Universidade Federal de Pernambuco
Centro de Tecnologia e Geociências
Departamento de Engenharia Mecânica
Programa de Pós Graduação em Engenharia Mecânica

Saulo Menezes de Melo

Estratégia de adaptação local na simulação de escoamentos bifásicos em meios porosos

Recife/2014

Saulo Menezes de Melo

Estratégia de adaptação local na simulação de escoamentos bifásicos em meios porosos

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Engenharia Mecânica, PPGEM, da Universidade Federal de Pernambuco como parte dos requisitos necessários à obtenção do grau de Mestre em Engenharia Mecânica.

Área de concentração: Mecânica computacional / Projetos

Orientador: Paulo Roberto Maciel Lyra , *PhD.*

Co-orientador: Rogério Soares da Silva, *Dr.*

Recife/2014

Catalogação na fonte
Bibliotecária Valdicea Alves Silva, CRB-4 / 1260

M528e Melo, Saulo Menezes de.
Estratégia de adaptação local na simulação de escoamentos bifásicos em meios porosos/ Saulo Menezes de Melo - Recife: O Autor, 2014.
80folhas, Il.; Acro.,e Sig., Sim. e Tabs.

Orientador: Prof. Dr. Paulo Roberto Maciel Lyra.
Coorientador: Prof. Dr. Rogério Soares da Silva.

Dissertação (Mestrado) – Universidade Federal de Pernambuco. CTG. Programa de Pós-Graduação em Engenharia Mecânica, 2014.
Inclui Referências e Apêndices.

1. Engenharia Mecânica. 2. Simulação adaptativa. 3. Redefinição de malhas. 4. Malha de "background". 5. Gmsh. I. Método dos Volumes finitos. 6. Reservatórios de petróleo. I. Lyra, Paulo Roberto Maciel. (Orientador). II. Silva, Rogério Soares da. (Coorientador) III. Título.

UFPE

621 CDD (22. ed.) BCTG/2015 – 91

07 de agosto de 2014

"ESTRATÉGIA DE ADAPTAÇÃO VIA REDEFINIÇÃO LOCAL DE MALHA NA SIMULAÇÃO DE ESCOMENTOS BIFÁSICOS EM MEIOS POROSOS"

SAULO MENEZES DE MELO

ESTA DISSERTAÇÃO FOI JULGADA ADEQUADA PARA OBTENÇÃO DO TÍTULO DE MESTRE EM ENGENHARIA MECÂNICA

ÁREA DE CONCENTRAÇÃO: MECÂNICA COMPUTACIONAL E PROJETO MECÂNICO

APROVADA EM SUA FORMA FINAL PELO PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA/CTG/EEP/UFPE

_____

Prof. Dr. PAULO ROBERTO MACIEL LYRA

ORIENTADOR/PRESIDENTE

_____

Prof. Dr. ROGÉRIO SOARES D SILVA

CO-ORIENTADOR/PRESIDENTE

_____

Prof. Dr. JORGE RECARTE HENRÍQUEZ GUERRERO

COORDENADOR DO PROGRAMA

BANCA EXAMINADORA:

_____

Prof. Dr. PAULO ROBERTO MACIEL LYRA (UFPE)

_____

Prof. Dr. ROGÉRIO SOARES DA SILVA (UFPE)

_____

Prof. Dr. DARLAN KARLO ELISIÁRIO DE CARVALHO (UFPE)

_____

Prof. Dr. JOAQUIM BENTO CAVALCANTE NETO (UFC)

AGRADECIMENTOS

A **Deus** pela força e discernimentos concedidos.

Ao professor **Paulo R. M. Lyra**.

Ao professor **Rogério Soares**, peça de fundamental importância durante o desenvolvimento do trabalho.

À minha mãe **Deuzeni Melo** pelo apoio incondicional em todos os momentos.

Á minha sogra **Alaíde Santos**, por todo apoio e incentivo em todos os momentos dessa caminhada, sua participação ainda que indireta, impacta fortemente sobre este trabalho.

A todos os amigos do PADMEC pelo bom ambiente de trabalho proporcionado no convívio diário, **Marcelo, Guilherme K. Caminha, Márcio e Bruno.**

À **CAPES- Coordenação de Aperfeiçoamento de Pessoal de Nível Superior**, pelo suporte financeiro durante o período de desenvolvimento deste trabalho.

A todos que contribuíram direta ou indiretamente para realização deste trabalho, professores **Darlan Karlo Elisiário, Ramiro Brito e Joaquim Bento.**

"Inteligência é a capacidade de se adaptar às mudanças."

Stephen Hawking

RESUMO

Neste trabalho, abordamos o processo de adaptação de malhas via "remeshing", a adaptação de malhas não estruturadas é usada aqui como uma ferramenta de alto desempenho que visa simultaneamente garantir a qualidade do resultado e reduzir o tempo computacional envolvido em toda a simulação. Usamos um simulador numérico de escoamento óleo-água em reservatório de petróleo em duas dimensões. Os fluidos e a rocha reservatório são tidos como incompressíveis e não se considera variações térmicas. Um método de volumes finitos centrado nos nós e baseado em uma estrutura de dados por arestas é empregado na discretização das equações de pressão e de saturação. Uma formulação agregada IMPES (*Implicit Pressure Explicit Saturation*) é utilizada para resolver as variáveis de saturação e de pressão. Aproximações de baixa e alta ordem na eq. de saturação e seus efeitos na propagação dos erros é calculada. A cada avanço no tempo, novos campos de saturações e de pressões são avaliados e, em seguida, a qualidade destes resultados é estimada. Isto é realizado por meio de um estimador de erros baseado na hessiana dos campos de pressão e de saturação, os quais são usados para calcular os erros local e global referentes a cada campo. Estes erros são comparados a uma tolerância que irá definir se a malha deve ou não ser adaptada, onde e em que grau de refinamento. O método de adaptação desenvolvido é o do "*remeshing*" ou redefinição de malhas, que se baseia na reconstrução total ou parcial da malha. A redefinição da malha é feita com o gerador *open-source Gmsh* que possibilita a criação da malha com controle do grau de refinamento local através da malha de "*background*". Após a adaptação, os campos de pressão e de saturações são interpolados da malha anterior para a nova malha. Métodos lineares, quadráticos e adaptativos de interpolação são explorados e avaliados. Para um mesmo instante de tempo, o processo de adaptação se repete até que a qualidade dos resultados atinja a tolerância exigida. Para a integração e conformidade de todas estas etapas foi desenvolvido um software na linguagem C++ usando a biblioteca para gerenciamento de malhas FMDB (*Flexible Distributed Mesh Data Base*). Foi efetuado um estudo da qualidade dos resultados obtidos e da eficiência da simulação quando da utilização do procedimento de adaptação de malhas desenvolvido em simulações de escoamento monofásico e bifásico óleo-água em meios porosos usando malhas não estruturadas 2D triangulares, considerando meios homogêneos e heterogêneos, isotrópicos e anisotrópicos, comprovando a robustez da metodologia implementada nas simulações efetuadas.

PALAVRAS-CHAVE: Simulação adaptativa. redefinição de malhas. malha de *background*. *Gmsh*. método dos volumes finitos. reservatórios de petróleo.

ABSTRACT

In this work, we address the mesh adaptation process using "remeshing" tehcnique. The adaptation of unstructured meshes is used here as a high performance tool intended both to ensure the quality of the result and to reduce the computational time involved in the whole simulation. The numerical simulation are performed using a two dimensional two phase flow numerical simulation of oil and water in porous media. Fluids and reservoir rock are considered incompressible and does not consider thermal variations are considered. A cell centered finite volume method using an edge based data structure used for of the error the discretization of the pressure and saturation equations. The IMPES formulation (Implicit Pressure Explicit Saturation) composes the structure of the simulator.The hyperbollic saturation equation is discretized using either a first or higher order aproximation and the effect of such choice is analysed. After advancing the solution in time the obtained result is analysed using an a-posteriori erro indicator. At each time step a new field of saturation and pressure is evaluated, and then, the quality of results is assessed. This is accomplished by means of an estimator based on the Hessian of the fields of pressure and saturation, which are used to calculate both the local and global errors for each field. These errors are compared to a tolerance that will define whether or not the mesh must be adapted where the mesh density required. The adaptive method developed is refered to as "remeshing" and considers either total or partial reconstruction of the mesh. The remeshing is done with the open-source mesh generator named Gmsh that enables to build mesh containing the level of refinement through the mesh refinement "background." mesh. After adaptation, the fields of pressure and saturations are interpolated from the previous mesh to the new mesh. Linear and quadratic interpolation methods are explored and evaluated. For the same level of time, the adaptation process is repeated until the quality of results reached the required tolerance. For the integration and accomplishement of all these steps a software was developed in C + + language and using the library for unstructured mesh management FMDB (Flexible Distributed Mesh Data Base) mesh. A study of the quality of the analysed results and of the efficiency of the simulation was performed when using the adaptive mesh procedure developed through the analysis of single-phase flow and two-phase oil-water in porous media using 2D models with triangular unstructured meshes. The analysed examples considers homogeneous and heterogeneous media, isotropic and anisotropic, demonstrating the robustness and efficiency of the methodology implemented for the cases analysed in the simulations performed.

KEYWORDS: Adaptive simulation. background mesh. Gmsh. finite volumn method, oil reservoirs.

LISTA DE FIGURAS

FIGURA 2.1 - MODELO ESQUEMÁTICO DE UM RESERVATÓRIO E SUAS RESPECTIVAS FRONTEIRAS. 21
FIGURA 3.1 – FLUXOGRAMA GERAL DO PROCESSO DE SIMULAÇÃO COM ADAPTAÇÃO NO REGIME ESTACIONÁRIO. 24
FIGURA 3.2 – FLUXOGRAMA GERAL DO PROCESSO DE SIMULAÇÃO COM ADAPTAÇÃO NO REGIME TRANSIENTE. 25
FIGURA 3.3 – A) MALHA INICIAL B) CAMPO DE NOVOS TAMANHOS PARA AS FACES $hnew$ C) MALHA GERADA FINAL. 31
FIGURA 3.4 – INTERFACE GRÁFICA DO *GMSH*. 32
FIGURA 3.5 – DECOMPOSIÇÃO ESPACIAL BIDIMENSIONAL POR ESTRUTURA HIERÁRQUICA *QUADTREE*. 34
FIGURA 3.6 – PASSOS DA GERAÇÃO DA MALHA PELO JMESH A) DISCRETIZAÇÃO DA GEOMETRIA B) GERAÇÃO DA QUADTREE C) FORMAÇÃO DA MALHA NAS REGIÕES INTERNAS ÀS CÉLULAS E REMOÇÃO DAS CÉLULAS PRÓXIMAS AOS CONTORNOS. 35
FIGURA 3.7 – MALHA GERADA PELO JMESH A) RESULTADO FINAL DA ETAPA DE AVANÇO FRONTAL B) MALHA FINAL APÓS MAPEAMENTO DOS NÓS NA SUPERFÍCIE TRIDIMENSIONAL INICIAL 35
FIGURA 3.8 – INTERPOLAÇÃO EM UM ELEMENTO TRIANGULAR 37
FIGURA 3.9 – REGIÕES DE SINGULARIDADES A) CURVA DE PRESSÃO AO LONGO DA DIAGONAL DOS POÇOS E B) CURVA DE SATURAÇÃO AO LONGO DA DIAGONAL ENTRE OS POÇOS. 38
FIGURA 3.10 – REGIÕES DE SINGULARIDADES EM ESCOAMENTO BIFÁSICO. 39
FIGURA 4.1 – UNIAO DE *LINELOOPS* A) "STGNODE" CAUSANDO JUNÇÃO DE *LOOP´S* E "DENTE" B) CONTORNO BEM DEFINIDO COM APENAS UM CAMINHO POSSÍVEL. 41
FIGURA 4.2 – FLUXOGRAMA DE OPERAÇÕES DA FUNÇÃO IDENTIFY_STGELEMENTS. 42
FIGURA 4.3 – RESULTADO DA FUNÇÃO IDENTIFY_STGELEMENTS. 43
FIGURA 4.4 – FLUXOGRAMA FUNÇÃO SAVEBGMVIEW 44
FIGURA 4.5 – REGIÃO DE ADAPTAÇÃO LOCAL ENTRE DOMÍNIOS. 45
FIGURA 4.6 – SEQUÊNCIA DE BUSCA EM ÁRVORE LOCAL ALGORITMO BFS. 46
FIGURA 4.7 – REGIÃO DE ADAPTAÇÃO IDENTIFICADA NA BUSCA BFS. 46
FIGURA 4.8 – REMOÇÃO DAS ARESTAS E FACES. 46
FIGURA 4.9 – CONJUNTO DE ARESTAS INTERNAS DELIMITANDO UMA REGIÃO DE ADAPTAÇÃO 46
FIGURA 4.10 – CONTAINER DA STL MAP COM AS REGIÕES DE ADAPTAÇÃO RESPECTIVAS A CADA SUBDOMÍNIO. 47
FIGURA 4.11 – FLUXOGRAMA DAS OPERAÇÕES DA FUNÇÃO IDENTIFY_AND_REMOVE_EDGES. 48
FIGURA 4.12 – REGIÃO DE ADAPTAÇÃO DELIMITADA POR MAIS DE UM LOOP DE ARESTAS. 49
FIGURA 4.13 - FLUXOGRAMA DA FUNÇÃO SAVEGEOMETRY_2D. 50
FIGURA 4.14 – PROCESSO DE CÓPIA DOS ELEMENTOS TRIANGULARES DA MALHA GERADA PARA A MALHA ORIGINAL. 51
FIGURA 4.15 – CONECTIVIDADES DOS ELEMENTOS TRIÂNGULOS E ARESTAS DO *GMSH*. ADAPTADO DE SILVA (2008) 52
FIGURA 5.1 – A) MODELO "FIVE SPOT" E B) ¼ DE "FIVE SPOT". 53
FIGURA 5.2 – RESULTADO ESTACIONÁRIO COM A MALHA INICIAL. 55
FIGURA 5.3 – 1ª ADAPTAÇÃO E RESULTADOS DO CASO ESTACIONÁRIO. 55
FIGURA 5.4 – RESULTADO APÓS 2ª E ÚLTIMA ADAPTAÇÃO. 55
FIGURA 5.5 – ESTUDO TRANSIENTE DO AVANÇO DA SATURAÇÃO DE ÁGUA COM ADAPTAÇÃO POR REDEFINIÇÃO LOCAL. 58
FIGURA 5.6 – EVOLUÇÃO DO ERRO GLOBAL COM E SEM SINGULARIDADES PARA O CAMPO DE PRESSÃO. 58

FIGURA 5.7 – EVOLUÇÃO DO ERRO GLOBAL COM E SEM SINGULARIDADES PARA O CAMPO DE SATURAÇÃO DE ÁGUA. 59
FIGURA 5.8 – COMPARAÇÃO DO TEMPO NECESSÁRIO PARA A CONCLUSÃO DA SIMULAÇÃO COM E SEM ADAPTAÇÃO AUTOMÁTICA..... 59
FIGURA 5.9 – AVANÇO DA SATURAÇÃO DE ÁGUA DE DIFERENTES ESTRATÉGIAS DE INTERPOLAÇÃO PVI 0,09.....60
FIGURA 5.10 – AMPLIAÇÃO DA FRENTE DE SATURAÇÃO (FIG. 5.9) PARA DIFERENTES TÉCNICAS DE INTERPOLAÇÃO PVI 0,09. 60
FIGURA 5.11 – CURVAS DE SATURAÇÃO PARA O CASO HOMOGÊNEO ISOTRÓPICO PVI 0,08.....61
FIGURA 5.12 – AMPLIAÇÃO DA FIG. 5.11..... 61
FIGURA 5.13 – TEMPO PARA CONCLUSÃO DA ANÁLISE. 61
FIGURA 5.14 – TAMANHOS LOCAIS DOS ELEMENTOS DA MALHA ADAPTADA – CL´S. 62
FIGURA 5.15 – A) CURVAS DE SATURAÇÃO DE ÁGUA PARA APROXIMAÇÕES DE ALTA E BAIXA ORDEM COM APROXIMAÇÕES DE ALTA E BAIXA ORDEM PVI 0,08 B) AMPLIAÇÃO NA REGIÃO DE INTERFACÓLEO-ÁGUA. 63
FIGURA 5.16 – TAXA DE PRODUÇÃO DE ÓLEO FORMULAÇÕES DE ALTA ORDEM E BAIXA ORDEM. 63
FIGURA 5.17 – AMPLIAÇÃO DA FIG. 5.16 ANTECIPAÇÃO DO "BREAKTHROUGH" NA BAIXA ORDEM. 64
FIGURA 5.18 – PRODUÇÃO TOTAL DE ÓLEO ACUMULADO FORMULAÇÕES DE ALTA E BAIXA ORDEM.....64
FIGURA 5.19 – QUANTIDADE DE ELEMENTOS A CADA ADAPTAÇÃO EM BAIXA E ALTA ORDEM.....64
FIGURA 5.20 – A) GEOMETRIA DO PROBLEMA HETEROGÊNEO ISOTRÓPICO B) MALHA INICIAL 65
FIGURA 5.21 – AVANÇO DA SATURAÇÃO DE ÁGUA A) PVI 0,2 B) PVI 0,3 66
FIGURA 5.22 – A) AVANÇO DA SATURAÇÃO DE ÁGUA PVI 0,45 B) MALHA ADAPTADA AOS RESULTADOS NO PVI 0,45 66
FIGURA 5.23 – PROBLEMA HETEROGÊNEO E ANISOTRÓPICO.....67
FIGURA 5.24 – MALHA INICIAL TRIANGULAR 67
FIGURA 5.25 – CAMPO DE RESULTADOS DE PRESSÃO.....68
FIGURA 5.26 – A) 1ª ADAPTAÇÃO PELO CRITÉRIO DA HESSIANA DA PRESSÃO B) 1ª ADAPTAÇÃO PELOS GRADIENTES DO FLUXO 69
FIGURA 5.27 – COMPARAÇÃO DOS TEMPOS DESPENDIDOS NA SIMULAÇÃO COM CRITÉRIOS DE ADAPTAÇÃO PELA HESSIANA DA PRESSÃO E PELOS GRADIENTES DE FLUXO 69
FIGURA 5.28 – RESULTADOS PARA A SATURAÇÃO DE ÁGUA PVI 0,01 A) ADAPTAÇÃO PELA HESSIANA DA PRESSÃO B) ADAPTAÇÃO PELOS GRADIENTES DO FLUXO.....70
FIGURA 5.29 – RESULTADOS PARA A SATURAÇÃO DE ÁGUA PVI 00,03 A) RESULTADOS DE SATURAÇÃO DE ÁGUA COM ADAPTAÇÃO PELOS GRADIENTES DO FLUXO B) MALHA ADAPTADA RESULTANTE 70
FIGURA 5.30 – RESULTADOS PARA A SATURAÇÃO DE ÁGUA PVI 0,08 A) RESULTADOS DE SATURAÇÃO DE ÁGUA COM ADAPTAÇÃO PELOS GRADIENTES DO FLUXO B) MALHA ADAPTADA RESULTANTE.....70

LISTA DE TABELAS

TABELA 4.1 – COMPARAÇÃO DE DESEMPENHO DOS ALGORITMOS.....51
TABELA 5.1 – PARÂMETROS DAS SIMULAÇÕES ADAPTATIVAS DO CASO ESTACIONÁRIO. 54
TABELA 5.2 – RESULTADOS DO CASO ESTACIONÁRIO. 56
TABELA 5.3 – RESULTADOS DA SIMULAÇÃO DO CASO DE REFERÊNCIA BIFÁSICO EM MEIO HOMOGÊNEO E ISOTRÓPICO. 56
TABELA 5.4 – PARÂMETROS DA SIMULAÇÃO DO CASO ADAPTATIVO HOMOGÊNEO E ISOTRÓPICO. 57
TABELA 5.5 – PARÂMETROS DA SIMULAÇÃO USADOS EM TODAS AS ANÁLISES. 57
TABELA 5.6 – PARÂMETROS PARA AS SIMULAÇÕES COM FORMULAÇÕES DE BAIXA E ALTA ORDEM. 62
TABELA 5.7 – PARÂMETROS PARA A SIMULAÇÃO HETEROGÊNEA ISOTRÓPICA.....65

LISTA DE SÍMBOLOS

$\vec{v}$ –Vetor velocidade aparente do escoamento.

$\underline{K}$– Tensor permeabilidade do domínio.

μ –Viscosidade do fluido.

p – Pressão.

ρ – Massa específica do fluido.

g – Aceleração gravitacional.

z – Altura da coluna de fluido.

λ – Mobilidade de um fluido.

S – Saturação de uma fase fluida.

φ – Porosidade do reservatório

Ω – Domínio discretizado.

Γ – Contorno do domínio discretizado.

LISTA DE ACRÔNIMOS E SIGLAS

EDP – Equação diferencial parcial
MVF – Método dos volumes finitos
EBFV – *Edge based finite volumn*
IMPES – *Implicit pressure explicit saturation*
CL– *Characteristic Length*
GUI – *Graphics User Interface*
CAD – *Computer Aided Design*
BFS – *Breadth-First Search*
FMDB – *Flexible Distributed Mesh Database*
TOA – Termos de Ordem Alta

SUMÁRIO

1 Introdução 16

1.1 Motivação 16

1.2 Objetivo....17

1.3 Organização....17

2 Modelos Matemáticos e Físicos 19

2.1 Introdução....19

2.2 Equação de Pressão....20

2.3 Equação de Saturação 20

2.4 Condições de contorno 20

3 Estratégias de Adaptação de Malhas 23

3.1 Refinamento adaptativo....23

3.2 Estratégia para problemas no regime estacionário....23

3.3 Estratégia para problemas no regime transiente 24

3.4 Processo geral de simulação com adaptação 25

3.5 O Simulador MVF 26

3.6 O Indicador de erros....27

3.7 Definição dos parâmetros da malha para adaptação via redefinição de malhas 30

3.8 Gerador de malhas gmsh 31

3.8.1 Criação de Geometrias 31

3.8.1.1 Utilização do Gmsh através de interface gráfica 32

3.8.1.2 Utilização do Gmsh através de script 32

3.8.2 Geração da malha usando o Gmsh 33

3.8.3 Geração de malhas usando o JMesh 34

3.9 O interpolador 35

3.9.1 O método linear de interpolação 36

3.9.2 O método quadrático de interpolação 37

3.9.3 O método de interpolação adaptativo 38

4 Adaptação automática de malhas através de redefinição local 40

4.1 Etapas seguidas pelo adaptador 40

4.2 Recebimento da lista de elementos a remover 40

4.3 Tratamento das regiões a ser adaptadas 41

4.4 Geração do arquivo .pos (malha de background) 43

4.5 Identificação das regiões de adaptação local 44

4.6 Criação da geometria necessária à geração da nova malha 48

4.7 Geração da nova malha local. 50

4.8 União das malhas – “merge” 51

5 Resultados 53

5.1 Introdução 53

5.2 Análise do campo de pressão em um escoamento estacionário, monofásico em meio homogêneo e isotrópico. 53

5.3 Comportamento do campo de saturação em um escoamento bifásico em meio homogêneo isotrópico 56

5.4 Simulação de escoamento em meio heterogêneo isotrópico 65

5.5 Simulação de escoamento em meio heterogêneo anisotrópico 67

6 Conclusões 72

6.1 Trabalhos futuros 72

REFERÊNCIAS 74

APÊNDICE 77

A.1 Aspectos relevantes do C++ 77

A.2 O FMDB 79

1 Introdução

1.1 Motivação

Métodos que utilizam adaptação de malhas podem ser aplicados em uma variedade de áreas da física e engenharia, tais como dinâmica dos fluidos, combustão, transferência de calor, resistência dos materiais, etc. Os fenômenos físicos nessas áreas desenvolvem uma dinâmica natural, em geral dependente do tempo, ocorrendo mudanças rápidas na solução ou na sua derivada em certas regiões do domínio. Como exemplos podem-se citar ondas de choque, zonas de reação em processos de combustão. Seja em casos estacionários ou transientes é necessário um procedimento de análise de erro para avaliar a acurácia dos resultados e se necessário o grau e local a ser refinada a malha.

A área de aplicação deste trabalho refere-se ao estudo do escoamento óleo-água em reservatórios de petróleo. A capacidade de simular o escoamento deste fenômeno físico é de interesse para as empresas exploradoras.

A partir do equacionamento dos fenômenos e da modelagem e simulação numérica é possível estudar o comportamento do reservatório para auxiliar na definição de uma estratégia otimizada para a injeção de água aumentando o a recuperação total final de óleo.

