Simulação de Reservatórios de Petróleo em Arquiteturas Paralelas com Memória Distribuída

Adriano Augusto Mucarbel Soares

Dissertação de Mestrado submetida ao Corpo Docente do Curso de Pós-Gradução em Engenharia Civil da Universidade Federal de Pernambuco, como parte dos requisitos necessários à obtenção do grau de Mestre em Ciências em Engenharia Civil.

Área de Concentração: Engenharia de Petróleo

Ézio da Rocha Araújo, D.Sc.
Orientador
Ramiro Brito Willmersdorf, Ph. D.
Co-Orientador

Recife, Pernambuco, Brasil

Simulação de Reservatórios de Petróleo em Arquiteturas Paralelas com Memória Distribuída

Adriano Augusto Mucarbel Soares

Dissertação de Mestrado apresentada em abril de 2002

Ézio da Rocha Araújo, D.Sc.
(Orientador)

Ramiro Brito Willmersdorf, Ph. D.
(Co-Orientador)

Silvana Maria Bastos Afonso da Silva, Ph. D.
(Examinadora Interna)

Denis José Schiozer, Ph. D.
(Examinador Externo)

Recife, Pernambuco, Brasil, abril de 2002

Dedicatória

Dedico este trabalho a meus pais Telmo e Socorro, pois sem eles não estaria concluindo mais esta etapa da minha vida, meus queridos irmãos Felipe e Thiago, e minha eterna namorada Patricia.

Agradecimentos

À ANP - Agência Nacional do Petróleo, pelo apoio ...nanceiro e por permitir a realização deste trabalho como parte do programa de recursos humanos PRH-ANP/MME/-MCT - Edital CTPETRO 02/99, PRH-26 - Arquitetura de Depósitos Sedimentares para Análogos de Hidrocarbonetos.

À Professora Margareth Alheiros e ao Prof. Virgínio Henrique Neumann pela organização e coordenação do PRH-26.

Ao amigo Prof. Ézio da Rocha Araújo pela valiosa orientação e incentivo para o desenvolvimento deste trabalho.

Ao Prof. Ramiro Brito Willmersdorf pela ajuda e conselhos que auxiliaram na utilização do sistema operacional Linux e do cluster de PCs, possibilitando a obtenção dos resultados de desempenho do simulador.

À Prof. Silvana Maria Bastos Afonso da Silva pelo incentivo e apoio para o desenvolvimento deste trabalho.

Ao Professor Rafael Dueire Linz e aos colegas da Sala de Cursos Especiais do Departamento de Eletrônica e Sistemas por possibilitarem a utilização do cluster de PCs para a análise de desempenho do simulador.

Aos professores e amigos dos cursos de graduação e pós-graduação do Departamento de Geologia da UFPE e que compõem o PRH-26, por todos os momentos em que compartilhamos novas idéias e pelo convívio durante este período.

Aos professores do curso de Pós-graduação em Engenharia Civil da UFPE, pelos ensinamentos e troca de experiências que contribuíram para a minha formação, e especialmente aos professores Paulo Régis, Evandro e Áurea, pela amizade e apoio.

Aos amigos e companheiros do curso de Pós-graduação em Engenharia Civil da UFPE, Adriano Siebra, Ana Paula, Carla, Carlúcio, Daniel, Darlan, Érico, Gabriela, Isaac, Joelma, Juliana, Natécia, Nilson, Simone, pelo convívio e amizades conquistadas.

Resumo

Novos códigos para simulação de reservatórios de petróleo vêm sendo desenvolvidos para tratar problemas de larga escala usando processamento de alto desempenho. Este trabalho consiste no desenvolvimento de um simulador de reservatórios usando técnicas de programação para computadores paralelos com memória distribuída. É implementado o modelo Black-Oil bifásico óleo/água, cujas equações são discretizadas usando o método das diferenças finitas em malhas retangulares de blocos centrados. As equações discretas são linearizadas utilizando a formulação IMPES e a totalmente implícita. As não-linearidades são tratadas utilizando o método de Newton-Inexato, com ciclo interno - o sistema linear - resolvido por diversos métodos iterativos com diferentes precondicionadores. Para isto é utilizado o software PETSc, que utiliza o MPI para efetuar toda comunicação de dados entre processadores. Os resultados obtidos, comparados com os do simulador comercial BOAST-98, validam o modelo implementado. A análise de desempenho realizada em um computador paralelo, especificamente um cluster de PCs, apresenta eficiências bastante elevadas de até 97%, validando também a implementação do modelo de programação paralela.

Abstract

New codes have been developed for the solution of large scale models in reservoir simulations with high performance computing. This work shows the development of a petroleum reservoir simulator using distributed memory models for parallel computers. A two phase oil/water Black-Oil model was implemented and the resulting equations are discretized by …nite di¤erences approximations in a 2D rectangular cell-centered grid. The discrete equations are linearized using IMPES and Fully Implicit numerical formulations. The nonlinearities are handled by inexact Newton method, with the inner looping - linear system - solved by di¤erent iterative methods and di¤erent preconditioners. The resulting systems of linear and nonlinear equations are solved using the software PETSc, which uses MPI for communications between processors. The results of the simulator are compared with the commercial simulator BOAST-98 for applications purposes. The performance of the simulator was measured in a parallel computer, actually a cluster of PCs, with e¢ciencies up to 97%, showing the applicability of the parallel model.

Lista de Simbolos

Letras Romanas

$a_{i;j}$	¡	elemento da linha i e coluna j da matriz
$A_x; A_y; A_z$	¡	área das faces do elemento de volume de controle segundo as direções x, y e z:
A	¡	matriz do sistema de equações lineares
b_i	¡	elemento da linha i do vetor
$b_{i;j}$	¡	elemento do bloco i; j, do vetor de termos independentes
b	¡	vetor de termos independentes
B_l	¡	fator volume de formação do componente l
B_l^{ref}	¡	fator volume de formação de referência para o componente l
c_l	¡	compressibilidade do ‡uido do componente l
c_r	¡	compressibilidade da rocha
C	¡	elemento da matriz do sistema linear
$C_{poo_{i;j}}^{n+1}; C_{pow_{i;j}}^{n+1}$	¡	coe...cientes que multiplicam as pressões de óleo nos termos de acumulação nas equações de óleo e água, respectivamente
$C_{swo_{i;j}}^{n+1}; C_{sww_{i;j}}^{n+1}$	¡	coe...cientes que multiplicam as saturações de água nos termos de acumulação nas equações de óleo e água, respectivamente
C	¡	submatriz que forma a matriz Jacobiana
D	¡	matriz diagonal formada pela diagonal principal da matriz
E	¡	elemento da matriz do sistema linear
E_P	¡	medida de e...ciência usando P processadores
E	¡	submatriz que forma a matriz Jacobiana; matriz triangular composta por elementos estritamente abaixo da diagonal principal
f	¡	função ou variável qualquer
f	¡	vetor de iteração do método iterativo
F	¡	matriz triangular composta por elementos estritamente acima da diagonal principal da matriz
g	¡	aceleração da gravidade; variável qualquer
$G_{x_{i\,\S\,1=2;j}}; G_{y_{i;j\,\S\,1=2}}$	¡	fatores geométricos nas faces do bloco, nas direções x e y
G_w	¡	fator geométrico de poço
G	¡	matriz de iteração do método iterativo
i	¡	índice que representa o bloco da malha

$I_{MB_l}^{n+1}$ ¡ índice de balanço de massa incremental para o componente l, no passo de tempo n + 1

I ¡ matriz identidade

j ¡ índice que representa o bloco da malha

J ¡ matriz Jacobiana usada na formulação totalmente implícita

k_H ¡ permeabilidade absoluta horizontal média no bloco do poço

$k_x; k_y; k_z$ ¡ permeabilidades absolutas segundo as direções x; y e z

k_{rl} ¡ permeabilidade relativa do componente l

$k_x; k_y$ ¡ vetores de permeabilidades absolutas

K ¡ tensor de permeabilidades absolutas

l ¡ índice usado para representar os componentes do sistema (l = o; w)

m ¡ dimensão do subespaço de Krylov; massa; índices dos blocos da malha

$\underline{m}_{l_x}; \underline{m}_{l_y}; \underline{m}_{l_z}$ ¡ ‡uxo de massa do componente l que entra ou sai do elemento de volume através de suas faces segundo as direções x; y e z

$(m_e)_l ; (m_s)_l$ ¡ massa do componente l que entra ou que sai do elemento através de suas faces, respectivamente

$(m_f)_l$ ¡ massa do componente l que entra no elemento devido a uma fonte localizada no seu interior

$(m_a)_l$ ¡ massa do componente l que …ca acumulada no elemento em um determinado intervalo de tempo

m_{v_l} ¡ massa do componente l contida no elemento

M ¡ matriz de precondicionamento ou precondicionador

n ¡ índice do passo de tempo anterior; dimensão da matriz e vetor do sistema; índices dos blocos da malha

$n_x; n_y$ ¡ número de blocos da malha segundo as direções x e y

N ¡ dimensão do vetor; elemento que forma a matriz da formulação IMPES

N ¡ submatriz que forma a matriz Jacobiana

o ¡ índice do componente óleo

O () ¡ ordem do erro da aproximação da série de Taylor

p_l ¡ pressão do componente l

p_l^{ref} ¡ pressão de referência do componente l

p_{wf} ¡ pressão de fundo de poço

p_o ¡ vetor de pressões de óleo usado na formulação IMPES

P	¡	número de processadores
$P_c; P_{cow}$	¡	pressão de capilaridade
q_{es}	¡	vazão especi…cada no poço
q_l	¡	vazão do componente l em condições de reservatório
q_{lsc}	¡	vazão do componente l em condições padrão; termo de poço
q_{m_l}	¡	‡uxo de massa do componente l em condições de reservatório
r_{eq}	¡	raio equivalente do poço ou raio de drenagem
$r_{l_{i;j}}$	¡	resíduo da equação de ‡uxo do componente l, no bloco i; j
r_w	¡	raio do poço
r	¡	vetor de resíduo do sistema linear
R	¡	vetor de resíduos multicomponente usado na formulação totalmente implícita;
s	¡	fator de skin
S	¡	elemento que forma a matriz da formulação IMPES
S_l	¡	saturação do componente l
S_P	¡	medida de speedup usando P processadores
S	¡	submatriz que forma a matriz Jacobiana
S_w	¡	vetor de saturações de água usado na formulação IMPES
t	¡	tempo; tempo de simulação
$T_{lx}; T_{ly}$	¡	transmissibilidades do componente l, segundo as direções x e y
$T_1; T_P$	¡	tempos de execução usando 1 e P processadores
$u_{l_x}; u_{l_y}; u_{l_z}$	¡	velocidade de Darcy segundo as direções x; y e z
u_l	¡	vetor de velocidades de Darcy
V_l	¡	volume do componente l em condições de reservatório
V_{lsc}	¡	volume do componente l em condições padrão
$V_{b_{i;j}}$	¡	volume do bloco i; j
w	¡	índice do componente água
W	¡	elemento que forma a matriz da formulação IMPES
W	¡	submatriz que forma a matriz Jacobiana
x	¡	variável espacial
x	¡	vetor de incógnitas do sistema linear; vetor qualquer
X	¡	vetor multicomponente com incógnitas do sistema não-linear
y	¡	variável espacial
z	¡	variável espacial
Z	¡	profundidade dos pontos centrais dos blocos, medida em relação a um ponto …xo
Z	¡	vetor de profundidades dos blocos

Letras Gregas

α	¡	variável qualquer
γ_l	¡	peso específico do componente l
$\pm X$	¡	vetor de incrementos do sistema não-linear
$\Delta x; \Delta y; \Delta z$	¡	dimensões dos blocos
Δt	¡	intervalo ou passo de tempo
$\Delta x; \Delta y; \Delta z$	¡	vetor contendo as dimensões dos blocos
ε	¡	variável de incremento
$\eta_{l\,i;j}$	¡	mobilidade do componente l no bloco i; j
$\eta_{l\,i+1=2;j}; \eta_{l\,i;j+1=2}$	¡	mobilidades médias entre blocos
λ_l	¡	viscosidade do componente l
$(^o)$	¡	índice de iteração
ρ_l	¡	densidade do componente l em condições de reservatório
ρ_l^{ref}	¡	densidade de referência do componente l
ρ_{lsc}	¡	densidade do componente l em condições padrão
ϕ	¡	porosidade da rocha
ϕ^{ref}	¡	porosidade da rocha correspondente a uma pressão de referência
Φ	¡	vetor de porosidades da rocha
Φ_l	¡	potencial gravitacional do componente l
ω	¡	fator de sobre-relaxação usado no método SOR

Índice

1 Introdução 1

1.1 Descrição do Problema 1

1.2 Objetivos do Trabalho 4

1.3 Organização dos Capítulos 5

2 Formulação Matemática 6

2.1 Descrição das Equações Básicas 6

2.1.1 Equações de Estado 6

2.1.2 Lei de Darcy 9

2.1.3 Lei da Conservação de Massa 10

2.2 Equações de Fluxo e Relações Constitutivas 13

2.3 Condições de Contorno e Condições Iniciais 14

2.3.1 Condições de Contorno 14

2.3.2 Condições Iniciais 15

3 Discretização e Formulação Numérica 16

3.1 Método das Diferenças Finitas 16

3.1.1 Aproximação das Derivadas no Espaço 17

3.1.2 Aproximação das Derivadas no Tempo 18

3.2 Discretização das Equações de Fluxo 19

3.2.1 Termos de Fluxo 20

3.2.2 Termos de Acumulação 21

3.2.3 Termos de Poços 23

3.2.4 Forma Final da Equação de Fluxo Discretizada 25

3.3 Introdução das Condições Iniciais e de Contorno 26

3.3.1 Inicialização das Pressões e Saturações 26

3.3.2 Introdução das Condições de Contorno 27

3.4 Linearização das Equações de Fluxo 27

3.4.1 Formulação Totalmente Implícita 27
3.4.2 Formulação IMPES 33
3.5 Detalhes Adicionais 35
3.5.1 Controle do Passo de Tempo 35
3.5.2 Veri...cação de Balanço de Massa 36

4 Métodos para Solução de Sistemas Lineares 37
4.1 Métodos Iterativos e Subespaços de Krylov 38
4.1.1 Métodos Estacionários 38
4.1.2 Métodos Não-Estacionários 40
4.2 Técnicas de Precondicionamento 45
4.2.1 Introdução 45
4.2.2 Formas de Precondicionamento 46

5 Noções sobre Computação Paralela 49
5.1 Introdução 49
5.1.1 Computadores Sequenciais 49
5.1.2 Computadores Paralelos 50
5.2 Arquiteturas de Computadores Paralelos 51
5.2.1 Taxonomia de Flynn 51
5.2.2 Outras Classi...cações 52
5.3 Métodos para Paralelização de um Programa 53
5.3.1 Partição do Programa 53
5.3.2 Comunicação entre Processadores 54
5.3.3 Efeito da Granularidade 56
5.3.4 Mapeamento e Balanceamento de Carga 57
5.4 Medidas de Desempenho 57
5.5 Paralelização de Algumas Operações Básicas 58
5.6 Modelos de Programação Paralela 59
5.7 Bibliotecas Paralelas 60
5.7.1 MPI - Message Passing Interface 60
5.7.2 PETSc - Portable, Extensible Toolkit for Scienti...c Computing . 64

6 Implementação do Simulador 67
6.1 Características do Simulador 67
6.1.1 Malha de Blocos Centrados 67

6.1.2 Dados de Entrada 68
6.1.3 Impressão e Visualização dos Resultados 70
6.1.4 Paralelização do Simulador 71
6.2 Etapas da Simulação 72
6.2.1 Inicialização do Programa 72
6.2.2 De…nições Iniciais 72
6.2.3 Criação da Interface de Visualização Grá…ca 74
6.2.4 Criação da Interface de Particionamento da Malha 74
6.2.5 Criação dos Vetores Globais e Locais 75
6.2.6 Criação da Matriz 77
6.2.7 Criação do Solver 78
6.2.8 Leitura e Transferência dos Dados de Entrada 79
6.2.9 Inicialização das Pressões e Saturações 80
6.2.10 Montagem do Vetor do Sistema 81
6.2.11 Montagem da Matriz do Sistema 82
6.2.12 Solução do Sistema de Equações 83
6.2.13 Cálculo Explícito das Saturações de Água 83
6.2.14 Visualização dos Resultados 84
6.2.15 Impressão dos Resultados 84
6.2.16 Controle do Passo de Tempo 86
6.2.17 Destruição dos Objetos e Finalização do Programa 86

