

ELEMENTOS FINITOS ADAPTATIVOS PARA ESCOAMENTOS BI-FÁSICOS
IMISCÍVEIS LIGEIRAMENTE COMPRESSÍVEIS

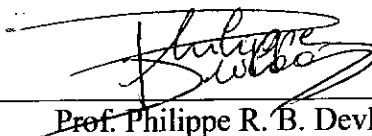
Arlenés Silvino da Silva

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

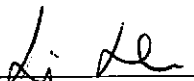
Aprovada por:



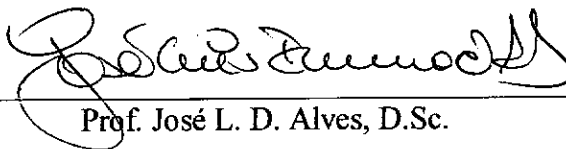
Prof. Alvaro L. G. A. Coutinho, D.Sc.



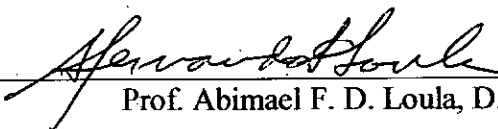
Prof. Philippe R. B. Devloo, PhD.



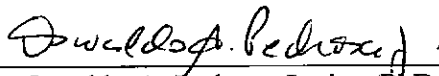
Prof. Luiz Landau, D.Sc.



Prof. José L. D. Alves, D.Sc.



Prof. Abimael F. D. Loula, D.Sc.



Dr. Osvaldo A. Pedrosa Junior, PhD.

RIO DE JANEIRO, RJ – BRASIL

MAIO DE 2000

SILVA, ARLENES SILVINO DA

Elementos Finitos Adaptativos para
Escoamentos Bi-fásicos Imiscíveis Ligeira-
mente Compressíveis [Rio de Janeiro] 2000

VIII, 97 p., 29.7 cm, (COPPE/UFRJ, D.Sc.,
ENGENHARIA CIVIL 2000)

Tese - Universidade Federal do Rio de
Janeiro, COPPE

1. Métodos de Elementos Finitos
2. Métodos Adaptativos
3. Escoamentos Imiscíveis em Meios Porosos
I. COPPE/UFRJ II. Título (série)

À minha esposa Terezinha e às minhas filhas Monika e Flavia

Em memória de meus Pais

AGRADECIMENTOS

Desejo expressar meus sinceros agradecimentos aos meus orientadores e amigos, Prof. Alvaro L. G. A. Coutinho e Prof. Philippe R. B. Devloo pela orientação, incentivo e compreensão.

Ao Prof. Fernando L. B. Ribeiro por disponibilizar o visualizador científico VIEW3D utilizado neste trabalho.

Ao Prof. Dennis Parsons pela valiosa discussão sobre o método multigríd.

Aos colegas da Engenharia Civil da COPPE, pelo convívio diário sempre agradável. Em especial agradeço a Lucia Catabriga, Marcos A. D. Martins e Cláudia Mazza Dias.

Ao programa CAPES/PICD/UFMT, cujo apoio financeiro possibilitou a execução deste trabalho.

Ao LAMCE e ao NACAD da COPPE/UFRJ pelo apoio computacional.

Aos colegas do Departamento de Ciências da Computação da Universidade Federal de Mato Grosso, pelo apoio e compreensão.

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

ELEMENTOS FINITOS ADAPTATIVOS PARA ESCOAMENTOS BI-FÁSICOS
IMISCÍVEIS LIGEIRAMENTE COMPRESSÍVEIS

Arlenés Silvino da Silva

Maio/2000

Orientadores: Alvaro L. G. A. Coutinho

Philippe R. B. Devloo

Programa: Engenharia Civil

A simulação numérica de escoamentos multi-fásicos em meios porosos rígidos é um problema que apresenta várias dificuldades do ponto de vista numérico em função das não-linearidades do acoplamento das equações e da presença de termos convectivos. Muitos trabalhos nesta área adotam modelos simplificados para evitar estas dificuldades, portanto descartando os efeitos da força da gravidade, pressão capilar entre outros. É também comum considerar que o tensor de permeabilidade absoluta é isotrópico.

Neste trabalho apresenta-se uma aproximação por elementos finitos do conjunto de equações que modela o deslocamento de fluidos bi-fásicos imiscíveis ligeiramente compressíveis, considerando-se os efeitos do campo gravitacional, pressão capilar e o tensor de permeabilidade anisotrópico.

Para modelar termos convectivos na equação da saturação, adotou-se uma formulação estabilizada do tipo Streamline Upwind/Petrov-Galerkin (SUPG) acrescida de um operador de captura de descontinuidade do tipo CAU. A equação da pressão é aproximada por uma formulação de Galerkin. Para melhorar a qualidade da velocidade computada uma técnica de pós-processamento é utilizada. Desta forma pode-se empregar aproximações padrão de elementos finitos lineares para todas as variáveis. A formulação semi-discreta resultante é aproximada no tempo por um algoritmo bloco-iterativo preditor multi-corretor. Para modelar problemas transientes os quais incluem fontes pontuais e sumidouros, um algoritmo dinâmico para refinar/desrefinar localmente a malha foi utilizado. Tal modelo adaptativo permite obter uma solução com maior precisão reduzindo o custo computacional em termos de tempo de CPU e memória.

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

ADAPTIVE FINITE ELEMENT PROCEDURES FOR SLIGHTLY COMPRESIBLE
IMMISCIBLE TWO-PHASE FLOW.

Arlenes Silvino da Silva

May/2000

Advisors: Alvaro L. G. A. Coutinho

Philippe R. B. Devloo

Department: Civil Engineering

A numerical method is presented for the simulation of immiscible two-phase flow through a rigid porous medium. Such simulations present several difficulties from the numerical point of view due to the non-linearity of the coupled system of equations and the dominant convective effects. It is usual in this area to adopt simplified models to avoid the above mentioned difficulties, thus discarding the effects of gravity, capilar pressure, among others. It is also common to consider the permeability tensor isotropic.

In this work, a finite element approximation is presented for the system of equations which models slightly compressible immiscible two-phase flow, considering the effects of gravity, capilar pressure and anisotropic permeability tensor.

In order to handle the convective terms present in the saturation equation, we employ the Stream-line Upwind Petrov Galerkin (SUPG) method supplemented with a CAU type shock capturing term. Pressure equation is approximated by Galerkin method. A post processing algorithm was used in order to improve the approximation of the computed velocity term. The resulting method employs equal order standard interpolations for all variables.

The resulting system of equations is integrated in time using a block iterative predictor-multicorrector scheme. In order to model transient problems which include point sources and sinks, a dynamic local refinement/coarsening algorithm was used. Such adaptive models allow to obtain a higher precision while reducing CPU time and memory.

Índice

1	Introdução	1
2	Formulação Matemática	6
2.1	Equação da pressão	9
2.2	Equação da saturação	11
2.3	Condições de contorno e condições iniciais	13
3	Formulação Aproximada de Elementos Finitos	17
3.1	Discretização espacial	17
3.2	Discretização temporal	20
3.3	Pós-processamento do campo de velocidades	23
3.4	Controle do Passo de Tempo	25
3.5	Implementação Elemento por Elemento	27
3.5.1	Pré-condicionamento	29
3.5.2	Método Multimalha	30
3.5.3	Imposição das Condições de Contorno no Pós-processamento das Velocidades	33
3.6	Validação da Solução da Pressão	34
3.7	Validação do Esquema de Estabilização	38
3.7.1	Caso Unidimensional	38
3.7.2	Problema de Cinco Poços	40
4	Implementação do Refinamento Adaptativo de Malha	46
4.1	Estimador de erro	47

4.2	Estratégia adaptativa	49
4.2.1	Algoritmo adaptativo com imposição de nível máximo	50
4.3	Estratégia para refinar e/ou desrefinar a malha	51
4.3.1	Refinamento	52
4.3.2	Desrefinamento.....	54
4.3.3	Estruturas de dados	55
4.3.4	Interpolação e restrição da solução	57
4.4	Implementação da Estratégia Adaptativa em um código existente	58
5	Exemplos Numéricos	61
5.1	Arranjo de cinco poços com razão de viscosidade unitária	61
5.1.1	Avaliação de desempenho	62
5.2	Arranjo de cinco poços com razão de viscosidade desfavorável	67
5.3	Injeção de água pela lateral do domínio	71
5.3.1	Injeção de água pela lateral do domínio sem gravidade	71
5.3.2	Injeção de água pela lateral do domínio com gravidade	73
5.4	Tensor de Permeabilidade Completo.....	76
5.5	Solução do Arranjo de Cinco Poços em Meio Heterogêneo	78
6	Conclusões e Trabalhos Futuros	90
	Bibliografia	92

Capítulo 1

Introdução

Nas últimas décadas muito esforço tem sido empregado em pesquisas que visam descrever com precisão o processo de recuperação de óleo em reservatórios de petróleo. O que se busca é maximizar o processo de recuperação de óleo, minimizando os custos de produção.

O aumento da capacidade de processamento dos computadores permitiu o desenvolvimento de simuladores mais completos, onde uma maior quantidade de fenômenos físicos ocorrendo no interior do reservatório podem ser considerados, produzindo assim resultados mais realísticos. O processo de simulação depende de um modelo matemático que represente o fluxo do fluido dentro do reservatório. Este modelo matemático do sistema físico é representado por um conjunto de equações diferenciais parciais, obtidas a partir das leis de Darcy e da equação de conservação de massa para cada fase do fluido. Para fluidos imiscíveis ligeiramente compressíveis, que será objeto de estudo neste trabalho, o modelo matemático é formado por duas equações acopladas, uma parabólica que descreve a variação da pressão com o tempo e posição; e uma equação hiperbólica que descreve a variação do fluido injetado (saturação da água) com o tempo e posição. Para completar o modelo acrescenta-se um conjunto apropriado de condições de contorno [20][4].

A utilização de métodos numéricos na simulação de reservatório de petróleo é uma necessidade, uma vez que as equações envolvidas são altamente complexas, dificultando a utilização de métodos analíticos. Para se obter uma solução que represente os fenômenos ocorrendo dentro do reservatório, métodos aproximados devem ser desenvolvidos.

Assim, a utilização de computadores é a única forma de se obter uma solução realística para o conjunto de equações associado à simulação de reservatórios. Mesmo com a utilização de computador, para que se obtenha uma solução de boa qualidade a um custo computacional aceitável, a eficiência do conjunto de algoritmos matemáticos envolvidos é um aspecto fundamental.

A simulação numérica das equações que governam o deslocamento imiscível de dois fluidos em um meio poroso rígido apresenta várias dificuldades, entre elas podemos citar: (a) o campo de velocidades é determinado pela Lei de Darcy, que depende do gradiente de pressão, (b) a presença de elevados gradientes de saturação da água, (c) a presença de fortes não-linearidades, provenientes de relações constitutivas não lineares, tais como, permeabilidades relativas, pressão capilar, etc.[20]. Estas dificuldades foram discutidas por vários autores.

Tradicionalmente os simuladores utilizam o método das diferenças finitas para discretizar o conjunto de equações diferenciais envolvido, no entanto esta técnica apresenta limitações, como por exemplo, na representação de geometrias complexas. Nos últimos anos outros métodos de discretização, tais como elementos finitos e volumes finitos tem sido utilizados. O método dos elementos finitos apresenta flexibilidade para representar geometrias complexas e utiliza malhas não estruturadas, favorecendo a utilização de técnicas adaptativas, conduzindo a um aumento da exatidão e eficiência da solução. Os primeiros trabalhos para deslocamentos imiscíveis na área de elementos finitos foram apresentados por SPIVAK, PRICE e SETTARI [52], que utilizou basicamente o método de Galerkin, com elementos finitos cúbicos de Hermite. Já neste trabalho SPIVAK, PRICE e SETTARIS [52] mostrou que os efeitos de orientação de malha são reduzidos. LEWIS, MORGAN e JOHNSON [30] utilizam elementos isoparamétricos padrão, enquanto que

HUYAKORN e PINDER [27] apresentam o primeiro método Petrov-Galerkin. PEDROSA Jr. e AZIZ [28], com volumes finitos mostram as vantagens do refinamento local de malha. Além disso, existem métodos híbridos, como FUNG, HIEBERT e NGHIEM [21] que utilizaram elementos finitos triangulares para a pressão e volumes finitos com upwind de primeira ordem para os termos convectivos, ou ainda, ESPEDAL e EWING [19] que combinam o método das características com métodos Petrov-Galerkin.

LANGTANGEN [29] mostrou que as dificuldades provenientes da existências de elevados gradientes de saturação poderiam ser superadas com a utilização de formulação tipo Petrov-Galerkin com operador de captura de descontinuidade. No mesmo trabalho o autor discute esquemas de discretização temporal totalmente implícitos e implícitos sequenciais. LOULA, GARCIA, MURAD et al. [33] apresentam uma técnica de pós-processamento para aumentar a qualidade da aproximação do campo de velocidades, para fluidos imiscíveis incompressíveis, onde pressão, velocidade e saturação são calculadas com elementos finitos Lagrangianos de mesma ordem. COUTINHO, ALVES, GARCIA et al.[8] discutiram a utilização de métodos adaptativos implícito/explicito (AIE) para a equação da saturação. COUTINHO, SILVA e DEVLOO [10] mostraram a importância do operador de captura de descontinuidade para evitar o surgimento de oscilações espúrias nas frentes de saturação. Nos trabalhos de LOULA, GARCIA, MURAD et al. [33], COUTINHO, ALVES, GARCIA et al. [8], COUTINHO, SILVA e DEVLOO [10], SILVA, COUTINHO e PARSONS [47], PARSONS e COUTINHO [38] a equação da pressão foi aproximada no espaço pelo métodos de Galerkin, as velocidades avaliadas por técnicas de pós-processamento e a equação da saturação aproximada no espaço através de formulações estabilizadas do tipo Petrov-Galerkin. Em PARSONS e COUTINHO [38] o

sistema de equações lineares da equação da pressão é solucionado pelo método multigrid.

Em função das dificuldades apresentadas, a simulação numérica do processo de recuperação de óleo é computacionalmente cara. Métodos convencionais têm dificuldades de representar fenômenos ocorrendo dentro do reservatório, tais como fontes pontuais e regiões de altos gradientes que se movem com o tempo. Para que se consiga captar informações sobre o comportamento do fluxo nestas regiões, ou utiliza-se uma malha muito fina em todo o domínio, o que resulta em um custo computacional excessivamente alto, ou utiliza-se métodos adaptativos. Com malhas adaptativas é possível obter uma solução de boa qualidade sem um custo computacional elevado em termos de memória e tempo de CPU. As técnicas adaptativas são divididas em três categorias:

- a) Refinamento h , onde o erro da aproximação de elementos finitos é reduzido pela divisão dos elementos cujo erro é maior que um dado valor;
- b) Refinamento p , onde o erro é reduzido aumentando-se a ordem do polinômio do elemento localmente;
- c) Refinamento h - p , onde as duas técnicas anteriores são usadas conjuntamente para produzir a solução desejada.

O refinamento h pode ser aplicado a qualquer problema que utilize aproximação de elementos finitos [12], sendo também o mais utilizado. Neste trabalho trataremos apenas este tipo de refinamento. Uma discussão geral sobre estas três técnicas de adaptatividade pode ser encontrada em [12]. Métodos adaptativos para problemas de escoamentos de fluidos incompressíveis em meios porosos foram utilizados por HORNUNG e TRANGESTEN [24], HANSEN e SPEDAL [23]. Neste último o refinamento utilizado é fixo, isto é, cada elemento selecionado para ser refinado é dividido um número fixo de ve-

zes. SILVA, DEVLOO e COUTINHO [48] apresentaram métodos adaptativos aplicados à simulação de fluidos ligeiramente compressíveis. No campo da Dinâmica dos Fluidos Computacional, métodos adaptativos têm sido largamente utilizados nos últimos anos e a bibliografia é muito ampla. Veja por exemplo [32][37][49], para uma discussão sobre métodos adaptativos.

Para superar as dificuldades apresentadas pelo sistema acoplado de equações diferenciais que descreve o escoamento de fluidos imiscíveis bi-fásicos compressíveis em um meio poroso rígido, utilizou-se uma formulação estabilizada tipo Petrov-Galerkin (SUPG) acrescida do operador de captura de descontinuidade tipo CAU para a equação da saturação. A discretização por elementos finitos deste conjunto de equações conduz a três sistemas de equações lineares que devem ser resolvidos a cada passo de tempo. Para se alcançar a eficiência computacional desejada a escolha dos métodos de solução destes sistemas é uma etapa importante. Assim optou-se por métodos iterativos associados à técnica de armazenamento elemento por elemento que, além de propiciar uma economia em termos de memória, é adequada para utilização com métodos adaptativos e implementação em paralelo [7]. A integração no tempo é feita através de um algoritmo bloco-iterativo preditor-multicorretor.

O segundo capítulo apresenta o modelo matemático utilizado conforme descrito em PEACEMAN [39]. A formulação de elementos finitos para o conjunto de equações é apresentada no capítulo 3. O método adaptativo utilizado, incluindo o método de refinamento/desrefinamento, estrutura de dados e alguns detalhes de implementação são discutidos no capítulo 4. O capítulo 5 é dedicado à apresentação dos exemplos numéricos para várias situações e modelos de reservatórios. O último capítulo resume as principais conclusões e propostas para trabalhos futuros.

Capítulo 2

Formulação Matemática

Na construção de um modelo para o escoamento de fluidos em um meio poroso devem ser considerados os mecanismos físicos e químicos ocorrendo dentro do reservatório, a formulação matemática das leis de conservação e um modelo numérico eficiente para resolver estas equações.

A formulação matemática para o escoamento de fluidos em um meio poroso rígido, conforme PEACEMAN [39], é obtida combinando-se a lei de Darcy com a equação de balanço para cada fase, resultando em um sistema acoplado de equações diferenciais parciais altamente não linear, que será descrito a seguir. Considera-se o deslocamento simultâneo de dois fluidos imiscíveis, isto é, não existe transferência de massa entre os dois fluidos, em um meio poroso rígido $\Omega \subset R^2$, de contorno Γ , equipado com um vetor normal unitário externo $\mathbf{n} = (n_x, n_y)$. O vetor posição é representado por $\mathbf{x} = (x, y)$ e o tempo por $t \in [0, T]$. Sendo a saturação de uma fase definida como a fração do meio poroso ocupado por esta fase, temos que para o caso do escoamento bifásico,

$$s_o + s_w = 1 \quad (2.1)$$

onde os subscritos o e w referem-se respectivamente às fases molhantes e não molhantes, por exemplo, óleo e água. Assim, s_o e s_w representam a saturação das fases.

A velocidade superficial do fluido é dada pela lei de Darcy para cada fase, que pode ser escrita como:

$$v_i = -\lambda_i \mathbf{K} (\nabla p_i - \rho_i g \nabla z), \quad i = o, w \quad (2.2)$$

onde:

$$\lambda_i = \frac{k_{ri}}{\mu_i}; \quad i = o, w \quad (2.3)$$

Nestas equações, λ_i são as mobilidades das fases (água e óleo), \mathbf{K} é o tensor de permeabilidade definido por ora como $\mathbf{K} = k\mathbf{I}$, $\kappa = k(\mathbf{x})$ é a permeabilidade absoluta da rocha, k_{ri} são as permeabilidades relativas das fases, μ_i as viscosidades, p_i as pressões, g é a força da gravidade e ∇z é uma função vetorial que aponta na direção em que a força da gravidade atua. É usual no caso de meio isotrópico se reescrever as mobilidades das fases como,

$$\lambda_i = \mathbf{K} \frac{k_{ri}}{\mu_i}; \quad i = o, w \quad (2.4)$$

As funções k_{ri} dependem somente da saturação da água e introduzem não linearidade no fluxo dos fluidos, isto é,

$$k_{ri} = k_{ri}(s_w) \quad (2.5)$$

As equações de conservação de massa para cada fase são:

$$-\nabla \cdot (\rho_o v_o) + q_o = \frac{\partial}{\partial t} (\phi \rho_o s_o) \quad (2.6)$$

$$-\nabla \cdot (\rho_w v_w) + q_w = \frac{\partial}{\partial t} (\phi \rho_w s_w) \quad (2.7)$$

onde ρ_o e ρ_w , representam respectivamente as massas específicas do óleo e da água, dependentes da pressão das fases. As vazões mássicas do óleo e água são representadas, respectivamente, por q_o e q_w . A porosidade efetiva do meio é representada por $\phi = \phi(\mathbf{x})$.

Introduzem-se as seguintes definições úteis. A pressão média é definida em função da pressão em cada fase como

$$p_m = \frac{1}{2} (p_o + p_w) \quad (2.8)$$

A diferença entre as pressões dos dois fluidos é a pressão capilar, p_c , que depende apenas da saturação da água, e é dada por,

$$p_c = p_o - p_w = p_c(s_w) \quad (2.9)$$

Uma curva típica da pressão capilar em função da saturação da água é apresentada na Figura 1.

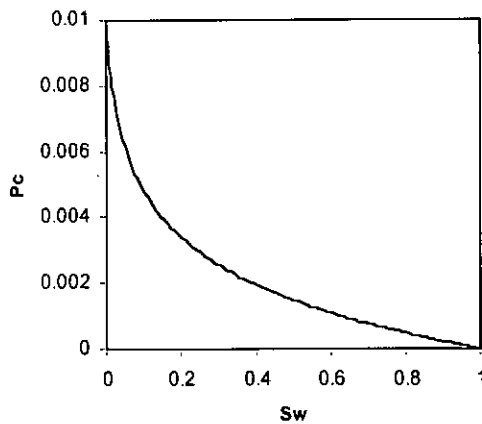


Figura 1: Curva típica da pressão capilar (PEACEMAN[39]).

A pressão de cada fase pode ser escrita em função da pressão média e pressão capilar como,

$$p_o = p_m + \frac{1}{2}p_c \quad (2.10)$$

$$p_w = p_m - \frac{1}{2}p_c \quad (2.11)$$

De acordo com PEACEMAN [39] as equações (2.6) e (2.7) podem ser reduzidas a duas equações, uma parabólica, conhecida como equação da pressão e outra essencialmente hiperbólica, conhecida como equação da saturação. A dedução destas equações será vista a seguir.