Temos então um problema de escoamento bifásico em um meio poroso (rocha reservatório), caracterizado por ser um fenômeno transiente. Nesses casos, para que se consiga capturar adequadamente o fenômeno, a malha deve conter uma quantidade significativa de pontos nas regiões onde ele ocorre. Tendo em vista a dimensão de uma rocha reservatório, dos custos envolvidos no processo de perfuração, da quantidade de dados a considerar, geológicos, econômicos, tecnológicos, fica evidente a importância do aprimoramento de ferramentas computacionais para a análise numérica do escoamento em reservatórios de petróleo.

Fazer com que todo o domínio tenha uma grande quantidade de pontos implicaria num esforço computacional muito alto para a solução do problema. Uma alternativa é o uso de técnicas adaptativas que oferecem certo grau de robustez, eficiência e confiança.

Um refinamento computacionalmente eficiente é fundamental para a solução de equações diferenciais devido à quantidade de variáveis envolvidas. Objetivando eficiência computacional, surgiu a idéia de refinamento das malhas de elementos finitos através da análise do erro de aproximação (ZIENKIEWICZ e TAYLOR, 2000).

A partir dos resultados obtidos da análise de erros e no intuito de manter o erro estimado dentro de uma tolerância estabelecida aceitável, dá-se início aos procedimentos de adaptação automática. As técnicas de adaptação utilizam varias estratégias tais como (ARAÚJO, 2004) reposicionamento dos nós, alteração da ordem das funções de aproximação, subdivisão ou agrupamento dos elementos da malha, redefinição global ou local de malhas através de um gerador de malhas, combinação dessas estratégias.

Dentre estes tipos se destaca a técnica da redefinição de malhas. Devido a sua alta capacidade de modificar a malha em um único ciclo de adaptação. Essa estratégia aliada ao uso de malhas triangulares não estruturadas fornece uma ferramenta poderosa de adaptação e controle do erro da simulação.

1.2 Objetivo

O objetivo deste trabalho é o desenvolvimento de um algoritmo eficiente de remoção de elementos para psoterior refinamento através de um gerador de malhas não estruturadas, triangulares, em casos bidimensionais para ser aplicado em simulação numérica de escoamento de fluidos em reservatórios de petróleo.

O procedimento adaptativo desenvolvido é denominado de redefinição local ou "remeshing local" de malhas não estruturadas e triangulares controlado por uma ferramenta de análise de erros a posteriori baseada no cálculo da hessiana da solução que fornece os parâmetros necessários à adaptação local da malha.

O programa supracitado deve realizar a integração de três módulos, a ferramenta de análise de erro, o gerador de malhas e o interpolador dentro de um processo de simulação com adaptação.

A fim de testar as ferramentas desenvolvidas, são realizadas simulações de escoamento de fluidos em meios porosos em caso horizontal e bidimensional de um modelo de injeção em malha conhecido como "five spot" onde as variáveis de interesse são pressão e saturação.

Os casos de escoamento estudados são casos modelo da engenharia de reservatórios de petróleo, considerando escoamento bifásico, bidimensional de fluidos imiscíveis e incompressíveis em meios porosos.

1.3 Organização

Após este primeiro capítulo introdutório temos outros cinco capítulos.

O segundo capítulo traz uma descrição dos modelos físico e matemático dos casos analisados, sendo usado o método dos volumes finitos com estrutura de dados por aresta para a obtenção das equações algébricas discretas. Nele são vistas as hipóteses simplificadoras consideradas para a classe de aplicações estudadas e são apresentadas as equações diferenciais que representam o modelo do fenômeno físico.

No terceiro capítulo inicia-se a análise computacional e uma visão global acerca das ferramentas de simulação, análise de erros, adaptação, geração de malha e interpolação utilizadas. Neste capitulo serão abordados os aspectos relacionados aos métodos utilizados em cada etapa do ciclo do métodos de acoplamento das equações aos critérios de adaptação e análise de erros, bem como estratégias de interpolação.

O quarto capítulo aborda o algoritmo do módulo adaptador, sua sequência de operações, funções principais internas e no macro-algoritmo, forma de adaptação e controle dos parâmetros para a geração da nova malha adaptada e como o adaptador se encaixa no processo global de uma forma um pouco mais detalhada.

O quinto capítulo mostra os resultados dos problemas estudados, mostra aplicações do software desenvolvido e compara os resultados com problemas modelo de engenharia de petróleo. Exploramos casos de escoamento monofásico e bifásico, homogêneo e heterogêneo, isotrópicos e anisotrópicos. Verificamos diferentes técnicas de interpolação, níveis de tolerância, análise do tempo decorrido e do ganho de eficiência, observamos a acurácia das soluções adaptadas em comparação com casos de referência sem adaptação. Procedemos a simulação de casos modelo para testar e comprovar a robustez da ferramenta desenvolvida.

O sexto capítulo apresenta algumas conclusões e sugestões de trabalhos futuros dentro desta linha de pesquisa.

2 Modelos Matemáticos e Físicos

2.1 Introdução

Nesta seção apresentamos a formulação matemática que descreve o escoamento bifásico e incompressível de óleo e água num reservatório de petróleo não deformável. A equação de conservação da massa é convenientemente manipulada junto com a velocidade de Darcy de modo a produzir uma formulação segregada com uma equação elíptica de pressão e uma equação hiperbólica de saturação das fases.

De modo a utilizar as equações apresentadas ao longo deste capítulo, é necessária a adoção de algumas hipóteses simplificadoras (CARVALHO, 2005):

• Meio poroso saturado;
• Rochas e fluidos incompressíveis;
• Fluidos imiscíveis;
• Temperatura constante - escoamento isotérmico;
• Lei de Darcy válida para as velocidades consideradas.

Inicialmente, podemos escrever a conservação da massa para uma fase *i (i=o (óleo), w (água))*, como:

$$\frac{\partial(\emptyset\rho_i S_i)}{\partial t} = -\vec{\nabla}.(\rho_i \vec{V}_i) + q_i \tag{2.1}$$

onde $\vec{v}_i$ e ρ_i, representam, respectivamente, a velocidade de Darcy e a densidade da fase *i*, e ϕ e q_i são a porosidade, i.e. a fração da rocha que pode ser ocupada por fluidos (HURTADO, 2005), e os termos de fonte (ou sumidouro), que representam poços de injeção ou produção. S_i é a saturação da fase *i*, ou seja, a fração do volume poroso ocupado pela fase onde o volume poroso é considerado totalmente saturado. Dessa última definição, segue a relação de restrição das saturações:

$$S_o + S_w = 1 \tag{2.2}$$

Quando duas ou mais fases estão presentes num mesmo meio poroso, a presença de uma fase dificulta o escoamento da outra. Neste caso, onde coexiste mais de uma fase, a definição da permeabilidade relativa torna-se necessária. A permeabilidade relativa pode variar entre zero e um. De forma geral, quanto maior a saturação de uma das fases, a permeabilidade relativa respectiva tenderá a ter maior valor assim a equação da velocidade de Darcy deverá ser escrita considerando a permeabilidade relativa para a fase i. Desprezando os efeitos capilares e gravitacionais, esta equação é escrita (PEREIRA, 2009; ARAÚJO, 2004):

$$\vec{V}_i = -\frac{K_{ri}}{\mu_i}\underline{K}\vec{\nabla}\rho_i \tag{2.3}$$

onde p_i , k_{ri} e μ_i representam, respectivamente, a pressão, a permeabilidade relativa e a viscosidade da fase *i*, e $\underline{K}$ é o tensor de permeabilidade absoluta da rocha, que, em duas dimensões, é representado por uma matriz *2x2*.

2.2 Equação de Pressão

Substituindo a velocidade de Darcy apresentada na Eq. (3) na lei de conservação da massa, Eq. (1) e após alguma manipulação algébrica, escrevemos a equação da pressão como:

$$\vec{\nabla}.\vec{V} = Q \tag{2.4}$$

onde:

$$\vec{V} = -\lambda_T \underline{K} \vec{\nabla} P \tag{2.5}$$

onde $\lambda_T = \lambda_O + \lambda_w$ e $Q = Q_w + Q_O$ são, respectivamente, a mobilidade total e a vazão específica total.

2.3 Equação de Saturação

Novamente, fazendo uso das leis de conservação da massa, Eq. (2.1), e de Darcy, Eq. (2.4), utilizando ainda a equação de restrição das saturações, Eq. (2.2), e as hipóteses assumidas, e novamente, após alguma manipulação algébrica, podemos escrever a equação de saturação da água, como:

$$\emptyset \frac{\partial S_w}{\partial t} = -\vec{\nabla}.\left(f_w \vec{V}\right) + Q_w \tag{2.6}$$

onde f_w é a função de fluxo fracionário e Q_w é a vazão específica de água. A equação (2.6) é não linear e de natureza hiperbólica (EWING, 1983; CARVALHO, 2005).

2.4 Condições de contorno

As equações de pressão e de saturação apresentadas definem um problema de valor inicial e de contorno cuja solução exige uma descrição do que ocorre nas fronteiras e como o reservatório se apresenta no início da simulação.

A partir das condições especificadas para um determinado instante de tempo, os cálculos de pressão para as células da malha de simulação são feitos através da marcha em passos de tempo, ou "*timesteps"*. Para os problemas de simulação de reservatórios, as condições iniciais necessárias são a pressão inicial do reservatório e a distribuição de saturações.

As condições de contorno usadas em simulação de reservatórios podem ser de diferentes naturezas. Isto inclui as fronteiras externas (limites dos reservatório) e as fronteiras internas (poços de produção e de injeção).

As condições aplicadas a poços produtores e injetores, sendo esses poços tratados como "fronteiras internas" do reservatório são de três tipos:

- Primeiro tipo – Dirichlet – Pressão especificada nas fronteiras ou poços
- Segundo tipo – Neumman – fluxo nas fronteiras – representa a produção de um poço ou influência de um aquífero nos limites do reservatório
- Terceiro tipo – Cauchy – combinação dos dois anteriores e pode acontecer quando as fronteiras de dois reservatórios coincidirem

Para as fronteiras externas é assumido que não há fluxo atravessando-as. Matematicamente esta condição de contorno é representada por:

$$\vec{V} \cdot \vec{n}\big|_{\Gamma} = 0 \qquad (2.9)$$

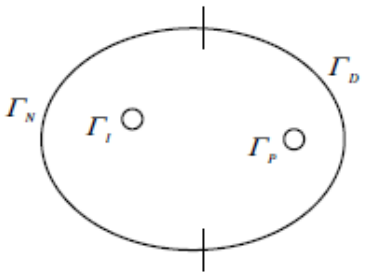
Onde n representa a direção normal à fronteira do reservatório. Em termos das equações discretizadas, esta condição pode ser implementada especificando-se transmissibilidades nulas nas faces externas das células situadas nos limites dos reservatórios. As condições de contorno que impõem gradientes de pressão nas fronteiras do reservatório são conhecidas como condições do tipo Neumann.

Como nos mostra a lei de Darcy, para as fronteiras internas, especificando-se um gradiente de pressão especifica-se por consequência uma vazão, haja vista as outras variáveis da equação serem conhecidas.

Para que possamos tornar o problema representado pelas equações de pressão e saturação, completamente determinado é necessário que utilizemos um conjunto apropriado de condições iniciais e de contorno. Dependendo das características geológicas das vizinhanças do reservatório, bem como, das condições de produção impostas a ele, várias configurações podem ser obtidas.

A seguir apresentaremos um modelo de condições iniciais e de contorno associadas a um reservatório com poços injetores (*I*) e produtores (*P*). Estes poços são tratados como "fronteiras internas" dos reservatórios e as condições aplicadas a estes poços são chamadas de condições de "contorno internas" (CARVALHO, 2005).

Figura 2.1 - Modelo esquemático de um reservatório e suas respectivas fronteiras.



Fonte: Carvalho, 2005.

Assim:

Γ_n – Fronteiras externas com fluxo prescrito;
Γ_D – Fornteiras externas com pressão prescrita;
Γ_I – Poços injetores;
Γ_P – Poços produtores

3 Estratégias de Adaptação de Malhas

3.1 Refinamento adaptativo

Uma maneira de obter uma solução melhor do ponto de vista da acurácia da solução e da eficiência computacional, é gerar uma malha nova através de um processo adaptativo, o que pode ser feito por diversas técnicas. Especificamente, para o método da redefinição de malha, a técnica consiste em criar uma malha nova local ou globalmente. Para esse fim existem algoritmos que permitem gerar malhas a partir de uma distribuição de tamanhos característicos dos elementos. Estes algoritmos geram malhas não estruturadas de elementos triangulares ou quadrangulares em 2D e tetraédricos em 3D. Utilizando um algoritmo deste tipo é possível gerar uma malha adaptada à solução e à precisão pretendida (BARREIROS J. O. 1996). Alguns exemplos:

Suhara e Fukuda (1972) utilizaram em refinamento adaptativo um algoritmo que permite gerar malhas e triângulos com densidades diferentes em diferentes subdomínios de um domínio bidimensional.

Lohner R. e Parikh P. (1988) desenvolveram um algoritmo que permite gerar malhas bidimensionais de elementos triangulares com uma distribuição de tamanhos arbitrária, muito utilizado em refinamento adaptativo que se baseia na técnica de avanço frontal.

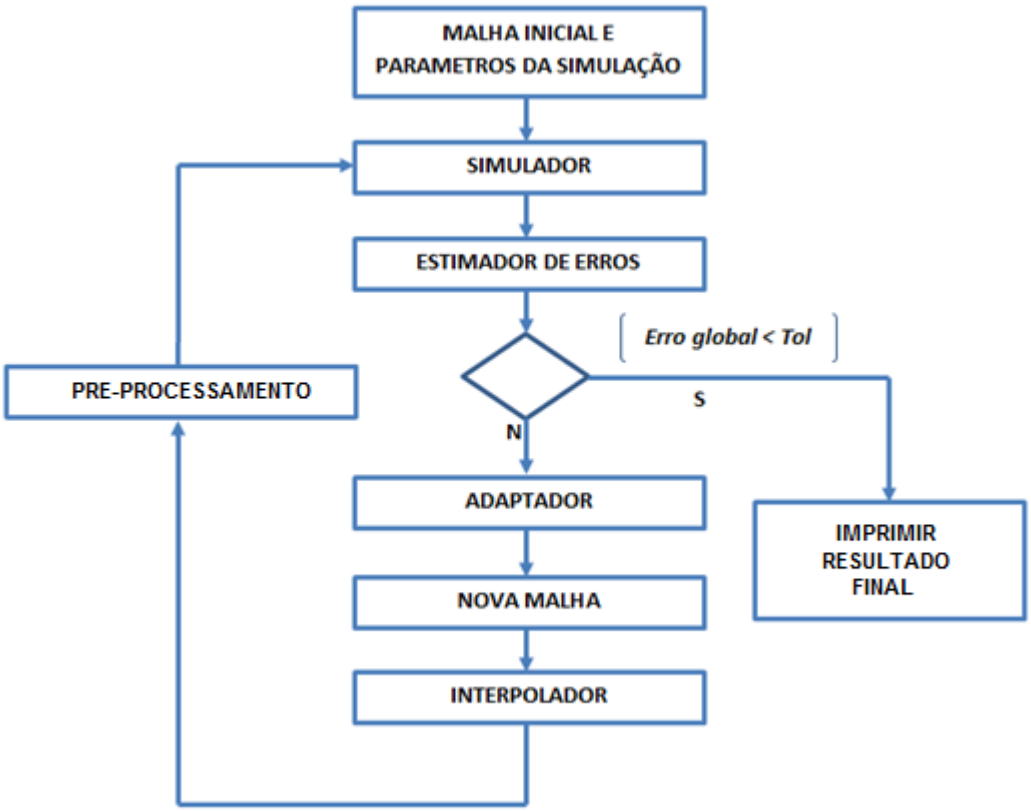
Popiolek T. L. e Santos M. A. publicaram em 2001 um trabalho onde a adaptação de malhas é feita sob um critério estatístico que identifica as sub-regiões da malha que devem ser refinadas. Em 2008 Cezano M. T. e Carvalho D. K. apresentaram todo um módulo de adaptação de malhas utilizando estratégia de análse de erros “a posteriori” para fenômenos multifásicos em meios porosos.

3.2 Estratégia para problemas no regime estacionário

Problemas no regime estacionário são aqueles em que os resultados de interesse da simulação não se alteram com o passar do tempo. Nesses casos não cabe falar em passos no tempo e a análise de erros só precisa ser realizada após o término da simulação numérica, consequentemente, o critério de parada deste algoritmo está unicamente relacionado ao resultado da análise de erros obtidos pelo estimador de erro. Caso o erro global não esteja reduzido a um valor abaixo da tolerância pré estabelecida a malha é adaptada e refeita a análise numérica até que seja obtida a acurácia exigida pelo usuário.

A estratégia para o processo global de adaptação adotada (ARAÚJO, 2004; SILVA E. 2008) para a solução de problemas no regime estacionário é a apresentada na figura 3.1.

Figura 3.1 – Fluxograma geral do processo de simulação com adaptação no regime estacionário.

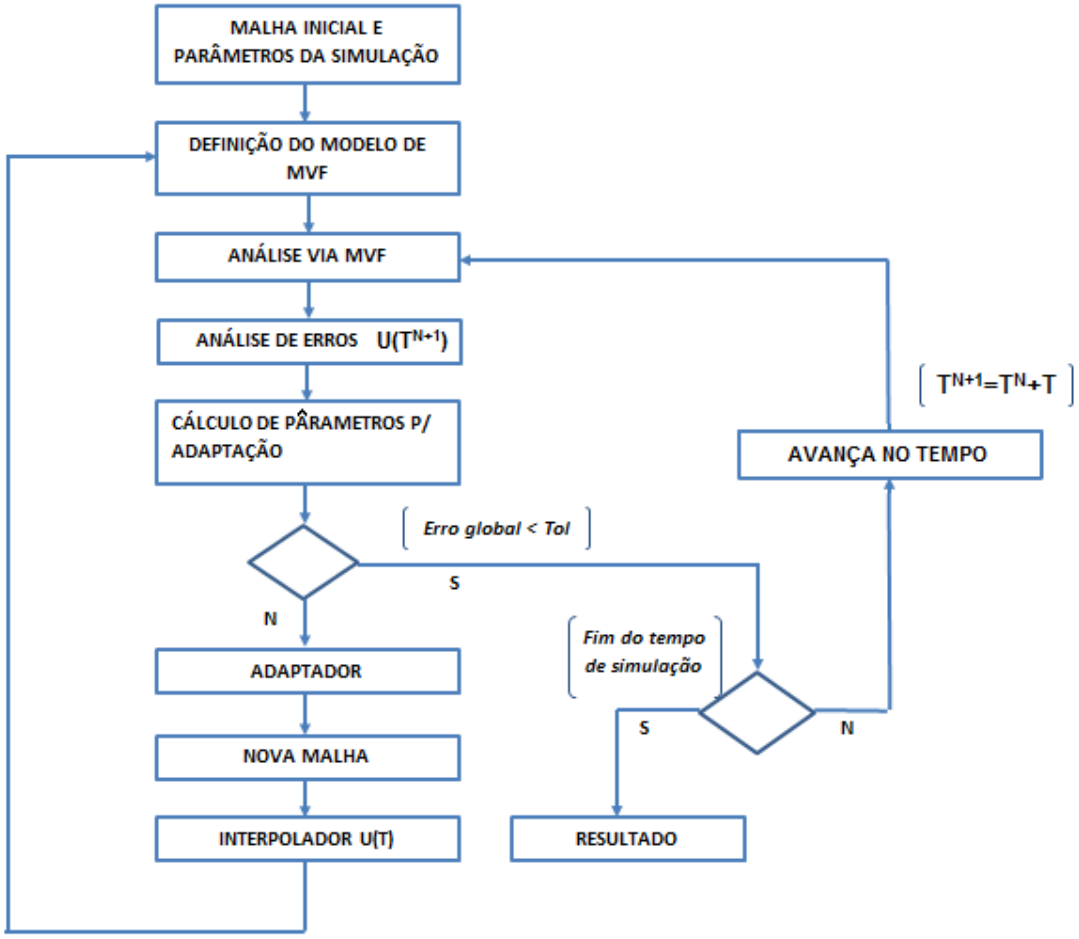


Fonte: O autor.

3.3 Estratégia para problemas no regime transiente

Os problemas de regime transiente envolvem a variação temporal das grandezas físicas de interesse, isto é, a partir dos valores iniciais desta grandeza, se procede a análise de erro, adaptação da malha, definição da grandeza dos passos de tempo baseado em critérios de estabilidade e cálculos sucessivos dos resultados de interesse com constante monitoração pela análise de erros a cada passo no tempo. A estratégia para problemas no regime transiente está esquematizada no fluxograma da figura 3.2.

Figura 3.2 – Fluxograma geral do processo de simulação com adaptação no regime transiente.



Fonte: O autor.

3.4 Processo geral de simulação com adaptação

Para elaboração do presente estudo houve a necessidade de desenvolver um software para execução da adaptação local. O programa funciona dentro do seguinte contexto:

1 Os resultados da simulação seguem para a análise do estimador de erros que possui critérios pré-estabelecidos de tolerância máxima para todo o domínio e excluindo singularidades (regiões onde a análise de erro indica um grau de refinamento que inviabiliza a simulação).
2 Feita a análise do estimador de erros, caso excedida a tolerância do erro, o estimador de erros aciona o adaptador e indica os elementos que deverão sofrer modificação e a dimensão que os novos elementos deverão ter.
3 O adaptador procede a remoção dos elementos da malha, redefinição da geometria e o controle do refinamento local e reconstrução das malhas das regiões indicadas da malha concluindo o processo de "remeshing" local.
4 A malha anterior com a solução e a nova malha adaptada seguem para o interpolador de dados entre malhas que interpolará os valores dos campos de interesse (pressão e saturação de água neste trabalho) para a nova malha e retornará ao simulador que prossegue com os cálculos.

3.5 O Simulador MVF

O problema físico de interesse é descrito admitindo-se a hipótese do contínuo e, portanto, para que o mesmo possa ser resolvido em um computador é necessário que as equações diferenciais que o governam sejam discretizadas e transformadas em um conjunto finito de equações algébricas. Para tal se faz necessário primeiramente a discretização espacial e temporal do problema.

Quando este conjunto de equações é resolvido empregando um método numérico, apenas é possível obter uma solução aproximada. Tal solução consiste em um conjunto finito de valores das variáveis dependentes, correspondendo a um conjunto discreto de localizações no espaço e no tempo. A distribuição de tais localizações no domínio de solução está definida pela malha computacional, a qual simplesmente é uma representação discreta da geometria do domínio (HURTADO, 2005). Daí a importância de realizar o controle do refinamento nas regiões de interesse afim de capturar/prever fenômenos físicos que possam vir a ocorrer.

Observamos do fluxograma geral que uma malha é fornecida ao simulador; esta malha discretiza o domínio da simulação e contém uma parcela das informações necessárias para iniciar o processo de simulação. São elas a geometria dos domínios estudados, a resolução inicial definida pela posição e quantidade de elementos, coordenadas dos nós e matriz de conectividades. No processo de adaptação modifica-se a densidade da malha porém conserva-se a geometria do domínio, os limites internos e externos dos domínios.

O MVF utiliza como ponto de partida a forma integral da equação da conservação. O domínio de solução é dividido num número finito de volumes de controle contíguos, e a equação da conservação é aplicada a cada volume de controle. No centróide de cada volume localiza-se um nó computacional originário da malha, para o qual são calculados os valores das variáveis, sendo os valores das variáveis nas superfícies dos volumes de controle obtidos por interpolação em função dos valores nodais (centros dos volumes). Os integrais de volume e de superfície são aproximados usando fórmulas de quadratura apropriadas. Como resultado, obtém-se uma equação algébrica para cada VC, na qual aparecem os valores das variáveis no nó em causa e nos nós vizinhos. O MVF pode ser aplicado a qualquer tipo de malha, por isso adapta-se a geometrias complexas. A malha define apenas as fronteiras do volume de controle e não necessita estar relacionada com um sistema de coordenadas. O método é inerentemente conservativo, contando que os integrais de superfície (que representam fluxos convectivos e difusivos) sejam os mesmos em faces partilhadas por VC.