7 Aplicações e Resultados Numéricos 87
7.1 Comparação dos Resultados do Simulador 87
7.1.1 Caso 1: Reservatório 1D 88
7.1.2 Caso 2: Reservatório 2D com modelo 5-spot 94
7.1.3 Caso 3: Reservatório Heterogêneo 96
7.2 Análise de Desempenho do Simulador 103
7.2.1 Descrição do Cluster de PCs 103
7.2.2 Modelo Analisado 103
7.2.3 Resultados com a Formulação IMPES 105
7.2.4 Resultados com a Formulação Totalmente Implícita 105
7.2.5 Speedup Escalável 108

8 Conclusões e Trabalhos Futuros 111
8.1 Conclusões 111

8.1.1 Trabalhos Futuros 113

Lista de Tabelas

5.1 Relação entre tipos de dados do MPI com o Fortran 61
5.2 Subrotinas de comunicação ponto a ponto 62
5.3 Operações de redução global 64

6.1 Vetores criados e suas respectivas dimensões 75

7.1 Dados do reservatório - Caso 1 88
7.2 Propriedades dos ‡uidos - Caso 1 88
7.3 Permeabilidade relativa e pressão de capilaridade - Casos 1 e 2 89
7.4 Dados de poços - Caso 1 89
7.5 Dados de poços - Caso 2 94
7.6 Dados do reservatório - Caso 3 101
7.7 Propriedades dos ‡uidos - Caso 3 101
7.8 Permeabilidades relativas e pressão de capilaridade - Caso 3 102
7.9 Dados de poços - Caso 3 102
7.10 Tolerâncias utilizadas nos solvers 104
7.11 Medidas de desempenho usando a formulação IMPES. 105
7.12 Medidas de desempenho usando a formulação totalmente implícita. . . 108
7.13 Medidas de desempenho obtidas variando o tamanho da malha em função do número de processadores 109

Lista de Figuras

2.1 Elemento de volume de controle. 11

3.1 Discretização unidimensional usando o esquema de blocos centrados. . . 17
3.2 Esquema de discretização com 5 pontos - modelo bidimensional. 19
3.3 Malha de blocos centrados, 3x3. 28
3.4 Esquema de diferenças ...nitas com 5 pontos - duas variáveis por bloco. 29

5.1 Arquiteturas MIMD. (a) memória compartilhada; (b) memória distribuída. 52
5.2 Algumas topologias de conexão entre processadores. 53
5.3 Malha regular 5x5 particionada entre 4 processadores - decomposição regular 2D. 55
5.4 Malha global particionada e malha local do processador 0, incluindo valores ocultos referentes ao esquema de diferenças ...nitas com 5 pontos. 55
5.5 Esquema de comunicação das subrotinas coletivas. 63
5.6 Organização dos componentes do PETSc (BALAY et al., 2001). 65
5.7 Detalhes dos objetos do PETSc (BALAY et al., 2001). 65

6.1 Malha retangular de blocos centrados utilizada na aproximação da geometria de um reservatório com contorno irregular. 68
6.2 Fluxograma principal do simulador. 72
6.3 Fluxograma da implementação da formulação IMPES. 73
6.4 Fluxograma da implementação da formulação totalmente implícita. . . 73
6.5 Malha 5x5 particionada entre 4 processos - numeração natural e numeração do PETSc. 76
6.6 Vetor global paralelo e vetores locais incluindo valores ocultos. 76
6.7 Matriz partionada entre 4 processadores. 77
6.8 Processo de leitura de dados. 79
6.9 Fluxograma da rotina de inicialização das variáveis primárias. 80
6.10 Fluxograma das rotinas para montagem do vetor b e vetor R: 82

6.11 Fluxograma das rotinas para montagem das matrizes. 83
6.12 Fluxograma da rotina para o cálculo explícito das saturações de água usando a formulação IMPES. 84
6.13 Fluxograma da rotina de impressão de resultados. 85
6.14 Processo de impressão dos resultados para cada bloco da malha global. 85

7.1 Caso 1 - Modelo do reservatório. 88
7.2 Caso 1 - vazão de óleo no poço produtor, q_{osc}, para Δt = 1 dia e n_x = 20. 91
7.3 Caso 1 - pressão no poço injetor, p_{wf}, para Δt = 1 dia e n_x = 20. . . . 91
7.4 Caso 1 - pressão média no reservatório, para Δt = 1 dia e n_x = 20. . . . 91
7.5 Caso 1 - vazão de óleo no poço produtor, q_{osc}, para Δt = 1 dia e n_x = 10; 20; 40 - IMPES e totalmente implícito. 92
7.6 Caso 1 - vazão de óleo no poço produtor, q_{osc}, para Δt = 1 dia e n_x = 10; 20; 40 - BOAST-98. 92
7.7 Caso 1 - curvas da frente de saturação de água nos tempos de simulação 90, 180 e 360 dias. 92
7.8 Caso 1 - vazão de óleo no poço produtor, q_{osc}, para n_x = 10 e Δt = 1; 10 e 20 dias - IMPES. 93
7.9 Caso 1 - vazão de óleo no poço produtor, q_{osc}, para n_x = 10 e Δt = 1; 10 e 20 dias - totalmente implícito. 93
7.10 Caso 1 - vazão de óleo no poço produtor, q_{osc}, para n_x = 10 e Δt = 1; 10 e 20 dias - BOAST-98. 93
7.11 Caso 2 - Modelo do reservatório. 94
7.12 Caso 2 - vazão de óleo no poço produtor, q_{osc}. 95
7.13 Caso 2 - vazão de água no poço produtor, q_{wsc}. 95
7.14 Caso 2 - pressão no poço injetor, p_{wf}: 95
7.15 Caso 2 - pressão média no reservatório. 96
7.16 Caso 3 - Modelo do reservatório. 97
7.17 Dimensões, Δx e Δy, e espessura dos blocos, Δz, em ft - Caso 3. 97
7.18 Profundidade das faces superiores dos blocos, ft - Caso 3. 98
7.19 Permeabilidade absoluta na direção x, mD - Caso 3. 98
7.20 Porosidade da rocha - Caso 3. 98
7.21 Caso 3 - vazão de óleo no poço produtor W-2, q_{osc}: 99
7.22 Caso 3 - vazão de água no poço produtor W-2, q_{wsc}: 99
7.23 Caso 3 - vazão de óleo no poço produtor W-4, q_{osc}: 99
7.24 Caso 3 - vazão de água no poço produtor W-4, q_{wsc}: 100

7.25 Caso 3 - pressão no poço injetor W-3, p_{wf}: 100
7.26 Esquema de partição das malhas do modelo 1/4 de 5-spot. 104
7.27 Speedup para a malha 200x200 (40.000 eqs.) - IMPES. 106
7.28 Speedup para a malha 300x300 (90.000 eqs.) - IMPES. 106
7.29 Speedup para a malha 400x400 (160.000 eqs.) - IMPES. 106
7.30 Speedup para a malha 150x150 (45.000 eqs.) - totalmente implícita. . . 107
7.31 Speedup para a malha 200x200 (80.000 eqs.) - totalmente implícita. . . 107
7.32 Speedup para a malha 240x240 (115.200 eqs.) - totalmente implícita. . . 107
7.33 Speedup escalável e tempo de execução com malha variável - 400x400 (160.000 eqs.), 566x566 (320.356 eqs.) e 800x800 (640.000 eqs.) - IMPES. 110
7.34 Speedup escalável e tempo de execução com malha variável - 240x240 (115.200 eqs.), 340x340 (231.200 eqs.), 480x480 (460.800 eqs.) - totalmente implícita. 110

Capítulo 1

Introdução

1.1 Descrição do Problema

A simulação de reservatórios é uma ferramenta muito importante para a avaliação do comportamento do ‡uxo multifásico de óleo, água e gás em reservatórios de petróleo, permitindo ao engenheiro tirar conclusões importantes sobre a dinâmica destes ‡uidos durante o processo de produção. Algumas informações obtidas com o auxílio da simulação numérica são a previsão do tempo de vida produtiva dos reservatórios e a avaliação do número de poços necessário para uma estratégia de produção otimizada. De fato, a simulação de reservatórios tem se tornado cada dia mais comum e essencial para as tomadas de decisão na indústria do petróleo (MATTAX e DALTON, 1990).

As equações utilizadas no problema de simulação são oriundas das equações que descrevem o ‡uxo multifásico de ‡uidos em meios porosos (MATTAX e DALTON, 1990; ERTEKIN et al., 2001). Tais equações são obtidas combinando-se a lei de conservação de massa, equações de estado e lei de Darcy, e aplicando-as a cada componente presente no sistema. A teoria de ‡uxo em meios porosos possui um campo de aplicação bastante abrangente, sendo empregada não só na engenharia de reservatórios como também na solução de problemas em diversas outras áreas da ciência e engenharia (BEAR, 1972) como modelagem de aquíferos, ciência do solo, mecânica dos solos, análises de ‡uxo de contaminantes, análises de ‡uxo sanguíneo no interior do corpo humano, etc.

Na engenharia de reservatórios, existem alguns modelos para o comportamento do ‡uxo de acordo com o fenômeno físico predominante no sistema, e estes modelos tornam-se mais complexos à medida que se deseja representar o sistema com mais detalhes. Alguns modelos de ‡uxo mais utilizados em simulações de reservatórios são o modelo Black-Oil, modelo composicional, modelos térmicos e modelos de ‡uxo miscível

(MATTAX e DALTON, 1990).

Neste trabalho são utilizados os conceitos estabelecidos no modelo Black-Oil convencional (MATTAX e DALTON, 1990; ERTEKIN et al., 2001), que considera o sistema com apenas três componentes - óleo, água e gás - representados pelos índices o; w e g, e três fases também designadas por óleo, água e gás. Assume-se que os três componentes são imiscíveis, o componente óleo só existe na fase óleo, o componente água só existe na fase água e o componente gás pode encontrar-se na fase gás, livre no reservatório, ou na fase óleo, associado ao componente óleo.

O modelo Black-Oil resulta em um sistema de equações diferenciais parciais não-lineares e dependentes no tempo, cuja solução geralmente só é possível empregando-se métodos numéricos de discretização apropriados. O processo de discretização numérica consiste em aproximar a geometria do reservatório através de uma malha e aplicar as equações do problema sobre os pontos discretos desta malha, a cada passo de tempo, resultando com isso em uma série de sistemas de equações lineares discretas, que são resolvidos empregando-se os métodos de solução de sistemas lineares adequados.

Atualmente existem diversos métodos numéricos de discretização utilizados em simulações de reservatórios, tais como: diferenças …nitas, elementos …nitos, volumes …nitos, streamlines, entre outros. O método das diferenças …nitas é o mais comum e, aparentemente, também o mais adequado para a solução deste tipo de problema, tendo sido adotado no desenvolvimento deste trabalho.

Para representar da melhor maneira possível as heterogeneidades existente em reservatórios reais é necessário usar malhas bastante re…nadas, o que resulta geralmente em modelos de larga escala, com grande quantidade de dados a serem armazenados em memória e elevado número de equações no sistema, sendo na maioria das vezes impossível resolvê-los utilizando os recursos computacionais disponíveis em um único microcomputador ou estação de trabalho atuais. Existem duas formas para se tratar estes problemas de larga escala: a primeira consiste em utilizar técnicas de upscaling, trabalhando-se com malhas menos re…nadas e tendo que levar em consideração as maiores incertezas envolvidas no processo de simulação; outra alternativa é utilizar a computação de alto desempenho para resolver o modelo mais re…nado. TCHELEPI et al. (1999) comparam aplicações destas duas alternativas em simulações de reservatórios heterogêneos reais, mostrando que, de fato, somente com o uso de computação de alto desempenho é possível resolver estes problemas e obter resultados satisfatórios em termos de vazões nos poços de produção.

Computadores paralelos têm sido a melhor alternativa para a solução de problemas

de larga escala em várias áreas da engenharia, possibilitando tratar modelos cada vez mais detalhados e representativos do problema real. Contudo, é fundamental que o código computacional tenha sido desenvolvido e implementado usando as técnicas específicas de programação paralela, lavando em consideração alguns aspectos relevantes como localização de dados, sincronização e balanceamento do trabalho computacional entre processadores. O código do simulador sequencial deve ser, portanto, reformulado e implementado visando tirar proveito das características específicas da arquitetura de cada máquina paralela.

Segundo KILLOUGH (1995) a paralelização de códigos de simuladores de reservatórios tiveram início em meados dos anos 70, com o surgimento dos primeiros computadores vetoriais, aparecendo somente no final dos anos 80 as primeiras aplicações em computadores paralelos com memória compartilhada e distribuída. A partir de então, várias aplicações em computadores paralelos ganharam espaço no meio científico, tornando-se cada vez mais comum sua utilização à medida que novas arquiteturas eram projetadas (RAMÉ e DELSHAD, 1995; SHIRALKAR et al., 1997; WANG et al., 1997; PARASHAR et al., 1997; ZHANG et al., 2001).

Como os computadores paralelos proprietários são normalmente muito caros, limitando seu uso a apenas algumas grandes empresas ou grandes centros de pesquisa, uma alternativa para o processamento paralelo a um custo menor é utilizar clusters de PCs (STERLING et al., 1995a) ou estações de trabalho. Com o acelerado avanço na tecnologia de microprocessadores, associado ao desenvolvimento de redes de conexão de alta velocidade, tem sido viável a utilização de clusters de PCs na solução de problemas de larga escala, apresentando uma relação custo-desempenho entre 3 a 10 vezes menor que os supercomputadores proprietários usuais. Atualmente, a tecnologia de clusters de PCs vem sendo utilizada em diversos projetos em diferentes áreas (BARRA et al., 1999; SILVA e RIVELLO, 2000; CAI e ODEGARD, 2001) , inclusive em simulações de reservatórios (ABATE et al., 1999, WANG et al., 1999).

Assim como qualquer computador paralelo com arquitetura de memória distribuída, a paralelização de códigos para aplicação em clusters de PCs exige o emprego de técnicas de programação específicas com o auxílio de softwares para controlar toda comunicação entre processadores. Isto exige do programador conhecimentos específicos sobre a utilização destes softwares de comunicação e detalhes sobre a paralelização dos algorítmos numéricos necessários à solução do problema.

1.2 Objetivos do Trabalho

O principal objetivo deste trabalho é examinar a possibilidade de desenvolver um simulador de reservatórios e…ciente para computadores paralelos com memória distribuída, utilizando apenas ferramentas de programação paralela de distribuição livre. Os requesitos de e…ciência desejados são escalabilidade e portabilidade. É usado o modelo de programação com memória distribuída onde toda comunicação entre processadores é realizada através da biblioteca de comunicação MPI - Message Passing Interface (MPI Forum, 1994; GROPP et al, 1999).

O código do simulador paralelo foi projetado aplicando-se o método de decomposição de domínio cuja idéia consiste em distribuir os dados da malha entre processadores de modo que cada um …que encarregado de efetuar operações sobre seus próprios dados locais. As soluções dos sistemas lineares ou não-lineares resultantes no simulador são obtidas através dos solvers do software PETSc - Portable, Extensible Toolkit for Scienti…c Computing (BALAY et al., 1997), que utiliza o MPI para a comunicação de dados entre processadores.