2.1 Equação da pressão

A primeira equação do sistema é conhecida como equação da pressão e descreve sua variação com o tempo e posição. Expandindo a derivada no tempo das equações (2.6) e (2.7) obtemos:

$$-\nabla \cdot (\rho_o v_o) + q_o = \rho_o s_o \frac{\partial \phi}{\partial t} + \phi s_o \frac{d\rho_o}{dp_o} \frac{\partial p_o}{\partial t} + \phi \rho_o \frac{\partial s_o}{\partial t} \quad (2.12)$$

$$-\nabla \cdot (\rho_w v_w) + q_w = \rho_w s_w \frac{\partial \phi}{\partial t} + \phi s_w \frac{d\rho_w}{dp_w} \frac{\partial p_w}{\partial t} + \phi \rho_w \frac{\partial s_w}{\partial t} \quad (2.13)$$

Dividindo (2.12) por ρ_o e (2.13) por ρ_w , obtemos:

$$-\frac{1}{\rho_o} \nabla \cdot (\rho_o v_o) + \frac{q_o}{\rho_o} = s_o \frac{\partial \phi}{\partial t} + \phi \frac{s_o}{\rho_o} \frac{d\rho_o}{dp_o} \frac{\partial p_o}{\partial t} + \phi \frac{\partial s_o}{\partial t} \quad (2.14)$$

$$-\frac{1}{\rho_w} \nabla \cdot (\rho_w v_w) + \frac{q_w}{\rho_w} = s_w \frac{\partial \phi}{\partial t} + \phi \frac{s_w}{\rho_w} \frac{d\rho_w}{dp_w} \frac{\partial p_w}{\partial t} + \phi \frac{\partial s_w}{\partial t} \quad (2.15)$$

Somando estas duas expressões e usando o fato que $s_o + s_w = 1$, temos:

$$\begin{aligned} -\frac{1}{\rho_o} \nabla \cdot (\rho_o v_o) - \frac{1}{\rho_w} \nabla \cdot (\rho_w v_w) + \left(\frac{q_o}{\rho_o} + \frac{q_w}{\rho_w} \right) = \\ \frac{\partial \phi}{\partial t} + \phi s_o \frac{1}{\rho_o} \frac{d\rho_o}{dp_o} \frac{\partial p_o}{\partial t} + \phi s_w \frac{1}{\rho_w} \frac{d\rho_w}{dp_w} \frac{\partial p_w}{\partial t} \end{aligned} \quad (2.16)$$

ou

$$-\frac{1}{\rho_o} \nabla \cdot (\rho_o v_o) - \frac{1}{\rho_w} \nabla \cdot (\rho_w v_w) + Q_T = \frac{\partial \phi}{\partial t} + \phi s_o c_o \frac{\partial p_o}{\partial t} + \phi s_w c_w \frac{\partial p_w}{\partial t} \quad (2.17)$$

onde, Q_T é a vazão volumétrica total, dada por,

$$Q_T = \frac{q_o}{\rho_o} + \frac{q_w}{\rho_w} \quad (2.18)$$

As constantes de compressibilidade das fases, c_o e c_w são definidas pelas relações,

$$c_o = \frac{1}{\rho_o} \frac{d\rho_o}{dp_o} \quad (2.19)$$

$$c_w = \frac{1}{\rho_w} \frac{d\rho_w}{dp_w} \quad (2.20)$$

Substituindo as equações (2.2) em (2.17) e usando o fato de que a pressão de cada fase pode ser escrita em função da pressão média e pressão capilar, obtemos:

$$\begin{aligned} \nabla \cdot \left[\left(-\frac{1}{\rho_o} (\rho_o \lambda_o) - \frac{1}{\rho_w} (\rho_w \lambda_w) \right) \mathbf{K} \nabla p_m \right] - \nabla \cdot \left[\left(\frac{1}{2\rho_o} (\rho_o \lambda_o) - \frac{1}{2\rho_w} (\rho_w \lambda_w) \right) \mathbf{K} \nabla p_c \right] + \\ Q_T = \left[\frac{d\phi}{dp_m} + \phi (s_o c_o + s_w c_w) \right] \frac{\partial p_m}{\partial t} + \left[\frac{1}{2} (s_o c_o - s_w c_w) \right] \frac{\partial p_c}{\partial t} \\ + \nabla \cdot \left[\left(\frac{1}{\rho_o} (\rho_o^2 \lambda_o) + \frac{1}{\rho_w} (\rho_w^2 \lambda_w) \right) \mathbf{K} g \nabla z \right] \end{aligned} \quad (2.21)$$

Uma vez que as compressibilidades são baixas, podemos desprezar o termo envolvendo a derivada temporal de p_c , por ser pequena em relação ao termo envolvendo a derivada temporal da pressão média. Logo rearranjando os termos desta equação, temos:

$$\begin{aligned} \phi c_t \frac{\partial p_m}{\partial t} = -\nabla \cdot [(\lambda_o + \lambda_w) \mathbf{K} \nabla p_m] - \frac{1}{2} \nabla \cdot [(\lambda_o - \lambda_w) \mathbf{K} \nabla p_c] + \\ \nabla \cdot [(\rho_o \lambda_o + \rho_w \lambda_w) \mathbf{K} g \nabla z] + Q_T \end{aligned} \quad (2.22)$$

onde,

$$c_t = \frac{1}{\phi} \frac{d\phi}{dp_m} + (s_o c_o + s_w c_w) \quad (2.23)$$

é a compressibilidade total do sistema. As compressibilidades do óleo e da água, são obtidas em laboratório ou por correlações, e seus valores estão entre 6×10^{-6} e $2 \times 10^{-5} \text{ psi}^{-1}$ para o óleo e de $3 \times 10^{-6} \text{ psi}^{-1}$ para a água. A porosidade é a razão entre o volume de vazios e o volume total e, seu valor, obtido em laboratório, depende da rocha e está entre 15 e 35%. Para fluidos moderadamente compressíveis c_t é da ordem de 10^{-6} psi^{-1} .

A equação (2.22), de natureza parabólica, é a equação da pressão. Deste ponto em diante a pressão média p_m será denotada simplesmente por p .

Definindo-se a velocidade total do fluido como,

$$v_T = v_o + v_w \quad (2.24)$$

a equação (2.22) pode ser escrita de forma mais compacta como,

$$\phi c_i \frac{\partial p}{\partial t} + \nabla \cdot v_T = Q_T \quad (2.25)$$

onde a expressão de v_T é,

$$v_T = -(\lambda_o + \lambda_w) \mathbf{K} \nabla p - \frac{1}{2} (\lambda_o - \lambda_w) \mathbf{K} \nabla p_c + [(\rho_o \lambda_o + \rho_w \lambda_w) \mathbf{K} g \nabla z] \quad (2.26)$$

2.2 Equação da saturação

Para chegar à equação da saturação, inicialmente escrevemos a velocidade da fase s_w em função da velocidade total. Isto é alcançado combinando-se as equações (2.2) e (2.9), o que nos conduz a:

$$v_w = f_w v_T + h_w \mathbf{K} \nabla s_w + \lambda_o f_w (\rho_w - \rho_o) \mathbf{K} g \nabla z \quad (2.27)$$

onde f_w é a função de fluxo fracionário da água definida como,

$$f_w (s_w) = \frac{\lambda_w}{\lambda_o + \lambda_w} \quad (2.28)$$

e a função h_w é dada por,

$$h_w (s_w) = -\frac{\lambda_o \lambda_w}{\lambda_o + \lambda_w} \frac{dp_c}{ds_w} \quad (2.29)$$

Uma curva típica da função de fluxo fracionário da água em função da saturação da água é mostrada na Figura 2.

Substituindo a equação (2.27) na equação (2.7) obtemos a equação da saturação,

$$-\nabla \cdot [\rho_w f_w v_T + \rho_w h_w \mathbf{K} \nabla s_w + \rho_w \lambda_o f_w (\rho_w - \rho_o) \mathbf{K} g \nabla z] - q_w = \frac{\partial}{\partial t} (\phi \rho_w s_w) \quad (2.30)$$

Multiplicando a equação (2.30) por $\frac{1}{\rho_w}$ e assumindo ϕ , ρ_o e ρ_w constantes,

temos:

$$-\nabla \cdot [f_w v_T + h_w \mathbf{K} \nabla s_w + \lambda_o f_w (\rho_w - \rho_o) \mathbf{K} g \nabla z] - Q_w = \phi \frac{\partial s_w}{\partial t} \quad (2.31)$$

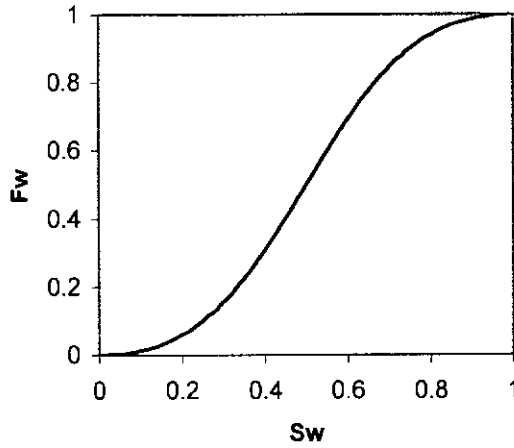


Figura 2: Curva típica da função de fluxo fracionário da água, (PEACEMAN[39]).

onde $Q_w = \frac{q_w}{\rho_w}$.

Definindo o tensor de difusão,

$$\mathbf{D} = \lambda_o f_w \frac{dp_c}{ds_w} \mathbf{K} \quad (2.32)$$

a velocidade aparente de transporte do fluido por:

$$v_a = f_w [v_T + \lambda_o (\rho_w - \rho_o) \mathbf{K} \mathbf{g}] \quad (2.33)$$

o vetor \mathbf{g} , como $\mathbf{g} = g \nabla z$ e lembrando que o tensor de permeabilidade absoluta da rocha no caso geral é,

$$\mathbf{K} = \begin{bmatrix} k_x & k_{xy} \\ k_{yx} & k_y \end{bmatrix} \quad (2.34)$$

onde k_x, k_{xy}, k_{yx}, k_y , são os valores das permeabilidades absolutas em relação aos eixos cartesianos, a Equação (2.31) pode agora ser reescrita como,

$$\phi \frac{\partial s_w}{\partial t} + \nabla \cdot v_a + \nabla \cdot [\mathbf{D} \nabla s_w] + Q_w = 0 \quad (2.35)$$

Esta é a equação da conservação de massa para a fase s_w . Do ponto de vista físico o primeiro termo é o termo de acumulação, o segundo é o termo convectivo e o último é o termo de dispersão-difusão. Note que a pressão capilar tem um papel fundamental

na natureza desta equação, pois dependendo do seu valor a equação da saturação pode ser mais ou menos difusiva. Além disso, a presença da pressão capilar acentua a não linearidade do sistema de equações diferenciais parciais.

Para fluidos moderadamente compressíveis o único termo contribuindo para os efeitos de compressibilidade é a derivada no tempo da equação da pressão (2.22). Como evidenciado em DURLOFSKY [17], este modelo só é válido para $c_t \ll 1$. Para escoamentos de água e óleo em reservatórios de petróleo, $c_t \approx 10^{-6} \text{psi}^{-1}$.

Pode-se agora escrever o sistema completo de equações diferenciais parciais que é:

$$\phi c_t \frac{\partial p}{\partial t} + \nabla \cdot v_T = Q_T \quad (2.36)$$

$$v_T = -\Lambda_p \nabla p - \Lambda_m \frac{dp_c}{ds_w} \nabla s_w + (\rho_o \lambda_o + \rho_w \lambda_w) \mathbf{K} \mathbf{g} \quad (2.37)$$

$$\phi \frac{\partial s_w}{\partial t} + \nabla \cdot v_a + \nabla \cdot [\mathbf{D} \nabla s_w] + Q_w = 0 \quad (2.38)$$

onde

$$\Lambda_p = \mathbf{K} (\lambda_o + \lambda_w) \quad (2.39)$$

$$\Lambda_m = \frac{1}{2} \mathbf{K} (\lambda_o - \lambda_w) \quad (2.40)$$

A velocidade total do fluido, representada pela equação (2.37) é responsável pelo acoplamento entre a equação de conservação de massa do fluido, equação (2.36) e a equação da saturação (2.38).

2.3 Condições de contorno e condições iniciais

Para completar a descrição do problema precisamos especificar um conjunto apropriado de condições de contorno e condições iniciais. A fronteira Γ do domínio Ω pode ser particionada de tal forma que $\Gamma = \Gamma_i \oplus \Gamma_p \oplus \Gamma_d \oplus \Gamma_n$, onde Γ_i e Γ_p denotam, respectivamente a fronteira nos poços injetor e produtor, Γ_d denota a fronteira onde a pressão é prescrita

e Γ_n a parte da fronteira onde a condição de escoamento nulo é especificada, conforme mostra a Figura 3. Portanto as condições de contorno são:

Para a equação da pressão,

$$v_T \cdot \mathbf{n} = Q_i \text{ em } \Gamma_i \times [0, T] \quad (2.41)$$

$$v_T \cdot \mathbf{n} = Q_p \text{ em } \Gamma_p \times [0, T] \quad (2.42)$$

$$v_T \cdot \mathbf{n} = 0 \text{ em } \Gamma_n \times [0, T] \quad (2.43)$$

$$p(\mathbf{x}, t) = \bar{p} \text{ em } \Gamma_d \times [0, T] \quad (2.44)$$

e para a equação da saturação,

$$s_w(\mathbf{x}, t) = \bar{s} \text{ em } \Gamma_i \times [0, T] \quad (2.45)$$

$$\mathbf{D}\nabla s_w \cdot \mathbf{n} = 0 \text{ em } \Gamma_n \times [0, T] \quad (2.46)$$

As condições iniciais para pressão e saturação são respectivamente:

$$p(x, 0) = p^0 \text{ em } \Omega \quad (2.47)$$

$$s(x, 0) = s^0 \text{ em } \Omega \quad (2.48)$$

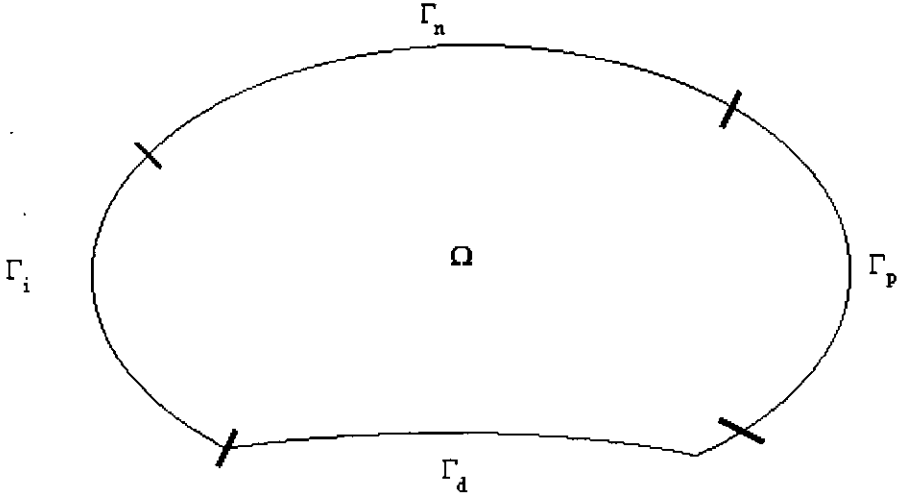


Figura 3: Domínio contendo a partição da fronteira Γ

Isto completa o conjunto de equações diferenciais parciais para o escoamento de fluidos imiscíveis bi-fásicos moderadamente compressíveis, em meios porosos rígidos.

De acordo com DURLOFSKY [17], o sistema de equações (2.36) e (2.38) pode ser adimensionalizado identificando-se valores característicos para cada variável, sendo que as equações adimensionais resultantes dependem de dois números adimensionais. O primeiro quantifica a relação entre os efeitos gravitacionais e convectivos e é dado por,

$$G_d = \frac{K_c \Delta \rho \|\mathbf{g}\|}{\mu_c v_{Tc}} \quad (2.49)$$

onde K_c é a permeabilidade característica do reservatório, avaliada como a média geométrica das permeabilidades do reservatório, $\Delta \rho = \rho_w - \rho_o$ e v_{Tc} é a velocidade total característica. O outro número representa a relação entre os termos convectivos e os termos relacionados à pressão capilar e é designado por,

$$P = \frac{\mu_o v_{Tc} L}{\bar{p}_c K_c} \quad (2.50)$$

onde L é o comprimento característico do reservatório e \bar{p}_c representa a pressão capilar característica, sendo que este número pode ser entendido como um número de Peclet.

Capítulo 3

Formulação Aproximada de Elementos Finitos

3.1 Discretização espacial

Uma forma de calcular a solução aproximada das equações (2.36-2.38) é através do método dos elementos finitos. Isto é alcançado subdividindo-se o domínio Ω em um conjunto de subdomínios Ω^e (elementos finitos) tal que, $\Omega = \bigcup_{e=1}^{Nel} \Omega^e$ e $\bigcap_{e=1}^{Nel} \Omega^e = \emptyset$, onde Nel é o número total de elementos na malha. Em seguida define-se os conjuntos de funções admissíveis para a pressão \mathbb{P}^h , para a saturação \mathbb{S}^h e o conjunto das funções peso \mathbb{W}^h ,

$$\mathbb{P}^h = \{p^h/p^h \in H^h(\Omega), p^h(t) = \bar{p} \text{ em } \Gamma_d\} \quad (3.51)$$

$$\mathbb{S}^h = \{s^h/s^h \in H^h(\Omega), s^h(t) = \bar{s} \text{ em } \Gamma_i\} \quad (3.52)$$

$$\mathbb{W}^h = \{w^h/w^h \in H^h(\Omega), w^h = 0 \text{ em } \Gamma\} \quad (3.53)$$

sendo $H^h(\Omega) \subset H^1(\Omega)$ um espaço de dimensão finita sobre o domínio Ω e $H^1(\Omega)$ é o espaço das funções quadrado integráveis com a primeira derivada também quadrado integrável, isto é,

$$f \in H^1(\Omega) \Leftrightarrow \int \left(\frac{df}{dx} \right)^2 d\Omega < \infty \quad (3.54)$$

Para problemas transientes a formulação semi-discreta de elementos finitos envolve duas etapas: aproximação espacial, cujo resultado é um conjunto de equações diferenciais ordinárias, e a aproximação temporal onde o conjunto de equações diferenciais ordinárias são aproximadas no tempo.

Como a equação da pressão é uma equação diferencial parcial do tipo parabólica, porém quase elíptica, pois o modelo é moderadamente compressível, o método de Galerkin é eficaz para obter uma boa aproximação da solução. A aproximação de elementos finitos, baseada no método de Galerkin, é alcançada como se segue:

Multiplicando a equação (2.36) por uma função peso $w^h \in H^h$ e integrando em Ω obtemos,

$$\int_{\Omega} w^h \left(\phi c_t \frac{\partial p}{\partial t} + \nabla \cdot v_T - Q_T \right) d\Omega = 0 \quad (3.55)$$

Usando a regra do produto para diferenciação,

$$\nabla \cdot (FA) = (\nabla F) \cdot A + F(\nabla \cdot A) \quad (3.56)$$

e o teorema da divergência de Gauss, obtemos, já que $v_T \cdot \mathbf{n} = 0$ em Γ_n ,

$$\int_{\Omega} w^h \phi c_t \frac{\partial p}{\partial t} d\Omega - \int_{\Omega} \nabla w^h \cdot v_T d\Omega = \int_{\Omega} w^h Q_T d\Omega \quad (3.57)$$

Substituindo-se a expressão da velocidade na equação anterior temos que,

$$\int_{\Omega} w^h \phi c_t \frac{\partial p}{\partial t} d\Omega + \int_{\Omega} \nabla w^h \cdot \Lambda_p \nabla p d\Omega = \int_{\Omega} \nabla w^h \cdot \left[\Lambda_m \frac{dp_c}{ds_w} \nabla s_w + (\lambda_o \rho_o + \lambda_w \rho_w) \mathbf{K} \mathbf{g} \right] d\Omega \quad (3.58)$$

Por outro lado a equação da saturação é predominantemente convectiva e, neste caso, soluções obtidas pelo método de Galerkin apresentam fortes oscilações à medida que o número de Peclet, $P^e = |\beta| h/\epsilon$, aumenta. Na definição de P^e , β é o campo de velocidades, ϵ é o coeficiente de difusão e h é o tamanho característico do elemento. Para evitar o surgimento destas oscilações ou refina-se drasticamente a malha, aumentando o esforço computacional, ou se pode utilizar uma formulação adequada para tratar os efeitos dos termos convectivos. Entre as diversas técnicas existentes, adota-se neste trabalho uma formulação estabilizada de Petrov-Galerkin. Nesta formulação estabilizada,

a função peso do método de Galerkin é acrescida da correção de Petrov-Galerkin com o objetivo de acrescentar estabilidade na direção das linhas de fluxo, e do operador de captura de descontinuidade (CAU), que tem por objetivo adicionar estabilidade na direção do gradiente da solução. Definindo-se o operador diferencial associado a equação da saturação,

$$L(s_w^h, v_a^h) = \phi \frac{\partial s_w^h}{\partial t} + \nabla \cdot v_a^h + \nabla \cdot [\mathbf{D} \nabla s_w^h] + Q_w \quad (3.59)$$

a formulação de Petrov-Galerkin com operador de captura de descontinuidades é dada por:

$$\begin{aligned} 0 = & \int_{\Omega} w^h (L^h(s_w^h, v_a^h)) d\Omega + \sum_{e=1}^{Nel} \int_{\Omega^e} \tau_1 \frac{v_a^h}{\|v_a^h\|} \cdot \nabla w^h L^h(s_w^h, v_a^h) d\Omega \\ & + \sum_{e=1}^{Nel} \int_{\Omega^e} \tau_2 \frac{|L^h(s_w^h, v_a^h)|}{\|\nabla s_w^h\|} \nabla w^h \cdot \nabla s_w^h d\Omega \end{aligned} \quad (3.60)$$

onde a primeira integral na equação (3.60) representa o termo de Galerkin, o primeiro somatório de integrais nos elementos é o termo Streamline-Upwind/Petrov-Galerkin (SUPG) [25] e o segundo somatório é o operador de captura de descontinuidades do tipo CAU [22].

Os parâmetros τ_1 e τ_2 são dependentes da malha e dados por :

$$\tau_1 = \frac{\delta_1^e}{2 \|v_a^e\|} \min\left(\frac{Pe^e}{3}, 1\right) \quad (3.61)$$

$$Pe^e = \delta_1^e \frac{\|v_a^e\|^3}{(v_a^e)^T \cdot \mathbf{D} v_a^e} \quad (3.62)$$

$$\tau_2 = \frac{\delta_2^e}{\|v_{a//}^e\|} \min(Pe//^e, 0.7) \quad (3.63)$$

$$Pe//^e = \delta_2^e \frac{\|v_{a//}^e\|^3}{(v_{a//}^e)^T \cdot \mathbf{D} v_{a//}^e} \quad (3.64)$$

$$v_{a//}^e = \left(v_a^e \cdot \frac{\nabla s_w^e}{\|\nabla s_w^e\|} \right) v_a^e \quad (3.65)$$

onde δ_1^e e δ_2^e são parâmetros que dependem da malha e o superíndice e indica que a variável foi calculada no interior do elemento.

3.2 Discretização temporal

A aplicação do método dos elementos finitos resulta em um sistema acoplado de equações diferenciais ordinárias, que necessita ser integrado no tempo. Várias formas de integração no tempo são propostos na literatura, veja por exemplo AZIZ e SETTARI [4], EWING [20]. A forma mais simples é conhecida como IMPES, onde a equação da pressão é tratada de forma implícita e a equação da saturação é tratada de forma explícita. Este método reduz o custo computacional, porém impõe severas restrições ao passo de tempo, sendo indicado apenas para problemas com moderado grau de dificuldade segundo STEVENSON [53]. Neste trabalho adota-se uma filosofia de solução onde o conjunto de equações semi-discreto é resolvido por um algoritmo iterativo preditor-multicorretor [6].

Assumindo que a dependência do tempo pode ser separada da variação espacial temos,

$$p^h(\mathbf{x}, t) = \sum_{i=1}^{nnos} N_i p_i(t) \quad (3.66)$$

$$s^h(\mathbf{x}, t) = \sum_{i=1}^{nnos} N_i s_i(t) \quad (3.67)$$

onde p_i e s_i são respectivamente os valores de p^h e s^h no tempo t , no i –ésimo nó da malha, $N_i = N_i(\mathbf{x})$, $i = 1, 2, \dots, nnos$ são as funções de interpolação e $nnos$ o número de nós na malha de elementos finitos. Introduzindo (3.66) e (3.67) nas equações (3.58) e (3.60) chegamos a um sistema não linear de equações semi-discretas da forma:

$$\begin{bmatrix} \mathbf{F}_p(p, \dot{p}, s, t) \\ \mathbf{F}_s(s, \dot{s}, p, t) \end{bmatrix} = \begin{bmatrix} \mathbf{Q}_p \\ \mathbf{Q}_s \end{bmatrix} \quad (3.68)$$

onde os termos Q_p e Q_s levam em conta os termos de fonte e condições de contorno. As derivadas em relação ao tempo de p e s são indicadas respectivamente por \dot{p} e \dot{s} .

Um algoritmo de integração no tempo muito utilizado, é conhecido como método trapezoidal generalizado [25] e será utilizado neste trabalho para discretizar no tempo o sistema (3.68). A solução no tempo $n + 1$ é alcançada como segue.