A aproximação com o MVF é talvez a de compreensão mais simples, pois todos os termos que precisam de ser aproximados têm significado físico, razão pela qual é popular entre engenheiros (FERREIRA, 2007). O método dos volumes finitos adotado foi implementado para lidar com malhas não estruturadas, o que permite a modelagem de problemas envolvendo geometrias complexas, assim como o uso de adaptação local de malhas. O simulador adotado utiliza uma formulação de volumes finitos (MVF) baseado em uma estrutura de dados por aresta e centrada nos nós (EBFV). A formulação foi desenvolvida de forma a permitir associar-se a cada sub-região contendo elementos da malha uma propriedade física do meio poroso, permitindo tratar problemas que apresentam variação abrupta de permeabilidade e porosidade, comuns em simulação de escoamento de fluidos em reservatórios de petróleo devido à própria natureza do processo de formação geológica das rochas sedimentares (CARVALHO, 2005).

No uso de formulação com estrutura de dados por arestas verifica-se uma economia significativa de memória e de operações de ponto flutuante por evitar redundância de armazenamento e distribuição de dados (SILVA R., 2008), particularmente no caso tridimensional (CARVALHO, 2005).

O Método de resolução das equações é do tipo IMPES – Implicit Pressure, Explicit Saturation, um dos primeiros trabalhos publicados nesta direção por Fagin e Stewart (1966). Este é um método de resolução das equações para as variáveis de pressão e saturação, sendo a pressão resolvida de forma implícita e a saturação de forma explícita. O acoplamento entre essas equações é feito pela velocidade de Darcy. Sua maior limitação é a restrição do passo de tempo devido à avaliação explícita das saturações (CORDAZZO, 2006). Sua principal vantagem é a simplicidade de implementação mas somase a isso a baixa demanda computacional, produzindo bons resultados para um conjunto de problemas de fluxo bifásico e incompressível.

3.6 O Indicador de erros

Desde as primeiras pesquisas que demonstraram que os modelos matemáticos de problemas em engenharia poderiam ser resolvidos de forma aproximada com a ajuda de métodos numéricos, surgiu a dúvida quanto ao grau de precisão de tais soluções A diferença entre o resultado exato e o resultado aproximado por um método numérico é o erro de truncamento. Além dele existem outras fontes de erro provenientes da modelagem matemática, processo de interpolação, modelagem geométrica, etc. Enfim, o grande desafio é saber quantificar, controlar e até mesmo minimizar este erro.

O primeiro passo antes de aprofundar-se na discussão a respeito desse tópico é definir de maneira clara o que se entende neste trabalho por erro de aproximação. O conceito de erro é a diferença entre a resposta dita exata e uma solução aproximada oriunda de um procedimento numérico (MARQUES R. 2011):

$$E = u - \hat{u} \tag{3.1}$$

onde û representa uma solução obtida através de um procedimento numérico e u representa a solução exata.

Infelizmente a determinação do erro como colocado na eq. (3.1) de maneira geral não é possível, pois, na grande maioria dos problemas, não se conhece a solução exata. Dessa forma é necessário especular uma maneira de ao menos ter uma idéia da medida desse erro. Os estimadores de erro surgiram justamente para preencher essa lacuna.

A ideia básica dos estimadores de erro é calcular um valor estimado para a solução exata do problema, isto é, sem erro de discretização. A diferença entre o valor estimado para a solução exata e o resultado numérico obtido representa o erro aproximado (MARCHI, 1999).

Essa resposta gerada, muitas vezes chamada de suavizada ou recuperada é então usada para gerar a estimativa de erro. Logo a equação 3.1 passa a ser escrita para o erro estimado como (ZIENKIEWICZ e TAYLOR, 2000):

$$\hat{E} = u^{*} - \hat{u} \tag{3.2}$$

onde a solução estimada u^* é obtida com o estimador.

Muitos pesquisadores empregaram esforços em diversas estratégias de adaptação de malhas usando como critério a indicação ou estimativa de erros foram elaboradas ao longo dos anos (ZIENKIEWICZ e TAYLOR, 2000; POPIOLEK T. L. e SANTOS M., 2001; BOUSSETTA R, 2006; P. COOREVITS e BELLENGER, 2004), e a partir dos resultados obtidos por essas pesquisas se desenvolveram as estratégias de adaptação de malhas.

A ferramenta de análise de erros é a que controla todo o processo adaptativo; ela é quem indica as regiões a serem refinadas ou desrefinadas, calcula o novo tamanho de elemento em cada local, repassando através da malha de "background" e monitora a evolução do erro global dando início ao processo adaptativo quando necessário. Primeiramente, esta ferramenta avalia a qualidade da solução. Se após a verificação da acurácia da aproximação, constatar-se uma baixa resolução, i.e., abaixo daquela requerida pelo usuário, a própria ferramenta de estimativa de erros, através de procedimentos numéricos baseados no estudo da convergência da solução, fornece os parâmetros para a formação da nova malha (LYRA, 1994).

O campo de tamanhos é definido por pontos, assim se a malha for muito grossa (poucos pontos para efetuar o controle) o resultado nao terá a acurácia necessária. Se por outro lado a taxa de refinamento de alguns elementos for muito acentuada pode resultar em inviabilidade do avanço da simulação pelo alto custo computacional. Uma malha intermediária deve ser criada. Esta situação é levada em conta a cada passo de adaptação.

R. Boussetta (2006) trata este problema com um algoritmo que redefine a tolerância especificada se a razão entre os novos tamanhos de elementos ($h_{new)}$ e os tamanhos antigos (h_{old}) exceder limites pré estabelecidos.

Um algoritmo similar é proposto por Coorevits (2004) onde é prescrita a memória computacional disponível e o tempo de análise.

Em nossa estratégia também são esses os parâmetros limitantes. Definimos o critério para o simulador definir como elementos singulares os que, no cálculo do novo tamanho, exigirem um grau de refinamento acima do valor limite a ser definido pelo usuário em arquivo de inicialização. Dessa forma se o estimador de erros indica refinamento para um novo tamanho menor que o especificado o elemento é classificado como singularidade. Exemplos de regiões que frequentemente são marcados como singularidades devido aos altos gradientes da variável em estudo são os arredores dos poços nas análises de pressão e a interface óleo-água em análises de saturação. Como estas regiões vão requerer níveis de refinamento proporcionais aos gradientes em estudo, capazes de inviabilizar o avanço da análise, é necessário limitar de certa forma esse refinamento da malha.

Assim, no presente trabalho, adotamos duas tolerâncias diferentes para os erros relativos. A primeira computa os erros de todos os elementos da malha (Tol_1). Visando tratar problemas com singularidades na solução, como no caso dos poços injetores e produtores, adotamos uma segunda tolerância (Tol_2), retirando-se dos cálculos todos os elementos que atingem dimensões abaixo de valores mínimos pré-estabelecidos. Notadamente, como o cálculo com a Tol_1 leva em conta elementos com singularidades é preciso fornecer um maior valor de tolerância, que proporcione menor refinamento da malha para esse caso.

Se uma das duas percentagens de erro estiver acima da respectiva tolerância, o programa de análise de erros define os parâmetros que conduzirão à construção da malha adaptada. Estes parâmetros são os tamanhos desejados para os elementos da nova malha.

Desta forma, o indicador produz para o processo de adaptação uma distribuição de comprimentos característicos (CL´s), para os elementos da malha adaptada, que conduzam a um erro relativo menor ou igual às tolerâncias predefinidas.

Podemos obter os valores do erro em uma aresta pela equação (Silva, E. et al, 2008):

$$E = \left|\left(\nabla\hat{u}_j - \nabla\hat{u}_i\right)\vec{L}_{ij}\right| \tag{3.3}$$

onde:

E - erro na aresta formada pelos pontos i e j;

$\nabla\hat{u}_j$ - gradiente aproximado no nó j;

$\vec{L}_{ij}$ - o vetor que define a aresta.

O erro associado ao elemento triangular é a média dos produtos escalares das diferenças de solução entre os pontos de uma aresta e a própria aresta. Assim obtemos o gradiente dos gradientes da solução, ou seja, a "hessiana". A média da "hessiana" nas arestas de uma face fornecerá a indicação do "erro" do elemento. Temos assim uma tamanho de elemento associado à "hessiana" da solução, onde ela for maior, menor será o novo tamanho calculado "h_{new}", conforme a eq. 3.5:

$$e_e = \frac{1}{3}\left(\sum_1^3\left|\left(\nabla\hat{u}_j - \nabla\hat{u}_i\right).\overrightarrow{L_{ij}}\right|\right) \tag{3.4}$$

Podemos obter os espaçamentos requeridos para a malha a cada adaptação, pela seguinte expressão:

$$h_{new} = \frac{h_{old}\,.\,e_m}{\|e_e\|} \tag{3.5}$$

Sendo e_m o erro médio, calculado coforme a eq. (3.6):

$$e_m = Tol_i\,.\frac{\|\nabla\hat{u}^*\|}{\sqrt{N_e}} \tag{3.6}$$

onde:

$\nabla\hat{u}_i$ – gradiente aproximado da solução no nó "*i*";

e_e – erro associado ao elemento triangular;

h_{old} – dimensão característica do elemento na malha corrente;

h_{new} – dimensão característica desejada para o elemento *e* na malha adaptada;

N_e – número de elementos da malha;

Tol_i – Tol_1 *ou* Tol_2 associadas ao erro em todo o domínio e do domínio expurgando os elementos com singularidades, respectivamente.

Pelo fato de adotarmos duas tolerâncias distintas, a Eq. 3.5 que calcula as dimensões desejadas para os elementos da nova malha conduzirá a dois espaçamentos distintos para os elementos que deverão ser adaptados (ARAÚJO, 2004). Nesse caso adota-se o parâmetro mais conservador, isto é, o que conduza a uma malha a favor da segurança, ou seja, o menor dos dois.

Como estamos utilizando uma única malha para o cálculo simultâneo dos resultados de pressão e de saturação, o procedimento de análise de erros tem que ser realizado duas vezes. Uma análise é feita para o campo de pressão e outra análise para o campo de saturação. Ao final da análise para o campo de saturação é feita a comparação com o novo h_{new} calculado para a pressão e é então utilizada a malha atual associada com o h_{new} que for o menor dos dois para formar a malha de "background", isto é, novamente, o que conduza a uma malha a favor da segurança.

O critério de adaptação no entanto é o erro global e é definido como a média dos erros locais de cada elemento da malha, assim quando o valor do erro global ultrapassar a tolerância predefinida pelo usuário, se iniciará o processo de adaptação.

Como temos dois erros locais calculados cada um com uma tolerância Tol_1 *ou* Tol_2 teremos consequentemente dois erros globais a considerar. Quando um desses valores ultrapassar a respectiva tolerância é iniciado o processo de adaptação da malha.

3.7 Definição dos parâmetros da malha para adaptação via redefinição de malhas

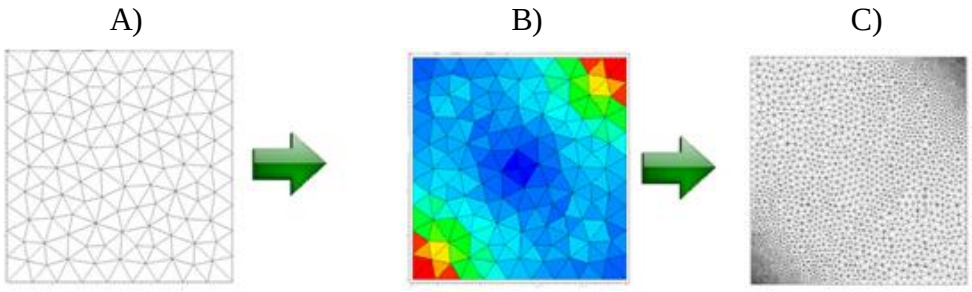
A geração das malhas é feita pelo *Gmsh* utilizando a técnica de controle de refinamento através do conceito de malha de "background" descrito a seguir.

Para uma malha inicial é comum usar uma malha grosseira que cobre todo o domínio onde a distribuição espacial dos nós é consistente com a geometria.

Para a simulação com adaptação a solução encontrada com a malha inicial é utilizada para encontrar o erro local dos elementos e a partir dele e da tolerância pré-estabelecida, definir os novos tamanhos de elementos na nova malha a ser gerada. Espacialmente é usada a posição dos nós da malha atual para indicar os tamanhos com os quais serão criados os novos elementos. Esta informação dos tamanhos é conhecida como CL, ou comprimento característico ("*characteristic lenght*") é calculado conforme apresentado na seção 4.4. Desta forma na malha atual cada nó em dada região a ser adaptada, recebe um valor para o comprimento característico, formando um campo de CL´s. Agora a malha atual passa a ser conhecida como malha de "background" pois é usada desta maneira para controlar o processo de adaptação ou de geração de nova malha que ocupará o seu lugar na simulação.

A figura 3.3 mostra um caso de adaptação global para o problema de escoamento monofásico no regime estacionário para ¼ de "*five spot*" com os poços nos cantos inferior esquerdo e superior direito. Na escala verificamos as regiões de maior ou menor refinamento definidas pelo indicador de erros (campo de tamanho de elementos h_{new} para as faces triangulares) e a nova malha resultante, adequada àquela simulação *steady state* (cálculo de pressão no campo).

Figura 3.3 – A) Malha inicial B) Campo de novos tamanhos para as faces h_{new} C) Malha gerada final.



Fonte: O autor.

Para proceder a modelagem geométrica, discretização espacial inicial (malha, condições de contorno, etc) e geração da nova malha adaptada é usado o software aberto (*open source*) *Gmsh*.

3.8 Gerador de malhas gmsh

Gmsh é um gerador de malhas 2D/3D para simulações com elementos finitos. Suas vantagens são requerer relativamente pouco uso de memória e facilidade de utilização, além de possuir capacidades avançadas de visualização. Além disso, é um software de código aberto, onde é possível o seu uso como biblioteca do C++. Ainda é um programa em constante atualização contando com excelente documentação disponível no site do autor (http://www.geuz.org/gmsh), e com extensa comunidade em uso, dispondo assim de fonte bastante acessível de informações quanto à utilização e funcionamento. Além do formato do *Gmsh* nativo ".msh" o *Gmsh* pode ler / escrever muitos arquivos de malha padrão, contudo o arquivo utilizado neste trabalho é o próprio arquivo .msh, devido à sua simplicidade e ao fato da biblioteca adotada (FMDB) já possuir funções de leitura no formato .msh. Essas características fizeram do *Gmsh* o programa escolhido como núcleo do processo de geração das malhas associado à adaptação por redefinição de malhas ou “remeshing”.

O *Gmsh* é construído em torno de quatro módulos: geometria, malha, solver e pós-processamento. A especificação de qualquer entrada para estes módulos é feito de 3 formas a saber: utilizando a interface gráfica do usuário, em arquivos de texto usando a própria linguagem de script do *Gmsh*, ou utilizando-o como biblioteca e suas funções internas diretamente no escopo do programa em desenvolvimento (GEUZAINE C., 2014).

3.8.1 Criação de Geometrias

O primeiro requisito para a criação de uma malha, qualquer que seja o software utilizado para isso, é a modelagem da geometria a ser discretizada, seja ela 1D, 2D, ou 3D.

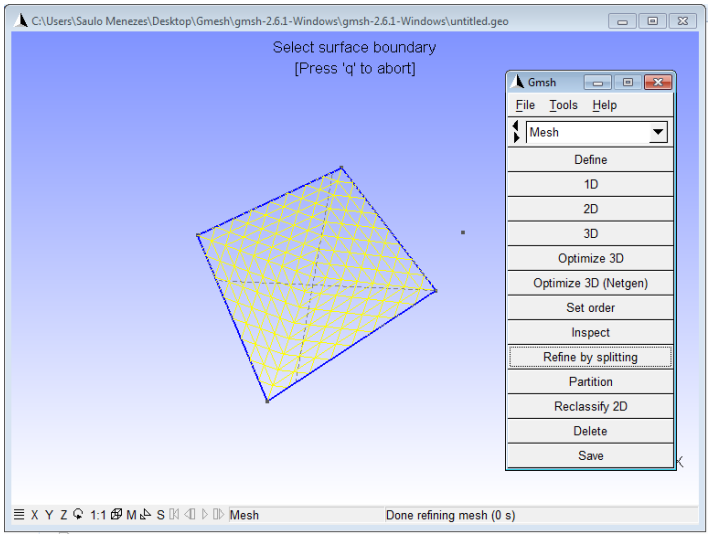
Como já mencionado, existem 3 formas de se criar uma geometria no *Gmsh* a saber:

- Como uma biblioteca de funções em C++;

- Através da programação do *Gmsh* a partir de linguagem própria em arquivos de texto (scripts);
- Através de comandos da interface gráfica ou GUI (Graphics User Interface) do *Gmsh*.

Essas alternativas representam três formas distintas de utilizar o *Gmsh*, seja para criar geometrias, malhas ou pós processamento. A figura 3.4 apresenta a interface gráfica do *Gmsh* com o menu de ferramentas posicionado lateralmente.

Figura 3.4 – Interface gráfica do *Gmsh*.



Fonte: O autor.

3.8.1.1 Utilização do *Gmsh* através de interface gráfica

Usando a interface gráfica, possuímos ferramentas 3D de criação de geometrias em formato CAD clássico com os principais elementos de desenho, tais como pontos, linhas restas, arcos ou círculos, Splines, e superfícies. Para a geração de superfícies tem-se duas opções: *Plane Surfaces* (superfícies planas) e *Rulled Surfaces* (Não planas), são compostas por "*loops*" de linhas fechados (*line loops*) e são elementos geométricos necessários à criação de malhas 2D. Temos ainda os Volumes que são elementos geométricos tridimensionais formados por superfícies conectadas originando uma única superfície fechada contínua ou não, e são necessários à criação de malhas 3D.

A utilização do software desta forma é intuitiva e será através do menu de ferramentas, dispensando qualquer curso especializado, apenas é necessário especificar com qual dos módulos se deseja trabalhar e usar as ferramentas através dos botões do menu, além disso existem vários tutoriais disponíveis na página oficial do software.

3.8.1.2 Utilização do *Gmsh* através de script

O uso de arquivos de texto "script", deve obedecer a uma linguagem própria do *Gmsh*. Um arquivo de "script" é um arquivo de texto ASCII que contém instruções em

linguagem de "script" do *Gmsh*. Esse arquivo é interpretado pelo analisador do *Gmsh*, e pode ser dada qualquer extensão (ou nenhuma extensão a todos). Por convenção, *Gmsh* usa a extensão .geo para "scripts" de geometria, .msh para arquivos de malha e a extensão .pos para conjuntos de dados para pós-processamento, utilizados por exemplo, no caso, para guardar informações da malha de "background". A maioria das ações interativas do GUI (Grafics User Interface ou Interface gráfica) têm um equivalente direto na linguagem de "script".

3.8.2 Geração da malha usando o Gmsh

Uma malha de elementos finitos é uma discretização de um dado subconjunto do espaço tridimensional por elementos geométricos elementares de várias formas. No caso do *Gmsh*: linhas, triângulos, tetraedros, prismas, hexahedros e pirâmides.

A malha de elementos finitos produzida pelo *Gmsh* é considerada como "não estruturada", mesmo que tenha sido gerada de uma maneira "estruturada" (por exemplo, extrusão). Isto implica que os elementos geométricos são definidos por uma lista ordenada dos seus nós, mas é assumido que não têm relação de ordem predefinida entre quaisquer dois elementos e, portanto, informando também a topologia através de matriz de conectividades (GEUZAINE C., 2014).

No caso de uma malha não estruturada, os pontos não podem ser representados através de uma lógica igual à estruturada. Enquanto na estruturada os vizinhos podem ser facilmente identificados através de uma regra geral, na malha não estruturada essa lei não existe, devendo a identificação das conectividades ser explicitada através de uma matriz conhecida como matriz de conectividades (THOMPSON J. et al 1998). Apesar desta característica ter um impacto computacional negativo, i.e requerer maior memória e endereçamento indireto, as malhas não estruturadas por essa mesma razão possuem relevantes vantagens sobre as estruturadas.

Primeiro, por não possuírem uma lógica de identificação de nós e conectividades as malhas não estruturadas permitem a livre inclusão e remoção de nós, tantos quantos forem necessários em qualquer região da malha.

Segundo, permitem o uso de qualquer tipo de elemento por exemplo no caso bidimensional, sejam quadriláteros, triângulos, pentágonos ou mesmo o uso de ambos numa mesma malha, fazendo com que a malha possua uma capacidade de adaptação a qualquer tipo de geometria, permitindo a inserção de tantos elementos quanto for preciso para alcançar a acurácia desejada na análise.

Como o objetivo deste trabalho é a implementação de uma ferramenta de adaptação local a escolha recai sobre o uso de malhas não estruturadas afim de dispor de liberdade para refinamento local ao nível que for necessário.

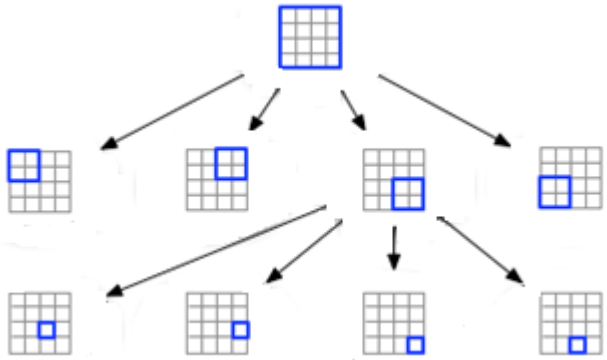
Os casos que vamos estudar são casos de simulação de reservatórios de petróleo em duas dimensões incorporando adaptação automática, através de malhas triangulares bidimensionais e não estruturadas.

3.8.3 Geração de malhas usando o JMesh

O termo *Quadtree* é utilizado para descrever uma classe de estruturas hierárquicas cuja propriedade em comum é o princípio da decomposição recursiva do espaço.

Atualmente as *Quadtrees* são usadas para representar pontos, regiões, curvas, superfícies e volumes. A decomposição espacial de um domínio pode ser feita em partes iguais em cada nível (decomposição regular), ou pode depender dos dados de entrada. A resolução da decomposição, ou seja, o número de vezes que o processo de decomposição é aplicado, pode ser fixado com antecedência, ou pode depender das propriedades dos dados de entrada. Exemplo de estrutura hierárquica *Quadtree* e seu uso para decomposição espacial na figura 3.5.

Fig. 3.5 Decomposição espacial bidimensional por estrutura hierárquica *Quadtree.*



Fonte: O autor.

A aplicação de métodos de decomposição espacial para geração de malhas para simulação numérica teve início a aproximadamente três décadas atrás, alguns desses métodos mostraram-se robustos e confiáveis e continuam sendo usados em uma variedade de aplicações de engenharia (FREY, 1998) incluindo geradores de malhas para simulação numérica.

Vem sendo desenvolvido um gerador de malhas pelo Computer Graphics, Virtual Reality and Animation Group da Universidade Federal do Ceará. O gerador chamado de JMesh vem sendo utilizado no desenvolvimento deste adaptador. Suas principais vantagens para este trabalho são (FREITAS, 2011) :

- Gerar malhas triangulares bidimensionais utilizando computadores paralelos de memória distribuída;
- Produzir elementos de boa qualidade, evitando elementos com razão de aspecto ruins;
- Proporcionar boas transições entre regiões da malha com grande densidade de elementos e regiões com baixa densidade de elementos da malha;
- Possibilidade de ser utilizado como biblioteca de funções.

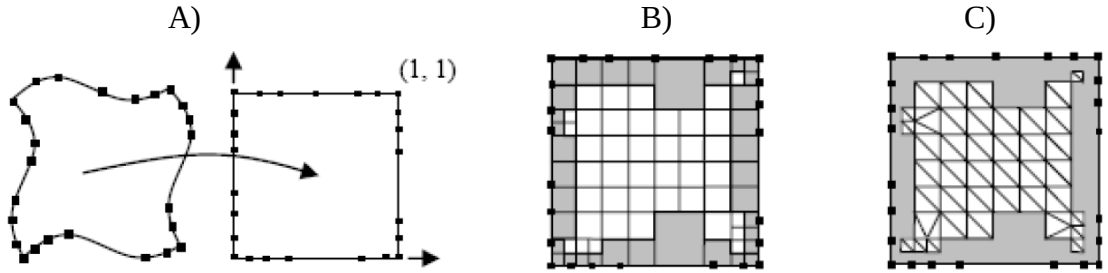
O JMesh possui algoritmos específicos que visam assegurar cada uma das características supracitadas, portanto as vantagens aqui evidenciadas não são inerentes ao método utilizado pelo gerador e sim asseguradas por rotinas próprias do gerador JMesh.

O JMesh utiliza as técnicas de geração a partir de uma *quadtree* e a técnica de avanço frontal. Os passos seguidos são:

1) Discretização da geometria;
2) Geração da *quadtree*, ajustes e otimização;
3) Eliminação das células perto dos contornos;
4) Geração dos elementos no interior das células;
5) Geração dos elementos exteriores às células por avanço frontal.