Devido ao tempo disponível para o desenvolvimento deste trabalho e à necessidade de dedicar parte do tempo ao estudo da computação paralela e algorítmos de paralelização, o simulador foi desenvolvido usando o modelo Black-Oil sem levar em conta o componente gás na formulação matemática, visando simpli…car a implementação computacional. Foi implementado no simulador o modelo Black-Oil bifásico óleo/água (ERTEKIN et al., 2001) visando explorar da melhor forma possível os detalhes de paralelização do problema. O sistema é considerado isotérmico, não levando em conta os efeitos de variação de temperatura no reservatório.

As equações que regem o problema são discretizadas usando o método das diferenças …nitas aplicado a uma malha bidimensional retangular com o esquema de blocos centrados. A linearização do sistema de equações discretas não-lineares e dependentes do tempo é realizada através das formulações IMPES (IMplicit Pressure, Explicit Saturation) e totalmente implícita, optando-se por uma das duas para a solução do problema.

São mostrados os detalhes envolvidos no desenvolvimento do simulador paralelo, descrevendo-se basicamente os modelos físico e matemático utilizados, o processo de discretização e formulações numéricas empregadas, alguns métodos utilizados na solução de sistemas de equações lineares, e o modelo computacional do simulador. São mostrados resultados de sua aplicação na solução de alguns problemas de ‡uxo em reservatórios homogêneos e heterogêneos, comparando estes resultados com os obtidos com o simulador comercial BOAST-98 (FANCHI et al., 1982). Finalmente, é realiza-

da a análise de desempenho do simulador em um computador paralelo com memória distribuída, especi...camente um cluster de PCs tipo NOW - network of workstations, obtendo-se medidas de speedup e e...ciência.

1.3 Organização dos Capítulos

Os demais capítulos deste trabalho estão organizados da seguinte forma.

No Capítulo 2 são descritos os detalhes da formulação matemática do problema, envolvendo as equações que regem o modelo Black-Oil bifásico óleo/água implementado no simulador.

A discretização das equações e as formulações numéricas empregadas na solução do sistema de equações discretas são tratados no Capítulo 3.

Detalhes sobre os métodos iterativos para solução de sistemas de equações lineares (SAAD, 1996; BARRET et al., 1994; TREFETHEN e BAU, 1997) são discutidos no Capítulo 4, onde são descritos alguns métodos estacionários e não-estacionários. Estes últimos compreendem os métodos baseados em subespaços de Krylov, cuja aplicação normalmente está associada a técnicas de precondicionamento necessárias para acelerar a convergência.

No Capítulo 5 são mostrados os conceitos básicos envolvidos na computação paralela (FOSTER, 1994; TOLEDO, 1997), descrevendo-se as principais classi...cações atribuídas a estes computadores, as diferentes formas de paralelização de códigos em arquiteturas paralelas com memória distribuída, as principais medidas de desempenho utilizadas, e uma descrição mais detalhada do modelo de programação com memória distribuída, sendo mostrado o funcionamento das bibliotecas MPI e PETSc.

O Capítulo 6 traz detalhes sobre a implementação do simulador paralelo, descrevendo-se a paralelização de cada etapa do processo de simulação e as principais subrotinas utilizadas.

O Capítulo 7 mostra aplicações para calibração e validação do modelo implementado no simulador, comparando-se os resultados do simulador com os do simulador BOAST-98 em um computador sequencial, e analisando o desempenho em um cluster de PCs.

No Capítulo 8 são encontradas as conclusões e recomendações para futuros trabalhos.

Capítulo 2

Formulação Matemática

Este capítulo descreve a formulação matemática necessária à implementação do simulador bifásico paralelo, sendo mostradas as equações básicas que, quando combinadas, resultam no sistema de equações diferenciais parciais, não-lineares e dependentes do tempo, que regem o problema de ‡uxo.

Para uma descrição da formulação matemática completa é necessário de…nir as equações de ‡uxo, relações constitutivas, condições inicias e condições de contorno. As seções seguintes descrevem estas equações.

2.1 Descrição das Equações Básicas

As equações de ‡uxo para o modelo Black-Oil óleo/água são obtidas a partir de equações básicas, que são: equações de estado, equação de conservação de massa e lei de Darcy, descritas resumidamente a seguir. DAKE (1978) e ERTEKIN et al.(2001) descrevem estas equações com mais detalhes.

2.1.1 Equações de Estado

As equações de estado descrevem o comportamento das propriedades do meio poroso e dos ‡uidos presentes no sistema. Geralmente estas propriedades podem variar em função das pressões e/ou saturações dos ‡uidos, sendo responsáveis pelo comportamento não-linear do sistema. Algumas destas propriedades importantes para o modelo bifásico óleo/água são descritas a seguir.

Propriedades da Rocha

As propriedades da rocha que interessam ao problema de ‡uxo em reservatórios de petróleo são a porosidade, Á; e a permeabilidade absoluta, k, que apresentam valores distintos para cada ponto, no caso de reservatórios heterogêneos.

A porosidade Á é de…nida em um determinado volume da rocha como sendo a relação entre o volume de vazios e o volume de rocha total considerado. Admitindo que o volume de vazios pode variar em função da pressão p no reservatório, tem-se a seguinte relação para expressar a variação de porosidade:

$$Á(p) = Á^{ref}\left[1 + c_r\left(p - p^{ref}\right)\right] \tag{2.1}$$

onde $Á^{ref}$ é a porosidade obtida em relação a uma pressão de referência p^{ref} e c_r é a compressibilidade da rocha.

Este efeito de variação da porosidade leva em conta somente a variação do volume de vazios da rocha, considerando as partículas sólidas incompressíveis. Em certos casos é importante considerar o efeito da deformação dos sólidos do meio poroso, devendo-se incorporar equações adicionais estabelecendo as relações entre deformações e porosidade (BEAR, 1972).

A permeabilidade absoluta k representa uma medida da facilidade apresentada pelo meio para o ‡uido escoar através de seus poros, e é representada através de um tensor de permeabilidades absolutas K, que contém os valores de permeabilidades segundo as direções x; y e z; admitindo um sistema de coordenadas cartesiano, mostrado abaixo:

$$K = \begin{bmatrix} k_x & k_{xy} & k_{xz} \\ k_{yx} & k_y & k_{yz} \\ k_{zx} & k_{zy} & k_z \end{bmatrix} \tag{2.2}$$

Em simulação de reservatórios, é comum admitir que as direções preferenciais de ‡uxo coincidem com as direções principais do tensor de permeabilidades absolutas, podendo portanto ser substituido por um tensor diagonal representado por:

$$K = \begin{bmatrix} k_x & & \\ & k_y & \\ & & k_z \end{bmatrix} \tag{2.3}$$

Esta hipótese de ‡uxo segundo as direções principais nem sempre ocorre, trazendo alguns problemas nos resultados da simulação, dependendo da orientação da malha utilizada na discretização do modelo (MATTAX e DALTON, 1990).

Admite-se que não há variação das permeabilidades absolutas durante o processo de simulação.

Propriedades dos Fluidos

Algumas propriedades dos ‡uidos são importantes para a avaliação do comportamento do ‡uxo nas condições especí…cas dos reservatórios. Estas propriedades são: compressibilidade, fator volume de formação, densidade e viscosidade. Estas propriedades são intrínsecas ao ‡uido e podem variar com a pressão e temperatura no reservatório.

Para o ‡uido l, a compressibilidade c_l é de…nida como uma medida da variação do volume do ‡uido em relação à pressão. Admitindo uma temperatura constante no reservatório, tem-se as relações:

$$c_l = -\frac{1}{V_l}\frac{@V_l}{@p_l} \tag{2.4}$$

$$c_l = \frac{1}{\rho_l}\frac{@\rho_l}{@p_l} \tag{2.5}$$

onde V_l e ρ_l são o volume e a densidade em condições de reservatório, respectivamente.

Uma de…nição bastante utilizada no modelo Black-Oil é o fator volume de formação B_l, representado pela relação entre o volume do ‡uido em condições de reservatório, V_l ;sobre o volume do ‡uido em condições padrão, V_{lsc}. Geralmente, admite-se para a condição padrão uma pressão de 14.7 psi e uma temperatura de 60^{o}F. Tem-se, portanto, as seguintes relações para o fator volume de formação:

$$B_l = \frac{V_l}{V_{lsc}} \tag{2.6}$$

Para ‡uidos levemente compressíveis, como água e óleo pesado, a expressão para a variação do fator volume de formação com a pressão no reservatório pode ser obtida integrando-se a eq. (2.5), admitindo uma temperatura constante, resultando em:

$$B_l(p_l) = \frac{B_l^{ref}}{[1 + c_l(p_l - p^{ref})]} \tag{2.7}$$

onde B_l^{ref} é o fator volume de formação correspondente a uma pressão de referência p^{ref}, normalmente igual a 14.7 psi:

A densidade ou massa especí…ca também varia com a pressão. Partindo da eq. (2.6), a densidade em condições de reservatório pode ser escrita em função da densidade em condições padrão ρ_{lsc}:

$$\rho_l = \frac{\rho_{lsc}}{B_l} \tag{2.8}$$

Utilizando as eqs. (2.7) e (2.8), a variação da densidade em função da pressão no reservatório pode ser expressa por:

$$\rho_l(p_l) = \rho_l^{ref}\left[1 + c_l\left(p_l - p^{ref}\right)\right] \tag{2.9}$$

onde ρ_l^{ref} é a densidade de referência a uma pressão p^{ref}.

Outra propriedade importante é a viscosidade μ_l, que representa a dificuldade que o fluido apresenta para escoar quando submetido a um gradiente de pressão. Admitindo sistemas de reservatórios com temperatura constante, para a água e o óleo pesado a variação da viscosidade é muito pequena, crescendo à medida que a pressão aumenta.

Propriedades Rocha-Fluido

Algumas propriedades variam em função de características conjuntas da rocha e do fluido considerado, como saturação S_l, permeabilidade relativa k_{rl} e pressão de capilaridade P_c, descritas a seguir.

A saturação S_l em um sistema multifásico representa o percentual do volume poroso ocupado pelo fluido, calculada pela relação entre o volume ocupado em condições de reservatório e o volume de vazios. Em sistemas bifásicos óleo/água, tem-se a seguinte equação de restrição das saturações:

$$S_o + S_w = 1 \tag{2.10}$$

A permeabilidade relativa $k_{rl}(S_w)$ representa um fator de redução da permeabilidade absoluta devido à presença de outros fluidos no espaço poroso. Em sistemas óleo/água os valores das permeabilidades para cada componente variam em função da saturação de água.

A pressão de capilaridade P_c consiste na diferença de pressão entre a fase não-molhada e a fase molhada. Neste caso a fase não-molhada é representada pelo componente óleo e a fase molhada pelo componente água, resultando em:

$$P_{cow}(S_w) = p_o - p_w \tag{2.11}$$

Esta propriedade varia com a saturação de água e, juntamente com as permeabilidades relativas, é responsável pelo elevado grau de não-linearidade atribuído ao problema.

2.1.2 Lei de Darcy

A lei de Darcy (BEAR, 1972; DAKE, 1978) é uma expressão empírica para descrever o fluxo de fluidos através de um meio poroso, estabelecendo uma relação entre a velocidade superficial e o gradiente do potencial. O potencial Φ_l é definido pela fórmula abaixo:

$$\Phi_l = p_l - \gamma_l Z \tag{2.12}$$

onde p_l é a pressão a uma profundidade Z; calculada em relação a um ponto de referência e positiva para baixo, $°_l$ é o peso específico do fluido definido por $°_l = ½_l g$, onde g é a aceleração da gravidade e $½_l$ é a densidade. Assumindo um fluxo multifásico unidimensional, a velocidade superficial, ou velocidade de Darcy, segundo a direção x pode ser escrita para o fluido l através da expressão:

$$u_{lx} = - \frac{k_x k_{rl}}{¹_l} \frac{@©_l}{@x} \tag{2.13}$$

onde k_x é a permeabilidade absoluta na direção x, $¹_l$ é a viscosidade e k_{rl} é a permeabilidade relativa.

Substituindo a eq. (2.12) na eq. (2.13), tem-se:

$$u_{lx} = - \frac{k_x k_{rl}}{¹_l} \left(\frac{@p_l}{@x} - °_l \frac{@Z}{@x} \right) \tag{2.14}$$

Na equação acima a densidade é considerada constante, embora seja sempre atualizada em função da pressão calculada no passo de tempo anterior, durante o processo de simulação. Esta consideração pode ser aplicada também a fluidos levemente compressíveis (ERTEKIN et al., 2001).

No caso multidimensional, admitindo um sistema de coordenadas cartesiano, a eq. (2.13) pode ser escrita da seguinte forma:

$$u_l = - K \frac{k_{rl}}{¹_l} r ©_l \tag{2.15}$$

onde u_l é o vetor de velocidades de Darcy, composto pelos componentes de velocidades nas direções x, y e z, e K é o tensor diagonal de permeabilidades absolutas representado por (2.3).

2.1.3 Lei da Conservação de Massa

A lei da conservação de massa, ou equação de continuidade, estabelece o balanço de massa para um determinado fluido, tomando como referência um elemento de volume de controle. Nesta seção, a expressão que estabelece o balanço de massa é deduzida utilizando uma abordagem física do problema, destacando-se que a mesma expressão também pode ser deduzida através de uma análise integral para estabelecer o equilíbrio de massa no elemento de controle (BEAR, 1972).

Para o sistema de coordenadas cartesiano, o elemento de controle representado pela Figura 2.1 é utilizado na dedução das equações de balanço de massa, assumindo que o volume do elemento não varia com o tempo.

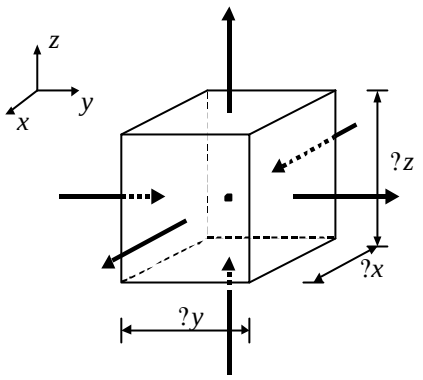

Figura 2.1: Elemento de volume de controle.

Admitindo que o ‡uxo de massa entra ou sai do elemento, paralelo aos eixos do sistema cartesiano, através das faces do elemento, têm-se a seguinte expressão para o balanço de massa do componente l:

$$(m_e)_l \; ¡ \; (m_s)_l + (m_f)_l = (m_a)_l \tag{2.16}$$

onde

$(m_e)_l$ - massa que entra no elemento através das faces;

$(m_s)_l$ - massa que sai do elemento através das faces;

$(m_f)_l$ - massa que entra no elemento devido a uma fonte no seu interior;

$(m_a)_l$ - massa acumulada no elemento durante um intervalo de tempo ¢t:

No lado esquerdo da eq. (2.16), os termos de massa que entra e sai do elemento por suas faces e o termo de massa devido à fonte localizada no seu interior, são chamados termos de ‡uxo e termo de fonte, respectivamente, enquanto o termo do lado direito é chamado termo de acumulação.

Os termos de ‡uxo podem ser escritos, em relação ao ponto central de coordenadas $(x; y; z)$; da seguinte forma:

$$\begin{aligned}(m_e)_l &= \left[(\underline{m}_{l_x} A_x)_{x\,¡\,¢x=2} + \left[\underline{m}_{l_y} A_y\right]_{y\,¡\,¢y=2} + (\underline{m}_{l_z} A_z)_{z\,¡\,¢z=2}\right] ¢t \\ (m_s)_l &= \left[(\underline{m}_{l_x} A_x)_{x+¢x=2} + \left[\underline{m}_{l_y} A_y\right]_{y+¢y=2} + (\underline{m}_{l_z} A_z)_{z+¢z=2}\right] ¢t \end{aligned} \tag{2.17}$$

onde

$A_x; A_y; A_z$ são as áreas das faces do elemento segundo as direções $x; y$ e z respectivamente;

$(\underline{m}_{l_x})_{x \pm \Delta x/2}$ representa o fluxo de massa do componente l por unidade de área que entra ou sai do elemento pela face $x \pm \Delta x/2$, podendo ser definido como:

$$\underline{m}_{l_x} = \rho_l u_{l_x} \tag{2.18}$$

onde ρ_l é a densidade do componente l e u_{lx} é a velocidade de Darcy, que representa uma vazão por unidade de área. Definição análoga vale para o fluxo de massa segundo as direções y e z.