Dado p_0 e s_0 calcule

$p_n, s_n, n \in \{0, 1, 2, \dots, Nsteps\}$ tal que:

$$\begin{bmatrix} \mathbf{F}_p(p_{n+1}, \dot{p}_{n+1}, s_{n+1}, t_{n+1}) \\ \mathbf{F}_s(p_{n+1}, s_{n+1}, \dot{s}_{n+1}, t_{n+1}) \end{bmatrix} = \begin{bmatrix} Q_p \\ Q_s \end{bmatrix} \quad (3.69)$$

$$\mathbf{p}_{n+1} = \mathbf{p}_n + (1 - \alpha) \Delta t \dot{\mathbf{p}}_n + \alpha \Delta t \dot{\mathbf{p}}_{n+1}$$

$$s_{n+1} = s_n + (1 - \alpha) \Delta t \dot{s}_n + \alpha \Delta t \dot{s}_{n+1}$$

onde $Nsteps$ é o número de passos de tempo e Δt é o intervalo de tempo especificado. Adota-se $\alpha = \frac{1}{2}$ o que corresponde ao método implícito de Crank-Nicolson. Eliminando-se \dot{p}_{n+1} e \dot{s}_{n+1} em \mathbf{F}_p e \mathbf{F}_s chega-se a uma equação não linear a ser resolvida a cada passo de tempo, que pode ser escrita como,

$$\begin{bmatrix} \mathbf{F}_p(p_{n+1}, s_{n+1}) \\ \mathbf{F}_s(p_{n+1}, s_{n+1}) \end{bmatrix} - \begin{bmatrix} Q_p \\ Q_s \end{bmatrix} = \mathbf{0} \quad (3.70)$$

Para resolver o sistema não linear (3.70) utiliza-se o método de Newton-Raphson, o que nos leva a,

$$\begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix}^{(i)} \begin{bmatrix} \Delta \dot{\mathbf{p}}^{(i)} \\ \Delta \dot{\mathbf{s}}^{(i)} \end{bmatrix} = \begin{bmatrix} Q_p - \mathbf{F}_p(p_{n+1}^{(i)}, s_{n+1}^{(i)}) \\ Q_s - \mathbf{F}_s(p_{n+1}^{(i)}, s_{n+1}^{(i)}) \end{bmatrix} \quad (3.71)$$

onde i é o contador de iterações do método de Newton-Raphson, e as submatrizes,

$$\mathbf{A}_{11} = \partial \mathbf{F}_p^{(i)} / \partial \mathbf{p}$$

$$\mathbf{A}_{12} = \partial \mathbf{F}_p^{(i)} / \partial \mathbf{s}$$

$$\mathbf{A}_{21} = \partial \mathbf{F}_s^{(i)} / \partial \mathbf{p}$$

$$\mathbf{A}_{22} = \partial \mathbf{F}_s^{(i)} / \partial \mathbf{s}$$

são as componentes da matriz Jacobiana, e, $\Delta \dot{\mathbf{p}}^{(i)}$, $\Delta \dot{\mathbf{s}}^{(i)}$ são os incrementos das variáveis nodais das derivadas temporais da pressão e saturação. A matriz Jacobiana é normalmente não simétrica e esparsa. A experiência indica que a resolução iterativa destes sistemas é dominada pelos termos provenientes da equação da pressão. Sendo assim, observa-se na solução iterativa dos mesmos um comportamento típico de problemas quasi-elípticos uma vez que a compressibilidade é baixa. Uma forma de se tentar otimizar os procedimentos de solução é desprezar os termos de acoplamento, isto é, resolver o problema de forma bloco-iterativa,

$$\mathbf{A}_{11} \Delta \dot{\mathbf{p}} = \mathbf{Q}_p - \mathbf{F}_p^{(i)} \left(p_{n+1}^{(i-1)}, s_{n+1}^{(i-1)} \right) \quad (3.72)$$

$$\dot{\mathbf{p}}_{n+1}^{(i)} = \dot{\mathbf{p}}_{n+1}^{(i-1)} + \Delta \dot{\mathbf{p}}_{n+1}^{(i)}$$

$$\mathbf{p}_{n+1}^{(i)} = \mathbf{p}_{n+1}^{(i-1)} + \alpha \Delta t \Delta \dot{\mathbf{p}}^{(i)}$$

$$\mathbf{A}_{22} \Delta \dot{\mathbf{s}} = \mathbf{Q}_s - \mathbf{F}_s^{(i)} \left(p_{n+1}^{(i-1)}, s_{n+1}^{(i-1)} \right) \quad (3.73)$$

$$\dot{\mathbf{s}}_{n+1}^{(i)} = \dot{\mathbf{s}}_{n+1}^{(i-1)} + \Delta \dot{\mathbf{s}}_{n+1}^{(i)}$$

$$\mathbf{s}_{n+1}^{(i)} = \mathbf{s}_{n+1}^{(i-1)} + \alpha \Delta t \Delta \dot{\mathbf{s}}^{(i)}$$

Os vetores $\mathbf{F}_p^{(i)}$, $\mathbf{F}_s^{(i)}$ e a matriz \mathbf{A}_{22} dependem da velocidade v^h , equação (2.37), que por sua vez é função da pressão e da saturação da água. Uma forma de calcular o campo de velocidades é usar diretamente as leis de Darcy, isto é, a equação (2.37). No entanto este procedimento apresenta baixa precisão, uma vez que o campo de velocidades é aproximado calculando-se diretamente os gradientes de pressão, o que pode comprometer a aproximação da saturação.

3.3 Pós-processamento do campo de velocidades

Com o objetivo de melhorar a aproximação do campo de velocidades utiliza-se a técnica de pós-processamento desenvolvida por MALTA, LOULA e GARCIA [35] para escoamentos miscíveis. Esta técnica baseia-se na formulação variacional das leis de Darcy combinada com a equação da conservação de massa. A análise numérica desta técnica, indica um ganho nas taxas de convergência para o campo de velocidades em relação a métodos mistos do tipo Raviart-Thomas de $O(h^{0.5})$. No presente trabalho vamos adaptar diretamente as idéias de MALTA, LOULA e GARCIA [35], sem no entanto efetuar uma análise numérica da técnica. Portanto sendo p^h o campo de pressão calculado pelo método de Galerkin e

$$E^h = \{\mathbf{w}^h \in H^h \times H^h, \mathbf{w}^h = 0 \text{ em } \Gamma\}$$

calcular $\mathbf{v}_p^h \in E^h$ tal que,

$$0 = \int_{\Omega} \mathbf{w}^h \cdot (\Lambda_p^{-1} v_p^h + \nabla p^h) d\Omega + \sum_{e=1}^{Nel} \int_{\Omega^e} \delta_e \nabla \cdot \mathbf{w}^h \left[\nabla \cdot \left(v_p^h + \Lambda_m \frac{dp_c}{ds_w} \nabla s_w - \mathbf{K} (\rho_o \lambda_o + \rho_w \lambda_w) \mathbf{g} \right) - Q_T + \phi c_t \frac{\partial p}{\partial t} \right] d\Omega \quad (3.74)$$

onde δ_e é um parâmetro que depende da malha, geralmente tomado como $h^e/2$. Com a utilização desta técnica de pós-processamento observa-se que as variáveis p^h , s^h e v^h são aproximadas por interpolações padrão de mesma ordem. Sendo o cálculo das velocidades efetuado logo após a atualização do campo de pressão.

Com isto obtemos a forma final para os sistemas de equações diferenciais ordinárias resultantes da discretização por elementos finitos das equações (3.58), (3.74), (3.60)

$$[\mathbf{M}] \dot{\mathbf{p}} + [\mathbf{K}] \mathbf{p} = \mathbf{F}_p \quad (3.75)$$

$$\tilde{\mathbf{K}} \mathbf{v}_p = \tilde{\mathbf{F}}_v \quad (3.76)$$

$$[\bar{\mathbf{M}}] \dot{\mathbf{s}} + [\bar{\mathbf{K}}] \mathbf{s} = \bar{\mathbf{F}}_s \quad (3.77)$$

As matrizes \mathbf{M} , \mathbf{K} e o vetor \mathbf{F}_p , são as matrizes de massa e de Galerkin para a equação da pressão (3.58), ambas simétricas, e seu correspondente vetor de termos independentes. A matriz $\tilde{\mathbf{K}}$, simétrica, e o vetor \mathbf{F}_v correspondem à forma discreta (3.74). As matrizes $\bar{\mathbf{M}}$ e $\bar{\mathbf{K}}$, não simétricas, são, respectivamente, as matrizes correspondentes a formulação estabilizada para a equação da saturação (3.60) e \mathbf{F}_s seu vetor de termos independentes. Os termos independentes levam em conta a presença de termos fontes/sumidouros (se houverem) e a influência das condições de contorno.

A solução em cada passo de tempo, $t_i \in [0, T]$ é calculada de forma implícita utilizando o método trapezoidal generalizado, conforme [25], através do seguinte algoritmo bloco-iterativo preditor-multicorretor:

Inicialize: $\mathbf{p}_{n+1}^0 = \mathbf{p}_n^0$; $\dot{\mathbf{p}}_{n+1}^0 = \dot{\mathbf{p}}_n^0$; $\mathbf{v}_{n+1}^0 = \mathbf{v}_n^0$; $u_{n+1}^0 = u_n^0$; $\dot{\mathbf{u}}_{n+1}^0 = \dot{\mathbf{u}}_n^0$

Para $i = 0, 1, 2, \dots$ *max iter*, **Faça**

Bloco 1 : Resolve a equação da pressão

$$\mathbf{M}_{n+1}^* \Delta \dot{\mathbf{p}}_{n+1}^{i+1} = \mathbf{R}_{n+1}^{i+1}$$

onde \mathbf{M}_{n+1}^* é uma matriz simétrica definida como

$$\mathbf{M}_{n+1}^* = \mathbf{M} + \alpha \Delta t \mathbf{K}$$

$$\mathbf{R}_{n+1}^{i+1} = \mathbf{F}_{n+1} - \mathbf{M} \dot{\mathbf{p}}_{n+1}^i - \mathbf{K} \mathbf{s}_{n+1}^i (s_{n+1}^i) \mathbf{p}_{n+1}^i \quad (\text{vetor de resíduos})$$

Atualizar a pressão e sua derivada:

$$\mathbf{p}_{n+1}^{i+1} = \mathbf{p}_{n+1}^i + \alpha \Delta t \Delta \dot{\mathbf{p}}_{n+1}^{i+1}$$

$$\dot{\mathbf{p}}_{n+1}^{i+1} = \dot{\mathbf{p}}_{n+1}^i + \Delta \dot{\mathbf{p}}_{n+1}^{i+1}$$

Bloco2: Calcula o campo de velocidades

$$\tilde{\mathbf{K}} (s_{n+1}^i) \mathbf{v}_{n+1}^{i+1} = \mathbf{F} (\mathbf{p}_{n+1}^{i+1})$$

Bloco3: Resolve a equação da saturação

$$\bar{\mathbf{M}}_{n+1}^* \Delta \dot{\mathbf{s}}_{n+1}^{i+1} = \bar{\mathbf{R}}_{n+1}^{i+1}$$

onde $\bar{\mathbf{M}}_{n+1}^*$ é uma matriz não simétrica definida como

$$\bar{\mathbf{M}}_{n+1}^* = \bar{\mathbf{M}} + \alpha \Delta t \bar{\mathbf{K}}$$

$$\bar{\mathbf{R}}_{n+1}^{i+1} = \bar{\mathbf{F}}_{n+1} - \bar{\mathbf{M}} (\dot{\mathbf{s}}_{n+1}^i) \dot{\mathbf{s}}_{n+1}^i - \bar{\mathbf{K}} (\mathbf{s}_{n+1}^i, \mathbf{p}_{n+1}^i, \mathbf{v}_{n+1}^{i+1}) \mathbf{s}_{n+1}^i \text{ (vetor de resíduos)}$$

Atualizar a saturação e sua derivada:

$$\mathbf{s}_{n+1}^{i+1} = \mathbf{s}_{n+1}^i + \alpha \Delta t \Delta \dot{\mathbf{s}}_{n+1}^{i+1}$$

$$\dot{\mathbf{s}}_{n+1}^{i+1} = \dot{\mathbf{s}}_{n+1}^i + \Delta \dot{\mathbf{s}}_{n+1}^{i+1}$$

Fim

No algoritmo acima, i é o contador de iterações Δt é o passo de tempo e α é o parâmetro que controla precisão e estabilidade da integração no tempo. Normalmente assume-se $\alpha = \frac{1}{2}$ o que conduz a um método implícito incondicionalmente estável. O processo iterativo continua até que o seguinte critério de convergência seja satisfeito.

$$\frac{\|\Delta \dot{\mathbf{s}}_{n+1}^{i+1}\|}{\|\dot{\mathbf{s}}_{n+1}^{i+1}\|} \leq \text{etol} \quad (3.78)$$

onde o parâmetro *etol* é fornecido pelo usuário.

3.4 Controle do Passo de Tempo

Em problemas transientes, a escolha do passo de tempo Δt é uma decisão importante. Um passo de tempo muito grande pode comprometer a exatidão da solução procurada, enquanto que um passo de tempo muito pequeno, eleva o custo computacional. O ideal é obter uma solução de boa qualidade com o maior passo de tempo possível determinando o valor de Δt mais conveniente para o problema. Uma estratégia automática para seleção do passo de tempo é apresentada em COUTINHO e ALVES [6], onde se procura encontrar a cada intervalo de tempo o Δt mais conveniente para o problema.

A estratégia baseia-se em um estimador de erro *a posteriori* da solução. Após o cálculo da solução em um passo de tempo, calcula-se o erro. Se o erro é muito grande a solução é rejeitada, determina-se o novo Δt , retorna-se para o passo de tempo anterior e tenta-se novamente com o novo Δt . Um erro muito grande indica que o passo de tempo deve ser diminuído, caso contrário deve ser aumentado. Se o erro na solução é aceitável, continua-se com o algoritmo de integração no tempo.

O algoritmo é composto de duas etapas fundamentais:

1. Um estimador de erro para a solução calculada;
2. Um algoritmo que usa este erro para rejeitar ou aceitar o passo de tempo e calcular o novo Δt .

Dois algoritmos para seleção do passo de tempo são apresentados em [6]. O primeiro é dado por,

$$\Delta t_{n+1} = \left(\frac{tol}{\varepsilon_n} \right) \Delta t_n \quad (3.79)$$

onde *tol* é uma tolerância ou mudança desejada, suprida pelo usuário, e ε_n é uma medida da mudança no passo de tempo Δt_n .

O segundo algoritmo avalia o novo passo de tempo pela relação,

$$\Delta t_{n+1} = \left(\frac{\varepsilon_{n-1}}{\varepsilon_n} \right)^{k_P} \left(\frac{tol}{\varepsilon_n} \right)^{k_I} \left(\frac{\varepsilon_{n-1}^2}{\varepsilon_n \varepsilon_{n-2}} \right)^{k_D} \Delta t_n \quad (3.80)$$

onde os parâmetros k_P , k_I , k_D , devem ser ajustados de forma a minimizar o esforço computacional. A equação (3.80) é obtida supondo-se que o problema de seleção do passo de tempo é um problema de controle ótimo, onde emprega-se um controlador Proporcional, Integral e Derivativo. Neste trabalho os valores adotados para estes parâmetros são: $k_P = 0.05$, $k_I = 0.15$ e $k_D = 0.01$. Sugere-se a leitura de [6] para uma discussão mais aprofundada sobre este algoritmo.

A medida da mudança da quantidade de interesse, ε , no passo de tempo Δt_{n+1} , é avaliada usando-se as normas,

$$\varepsilon'_p = \frac{\|\mathbf{p}_{n+1} - \mathbf{p}_n\|}{\|\mathbf{p}_{n+1}\|}; \varepsilon_p = \frac{\varepsilon'_p}{tol_p} \quad (3.81)$$

$$\varepsilon'_{\dot{p}} = \frac{\|\dot{\mathbf{p}}_{n+1} - \dot{\mathbf{p}}_n\|}{\|\dot{\mathbf{p}}_{n+1}\|}; \varepsilon_{\dot{p}} = \frac{\varepsilon'_{\dot{p}}}{tol_{\dot{p}}} \quad (3.82)$$

$$\varepsilon'_s = \frac{\|\mathbf{s}_{n+1} - \mathbf{s}_n\|}{\|\mathbf{s}_{n+1}\|}; \varepsilon_s = \frac{\varepsilon'_s}{tol_s} \quad (3.83)$$

$$\varepsilon'_{\dot{s}} = \frac{\|\dot{\mathbf{s}}_{n+1} - \dot{\mathbf{s}}_n\|}{\|\dot{\mathbf{s}}_{n+1}\|}; \varepsilon_{\dot{s}} = \frac{\varepsilon'_{\dot{s}}}{tol_{\dot{s}}} \quad (3.84)$$

e tomando ε como, $\varepsilon = \max(\varepsilon_p, \varepsilon_{\dot{p}}, \varepsilon_s, \varepsilon_{\dot{s}})$.

As quantidades $\varepsilon_p, \varepsilon_{\dot{p}}, \varepsilon_s, \varepsilon_{\dot{s}}$ representam, respectivamente as mudanças das grandezas nodais de pressão, saturação e derivada com relação ao tempo da saturação. As tolerâncias $tol_p, tol_{\dot{p}}, tol_s$ e $tol_{\dot{s}}$ são fornecidas pelo usuário.

3.5 Implementação Elemento por Elemento

Para problemas práticos de engenharia os métodos diretos, tipo eliminação de Gauss, tornam-se impraticáveis em função de seu alto custo em termos de memória requerida e tempo de CPU [3]. A alternativa são os métodos iterativos, que além de requererem uma quantidade menor de memória são adequados para implementação usando técnicas elemento por elemento, o que os torna altamente interessante para utilização com o método dos elementos finitos. No momento atual a utilização de métodos iterativos e técnicas elemento por elemento estão presentes em boa parte dos códigos que buscam eficiência na solução de problemas de grande porte. Outro aspecto importante da implementação elemento por elemento é que esta favorece a implementação de refinamento adaptativo de malha, onde graus de liberdade são adicionados ou eliminados durante a análise e as conectividades dos elementos são rearranjadas a cada atualização da malha. Neste tra-

balho optou-se por uma implementação elemento por elemento e por técnicas iterativas para resolver os sistemas de equações lineares oriundos da discretização por elementos finitos do conjunto de equações diferenciais governante.

Para integração no tempo do conjunto de equações semi-discretas (3.75), (3.76) e (3.77) através do algoritmo bloco-iterativo, três sistemas de equações lineares devem ser resolvidos a cada iteração. Estes sistemas, por sua vez, são resolvidos através de técnicas iterativas. Uma vantagem da estratégia por bloco é permitir a escolha do algoritmo adequado para cada sistema de equações lineares a ser resolvido.

Da mesma forma que em [6], para a solução do sistema de equações no bloco 1, correspondente à equação da pressão, utilizou-se o método dos gradientes conjugados. Para o bloco 2, pós-processamento do campo de velocidades, o método de Jacobi, e o bloco 3, correspondente à equação da saturação, é resolvido pelo algoritmo GMRES. A descrição destes algoritmos encontra-se, por exemplo, em SAAD [45]. De forma geral, os métodos iterativos se baseiam em operações vetor-vetor tais como, produtos escalares, atualização de vetor, etc., e produtos matriz-vetor.

Na técnica elemento por elemento a matriz de coeficientes e o vetor de termos independentes nunca são montados, sendo o produto matriz-vetor computado como,

$$\mathbf{Ax} = \sum_{e=1}^{nel} \mathbf{A}_e \mathbf{x}_e \quad (3.85)$$

onde \mathbf{A}_e é a matriz de elemento e \mathbf{x}_e as componentes do vetor \mathbf{x} restrita ao elemento e .

A implementação da multiplicação matriz-vetor na forma elemento por elemento é executada em três passos como segue,

1. Localiza as componentes do vetor de elementos local no vetor global:

$$\mathbf{x}_e \leftarrow \mathbf{x}$$

2. Executa a operação a nível de elemento:

$$\mathbf{u}_e = \mathbf{A}_e \mathbf{x}_e$$

3. Espalha as componentes do vetor local no vetor global:

$$\mathbf{u} \leftarrow \mathbf{u} + \mathbf{u}_e$$

Este algoritmo pode ser executado de forma paralela ou vetorial sem maiores dificuldades. Os passos 1 e 2 podem ser executados individualmente para cada elemento. O passo três, que envolve escrita sobre o vetor global, só pode ser executado em grupos de elementos disjuntos. Se este cuidado não for tomado os resultados serão errados, em função do compartilhamento de nós pelos elementos adjacentes.

A distribuição de elementos em blocos disjuntos é obtida a partir de um algoritmo de coloração de malha descrito em [26][6]. Este algoritmo garante que nenhum elemento de um bloco possui qualquer nó em comum com qualquer elemento de outro bloco. Com isto os três passos do algoritmo podem ser executados de forma paralela ou vetorial.

3.5.1 Pré-condicionamento

Para acelerar a convergência do método dos gradientes conjugados e do GMRES, empregou-se o pré-condicionador elemento por elemento descrito a seguir.

Seja o sistema de equações lineares

$$\mathbf{A} \mathbf{u} = \mathbf{b} \tag{3.86}$$

e \mathbf{P} uma matriz simétrica, positiva definida dada por,

$$\mathbf{P} = \mathit{diag}(\mathbf{A}) \tag{3.87}$$

Transformamos o sistema (3.86) para a forma,

$$\bar{\mathbf{A}}\bar{\mathbf{u}} = \bar{\mathbf{b}} \quad (3.88)$$

onde

$$\bar{\mathbf{A}} = \mathbf{P}^{-\frac{1}{2}}\mathbf{A}\mathbf{P}^{-\frac{1}{2}} \quad (3.89)$$

$$\bar{\mathbf{u}} = \mathbf{P}^{-\frac{1}{2}}\mathbf{u} \quad (3.90)$$

$$\bar{\mathbf{b}} = \mathbf{P}^{-\frac{1}{2}}\mathbf{b} \quad (3.91)$$

Estas operações sobre o sistema original (3.86) são consideradas como um pré-condicionamento. Em seguida, em cada iteração aplica-se o preconditionador elemento-por-elemento,

$$\mathbf{L}^{-1}\bar{\mathbf{A}}\mathbf{U}^{-1}\mathbf{x} = \mathbf{L}^{-1}\bar{\mathbf{b}} \quad (3.92)$$

onde

$$\mathbf{L} = \prod_{e=1}^{nel} \mathbf{L}_e; \quad \mathbf{U} = \prod_{e=1}^{nel} \mathbf{U}_e \quad (3.93)$$

As matrizes \mathbf{L}_e e \mathbf{U}_e são obtidas da partição Gauss-Seidel da matriz $\bar{\mathbf{A}}$, tal que,

$$\mathbf{L}_e + \mathbf{U}_e = \bar{\mathbf{A}}_e + \mathbf{I} \quad (3.94)$$

O preconditionador resultante é conhecido como preconditionador elemento-por-elemento Gauss-Seidel. Note que o preconditionador não envolve nenhuma área de memória adicional além das matrizes de elemento. Além disso, no caso simétrico, $\mathbf{U}_e = \mathbf{L}_e^T$. Outra alternativa para resolver a equação da pressão é usar um solucionador multimalha.

3.5.2 Método Multimalha

Uma forma de resolver o sistema de equações lineares,

$$\mathbf{A}\mathbf{u} = \mathbf{f} \quad (3.95)$$

é usando o método multimalha, cuja idéia central é o uso de um conjunto de malhas, M_1, M_2, \dots, M_k , com espaçamentos $h_1 > h_2 > \dots > h_k$, respectivamente, para aproximar o mesmo domínio Ω do problema. Um aspecto importante deste método é a transferência de informações entre as diversas malhas. Esta questão é resolvida através da definição de operadores de restrição e interpolação.

(i) Operador de restrição R , que transfere as informações da malha *fina* , Ω_h , para a malha *grossa* , Ω_{2h} :

$$\Omega_{2h} = R(\Omega_h) \quad (3.96)$$

(ii) Operador de interpolação ou prolongamento P , que transfere informações da malha *grossa* Ω_{2h} , para malha *fina* Ω_h :

$$\Omega_h = P(\Omega_{2h}) \quad (3.97)$$

A forma destes operadores pode variar com o tipo de elemento ou com o tipo de problema. Considere por exemplo uma discretização de elementos finitos triangulares lineares conforme mostra a Figura 4. Os operadores de restrição e interpolação podem ser definidos, para uma malha aninhada, como segue:

Operador de interpolação P

$$u_i^h = u_i^{2h}; \text{ para } i = 1, 2, 3$$

$$u_j^h = \frac{1}{2} (u_m^{2h} + u_n^{2h}); \text{ para } j = 4, 5, 6; m = 1, 2, 3; n = 2, 3, 1$$

Operador de restrição R

$$u_i^{2h} = \frac{1}{2} [u_i^h + \frac{1}{2} (u_m^h + u_n^h)]; \text{ para } i = 1, 2, 3; m = 4, 4, 5; n = 6, 5, 6$$

De posse da definição dos operadores de restrição e interpolação podemos sintetizar o conceito do método como segue. Seja u a solução exata do sistema dado pela Equação

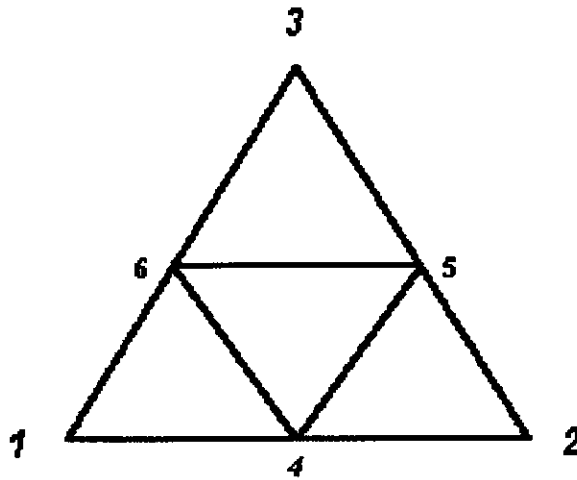


Figura 4: Operadores de restrição e interpolação. Os nós 1, 2 e 3 pertencem à malha *grossa*, os nós 4, 5 e 6 pertencem à malha *fin*a.