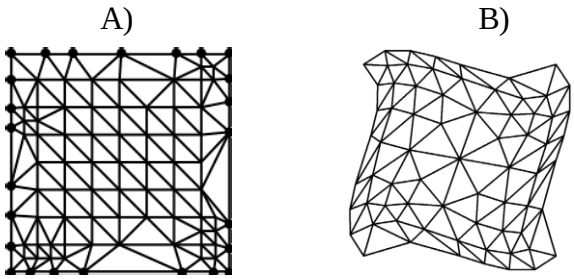
O gerador utiliza a *quadtree* para auxiliar a determinar os pontos que formarão os elementos triangulares ideais em uma técnica de avanço frontal (MIRANDA, 1999).

Fig. 3.6 Passos da geração da malha pelo JMesh A) discretização da geometria B) geração da quadtree C) Formação da malha nas regiões internas às células e remoção das células próximas aos contornos.



Fonte: Miranda, 1999.

Fig. 3.7 Malha gerada pelo JMesh A) Resultado final da etapa de avanço frontal B) Malha final após mapeamento dos nós na superfície tridimensional inicial.



Fonte: Miranda, 1999.

3.9 O interpolador

Após o procedimento de adaptação é necessário interpolar as soluções de todas as variáveis da malha atual para a nova malha encontrando dessa forma os valores da solução nos novos pontos criados na malha. Encontrado o novo nó, são utilizadas as funções de forma lineares (ou bilineares no caso de quadriláteros). Utilizamos uma interpolação pontual onde cada ponto criado na nova malha P_{new} deve ser primeiramente localizado no elemento bidimensional K_{old} da antiga (ZHENG et al, 2005).

Esta localização é efetuada através de uma função FMDB que procede uma busca em sua estrutura de armazenamento de elementos. Como o os elementos são organizados em forma de árvore *quadtree* a busca é eficiente e requer relativamente pouco custo computacional.

Os métodos de interpolação usados neste trabalho são linear, quadrática e adaptativa (LYRA, 1994). Seguem esclarecimentos a respeito de cada um deles.

3.9.1 O método linear de interpolação

Em um algoritmo de adaptação por redefinição de malhas é sempre necessário interpolar resultados da malha antiga para a nova malha. Durante uma análise transiente um grande número de interpolações pode ser requerido, resultando em uma acumulação do erro de interpolação, transcrevemos a seguir a dedução matemática da difusão numérica tal como é encontrada em Lyra (1994).

Considere inicialmente um problema unidimensional onde a derivada temporal pode ser discretizada segundo a fórmula:

$$\frac{\partial u_p}{\partial t} \cong \frac{u_p^{n+1} - u_p^n}{\Delta t} \tag{3.7}$$

Que representa uma aproximação convencional por diferença finita. Quando a mesma malha é usada nos tempos T^n e T^{n+1}, neste caso a acurácia do resultado depende apenas da acurácia do algoritmo escolhido para atualizar o campo de fluxo. Contudo devido o processo de adaptação, no instante T^{n+1} eventualmente acontecerá de termos de atualizar um ponto na malha nova que não existe na antiga, então deve ser interpolado usando-se os valores dos nós da malha antiga. A derivada temporal na prática é calculada desta forma:

$$\frac{\partial u'_p}{\partial t} \cong \frac{u_p^{n+1} - \hat{u}_p^n}{\Delta t} \tag{3.8}$$

onde $\hat{u}_p^n$ é um valor interpolado.

Esta forma de calculá-la inclui termos de ordem mais alta resultando em uma difusão numérica representada por:

$$\frac{u_p^{n+1} - \hat{u}_p^n}{\Delta t} = \frac{u_p^{n+1} - u_p^n}{\Delta t} - \frac{(x_2 - x_p)(x_p - x_1)}{2\Delta t} \frac{\partial^2 u^n}{\partial x^2}\Big|_{x_p} + T.O.A. \tag{3.9}$$

onde o termo:

$$\frac{(x_2 - x_p)(x_p - x_1)}{2\Delta t} \frac{\partial^2 u^n}{\partial x^2} \tag{3.10}$$

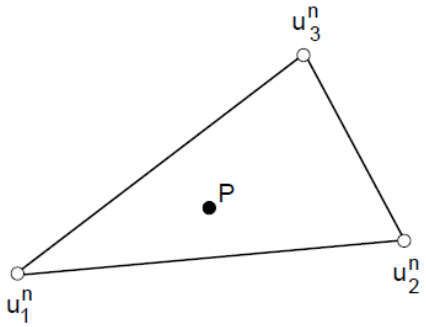
representa uma difusão numérica proveniente da interpolação linear.

Com o propósito de evitar/minimizar um problema igual em análises adaptativas bidimensionais foi desenvolvido um procedimento de segunda ordem (quadrático). Esta interpolação utiliza não apenas os valores dos vértices dos elementos triangulares mas também os gradientes suavizados já calculados pelas rotinas do estimador de erros para o campo de velocidades.

3.9.2 O método quadrático de interpolação

Suponha que u seja um componente genérico do vetor de variáveis primitivas *U* e que as direções cartesianas sejam representadas por *x* e *y*, para conveniência de notação nesta seção, u_p^n é a interpolação linear usando a funções de forma e os valores nodais $u^n{}_k$ indicados na figura 3.5.

Figura 3.8 – Interpolação em um elemento triangular.



Fonte: Lyra (1994).

$$\widehat{u_p^n} = \sum_{k=1}^{3} N_k(x_p, y_p) u_k^n \tag{3.11}$$

Usando a série de Taylor na vizinhança do nó P temos:

$$u^n(x_p, y_p) = u^n(x_k, y_k)\ (x_p - x_k)\left.\frac{\partial u^n}{\partial x}\right|_k + (y_p - y_k)\left.\frac{\partial u^n}{\partial y}\right|_k + \frac{(x_p - x_k)^2}{2}\left.\frac{\partial^2 u^n}{\partial x^2}\right|_k + \frac{(y_p - y_k)^2}{2}\left.\frac{\partial^2 u^n}{\partial y^2}\right|_k + (x_p - x_k)(y_p - y_k)\left.\frac{\partial^2 u^n}{\partial x \partial y}\right|_k + T.O.A. \tag{3.12}$$

Fazendo uso destas duas equações obtemos os valores interpolados u_p^n, corretos para segunda ordem, após alguma manipulação algébrica a partir da relação:

$$\tilde{u}_p^n = \hat{u}_p^n + \sum_{k=1}^{3} N_k(x_p, y_p) R_k \tag{3.13}$$

Com:

$$R_k = \left(x_p - x_k\right)\left.\frac{\partial u^n}{\partial x}\right|_k + \left(y_p - y_k\right)\left.\frac{\partial u^n}{\partial y}\right|_k + \left.\frac{\left(x_p - x_k\right)^2}{2}\frac{\partial^2 u^n}{\partial x^2}\right|_k + \left.\frac{\left(y_p - y_k\right)^2}{2}\frac{\partial^2 u^n}{\partial y^2}\right|_k + \left(x_p - x_k\right)\left(y_p - y_k\right)\left.\frac{\partial^2 u^n}{\partial x \partial y}\right|_k \quad (3.14)$$

Onde os termos de ordem mais alta foram desconsiderados e as derivadas de segunda ordem são calculadas por diferenciação dos gradientes contínuos suavizados.

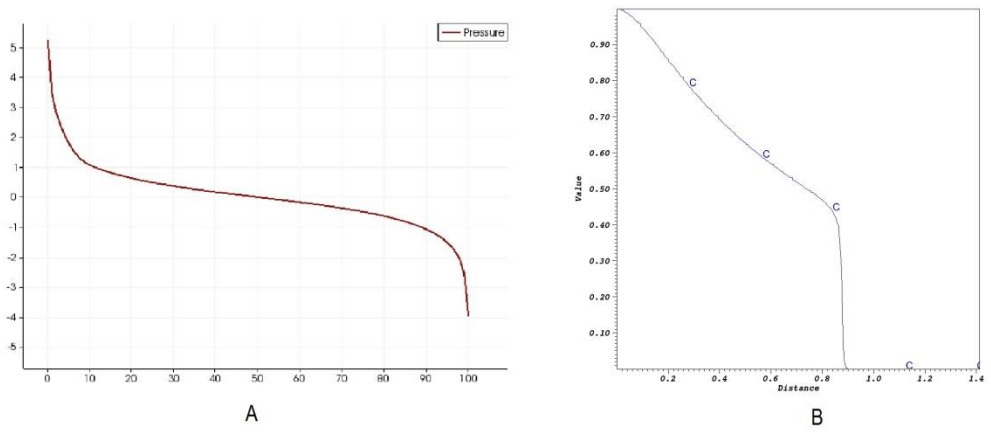
Note que esta interpolação de segunda ordem é obtida adicionando um termo de correção à interpolação de primeira ordem. Em alguns casos pode ser conveniente manter a interpolação em primeira ordem não incorporando este fator de correção. Este caso será explicado no item seguinte com a interpolação adaptativa.

3.9.3 O método de interpolação adaptativo

Para ilustrar consideremos o caso do problema de ¼ dos cinco poços. Observa-se que entre a interface óleo-água e o poço injetor observamos no gráfico de saturação (fig. 3.9), um gradiente suave negativo da saturação até próximo da interface óleo-água quando há uma acentuação deste gradiente formando praticamente uma queda linear até o zero de saturação de água.

Figura 3.9 – Regiões de singularidades A) curva de pressão ao longo da diagonal dos poços e B) curva de saturação ao longo da diagonal entre os poços.

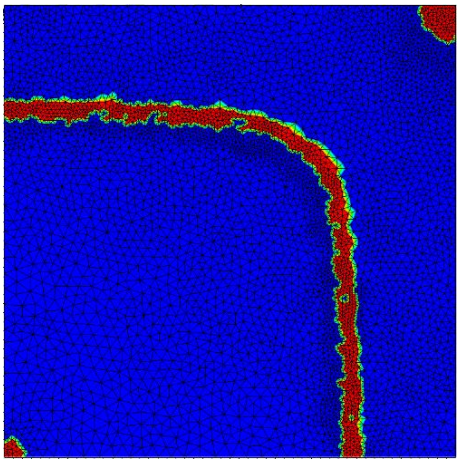
A) B)



A B

Fonte: O autor.

Figura 3.10 – Regiões de singularidades em escoamento bifásico.



Fonte: O autor.

Este comportamento dos perfis de saturação sugere que, neste caso, para que se tenha melhor acurácia se utilize em regiões distintas estratégias de interpolação adequadas a cada situação. Da mesma forma observamos no gráfico de pressão, fig 3.9(a), que as regiões dos arredores dos poços as pressões possuem gradientes acentuados essas regiões são tratadas nesta metodologia como singularidades e no procedimento de interpolação adaptativo é utilizada a interpolação linear. Para locais onde os gradientes são mais suaves (anteriores ao choque água-óleo e distantes dos poços) é mais adequado a aproximação pela interpolação de segunda ordem ou quadrática.

O critério usado para esta identificação é o de detecção de singularidades discutido na seção 3.6, nestas regiões (fig. 3.10), onde os gradientes são muito elevados dentro de uma área pequena, o processo de interpolação é linear e fora, onde os gradientes possuem menor módulo, o processo é quadrático. No capítulo 5 procederemos estudos comparativos entre os métodos de interpolação linear, quadrático e adaptativo.

4 Adaptação automática de malhas através de redefinição local

Este capítulo trata do foco principal nesta etapa do trabalho que é o módulo adaptador, núcleo do processo de adaptação e ponte de integração dos módulos de análise de erro, geração de malhas e interpolação de resultados entre as malhas.

4.1 Etapas seguidas pelo adaptador

O processo de redefinição local de malha consiste em gerar uma nova malha com o tamanho de elementos especificado que atenda às necessidades impostas pela física do fenômeno em estudo, global ou localmente e a minimização do erro global aos padrões de tolerância pré-estabelecidos. Para tanto é necessário um mecanismo de controle dos tamanhos desses elementos.

O adaptador é acionado pelo o estimador de erros quando o valor do erro global ultrapassa a tolerância especificada. A partir daí os passos que o adaptador segue são os seguintes:

1- Uma lista de ponteiros para os elementos da malha que forma as regiões a serem adaptadas;
2- Tratamento das regiões a serem adaptadas;
3- Geração do arquivo .pos (malha de background);
4- Agrupamento das regiões da malha a serem adaptadas e remoção dos elementos internos às mesmas;
5- Criação do arquivo de geometria necessário à geração da nova malha;
6- Geração da nova malha local;
7- União das malhas (atual e nova) – "merge";

A seguir destacamos cada uma das etapas do algoritmo listadas acima.

4.2 Recebimento da lista de elementos a remover

O adaptador recebe do simulador uma lista de ponteiros para faces que apontam os elementos triangulares a serem removidos. Através da função **getRefUnrefElementsList** da classe **ErrorAnalysis.** Essa lista entra como parâmetro da função **Identify_StgElements** que recebe também a lista de domínios da malha.

A rotina **Identify_StgElements** objetiva identificar elementos "estranhos", "dentes" nos contornos da região a ser remalhada e "buracos" que se "tocam" (fig. 4.1a e 4.1b) e os acrescenta à lista de elementos a serem removidos.

4.3 Tratamento das regiões a ser adaptadas

É preciso manter regiões de adaptação distintas de tal forma que seja possível identificar os contornos de arestas que delimitam cada uma dessas regiões.

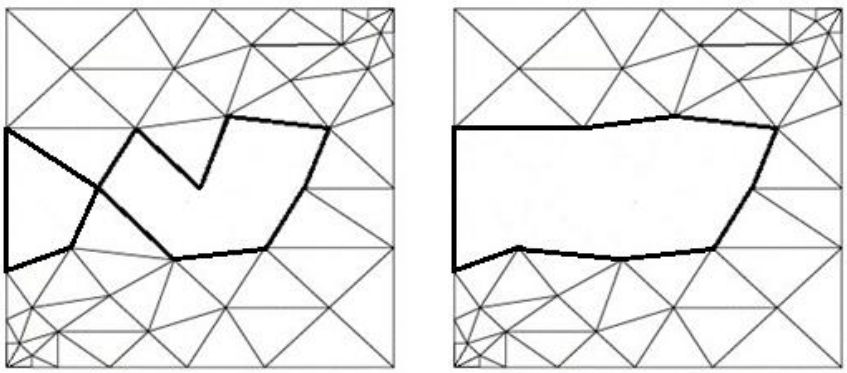
Para o gerador de malhas é necessário que sejam indicadas as arestas em uma ordem específica formando um "*loop*" fechado. A função **Create_LineLoop** ordena estas arestas para o gerador de malhas, mas ela requer uma região bem delimitada. Cada nó do contorno de uma região de adaptação deve pertencer a apenas uma região para que haja apenas um caminho possível para um "*loop*".

Assim o objetivo desta função é acrescentar mais alguns elementos à lista de elementos a ser removidos de forma a possibilitar a geração de malhas de boa qualidade nas regiões indicadas pelo estimador de erros. Esta função não procede a remoção de nenhum elemento. Esses elementos na malha serão necessários para compor a malha de "background" (que associa a seus vértices os valores de tamanho local dos novos elementos da malha a ser gerada) e só depois é que é feita a remoção das faces triangulares.

Basicamente ela identifica os vértices que possuem apenas uma face nas suas conectividades e os que possuem mais arestas do que faces conectadas com uma diferença de duas ou mais arestas do que faces. Estes vértices são os pontos de contato entre duas regiões de adaptação local, "StgNodes", e podem ser visualizados na fig. 4.1.

Figura 4.1 – Uniao de *Lineloops* A) "StgNode" causando junção de *loop´s* e "dente" B) Contorno bem definido com apenas um caminho possível.

A) B)



Fonte: O autor.

Posteriormente existe uma iteração sobre estes vértices e a remoção de todas as faces conectadas a ele.

Inicia-se uma nova verificação pois este processo pode causar a criação de novos pontos de contato entre regiões, assim a função permanece neste laço recursivamente até que não exista mais nenhum "StgNode".

Para fins de esclarecimento é apresentado um pseudocódigo da função descrita.

Função IdentifySTGNodes

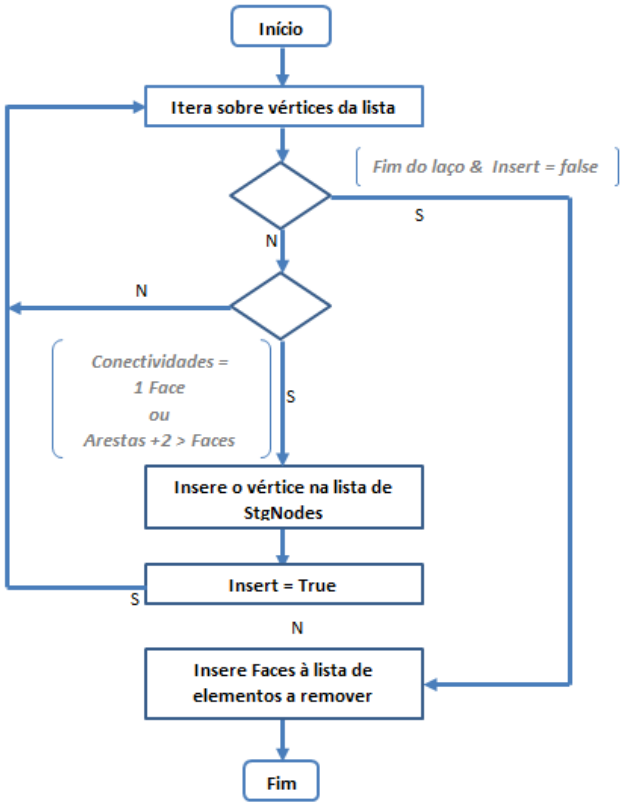
Loop nos vértices da lista de remoções

Se o vértice é um StgNode

Insere as faces conectadas ao vértice à lista de remoções

A fig. 4.2 apresenta o fluxo de operações desta função e a fig. 4.3 apresenta o resultado do antes e depois de acionada esta função bem como casos onde foi criado um novo StgNode.

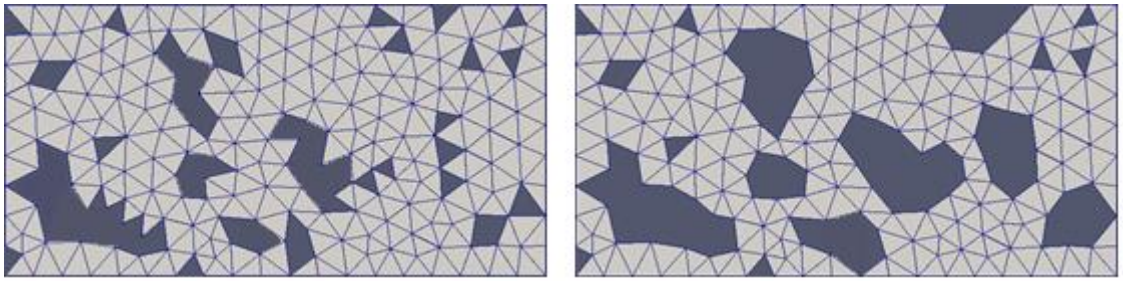
Figura 4.2 – Fluxograma de operações da função **Identify_StgElements.**



Fonte: O autor.

Figura 4.3 – Resultado da função Identify_StgElements.

A) Antes B) Depois



Fonte: O autor.

4.4 Geração do arquivo .pos (malha de background)

Vimos na seção 3.6 que a cada elemento triangular é atribuído segundo a eq. (3.5) Um valor h_{new}.

A malha de "*background*" recebe o valor dos *CL* atribuídos a cada vértice**.** Para tanto é necessário proceder uma iteração sobre os vértices e para cada um calcular a média dos h_{new} de cada face a qual o dado vértice se conecta. Este valor é guardado no flag "elem_height" da face pelo estimador de erro que calcula os valores dos *CL´s*. Dessa forma o *CL* atribuído a cada vértice da malha de "*background*" é a média dos valores "elem_height" das faces conectadas ao vértice.

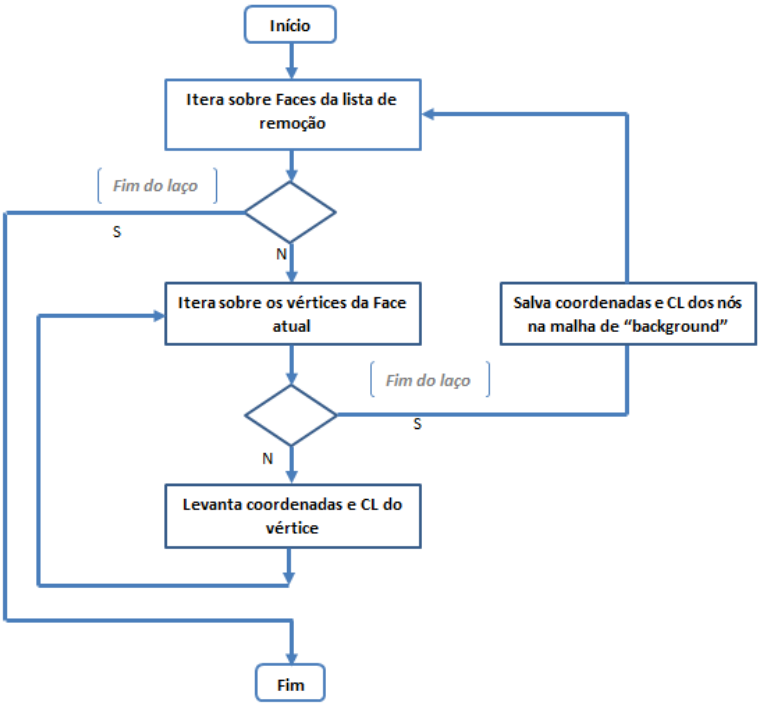
$$CL = \frac{\sum_1^N h_{new}}{N} \tag{4.1}$$

N – quantidade de faces conectadas a cada vértice

Além dos valores dos comprimentos característicos (*CL´s)* são levantadas as coordenadas dos pontos de cada vértice para formar a malha.

A figura 4.4 mostra o processo onde existe um laço iterando sobre as faces e para cada face um laço iterando sobre os vértices e levantando as informações necessárias para a malha de "*background*" e a seguir é mostrado o pseudocódigo com as rotinas da função.

Figura 4.4 – Fluxograma função SaveBGMView.



Fonte: O autor.

Função SaveBGMView

Loop nos elementos da lista de remoções

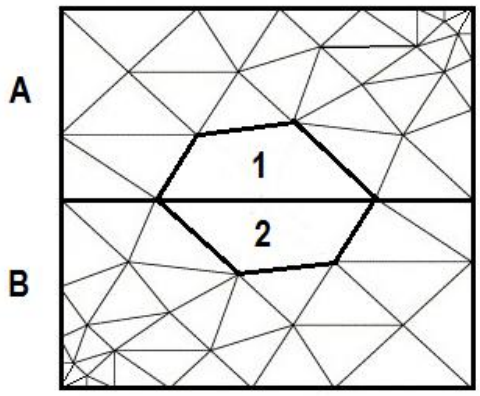
Loop nos vértices das faces

Especificação dos CL´s na malha de "background"

4.5 Identificação das regiões de adaptação local

Um dos principais requisitos do adaptador é que mantenha os limites geométricos dos N domínios existentes na malha, ou seja, a geometria interna deve ser preservada. Para tanto é necessário identificar as regiões de adaptação local pois uma determinada região pode atravessar dois ou mais domínios e afim de preservar os limites geométricos entre domínios é necessário dividir o que seria uma região de adaptação em duas ou mais regiões. Esta função é realizada pela rotina **Identify_and_Remove_Edges.** A figura 4.5 apresenta este caso onde os elementos a remover geram uma região entre domínios.

Figura 4.5 – Região de adaptação local entre domínios.



Fonte: O autor.

Apresento o pseudocódigo geral da função e a seguir esclarecimentos sobre as sub-rotinas.

Pseudocódigo Função Identify_and_Remove_Edges

Loop nos elementos Faces da lista

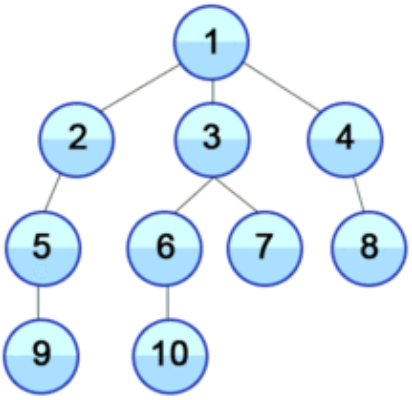
1. Identifica um elemento do domínio;
2. procede a busca em largura identificando uma região de adaptação;
3. Remove todas as faces da região da lista de remoções e da malha;
4. Remove da malha as arestas e vértices internos à região;
5. Insere o conjunto de arestas restantes em uma lista de conjuntos do respectivo domínio

A função **Identify_and_Remove_Edges** inicia iterando sobre a lista de domínios, assim para cada domínio a função identifica os elementos de uma região de adaptação. Faz isso através de um algoritmo de busca em largura, ou busca em amplitude (*Breadth-First Search BFS*).