Definindo q_{m_l} como sendo o fluxo de massa do componente l que entra no elemento devido à fonte localizada no seu interior, tem-se:

$$(m_f)_l = q_{m_l} \Delta t \tag{2.19}$$

Escrevendo o fluxo de massa q_{m_l} em termos da vazão q_l, dada em volume por unidade de tempo, tem-se:

$$q_{m_l} = \rho_l q_l \tag{2.20}$$

O termo de acumulação representa a variação da massa acumulada entre os tempos t e $t + \Delta t$, sendo escrito por:

$$(m_a)_l = (m_{v_l})^{t+\Delta t} - (m_{v_l})^t \tag{2.21}$$

onde $m_{v_l} = S_l \phi \rho_l V_b$ é a massa do componente l contida no elemento, $V_b = \Delta x \Delta y \Delta z$ é o volume do elemento e S_l é a saturação do componente, resultando em:

$$(m_a)_l = (S_l \phi \rho_l \Delta x \Delta y \Delta z)^{t+\Delta t} - (S_l \phi \rho_l \Delta x \Delta y \Delta z)^t \tag{2.22}$$

Substituindo as eqs. (2.17), (2.19) e (2.22) na eq. (2.16), rearrumando os termos e dividindo o resultado pelo volume do elemento, tem-se:

$$-\left[\frac{(\underline{m}_{l_x})_{x+\Delta x/2} - (\underline{m}_{l_x})_{x-\Delta x/2}}{\Delta x}\right] - \left[\frac{\underline{m}_{l_y}{}_{y+\Delta y/2} - \underline{m}_{l_y}{}_{y-\Delta y/2}}{\Delta y}\right]$$

$$-\left[\frac{(\underline{m}_{l_z})_{z+\Delta z/2} - (\underline{m}_{l_z})_{z-\Delta z/2}}{\Delta z}\right] + \frac{q_{m_l}}{V_b} = \left[\frac{(S_l \phi \rho_l)^{t+\Delta t} - (S_l \phi \rho_l)^t}{\Delta t}\right] \tag{2.23}$$

Escrevendo a equação acima em termos de derivadas parciais e substituindo os termos de fluxo de massa pelas eqs. (2.18), deduzida para cada direção, e (2.20), tem-se:

$$-\frac{\partial}{\partial x}(\rho_l u_{l_x}) - \frac{\partial}{\partial y}\rho_l u_{l_y} - \frac{\partial}{\partial z}(\rho_l u_{l_z}) + \frac{\rho_l q_l}{V_b} = \frac{\partial}{\partial t}(S_l \phi \rho_l) \tag{2.24}$$

ou

$$-\frac{\partial}{\partial x}(\rho_l u_{l_x} A_x)\,\Delta x - \frac{\partial}{\partial y}\left[\rho_l u_{l_y} A_y\right]\Delta y - \frac{\partial}{\partial z}(\rho_l u_{l_z} A_z)\,\Delta z + \rho_l q_l = V_b \frac{\partial}{\partial t}(S_l \phi \rho_l) \quad (2.25)$$

A eq. (2.25) representa a forma final da equação de balanço de massa para o componente l, admitindo um sistema de eixos cartesiano. Esta equação pode ser obtida em relação a outros sistemas de eixos, tornando-se mais conveniente de acordo com as características do problema que se deseja resolver (ERTEKIN et al., 2001).

2.2 Equações de Fluxo e Relações Constitutivas

Tendo estabelecido as equações básicas para o problema, a forma final das equações que regem o fluxo bifásico óleo/água são obtidas combinando-se estas equações. Considerando a densidade dos componentes em condições padrão, ρ_{lsc}, constante e substituindo a eq. (2.8) e a eq. (2.14), escrita para as direções y e z, na eq. (2.25), chega-se a:

$$\frac{\partial}{\partial x}\left[A_x k_x \frac{k_{rl}}{\mu_l B_l}\left(\frac{\partial p_l}{\partial x} - \gamma_l \frac{\partial Z}{\partial x}\right)\right]\Delta x + \frac{\partial}{\partial y}\left[A_y k_y \frac{k_{rl}}{\mu_l B_l}\left(\frac{\partial p_l}{\partial y} - \gamma_l \frac{\partial Z}{\partial y}\right)\right]\Delta y +$$
$$\frac{\partial}{\partial z}\left[A_z k_z \frac{k_{rl}}{\mu_l B_l}\left(\frac{\partial p_l}{\partial z} - \gamma_l \frac{\partial Z}{\partial z}\right)\right]\Delta z = V_b \frac{\partial}{\partial t}\left(\frac{S_l \phi}{B_l}\right) - q_{lsc} \qquad l = o, w \quad (2.26)$$

onde, $q_{lsc} = q_l / B_l$ é a vazão do termo de fonte ou termo de poço em condições padrão.

A eq. (2.26) representa o sistema de equações que rege o problema de fluxo tridimensional para o modelo Black-Oil bifásico óleo/água em reservatórios, usando um sistema de coordenadas cartesiano. O sistema é composto por duas equações diferenciais parciais dependentes do tempo, uma para cada componente, cujas incógnitas são as pressões e saturações de cada componente, no caso p_o; p_w; S_w e S_o. Como o número de incógnitas é maior que o número de equações, é necessário introduzir algumas relações constitutivas para que seja possível resolver o problema.

As relações constitutivas que relacionam as incógnitas da eq. (2.26) são a equação de restrição das saturações (2.10) e a equação da pressão de capilaridade (2.11), reescritas abaixo:

$$S_o + S_w = 1 \quad (2.27)$$

$$P_{cow}(S_w) = p_o - p_w \quad (2.28)$$

Usando estas equações, é possível rescrever o sistema (2.26) eliminando duas incógnitas e deixando-o apenas em função das duas variáveis primárias, pressão de óleo p_o e

saturação de água S_w; resultando no sistema de equações diferenciais …nal com duas equações e duas incógnitas:

$$\begin{aligned}
&\frac{\partial}{\partial x}\left[A_x k_x \frac{k_{ro}}{\mu_o B_o}\left(\frac{\partial p_o}{\partial x} - \gamma_o \frac{\partial Z}{\partial x}\right)\right]\Delta x + \\
&\frac{\partial}{\partial y}\left[A_y k_y \frac{k_{ro}}{\mu_o B_o}\left(\frac{\partial p_o}{\partial y} - \gamma_o \frac{\partial Z}{\partial y}\right)\right]\Delta y + \\
&\frac{\partial}{\partial z}\left[A_z k_z \frac{k_{ro}}{\mu_o B_o}\left(\frac{\partial p_o}{\partial z} - \gamma_o \frac{\partial Z}{\partial z}\right)\right]\Delta z = \\
&V_b \frac{\partial}{\partial t}\left[\frac{(1 - S_w)\phi}{B_o}\right] - q_{osc}
\end{aligned}$$

$$\begin{aligned}
&\frac{\partial}{\partial x}\left[A_x k_x \frac{k_{rw}}{\mu_w B_w}\left(\frac{\partial p_o}{\partial x} - \frac{\partial P_{cow}}{\partial x} - \gamma_w \frac{\partial Z}{\partial x}\right)\right]\Delta x + \\
&\frac{\partial}{\partial y}\left[A_y k_y \frac{k_{rw}}{\mu_w B_w}\left(\frac{\partial p_o}{\partial y} - \frac{\partial P_{cow}}{\partial y} - \gamma_w \frac{\partial Z}{\partial y}\right)\right]\Delta y + \\
&\frac{\partial}{\partial z}\left[A_z k_z r_w \mu_w B_w\left(\frac{\partial p_o}{\partial z} - \frac{\partial P_{cow}}{\partial z} - \gamma_w \frac{\partial Z}{\partial z}\right)\right]\Delta z = \\
&V_b \frac{\partial}{\partial t}\left[\frac{S_w \phi}{B_w}\right] - q_{wsc}
\end{aligned} \tag{2.29}$$

2.3 Condições de Contorno e Condições Iniciais

Para que o sistema (2.29) apresente solução única, é necessário especi…car as condições de contorno e condições iniciais do problema.

2.3.1 Condições de Contorno

As condições de contorno envolvem as condições prescritas nas fronteiras do reservatório e as condições de poços, também chamadas condições de contorno externas e condições de contorno internas, respectivamente.

As codições de contorno externas podem ser fornecidas através de um ‡uxo prescrito (condição de Neumann) ou através de uma pressão prescrita (condição de Dirichlet) nas fronteiras. Em reservatórios de petróleo, normalmente admite-se que o ‡uxo nas fronteiras é nulo, sendo comum encontrar esta condição implementada como padrão na maioria dos simuladores, tendo sido adotada também neste trabalho.

As condições de poços são especi…cadas fornecendo-se vazões ou pressões prescritas nos poços, incluídas nas equações de ‡uxo através dos termos de fonte. No modelo

considerado, os poços classi...cados como poços de produção podem produzir óleo e água ao mesmo tempo, enquanto os poços de injeção apenas injetam água no reservatório.

2.3.2 Condições Iniciais

As condições iniciais são introduzidas com o fornecimento dos campos de pressões e saturações no reservatório no tempo inicial, representadas pelas variáveis primárias, pressões de óleo p_o ($t = 0$) e saturações de água S_w ($t = 0$).

Capítulo 3

Discretização e Formulação Numérica

Na grande maioria dos casos de simulação de reservatórios, o sistema de equações diferenciais parciais (2.29) não apresenta uma solução analítica, sendo necessário o emprego de técnicas de discretização numérica para obter uma solução aproximada para o problema.

Neste capítulo, o sistema (2.29) é discretizado usando o método das diferenças …nitas para aproximar as derivadas no espaço e no tempo, obtendo-se então um sistema de equações discretas não-lineares e dependentes no tempo. Em seguida o sistema discreto é linearizado usando a formulação IMPES ou a totalmente implícita, ambas implementadas no simulador.

3.1 Método das Diferenças Finitas

O método das diferenças …nitas (ZIENKIEWICS e MORGAN, 1982) é bastante utilizado em simuladores de reservatórios (MATTAX e DALTON, 1990; ERTEKIN et al., 2001) e baseia-se em expansões por Séries de Taylor para aproximar as derivadas no espaço e no tempo das equações de ‡uxo. Seja uma função qualquer dependente do espaço e do tempo $f(x;t)$. As expansões em Series de Taylor aplicadas a esta função nos pontos $f(x + ¢x;t)$ e $f(x;t + ¢t)$; variando-se as variáveis x e t independentemente uma da outra, podem ser escritas em termos das derivadas em relação a cada uma destas variáveis, segundo as expressões abaixo:

$$f(x + ¢x;t) = f(x;t) + \frac{¢x}{1!}\frac{@f(x;t)}{@x} + \frac{¢x^2}{2!}\frac{@^2f(x;t)}{@x^2} + \frac{¢x^3}{3!}\frac{@^3f(x;t)}{@x^3} + O\left(¢x^4\right) \quad (3.1)$$

$$f(x;t+\Delta t) = f(x;t) + \frac{\Delta t}{1!}\frac{\partial f(x;t)}{\partial t} + \frac{\Delta t^2}{2!}\frac{\partial^2 f(x;t)}{\partial t^2} + \frac{\Delta t^3}{3!}\frac{\partial^3 f(x;t)}{\partial t^3} + O\left(\Delta t^4\right) \quad (3.2)$$

Em problemas de reservatórios existem dois níveis de discretização: discretização no espaço e no tempo. A seguir são obtidas as expressões gerais para a aproximação das derivadas no espaço e no tempo usadas nas equações de fluxo.

3.1.1 Aproximação das Derivadas no Espaço

Para aproximar as derivadas no espaço é necessário definir uma malha de discretização com o objetivo de tratar o eixo coordenado contínuo através de pontos isolados. Existem dois esquemas mais usados na discretização da geometria dos reservatórios, que são os esquemas de pontos centrados e os esquemas de blocos centrados (MATTAX e DALTON, 1990; ERTEKIN et al., 2001). Neste trabalho é utilizado o esquema de blocos centrados, onde a malha é dividida em blocos e cada bloco possui um ponto central para representar as propriedades médias no bloco. O esquema de blocos centrados está mostrado na Figura 3.1 para uma malha de discretização unidimensional.

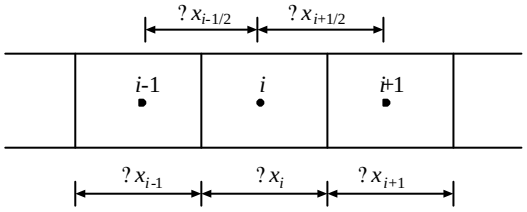

Figura 3.1: Discretização unidimensional usando o esquema de blocos centrados.

Na eq. (2.29), o termo de fluxo na direção x é formado pelas seguintes parcelas:

- componente óleo:

$$\frac{\partial}{\partial x}\left[A_x k_x \frac{k_{ro}}{\mu_o B_o}\frac{\partial p_o}{\partial x}\right]\Delta x \quad \text{e} \quad \frac{\partial}{\partial x}\left[A_x k_x \frac{k_{ro}}{\mu_o B_o}\gamma_o \frac{\partial Z}{\partial x}\right]\Delta x$$

- componente água:

$$\frac{\partial}{\partial x}\left[A_x k_x \frac{k_{rw}}{\mu_w B_w}\frac{\partial p_o}{\partial x}\right]\Delta x; \quad \frac{\partial}{\partial x}\left[A_x k_x \frac{k_{rw}}{\mu_w B_w}\frac{\partial P_{cow}}{\partial x}\right]\Delta x$$
$$\text{e} \quad \frac{\partial}{\partial x}\left[A_x k_x \frac{k_{rw}}{\mu_w B_w}\gamma_w \frac{\partial Z}{\partial x}\right]\Delta x$$

Observa-se que todas as parcelas mostradas acima são da forma $\frac{\partial}{\partial x}\left[f(x;t)\frac{\partial v}{\partial x}\right]\Delta x$, considerando v uma variável qualquer e f uma função que varia no espaço e no tempo.

Admitindo a malha unidimensional da Figura 3.1, uma aproximação para este termo pode ser obtida a apartir da eq. (3.1), usando uma aproximação por diferença central aplicada ao bloco i e desprezando o erro de discretização, resultando em:

$$\frac{@}{@x}\left[\Gamma(x;t)\frac{@v}{@x}\right]_i \Delta x_i = \left[\Gamma(x;t)\frac{@v}{@x}\right]_{i+1=2} - \left[\Gamma(x;t)\frac{@v}{@x}\right]_{i-1=2} \tag{3.3}$$

Os índices $i+1=2$ e $i-1=2$ referem-se a valores nas faces do bloco i. Desprezando o erro de discretização e aplicando aproximação por diferenças centrais nos pontos $i+1=2$ e $i-1=2$; chega-se à seguinte expressão para a primeira derivada:

$$\left(\frac{@v}{@x}\right)_{i\pm1=2} = \pm\frac{v_{i\pm1} - v_i}{\Delta x_{i\pm1=2}} \tag{3.4}$$

Substituindo a eq. (3.4) na eq. (3.3) chega-se à expressão geral usada para aproximar os termos de fluxo:

$$\frac{@}{@x}\left[\Gamma(x;t)\frac{@v}{@x}\right]_i \Delta x_i = \left[\frac{\Gamma(x;t)}{\Delta x}\right]^{i+1=2}(v_{i+1} - v_i) + \left[\frac{\Gamma(x;t)}{\Delta x}\right]^{i-1=2}(v_{i-1} - v_i) \tag{3.5}$$

A expressão (3.5) pode ser representada de forma compacta, definindo-se o seguinte operador:

$$\Delta_x[g(\Delta_x v)] = g_{x_{i+1=2}}(v_{i+1} - v_i) + g_{x_{i-1=2}}(v_{i-1} - v_i) \tag{3.6}$$

Expressões similares podem ser obtidas para os termos nas direções y e z.