(3.95) e v uma aproximação de u . Assim o erro é definido por,

$$e = u - v \quad (3.98)$$

que não pode ser computado, pois não conhecemos a solução exata u . Uma medida que pode ser calculada é o resíduo,

$$r = f - Av \text{ ou } Av = f - r \quad (3.99)$$

que nos dá uma idéia do quanto v está próximo de u . Subtraindo esta equação da original, Equação (3.95), temos,

$$Ae = r \quad (3.100)$$

conhecida como equação do resíduo. Usamos a equação do resíduo (3.100) para obter uma aproximação para o erro na malha mais *grossa* que será usada para corrigir a solução na malha mais *fin*a. Através de recursão, pode-se definir uma família de métodos multimalha, chamada Ciclo- μ , que é dada no algoritmo abaixo:

$$\text{Ciclo-}\mu : \mathbf{v}^h \leftarrow M\mu^h(\mathbf{v}^h, \mathbf{f}^h)$$

1. Relaxar ν_1 vezes em $\mathbf{A}\mathbf{u}^h = \mathbf{f}^h$ com aproximação inicial \mathbf{v}^h
2. Se $\Omega^h =$ malha mais grossa, então vá para o passo 4

$$\text{Senão } \mathbf{f}^{2h} \leftarrow R_h^{2h}(\mathbf{f}^h = \mathbf{A}^h \mathbf{v}^h)$$

$$\mathbf{v}^{2h} \leftarrow 0$$

$$\mathbf{v}^{2h} \leftarrow M\mu(\mathbf{v}^{2h}, \mathbf{f}^{2h}), \mu \text{ vezes}$$

3. Corrigir $\mathbf{v}^h \leftarrow \mathbf{v}^h + P_{2h}^h \mathbf{v}^{2h}$
4. Relaxar ν_2 vezes em $\mathbf{A}\mathbf{u}^h = \mathbf{f}^h$ com aproximação inicial \mathbf{v}^h

Os dois tipos de ciclos multimalha mais usados são o ciclo V , quando $\mu = 1$ e o ciclo W , quando $\mu = 2$. Os valores de ν_1 e ν_2 que determinam o número de relaxações em cada passo são fixados no início do algoritmo, e seus valores geralmente variam entre 1 e 5. As matrizes \mathbf{A}^h são calculadas para cada malha. Como método de relaxação emprega-se o método dos gradientes conjugados com condicionamento diagonal.

3.5.3 Imposição das Condições de Contorno no Pós-processamento das Velocidades

As velocidades pós-processadas são obtidas por iterações de Jacobi. A condição de fluxo nulo é aplicada através da introdução de uma técnica de projeção no processo iterativo, conforme DEVLOO [13]. O cálculo do campo de velocidades e aplicação das condições de contorno são executadas de forma muito compacta através de dois *loops*, como ilustrado no algoritmo a seguir, onde $\mathbf{D} = \text{diag}(\tilde{\mathbf{K}})$ e \mathbf{n}_i o vetor normal unitário em algum nó i localizado na fronteira

Para $n = 1, 2, \dots, \text{até } iter$

$$\mathbf{v}_{n+1} = \mathbf{v}_n + \mathbf{D}^{-1}(\tilde{\mathbf{F}} - \tilde{\mathbf{K}}\mathbf{v}_n)$$

Para todo i na fronteira faça

$$\mathbf{v}_{n+1}^i = \mathbf{v}_{n+1}^i - (\mathbf{v}_{n+1}^i \cdot \mathbf{n}_i) \mathbf{n}_i$$

Fim

Fim

Sendo a matriz \mathbf{D} diagonal, o processo iterativo é trivial, sendo que as matrizes de elemento nunca são efetivamente armazenadas.

3.6 Validação da Solução da Pressão

Para avaliar a precisão e desempenho do método multimalha, resolvemos a equação da pressão no problema de cinco poços para o caso incompressível. Neste problema clássico, um poço central é usado como injetor e os outros quatro à sua volta como produtores. O domínio do problema é o quadrado unitário correspondendo a um quarto do modelo de cinco poços. Para este teste assumiu-se $k_x = k_y = 1.0$, o que corresponde a um meio homogêneo. O objetivo é comparar a solução obtida utilizando o método multimalha com a solução exata, que longe dos poços é dada, de acordo com TEIGLAND [55], pela equação,

$$p(x, y) = \frac{1}{4\pi} \sum_{m=-\infty}^{\infty} (-1^{|m|}) \ln \left\{ \frac{\cosh\left(\frac{\pi}{2}(x+y-2m)\right) + \cos\left(\frac{\pi}{2}(x-y)\right)}{\cosh\left(\frac{\pi}{2}(x+y-2m)\right) - \cos\left(\frac{\pi}{2}(x-y)\right)} \right\} \quad (3.101)$$

Sendo $p(x, y)$ a solução exata dada pela Equação (3.101) e $\bar{p}(x, y)$ a solução computada através do método numérico, o erro é dado por,

$$\varepsilon_p = \left\{ \sum_{(x,y) \in \Omega} h^2 (p(x, y) - \bar{p}(x, y))^2 \right\}^{\frac{1}{2}} \quad (3.102)$$

sendo que para fins de cálculo é excluída a região em torno da singularidade, isto é, o domínio do problema é redefinido como,

$$\Omega_E = [0, 1] \times [0, 1] - \left(\left[0, \frac{1}{8}\right] \times \left[0, \frac{1}{8}\right] + \left[\frac{7}{8}, 1\right] \times \left[\frac{7}{8}, 1\right] \right) \quad (3.103)$$

Os resultados obtidos com o método multimalha que implementamos são mostrados na Tabela 1 e 2. A Tabela 1 mostra os resultados para um refinamento uniforme, enquanto que a Tabela 2 apresenta os resultados obtidos com o método multimalha usando a malha com refinamento não uniforme mostrada na Figura 5. Nas Tabelas 1 e 2, Nel_f é o número de elementos na malha mais fina. Em ambos os casos utilizamos o ciclo- V com $\nu_1 = \nu_2 = 1$. A tolerância dos métodos iterativos foi fixada em 10^{-4} . Ao solucionarmos este problema com o ciclo- W notamos que os resultados foram os mesmos. Podemos notar que quanto maior é o número de níveis, menor o erro, tanto para os casos com malhas uniformes, quanto para os casos com malhas não uniformes. Além disso, ao compararmos os nossos resultados com aqueles de TEIGLAND [55], podemos observar que ambos estão em boa concordância.

Tabela 1: Resultados obtidos pelo método multimalha para o problema de cinco poços usando uma malha com refinamento uniforme.

Elementos na Malha Inicial	Níveis	ϵ_p	Nel_f
2	5	$3.6e - 04$	512
2	6	$8.6e - 05$	2048
8	6	$2.1e - 05$	8192

Tabela 2: Resultados obtidos pelo método multimalha para o problema de cinco poços usando uma malha com refinamento não uniforme.

Elementos na Malha Inicial	Níveis	ϵ_p	Nel_f
24	3	$6.3e - 04$	384
24	4	$1.6e - 04$	1536
24	5	$4.2e - 05$	6144

A Figura 6 mostra a comparação da solução analítica e numérica ao longo da linha entre o poço de injeção e o poço de produção. Nesta comparação empregou-se a solução computada usando-se o método multimalha em malha com refinamento uniforme, ciclo- V e seis níveis. Pode-se observar que os resultados são praticamente coincidentes.

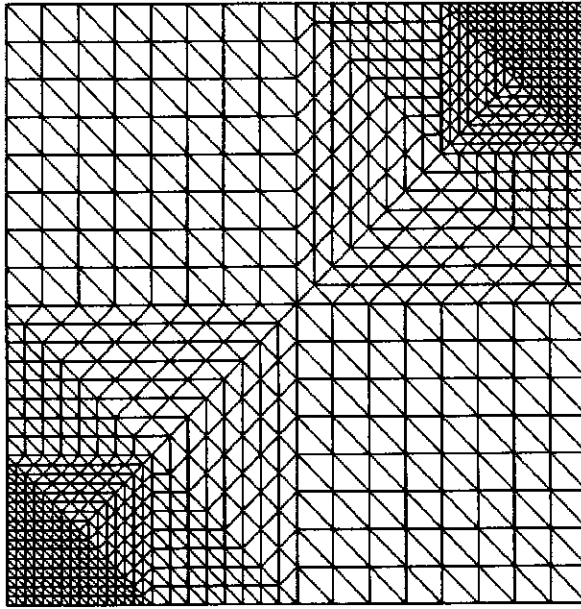


Figura 5: Malha com refinamento não uniforme utilizada na solução com o método multimalha.

Para comparar o desempenho computacional do método multimalha com o método dos gradientes conjugados resolvemos o problema de cinco poços incompressível, considerando as viscosidades $\mu_o = \mu_w = 1.0$, as densidades $\rho_o = \rho_w = 1.0$ e a porosidade $\phi = 0.2$. O tensor de permeabilidade absoluta é $\mathbf{K} = \mathbf{I}$. Assumiu-se ainda que as permeabilidades relativas da água e do óleo dadas, respectivamente, por $k_{rw} = s_w^2$ e $k_{ro} = (1 - s_w)^2$. Os efeitos da gravidade e pressão foram desprezados neste exemplo e empregou-se malha com refinamento uniforme. A tolerância para os métodos iterativos foi fixada em 10^{-4} e tolerância para variação não linear em 10^{-1} . Dois casos foram analisados. No primeiro caso usou-se uma malha com 16 divisões na malha mais fina e 5 níveis, e no segundo caso, uma malha com 32 divisões na malha mais fina e 6 níveis. O problema foi solucionado durante 1000 passos de tempo. A Tabela 3 mostra o tempo médio normalizado correspondente à solução da equação da pressão pelo método multimalha, identificado por MG e pelo método dos gradientes conjugados com precon-

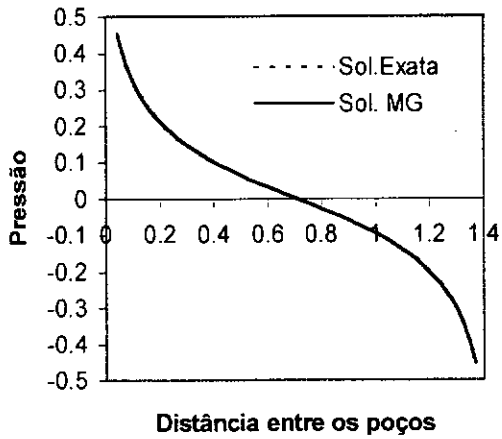


Figura 6: Comparação da solução exata e da solução multimalha com ciclo-V, $\nu_1 = \nu_2 = 1$ em uma malha uniforme com 32 divisões na malha mais fina.

dicionador elemento-por-elemento Gauss-Seidel, identificado simplesmente como PCG. Podemos notar que para este problema o método multimalha é mais eficiente do ponto de vista computacional que o método dos gradientes conjugados utilizado.

Tabela 3: Tempo de processamento médio normalizado por iteração não linear para a solução da pressão no problema de cinco poços. Malha 1: 512 elementos; Malha 2: 2048 elementos.

Método	Tempo de processamento	
	Malha 1	Malha 2
MG	0.22	0.12
PCG	1.0	1.0

3.7 Validação do Esquema de Estabilização

3.7.1 Caso Unidimensional

Para avaliar a formulação estabilizada descrita na Seção 3.1, vamos apresentar resultados para a solução da equação da saturação em um problema unidimensional. Neste exemplo não foram considerados os efeitos da pressão capilar e nem termos fonte. A velocidade total é $\mathbf{v} = \mathbf{i}_x$ e a saturação da água em $x = 0$ é fixa em $s_w = 1.0$. No primeiro caso adotou-se $k_{rw} = s_w^2$; $k_{ro} = (1 - s_w)^2$; $\mu_o/\mu_w = 5$ e os efeitos da gravidade não foram incluídos, isto é $G_d = 0$. Os perfis de saturação obtidos neste caso para $t = 0.5$ Volume Poroso Injetado (VPI) encontram-se na Figura 7, para simulação empregando uma malha uniforme com 80 elementos lineares. Na Figura 7, a solução com o método SUPG é identificada simplesmente por SUPG. Quando o operador de captura de descontinuidade é levado em consideração as soluções são identificadas por SUPG+CAU. Também na Figura 7 encontram-se duas soluções com a formulação CAU somente. Note que em algumas soluções a contribuição do termo CAU foi ponderada por $\delta_2^e = h^e/2$ enquan-

to que na outra solução o mesmo termo foi ponderado por $\delta_2^e = h^e$. Em todos os casos $\delta_1^e = h^e/2$. Observou-se na Figura 7 que a solução com o método SUPG é menos difusiva, porém com oscilações espúrias no pé da frente. Já as soluções com a presença do operador de captura de descontinuidade são mais difusivas, especialmente aquelas onde somente o operador foi utilizado. De acordo com DURLOFSKY [17], a altura teórica para o choque neste caso é 0.41, o que está em boa concordância com as soluções numéricas obtidas com as formulações SUPG+CAU e CAU.

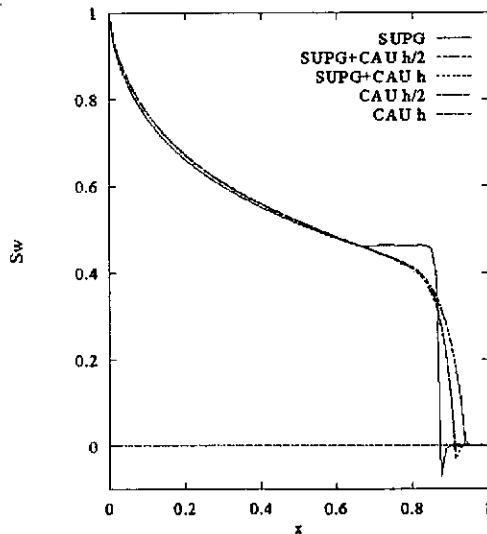


Figura 7: Frente de saturação de água em $t = 0.5$ VPI para o problema unidimensional sem gravidade.

O próximo exemplo considera os efeitos gravitacionais. Neste caso $G_d = 2$. Considera-se ainda que os efeitos gravitacionais agem na direção oposta à \mathbf{v} , ou seja, na direção negativa do eixo x . Os perfis de saturação para $t = 0.5$ VPI para as diferentes simulações encontram-se na Figura 8. Novamente a solução SUPG apresentou oscilações indesejáveis, enquanto que as soluções SUPG+CAU e CAU estão em boa concordância com a altura teórica do choque, que é de 0.76 para este caso. Ainda verifica-se que as

oscilações espúrias são comparativamente menos acentuadas do que no caso sem gravidade.

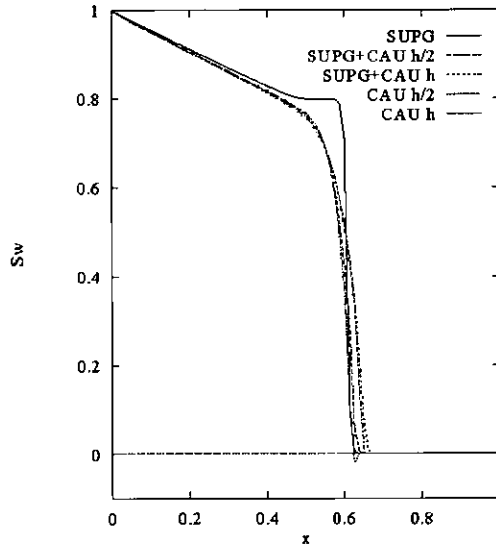


Figura 8: Frente de saturação de água em $t = 0.5$ VPI para o problema unidimensional com gravidade

3.7.2 Problema de Cinco Poços

Seja o problema da simulação do escoamento de água e óleo numa configuração do arranjo de cinco poços com dupla simetria. Neste caso, o domínio computacional é uma quadrado de lado unitário, com um poço injetor e um poço produtor alinhados na diagonal principal. Considera-se o reservatório homogêneo, com permeabilidade absoluta constante e isotrópica, ou seja, $\mathbf{K} = \mathbf{I}$. A porosidade do reservatório é constante e seu valor é irrelevante, pois serve apenas como escalonamento do tempo. Tomou-se $c_t = 10^{-6}$ e não foram considerados os efeitos da gravidade e da pressão capilar. Neste exemplo $k_{rw} = s_w^2$; $k_{ro} = (1 - s_w)^2$ e $\mu_o/\mu_w = 4$. Os resultados serão apresentados em termos de Volumes Porosos Injetados (VPI), que é uma grandeza análoga ao tempo adimensional-

zado. Adota-se uma malha regular composta de 20×20 células, onde cada célula é subdividida em dois elementos triangulares lineares, totalizando 800 elementos. Dois cenários foram considerados. O primeiro, identificado como diagonal, os poços injetor e produtor estão na diagonal perpendicular às linhas diagonais de disposição dos elementos triangulares. O segundo, identificado como paralelo, possui a orientação das linhas diagonais dos elementos paralela à diagonal poço injetor-poço produtor. Idealmente, os resultados para ambas configurações deveriam ser idênticos. Entretanto, devido a um maior ou menor efeito de orientação de malha, os resultados poderão diferir para os dois cenários. Além disso este exemplo permite validar a solução do problema pressão \times velocidade e seu acoplamento com a equação da saturação.

Os resultados para a recuperação de óleo normalizada pelo volume total injetado para os dois cenários encontram-se respectivamente nas Figuras 9 e 10. Pode-se notar nestas Figuras que para os dois cenários as soluções com a formulação SUPG apresentaram oscilações indesejáveis, notadamente na configuração diagonal. Os resultados com as formulações com operador de captura de descontinuidade não apresentam oscilações.

A Figura 11 mostra, para a formulação CAU ponderada por $\delta_2^e = h^e/2$, uma comparação da recuperação de óleo obtida para os dois cenários de escoamento. Verifica-se que estas formulações exibem pouco efeito de orientação de malha. Resultados semelhantes foram observados para as formulações SUPG+CAU ($h^e/2$ e h^e) e CAU h^e .

Além disso, as simulações SUPG+CAU e CAU estão em boa concordância com os resultados de DURLOFSKY [17], que emprega uma formulação combinando elementos finitos mistos para a equação da pressão e volumes finitos (TVD de segunda ordem) para a equação da saturação. As Figuras 12 e 13 apresentam as curvas de volume de óleo recuperado (VOR) respectivamente para as duas orientações de malha.

Nota-se que todas as formulações conduziram a uma boa recuperação do volume de óleo, com muito pouca diferença entre as soluções para as orientações diagonal e paralela, conforme evidenciado na Figura 14, que apresenta uma comparação direta para o volume de óleo recuperado obtido com a formulação CAU $h^e/2$ em ambos cenários de escoamento. Resultados semelhantes também foram observados para as formulações SUPG+CAU.

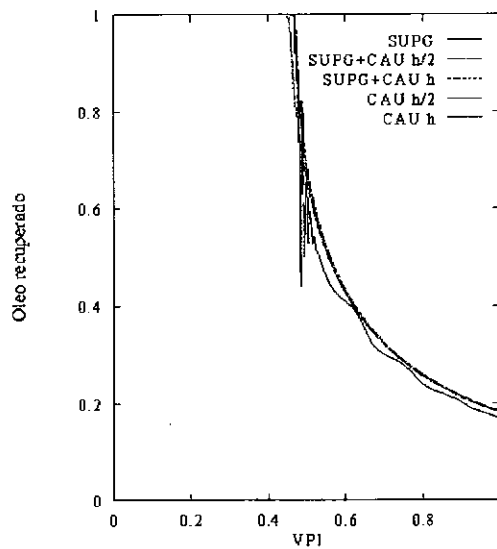


Figura 9: Recuperação de óleo para o problema de cinco poços na configuração diagonal

As Tabelas 4 e 5 apresentam respectivamente o número médio de iterações por bloco-iteração dos algoritmos de solução dos sistemas de equações lineares empregados na solução da equação da pressão, no pós-processamento das velocidades e na solução da equação da saturação, identificados respectivamente por PCG, Jacobi e GMRES. Nestas simulações, a tolerância dos métodos iterativos foi fixada em 10^{-4} , o número de iterações por bloco foi fixada em 3, e o passo de tempo foi $t = 0.004$. Adotou-se um número de 10

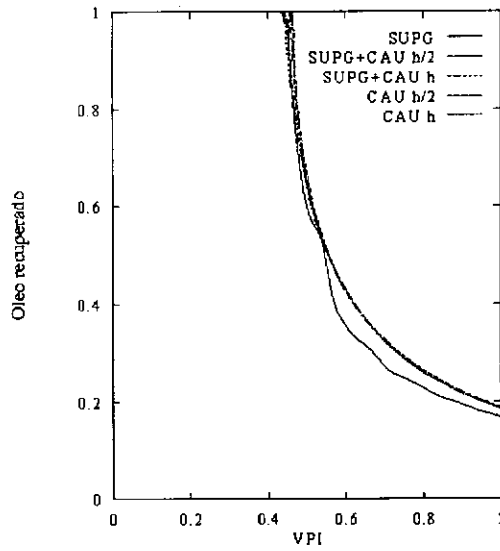


Figura 10: Recuperação de óleo para o problema de cinco poços na configuração paralela

vetores de Krylov para o algoritmo GMRES. Com a formulação SUPG as iterações por bloco convergem em 2 iterações, para ambas orientações de malha.

Tabela 4: Resultados computacionais para o problema de cinco poços na malha diagonal.

Método	SUPG	SUPG+CAU $h^e/2$	SUPG+CAU h^e	CAU $h^e/2$	CAU h^e
PCG	39.74	46.30	40.55	39.93	39.72
Jacobi	4.47	3.10	3.61	4.06	4.33
GMRES	8.80	6.52	6.66	4.25	4.23

Pode-se notar nas Tabelas 4 e 5 que os custos computacionais são dominados pela solução da equação da pressão. O número de iterações GMRES é ligeiramente superior na malha diagonal para as soluções que empregam a formulação SUPG, ainda que em todos os casos o número de iterações GMRES possa ser considerado muito pequeno. Deve-se destacar ainda o baixo número de iterações de Jacobi em todas as simulações.

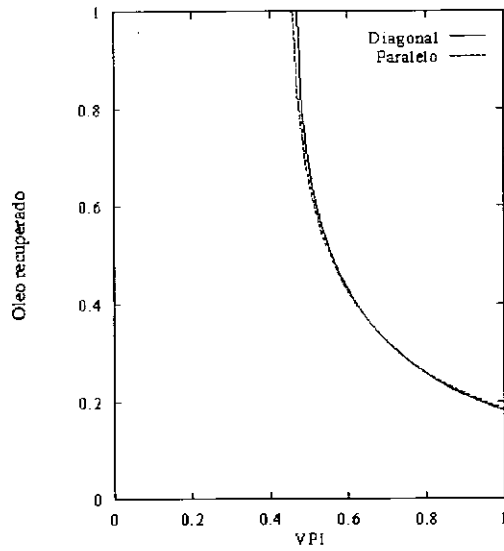


Figura 11: Recuperação de óleo para o problema de cinco poços em diferentes cenários

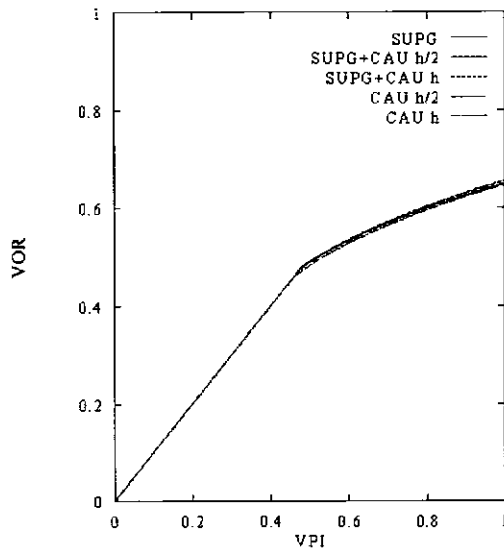


Figura 12: Volume de óleo recuperado para o problema de cinco poços na configuração diagonal.

Tabela 5: Resultados computacionais para o problema de cinco poços na malha paralela.

Método	SUPG	SUPG+CAU $h^e/2$	SUPG+CAU h^e	CAU $h^e/2$	CAU h^e
PCG	39.55	39.87	39.99	39.71	39.64
Jacobi	5.03	3.64	4.17	3.81	4.57
GMRES	5.59	5.22	5.29	4.28	4.17

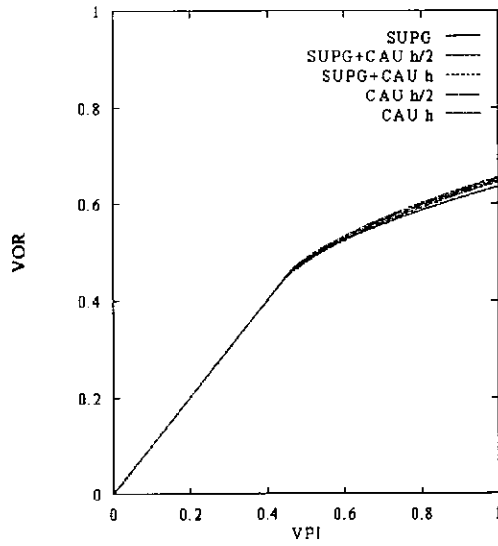


Figura 13: Volume de óleo recuperado para o problema de cinco poços na configuração paralela.