Busca em largura é um algoritmo de busca em grafos utilizado para realizar uma busca ou travessia num grafo e estrutura de dados do tipo árvore. Intuitivamente, você começa pelo vértice raiz e explora todos os vértices vizinhos. Então, para cada um desses vértices mais próximos, a busca explora os seus vértices vizinhos inexplorados e assim por diante, até que ele encontre o alvo da busca (CORMEM T. H., 2001).

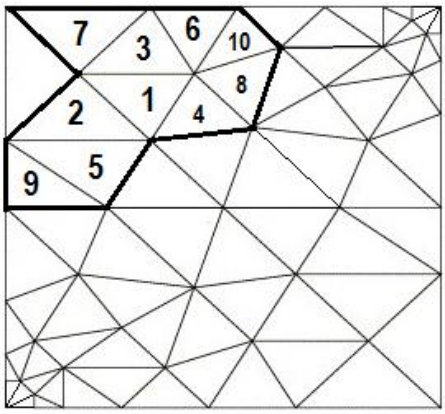
A figura 4.6 demonstra essa ordem de busca em grafo pelos elementos numerados e a figura 4.7 ilustra a forma como este método foi empregado para identificar as regiões de adaptação a partir de uma primeira face da região e suas faces vizinhas. Por essa razão é preciso primeiro identificar algum elemento do domínio antes de começar a busca.

Figura 4.6 – Sequência de busca em árvore local algoritmo BFS.



Fonte: O autor.

Figura 4.7 - Região de adaptação identificada na busca BFS.



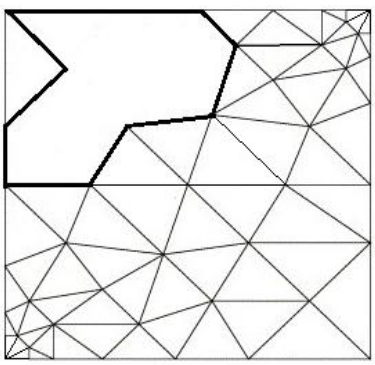
Fonte: O autor.

Utilizamos este método de busca para identificar as regiões de adaptação inicialmente iterando sobre os domínios e dentro deste laço sobre a lista de faces e acrescentando a partir da primeira, as suas vizinhas que estiverem na lista de remoção (as que estiverem receberam valor = 1 no "flag" "elementList" pela função **Identify_StgElements**). Até não haver mais faces vizinhas naquela região de adaptação sendo este critério de parada a única diferença do uso tradicional deste algoritmo.

Assim é preenchido um conjunto de faces relativa a alguma região enquanto houver faces pertencentes ao domínio na lista de remoções. Quando for concluída a busca (não houverem mais faces pertencentes ao domínio na lista de remoções) o processo é reiniciado para o domínio seguinte até a conclusão da iteração sobre a lista de domínios.

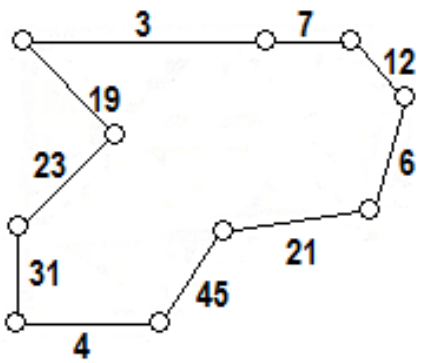
Dessa forma cada lista contém todas as faces de uma região a ser remalhada e apenas uma região. Simultaneamente à identificação das regiões é feita a identificação de todas as arestas conectadas às faces e posteriormente são apagados todos os elementos internos à região, i.e, que não forem do contorno da região. As fig. 4.8 e 4.9 apresentam as arestas que restaram e que servirão para formar a geometria para a nova malha.

Figura 4.8 – Remoção das arestas e faces.



Fonte: O autor.

Figura 4.9 – Conjunto de arestas internas delimitando uma região de adaptação.



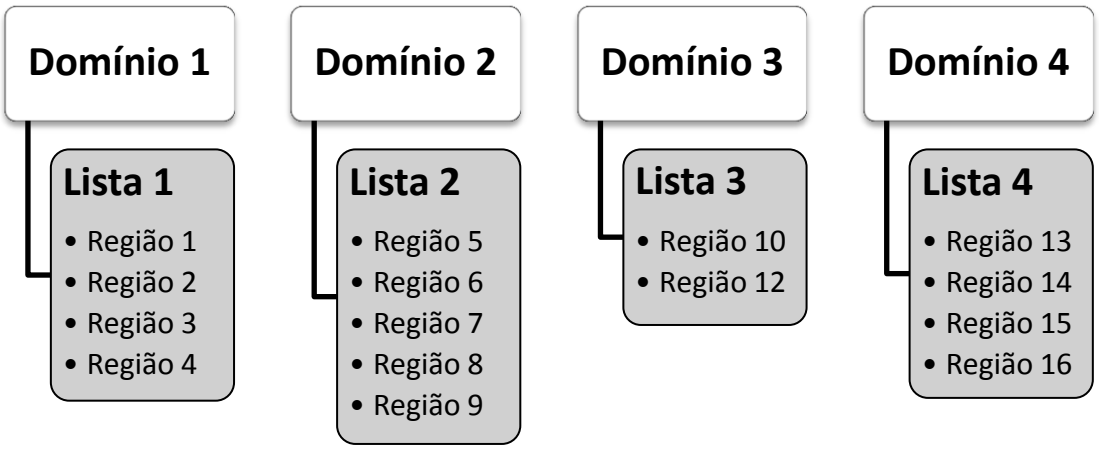
Fonte: O autor.

As arestas de contorno que não foram apagadas foram incluídas em um conjunto que por sua vez é inserido em uma lista. Assim para cada subdomínio temos uma lista de conjuntos de arestas, cada conjunto representando uma região de adaptação. Ainda no mesmo domínio segue a iteração em busca de outras regiões que pertençam ao subdomínio em análise. Quando esgotadas as faces de um determinado subdomínio o ID do subdomínio é inserido como chave de um *map* e a lista inserida como o conteúdo, o iterador sobre a lista de domínios é incrementado para o próximo domínio.

Ao final temos um *map* (container padrão da STL onde os conteúdos são classificados por uma chave) onde o ID do domínio é a chave para cada lista e os conteúdos das listas são de conjuntos de arestas onde cada conjunto guarda o contorno e assim define uma região de adaptação.

A figura 4.10 apresenta esta organização onde cada chave identifica uma lista de um determinado subdomínio.

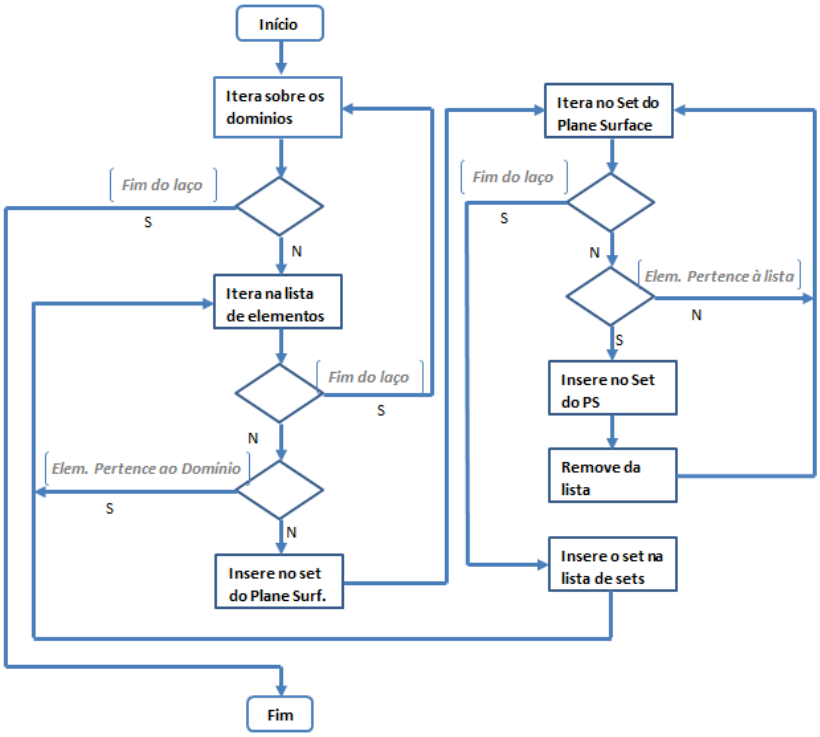
Figura 4.10 – Container da STL map com as regiões de adaptação respectivas a cada subdomínio.



Fonte: O autor.

As arestas dentro de cada conjunto (região) estão organizadas em ordem crescente do *ID*, a próxima função trata de reordenar as arestas em sequências de "loops" (laços) fechados necessários para a criação das superfícies que por sua vez são os requisitos para a geração das malhas pelo *Gmsh*.

Figura 4.11 – Fluxograma das operações da função Identify_and_Remove_Edges.



Fonte: O autor.

Os elementos em um container do tipo set são armazenados em forma de uma árvore binária usando como critério de ordenação o valor do *ID* do elemento em ordem crescente isso é feito no intuito de otimizar a busca, mas ao iterar sobre essa árvore os elementos serão ordenados de forma crescente resultando:

Line Loop (10) = {3, 4, 6, 7, 12, 19, 21, 23, 31, 45}

Este comando mostra o elemento Line Loop de ID 10 formado pelas dez arestas representadas pelos seus ID´s entre chaves. Notar que estão dispostas em ordem crescente do respectivo ID.

No entanto o que é necessário para a definição da região para o gerador de malhas é que as arestas sejam ordenadas seguindo um "*loop*", o que de acordo com a fig. 4.7 em sentido anti-horário é:

Line Loop (10) = {7, 3, 19, 23, 31, 4, 45, 21, 6, 12}

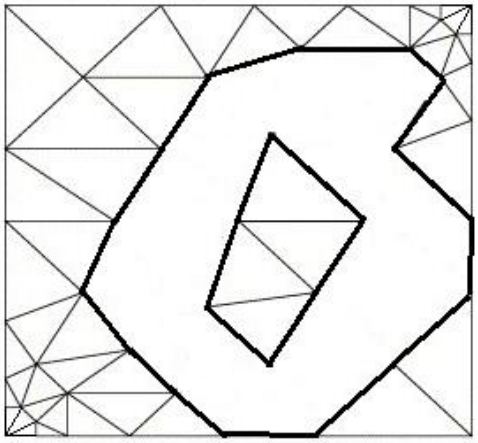
Assim é necessária uma rotina que as ordene dessa forma, esta função é a **Createlineloop** é chamada dentro da função **SaveGeometry_2D** descrita na seção 4.6.

4.6 Criação da geometria necessária à geração da nova malha

Esta função recebe como parâmetro o *map* contendo como as listas de conjuntos de arestas cada qual com as arestas do contorno de uma região. Este *map* foi criado na função **Identify_and_Remove_Edges** (ver seção 4.4), agora resta apenas iterar sobre ele para criar a geometria necessária ao gerador de malhas. Cabe observar que as arestas nos conjuntos além de não estarem ordenadas em *loop´s* fechados, não se sabe quantos

são estes *loop´s* para cada uma destas regiões. A figura 4.12 mostra um exemplo de região de adaptação delimitada por dois *loop´s* de arestas.

Figura 4.12 – Região de adaptação delimitada por mais de um loop de arestas.



Fonte: O autor.

Dessa forma é preciso uma função que remova dos conjuntos um *loop* por vez e os forneça à outra rotina que reordene as arestas. A função que separa é **SeparateCommonEdges** que remove do conjunto essas arestas, em seguida é chamada a função **Createlineloop** que as reordena em sequência de *loop*. Quando o conjunto esvazia-se completamente a região recebe um *ID* que é inserido em uma lista do grupo físico do subdomínio (os elementos das regiões destas listas receberão em um *flag* específico o *ID* do domínio a que pertencem quando for gerada a malha).

A seguir o iterador da lista é incrementado para o próximo conjunto que contém a região seguinte até esgotar a lista e conseguinte o subdomínio atual. O processo é repetido para todos os subdomínios que possuírem regiões a ser remalhadas até a conclusão da geração da geometria. Apresentamos agora o algoritmo desta função como pseudocódigo e em forma de fluxograma na fig. 4.13.

Função SaveGeometry_2D

Loop sobre os subdomínios

Cria grupo físico de regiões com o ID do subdomínio atual

Loop sobre a lista do subdomínio

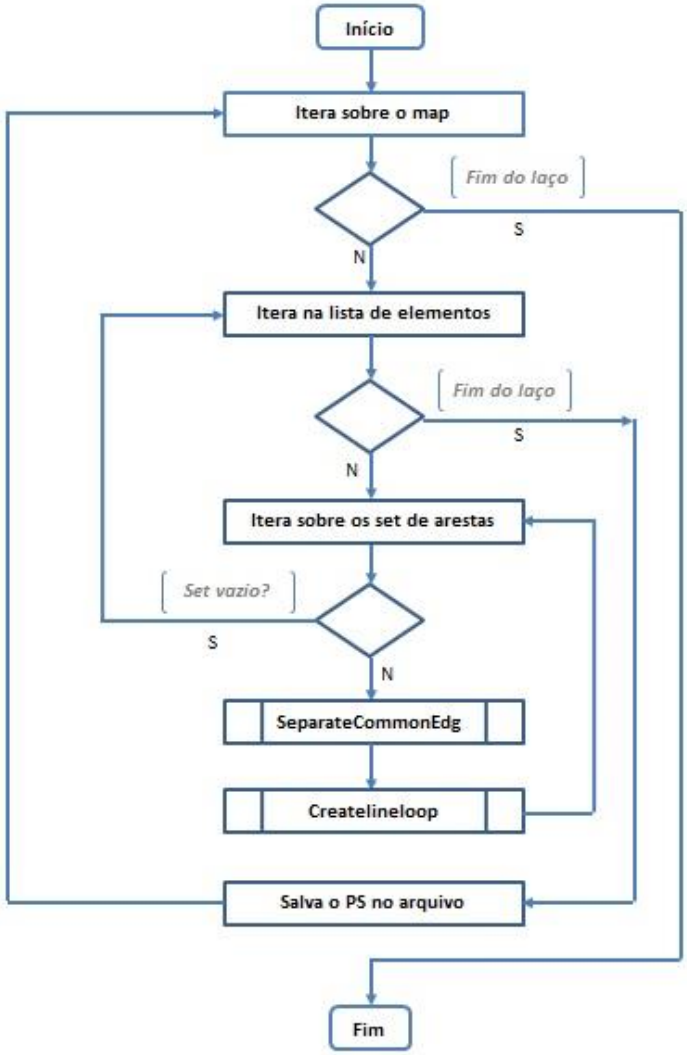
Loop sobre o conjunto de arestas de contorno

Chama a função SeparateCommonEdges

Aciona a função CreateLineLoops

Atribui um *ID* à região e a insere no grupo físico do subdomínio atual

Figura 4.13 - Fluxograma da função SaveGeometry_2D.



Fonte: O autor.

4.7 Geração da nova malha local.

A função MakeMesh_2D apenas procede o acionamento do gerador de malhas a partir do comando:

system("gmsh -2 Geometria2D.geo -o Geometria2D.msh -format msh1 -saveall");

Em resumo:

- Gmsh – O gerador de malhas utilizado
- -2 - Comando para iniciar a geração de malha bidimensional
- Geometria2D.geo – Arquivo de geometria indicado
- Geometria2D.msh – Nome do arquivo de malha a ser gerado pelo Gmsh
- -format msh1 – Formato de malha escolhido.
- -saveall – Ordena a gravação de todos os elementos, independente de grupos físicos indicados na malha.

O *Gmsh* fornece três opções de algoritmos para a geração da malha, o Meshadapt, Delaunay e *Advancing front*. Dos três tipos apenas o algoritmo Delaunay proporciona a opção de controle do grau de refinamento local através de malha de "*background*", sendo por esta razão o algoritmo utilizado neste trabalho. A tabela 4.1 faz uma comparação entre esses algoritmos. Na pontuação o valor 3 representa o melhor desempenho.

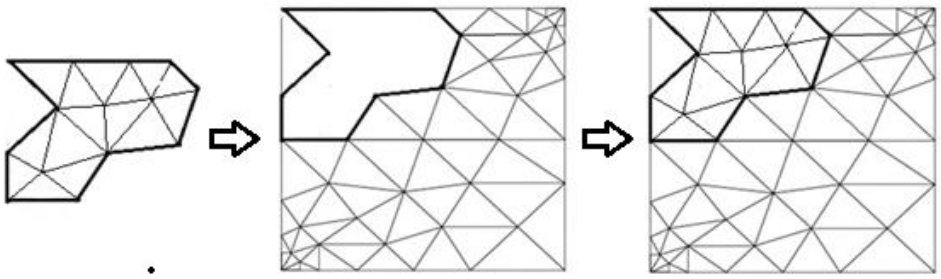
Tabela 4.1 – Comparação de desempenho dos algoritmos.

Algoritmo	**Robustez**	**Eficiencia**	**Qualidade do elemento**
Meshadapt	1	3	2
Delaunay	2	1	2
Avanço frontal	3	2	1

4.8 União das malhas – "merge"

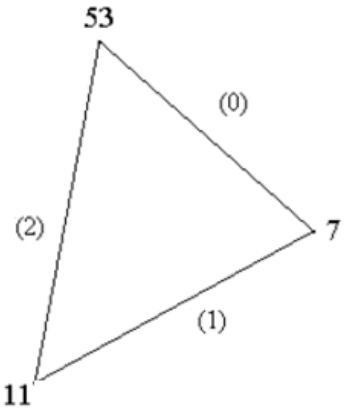
O passo final do processo no adaptador é a união das malhas. Neste processo a malha recém gerada é lida e seus elementos copiados para a malha original. Os primeiros são os vértices, apenas os do contorno não são copiados. Em seguida as faces triangulares com os Id´s somados da quantidade de faces da malha original. As arestas não são copiadas pois são criadas pelo FMDB bastando apenas criar os elementos triangulares com suas conectividades com os vértices. Os comandos para esta operação são os da biblioteca FMDB **M_createVP2** e **M_createFV**. Esta rotina é simples devido à estrutura flexível de dados da malha do FMDB e devido ao fato do *Gmsh* conservar os ID's dos elementos ao criar a malha e ainda à estrutura "de baixa para cima" do *Gmsh (*arestas, faces e volumes são formados por elementos de menor dimensão, no caso das arestas e face os elementos são os vértices), apenas é preciso iterar nos vértices internos copiando-os para a malha original e depois sobre as faces copiando-as integralmente com suas conectividades aos vértices.

Figura 4.14 – Processo de cópia dos elementos triangulares da malha gerada para a malha original.



Fonte: O autor.

Figura 4.15 – Conectividades dos elementos triângulos e arestas do *Gmsh*. Adaptado de Silva (2008).



Conectividades do triângulo
Face (1) = nós { 53, 11, 7 }

Conectividades das arestas
Aresta (0) = nós {53, 7}
Aresta (1) = nós {11, 7}
Aresta (2) = nós {53, 11}

Fonte: Silva, 2008.

5 Resultados

5.1 Introdução

Neste capítulo apresentamos os resultados obtidos na solução de problemas modelo através da utilização das ferramentas desenvolvidas neste trabalho. Serão feitas observações relevantes quando da interpretação dos resultados obtidos.

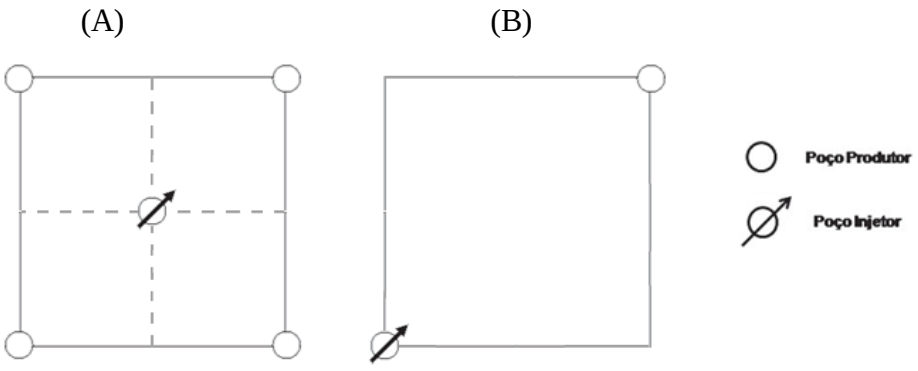
Os problemas estudados serão:

1- Comportamento do campo de pressão em um escoamento monofásico.
2- Comportamento do campo de saturação em um escoamento bifásico em um domínio homogêneo quanto à permeabilidade, usando com o parâmetro de referência a análise com uma malha fina (sem adaptação).
    - Comparação entre processamento com redefinição local de malha e o caso de referência.
    - Acompanhamento da evolução do erro global.
    - Análise e discussão dos processos de interpolação linear, quadrático e adaptativo.
3- Comportamento dos campos de saturação e pressão em um escoamento bifásico em um domínio heterogêneo e isotrópico.
    - Observação do comportamento do escoamento em um domínio heterogêneo quanto heterogêneo quanto à permeabilidade.
    - Comparativos entre métodos de discretização de baixa e alta ordem.
4- Comportamento dos campos de saturação e pressão em um escoamento bifásico em um domínio heterogêneo e anisotrópico.

5.2 Análise do campo de pressão em um escoamento estacionário, monofásico em meio homogêneo e isotrópico.

Para a verificação da robustez e eficiência da nossa metodologia na obtenção do campo de pressão, estudaremos o problema clássico da produção no problema de ¼ do "five-spot" monofásico onde a geometria da malha de injeção e produção é vista na Fig. 5.1.

Figura 5.1 – A) Modelo "five spot" e B) ¼ de "five spot".



Fonte: O autor.

Dada a simetria do problema, só é necessário simular um quarto do reservatório, onde um quadrado de lado unitário descreve a geometria para os casos "1/4 Five Spot". Os resultados foram plotados em uma curva na direção diagonal do reservatório ligando o poço injetor e o produtor.

Sobre as condições de contorno temos fluxo prescrito unitário nos poços injetor e produtor e pressão prescrita nos cantos restantes da geometria do domínio equivalente a zero.

No caso das adaptações a seguir foram usados os parâmetros da tabela 5.1:

Tabela 5.1 – Parâmetros das simulações adaptativas do caso estacionário.

Tolerância cálculo do erro com todos os elementos	0,15
Tolerância cálculo do erro desprezando as singularidades	0,09
Método de interpolação	Adaptativo
Critério de adaptação local	
Refinamento	0,9
Desrefinamento	1,4

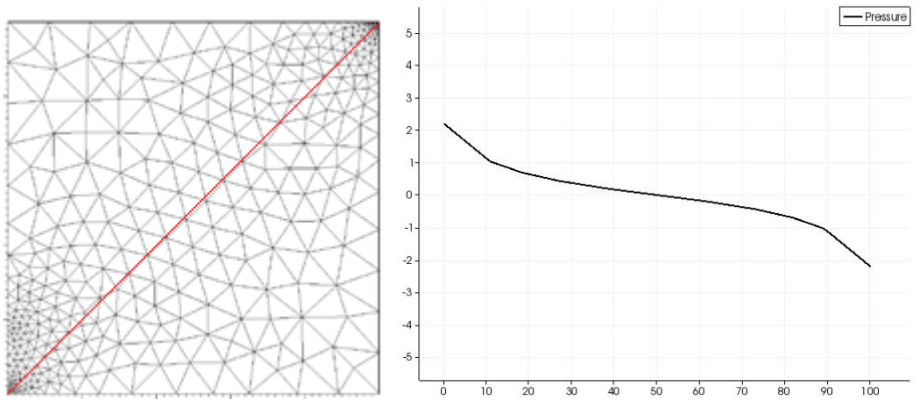
Cabe observar que nos parâmetros de adaptação sob o critério de porcentagem do tamanho do elemento foi escolhido 1,4 para o desrefinamento, sendo este um valor alto com o qual praticamente não haverá aumento do tamanho dos elementos da malha. Este valor é usado em todas as simulações do caso estacionário já que o critério de parada de simulação é a queda do erro a níveis abaixo da tolerância especificada, dessa forma é possível analisar o desempenho do controle do erro global através de refinamento, pode-se então comparar as estratégias de adaptação bem como parâmetros iniciais tais como tamanhos dos elementos da malha inicial.