3.1.2 Aproximação das Derivadas no Tempo

A discretização no tempo consiste em dividir o domínio do tempo em intervalos, onde em cada passo de tempo, são calculadas as incógnitas do problema. Do mesmo modo como foi feito para aproximar a derivada no espaço, partindo da eq. (3.2) e desprezando o erro de discretização é possível deduzir uma aproximação à esquerda para a primeira derivada no tempo atual $t + \Delta t$, dada por:

$$\frac{@f(x;t+\Delta t)}{@t} = \frac{f(x;t+\Delta t) - f(x;t)}{\Delta t} \tag{3.7}$$

Associando o passo de tempo $n+1$ ao tempo $t + \Delta t$; e n ao tempo t, a aproximação acima fica escrita de forma compacta como:

$$\left(\frac{@f}{@t}\right)^{n+1} = \frac{f^{n+1} - f^n}{\Delta t} = \frac{\Delta_t f}{\Delta t} \tag{3.8}$$

3.2 Discretização das Equações de Fluxo

Tendo desenvolvido as expressões aproximadas para as derivadas no espaço (3.5) e no tempo (3.8), é possível aplicar estas expressões às parcelas dos termos das equações de ‡uxo, obtendo-se assim o sistema de equações discretas resultante da discretização do sistema diferencial (2.29) usando aproximações por diferenças …nitas.

Para uma malha bidimensional como a ilustrada na Figura 3.2, e admitindo que existem n_x blocos na direção x e n_y blocos na direção y; o sistema de equações discreto para o bloco (i; j) em um passo de tempo n + 1; pode ser escrito a partir do sistema (2.29), resultando em:

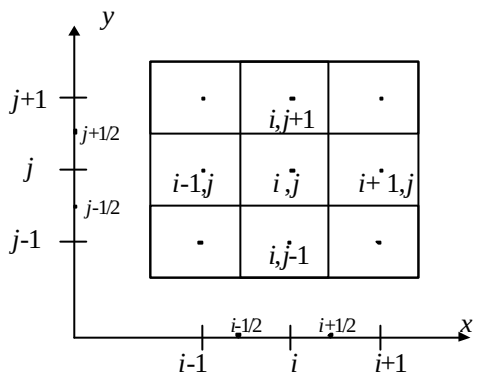

Figura 3.2: Esquema de discretização com 5 pontos - modelo bidimensional.

$$\frac{\partial}{\partial x}\left[A_x k_x \frac{k_{ro}}{\mu_o B_o}\left(\frac{\partial p_o}{\partial x} - \gamma_o \frac{\partial Z}{\partial x}\right)\right]^{n+1}_{i,j} \Delta x_{i,j} + \frac{\partial}{\partial y}\left[A_y k_y \frac{k_{ro}}{\mu_o B_o}\left(\frac{\partial p_o}{\partial y} - \gamma_o \frac{\partial Z}{\partial y}\right)\right]^{n+1}_{i,j} \Delta y_{i,j} = V_{b_{i,j}} \left[\frac{\partial}{\partial t}\left(\frac{(1 - S_w)\phi}{B_o}\right)\right]^{n+1}_{i,j} - q^{n+1}_{osc_{i,j}}$$

$$\frac{\partial}{\partial x}\left[A_x k_x \frac{k_{rw}}{\mu_w B_w}\left(\frac{\partial p_o}{\partial x} - \frac{\partial P_{cow}}{\partial x} - \gamma_w \frac{\partial Z}{\partial x}\right)\right]^{n+1}_{i,j} \Delta x_{i,j} + \frac{\partial}{\partial y}\left[A_y k_y \frac{k_{rw}}{\mu_w B_w}\left(\frac{\partial p_o}{\partial y} - \frac{\partial P_{cow}}{\partial y} - \gamma_w \frac{\partial Z}{\partial y}\right)\right]^{n+1}_{i,j} \Delta y_{i,j} = V_{b_{i,j}} \left[\frac{\partial}{\partial t}\left(\frac{S_w \phi}{B_w}\right)\right]^{n+1}_{i,j} - q^{n+1}_{wsc_{i,j}}$$

$$i = 1;:::;n_x \qquad j = 1;:::;n_y \qquad n = 0;1;2;::: \tag{3.9}$$

A seguir são desenvolvidas as aproximações para os termos de ‡uxo, termos de acumulação e termos de poço do sistema (3.9).

3.2.1 Termos de Fluxo

Partindo da eq. (3.5) e ocultando temporariamente o índice do passo de tempo, as parcelas que formam os termos de ‡uxo na direção x no sistema discreto (3.9), são expressas pelas seguintes equações para os componentes $l = o;w$:

$$\begin{aligned}\frac{@}{@x}\left(A_x k_x \frac{k_{rl}}{{}^1_l B_l}\frac{@p_o}{@x}\right)_{i;j} ¢x_{i;j} &= \left(\frac{A_x k_x}{¢x}\frac{k_{rl}}{{}^1_l B_l}\right)_{i+1=2;j}\left(p_{o_{i+1;j}} - p_{o_{i;j}}\right) + \\ &\quad \left(\frac{A_x k_x}{¢x}\frac{k_{rl}}{{}^1_l B_l}\right)_{i-1=2;j}\left(p_{o_{i-1;j}} - p_{o_{i;j}}\right) \\ &= T_{lx_{i+1=2;j}}\left(p_{o_{i+1;j}} - p_{o_{i;j}}\right) + \\ &\quad T_{lx_{i-1=2;j}}\left(p_{o_{i-1;j}} - p_{o_{i;j}}\right)\end{aligned} \tag{3.10}$$

$$\begin{aligned}\frac{@}{@x}\left(A_x k_x \frac{k_{rl}}{{}^1_l B_l}°_l\frac{@Z}{@x}\right)_{i;j} ¢x_{i;j} &= T_{lx_{i+1=2;j}}°_{l_{i+1=2;j}}\left(Z_{i+1;j} - Z_{i;j}\right) + \\ &\quad T_{lx_{i-1=2;j}}°_{l_{i-1=2;j}}\left(Z_{i-1;j} - Z_{i;j}\right)\end{aligned} \tag{3.11}$$

$$\begin{aligned}\frac{@}{@x}\left(A_x k_x \frac{k_{rw}}{{}^1_w B_w}\frac{@P_{cow}}{@x}\right)_{i;j} ¢x_{i;j} &= T_{wx_{i+1=2;j}}\left(P_{cow_{i+1;j}} - P_{cow_{i;j}}\right) + \\ &\quad T_{wx_{i-1=2;j}}\left(P_{cow_{i-1;j}} - P_{cow_{i;j}}\right)\end{aligned} \tag{3.12}$$

onde,

$$°_{l_{i§1=2;j}} = g\frac{½l_{i§1;j} + ½l_{i;j}}{2} \tag{3.13}$$

$$T_{lx_{i§1=2;j}} = \left(\frac{A_x k_x}{¢x}\right)_{i§1=2;j}\left(\frac{k_{rl}}{{}^1_l B_l}\right)_{i§1=2;j} = G_{x_{i§1=2;j}}, l_{i§1=2;j} \tag{3.14}$$

A eq. (3.13) representa o peso especí…co médio entre blocos, sendo calculado em função da média aritmética das densidades nos blocos.

Na equação (3.14), $T_{lx_{i§1=2;j}}$ é a transmissibilidade do componente l nas faces $i§1=2$ do bloco $(i;j)$, de…nida através do produto de um fator geométrico $G_{x_{i§1=2;j}}$ pela mobilidade do ‡uido $, l_{i§1=2;j}$. Como as transmissibilidades são propriedades calculadas nas faces dos blocos, seus valores são valores médios calculados em função de propriedades dos blocos vizinhos.

Pode-se deduzir uma expressão para o cálculo do fator geométrico (ERTEKIN et al., 2001) dada pela média harmônica das propriedades geométricas dos blocos, cuja fórmula é:

$$G_{x_{i\pm1=2;j}} = \frac{2A_{x_{i;j}}A_{x_{i\pm1;j}}k_{x_{i;j}}k_{x_{i\pm1;j}}}{A_{x_{i;j}}k_{x_{i;j}}¢x_{i\pm1;j} + A_{x_{i\pm1;j}}k_{x_{i\pm1;j}}¢x_{i;j}} \tag{3.15}$$

Para as mobilidades nas faces do bloco $¸_{l_{i\pm1=2;j}}$, está sendo adotado o modelo de ponderação com 1 ponto a montante (upstream weighting), que consiste em calcular a mobilidade média entre blocos como sendo a mobilidade do bloco que possuir o maior potencial gravitacional, ou seja:

$$¸_{l_{i\pm1=2;j}} = \begin{cases} ¸_{l_{i\pm1;j}} & \text{se} \quad ©_{l_{i\pm1;j}} ¸ ©_{l_{i;j}} \\ ¸_{l_{i;j}} & \text{se} \quad ©_{l_{i\pm1;j}} < ©_{l_{i;j}} \end{cases} \tag{3.16}$$

$$¸_{l_{i;j}} = (k_{rl}=¹_l B_l)_{i;j} \tag{3.17}$$

onde $¸_{l_{i;j}}$ é a mobilidade do ‡uido no bloco $(i;j)$ e $©_l$ é o potencial gravitacional. Existem outras formas para o cálculo das mobilidades médias entre blocos, entre elas o modelo de ponderação com 2 pontos a montante (ERTEKIN et al., 2001), ou a própria média aritmética entre mobilidades dos blocos. MATTAX e DALTON (1990) mostram que esta última técnica não é adequada para sistemas multifásicos, fornecendo resultados pouco acurados. Em sistemas unifásicos, o uso da média aritmética não apresenta os mesmos problemas, podendo ser utilizada com e…ciência.

Usando o operador de forma compacta (3.6), os termos de ‡uxo na direção x são escritos por:

$$\frac{@}{@x}\left[A_x k_x \frac{k_{ro}}{¹_o B_o}\left(\frac{@p_o}{@x} ¡ °_o\frac{@Z}{@x}\right)\right]_{i;j} ¢x_{i;j} = ¢_x[T_o(¢_x p_o ¡ ²_o ¢_x Z)] \tag{3.18}$$

$$\frac{@}{@x}\left[A_x k_x \frac{k_{rw}}{¹_w B_w}\left(\frac{@p_o}{@x} ¡ \frac{@P_{cow}}{@x} ¡ °_w\frac{@Z}{@x}\right)\right]_{i;j} ¢x_{i;j} = ¢_x[T_w(¢_x p_o ¡ ¢_x P_{cow} ¡ ²_w ¢_x Z)] \tag{3.19}$$

onde $²_l$ representa os pesos especí…cos médios entre blocos. Os termos de ‡uxo na direção y são desenvolvidos de forma similar.

3.2.2 Termos de Acumulação

Os termos de acumulação do sistema (3.9) são desenvolvidos usando a aproximação por diferença à esquerda para as derivadas no tempo, eq. (3.8), assumindo que a derivada é calculada no passo de tempo atual $n+1$, onde n refere-se ao passo de tempo anterior.

A forma discreta do termo de acumulação é escrita para o bloco $(i;j)$ por:

$$V_{b_{i;j}} \left[\frac{@}{@t} \left(\frac{S_l Á}{B_l} \right) \right]^{n+1}_{i;j} = \frac{V_{b_{i;j}}}{¢t} \left[\left(\frac{S_l Á}{B_l} \right)^{n+1} - \left(\frac{S_l Á}{B_l} \right)^{n} \right]_{i;j} = \frac{V_{b_{i;j}}}{¢t} \left[¢_t \left(\frac{S_l Á}{B_l} \right) \right]_{i;j} \tag{3.20}$$

Expansão do Termo de Acumulação

Para escrever a aproximação do termo de acumulação em função das variáveis primárias de modo a preservar o balanço de massa do sistema, é necessário expandir este termo de maneira adequada. A seguir, é mostrada a expansão do termo de acumulação usando um esquema conservativo (ERTEKIN et al., 2001).

Em um sistema conservativo, deve ser satisfeita a seguinte relação para uma função f qualquer dependente no tempo:

$$¢_t f = f^{n+1} - f^n \tag{3.21}$$

Considerando $f = abc$, e desenvolvendo a expressão, chega-se a:

$$\begin{aligned} ¢_t(abc) &= (abc)^{n+1} - (abc)^n \\ &= b^n c^n ¢_t a + a^{n+1} c^n ¢_t b + a^{n+1} b^{n+1} ¢_t c \end{aligned} \tag{3.22}$$

A eq. (3.22) corresponde à expansão do termo $¢_t(abc)$ usando um esquema conservativo. De…nindo $a = Á; b = 1/B_l; c = S_l$, substituindo na equação (3.22) e rearrumando os termos, chega-se a:

$$¢_t \left(\frac{ÁS_l}{B_l} \right) = \left[\frac{Á^0}{B_l^n} + Á^{n+1} (1/B_l)^0 \right] S_l^n ¢_t p_o + \left(\frac{Á}{B_l} \right)^{n+1} ¢_t S_l \tag{3.23}$$

onde,

$$Á^0 = \frac{Á^{n+1} - Á^n}{p_o^{n+1} - p_o^n} \tag{3.24}$$

$$(1/B_l)^0 = \frac{(1/B_l)^{n+1} - (1/B_l)^n}{p_o^{n+1} - p_o^n} \tag{3.25}$$

Embora a eq. (3.25) tenha sido escrita em função da pressão de óleo, esta consideração não in‡uencia signi…cativamente os resultados da solução. COATS et al. (1974) mostram que as propriedades dependentes unicamente das pressões de fase apresentam baixa não-linearidade, podendo ser expressas em função da pressão de óleo sem perder a acurácia da solução.

Substituindo a eq. (3.23) na eq. (3.20) tem-se a forma …nal do termo de acumulação desenvolvida usando um esquema de expansão conservativo:

$$V_{b_{i;j}} \left[\frac{@}{@t} \left(\frac{S_l Á}{B_l} \right)_{i;j}^{n+1} \right] = \frac{V_{b_{i;j}}}{¢t} \left\{ \left(\frac{Á^0}{B_l^n} + Á^{n+1} (1=B_l)^0 \right)^n S_l^n ¢_t p_o + \left(\frac{Á}{B_l} \right)_{i;j}^{n+1} ¢_t S_l \right\} \quad (3.26)$$

3.2.3 Termos de Poços

Nas equações de ‡uxo, os termos $q_{osc_{i;j}}$ e $q_{wsc_{i;j}}$ correspondem às vazões de óleo e água no poço do bloco $(i; j)$ em condições padrão, respectivamente. A convenção de sinais usada na formulação considera vazão positiva nos poços de injeção (entrando no bloco) e vazão negativa nos poços de produção (saindo do bloco), recebendo valor nulo quando não houver poço no bloco.

Como visto na seção 2.3.1, os termos de poços são de…nidos como condições de contorno internas do problema. No simulador, podem ser especi…cadas 4 tipos de condições de poços, dependendo se o poço é produtor ou injetor, ou se é fornecida a vazão ou a pressão no poço.

O acoplamento entre as equações que descrevem os modelos de poços com as equações de ‡uxo em reservatório é um problema ainda bastante discutido e existem vários modelos que tentam explicá-lo (MATTAX e DALTON, 1990; ERTEKIN et al., 2001). Neste trabalho são analisados apenas poços verticais.