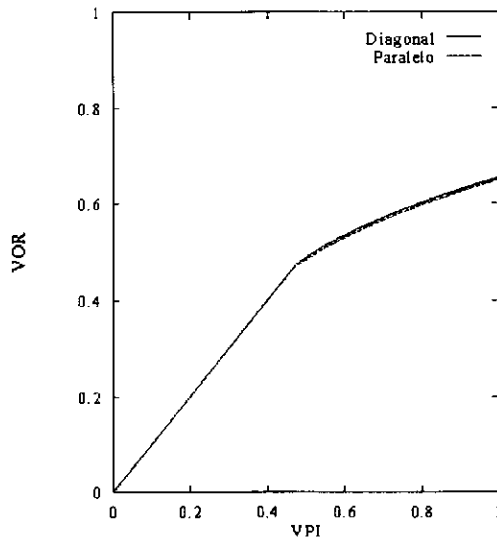


Figura 14: Volume de óleo recuperado para o problema de cinco poços em diferentes cenários.

Capítulo 4

Implementação do Refinamento Adaptativo de Malha

A simulação numérica de escoamentos de fluidos em meios porosos é um problema que apresenta alto grau de dificuldades no campo computacional em função da natureza do sistema de equações diferenciais governante. Entre estas dificuldades está a não linearidade do sistema de equações e a natureza convectiva-dominante da equação que descreve o comportamento do fluido injetado. Esta equação apresenta ainda regiões de altos gradientes que migram de uma região para outra à medida que a solução avança no tempo. Estas dificuldades tem motivado a busca de algoritmos numéricos que possam diminuir o esforço computacional envolvido e melhorar o desempenho da simulação numérica.

Uma proposta para resolver este problema de forma eficiente, sem ter que refinar demasiadamente todo o domínio, é a utilização de métodos adaptativos, que permitem, com o auxílio de um indicador de erro, que o domínio seja refinado apenas nas regiões de interesse. O objetivo é obter uma malha na qual o erro esteja igualmente distribuído sobre toda a malha. Métodos adaptativos tem sido utilizado por vários pesquisadores para uma vasta classe de problemas [2][41][46][49][42]. Para problemas onde as regiões a serem refinadas são pequenas em relação ao domínio, como é o caso em questão, a utilização de malhas adaptativas é determinante na obtenção de uma solução que represente o comportamento físico do problema, a um custo computacional aceitável.

Todo processo adaptativo é constituído de três etapas fundamentais:

1. Solução do problema em uma malha inicial.

2. Utilização de um medidor/indicador de erro para identificar as regiões que devem ser refinadas ou desrefinadas;
3. Uma estratégia para refinar/desrefinar a malha nas regiões identificadas na etapa anterior.

4.1 Estimador de erro

A peça fundamental no processo adaptativo é o indicador de erro. É ele que fornece as informações sobre a distribuição de erro no domínio, indicando as regiões que devem ser refinadas/desrefinadas.

Os estimadores de erro são classificados em duas categorias, os estimadores *a priori*, onde o erro é estimado antes que a solução seja conhecida, e os estimadores *a posteriori*, onde o erro é estimado após o cálculo da solução. Os estimadores *a priori*, não são computáveis, uma vez que a solução exata não é conhecida, sendo, no entanto, ferramentas úteis para estudos teóricos [16].

Os estimadores *a posteriori* baseiam-se na solução de elementos finitos calculada, portanto podem ser avaliados durante a análise. Isto os tornam mais úteis para a utilização em estratégias adaptativas e são largamente utilizados na prática. Veja por exemplo [5][40][36].

Os estimadores de erro *a posteriori* baseiam-se na avaliação do resíduo da equação governante por alguma norma, ou em técnicas de pós-processamento da solução, onde o erro é estimado pela diferença entre a solução pós-processada e a solução de elementos finitos [54].

Para problemas transientes onde o medidor de erro deve ser executado várias vezes durante a análise, uma característica a ser exigida é que ele seja rápido [31]. Um estimador de erro *a posteriori* baseado no pós-processamento da solução e que preenche este requisito, foi proposto por ZIENKIEWICZ e ZHU [56]. Este estimador é simples e muito eficiente para problemas práticos de engenharia [36]. Este estimador já foi utilizado com sucesso em problemas transientes [34], [42] e foi também utilizado neste trabalho.

Sendo ∇u^h o gradiente da solução aproximada pelo método dos elementos finitos e $\tilde{\nabla} u^h$ o gradiente pós-processado da solução aproximada através de uma média nodal [9], o erro é definido por,

$$\mathbf{e} = \left(\tilde{\nabla} u^h - \nabla u^h \right) \quad (4.104)$$

e a norma L_2 do erro pode então ser calculada como,

$$\|\mathbf{e}\| = \left(\int_{\Omega} \mathbf{e}^T \mathbf{e} d\Omega \right)^{\frac{1}{2}} \quad (4.105)$$

Em nível de elemento o erro pode ser calculado como segue:

$$\|\mathbf{e}_i^h\| = \left(\int_{\Omega_e} \mathbf{e}^T \mathbf{e} d\Omega \right)^{\frac{1}{2}} \quad (4.106)$$

Para uma malha com N_{el} elementos o erro médio é definido como,

$$E_m = \sqrt{\frac{\|\mathbf{E}\|^2}{N_{el}}} \quad (4.107)$$

onde $\|\mathbf{E}\|^2$ é dado por,

$$\|\mathbf{E}\|^2 = \sum_{e=1}^{N_{el}} \|\mathbf{e}_e^h\|^2 \quad (4.108)$$

4.2 Estratégia adaptativa

A decisão sobre quais elementos devem ser refinados ou desrefinados cabe à estratégia adaptativa, com o auxílio do indicador de erro. Isto permite que o programa decida automaticamente as regiões do domínio que devem ser refinadas ou desrefinadas.

Toda estratégia adaptativa visa alcançar uma malha ótima. Isto é, obter uma malha com o menor erro e o menor número possível de graus de liberdade. Para atingir este objetivo muitas estratégias adaptativas procuram distribuir o erro igualmente entre todos os elementos [14]. Esta idéia conduz ao seguinte algoritmo:

1. Computar a solução em uma malha inicial.
2. Para todo elemento e_i calcular o erro e determinar o erro máximo e_{\max} ,

$$e_{\max} = \max_{1 \leq e \leq Nel} \|e\|_{\Omega^e} \quad (4.109)$$

3. Refinar todos os elementos para os quais o indicador de erro satisfaz,

$$\|e_i\| \geq \Theta_1 e_{\max} \quad (4.110)$$

4. Desrefinar todos os elementos para os quais o indicador de erro satisfaz,

$$\|e_i\| \leq \Theta_2 e_{\max} \quad (4.111)$$

onde Θ_1 e Θ_2 são tolerâncias fixadas pelo usuário.

De acordo com DEVLOO [15], este algoritmo conduz a uma malha ótima após um determinado número de refinamentos.

Nem sempre a tarefa de determinar as regiões a serem refinadas deve ser confiada apenas ao indicador de erro. Para problemas não lineares que apresentam choque o critério de erro é difícil de ser alcançado resultando em uma malha onde certas regiões apresentam um número excessivo de refinamentos, elevando o custo computacional [49].

O que se deseja é obter uma malha, com um número mínimo de graus de liberdade, onde uma solução de boa qualidade possa ser obtida. Neste caso uma estratégia adaptativa com a imposição de um nível máximo de refinamento conduz a uma malha final onde nenhum elemento é refinado além de um valor pré-fixado para o número máximo de refinamentos permitido para cada elemento.

Neste trabalho adotou-se uma estratégia adaptativa combinando-se a imposição de um nível máximo de refinamento e o critério do erro médio E_m dado por (4.107). O algoritmo resultante é descrito a seguir.

4.2.1 Algoritmo adaptativo com imposição de nível máximo

Todo elemento na malha está associado a um nível. Os elementos da malha inicial possuem nível zero e quando um elemento é refinado os novos elementos gerados possuem níveis com uma unidade a mais que seus pais. A imposição sobre o mais alto nível de qualquer elemento conduz a este algoritmos.

1. Dados $0 \leq \Theta_1, \Theta_2 \leq 1$ e $NIVMAX$.
2. Resolver o problema para um intervalo de tempo fixo t_n , sobre uma malha inicial.
3. Calcule o erro médio E_m usando (4.107).
4. Calcular o erro máximo e o erro mínimo admissível

$$e_{\max} = \Theta_1 \cdot E_m \quad (4.112)$$

$$e_{\min} = \Theta_2 \cdot E_m \quad (4.113)$$

5. Refinar todos os elementos para os quais

$$\|e\| \geq e_{\max} \text{ e } nivel(e) < NIVMAX \quad (4.114)$$

6. Desrefinar todos os elementos para os quais

$$\|e\| \leq e_{\min} \quad (4.115)$$

7. Recalcular a solução

8. Retornar ao passo 3

A estratégia acima nos dá informações sobre a qualidade da aproximação de elementos finitos obtida, informando as regiões do domínio Ω que devem ser refinadas ou desrefinadas de acordo com erro calculado. Porém seu real valor está na sua combinação com uma estratégia eficiente para refinar e desrefinar a malha.

4.3 Estratégia para refinar e/ou desrefinar a malha

A última etapa no processo adaptativo é a escolha de uma estratégia eficiente para refinar e/ou desrefinar a malha. Três diferentes estratégias de refinamento são propostas na literatura:

1. Reposicionamento de nós (método-r)

Nesta técnica os nós da malha são redistribuídos para que uma discretização que melhor represente o problema seja alcançada [50][51].

2. Remalhamento (método-m)

Com o auxílio de um medidor, de erro uma discretização, local ou global, mais conveniente para o domínio computacional é gerada usando-se um gerador de malhas baseado em métodos tipo Delaunay ou avanços frontais [2][1].

3. Enriquecimento da malha (método-h)

Neste método, graus de liberdade são adicionados através da divisão de elementos existentes, onde a malha necessita ser refinada, ou elimina-se graus de liberdade, através do agrupamento de elementos, onde a malha pode ser desrefinada [42][14][15] [32].

Outro tipo de enriquecimento de malha é alcançado através do aumento da ordem das funções de forma, conhecido como **método-p**. A combinação dos métodos **h** e **p**, conhecida como **método-hp** também é muito utilizada [11][41].

A escolha de uma estratégia para refinar e/ou desrefinar a malha depende do tipo de problema em questão. Para problemas transientes onde a malha deve ser adaptada várias vezes durante o processo de solução e as fases de refinamento e desrefinamento são igualmente importantes, deve-se optar por uma estratégia eficiente que seja rápida e com baixo custo em termos de memória segundo MAVRIPLIS [37]. Conforme LÖHNER [32], a melhor estratégia é o refinamento-h com apenas um nível de refinamento/desrefinamento por mudança de malha. O procedimento utilizado para refinar/desrefinar a malha será detalhado a seguir.

4.3.1 Refinamento

O procedimento utilizado para refinar a malha segue o esquema proposto por RIVARA [43], onde cada elemento marcado para ser refinado é dividido pelo ponto médio do maior lado. De acordo com RIVARA [44] este procedimento garante uma malha final regular, aninhada, sem a presença de triângulos degenerados, isto é, todos os ângulos internos de qualquer triângulo são superiores a um valor calculado baseado na malha inicial. A transição entre os triângulos grandes e pequenos é feita de forma suave, isto é o refinamento é progressivo.

Para evitar o surgimento de nós irregulares durante o processo de refinamento utilizou-se o seguinte critério: cada elemento seleccionado para refinamento é dividido pelo maior lado juntamente com seu vizinho por este lado. Quando o maior lado do elemento a ser dividido é uma fronteira, apenas o elemento seleccionado é dividido. O procedimento de refinamento pode ser descrito resumidamente como se segue.

1. O maior lado do elemento a ser refinado é uma fronteira. O elemento é simplesmente dividido em dois por este lado.

2. O elemento A tem como vizinho pelo maior lado (l_a), o elemento B que tem como vizinho pelo maior lado (l_b) o próprio elemento A , conforme mostra na Figura 15(a). Então os dois elementos são divididos simultâneamente.

3. O vizinho pelo maior lado do elemento a ser dividido tem como vizinho pelo maior lado uma fronteira. Neste caso aplica-se o ítem 1 ao vizinho.

4. O elemento A tem como vizinho pelo maior lado (l_a), o elemento B que tem como vizinho pelo maior lado outro elemento, conforme mostra na Figura 15(b). Então aplica-se o ítem 2 ao vizinho.

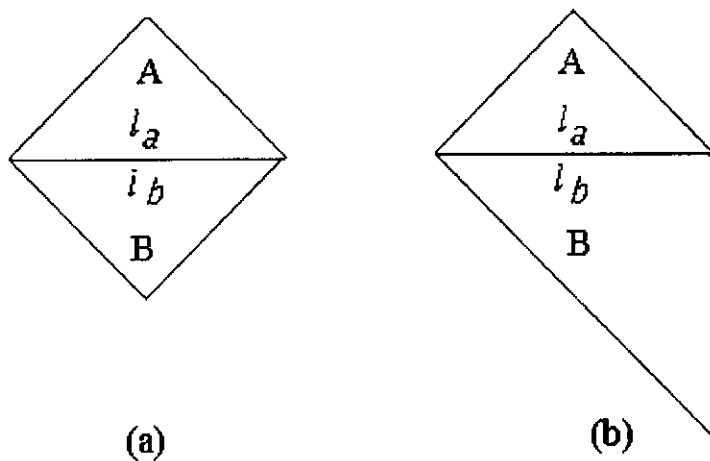


Figura 15: Casos de refinamento considerados

4.3.2 Desrefinamento

O procedimento para desrefinar um grupo de elementos é análogo ao refinamento. Um elemento marcado para desrefinamento só é eliminado se todo o grupo (elementos irmãos) é marcado para ser desrefinado. O processo de desrefinamento segue os seguintes passos:

1. O pai do grupo de elementos a ser desrefinado tem como vizinho pelo maior lado uma fronteira, conforme mostrado na Figura 16(a). Então o grupo de elementos é eliminado e retorna-se à lista de elementos a serem desrefinados.

2. O grupo de elementos a ser desrefinados tem como pai o elemento A cujo vizinho pelo maior lado é um elemento B . Verifica-se se os filhos do elemento B estão marcados para desrefinamento. Só quando todos os elementos do grupo estão marcados para desrefinamento os elementos são eliminados como exemplificado na Figura 16(b).

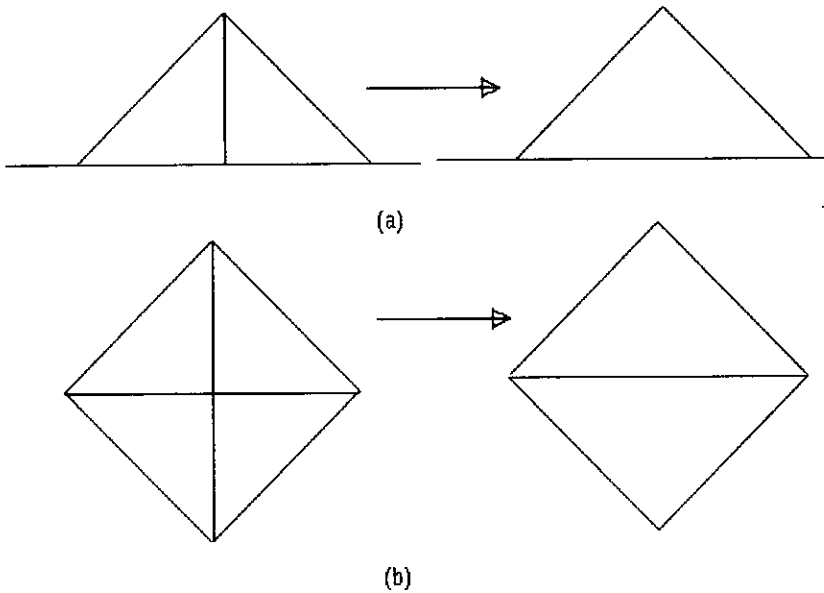


Figura 16: Casos de desrefinamento considerados. No caso (a) o maior lado do pai do grupo é uma fronteira. No caso (b), os dois elementos pais compartilham o maior lado.

4.3.3 Estruturas de dados

Para evitar a compactação da estrutura de dados sempre que a malha for atualizada, o que acarretaria em um consumo maior de tempo de CPU, utilizou-se o conceito de dados livres. Assim, os espaços deixados por nós e elementos eliminados durante a fase de desrefinamento são mantidos na estrutura para reutilização quando novos nós e elementos forem criados. Os ítems eliminados são chamados de dados livres e são mantidos em uma pilha para uso oportuno.

Durante a fase de refinamento, quando um novo nó ou elemento é criado, verifica-se na pilha a existência de dados livres. Se existirem dados livres na pilha os mesmos são desempilhados e reutilizados. Caso contrário um novo dado é criado. A estrutura de dados utilizada na implementação do conjunto de rotinas que executam o refinamento e desrefinamento da malha é:

$NELMX$, $NNOSMX$: representam, respectivamente, o número máximo de elementos e nós admitidos na malha.

$ICON(NELMX, 3)$: armazena as conectividades dos elementos.

$COORD(NPMX, 2)$: armazena as coordenadas dos nós.

$NELCON(NELMX, 3)$: armazena a vizinhança dos elementos. Sempre que um elemento é criado ou eliminado sua vizinhança é atualizada.

$NOCON(NNOSMX, 2)$: para cada novo nó criado, no ponto médio do maior lado do elemento a ser dividido, este vetor armazena os nós que definem este lado do elemento onde foi criado este novo nó, sendo que na primeira posição é armazenado o nó à sua esquerda e na segunda posição o nó à sua direita deste novo nó. Considere o caso onde o nó 3 é criado no ponto médio do lado formada pelos nós 1 e 2 conforme mostra a Figura 17. A configuração do arranjo $NOCON$, para este caso seria:

$$NOCON(3, 1) = 1$$

$$NOCON(3, 2) = 2$$

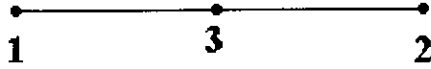


Figura 17: Divisão do maior lado do elemento pelo ponto médio.

Para atualizar a malha é necessário identificar os elementos que serão refinados e os elementos que serão eliminados. Estes dados são armazenados no arranjo *IREF* de dimensão *NELMX* como segue. Se

$IREF(IEL) = 1$: elemento a ser refinado

$IREF(IEL) = -1$: elemento a ser desrefinado

$IREF(IEL) = 0$: elemento não sofre alteração.

$NELRD(NELMX)$: armazena a nova ordem dos elementos. A função deste arranjo ficará clara adiante.

$IPAI(NELMX)$: o pai de cada elemento é armazenado neste arranjo. Sempre que um elemento é criado, o pai deste elemento é armazenado. Desta forma, se o elemento *nel* é dividido gerando os filhos *nel1* e *nel2*, temos que $IPAI(nel1) = IPAI(nel2) = nel$.

$IFILHO(NELMX)$: armazena o primeiro filho do elemento. Se um elemento *nel* tem como primeiro filho o elemento *iel*, então o outro filho será o elemento $iel + 1$.

Os vetores $IPAI(NELMX)$ e $IFILHO(NELMX)$ são calculados na fase de refinamento e suas informações utilizadas na fase de desrefinamento.

NOFREE(NNOSMX): arranjo que armazena a pilha de nós livres. Todo nó eliminado na fase de desrefinamento é armazenado em *NOFREE* para ser utilizado futuramente.

IELFREE(NELMX): neste arranjo é implementada a pilha de elementos livres. Apenas o primeiro filho é efetivamente armazenado, assume-se que o irmão do elemento também é livre.

NORD(NNOSMX): armazena a nova ordem dos nós. A função deste arranjo é a mesma do vetor *NELRD* e ficará clara oportunamente.

4.3.4 Interpolação e restrição da solução

Um dado importante é a frequência de atualização da malha. Aqui a malha é atualizada a cada np passos de tempo, onde geralmente tem-se que $5 \leq np \leq 10$. O parâmetro np é fornecido pelo usuário. Sempre que a malha é atualizada, um operador de restrição que projeta a solução da malha *fina* para a malha *grossa* na fase de desrefinamento, e um operador de interporlação que projeta a solução da malha *grossa* para a malha *fina* na fase de refinamento, devem ser definidos. Neste trabalho adotou-se a seguinte forma para estes operadores:

Na fase de desrefinamento, a solução nos nós eliminados é simplesmente desprezada. Na fase de refinamento, sempre que um novo nó é criado a solução neste nó é obtida por interpolação linear entre os nós que formam o lado pelo qual o elemento será dividido. Isto é,

$$s_3^f = \frac{1}{2} (s_1^g + s_2^g) \quad (4.116)$$

onde os índices f e g referem-se à malha *fina* e à malha *grossa* respectivamente.

4.4 Implementação da Estratégia Adaptativa em um código existente

O objetivo da presente tese é a implementação de uma estratégia adaptativa em um código de elementos finitos para simulação de fluidos bi-fásicos compressíveis em meios porosos heterogêneos. Este código foi concebido para trabalhar com malhas não-estruturadas fixas e emprega uma estrutura de dados compacta elemento por elemento.

Projetado para a arquitetura vetorial/paralela do supercomputador CRAYJ90, toda sua estrutura de dados foi idealizada para alcançar máxima eficiência neste tipo de arquitetura. Na implementação dos métodos iterativos as operações matriz-vetor e matriz-matriz foram otimizadas para alcançar máximo desempenho neste tipo de arquitetura. Para explorar as características das máquinas vetoriais um algoritmo para reordenar internamente os elementos em blocos disjuntos, onde nenhum elemento de um bloco possui um nó em comum com elementos de outros blocos, foi utilizado [26]. O tamanho destes blocos deve satisfazer as características do processador vetorial da máquina.

Visando eficiência em termos de tempo de CPU na implementação da estratégia adaptativa, optou-se por uma estrutura de dados que suportasse dados livres.

Alterar a estrutura de dados do código existente para acomodar dados livres seria uma tarefa árdua e poderia afetar severamente o desempenho do código. Optou-se então por projetar um código adaptativo independente, para ser incorporado ao código de elementos finitos existente, preservando sua estrutura de dados. Para isto permitiu-se que cada parte utilizasse sua própria estrutura de dados e desenvolveu-se um método para transformar os dados de uma estrutura para outra.

Os arranjos *NELRD* e *NORD* relacionam os elementos e nós nas duas estruturas. Assim a atribuição $nel = NELRD(iel)$ fornece o número do elemento *nel* na estrutura

adaptativa. Analogamente para a atribuição $no = \text{NORD}(ino)$. O núcleo do método utilizado para a transformação dos dados é apresentado no algoritmo à seguir:

(i) Determinar os elementos que deverão estar presentes na malha

$nel f = 0$

Do iel = 1, nelmx

If (i filho(iel) = 0) Then

If (iconect(iel, 1) \neq .0) Then

$nel f = nel f + 1$

$mat(nel f) = nel f$

Do ino = 1, nnoel

$ien(nel f, ino) = iconect(iel, ino)$

Enddo

$nelrd(nel f) = iel$

Endif

Endif

Enddo

(ii) Determinar os nós que deverão estar presentes na malha:

- – **determina a ocorrência de nós nas incidencias dos elementos**

Do iel = 1, nel f

Do no = 1, nnoel

$ino = ien(iel, no)$

$nodcon(ino) = nodcon(ino) + 1$

Enddo

Enddo

- - **enumera os nós de forma sequencial**

$nnosf = 0$

Do $ino = 1, nnosmx$

If ($nodcon(ino) \neq 0$) *Then*

$nnosf = nnosf + 1$

$xyz(nnosf, 1) = coord(ino, 1)$

$xyz(nnosf, 2) = coord(ino, 2)$

$inoa(ino) = nnosf$

$nord(nnosf) = ino$

Endif

Enddo

(iii) Renumeração dos nós dos elementos

Do $iel = 1, nel$

Do $no = 1, nnoel$

$ino = ien(iel, no)$

If ($inoa(ino) \neq ino$) *Then*

$ien(iel, no) = inoa(ino)$

Endif

Enddo

Endd

Capítulo 5

Exemplos Numéricos

Neste capítulo vários exemplos numéricos usados para validação do código desenvolvido serão apresentados. Estes exemplos envolvem tipos diferentes de reservatórios e condições de contorno. Todos os casos analisados são para fluidos imiscíveis bi-fásicos ligeiramente compressíveis usando o método dos elementos finitos adaptativo apresentado neste trabalho. Em todos os casos a malha inicial é uniforme com 16 divisões, o que corresponde a 512 elementos triangulares lineares e o mesmo indicador de erro é utilizado para a equação da pressão (2.25) e saturação (2.38).