Sobre as tolerâncias foi mostrado na seção 3.6 que existem dois critérios que a malha adaptada deve satisfazer, o primeiro é o cálculo do erro global considerando todos os elementos da malha e o segundo é o erro global desprezando as regiões de singularidades. Evidentemente a primeira tolerância deve ser maior do que a segunda. Na prática é necessário limitar o tamanho mínimo do elemento nessas regiões haja vista sua influência direta passo de tempo que pode se tornar tão pequeno que inviabilize a análise computacional (LYRA, 1994).

Esta equação estabelece uma relação fixa entre o passo de tempo ("*timestep*") e o tamanho mínimo do elemento e determina o critério de estabilidade.

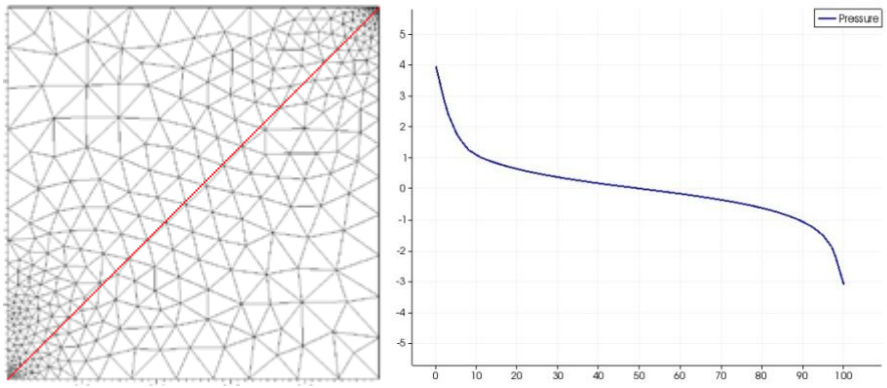
Vemos que pelas figuras 5.2 a 5.4 e pela tabela de resultados do caso estacionário (5.2) o processo automático efetuou o refinamento sucessivo das regiões com singularidades até que o erro nestas regiões tenha caído abaixo da tolerância especificada tendo conseguido a malha final com dois ciclos de adaptação. Resultado este considerado satisfatório.

Figura 5.2 – Resultado estacionário com a malha inicial.



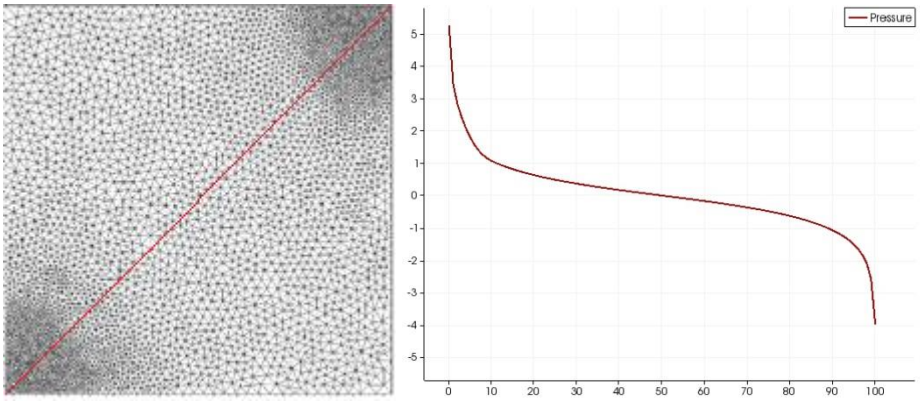
Fonte: O autor.

Figura 5.3 – 1ª adaptação e resultados do caso estacionário.



Fonte: O autor.

Figura 5.4 – Resultado após 2ª e última adaptação.



Fonte: O autor.

Tabela 5.2 – Resultados do caso estacionário.

	Malha Inicial	2ª malha	Malha Final
Triângulos	228	2.371	8.505
Arestas	360	3.621	12.853
Nós	133	1.251	4.349
Erro global excluindo singularidades	0,448	0,141	0,087

5.3 Comportamento do campo de saturação em um escoamento bifásico em meio homogêneo isotrópico

Como estamos em processo de teste do adaptador por redefinição, primeiramente simulamos um caso sem adaptação onde fizemos uso de uma malha com tamanho de elemento tal que o erro global foi mantido abaixo de 0,35 ao longo de toda a análise transiente efetuada, a saber o tamanho constante dos elementos usado foi 0,04, este caso será visto como referência para avaliação do desempenho do adaptador.

Observar que neste caso o erro final é maior que o inicial por não haver processo de adaptação. O erro final é utilizado como tolerância dos processos adaptativo a seguir que devem mantê-lo abaixo deste valor.

O caso apresentado a seguir é um estudo de um escoamento bifásico em meio poroso (1/4 five spot), homogêneo e isotrópico.

Para os casos de simulação de escoamentos bifásicos a saturação inicial de água é zero em todo o domínio e a saturação irredutível de água e a residual de óleo também valem zero. Além disso, temos a saturação de água unitária no poço injetor.

O modelo de permeabilidade relativa é o de Brooks-Corey onde os expoentes para a água e o óleo valem $n_w = n_o = 2$. O tensor de permeabilidade para este caso isotrópico é:

$$K = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Tabela 5.3 – Resultados da simulação do caso de referência bifásico em meio homogêneo e isotrópico.

Quantidade de faces da malha:	26.516
Quantidade de nós:	13.459
Tempo total de simulação	20 min 27,56 s
Erro global (todos os elementos) inicial:	0,125
Erro global (todos os elementos) final:	0,354.

No caso transiente comparamos os resultados entre a técnica de adaptação por redefinição local e um caso de referência sem adaptação onde acompanhou-se a evolução do erro e o máximo alcançado foi usado como tolerância do caso adaptativo. O objetivo é mostrar que o processo com adaptação pode manter o erro sob controle e proporcionar melhor desempenho computacional, ou seja, menos tempo dispendido na análise. A tabela 5.4 apresenta os parâmetros da simulação que foram adotados.

Tabela 5.4 – Parâmetros da simulação do caso adaptativo homogêneo e isotrópico.

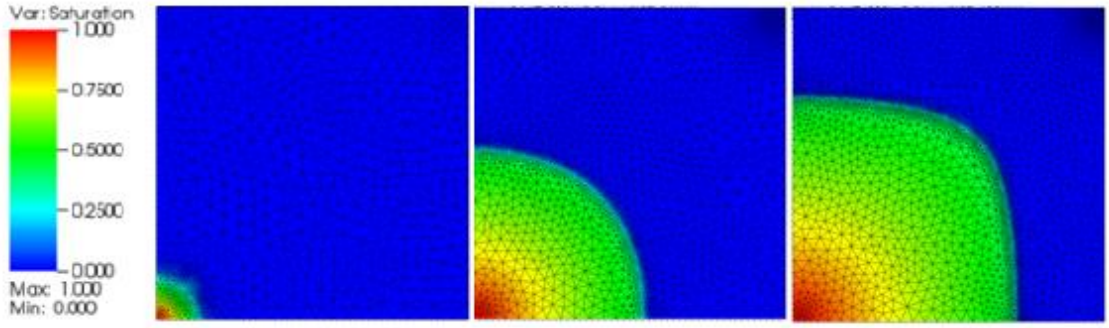
Quantidade de faces:	1.054
Quantidade de nós:	568
Tolerância para todos os elementos:	0,35
Tolerância para todos os elementos exceto as regiões de singularidades	0,3
Tempo de simulação para a conclusão em VPI	0,1
Saturação inicial de água	0
Saturação irredutível de água	0
Saturação residual de óleo	0
Expoente de Brooks-Corey para a água n_w	2
Expoente de Brooks-Corey para o óleo n_o	2
Tempo total de simulação	2 min 48 s
Erro global (todos os elementos) inicial:	0,448
Erro global (todos os elementos) final:	0,308

Tabela 5.5 – Parâmetros da simulação usados em todas as análises.

Tempo de simulação para a conclusão em VPI	0,1
Saturação inicial de água	0 (zero)
Saturação irredutível de água	0 (zero)
Saturação residual de óleo	0 (zero)
Expoente de Brooks-Corey para a água n_w	2
Expoente de Brooks-Corey para o óleo n_o	2

O resultado no regime transiente muda com o passar do tempo, a região com gradientes elevados varia sua posição, demonstrando o avanço da frente de saturação de água. Assim o refinamento da malha deve acompanhar este movimento. O erro global apresenta um contínuo aumento até o limite especificado pela tolerância e uma queda brusca quando da adaptação, e mantém este comportamento até o fim da simulação. Da mesma forma que na simulação da pressão, espera-se que para este caso haja um ganho considerável no desempenho do simulador quanto ao tempo necessário para concluir os cálculos.

Figura 5.5 – Estudo transiente do avanço da saturação de água com adaptação por redefinição local.

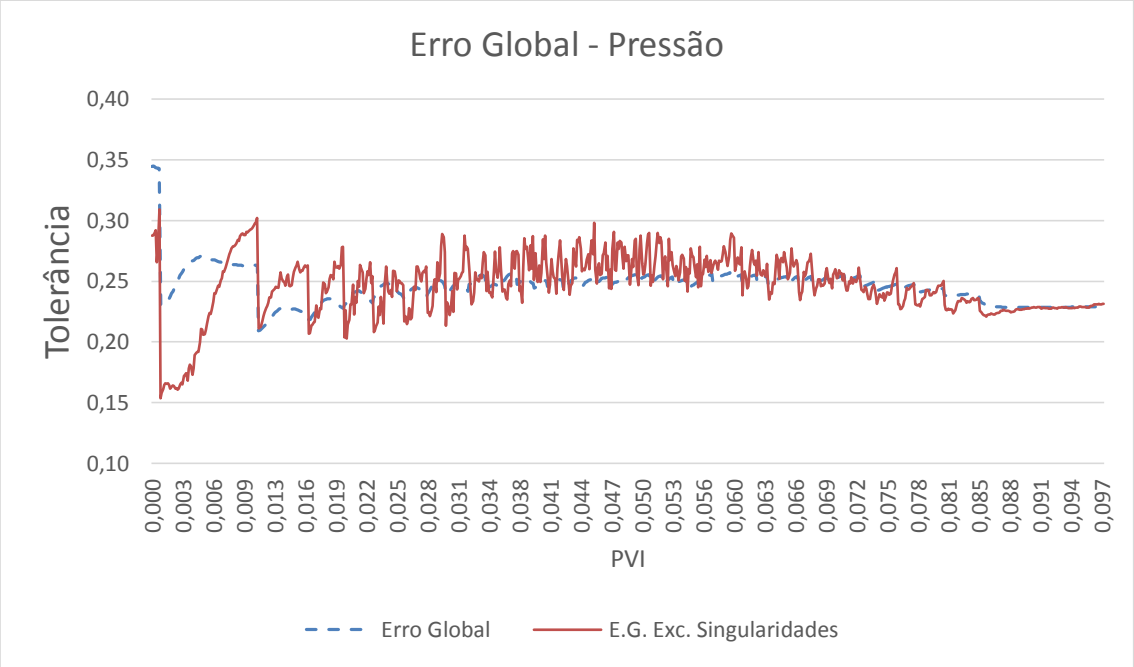


Fonte: O autor.

A escala de cores na fig. 5.5 apresenta os valores de saturação de água, assim o resultado apresenta a queda da quantidade de água na região de interface entre os fluidos. Observamos que a região refinada da malha acompanha perfeitamente as regiões de maior gradiente como é esperado.

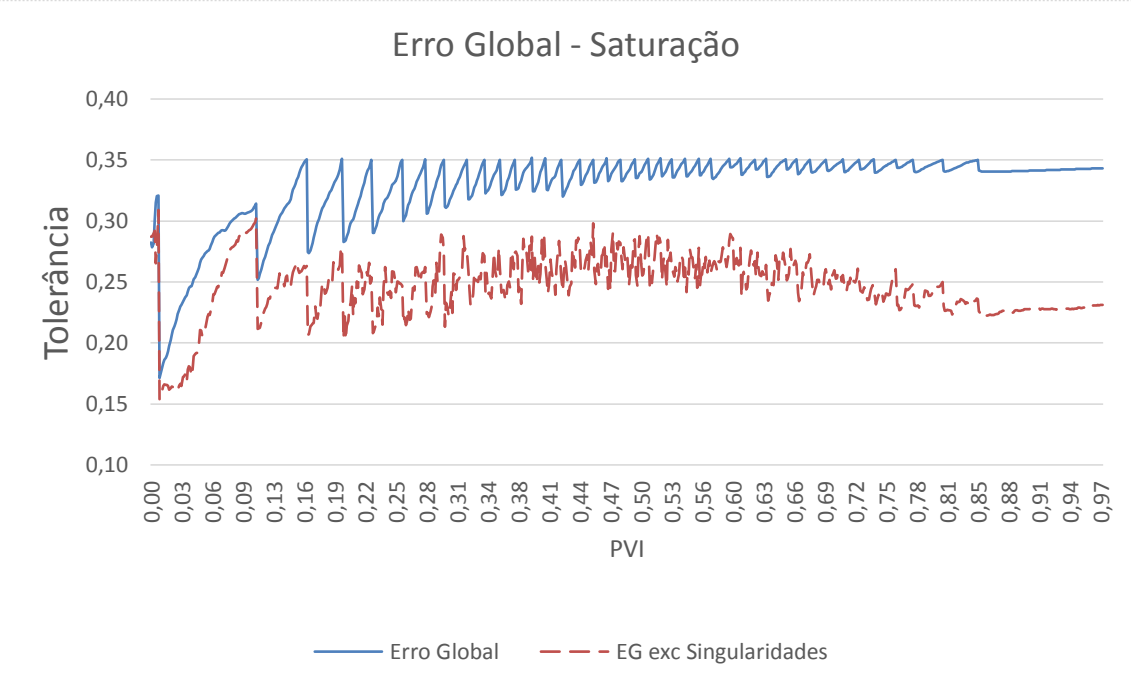
Vemos da análise da evolução do erro global mostrada nas figuras 5.6 e 5.7 que o processo adaptativo automático manteve o erro abaixo da tolerância exigida.

Figura 5.6 – Evolução do erro global com e sem singularidades para o campo de pressão.



Fonte: O autor.

Figura 5.7 – Evolução do erro global com e sem singularidades para o campo de saturação de água.

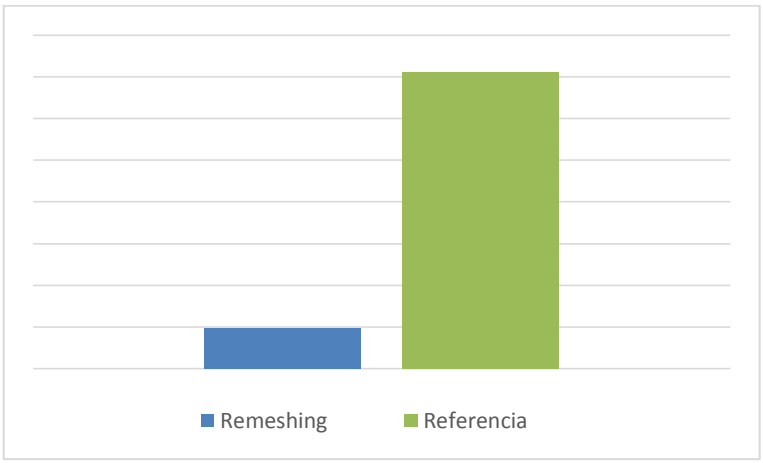


Fonte: O autor.

As figuras 5.6 e 5.7 mostram que o processo de adaptação foi claramente controlado pelo erro referente aos resultados da saturação de água, este resultado é razoável visto que os resultados para a pressão variam pouco no decorrer da análise.

O processo adaptativo consumiu cerca de 15% do tempo necessário para obtenção do resultado alcançado com uma malha fina que dispense adaptações conforme observado na figura 5.8.

Figura 5.8 – Comparação do tempo necessário para a conclusão da simulação com e sem adaptação automática.

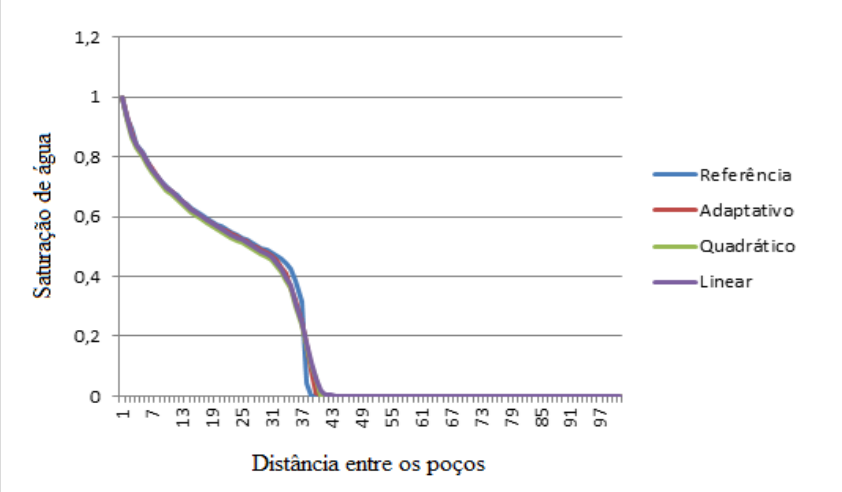


Fonte: O autor.

Utilizamos a estratégia de redefinição local para evidenciar os resultados dos diferentes tipos de interpolação de dados entre malhas para a mesma simulação, a saber interpolação linear, interpolação quadrática e interpolação adaptativa descritos no capítulo 3.

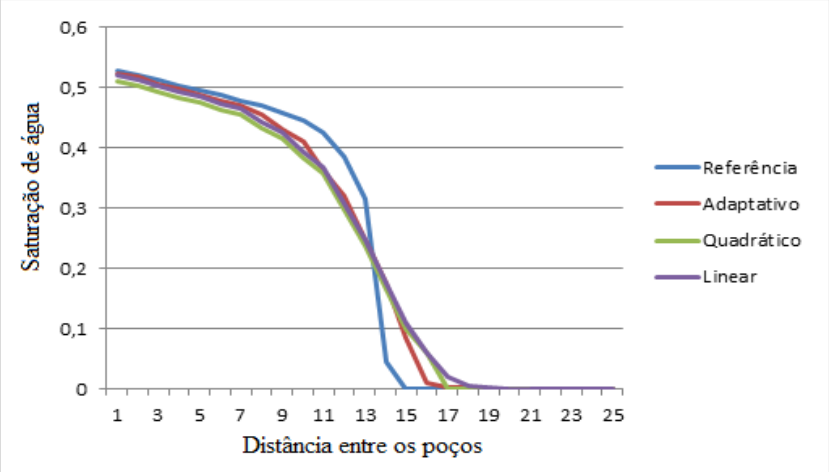
Observamos que em todos os casos os resultados apresentaram-se um pouco difusivos em relação ao caso de referência. Contudo observamos que a estratégia de interpolação adaptativa apresentou o melhor resultado dentre as três técnicas testadas, o resultado apresentado na figura 5.9 e ampliada na figura 5.10 confirmou esse resultado da saturação evidenciando o impacto na acurácia dependente do tipo de técnica de interpolação. A partir de agora para os próximos casos em estudo será utilizada a técnica de interpolação adaptativa.

Figura 5.9 – Avanço da saturação de água de diferentes estratégias de interpolação PVI 0,09.



Fonte: O autor.

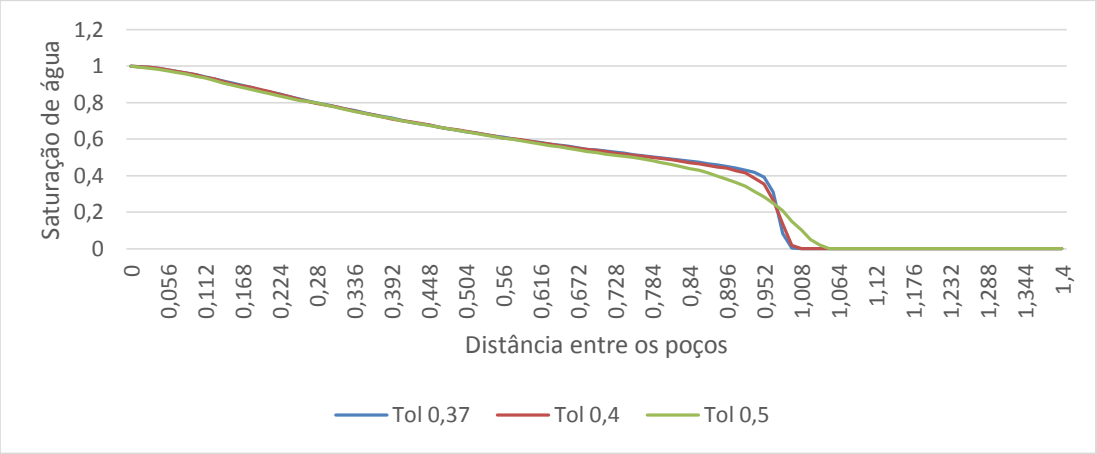
Figura 5.10 – Ampliação da frente de saturação (fig. 5.9) para diferentes técnicas de interpolação PVI 0,09.



Fonte: O autor.

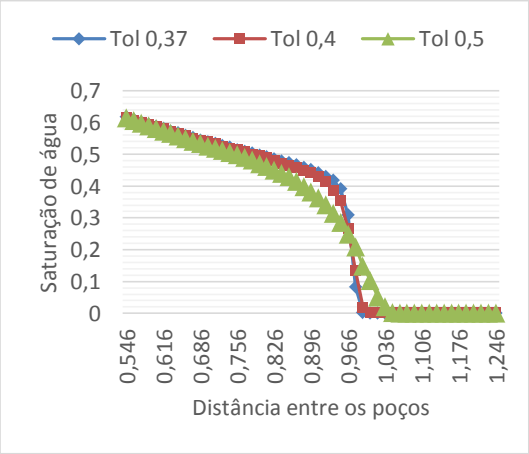
Os resultados apresentados nas fig. 5.11 e 5.12 são as curvas de saturação de água para três tolerâncias diferentes. O tamanho local dos elementos seguirá proporcionalmente os limites de tolerâncias adotados sendo menor nas regiões de gradientes acentuados. É possível observar claramente esse resultado na fig. 5.14 para cada uma das tolerâncias adotadas. As curvas de valores instantâneos da saturação de água, deixam claro a magnitude da difusão numérica. Alguns fatores influem diretamente no tempo, menor tamanho de elementos significam mais elementos para calcular a solução, bem como um "*timestep*" menor. Dessa forma teremos mais elementos que deverão ter sua solução calculada uma quantidade maior de vezes, sob pena de perder estabilidade da solução. Isso explica por que o tempo gasto para finalizar a simulação cresce desproporcionalmente à redução da tolerância como pode ser observado na fig. 5.13. A saber as tolerâncias utilizadas foram 0,37, 0,4 e 0,5.

Figura 5.11 – Curvas de saturação para o caso homogêneo isotrópico PVI 0,08.



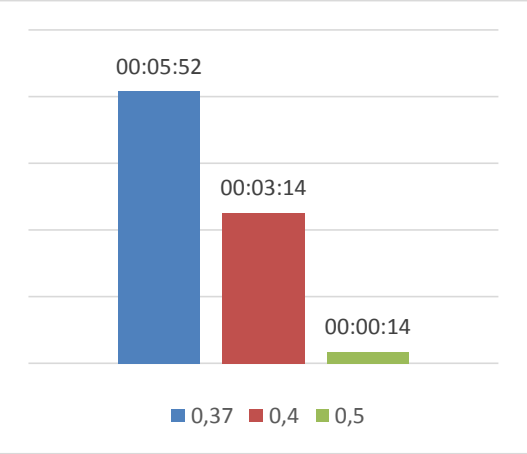
Fonte: O autor.

Figura 5.12 – Ampliação da fig. 5.11.