Partindo da lei de Darcy para um ‡uxo radial na vizinhança de um poço, e diante da necessidade de relacionar pressão e vazão no poço com a pressão no bloco, chega-se a uma expressão geral chamada equação de poço. O modelo de PEACEMAN (1977) foi utilizado na implementação da equação de poço, de…nida por:

$$q_{lsc_{i;j}} = ¡ G_w ¸_{l_{i;j}} \left[p_{l_{i;j}} ¡ p_{wf} \right] \quad (3.27)$$

onde

$p_{l_{i;j}}$ é a pressão do componente l no bloco $(i; j)$ e p_{wf} é a pressão no poço;

$q_{lsc_{i;j}}$ é a vazão do componente l no poço do bloco $(i; j)$ em condições padrão;

$¸_{l_{i;j}}$ é a mobilidade do componente l no bloco;

G_w é o fator geométrico do poço, calculado por:

$$G_w = \frac{2¼k_H ¢ z}{\ln (r_{eq}=r_w) + s} \quad (3.28)$$

r_w é o raio do poço;

$k_H = \sqrt{k_x k_y}$ é a média geométrica das permeabilidades horizontais no bloco;

¢z é a espessura do bloco, admitindo que o poço é completado ao longo de toda espessura;

s é o fator de skin;

r_{eq} é o raio equivalente do poço admitindo blocos retangulares e permeabilidades diferentes no plano horizontal, calculado da acordo com PEACEMAN (1997):

$$r_{eq} = 0;28 \frac{\sqrt{\left(\frac{k_y}{k_x}\right)^{1=2} ¢x^2 + \left(\frac{k_x}{k_y}\right)^{1=2} ¢y^2}}{\sqrt{\left(\frac{k_y}{k_y}\right)^{1=4} + \left(\frac{k_x}{k_y}\right)^{1=4}}} \tag{3.29}$$

¢x; ¢y são as dimensões do bloco nas direções x e y, respectivamente;

k_x; k_y são as permeabilidades absolutas do bloco nas direções x e y; respectivamente.

Outros modelos para tratar os termos de poço foram propostos na literatura admitindo outras hipóteses no modelo (ABOU-KASSEM e AZIZ, 1985; BABU e ODEH, 1989). A seguir são mostradas as fórmulas empregadas no cálculo dos termos de poço para cada tipo de condição de poço implementada no simulador. As forças capilares próximo ao poço são consideradas desprezíveis nas equações abaixo, ou seja $p_{o_{i;j}} ' p_{w_{i;j}}$:

a) Injeção de água com vazão constante: Deve ser especi...cada uma vazão de água constante em condições padrão, sendo computada diretamente nas equações do sistema discreto:

$$q_{wsc_{i;j}} = q_{es_{i;j}} \tag{3.30}$$

$q_{es_{i;j}}$ é a vazão de injeção de água especi...cada.

b) Injeção de água com pressão de poço constante: É fornecida a pressão no poço de injeção p_{wf} e a vazão de água no poço é calculada pela eq. (3.27), levando em conta a mobilidade total no bloco, considerada como a soma da mobilidade da água e do óleo:

$$q_{wsc_{i;j}} = ¡G_w \left(\frac{B_{o_{i;j}}}{B_{w_{i;j}}} ¸_{o_{i;j}} + ¸_{w_{i;j}}\right) \left(p_{o_{i;j}} ¡ p_{wf}\right) \tag{3.31}$$

c) Produção de óleo com vazão constante: Neste caso podem ser produzidos óleo e água ao mesmo tempo no poço; é especi...cada a vazão de óleo constante em condições padrão e a vazão de água é calculada em função da vazão de óleo:

$$q_{osc_{i;j}} = q_{es_{i;j}} \tag{3.32}$$

$$q_{wsc_{i;j}} = q_{osc_{i;j}} \frac{¸_{w_{i;j}}}{¸_{o_{i;j}}} \tag{3.33}$$

d) Produção de óleo com pressão de poço constante: A pressão constante no poço p_{wf} é especi…cada e as vazões de óleo e água no poço de produção são calculadas usando a eq. (3.27):

$$q_{lsc_{i;j}} = ¡ G_{w,l_{i;j}} \left[p_{o_{i;j}} ¡ p_{wf} \right] \qquad l = o; w \tag{3.34}$$

3.2.4 Forma Final da Equação de Fluxo Discretizada

Tendo desenvolvido cada termo das equações de ‡uxo discretas (3.9), é possível rescrever este sistema de forma compacta usando as eqs. (3.18), (3.19) e (3.26), aplicadas no passo de tempo atual n + 1; resultando em:

$$\begin{aligned} ¢ \left[T_o^{n+1} ¢ p_o^{n+1} ¡ ²_o^n ¢ Z \right] &= C_{poo_{i;j}}^{n+1} ¢_t p_{o_{i;j}} + C_{swo_{i;j}}^{n+1} ¢_t S_{w_{i;j}} ¡ q_{osc_{i;j}}^{n+1} \\ ¢ \left[T_w^{n+1} ¢ p_o^{n+1} ¡ ¢ P_{cow}^{n+1} ¡ ²_w^n ¢ Z \right] &= C_{pow_{i;j}}^{n+1} ¢_t p_{o_{i;j}} + C_{sww_{i;j}}^{n+1} ¢_t S_{w_{i;j}} ¡ q_{wsc_{i;j}}^{n+1} \end{aligned}$$

$$i = 1; ::; n_x \qquad j = 1; ::; n_y \tag{3.35}$$

onde,

$$¢ \left[T_o^{n+1} ¢ p_o^{n+1} ¡ ²_o^n ¢ Z \right] = ¢_x \left[T_o^{n+1} ¢_x p_o^{n+1} ¡ ²_o^n ¢_x Z \right] + ¢_y \left[T_o^{n+1} ¢_y p_o^{n+1} ¡ ²_o^n ¢_y Z \right] \tag{3.36}$$

$$¢ \left[T_w^{n+1} ¢ p_o^{n+1} ¡ ¢ P_{cow}^{n+1} ¡ ²_w^n ¢ Z \right] = ¢_x \left[T_w^{n+1} ¢_x p_o^{n+1} ¡ ¢_x P_{cow}^{n+1} ¡ ²_w^n ¢_x Z \right] + ¢_y \left[T_w^{n+1} ¢_y p_o^{n+1} ¡ ¢_y P_{cow}^{n+1} ¡ ²_w^n ¢_y Z \right] \tag{3.37}$$

$C_{poo_{i;j}}^{n+1}$, $C_{swo_{i;j}}^{n+1}$, $C_{pow_{i;j}}^{n+1}$, $C_{sww_{i;j}}^{n+1}$, são os coe…cientes do termo de acumulação que multiplicam pressões de óleo e saturações de água nas equações de óleo e água, respectivamente, e são expressos, a partir da eq. (3.26), por:

$$C_{poo_{i;j}}^{n+1} = \frac{V_{b_{i;j}}}{¢ t} \left[1 ¡ S_{w_{i;j}}^n ¡ \left(\frac{Á^0}{B_o^n} + Á^{n+1} (1=B_o)^0 \right)_{i;j} \right] \tag{3.38}$$

$$C_{swo_{i;j}}^{n+1} = ¡ \frac{V_{b_{i;j}}}{¢ t} \left(\frac{Á^{n+1}}{B_o^{n+1}} \right)_{i;j} \tag{3.39}$$

$$C_{pow_{i;j}}^{n+1} = \frac{V_{b_{i;j}}}{¢ t} S_{w_{i;j}}^n \left(\frac{Á^0}{B_w^n} + Á^{n+1} (1=B_w)^0 \right)_{i;j} \tag{3.40}$$

$$C_{sww_{i;j}}^{n+1} = \frac{V_{b_{i;j}}}{¢ t} \left(\frac{Á^{n+1}}{B_w^{n+1}} \right)_{i;j} \tag{3.41}$$

O fato da densidade ter sido admitida constante na eq. (2.14) re‡ete-se nas eqs. (3.35), sendo portanto atualizada sempre em função das pressões do passo de tempo anterior.

3.3 Introdução das Condições Iniciais e de Contorno

Na seção 2.3 foram mostradas as principais condições de contorno e condições iniciais necessárias para complementar o sistema de equações diferenciais do problema. A forma como estas condições são introduzidas no sistema discreto é descrita a seguir.

3.3.1 Inicialização das Pressões e Saturações

A inicialização das variáveis primárias do problema, pressões de óleo e saturações de água, é uma etapa muito importante no processo de simulação (DAKE, 1978; MATTAX e DALTON, 1990; ERTEKIN et al., 2001). Devem ser especi...cadas as condições inicias destas variáveis em cada bloco da malha antes de se iniciar as iterações no tempo.

Especi...cado-se a pressão de óleo e a saturação de água de referência, p_o^{ref} e S_w^{ref}; a uma profundiade Z^{ref}, é possível calcular a pressão de água de referência p_w^{ref} usando a relação de pressão de capilaridade com a saturação S_w^{ref}: Então, as pressões de óleo e água iniciais são calculadas em relação à profundidade $Z_{i;j}$ de cada bloco, considerando a relação de equilíbrio hidrostático abaixo, aplicada aos dois componentes:

$$©_{l_{i;j}} ¡ ©_l^{ref} = 0 \qquad l = o; w \tag{3.42}$$

Usando a eq. (2.12), tem-se:

$$p_{l_{i;j}} = p_l^{ref} + ³ Z_{i;j} ¡ Z^{ref} ´ ½_{l_{i;j}} g \tag{3.43}$$

Como a densidade $½_{l_{i;j}}$ na profundidade $Z_{i;j}$ depende da pressão nesta profundidade, que é uma incógnita, o problema é resolvido iterativamente calculando-se a densidade em função da pressão da iteração anterior até atingir a convergência. A equação usada no processo iterativo para cada componente é a recorrência de ponto ...xo abaixo:

$$p_{l_{i;j}}^{(À)} = p_l^{ref} + ³ Z_{i;j} ¡ Z^{ref} ´ ½_{l_{i;j}}^{(À¡1)} g \qquad À = 1; 2; ::: \tag{3.44}$$

onde (À) é a iteração atual.

Tendo calculado as pressões de óleo e água iniciais nos blocos, as saturações de água iniciais, $S_{w_{i;j}}$; são obtidas, usando a relação inversa da pressão de capilaridade, a partir da expressão $P_{cow}\ ³ S_{w_{i;j}} = p_{o_{i;j}} ¡ p_{w_{i;j}}$.

3.3.2 Introdução das Condições de Contorno

As condições de contorno compreendem as condições externas no reservatório e as condições de poços, estas últimas já discutidas na seção 3.2.3.

As condições de contorno externas são consideradas admitindo-se ‡uxo nulo nas fronteiras do reservatório. Para isto, basta assumir transmissibilidades nulas nas faces externas dos blocos de fronteira, e aplicar as equações de ‡uxo discretas tanto para os blocos internos como para os blocos da fronteira.

3.4 Linearização das Equações de Fluxo

Analisando as equações (3.35), veri…ca-se que transmissibilidades, pressões de capilaridade, vazões nos poços e coe…cientes dos termos de acumulação são calculados em função das pressões e saturações do passo de tempo atual $n+1$; que são as incógnitas do sistema. Devido a esta característica não-linear do problema, devem ser utilizadas técnicas de linearização para obter sua solução.

As formulações numéricas utilizadas para a linearização das equações discretas implementdas no simulador são: formulação totalmente implícita e IMPES (MATTAX E DALTON, 1990; ERTEKIN et al., 2001).

3.4.1 Formulação Totalmente Implícita

Na formulação totalmente implícita as variáveis primárias são tratadas implicitamente nas eqs. (3.35), sendo necessário empregar um método iterativo para resolver este sistema. Neste caso, é utilizado o método iterativo de Newton, que consiste em escrever expressões aproximadas para os resíduos, partindo de expansões em Séries de Taylor, e igualar estes resíduos a zero. A expressão dos resíduos das equações de ‡uxo pode ser escrita para a iteração (ν) como:

$$r_{o_{i;j}}^{(\nu)} = \left[T_o^{(\nu)}\right]\left[p_o^{(\nu)} - \left(\gamma_o^n\right)\left[Z\right]\right] - \frac{V_{b_{i;j}}}{\Delta t}\left[\frac{(1-S_w)\phi^{(\nu)}}{B_o} - \frac{(1-S_w)\phi^n}{B_o}\right]_{i;j} + q_{osc_{i;j}}^{(\nu)} \tag{3.45}$$

$$r_{w_{i;j}}^{(\nu)} = \left[T_w^{(\nu)}\right]\left[p_o^{(\nu)} - P_{cow}^{(\nu)} - \left(\gamma_w^n\right)\left[Z\right]\right] - \frac{V_{b_{i;j}}}{\Delta t}\left[\left(\frac{S_w\phi}{B_w}\right)^{(\nu)} - \left(\frac{S_w\phi}{B_w}\right)^n\right]_{i;j} + q_{wsc_{i;j}}^{(\nu)} \tag{3.46}$$

Seja uma malha bidimensional $n_x \times n_y$ e considere uma numeração natural por linhas para esta malha, usando a variável n para representar os índices $(i;j)$ de cada

bloco. O método de Newton resulta na solução do seguinte sistema iterativo:

$$J^{(o)} \pm X^{(o+1)} = \neg R^{(\grave{A})} \tag{3.47}$$

$$\pm X^{(o+1)} = X^{(o+1)} \neg X^{(o)} \tag{3.48}$$

onde,

$(o + 1)$ é a iteração de Newton atual e (o) é a iteração anterior;

$\pm X^{(o+1)}$ é o vetor de incremento na solução;

$X^{(\grave{A}+1)}$ é o vetor de soluções, formado pelos subvetores $X_n^{(\grave{A}+1)}$; dados por:

$$X^{(\grave{A}+1)} = \left[X_{1;1}^{(\grave{A}+1)}; X_{2;1}^{(\grave{A}+1)}; \ldots; X_n^{(\grave{A}+1)}; \ldots; X_{n_x;n_y}^{(\grave{A}+1)} \right]^T \tag{3.49}$$

$$X_n^{(\grave{A}+1)} = \begin{bmatrix} p_{o_n}^{(\grave{A}+1)} \\ S_{w_n}^{(\grave{A}+1)} \end{bmatrix} \tag{3.50}$$

$R^{(\grave{A})}$ é o vetor de resíduos do sistema, formado pelos subvetores $R_n^{(\grave{A})}$ referentes a cada bloco da malha,

dados por:

$$R^{(\grave{A})} = \left[R_{1;1}^{(\grave{A})}; R_{2;1}^{(\grave{A})}; \ldots; R_n^{(\grave{A})}; \ldots; R_{n_x;n_y}^{(\grave{A})} \right]^T \tag{3.51}$$

$$R_n^{(\grave{A})} = \begin{bmatrix} r_{o_n}^{(\grave{A})} \\ r_{w_n}^{(\grave{A})} \end{bmatrix} \tag{3.52}$$

$J^{(o)}$ é a matriz Jacobiana.

Para ilustrar a estrutura da matriz Jacobiana, considere a malha 3x3 na Figura 3.3. A matriz Jacobiana referente a esta malha possui uma estrutura bloco-pentadiagonal e está indicada a seguir:

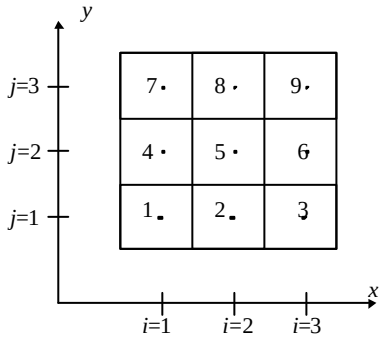

Figura 3.3: Malha de blocos centrados, 3x3.

$$J^{(\lambda)} = \begin{bmatrix} CE & N & & & & & \\ WCE & & N & & & & \\ & WC & & N & & & \\ S & & C & E & & N & \\ & S & & W & C & E & N \\ & & S & & W & C & \\ & & & S & & W & CE \\ & & & & S & & WC \end{bmatrix}^{(\lambda)}$$

Cada elemento desta matriz é uma submatriz 2x2 cujos elementos são compostos pelas derivadas parciais dos resíduos em relação às variáveis primárias. Observando as eqs. (3.45) e (3.46), veri…ca-se que o resíduo aplicado a cada bloco da malha 2D é função das incógnitas dos blocos de acordo com o esquema de diferenças …nitas com 5 pontos utilizado, indicado na Figura 3.4. Portanto, para o bloco $(i;j)$, tem-se:

$$r_{l_{i;j}} = r_{l_{i;j}}\left(p_{o_{i;j}}; S_{w_{i;j}}; p_{o_{i+1;j}}; S_{w_{i+1;j}}; p_{o_{i-1;j}}; S_{w_{i-1;j}}; p_{o_{i;j+1}}; S_{w_{i;j+1}}; p_{o_{i;j-1}}; S_{w_{i;j-1}}\right) \quad (3.53)$$

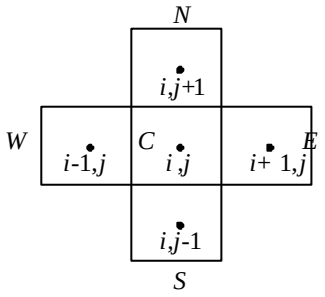

Figura 3.4: Esquema de diferenças …nitas com 5 pontos - duas variáveis por bloco.