Os elementos marcados para refinamento ou desrefinamento são aqueles determinados por ambas as equações na seguinte ordem:

1. aplica-se o indicador de erro na equação da pressão e determinam-se os elementos a serem refinados/desrefinados.
2. aplica-se o indicador de erro na equação da saturação e determinam-se os elementos a serem refinados/desrefinados, porém nesta fase o elemento só é marcado para desrefinamento se tiver sido marcado para desrefinamento pela equação da pressão.

5.1 Arranjo de cinco poços com razão de viscosidade unitária

Considera-se neste primeiro exemplo o arranjo de cinco poços, com tensor de permeabilidade absoluta $K = I$ e permeabilidades relativas $k_{rw} = s_w^2$ e $k_{ro} = (1 - s_w)^2$. A porosidade é $\phi = 0.2$, as massas específicas da água e do óleo são respectivamente, $\rho_w = 2$ e $\rho_o = 1$, em um sistema de unidades coerentes. O sistema é moderadamente

compressível, sendo a constante de compressibilidade $c_t = 10^{-6}$. Os efeitos da gravidade e pressão capilar são nulos para este problema e a razão de viscosidade é $\mu_o/\mu_w = 1.0$. Admite-se ainda que a saturação inicial $s_w = 0$ em todo o reservatório. O passo de tempo é fixado em $\Delta t = 0.004$, a tolerância dos métodos iterativos é 10^{-3} e a tolerância para iteração não linear é 0.5. Para este problema o número máximo de multicorrecções foi fixado em 5, sendo a malha atualizada a cada 5 passos de tempo com no máximo 8 níveis de refinamento, o que corresponde a uma malha com 256 divisões.

O poço injetor está localizado no canto inferior esquerdo do domínio e o produtor no canto superior direito. As Figuras 18 e 19 apresentam, respectivamente a frente de saturação em 0.25 VPI e 0.5 VPI. As malhas correspondentes são apresentadas nas Figuras 20 e 21. A malha obtida quando o medidor de erro é aplicado apenas à equação da pressão, em 0.5 VPI, é mostrada na Figura 22, e a malha obtida quando o medidor de erro é aplicado apenas à equação da saturação, também em 0.5 VPI, é mostrada na Figura 23.

Observa-se nas Figuras 20 e 21 que a região de refinamento está concentrada em torno dos poços e na frente de saturação. As mesmas Figuras mostram a evolução no tempo da frente de saturação.

5.1.1 Avaliação de desempenho

Para este teste considerou-se o arranjo de cinco poços com razão de viscosidade unitária, como descrito anteriormente, sendo que o número máximo de iterações foi fixado em 100 e o tempo máximo de análise foi fixado em 0.5 VPI. Comparou-se o tempo de execução com malha adaptativa e com malha uniforme, sendo que as malhas apresentavam aproximadamente o mesmo número de elementos. As malhas adaptativas foram atualizadas

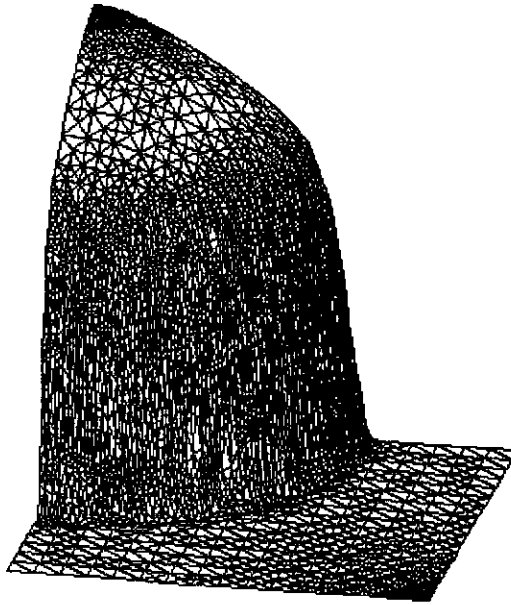


Figura 18: Frente de saturação para o arranjo de cinco poços com razão de viscosidade unitária, em 0.25 VPI.

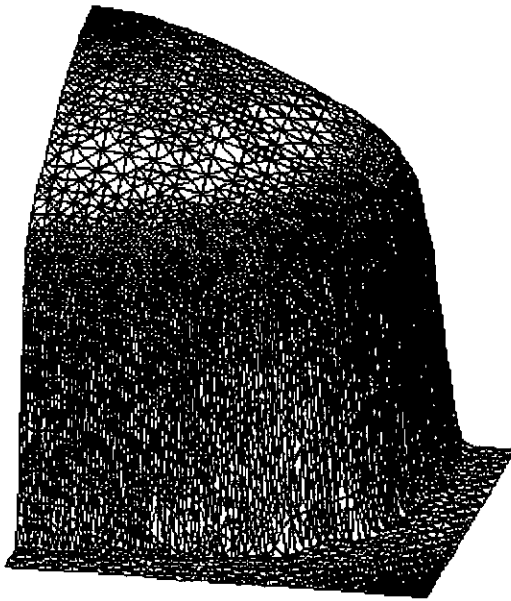


Figura 19: Frente de saturação para o arranjo de cinco poços com razão de viscosidade unitária, em 0.5 VPI

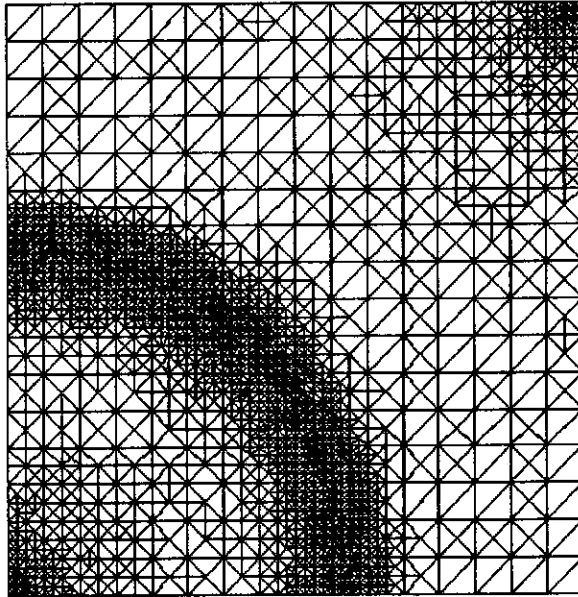


Figura 20: Malha com 2493 nós, 4852 elementos, correspondente à frente de saturação para o arranjo de cinco poços com razão de viscosidade unitária, em 0.25 VPI.

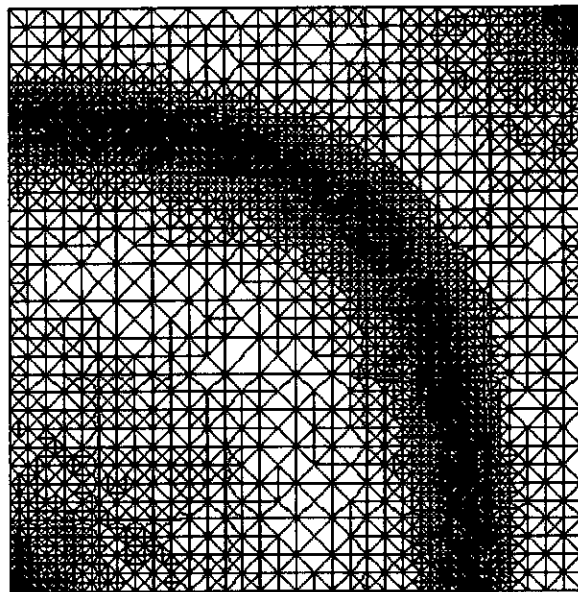


Figura 21: Malha com 5266 nós, 10341 elementos, correspondente à frente de saturação para o arranjo de cinco poços com razão de viscosidade unitária, em 0.5 VPI.

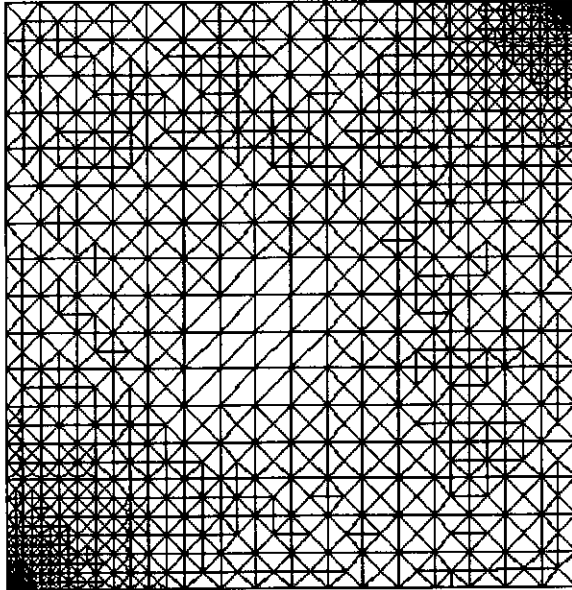


Figura 22: Malha com 1123 nós, 2123 elementos para o arranjo de cinco poços com razão de viscosidade unitária, em 0.5 VPI, com refinamento determinado pela equação da pressão.

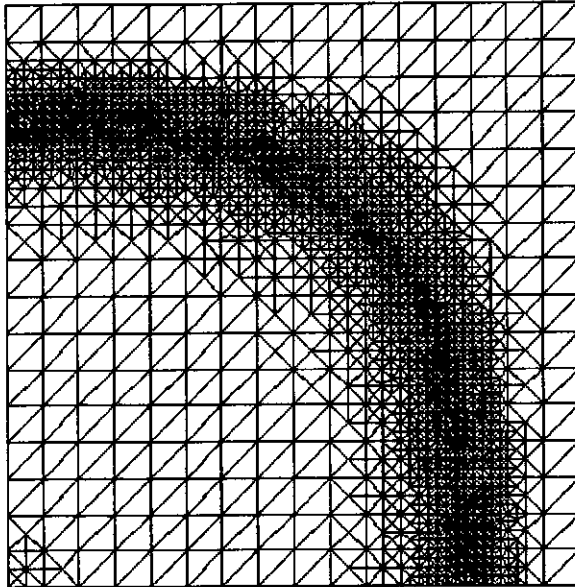


Figura 23: Malha com 3025 nós, 5951 elementos para o arranjo de cinco poços com razão de viscosidade unitária, em 0.5 VPI, com refinamento determinado pela equação da saturação.

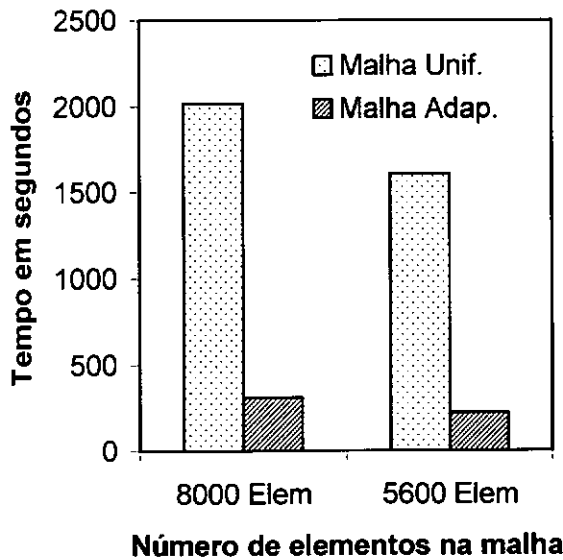


Figura 24: Comparação entre tempos de execução (malha uniforme \times malha adaptativa) para o arranjo de cinco poços com razão de viscosidade unitária, em 0.5 VPI.

a cada 5 passos. Considerou-se dois casos: um com aproximadamente 8000 elementos e o outro com aproximadamente 5600 elementos. A Figura 24 mostra os tempos de execução, em segundos, para cada caso. Como pode ser observado, a solução com malha adaptativa apresenta um tempo bem inferior à solução com malha uniforme. O número médio de multicorrecções executados para cada caso é apresentado na Tabela 6. Como pode ser observado o número médio de multicorrecções executados quando utiliza-se refinamento adaptativo é em média 4 vezes menor que aquele apresentado quando utiliza-se refinamento uniforme.

Tabela 6: Número médio de multicorrecções executadas quando utilizou-se malha uniforme e malha adaptativa para o arranjo de cinco poços, com viscosidade unitária em 0.5VPI.

Tipo de malha	Número médio de multicorrecções	
	8000 Elementos	5600 Elementos
Malha Uniforme	25	38
Malha Adaptativa	6.6	6.5

5.2 Arranjo de cinco poços com razão de viscosidade desfavorável

Considera-se neste exemplo o arranjo de cinco poços, com tensor de permeabilidade absoluta $\mathbf{K} = \mathbf{I}$ e permeabilidades relativas $k_{rw} = s_w^2$ e $k_{ro} = (1 - s_w)^2$. A porosidade é $\phi = 0.2$, as as massas específicas da água e do óleo são respectivamente, $\rho_w = 2$ e $\rho_o = 1$. O sistema é moderadamente compressível, sendo a constante de compressibilidade $c_t = 10^{-6}$, os efeitos da gravidade e pressão capilar são nulos para este problema e a razão de viscosidade é $\mu_o/\mu_w = 4.0$. Assume-se ainda que a saturação inicial $s_w = 0$ em todo o reservatório. O passo de tempo é fixado em $\Delta t = 0.004$, a tolerância dos métodos iterativos é 10^{-3} e a tolerância para iteração não linear é 0.5. Para este problema o número máximo de multiorreções foi fixado em 5, sendo a malha atualizada a cada 8 passos de tempo com no máximo 6 níveis de refinamento, o que corresponde a uma malha com 128 divisões.

As malhas correspondentes às frentes de saturação em 1 VPI, para os dois cenários, diagonal e paralelo, são apresentadas nas Figuras 25 e 26. Os resultados para a recuperação de óleo normalizada pelo volume total injetado para os dois cenários são apresentados na Figura 27, e na Figura 28 as curvas para o volume de óleo recuperado para as duas orientações de malha. Como pode ser observado os efeitos de orientação de malha são imperceptíveis.

A Tabela 7 apresenta o número médio de multiorreções por passo de tempo executado pelo algoritmo bloco-iterativo preditor-multicorretor e o número médio de iterações em cada multiorreção executado pelo método dos gradientes conjugados com preconditionador de Gauss-Seidel, empregados para resolver a equação da pressão identificado como PCG. Na mesma tabela apresenta-se também o número médio de iterações do algo-

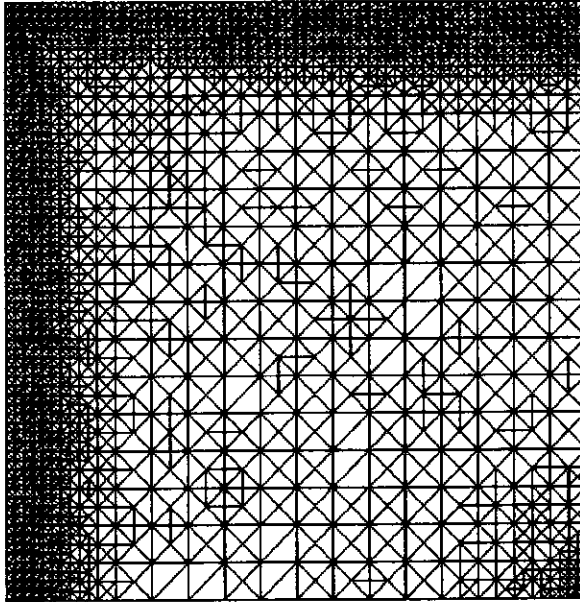


Figura 25: Malha com 3356 nós e 6495 elementos, correspondente à frente de saturação em 1 VPI, para o arranjo de cinco poços com razão de viscosidade 4, na orientação diagonal.

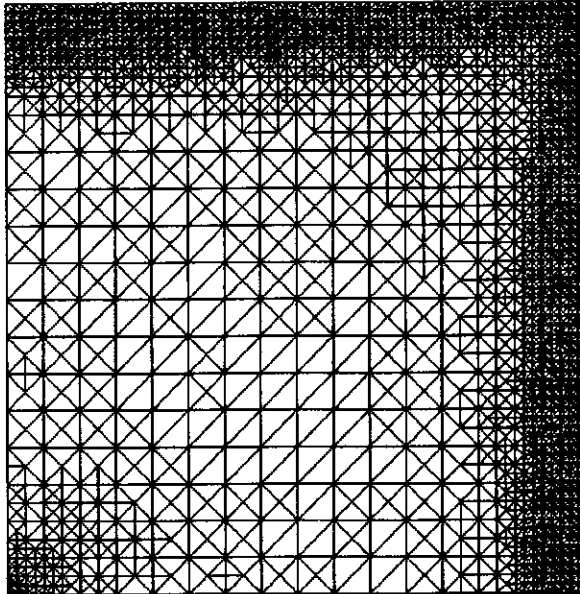


Figura 26: Malha com 3255 nós e 6284 elementos, correspondente à frente de saturação em 1 VPI, para o arranjo de cinco poços com razão de viscosidade 4, na orientação paralela.

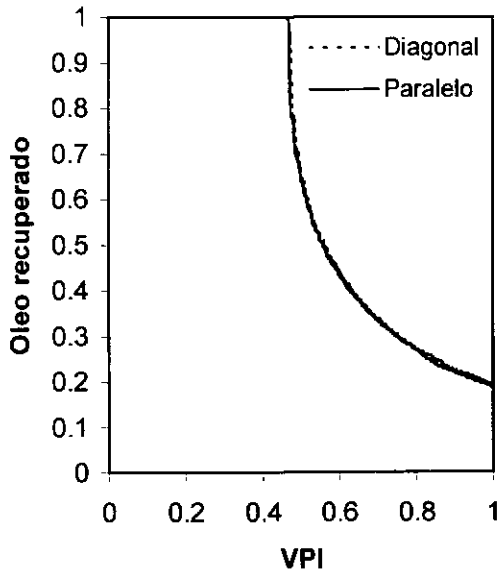


Figura 27: Recuperação para o arranjo de cinco poços com razão de viscosidade 4, para os cenários diagonal e paralelo.

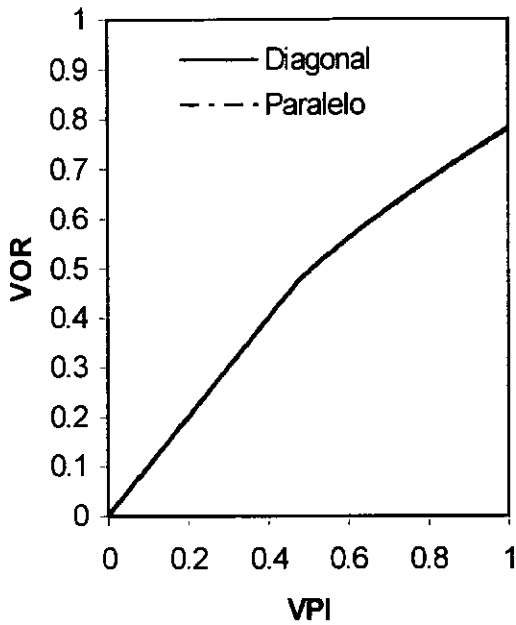


Figura 28: Volume total de óleo recuperado para o arranjo de cinco poços, com razão de viscosidade 4, para os cenários diagonal e paralelo.

ritmo GMRES com condicionador Gauss-Seidel para resolver a equação da saturação e o número médio de iterações de Jacobi livre de matrizes empregado para o cálculo das velocidades, identificado como JCB. Um baixo número de multicorreções foram necessárias para resolver o problema como pode ser visto na Tabela 7. Observa-se também que os custos computacionais são dominados pela equação da pressão.

Os tempos de execução para solução com malha adaptativa e uniforme para uma análise igual até 0.5 VPI, quando o número máximo de iterações foi fixado em 100, são dados na Tabela 8. Novamente observa-se que a solução adaptativa é mais rápida que a solução com malha uniforme. O tempo de CPU gasto pelas rotinas que implementam a lógica de refinamento/desrefinamento, incluindo a rotina que transforma os dados de uma estrutura para a outra é também muito pequeno. Para este exemplo foi de 0.3% do tempo total de CPU e o tempo gasto pelo indicador de erro foi de 0.2% do tempo total de CPU. Este cálculo foi avaliado para 1 VPI.

Tabela 7: Número médio de iterações por multicorreção para o arranjo de cinco poços com razão de viscosidade 4.

Método	Número Médio de Iterações
Multicorreções	4.7
PCG	62.0
GMRES	7.0
JCB	2.6

Tabela 8: Tempo de execução usando malha uniforme e malha adaptativa para o arranjo de cinco poços, com razão de viscosidade 4 em 0.5 VPI.

Tipo de malha	Tempo (minutos)	Número de Elementos
Malha Uniforme	4.2	5000
Malha Adaptativa	2.7	5261

5.3 Injeção de água pela lateral do domínio

Considera-se agora o modelo de reservatório apresentado em [23] onde o domínio computacional é o quadrado unitário, sendo injeção e produção tomadas respectivamente como $q = 1.0$ na fronteira esquerda e $q = -1.0$ na fronteira direita. A função de difusão é $D(s_w) = 0.04s_w(1 - s_w)$. O tensor de permeabilidade absoluta é $\mathbf{K} = \mathbf{I}$ e as permeabilidades relativas $k_{rw} = s_w^2$ e $k_{ro} = (1 - s_w)^2$. A porosidade é $\phi = 0.2$, as as massas específicas da água e do óleo são respectivamente, $\rho_w = 2.0$ e $\rho_o = 1.0$. O sistema é moderadamente compressível, sendo a constante de compressibilidade $c_t = 10^{-6}$ e a razão de viscosidade é $\mu_o/\mu_w = 1.0$. Assume-se ainda que a saturação inicial é $s_w = 0$ em todo reservatório. Os resultados, para este modelo, com e sem os efeitos da gravidade serão apresentados a seguir.

5.3.1 Injeção de água pela lateral do domínio sem gravidade

Para este exemplo os efeitos da gravidade não são considerados. O passo de tempo é fixado em $\Delta t = 0.001$, a tolerância dos métodos iterativos é 10^{-3} e a tolerância para iteração não linear em 0.5. O número máximo de multicorreções é fixado em 10, sendo a malha atualizada a cada 8 passos de tempo e o nível máximo de refinamento fixado em 6, correspondendo a uma malha com 128 divisões.

A frente de saturação para 100 passos de tempo é apresentada na Figura 29 e a malha correspondente na Figura 30.

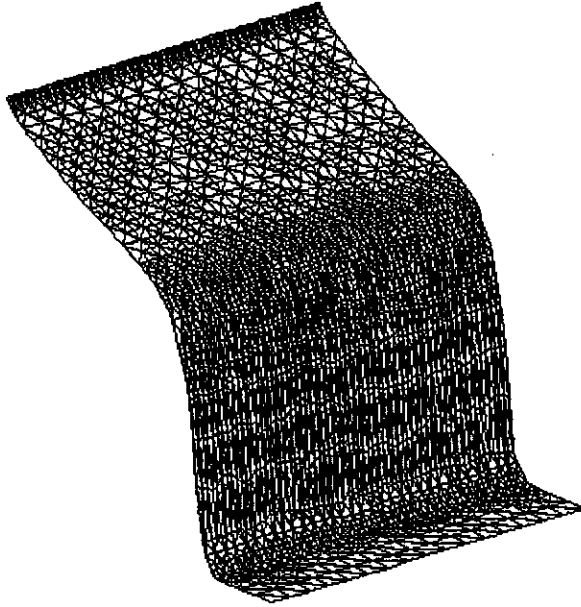


Figura 29: Frente de saturação para 100 passos de tempo sem os efeitos da gravidade.