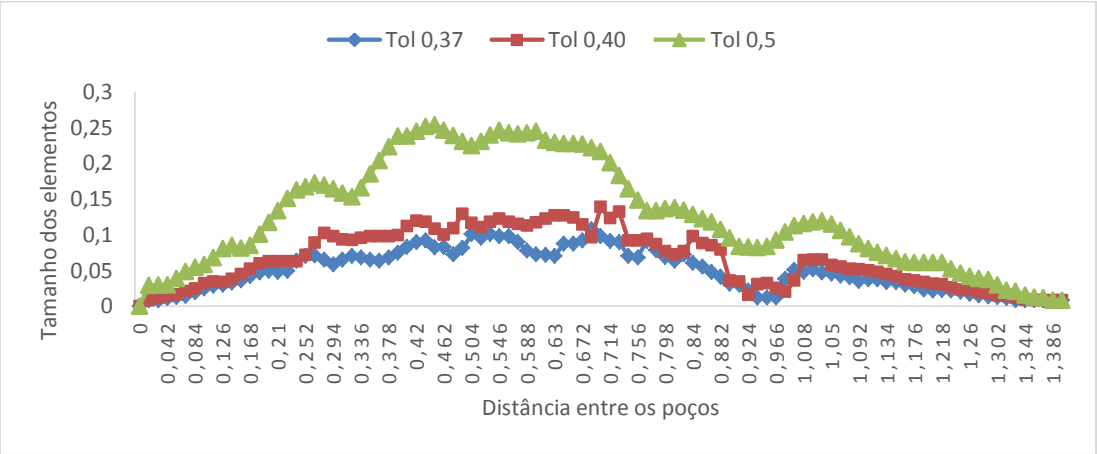
Fonte: O autor.

Figura 5.13 – Tempo para conclusão da análise.



Fonte: O autor.

Figura 5.14 – Tamanhos locais dos elementos da malha adaptada – CL´s.



Fonte: O autor.

No estudo seguinte fizemos a simulação de dois casos homogêneos isotrópicos, utilizando uma aproximação do método de volumes finitos de primeira ordem em comparação com uma aproximação de segunda ordem.

Claramente da comparação dos gráficos 5.16 e 5.14 vemos o efeito de difusão numérica associado ao tamanho dos elementos em casos com maiores valores de tolerância e do gráfico 5.15 o efeito do tamanho do elemento impactando diretamente no tempo necessário para conclusão da análise a partir do cálculo do "*timestep*" visto na eq. 5.1.

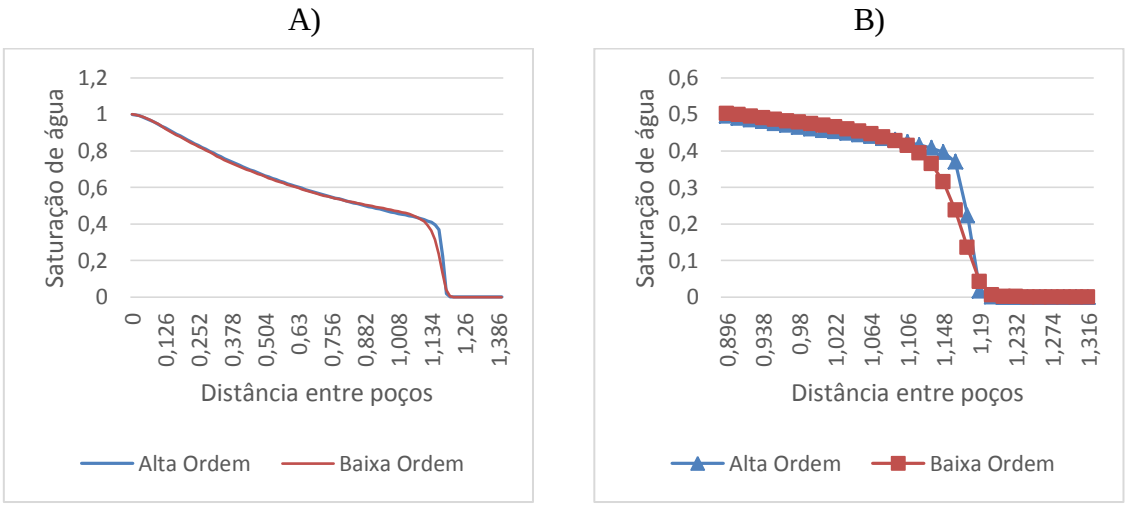
Seguem os parâmetros utilizados:

Tabela 5.6 – Parâmetros para as simulações com formulações de baixa e alta ordem.

Tolerância cálculo do erro com todos os elementos	0,35
Tolerância cálculo do erro desprezando as singularidades	0,3
Método de interpolação	Adaptativo
Refinamento	0,9
Desrefinamento	1,1
CFL Baixa Ordem	0,9
CFL Alta Ordem	0,48

Agora os gráficos de saturação para cada formulação, Fig. 5.15 onde podemos constatar maior difusão para a simulação com baixa ordem (upwind de 1ª ordem) quando comparado com formulação de alta ordem.

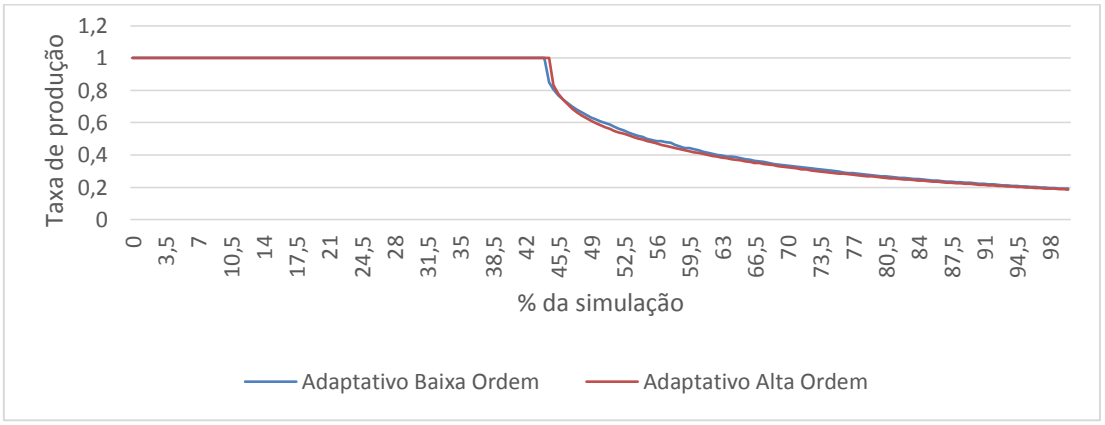
Figura 5.15 – A) Curvas de saturação de água para aproximações de alta e baixa ordem com aproximações de alta e baixa ordem PVI 0,08 B) Ampliação na região de interfacó-leo-água.



Fonte: O autor.

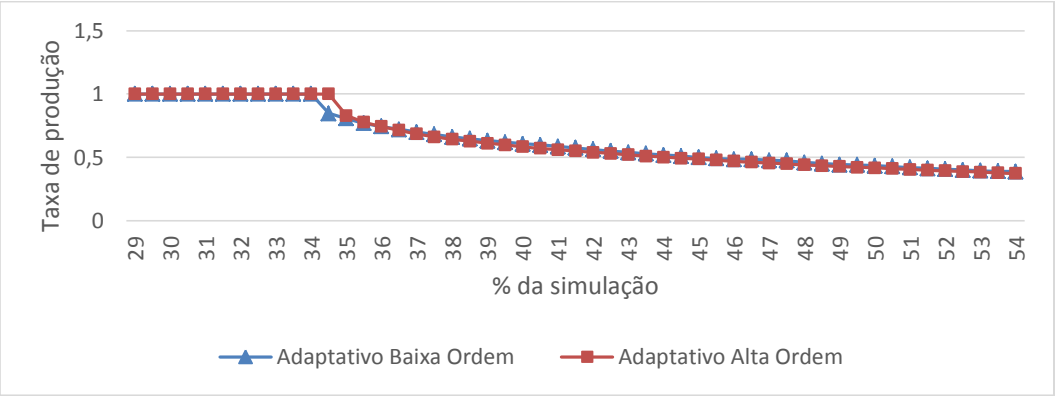
O impacto da difusão na formulação de baixa ordem do simulador pode ser observado na curva de produção de óleo (fig. 5.16 a 5.18) onde o "*breakthrough*", ponto de início de produção de água no poço produtor e acentuada queda na produção de óleo, ocorre antes do que nas formulações de alta ordem. Contudo nos casos com adaptação praticamente não se observa diferença na acurácia dos resultados mesmo utilizando-se a mesma tolerância para as simulações com alta ou baixa ordem.

Figura 5.16 – Taxa de produção de óleo formulações de alta ordem e baixa ordem.



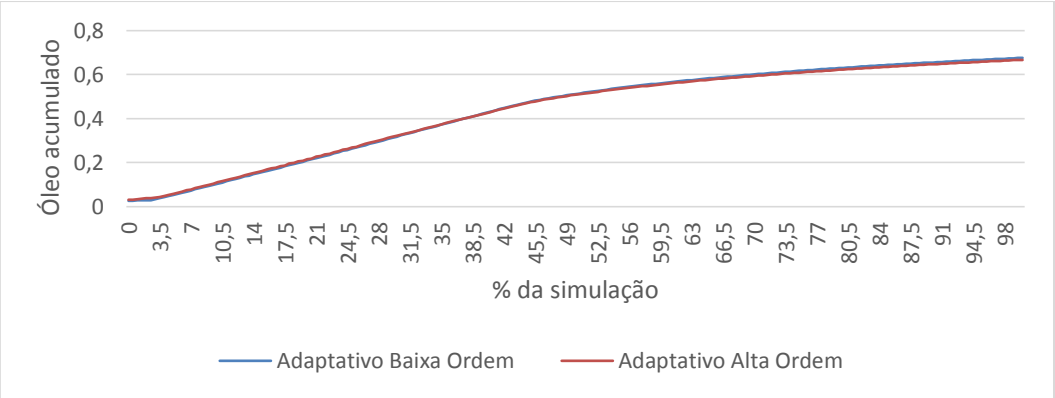
Fonte: O autor.

Figura 5.17 – Ampliação da fig. 5.16 antecipação do “breakthrough” na baixa ordem.



Fonte: O autor.

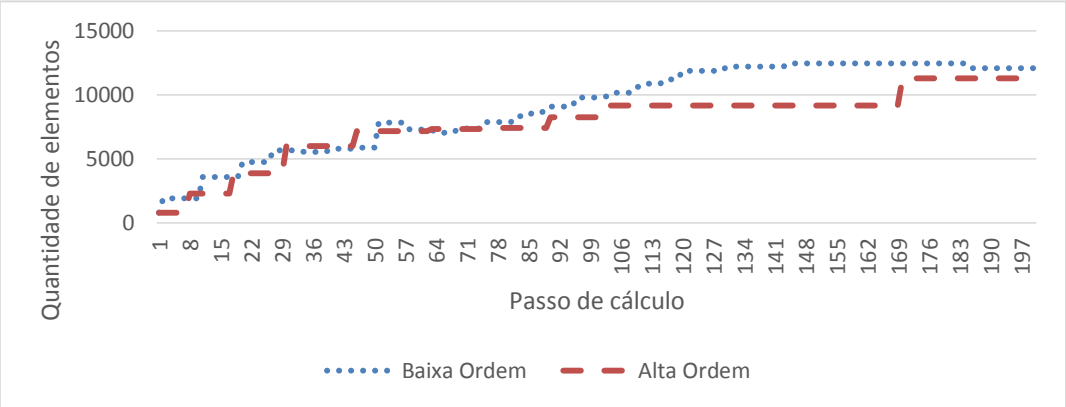
Figura 5.18 – Produção total de óleo acumulado formulações de alta e baixa ordem.



Fonte: O autor.

É de se esperar que em problemas com formulação de alta ordem seja requerida adaptações com dimensão dos elementos maior e consequentemente menos elementos na malha. Mantivemos o registro da quantidade de elementos durante a simulação e pudemos constatar esse resultado. Os gráficos da fig. 5.19 apresentam esses números para cada porcentagem de simulação concluída.

Figura 5.19 – Quantidade de elementos a cada adaptação em baixa e alta ordem.



Fonte: O autor.

5.4 Simulação de escoamento em meio heterogêneo isotrópico

O caso é o de um domínio quadrado de lado 1 com um subdomínio concêntrico e de baixíssima permeabilidade. O caso apresentado objetiva verificar a robustez do software adaptador em um processo de adaptação heterogêneo onde deve preservar as geometrias dos domínios. As propriedades de permeabilidade de cada domínio estão apresentadas nas eq. 5.1 e 5.2.

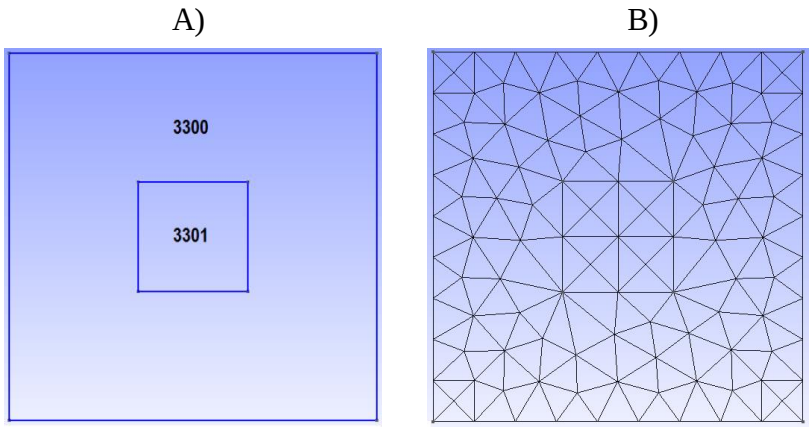
Como foi realizado em todas as análises neste estudo, temos para geometria um domínio quadrado de lado 1, representando a quarta parte de uma malha de injeção "*five spot*" como mostram a fig. 5.20 e os tensores de permeabilidade para as duas regiões são apresentados abaixo.

Parâmetros físicos e geométricos:

$$K_{3300} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (5.1)$$

$$K_{3301} = \begin{bmatrix} 10^{-6} & 0 \\ 0 & 10^{-6} \end{bmatrix} \quad (5.2)$$

Figura 5.20 – A) Geometria do problema heterogêneo isotrópico B) Malha inicial



Fonte: O autor.

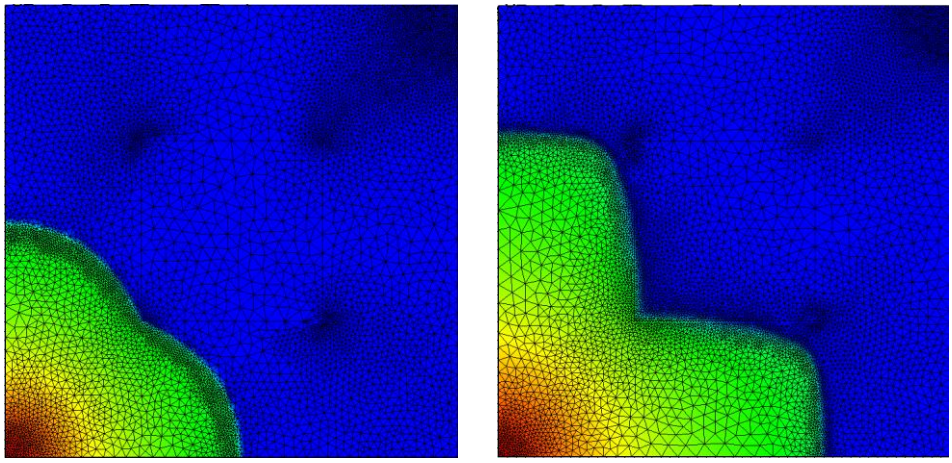
Os parâmetros principais utilizados nesta simulação estão apresentados na tabela 5.7 e os resultados para a saturação são os destacados nas fig. 5.21 e 5.22, na fig. 5.22 está a malha adaptada para a solução com PVI 0,45.

Tabela 5.7 – Parâmetros para a simulação heterogênea isotrópica.

Tolerância cálculo do erro com todos os elementos	0,35
Tolerância cálculo do erro desprezando as singularidades	0,3
Método de interpolação	Adaptativo
Refinamento	0,9
Desrefinamento	1,1
CFL Alta Ordem	0,48
PVI	0,45

Figura 5.21 – Avanço da saturação de água A) PVI 0,2 B) PVI 0,3.

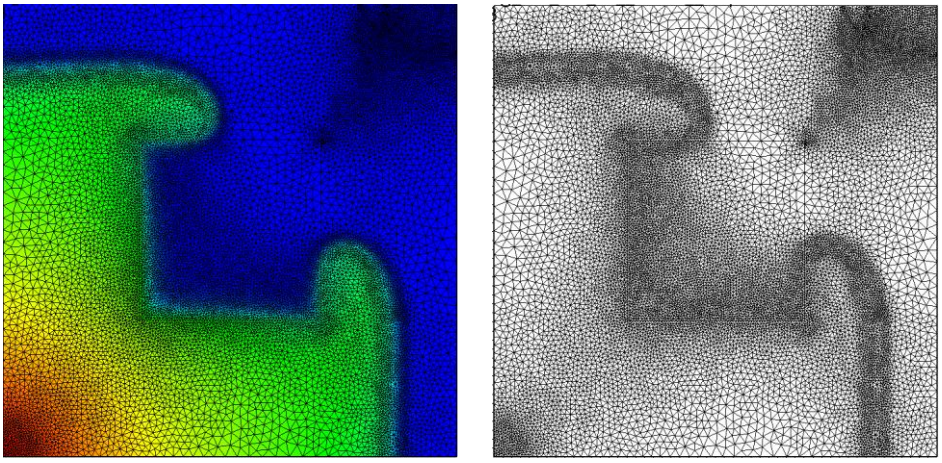
A) B)



Fonte: O autor.

Figura 5.22 – A) Avanço da saturação de água PVI 0,45 B) Malha adaptada aos resultados no PVI 0,45

A) B)



Fonte: O autor.

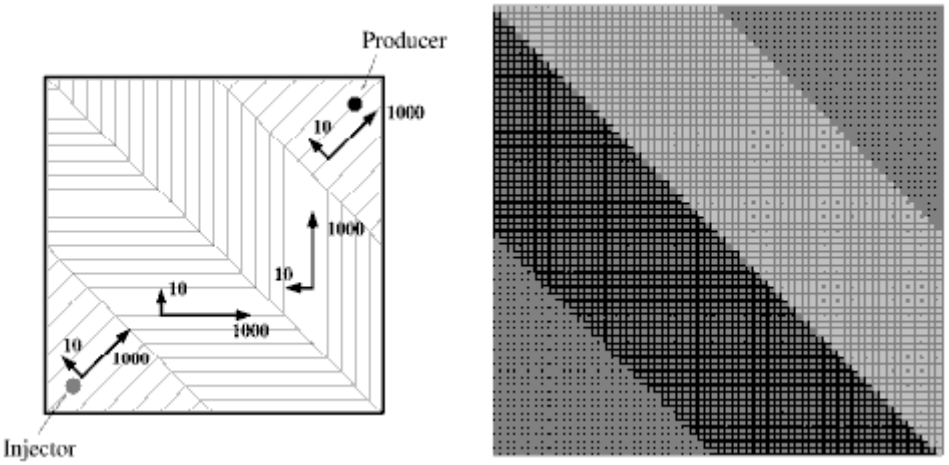
Esta simulação exigiu sucessivas adaptações utilizando a análise de erro para as variáveis de pressão e de saturação, vemos claramente o refinamento acompanhando perfeitamente as regiões de maiores gradientes de pressão e de saturação de água.

Este caso comprova a robustez do processo de adaptação quanto ao requisito de modificar a malha enquanto conserva os limites geométricos e acompanhamento do fenômeno transiente, demonstrando a capacidade de fornecer um processo adaptativo que conserve a qualidade e acurácia ao mesmo tempo que proporciona um aumento significativo da eficiência computacional.

5.5 Simulação de escoamento em meio heterogêneo anisotrópico

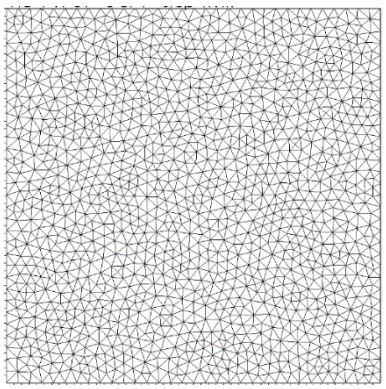
O caso que vamos estudar agora possui 4 subdomínios e anisotropia na permeabilidade e foi adaptado do problema apresentado por Kirill Nikitin (2010). A adaptação consiste no uso de malhas triangulares no lugar da quadrangular original.

Fig 5.23 Problema heterogêneo e anisotrópico.



Fonte: Nikitin, 2010.

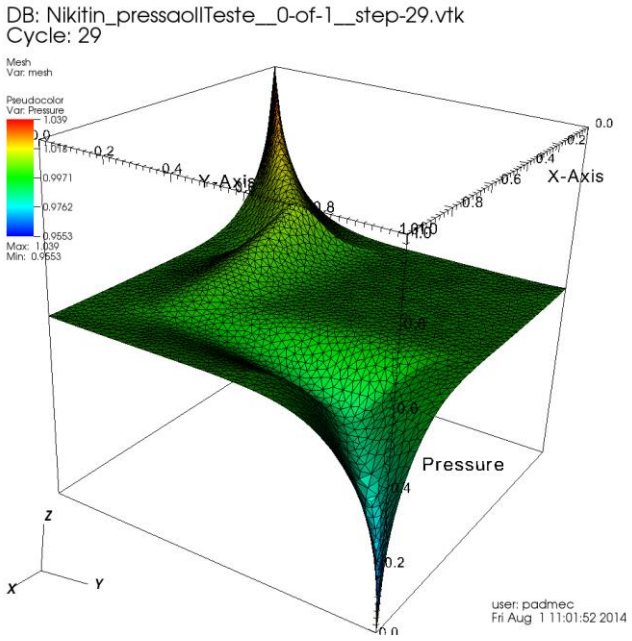
Fig. 5.24 Malha inicial triangular.



Fonte: O autor.

A adaptação vinha utilizando como critério o erro avaliado a partir dos resultados dos gradientes de pressão e saturação. Esses gradientes são calculados a partir uma formulação IMPES com o campo de saturação e de velocidades fornecidos pelas fórmulas (2.1) e (2.3) o fluxo (velocidade) está linearmente relacionado ao tensor permeabilidade, sendo este uma característica do subdomínio, de tal forma que existe uma descontinuidade da função gradiente de pressão entre subdomínios. Esta descontinuidade pode ser visualizada no campo de pressões exposto na Fig 5.25.

Fig. 5.25 Campo de resultados de pressão.



Fonte: O autor.

Pela equação (3.3) observamos que a adaptação baseada na análise de erros para o campo de pressão levará a resultados notadamente diferentes. Em simulações para problemas com meios heterogêneos se utlizarmos o gradiente de presões as regiões de adaptação para adaptações baseadas no critério da hessiana do campo de pressão tendem a ficar restrita aos limites dos domínios que contêm a interface óleo-água. Fato este que implica em problemas relativos à formação dos elementos triangulares nas interfaces entre domínios. Quando apenas um domínio sofre refinamento os elementos do contorno deste domínio têm que se adaptar ao novo tamanho requerido e ao contorno original da região afim de conformar-se à malha previamente existente. Isso pode levar à geração de faces triangulares "achatadas" ou "*bad*" *element* (MIRANDA, 1999).

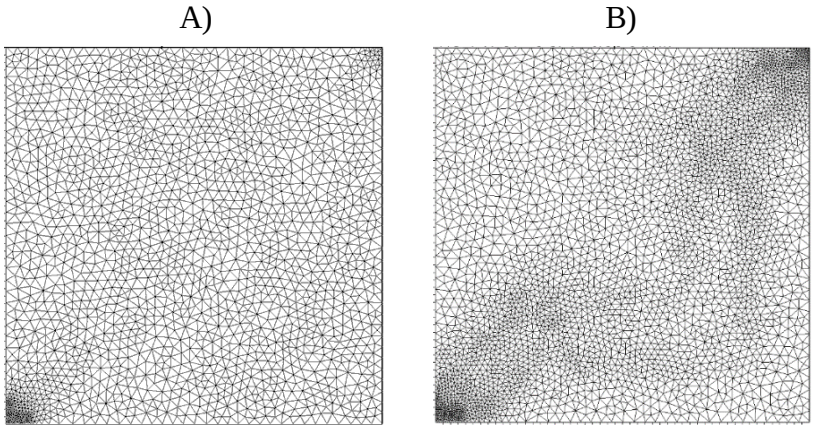
Os parâmetros utilizados nesta simulação estão resumidos na tabela 5.7

Tabela 5.7 – Parâmetros para a análise do problema em meio heterogêneo anisotrópico.