Para simpli…car as expressões das derivadas da matriz Jacobiana, considere a variável n representando os índices do bloco $(i;j)$, e a variável m representando o conjunto de índices dos blocos vizinhos ao bloco $(i;j)$: Ou seja, para o bloco $(i;j)$; $n = i;j$ e para os blocos vizinhos, $m = f(i-1;j);(i+1;j);(i;j-1);(i+1;j)g$:

Usando esta convenção indicial, as eqs. (3.45) e (3.46) são rescritas com a seguinte notação:

$$r_{o_n}^{(\lambda)} = \sum_m \left[T_{o_{n;m}}^{(\lambda)}\left(\phi_m p_o^{(\lambda)}\right) - \left(\phi_{o_{n;m}} \phi_m Z\right)\right] - \frac{V_{b_n}}{\phi t}\left[\frac{(1-S_w)\rho}{B_o}^{(\lambda)} - \frac{(1-S_w)\rho}{B_o}^{n}\right] + q_{osc_n}^{(\lambda)} \quad (3.54)$$

$$r_{w_n}^{(k)} = \sum_m \left[T_{w_{n;m}}^{(k)} \left(\Delta_m p_w^{(k)} - \Delta_m P_{cow}^{(k)} + \gamma_{w_{n;m}}^{n} \Delta_m Z \right) \right] - \frac{V_{b_n}}{\Delta t} \left[\left(\frac{S_w \phi}{B_w} \right)^{(k)} - \left(\frac{S_w \phi}{B_w} \right)^{n} \right] + q_{wsc_n}^{(k)} \tag{3.55}$$

onde,

$$\Delta_m p_l^{(k)} = p_{l_m}^{(k)} - p_{l_n}^{(k)} \qquad l = o; w \tag{3.56}$$

$$\Delta_m P_{cow}^{(k)} = P_{cow}\left(S_{w_m}^{(k)}\right) - P_{cow}\left(S_{w_n}^{(k)}\right) \tag{3.57}$$

$$\Delta_m Z = Z_m - Z_n \tag{3.58}$$

$$\Delta_t p_{o_n} = p_{o_n}^{(k)} - p_{o_n}^{n} \tag{3.59}$$

$$\Delta_t S_{w_n} = S_{w_n}^{(k)} - S_{w_n}^{n} \tag{3.60}$$

$$T_{l_{n;m}}^{(k)} = \begin{cases} T_{l x_{i \pm 1=2;j}}^{(k)} & \rightarrow m = i \pm 1; j \\ T_{l y_{i;j \pm 1=2}}^{(k)} & \rightarrow m = i; j \pm 1 \end{cases} \qquad l = o; w \tag{3.61}$$

$$\circ_{l_{n;m}}^{(k)} = \begin{cases} \circ_{l_{i \pm 1=2;j}}^{(k)} & \rightarrow m = i \pm 1; j \\ \circ_{l_{i;j \pm 1=2}}^{(k)} & \rightarrow m = i; j \pm 1 \end{cases} \qquad l = o; w \tag{3.62}$$

Portanto, as submatrizes que compõem a matriz Jacobiana apresentam a seguinte forma:

$$C_n^{(k)} = \begin{bmatrix} \partial r_{o_n}/\partial p_{o_n} & \partial r_{o_n}/\partial S_{w_n} \\ \partial r_{w_n}/\partial p_{o_n} & \partial r_{w_n}/\partial S_{w_n} \end{bmatrix}^{(k)} \tag{3.63}$$

$$W_n^{(k)}, E_n^{(k)}, S_n^{(k)}, N_n^{(k)} = \begin{bmatrix} \partial r_{o_n}/\partial p_{o_m} & \partial r_{o_n}/\partial S_{w_m} \\ \partial r_{w_n}/\partial p_{o_m} & \partial r_{w_n}/\partial S_{w_m} \end{bmatrix}^{(k)} \tag{3.64}$$

As expressões das derivadas do resíduo em relação às variáveis primárias são mostradas a seguir:

- em relação às variáveis dos blocos vizinhos ao bloco $(i; j)$:

$$\left(\frac{\partial r_{o_n}}{\partial p_{o_m}} \right)^{(k)} = T_{o_{n;m}}^{(k)} + \left(\Delta_m p_o^{(k)} - \gamma_{o_{n;m}}^{n} \Delta_m Z \right) \left(\frac{\partial T_{o_{n;m}}}{\partial p_{o_m}} \right)^{(k)} \tag{3.65}$$

$$\left(\frac{\partial r_{o_n}}{\partial S_{w_m}} \right)^{(k)} = \left(\Delta_m p_o^{(k)} - \gamma_{o_{n;m}}^{n} \Delta_m Z \right) \left(\frac{\partial T_{o_{n;m}}}{\partial S_{w_m}} \right)^{(k)} \tag{3.66}$$

$$\left(\frac{\partial r_{w_n}}{\partial p_{o_m}} \right)^{(k)} = T_{w_{n;m}}^{(k)} + \left(\Delta_m p_o^{(k)} - \Delta_m P_{cow}^{(k)} - \gamma_{w_{n;m}}^{n} \Delta_m Z \right) \left(\frac{\partial T_{w_{n;m}}}{\partial p_{o_m}} \right)^{(k)} \tag{3.67}$$

$$\left(\frac{\partial r_{w_n}}{\partial S_{w_m}} \right)^{(k)} = \left(\Delta_m p_o^{(k)} - \Delta_m P_{cow}^{(k)} - \gamma_{w_{n;m}}^{n} \Delta_m Z \right) \left(\frac{\partial T_{w_{n;m}}}{\partial S_{w_m}} \right)^{(k)} - T_{w_{n;m}}^{(k)} \left(\frac{\partial P_{cow}}{\partial S_{w_m}} \right)^{(k)} \tag{3.68}$$

- em relação às variáveis do bloco (i; j):

$$\left(\frac{\partial r_{o_n}}{\partial p_{o_n}}\right)^{(\lambda)} = \sum_m \left[-T^{(\lambda)}_{o_{n;m}} + \left(\phi_m p_o^{(\lambda)} - \varphi^n_{o_{n;m}} \phi_m Z \right) \left(\frac{\partial T_{o_{n;m}}}{\partial p_{o_n}}\right)^{(\lambda)} \right] - C^{(\lambda)}_{poo_n} + \left(\frac{\partial q_{osc_n}}{\partial p_{o_n}}\right)^{(\lambda)} \tag{3.69}$$

$$\left(\frac{\partial r_{o_n}}{\partial S_{w_n}}\right)^{(\lambda)} = \sum_m \left[\left(\phi_m p_o^{(\lambda)} - \varphi^n_{o_{n;m}} \phi_m Z \right) \left(\frac{\partial T_{o_{n;m}}}{\partial S_{w_n}}\right)^{(\lambda)} \right] - C^{(\lambda)}_{swo_n} + \left(\frac{\partial q_{osc_n}}{\partial S_{w_n}}\right)^{(\lambda)} \tag{3.70}$$

$$\left(\frac{\partial r_{w_n}}{\partial p_{o_n}}\right)^{(\lambda)} = \sum_m \left[-T^{(\lambda)}_{w_{n;m}} + \left(\phi_m p_o^{(\lambda)} - \phi_m P^{(\lambda)}_{cow} - \varphi^n_{w_{n;m}} \phi_m Z \right) \left(\frac{\partial T_{w_{n;m}}}{\partial p_{o_n}}\right)^{(\lambda)} \right] - C^{(\lambda)}_{pow_n} + \left(\frac{\partial q_{wsc_n}}{\partial p_{o_n}}\right)^{(\lambda)} \tag{3.71}$$

$$\left(\frac{\partial r_{w_n}}{\partial S_{w_n}}\right)^{(\lambda)} = \sum_m \left[\left(\phi_m p_o^{(\lambda)} - \phi_m P^{(\lambda)}_{cow} - \varphi^n_{w_{n;m}} \phi_m Z \right) \left(\frac{\partial T_{w_{n;m}}}{\partial S_{w_n}}\right)^{(\lambda)} + T^{(\lambda)}_{w_{n;m}} \left(\frac{\partial P_{cow}}{\partial S_{w_n}}\right)^{(\lambda)} \right] - C^{(\lambda)}_{sww_n} + \left(\frac{\partial q_{wsc_n}}{\partial S_{w_n}}\right)^{(\lambda)} \tag{3.72}$$

As derivadas das transmissibilidades são calculadas a partir das eqs. (3.14) e (3.16), segundo as expressões abaixo:

$$\left(\frac{\partial T_{l_{n;m}}}{\partial p_{o_m}}\right)^{(\lambda)} = \begin{cases} G_{n;m} \left(\partial_{,} l_m / \partial p_{o_m}\right)^{(\lambda)} & \text{se} \quad \gamma^{(\lambda)}_{l_m} \geq \gamma^{(\lambda)}_{l_n} \\ 0 & \text{se} \quad \gamma^{(\lambda)}_{l_m} < \gamma^{(\lambda)}_{l_n} \end{cases} \tag{3.73}$$

$$\left(\frac{\partial T_{l_{n;m}}}{\partial S_{w_m}}\right)^{(\lambda)} = \begin{cases} G_{n;m} \left(\partial_{,} l_m / \partial S_{w_m}\right)^{(\lambda)} & \text{se} \quad \gamma^{(\lambda)}_{l_m} \geq \gamma^{(\lambda)}_{l_n} \\ 0 & \text{se} \quad \gamma^{(\lambda)}_{l_m} < \gamma^{(\lambda)}_{l_n} \end{cases} \tag{3.74}$$

$$\left(\frac{\partial T_{l_{n;m}}}{\partial p_{o_n}}\right)^{(\lambda)} = \begin{cases} 0 & \text{se} \quad \gamma^{(\lambda)}_{l_m} \geq \gamma^{(\lambda)}_{l_n} \\ G_{n;m} \left(\partial_{,} l_n / \partial p_{o_n}\right)^{(\lambda)} & \text{se} \quad \gamma^{(\lambda)}_{l_m} < \gamma^{(\lambda)}_{l_n} \end{cases} \tag{3.75}$$

$$\left(\frac{\partial T_{l_{n;m}}}{\partial S_{w_n}}\right)^{(\lambda)} = \begin{cases} 0 & \text{se} \quad \gamma^{(\lambda)}_{l_m} \geq \gamma^{(\lambda)}_{l_n} \\ G_{n;m} \left(\partial_{,} l_n / \partial S_{w_n}\right)^{(\lambda)} & \text{se} \quad \gamma^{(\lambda)}_{l_m} < \gamma^{(\lambda)}_{l_n} \end{cases} \tag{3.76}$$

As derivadas das vazões nos poços são calculadas analiticamente partindo das eqs. (3.30) a (3.34), e dependem da condição de poço especi...cada:

1 - poço de injeção de água com vazão especi...cada:

$$\left(\frac{\partial q_{osc_n}}{\partial p_{o_n}}\right)^{(\lambda)} = \left(\frac{\partial q_{osc_n}}{\partial S_{w_n}}\right)^{(\lambda)} = \left(\frac{\partial q_{wsc_n}}{\partial p_{o_n}}\right)^{(\lambda)} = \left(\frac{\partial q_{wsc_n}}{\partial S_{w_n}}\right)^{(\lambda)} = 0 \tag{3.77}$$

2 - poço de injeção de água com pressão de poço especi...cada:

$$\left(\frac{\partial q_{osc_n}}{\partial p_{o_n}}\right)^{(k)} = \left(\frac{\partial q_{osc_n}}{\partial S_{w_n}}\right)^{(k)} = 0 \tag{3.78}$$

$$\left(\frac{\partial q_{wsc_n}}{\partial p_{o_n}}\right)^{(k)} = -G_w \begin{bmatrix} \partial\left(B_{o_n}\lambda_{o_n}/B_{w_n}\right)/\partial p_{o_n} + \\ \partial\lambda_{w_n}/\partial p_{o_n} \end{bmatrix}^{(k)} \left(p_{o_n} - p_{wf}\right) + \left(B_{o_n}/B_{w_n}\right)\lambda_{o_n} + \lambda_{w_n} \tag{3.79}$$

$$\left(\frac{\partial q_{wsc_n}}{\partial S_{w_n}}\right)^{(k)} = -G_w \left[\partial\left(B_{o_n}\lambda_{o_n}/B_{w_n}\right)/\partial S_{w_n} + \partial\lambda_{w_n}/\partial S_{w_n}\right]^{(k)} \left(p_{o_n} - p_{wf}\right)^{(k)} \tag{3.80}$$

3 - poço de produção com vazão de óleo especi...cada:

$$\left(\frac{\partial q_{osc_n}}{\partial p_{o_n}}\right)^{(k)} = \left(\frac{\partial q_{osc_n}}{\partial S_{w_n}}\right)^{(k)} = 0 \tag{3.81}$$

$$\left(\frac{\partial q_{wsc_n}}{\partial p_{o_n}}\right)^{(k)} = \left[q_{osc_n}\partial\left(\lambda_{w_n}/\lambda_{o_n}\right)/\partial p_{o_n}\right]^{(k)} \tag{3.82}$$

$$\left(\frac{\partial q_{wsc_n}}{\partial S_{w_n}}\right)^{(k)} = \left[q_{osc_n}\partial\left(\lambda_{w_n}/\lambda_{o_n}\right)/\partial S_{w_n}\right]^{(k)} \tag{3.83}$$

4 - poço de produção com pressão de poço especi...cada:

$$\left(\frac{\partial q_{osc_n}}{\partial p_{o_n}}\right)^{(k)} = -G_w\left[\left(\partial\lambda_{o_n}/\partial p_{o_n}\right)\left(p_{o_n} - p_{wf}\right) + \lambda_{o_n}\right]^{(k)} \tag{3.84}$$

$$\left(\frac{\partial q_{osc_n}}{\partial S_{w_n}}\right)^{(k)} = -G_w\left(\partial\lambda_{o_n}/\partial S_{w_n}\right)^{(k)}\left(p_{o_n} - p_{wf}\right)^{(k)} \tag{3.85}$$

$$\left(\frac{\partial q_{wsc_n}}{\partial p_{o_n}}\right)^{(k)} = -G_w\left[\left(\partial\lambda_{w_n}/\partial p_{o_n}\right)\left(p_{o_n} - p_{wf}\right) + \lambda_{w_n}\right]^{(k)} \tag{3.86}$$

$$\left(\frac{\partial q_{wsc_n}}{\partial S_{w_n}}\right)^{(k)} = -G_w\left(\partial\lambda_{w_n}/\partial S_{w_n}\right)^{(k)}\left(p_{o_n} - p_{wf}\right)^{(k)} \tag{3.87}$$

As derivadas da mobilidade no bloco em relação à pressão de óleo e à saturação de água são calculadas pelas seguintes expressões:

$$\left(\frac{\partial\lambda_{l_n}}{\partial p_{o_n}}\right)^{(k)} = k_{rl_n}^{(k)}\left[\frac{1}{B_{l_n}}\frac{\partial}{\partial p_{o_n}}\left(\frac{1}{\mu_{l_n}}\right) + \frac{1}{\mu_{l_n}}\frac{\partial}{\partial p_{o_n}}\left(\frac{1}{B_{l_n}}\right)\right]^{(k)} \tag{3.88}$$

$$\left(\frac{\partial\lambda_{l_n}}{\partial S_{w_n}}\right)^{(k)} = \left(\frac{1}{\mu_{l_n}B_{l_n}}\frac{\partial k_{rl_n}}{\partial S_{w_n}}\right)^{(k)} \tag{3.89}$$

As derivadas do fator volume de formação, viscosidade, permeabilidade relativa e pressão de capilaridade são calculadas numericamente, de acordo com a fórmula genérica aplicada a uma função f (x) qualquer, dada por:

$$\frac{@f\,(x)}{@x} = \frac{f\,(x + ") \;¡\; f\,(x)}{"} \tag{3.90}$$

onde " é um incremento no valor da variável x, podendo ser da ordem de $10^{¡6}$, por exemplo.

