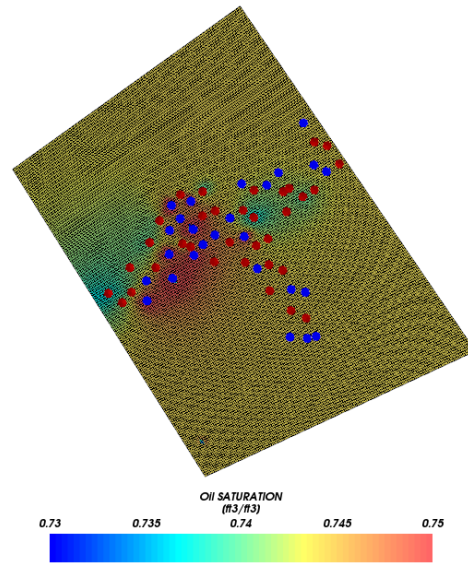
(b)

- Caso 3

Campos de saturação de óleo em 675 e 1021 dias:



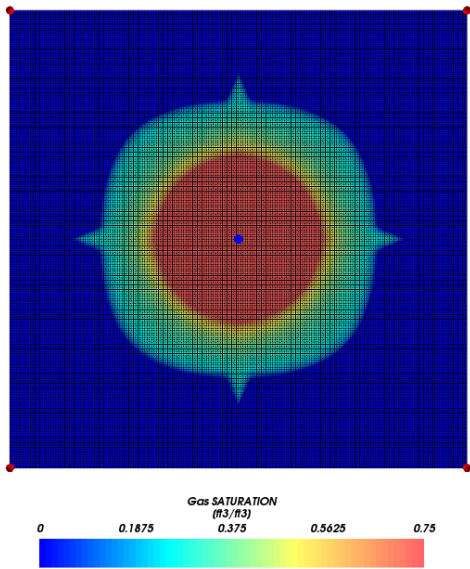
(a)



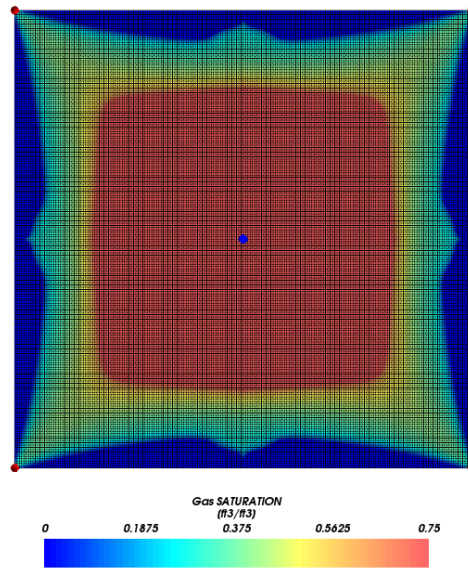
(b)

- Caso 4

Campos de saturação de gás em 80 e 300 dias:



(a)

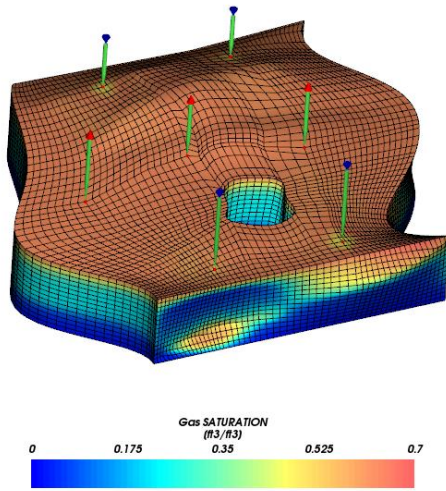


(b)

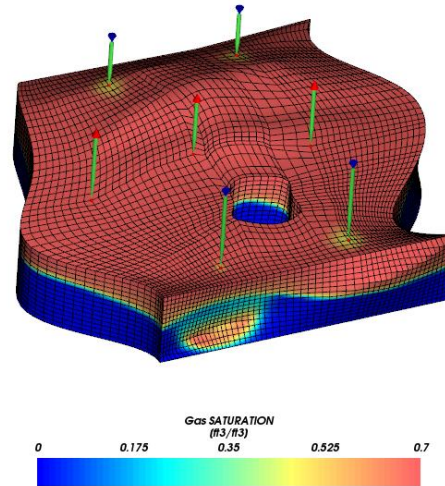


- Caso 5

Campos de saturação de gás em 200 e 2000 dias:



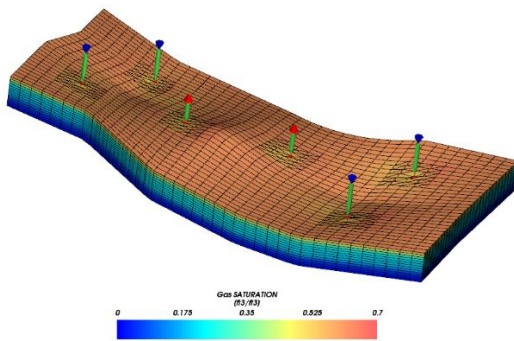
(a)



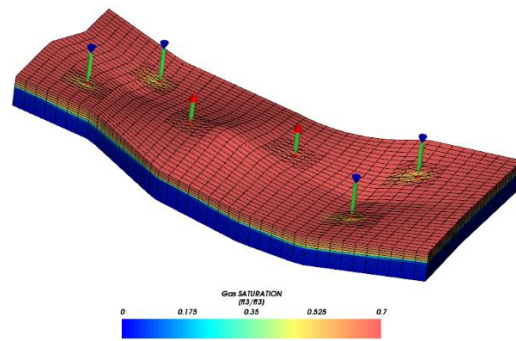
(b)

- Caso 6

Campos de saturação de gás em 100 e 500 dias:



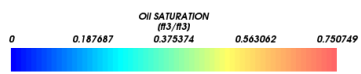
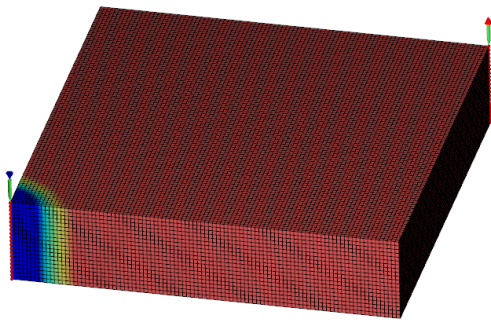
(a)



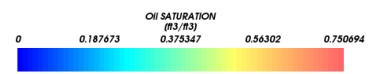
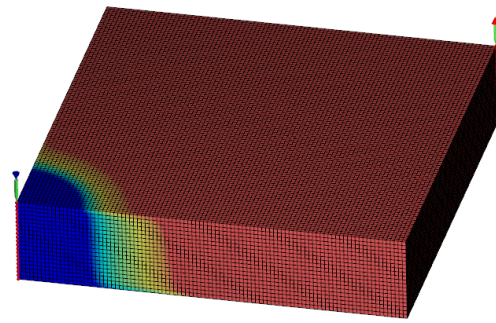
(b)

- Caso 7

Campos de saturação de óleo em 50 e 300 dias:



(a)



(b)