Tolerância cálculo do erro com todos os elementos – Pressão e Saturação	0,25
Tolerância cálculo do erro desprezando as singularidades – Pressão e Saturação	0,2
Método de interpolação	Adaptativo
Refinamento	0,9
Desrefinamento	1,2
CFL Alta Ordem	0,48
PVI	0,1

O exemplo na figura 5.27 (a) mostra a primeira adaptação referente ao campo de pressão. Embora haja gradientes de pressão em todos os domínios (fig. 5.25), a adaptação praticamente não captura o resultado. Em contrapartida a adaptação utilizando como parâmetro o fluxo, procedeu o refinamento "prevendo" o caminho que a fase injetada faz através do domínio.

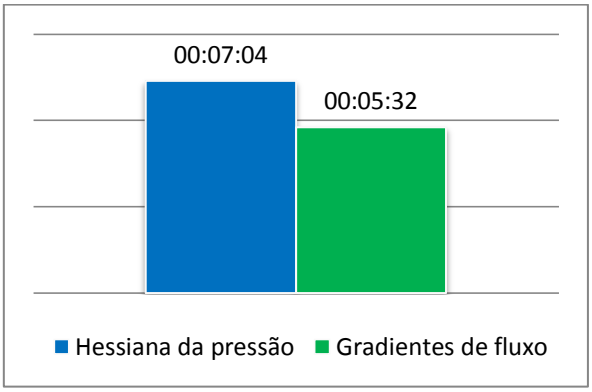
Fig. 5.26 A) 1ª adaptação pelo critério da hessiana da pressão B) 1ª adaptação pelos gradientes do fluxo.



Fonte: O autor.

A seguir são apresentados os resultados para os dois critérios de análise de erros citados. O caso baseado na hessiana da pressão completou apenas 10% da simulação e foi encerrado ao apresentar resultados incoerentes (refinamento excessivo em todo o domínio). A simulação com análise de erro baseada no fluxo chegou a ser concluída e seus resultados são apresentados nas imagens a partir da fig. 5.29, tendo cumprido a análise com maior eficiência como pode ser visualizado na fig. 5.27 (aproximadamente 78% do tempo gasto pela análise pela hessiana da pressão nos primeiros 10% de simulação).

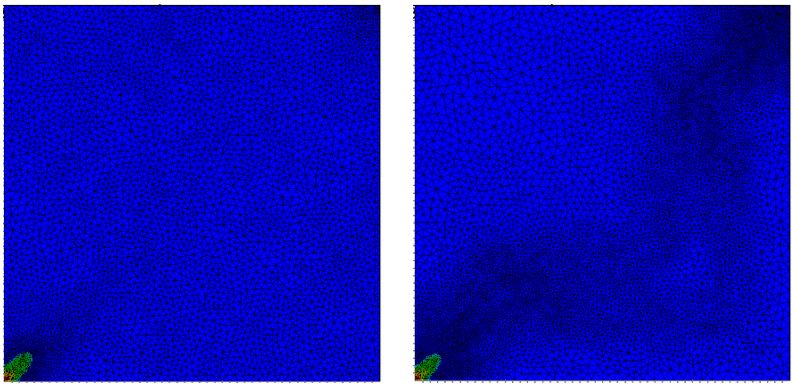
Fig. 5.27 Comparação dos tempos despendidos na simulação com critérios de adaptação pela hessiana da pressão e pelos gradientes de fluxo.



Fonte: O autor.

Fig. 5.28 Resultados para a saturação de água PVI 0,01 A) Adaptação pela hessiana da pressão B) adaptação pelos gradientes do fluxo.

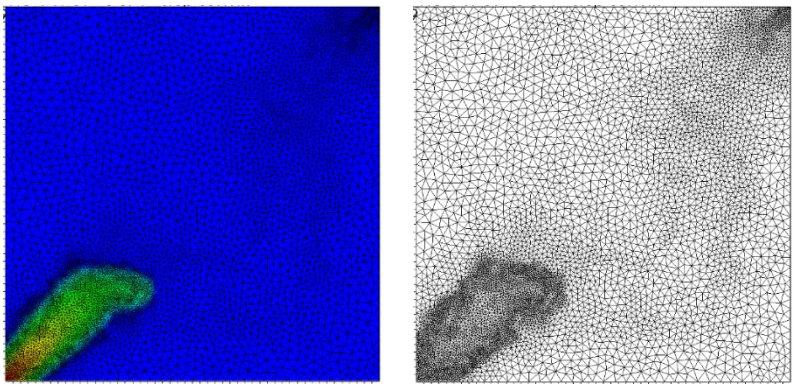
A) B)



Fonte: O autor.

Fig. 5.29 Resultados para a saturação de água PVI 0,03 A) Resultados de saturação de água com adaptação pelos gradientes do fluxo B) Malha adaptada resultante.

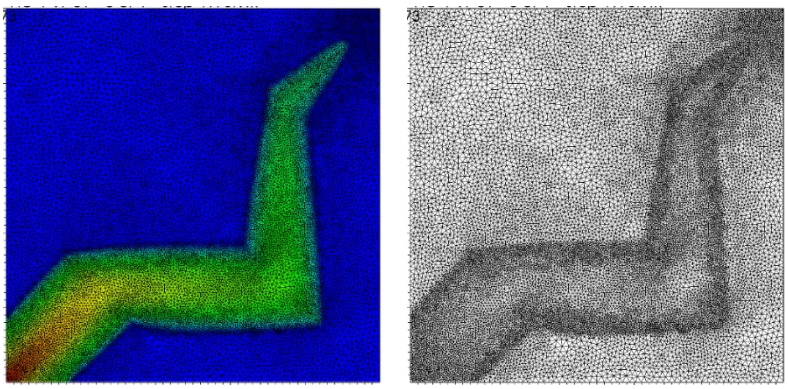
A) B)



Fonte: O autor.

Fig. 5.30 Resultados para a saturação de água PVI 0,08 A) Resultados de saturação de água com adaptação pelos gradientes do fluxo B) Malha adaptada resultante.

A) B)



Fonte: O autor.

Em resumo, a adaptação baseada na análise de erros em relação ao fluxo proporcionou uma malha mais adequada à captura do escoamento, procedendo o refinamento adaptativo em regiões de interesse, deixando menos refinada aquelas regiões com baixos gradientes o que nos leva a concluir que para meios com média a alta heterogeneidade é mais adequado utilizar a velocidade para captura do fenômeno físico de locomoção dos fluidos.

6 Conclusões

Nesse trabalho foi elaborada e implementada uma ferramenta numérica de adaptação de malhas por redefinição global ou local, capaz gerar malhas bidimensionais não estruturadas de elementos triangulares, flexível para atender a necessidade local em qualquer região da malha a ser determinada através de indicadores de erro para os campos de pressão e de saturação de água. O programa desenvolvido foi analisado na simulação numérica de escoamento de fluidos em meios porosos.

Para a implementação dessas estratégias se fez necessária a integração e utilização de diversos componentes numéricos, sendo eles o simulador em volumes finitos, o indicador de erros, o gerador de malhas e o interpolador, agindo o adaptador como ponte de integração entre essas ferramentas. fornecendo assim, sempre que necessário, a malha mais apropriada aos resultados parciais do fenômeno transiente ou estacionário e ainda proporcionando a otimização dos recursos de processamento do hardware aumentando a eficiência computacional.

O adaptador foi capaz de unificar várias outras já anteriormente desenvolvidas em programas da própria UFPE ou softwares open-source tais como FMDB, Madlib, *Gmsh*.. Esta ferramenta provou ser capaz de refinar ou desrefinar a malha localmente ou globalmente sempre a avaliação de erros indicar a necessidade.

Esta ferramenta foi testada na análise de casos modelos de escoamentos em meios porosos heterogêneos e anisotrópicos, onde ficou demonstrada a robustez do software de adaptação local ou global. Os exemplos adotados testam as ferramentas desenvolvidas e integradas em um único pacote e ilustram a aplicabilidade dos diversos procedimentos adaptativos, demonstrando ainda o potencial da utilização das estratégias adaptativas na simulação numérica de reservatórios de petróleo.

Os resultados do macro-algoritmo demonstraram que o ganho de eficiência alcançado foi da ordem de 10 vezes. Consideramos, dessa forma, que o objetivo de aumento da eficiência na utilização dos recursos computacionais ao mesmo tempo que garante acurácia e convergência da solução aproximada foi alcançado.

Ainda o adaptador desenvolvido em todos os testes conseguiu modificar a malha localmente no interior dos subdomínios e nas regiões de fronteira entre subdomínios e linites da geometria do domínio global sem perda de informação geométrica, mostrando-se uma ferramenta versátil de adaptação para qualquer geometria dada.

6.1 Trabalhos futuros

Aprimorar o módulo de adaptação para eliminar processos de gravação em arquivos aumentando a eficiência do módulo.

Utilizar o software *Gmsh* como biblioteca de funções, evitando o acionamento do programa como software durante a execução da simulação. Esta medida deve evitar o acionamento de rotinas desnecessárias do *Gmsh* implicando em melhoria do desempenho geral.

Adaptar o código para o uso de computação paralela.

Estender o processo de adaptação para o uso de malhas mistas (quadriláteros-triângulos-2D) e para lidar com modelos tridimensionais.

Explorar outras técnicas de adaptação, bem como, estimadores de erro e adaptação com malhas anisotrópicas.

Utilizar outros critérios para o procedimento adaptativo, tais como a análise temporal dos resultados para a captura do fenômeno no transcorrer da análise transiente.

REFERÊNCIAS

ARAÚJO F. S., Procedimentos adaptativos na simulação de reservatórios de petróleo em volumes finitos, Dissertação de Mestrado-Engenharia Civil – UFPE, 2004.

BARREIROS J. O., Utilização de elementos Finitos de Equilíbrio em Refinamento Adaptativo, Tese de Doutorado – UTL, 2006.

BOUSSETTA R., COUPEZ T., FOURMENT L., Adaptive Remeshing Based on a Posteriori Error Estimation, 2006.

BRENNER K., Hybrid finite volume scheme for two-phase flow in porous media, 2012.

CARVALHO, D. K. E., LYRA, P. R. M. Uma Formulação do Método dos Volumes Finitos com Estrutura de Dados por Aresta para a Simulação de Escoamentos em Meios Porosos, Tese de Doutorado – Engenharia Civil –UFPE, 2005;

CEZANO M. T., CARVALHO D. K., Utilização de técnicas de adaptação de malhas para a simulação numérica de escoamento em meios porosos, CONIC, 2008.

COMPASSO A. F., LYRA, P. M., SILVA R. S., CARVALHO D. K. E., SILVA E. O., Adaptação tipo h aplicada à simulação de reservatórios de petróleo 2d usando programação orientada a objetos em C++, VI Congresso Nacional de Engenharia Mecânica, 2010.

COOREVITS P., BELLENGER E., Alternative mesh optimality criteria for h-adaptive finite element method, Finite Elem. Anal. Des. 40 1195–1215, 2004.

CORDAZZO, J. Simulação de Reservatórios de Petróleo Utilizando o Método EbFVM e Multigrid Algébrico, Tese de Doutorado – Engenharia Mecânica –UFSC, 2006.

CORMEM, T. H., LEISERSON C. E., R. L. RIVEST, STEIN C., Introduction to Algorithms. Dubuque, IA, 2001. McGraw-Hill Book Company.

FANCHI, J. R. Principles of Applied Reservoir Simulation, Houston, TX, Elsevier, 2001.

FERREIRA N. D. Método dos Volumes Finitos em Malhas Não-Estruturadas Engenharia Matemática, FCUP, 2007.

FREITAS M. O., CAVALCANTE J. B., VIDALA C. A., MARTHAB L. F., INGRAFFEAC A., WAWRZYNEKC P., A Parallel Technique For Two-dimensional MeshGeneration For Arbitrary Regions With Cracks, CILAMCE, 2011.

FREITAS M. O., SIQUEIRA, D. M.B., CAVALCANTE J. B., VIDAL C. A., SILVA R. J., An Adaptive Parametric Surface Mesh Generation Method Guided by Curvatures, International Meshing Roundtable, 2013.

FREY P. J., MARECHAL L., Fast Adaptive Quadtree Mesh Generation, 1998.

GEUZAINE C., REMACLE J. F., Gmsh a three dimensional finite element mesh generator with built-in pre and post processing facilities. Disponível em: geuz.org/gmsh/. Acessado em: 5 de fevereiro de 2014.

HIRSCH, C. Numerical Computation of Internal and External Flows, Britain, Elsevier, 2007.

HURTADO F. S. Uma formulação de volumes finitos baseada em elementos para a simulação do deslocamento bifásico imiscível em meios porosos - Dissertação de Mestrado-Engenharia Mecânica – UFSC, 2005.

LOHNER R., PARIKH P., Generation of three-dimensional unstructured grids by the advancing front method, International Journal For Numerical Methods In Fluids, 1988.

LYRA, P. R. M, CARVALHO, D. K. E., Utilização de Estratégias de Adaptação de Malhas na Simulação Computacional de Escoamentos Bifásicos em Meios Porosos, Congresso ibero-americano de engenharia mecânica, 2007;

LYRA, P. R. M. Unstructured Grid Algorithms for Fluid Dynamics and Heat Conduction,Tese de Doutorado, University of Wales – Swansea, 1994;

MARCHI C. H., CARVALHO A. F., Condições Suficientes Para Estimar Com Acurácia e Confiabilidade Erros de Discretização em CFD, CILAMCE, 1999.

MARQUES R., Estimador de Erro a Posteriori Baseado em Recuperação do Gradiente para o Método dos Elementos Finitos Generalizados, Tese de Mestrado – EESC/USP, 2011.

MIRANDA A.C.O., CAVALCANTE J.B., e MARTHA L.F. An algorithm for two-dimensional mesh generation for arbitrary regions with cracks. In SIBGRAPI ’99: Proceedings of the II Brazilian Symposium on Computer Graphics and Image Processing, IEEE Computer Society, 1999.

NIKITIN, K., VASSILEVSKI Y. A Monotone Non-linear Finite Volume Method for Advection-diffusion Equations and Multiphase Flows , ECMOR, 2012.

NIKITIN, K. Nonlinear finite volume method for two phase flow in porous media.

Math. Model., 20:11:131–147, 2010.

PEREIRA M. T. Análise de esquemas numéricos de cinco e nove pontos para malhas tipo cornerpoint, Dissertação de Mestrado, Engenharia Mecânica, UFSC, 2009.

POPIOLEK T. L., SANTOS M. A., Um novo critério de adaptação de malhas. Revista Brasileira de Recursos Hídricos, Abr/Jun 2001

SAVITCH W. J. C++ Absoluto, São Paulo, Pearson, 2004.

SILVA, O. E., SILVA, R. S., LUNA, G. B. Adaptação tipo H Aplicada à Simulação de Reservatórios de Petróleo Usando Programação Orientada a Objetos em C++ CILAMCE 2008 - Iberian Latin American Congress on Computational Methods in Engineering, 2008 – Recife.

SIQUEIRA D. M., et al An Adaptive Parametric Surface Mesh Generation Method Guided by Curvatures, International Meshing Roundtable, 2013.

THE FMDB TEAM, The FMDB User´s Guide, Scientific Computation Research Center, 2013 Disponível em: http://www.scorec.rpi.edu/FMDB/. Acessado em: 17 de março de 2014.

THOMPSON J. F., B. K. SONI, WEATHERILL N. P., Handbook of Grid Generation, CRC Press, 1998.

ZIENKIEWICZ O. C., TAYLOR R. L. The Finite Element Method Vol 1, BH, 2000.

ZIENKIEWICZ O. C., RANK E. A Simple Error Estimator In The Finite Element Method, 1981.

ZHENG X., ANDERSON A., CRISTINI V., Adaptive unstructured volume remeshing – II: Application to two- and three-dimensional level-set simulations of multiphase flow, Journal of Computational Physics 208, 2005.

APÊNDICE

A.1 Aspectos relevantes do C++

Quando a linguagem C++ foi padronizada por um comitê ISO/ANSI em 1997, uma biblioteca padrão foi incluída ao seu desenvolvimento: A STL (Standard Template Library). Trata-se de uma biblioteca genérica que oferece soluções de gerenciamento de coleções de dados junto com o uso de algoritmos modernos e eficientes (SOUZA, 2008).

O STL dispensa o programador de desenvolver *arrays* dinâmicos, listas encadeadas, árvores binarias ou algoritmos de busca. Basta que use o tipo apropriado de container (Em programação orientada a objetos container é um objeto que contém outros objetos que podem ser removidos ou incluídos dinamicamente) e as funções membros e algoritmos para processar os dados.

Visando o funcionamento ótimo foram utilizados estrategicamente alguns tipos de containeres da biblioteca STL. Principalmente no módulo adaptador, foram usados o tipo list, o container tipo set e o tipo map por suas características intrínsecas.

O container tipo list é um tipo de lista ligada e um exemplo de estrutura de dados dinâmica. Uma estrutura de dados dinâmica é aquela que possui a capacidade de moldar-se em tempo de execução. O conceito de alocação dinâmica está fortemente ligado ao de estrutura dinâmica, pois estruturas dinâmicas geralmente requerem alocação dinâmica para comportar suas modificações internas. No caso da list, isso acontece pois para cada novo elemento inserido, um novo "slot" na lista tem que ser criado, onde esse slot deve ter o objeto inserido e dois ponteiros, um para o próximo elemento e outro pro anterior de tal forma que sempre é possível acrescentar elementos à lista (SAVITCH W., 2004).

Diferentemente de um *array* estático (a grande diferença entre arrays e lista é que a lista possui complexidade constante na inserção e remoção enquanto o *array* possui complexidade constante na busca) que precisa na sua criação de declaração de seu tamanho para fins de alocação de memória. A figura A.1 esquematiza a estrutura do list e do *array*.

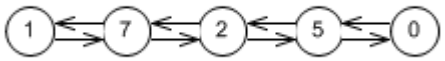
Fig. A.1 A) Representação de lista ligada B) *Array* previamente alocado e não utilizado completamente.

Fonte: O autor.

Essa característica pode parecer uma vantagem, mas para fins de desempenho, a operação de alocação e desalocação dinâmica de memória sempre que é inserido ou removido algum elemento implica em um custo computacional, ao contrário de vetores estáticos que realizam a alocação apenas na sua criação. Dessa forma os containeres dinâmicos devem ser usados apenas quando não se souber realmente com qual quantidade de elementos será operada a rotina.

Ainda o container list é do tipo duplamente ligado pois os elementos possuem ponteiros tanto para o elemento seguinte quanto para o anterior de forma que pode-se iterar na lista avançando ou recuando.

Fig. A.2 Container list duplamente ligado.



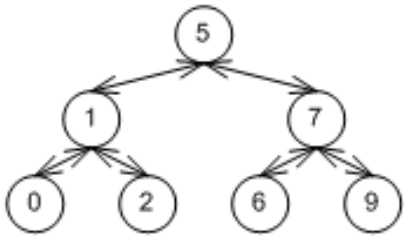
Fonte: O autor.

Contudo esse tipo de armazenamento simples não disponibiliza ferramentas otimizadas de busca, sendo que na procura de algum elemento específico seria necessário percorrer o container desde o início até encontrar o elemento desejado. Dessa forma ele é usado quando se requeira um armazenador simples com alocação dinâmica sem necessidade de se operar buscas por elementos específicos e sem necessidade de ordenação dos elementos. Ainda o list permite a inclusão de elementos repetidos o que em alguns algoritmos pode ser um problema, especialmente em algoritmos de busca como o BFS usado neste trabalho, esta característica do list não é vantajosa.

Neste trabalho afim de facilitação do entendimento das rotinas, nos referiremos ao list como lista deixando dessa forma implícita as funcionalidades deste tipo sem contudo entrar em detalhes na rotina das implicações da escolha deste tipo de armazenador.

O container tipo *set* também é um tipo de alocação dinâmica do tipo árvore. A diferença de sua estrutura de dados para o tipo lista é que cada elemento na árvore do set pode apontar para dois elementos. Isso o classifica como árvore binária e requer a organização dos elementos segundo uma ordem específica que agiliza o processo de busca (por padrão através do operador "<"). A figura A.3 mostra a organização de uma árvore segundo o critério de valor do elemento e ordem de inserção.

Fig. A.3 Organização tipo árvore binária do container set.



Fonte: O autor.

Set´s são containers que armazenam elementos únicos na sequência de uma ordem específica usando uma estrutura em árvore. São mais lentos do que os listas para inserir elementos devido ao processo de comparação com elementos previamente inseridos, mas bem mais eficazes quando se trata de proceder uma busca em seus elementos. São especialmente recomendados quando há o interesse de evitar a repetição de inclusão de elementos no container. Para fins de clareza das explicações em capítulos seguintes, os containeres do tipo set serão referenciados simplesmente por conjuntos.

Os containeres do tipo map armazenam os elementos formados por uma combinação de um valor chave e um valor mapeado, seguindo uma ordem específica. Em um map, os valores chave são geralmente utilizados para classificar e identificar exclusivamente os elementos, enquanto que os valores mapeados armazenar o conteúdo associado a esta chave. Os tipos de valor de chave e mapeados podem ser diferentes. Internamente, os elementos em um mapa são sempre ordenados por sua chave segundo uma estrutura de árvore binária. Assim é possível classificar um objeto armazenado nesse container por sua chave relacionada e fazer buscas no map a partir do valor da chave (SAVITCH W.,2004).

A.2 O FMDB

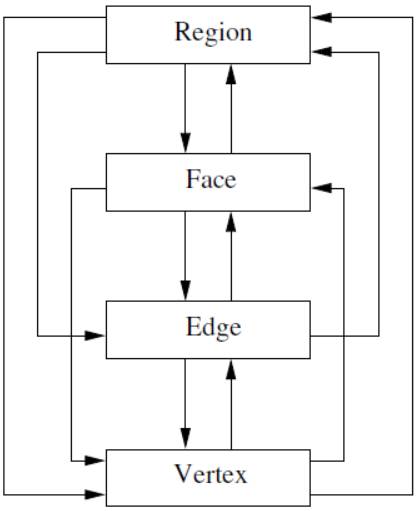
O ponto inicial para construir qualquer simulador que lida com malhas não estruturadas é definir que tipo de estrutura de dados serão usadas para acessar os elementos (nos, arestas, faces e volumes) da malha ao longo da simulação.

O FMDB é um pacote com uma série de rotinas escritas em C++ que se apresenta como uma boa opção no gerenciamento de malhas seriais ou distribuídas em computadores paralelos, fazendo uso constante da biblioteca padrão do C++ (STL), o FMDB também utiliza o ParMetis e o Autopack na distribuição da malha entre os processadores, tendo o MPI como base para todas as rotinas de envio e recebimento de dados remotos.

Adaptação em malhas estruturadas ou não-estruturadas é uma tarefa complexa para domínios em duas dimensões, 2D, e muito mais para três dimensões, o que torna imperativo o uso de um gerenciador de malhas robusto e flexível. O FMDB (Flexible Distributed Mesh Database) é uma alternativa que tem se mostrado bastante atraente em muitos aspectos. Por ser desenvolvido inteiramente em C++ e fazer uso constante da biblioteca padrão STL (Standard Template Library), este pacote foi empregado no programa de adaptação. As rotinas de refinamento e desrefinamento necessárias durante a adaptação são todas baseadas em subrotinas de gerenciamento de malha do FMDB que dispõe de funções de criação de elementos, de remoção, de acesso às conectividades e elementos vizinhos de um elemento, leitura de malhas, criação e reorganização da estrutura de dados, etc.

A flexibilidade que o próprio nome do FMDB passa vem do fato de ser possível criar estruturas de dados de acordo com a necessidade do usuário, ou seja, pode-se criar uma estrutura com entidades que geralmente não são fornecidas pelo gerador de malhas como as arestas. Isso pode ser visualizado pela estrutura de adjacências criada pelo FMDB na figura A.4.

Fig. A.4 Adjacências possíveis no gerenciamento com FMDB.



Fonte: Scientific Computation Research Center, 2013.

Essa estrutura "flexível" permite acionar ou não essas conectividades entre os elementos para fins de aumentar o desempenho computacional, bem como a possibilidade de utilizar e reorganizar a estrutura de dados permite a integração com qualquer software que manipule malhas como exemplo geradores de malha.

A construção da estrutura de dados do FMDB começa a definição de um objeto tipo mMesh. A classe mMesh representa uma interface para manipular todos os elementos da malha. Por meio desta classe são criadas as estruturas de nós e elementos baseadas nas conectividades dos mesmos oriundas do gerador de malhas. A tarefa de leitura da malha é totalmente automatizada através da função m_load que se encarrega automaticamente da criação dos elementos a partir do arquivo .msh padrão do *Gmsh*.