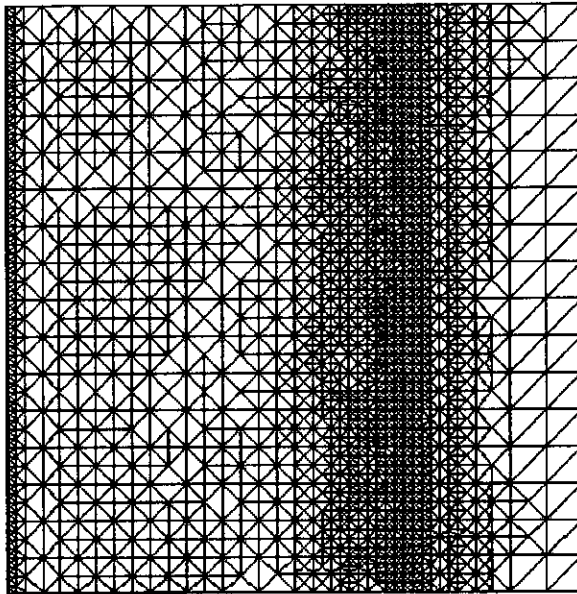


Figura 30: Malha com 4536 elementos, 2376 nós correspondente à frente de saturação.

Tabela 9: Número médio de iterações por multicorreção para o problema de injeção de água pela lateral do domínio sem gravidade

Método	Número Médio de Iterações
Multicorreções	7.4
PCG	42.7
GMRES	4.0
JCB	3.0

Novamente observa-se que o refinamento acompanha a frente de saturação como pode ser visto na Figura 30. A linha de refinamento paralela ao eixo vertical era esperada, uma vez que para este exemplo o efeito da gravidade não foi considerado. Nenhuma oscilação na frente de saturação é observada.

Para comparar o tempo de execução com malha adaptativa e uniforme em uma análise até 0.5 VPI, o número máximo de iterações foi fixado em 100. Os resultados são mostrados na Tabela 10. Novamente verifica-se que a solução adaptativa é aproximadamente 4 vezes mais rápida.

Tabela 10: Tempo de execução usando malha uniforme e malha adaptativa para injeção de água pela lateral do domínio sem gravidade em 0.5VPI.

Tipo de malha	Número de Elementos	Tempo (minutos)	Multicorreções
Malha Uniforme	2048	25.4	93
Malha Adaptativa	4542	5.7	23.6

5.3.2 Injeção de água pela lateral do domínio com gravidade

Para este exemplo onde os efeitos da gravidade são considerados, o vetor aceleração da gravidade é $\mathbf{g} = (0, -1)$, o passo de tempo é fixado em $\Delta t = 0.0001$, a tolerância dos métodos iterativos é 10^{-3} e a tolerância para iteração não linear em 0.5. O número máximo de multicorreções é fixado em 10, sendo a malha atualizada a cada 8 passos de

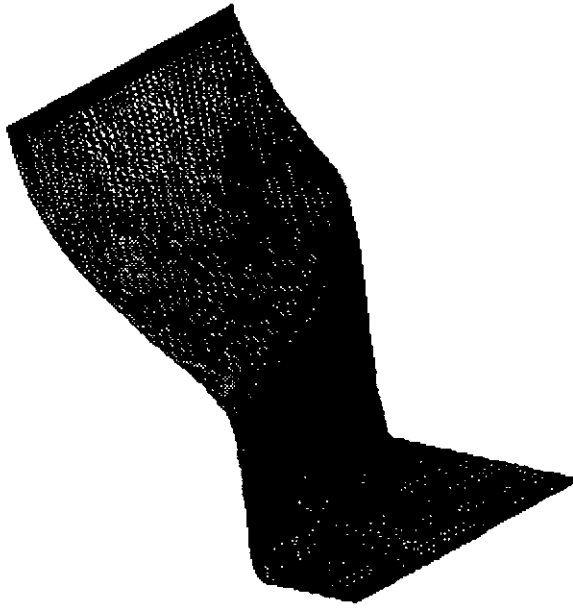


Figura 31: Frente de saturação considerando os efeitos da gravidade para 400 passos de tempo

tempo e o nível máximo de refinamento fixado em 10. Isto corresponde a uma malha com 512 divisões.

A frente de saturação após 400 passos de tempo é apresentada na Figura 31 e na Figura 32 a malha correspondente.

Tabela 11: Número médio de iterações por multicorreção para o problema de injeção de água pela lateral do domínio com gravidade

Método	Número Médio de Iterações
Multicorreções	9.8
PCG	32.3
GMRES	4.2
JCB	7.0

Para este caso pode ser observado na Figura 31 que o efeito da gravidade provoca um acúmulo de água na parte inferior da malha. Efeito semelhante foi verificado por Hansen e Spedal [23] para o caso incompressível. Comparando-se os resultados da Tabela 11 para o caso com gravidade com aqueles da Tabela 9 (sem gravidade), nota-se que

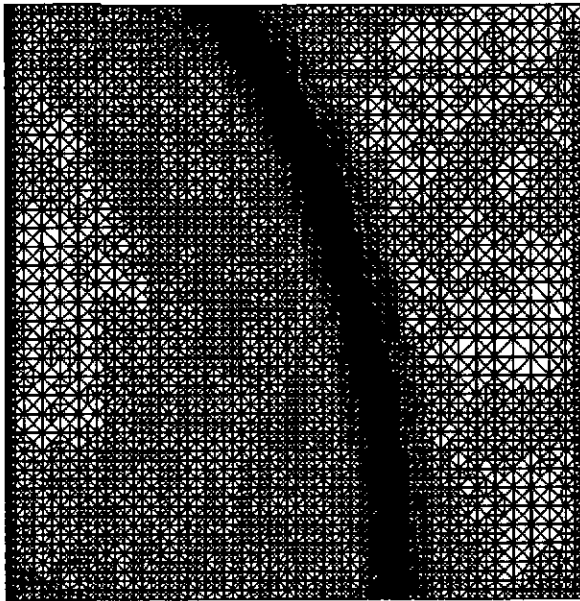


Figura 32: Malha com 19422 elementos, 9942 nós correspondente à frente de saturação.

para o caso com gravidade são necessárias mais multicorrekções e um maior número de iterações de Jacobi. No entanto o custo computacional gasto na solução da equação da pressão foi menor.

5.4 Tensor de Permeabilidade Completo

Neste exemplo utilizou-se um tensor permeabilidade completo dado por,

$$K = \begin{bmatrix} 5.0 & 4.0 \\ 4.0 & 5.0 \end{bmatrix} \quad (5.117)$$

como em Edwards [18] este tensor é obtido a partir de um tensor diagonal anisotrópico aplicando-se sobre este um giro de 45° . O domínio é o quadrado unitário com razão de injeção $q = 1.0$ na fronteira esquerda e, na fronteira direita, condições de Dirichlet nulas são especificadas. As permeabilidades relativas são $k_{rw} = s_w^2$ e $k_{ro} = (1 - s_w)^2$. A porosidade $\phi = 1.0$, as massas específicas da água e do óleo são respectivamente, $\rho_w = 2$ e $\rho_o = 1$. O sistema é moderadamente compressível, sendo a constante de compressibilidade $c_t = 10^{-6}$, os efeitos da gravidade e pressão capilar são nulos para este problema e a razão de viscosidade é $\mu_o/\mu_w = 1.0$. Assume-se ainda que a saturação inicial $s_w = 0$ em todo o reservatório. O passo de tempo é fixado em $\Delta t = 0.002$, a tolerância dos métodos iterativos é 10^{-3} e a tolerância para iteração não linear é 0.5. Para este problema o número máximo de multicorrekções foi fixado em 5, sendo a malha atualizada a cada 8 passos de tempo com nível máximo de refinamento igual a 6.

Os contornos da saturação da água para 0.5VPI são mostrados na Figura 33. A Tabela 12 apresenta o número médio de multicorrekções por passo de tempo executado pelo algoritmo bloco-iterativo preditor-multicorretor e o número médio de iterações em cada multicorrekção executado pelo método dos gradientes conjugados com precondu-

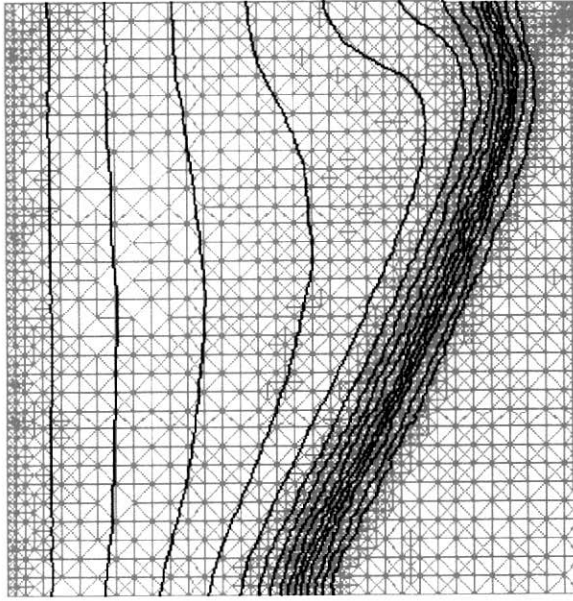


Figura 33: Contornos da frente de saturação em 0.5VPI, com tensor de permeabilidade absoluta completo

nador de Gauss-Seidel, empregados para resolver a equação da pressão identificado como PCG. Na mesma tabela apresenta-se também o número médio de iterações do algoritmo GMRES com preconditionador Gauss-Seidel para resolver a equação da saturação e o número médio de iterações de Jacobi livre de matrizes empregado para o cálculo das velocidades identificado como JCB.

Tabela 12: Número médio de iterações por multicorreção para o problema com tensor de permeabilidade completo

Método	Número Médio de Iterações
Multicorreções	4.9
PCG	94
GMRES	4.7
JCB	2.5

Como pode ser observado na Figura 33, a frente de saturação avança com maior rapidez na direção que apresenta maior permeabilidade. Este mesmo comportamento foi relatado em Edwards [18]. Mais uma vez o maior esforço computacional foi con-

centrado na resolução da equação da pressão, já a solução da equação da saturação e o pós-processamento da velocidade foram rápidos, como pode ser observado pelo baixo número de iterações executadas pelo algoritmo GMRES e JCB.

5.5 Solução do Arranjo de Cinco Poços em Meio Heterogêneo

Em um reservatório de petróleo a permeabilidade do meio raramente é uniforme sendo normalmente usado um valor médio. Neste exemplo estuda-se o arranjo de cinco poços, com tensor de permeabilidade absoluta K heterogêneo, gerado a partir de uma distribuição lognormal com média e desvio padrão iguais a 1.0. As permeabilidades relativas são $k_{rw} = s_w^2$ e $k_{ro} = (1 - s_w)^2$. A porosidade $\phi = 0.2$, as massas específicas da água e do óleo são respectivamente, $\rho_w = 2$ e $\rho_o = 1$. O sistema é moderadamente compressível, sendo a constante de compressibilidade $c_t = 10^{-6}$, os efeitos da gravidade e pressão capilar são nulos para este problema e a razão de viscosidade é $\mu_o/\mu_w = 1.0$. Assume-se ainda que a saturação inicial $s_w = 0$ em todo o reservatório. O passo de tempo é fixado em $\Delta t = 0.004$, a tolerância dos métodos iterativos é 10^{-3} e a tolerância para iteração não linear é 0.5. Para este problema o número máximo de multicorrecções foi fixado em 5, sendo a malha atualizada a cada 8 passos de tempo e o número máximo de refinamentos foi fixado em 6. O poço injetor está localizado no canto inferior esquerdo do domínio e o produtor no canto superior direito.

Dois casos são analisados:

- a. Distribuição de permeabilidades variável: neste caso as permeabilidades dos elementos filhos são geradas através da mesma distribuição lognormal, tendo como média e desvio padrão as permeabilidades do elemento pai.

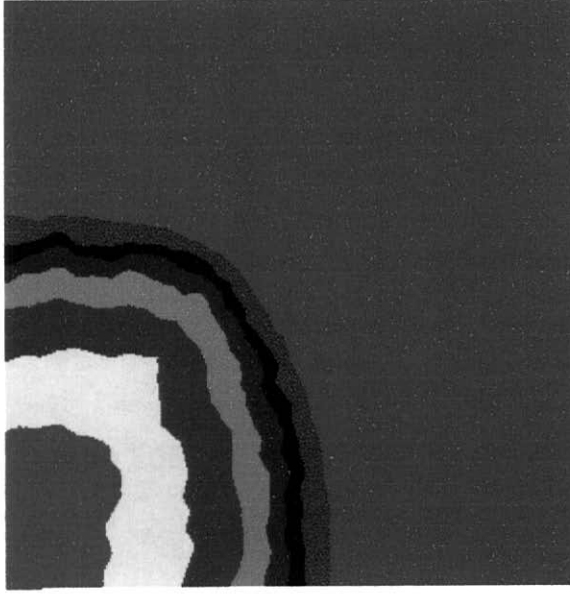


Figura 34: Frente de saturação em $t = 0.2$ para o arranjo de cinco poços com distribuição de permeabilidades heterogênea .

As frentes de saturação e respectivas malhas em, $t = 0.2$, $t = 0.4$ e $t = 0.6$ são apresentadas nas Figuras 34,35, 36, 37, 38 e 39. Na Tabela 13, está o número médio de multicorreções por passo de tempo executado pelo algoritmo bloco-iterativo preditor-multicorretor e o número médio de iterações em cada multicorreção executado pelo método dos gradientes conjugados com preconditionador de Gauss-Seidel, empregados para resolver a equação da pressão identificado como PCG. Na mesma tabela apresenta-se também o número médio de iterações do algoritmo GMRES com preconditionador Gauss-Seidel para resolver a equação da saturação e o número médio de iterações de Jacobi livre de matrizes empregado para o cálculo das velocidades, identificado como nos casos anteriores por JCB. As Figuras 40, 41, 42 e 43 apresentam, respectivamente, a distribuição inicial de permeabilidades, e as distribuições de permeabilidades em 0.25 VPI, 0.5 VPI e 0.75 VPI.

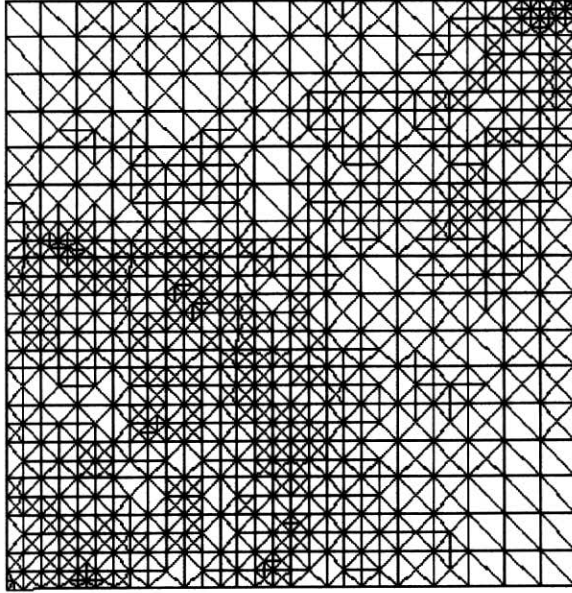


Figura 35: Malha com 1084 nós e 2065 elementos, para o arranjo de cinco poços com distribuição de permeabilidade heterogênea variável em 0.25 VPI.

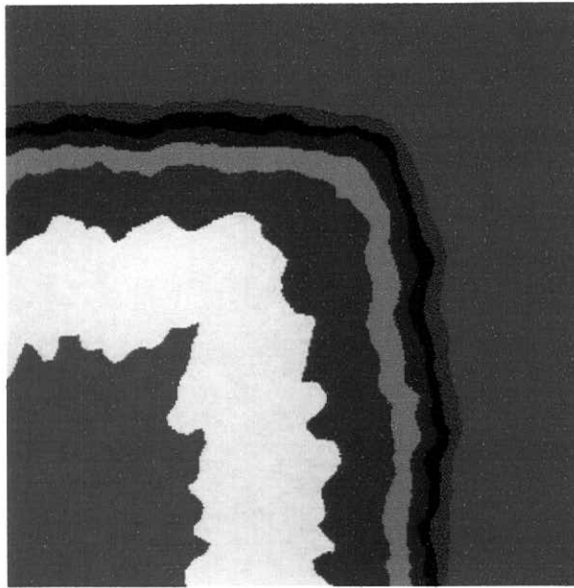


Figura 36: Frente de saturação em $t = 0.4$ para o arranjo de cinco poços com distribuição de permeabilidades heterogênea .

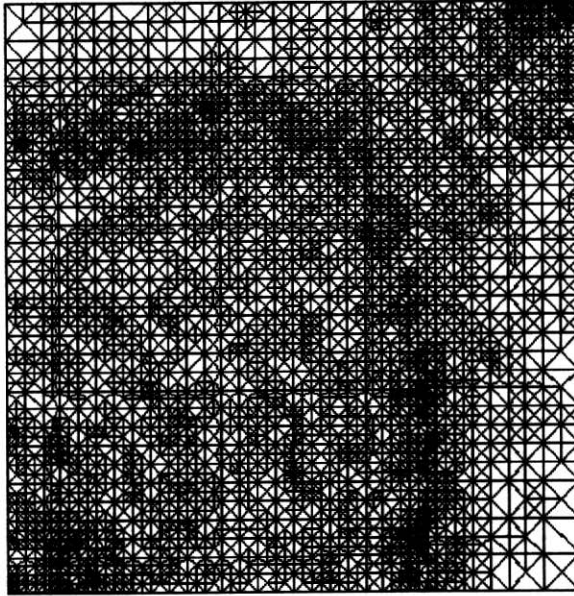


Figura 37: Malha com 4150 nós e 8115 elementos, para o arranjo de cinco poços com distribuição de permeabilidade heterogênea variável em 0.5 VPI.

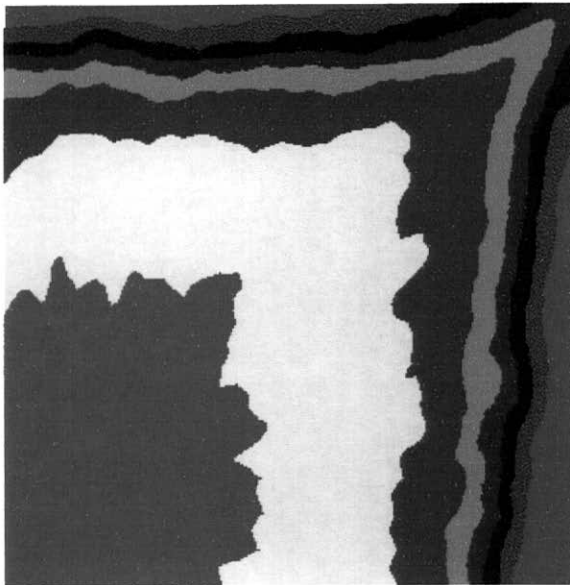


Figura 38: Frente de saturação em $t = 0.6$ para o arranjo de cinco poços com distribuição de permeabilidades heterogênea.

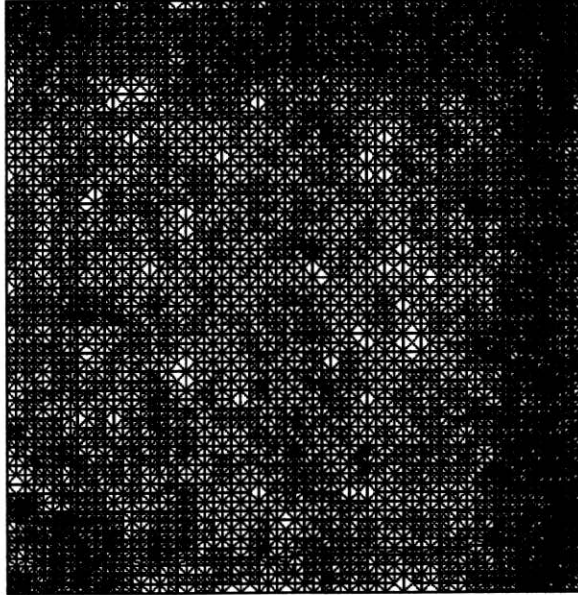


Figura 39: Malha com 8192 nós e 16085 elementos, para o arranjo de cinco poços com distribuição de permeabilidade heterogênea variável em 0.75 de VPI.



Figura 40: Distribuição inicial de permeabilidades para o arranjo de cinco poços.

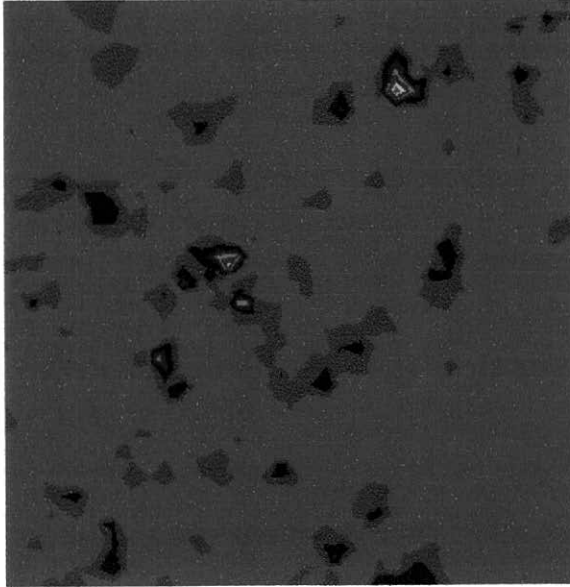


Figura 41: Distribuição de permeabilidades para o arranjo de cinco poços com distribuição variável em 0.25 VPI.

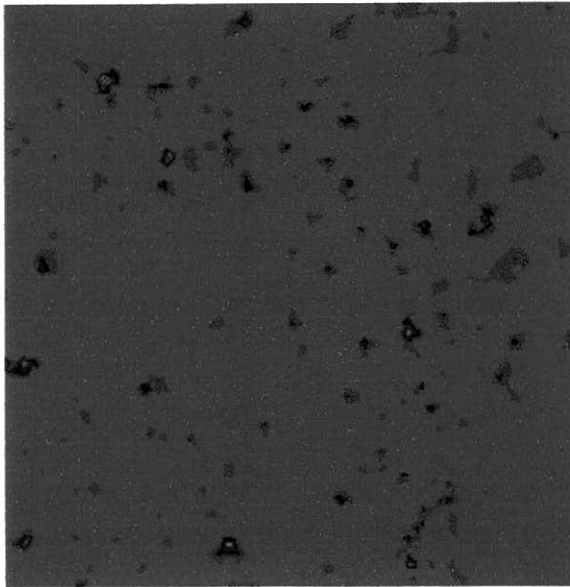


Figura 42: Distribuição de permeabilidades para o arranjo de cinco poços com distribuição variável em 0.5 VPI.

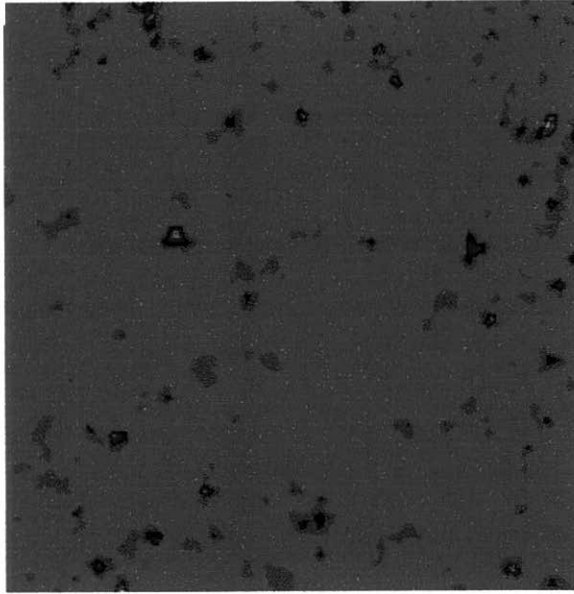


Figura 43: Distribuição de permeabilidades para o arranjo de cinco poços com distribuição variável em 0.75 VPI.

Tabela 13: Número médio de iterações por multicorreção para o arranjo de cinco poços com distribuição de permeabilidade heterogênea variável em 0.75 de VPI.

Método	Número Médio de Iterações
Multicorreções	4.4
PCG	72
GMRES	4.9
JCB	2.3

b. Distribuição de permeabilidade fixa (dada pela malha inicial, conforme Figura 40). Neste caso as permeabilidades dos elementos filhos são iguais à permeabilidade do elemento pai.

As frentes de saturação e respectivas malhas em, $t = 0.2$, $t = 0.4$ e $t = 0.6$ são apresentadas nas Figuras 44, 45, 46, 47, 48 e 49. Na Tabela 14, está o número médio de multicorreções por passo de tempo executado pelo algoritmo bloco-iterativo preditor-multicorretor e o número médio de iterações em cada multicorreção executado pelo método dos gradientes conjugados com preconditionador de Gauss-Seidel, empregados para resolver a equação da pressão identificado como PCG. Na mesma tabela apresenta-se também o número médio de iterações do algoritmo GMRES com preconditionador Gauss-Seidel para resolver a equação da saturação e o número médio de iterações de Jacobi livre de matrizes empregado para o cálculo das velocidades identificado como JCB.

Tabela 14: Número médio de iterações por multicorreção para o arranjo de cinco poços com distribuição de permeabilidade heterogênea fixa em 3/4 de VPI.

Método	Número Médio de Iterações
Multicorreções	4.8
PCG	74.6
GMRES	5.2
JCB	2.0

Para os dois casos de distribuição heterogênea de permeabilidades verifica-se que o custo computacional para ambos os casos é equivalente, sendo que o maior tempo foi

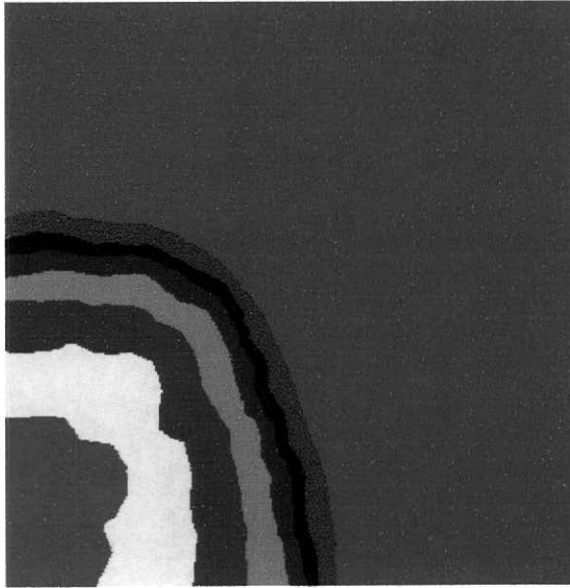


Figura 44: Frente de saturação em $t = 0.2$ para o arranjo de cinco poços com distribuição de permeabilidades heterogênea fixa.

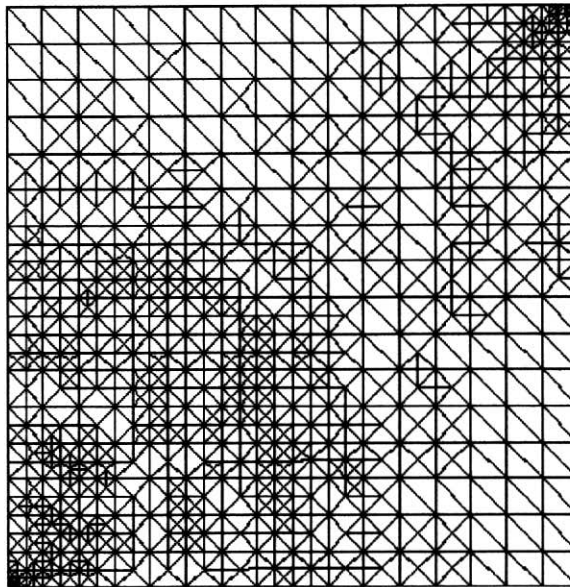


Figura 45: Malha com 908 nós e 1715 elementos, para o arranjo de cinco poços com distribuição de permeabilidade heterogênea fixa em 0.25 de VPI.

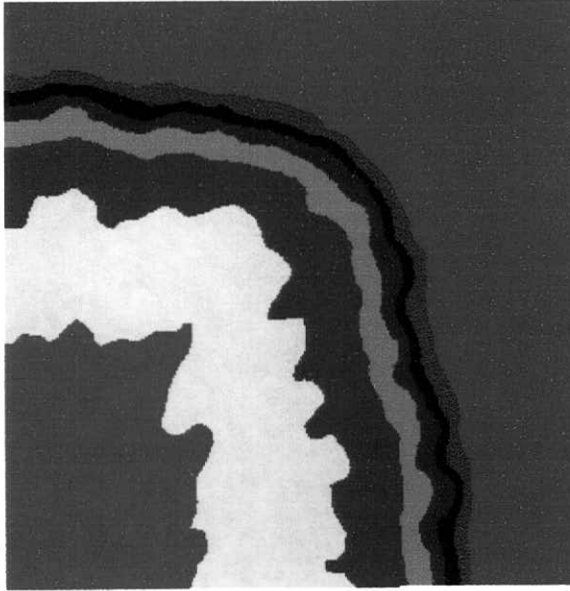


Figura 46: Frente de saturação em $t = 0.4$ para o arranjo de cinco poços com distribuição de permeabilidades heterogênea fixa.

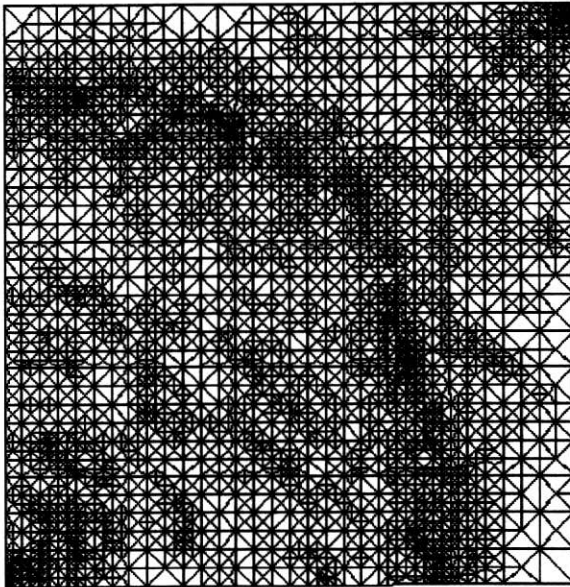


Figura 47: Malha com 3324 nós e 6494 elementos, para o arranjo de cinco poços com distribuição de permeabilidade heterogênea fixa em 0.5 de VPI.

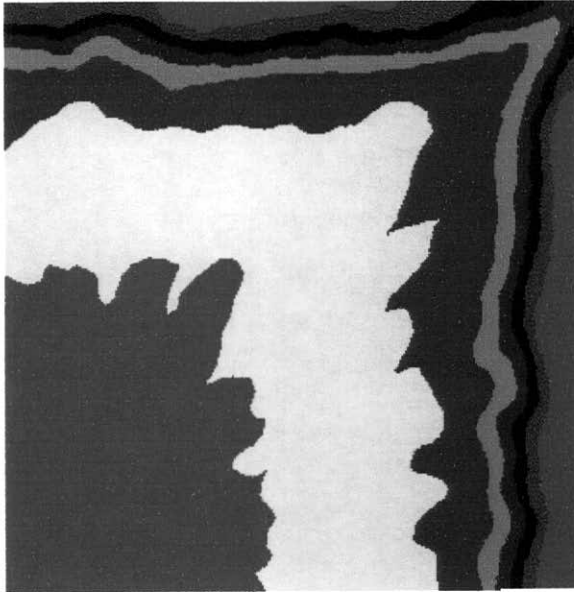


Figura 48: Frente de saturação em $t = 0.6$ para o arranjo de cinco poços com distribuição de permeabilidades heterogênea fixa.

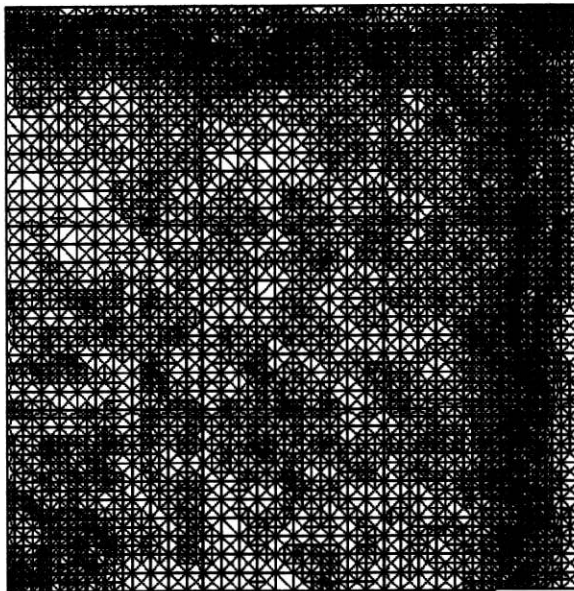


Figura 49: Malha com 6382 nós e 12531 elementos, para o arranjo de cinco poços com distribuição de permeabilidade heterogênea fixa em $3/4$ de VPI.

gasto na solução da equação da pressão. O número médio de multicorreções e o número médio de ciclos executados pelo algoritmo GMRES e JCB foi baixo.

Na solução com distribuição de permeabilidade fixa observa-se uma maior tendência à formação de *fingering*.

Capítulo 6

Conclusões e Trabalhos Futuros

Neste trabalho implementou-se um Método de Elementos Finitos Adaptativo em duas dimensões para problemas transientes utilizando-o para resolver problemas de escoamento de fluidos bi-fásicos imiscíveis ligeiramente compressíveis em meios porosos. A técnica de refinamento seletivo utilizada produz uma malha final aninhada com uma boa distribuição dos ângulos internos, garantido que a malha inicial seja regular e suave.

O código desenvolvido foi utilizado para simular diversos tipos de problemas de engenharia de petróleo e modelos diferentes de reservatório, sendo que os resultados obtidos apresentam boa concordância com a literatura.

Com o refinamento/desrefinamento aplicado apenas nas regiões onde a solução apresenta irregularidades, obteve-se uma solução de boa qualidade com um baixo custo computacional, como pode ser observado na comparação de desempenho com malhas uniformes. O tempo de CPU gasto pelas rotinas responsáveis pelo refinamento / desrefinamento é também muito pequeno em relação ao tempo total de CPU.

Observa-se em todos os casos analisados que, para um mesmo número de elementos, as soluções com malha adaptadas parecem representar melhor as não-linearidades, contribuindo para diminuir o número médio de multicorrecções.

A utilização de tensor de permeabilidade anisotrópico contribui para um decréscimo da taxa de convergência do método iterativo, como pode ser observado pelo aumento do número de iterações do método dos gradientes conjugados com preconditionador de Gauss-Seidel (PCG) utilizado para resolver a equação da pressão. Para o problema com distribuição heterogênea de permeabilidades, também se observou um acréscimo no

número de iterações executadas pelo método dos gradientes conjugados com preconditionador Gauss-Seidel. Nestes casos talvez fosse recomendado a utilização do método multimalha.

A utilização da formulação estabilizada tipo SUPG, juntamente com operador de captura de descontinuidade, mostrou-se eficiente para evitar o surgimento de oscilações espúrias nas frentes de saturação.

O indicador de erro aqui implementado mostrou-se capaz de identificar as regiões a serem refinadas/desrefinadas conforme observado nos exemplos e possui um custo computacional extremamente baixo.

As curvas de recuperação de óleo e volume de óleo recuperado mostram que o método adaptativo minimizou os efeitos de orientação de malha.

Uma extensão deste trabalho para três dimensões, é um tema para pesquisas futuras, incluindo aí a possibilidade de trabalhar com outros tipos de elementos e indicadores de erro. Entretanto tal extensão envolve um grande trabalho de implementação. Aplicar a estrutura adaptativa aqui apresentada para fluidos miscíveis também é promissor.

Como pode ser observado a maior carga computacional está concentrada na solução da equação da pressão, por isto, um solucionador multimalha associado ao esquema adaptativo pode contribuir para minimizar o custo computacional.

Pretende-se empregar este simulador adaptativo para a solução de casos práticos de interesse da indústria do petróleo, particularmente na simulação de migração de óleo. Além disso, dada a sua alta precisão, o simulador do presente trabalho seria uma ferramenta extremamente eficiente na compreensão dos fenômenos envolvidos no desenvolvimento de fingers viscosos, tanto em escoamentos imiscíveis quanto miscíveis.

Bibliografia

- [1] ALMEIDA, R. C., FEIJÓ, R. A., SILVA, R. S., "Adaptive Finite Element Computational Fluid Dynamics Using an Anisotropic Error Estimator". In: *Fourth World Congress on Computational Mechanics Buenos Aires Argentina*, 1998.
- [2] ALMEIDA, R. C., GALEÃO, A. C., "An Adaptive Petrov-Galerkin Formulation for the Compressible Euler and Navier-Stokes Equations". *Computer Methods in Applied Mechanics and Engineering*, v. 129, pp. 157-176, 1996.
- [3] AXELSSON, O., BARKER, V. A., *Finite Element Solution of Boundary Value Problems*. Academic Press, Inc, 1984.
- [4] AZIZ, K., SETTARI, A., *Petroleum Reservoir Simulation*. Applied Science Publishers, 1983.
- [5] CHRISTIAN, F., SANTOS, G., "A Posteriori Estimators for Nonlinear Elliptic Partial Differential Equations". *Journal of Computational and Applied Mathematics*, v. 103, pp. 99-114, 1999.
- [6] COUTINHO, A. L. G. A., ALVES, J. L. D., "Parallel Finite Element Simulation of Miscible Displacements in Porous Media". *SPE Journal 1 (4) SPE 37399*, pp. 487-500, 1996.
- [7] COUTINHO, A. L. G. A., ALVES, J. L. D., DEVLOO, P. R. B. "Uma Técnica Elemento por Elemento Livre de Matrizes para a Avaliação de Resíduos". *Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería*. v. 11, n. 4, pp. 487-497, 1995.
- [8] COUTINHO, A. L. G. A., ALVES, J. L. D., GARCIA, E. L. M., LOULA, A. F. D., "Solution of Miscible and Immiscible Flows Employing Element-by-Element Iterative Strategies". *3rd SPE Latin American/Caribbean Engin. Conference SPE 27050*, pp. 431-444, 1994.
- [9] COUTINHO, A. L. G. A., ALVES, J. L. D., LANDAU, L., EBECKEN, N. F. F., "A Dynamic Mesh Partition Algorithm for the Finite Element Solution of Two-Phase Immiscible Flow in Oil Reservoirs". *Finite Element in Fluids: New Trends and Applications.*, v. II, pp. 856-865, 1993.

- [10] COUTINHO, A. L. G. A., SILVA, A. S., DEVLOO, P. R. B., "Uma Comparação de Esquemas de Estabilização para a Simulação por Elementos Finitos de Escoamentos Imiscíveis Bi-Fásicos em Meios Porosos". In: *20^o Iberian Latin-American Congress on Computational Methods in Engineering, CD-ROM*, São Paulo, 1999.
- [11] DEB, M. K., REDDY, M. P., THUREN, J. B., ADAMS, W. T., "A New Generation Solution Adaptive Reservoir Simulator". *Society of Petroleum Engineers SPE 30720*, pp. 175-189, 1995.
- [12] DEVLOO, P. R. B., *An H-P Adaptive Finite Element Method for Steady Compressible Flow*. Ph.D. dissertation, University of Texas at Austin, Austin, Texas, USA, 1987.
- [13] DEVLOO, P. R. B., "Implementation of Directional Boundary Conditions for Finite Element Computations with Iterative Solvers". *Comm. in Applied Numerical Methods*, v. 3, pp. 259-264, 1987.
- [14] DEVLOO, P. R. B., "A Three-Dimensional Adaptive Finite Strategy". *Computers and Structures*, v. 38, pp. 121-130, 1991.
- [15] DEVLOO, P. R. B., ODEN, J. T., STROUBOULIS, T., "Implementation of an Adaptive Refinement Technique for the SUPG Algorithm". *Computer Methods in Applied Mechanics and Engineering*, v. 61, pp. 339-358, 1987.
- [16] DÍEZ, P., EGOZCUE, J. J., HUERTA, A., "A Posteriori Error Estimates for Standard Finite Element Analysis". *Computer Methods in Applied Mechanics and Engineering*, v. 163, pp. 141-157, 1998.
- [17] DURLOFSKY, L. J., "A Triangle Based Mixed Finite Element-Finite Volume Technique for Modeling Two Phase Flow Through Porous Media". *Journal of Computational Physics*, v. 105, pp. 252-266, 1993.
- [18] EDWARDS, M. G., ROGERS, C. F., "Finite Volume Discretization with Imposed Flux Continuity for the General Tensor Pressure Equation". *Computational Geosciences*, v. 2, pp. 259-290, 1998.
- [19] ESPEDAL, M. S., EWIND, R. E., "Characteristic Petrov-Galerkin Subdomain Methods for Two-Phase Immiscible Flow Problems". *Computer Methods in Applied Mechanics and Engineering*, v. 64, pp. 113-135, 1987.

- [20] EWIND, R. E., *The Mathematics of Reservoir Simulation*, SIAM, Philadelphia, 1983.
- [21] FUNG, L. S. K., HIEBERT, A. D., NGHIEM, L. "Reservoir Simulation with a Control-Volume Finite Element Method". *SPEERE Agosto*, pp. 349-357, 1992.
- [22] GALEÃO, A. C. R. N., CARMO, E. G. D., "A Consistent Approximate Upwind Petrov-Galerkin Formulation for Convection Dominated Problems". *Computer Methods in Applied Mechanics and Engineering*, v. 66, pp. 83-95, 1988.
- [23] HANSEN, R., ESPEDAL, M. S., "On the Numerical Solution of Non-linear Reservoir Flow Models with Gravity". *International Journal for Numerical Methods in Engineering*, v. 38, pp. 2017-2032, 1995.
- [24] HORNUNG, D., TRANGENSTEIN, J. A., "Adaptive Mesh Refinement and Multilevel Iteration for Flow in Porous Media". *Journal of Computational Physics*, v. 136, pp. 522-545, 1997.
- [25] HUGHES, T. J. R., *Finite Element Method, Linear Static and Dynamic Finite Element Analysis*. Prentice-hall, Englewood Cliffs, New Jersey, 1987.
- [26] HUGHES, T. J. R., FERENCZ, R. M., HALLQUIST, J. O., "Large-Scale Vectorized Implicit Calculations in Solid Mechanics on a CRAY x MP/48 Utilizing EBE Preconditioned Conjugate Gradients". *Computer Methods in Applied Mechanics and Engineering*, v. 61, pp. 215-248, 1987.
- [27] HUYAKORN, P.S., PINDER, G. F., "Solution of Two-Phase Flow Using a New Finite Element Technique". *Applied Numerical Modeling*, Ed. C. A. Brebbia, pp. 375-390, 1975.
- [28] PEDROSA JR., O. A., AZIZ, K., "Use of a Hybrid Grid in Reservoir Simulation". *SPEERE Novembro*, pp. 611-621, 1986.
- [29] LANGTANGEN, H. P., "Implicit Finite Element Methods for Two-Phase Flow in Oil Reservoirs". *International Journal for Numerical Methods in Fluids*, v. 10, pp. 651-681, 1990.
- [30] LEWIS, R. W., MORGAN, K., JOHNSON, K. H., "A Finite Element Study of Two-Dimensional Multiphase Flow with Particular Reference to the Five-Spot Problem". *Computer Methods in Applied Mechanics and Engineering*, v. 44, pp. 17-47, 1984.

- [31] LOHNER, R., "Some Useful Data Structures for the Generation of Unstructured Grids". *Communication in Applied Numerical Methods*, v. 4, pp. 123-135, 1988.
- [32] LOHNER, R., "Finite Element Methods in CFD Grid Generation Adaptivity and Parallelization". Tech. Rep. AGARD Report 787, 1992.
- [33] LOULA, A. F. D., GARCIA, E. L. M., MURAD, M. A., MALTA, S. M. C., SILVA, R. S., "Métodos de Elementos Finitos Aplicados à Simulação de Petróleo". Tech. Rep., Projeto CENPES/LNCC, 1992.
- [34] LOULA, A. F. D., GUERREIRO, J. N. C., RIBEIRO, F. L. B., LANDAU, L., "Tracer Injection Simulation by Finite Element Methods". *SPE Advanced Technology Series, SPE 27047*, v. 4, n. 1, pp. 150-156, 1996.
- [35] MALTA, S. M. C., LOULA, A. F. D., GARCIA, E. L. M., "A Post-Processing Technique to Approximate the Velocity Field in Miscible Displacement". *Contemporary Mathematics*, v. 8, pp. 239-268, 1995.
- [36] MATHISEN, K. M., HOPPERSTAD, O. S., OKSTAD, K. M., "Error Estimation and Adaptivity in Explicit Non-linear Finite Element Simulation of Quasi-Static Problems". *Computers & Structures*, v. 72, pp. 627-644, 1999.
- [37] MAVRIPLIS, D. J., "Unstructured Mesh Generation and Adaptivity". Tech. Rep. ICA-SE Report 95-26, 1995.
- [38] PARSONS, I. D., COUTINHO, A. L. G. A., "Finite Element Multigrid Methods for Two-Phase Immiscible Flow in Heterogeneous Media". *Communication in Numerical Methods in Engineering*, v. 15, pp. 1-7, 1999.
- [39] PEACEMAN, D. W., *Fundamentals of Numerical Reservoir Simulation*. Elsevier Scientific Publishing Company, Amsterdam, 1977.
- [40] PICASSO, M., "Adaptive Finite Elements for a Linear Parabolic Problem". *Computer Methods in Applied Mechanics and Engineering*, v. 167, pp. 223-237, 1998.
- [41] REDDY, M. P., DEB, M. K., BASS, J. M., NING, H., "Numerical Simulation of Non-Conventional Wells Using Adaptive Finite Element Analysis". *Computer Methods in Applied Mechanics and Engineering*, v. 150, pp. 109-124, 1997.
- [42] RIBEIRO, F. L. B., LANDAU, L., "Na H-Refinement Strategy for Convection Dominated Problems Using Triangular Finite Elements". In: *XIV Congresso*

Ibero Latino Americano sobre Métodos Computacionais para Engenharia, v. II, pp. 713-722, 1993.

- [43] RIVARA, M. C., "Selective Refinement/Derefinement Algorithms for Sequences of Nested Triangulations". *International Journal for Numerical Methods in Engineering*, v. 28, pp. 2889-2906, 1989.
- [44] RIVARA, M. C., "Local Modification of Meshes for Adaptive and/or Multigrid Finite-Element Methods". *Journal of Computational and Applied Mathematics*, v. 36, pp. 79-89, 1991.
- [45] SAAD, Y., *Iterative Methods for Sparse Linear Systems*. PWS Publishing Company, 1996.
- [46] SAMPAIO, P. A. B., COUTINHO, A. L. G. A., "Simulation of Free and Forced Convection Incompressible Flows Using an Adaptive Parallel/Vector Finite Element Procedure". *International Journal for Numerical Methods in Fluids*, v. 29, pp. 289-309, 1999.
- [47] SILVA, A. S., COUTINHO, A. L. G. A., PARSONS, I. D., "Métodos Multigrid para a Simulação por Elementos Finitos de Escoamentos Imiscíveis em Meios Porosos". *Revista Internacional de Metodos Numericos y Deseño en Ingenieria Aceito para Publicação*.
- [48] SILVA, A. S., DEVLOO, P. R. B., COUTINHO, A. L. G. A., "Métodos Adaptativos para Simulação de Fluidos Bi-Fásicos". In: *20º Iberian Latin American Congress on Computational Methods in Engineering, CD-ROM, 1999*.
- [49] SILVA, R. S., *Estratégias de Resolução em Sistemas Distribuídos de Problemas em Dinâmica dos Fluidos Computacionais Via Elementos Finitos Adaptativos*. Tese de D.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil, 1998.
- [50] SILVA, R. S., GALEÃO, A. C., CARMO, E. G. D. D., "Um Modelo de Petrov-Galerkin Combinado com Esquemas Auto Adaptativos e sua Aplicação na Resolução de Problemas com Camada Limite". In: *X Congresso Brasileiro de Engenharia Mecânica*, v. 1, pp. 185-188, 1989.
- [51] SILVA, R. S., GALEÃO, A. C., CARMO, E. G. D. D., "Um Estimador a-Posteriori para Modelos de Petrov-Galerkin Aplicados à Problemas com Camada Limites". In: *XI Congresso Brasileiro de Engenharia Mecânica*, v.2, pp.193-196, 1991.

- [52] SPIVAK, A., PRICE, H. S., SETTARI, A., "Solution of the Equations for Multidimensional, Two-Phase, Immiscible Flow by Variational Methods". *SPE Journal*, pp. 27-41, 1981.
- [53] STEVENSON, M. D., KAGAN, M., PINCZEWSKI, W. V., "Computational Methods in Petroleum Reservoir Simulation". *Computers & Fluids*, v. 19, pp. 1-19, 1991.
- [54] STEWART, J. R., HUGHES, T. J. R., "A Tutorial in Elementary Finite Element Error Analysis: A Systematic Presentation of a Priori and a Posteriori Error Estimates". *Computer Methods in Applied Mechanics and Engineering*, v. 158, pp. 1-22, 1998.
- [55] TEIGLAND, R., "On Some Variational Techniques and Related Methods for Local Refinement". *Int. Journal for Numerical Methods in Fluids*, v. 28, n. 6, pp. 945-960, 1998.
- [56] ZIENKIEWICZ, O. C., ZHU, J. Z., "A Simple Error Estimator and Adaptive Procedure for Practical Engineering Analysis". *International Journal for Numerical Methods in Engineering*, v. 24, pp. 337-357, 1987.