O processo iterativo na formulação totalmente implícita prossegue até que sejam atingidos os critérios de convergência estabelecidos no problema. São utilizados os critérios de convergência do solver não-linear do PETSc (BALAY et al., 2001) discutidos no Capítulo 6.

3.4.2 Formulação IMPES

Na formulação IMPES (IMplicit Pressure, Explicit Saturation) as pressões de óleo são calculadas implicitamente e os termos que variam com as saturações de água são calculados explicitamente.

A idéia consiste em escrever uma única equação para o sistema contendo apenas pressões de óleo como incógnitas. Para isto os termos de transmissibilidades, pressão de capilaridade e termos de poço são calculados explicitamente em função das pressões e saturações do passo de tempo anterior n. Então é possível rescrever as esquações de ‡uxo discretas (3.35) da seguinte forma:

$$¢\left[T_o^n\right]¢p_o^{n+1} ¡ \left[¢_o^n¢Z\right] = C_{poo_{i;j}}^{n+1}¢_t p_{o_{i;j}} + C_{swo_{i;j}}^{n+1}¢_t S_{w_{i;j}} ¡ q_{osc_{i;j}}^n \tag{3.91}$$

$$¢\left[T_w^n\right]¢p_o^{n+1} ¡ ¢P_{cow}^n ¡ \left[¢_w^n¢Z\right] = C_{pow_{i;j}}^{n+1}¢_t p_{o_{i;j}} + C_{sww_{i;j}}^{n+1}¢_t S_{w_{i;j}} ¡ q_{wsc_{i;j}}^n \tag{3.92}$$

É possível eliminar as saturações $¢_t S_{w_{i;j}}$ do sistema multiplicando a equação da água pelo fator ® = $¡C_{swo_{i;j}}^{n+1}/C_{sww_{i;j}}^{n+1}$ e somando o resultado com a equação do óleo. Como ® pode ser escrito por ® = $B_{w_{i;j}}^{n+1}/B_{o_{i;j}}^{n+1}$, chega-se à seguinte equação de pressões de óleo:

$$\begin{aligned} &B_{o_{i;j}}^{n+1}¢\left[T_o^n\right]¢p_o^{n+1} ¡ \left[¢_o^n¢Z\right] + B_{w_{i;j}}^{n+1}¢\left[T_w^n\right]¢p_o^{n+1} ¡ ¢P_{cow}^n ¡ \left[¢_w^n¢Z\right] = \\ &B_{o_{i;j}}^{n+1}C_{poo_{i;j}}^{n+1} + B_{w_{i;j}}^{n+1}C_{pow_{i;j}}^{n+1}¢_t p_{o_{i;j}} ¡ B_{o_{i;j}}^{n+1}q_{osc_{i;j}}^n ¡ B_{w_{i;j}}^{n+1}q_{wsc_{i;j}}^n \end{aligned} \tag{3.93}$$

Na equação acima, $C_{poo_{i;j}}^{n+1}$, $C_{pow_{i;j}}^{n+1}$, $B_{o_{i;j}}^{n+1}$ e $B_{w_{i;j}}^{n+1}$ devem ser calculados em função das pressões no passo de tempo atual $n+1$, sendo necessário portanto realizar iterações no processo de solução desta equação, de modo a preservar o balanço de massa do sistema. Normalmente, são necessárias apenas duas iterações até atingir a convergência.

Tendo-se calculado as pressões de óleo $p_{o_{i;j}}^{n+1}$, calcula-se as saturações de água $S_{w_{i;j}}^{n+1}$ explicitamente substituindo as pressões calculadas em uma das equações, (3.91) ou (3.92). Usando a eq. (3.91), a solução para as saturações de água é dada pela fórmula:

$$S_{w_{i;j}}^{n+1} = S_{w_{i;j}}^{n} + \frac{1}{C_{swo_{i;j}}^{n+1}} \left[T_o^n \Delta p_o^{n+1} - \left(\Delta p_c^n \Delta Z \right)^{i} - C_{poo_{i;j}}^{n+1} \Delta_t p_{o_{i;j}} + q_{osc_{i;j}}^{n} \right]^{o} \qquad (3.94)$$

Considerando a malha da Figura 3.3, a matriz do sistema de equações $A p_o = b$; obtido aplicando-se a eq. (3.93) a cada bloco da malha, apresenta a seguinte forma:

$$A = \begin{bmatrix}
C & E & & N & & & & & & & & \\
W & C & E & & N & & & & & & & \\
 & W & C & & & N & & & & & & \\
S & & & C & E & & N & & & & & \\
 & S & & W & C & E & & N & & & & \\
 & & S & & W & C & & & N & & & \\
 & & & S & & & C & E & & & & \\
 & & & & S & & W & C & E & & & \\
 & & & & & S & & W & C & & & \\
 & & & & & & & & & & & \\
 & & & & & & & & & & & \\
 & & & & & & & & & & &
\end{bmatrix}$$

Esta matriz possui uma estrutura similar à da matriz Jacobiana obtida na formulação totalmente implícita, porém, ao invés de ser bloco-pentadiagonal, para o caso 2D mostrado, sua estrutura é somente pentadiagonal, devido ao sistema só possuir uma incógnita por bloco. Desenvolvendo e aplicando a eq. (3.93) a cada bloco, tem-se as seguintes expressões para os elementos da matriz A:

$$W_{i;j} = B_{o_{i;j}}^{n+1} T_{o_{i-1=2;j}}^{n} + B_{w_{i;j}}^{n+1} T_{w_{i-1=2;j}}^{n} \qquad (3.95)$$

$$E_{i;j} = B_{o_{i;j}}^{n+1} T_{o_{i+1=2;j}}^{n} + B_{w_{i;j}}^{n+1} T_{w_{i+1=2;j}}^{n} \qquad (3.96)$$

$$S_{i;j} = B_{o_{i;j}}^{n+1} T_{o_{i;j-1=2}}^{n} + B_{w_{i;j}}^{n+1} T_{w_{i;j-1=2}}^{n} \qquad (3.97)$$

$$N_{i;j} = B_{o_{i;j}}^{n+1} T_{o_{i;j+1=2}}^{n} + B_{w_{i;j}}^{n+1} T_{w_{i;j+1=2}}^{n} \qquad (3.98)$$

$$\begin{aligned} C_{i;j} = & -\left[B_{o_{i;j}}^{n+1} \left(T_{o_{i-1=2;j}}^{n} + T_{o_{i+1=2;j}}^{n} + T_{o_{i;j-1=2}}^{n} + T_{o_{i;j+1=2}}^{n} \right) \right. \\ & \left. - B_{w_{i;j}}^{n+1} \left(T_{w_{i-1=2;j}}^{n} + T_{w_{i+1=2;j}}^{n} + T_{w_{i;j-1=2}}^{n} + T_{w_{i;j+1=2}}^{n} \right) \right] \\ & - B_{o_{i;j}}^{n+1} C_{poo_{i;j}}^{n+1} + B_{w_{i;j}}^{n+1} C_{pow_{i;j}}^{n+1} - ATP_{i;j}^{n+1} \end{aligned} \qquad (3.99)$$

onde $ATP_{i;j}^{n+1}$ contém a contribuição do termo de poço, que varia de acordo com a condição de poço especi...cada. Usando a mesma numeração usada na seção anterior para identi...car o tipo de poço, tem-se:

$$ATP_{i;j}^{n+1} = \begin{cases} 0 & \text{se tipo = 1} \\ B_{w_{i;j}}^{n+1} G_w \left[B_{o_{i;j}}^{n} = B_{w_{i;j}}^{n} \left(\lambda_{o_{i;j}}^{n} + \lambda_{w_{i;j}}^{n} \right) \right] & \text{se tipo = 2} \\ 0 & \text{se tipo = 3} \\ B_{o_{i;j}}^{n+1} G_w \lambda_{o_{i;j}}^{n} + B_{w_{i;j}}^{n+1} G_w \lambda_{w_{i;j}}^{n} & \text{se tipo = 4} \end{cases} \tag{3.100}$$

Os elementos do vetor b são calculados por:

$$\begin{aligned} b_{i;j} = {} & B_{o_{i;j}}^{n+1} \left[\begin{array}{l} T_{o_{i-1=2;j}}^{n} \Phi_{o_{i-1=2;j}}^{n} (Z_{i-1;j} - Z_{i;j}) + T_{o_{i+1=2;j}}^{n} \Phi_{o_{i+1=2;j}}^{n} (Z_{i+1;j} - Z_{a;j}) \\ + T_{o_{i;j-1=2}}^{n} \Phi_{o_{i;j-1=2}}^{n} (Z_{i;j-1} - Z_{i;j}) + T_{o_{i;j+1=2}}^{n} \Phi_{o_{i;j+1=2}}^{n} (Z_{i;j+1} - Z_{i;j}) \end{array} \right] \\ & + B_{w_{i;j}}^{n+1} \left[\begin{array}{l} T_{w_{i-1=2;j}}^{n} \Phi_{w_{i-1=2;j}}^{n} (Z_{i-1;j} - Z_{i;j}) + T_{w_{i+1=2;j}}^{n} \Phi_{w_{i+1=2;j}}^{n} (Z_{i+1;j} - Z_{i;j}) \\ + T_{w_{i;j-1=2}}^{n} \Phi_{w_{i;j-1=2}}^{n} (Z_{i;j-1} - Z_{i;j}) + T_{w_{i;j+1=2}}^{n} \Phi_{w_{i;j+1=2}}^{n} (Z_{i;j+1} - Z_{i;j}) \end{array} \right] \\ & + B_{w_{i;j}}^{n+1} \left[\begin{array}{l} T_{w_{i-1=2;j}}^{n} \left(P_{cow_{i-1;j}} - P_{cow_{i;j}} \right) + T_{w_{i+1=2;j}}^{n} \left(P_{cow_{i+1;j}} - P_{cow_{i;j}} \right) \\ T_{w_{i;j-1=2}}^{n} \left(P_{cow_{i;j-1}} - P_{cow_{i;j}} \right) + T_{w_{i;j+1=2}}^{n} \left(P_{cow_{i;j+1}} - P_{cow_{i;j}} \right) \end{array} \right] \\ & - \left[B_{o_{i;j}}^{n+1} C_{poo_{i;j}}^{n+1} + B_{w_{i;j}}^{n+1} C_{pow_{i;j}}^{n+1} \right] p_{o_{i;j}}^{n} - BTP_{i;j}^{n+1} \end{aligned} \tag{3.101}$$

onde $BTP_{i;j}^{n+1}$ contém a contribuição do termo de poço, e é calculado por:

$$BTP_{i;j}^{n+1} = \begin{cases} B_{w_{i;j}}^{n+1} q_{wsc_{i;j}}^{n} & \text{se tipo = 1} \\ B_{w_{i;j}}^{n+1} G_w \left[B_{o_{i;j}}^{n} = B_{w_{i;j}}^{n} \left(\lambda_{o_{i;j}}^{n} + \lambda_{w_{i;j}}^{n} \right) \right] p_{wf} & \text{se tipo = 2} \\ B_{o_{i;j}}^{n+1} q_{osc_{i;j}}^{n} + B_{w_{i;j}}^{n+1} q_{wsc_{i;j}}^{n} & \text{se tipo = 3} \\ B_{o_{i;j}}^{n+1} G_w \lambda_{o_{i;j}}^{n} p_{wf} + B_{w_{i;j}}^{n+1} G_w \lambda_{w_{i;j}}^{n} p_{wf} & \text{se tipo = 4} \end{cases} \tag{3.102}$$

Embora a formulação IMPES exija menos esforço computacional durante a simulação do que a formulação totalmente implícita, esta última tem a vantagem de ser incondicionalmente estável permitindo utilizar maiores intervalos de tempo do que a formulação IMPES. Com as saturações de água calculadas explicitamente, a formulação IMPES pode apresentar problemas de instabilidade na solução quando o tamanho dos blocos da malha são reduzidos ou quando o tamanho do passo de tempo é aumentado, limitando sua aplicação em alguns problemas (MATTAX e DALTON, 1990).

3.5 Detalhes Adicionais

3.5.1 Controle do Passo de Tempo

O controle do passo de tempo (AZIZ e SETTARI, 1979; ERTEKIN et al., 2001) durante o processo de simulação é importante pois possibilita alterar o intervalo de tempo em

função das máximas variações de pressão e saturação calculadas entre passos de tempo consecutivos, aumentando ou reduzindo o valor de Δt.

O intervalo de tempo no passo atual $n+1$ é controlado em função das máximas diferenças de pressão e saturação veri…cadas entre o passo de tempo atual e o anterior, de acordo com os critérios estabelecidos abaixo (AZIZ e SETTARI, 1979):

$$\Delta t^{n+1} = \begin{cases} \alpha \Delta t^n \left(\Delta p_{o_{max}} / \Delta p_o^{n+1}\right) & \text{se} \quad \Delta p_o^{n+1} < \Delta p_{max} \text{ e } \Delta S_w^{n+1} < \Delta S_{max} \\ 0{:}5\,\Delta t^n & \text{caso contrário.} \end{cases} \tag{3.103}$$

onde,

$\Delta p_o^{n+1} = \max \left[p_{o_{i;j}}^{n+1} - p_{o_{i;j}}^{n} \right]$;

$\Delta S_w^{n+1} = \max \left[S_{w_{i;j}}^{n+1} - S_{w_{i;j}}^{n} \right]$;

α é usado para reduzir o fator de aumento do intervalo de tempo, podendo ser utilizado igual a 0:8;

Δp_{max} e ΔS_{max} são as máximas variações de pressão e saturação admissíveis, respectivamente.

O valor de Δt é alterado obedecendo sempre os limites de intervalo de tempo máximo Δt_{max} e mínimo Δt_{min}, também fornecidos como parâmetros de entrada. Se o Δt calculado for menor que o Δt_{min}, a simulação é interrompida.

3.5.2 Veri…cação de Balanço de Massa

O balanço de massa fornece uma espressão para a conservação de massa em um determinado volume de controle. O cálculo do balanço de massa no …nal de cada passo de tempo é necessário para validar as soluções obtidas na simulação, veri…cando o balanço de massa em todo o volume do reservatório.

A veri…cação de balanço de massa é representada pela razão entre a massa acumulada e a massa total que entra ou sai pelas fronteiras do reservatório (ERTEKIN et al., 2001). Admitindo um reservatório com ‡uxo nulo nas fronteiras externas, a expressão para veri…cação do balanço de massa incremental de óleo e água , realizada no passo de tempo $n+1$, é escrita da seguinte forma:

$$I_{BM_l}^{n+1} = \frac{\sum_{i=1}^{n_x}\sum_{j=1}^{n_y} \left[V_{b_{i;j}} \left(S_l^{n+1} \acute{A}^{n+1} / B_l^{n+1} - S_l^{n} \acute{A}^{n} / B_l^{n} \right)_{i;j} \right]}{\Delta t \sum_{i=1}^{n_x}\sum_{j=1}^{n_y} q_{lsc_{i;j}}^{n+1}} \qquad l = o; w \tag{3.104}$$

A expressão acima deve fornecer valores próximos à unidade.

Capítulo 4

Métodos para Solução de Sistemas Lineares

Em simulações de reservatórios, a maior parte do tempo gasto no processo de simulação ocorre na etapa de solução dos sistemas de equações lineares, e por isso é de fundamental importância empregar métodos de solução e...cientes capazes de resolver o problema em um tempo menor e utilizando pouca memória computacional, sem perder a acurácia na solução.

Existem dois tipos de métodos para solução de sistemas de equações lineares (TREFETHEN e BAU, 1997): métodos diretos e métodos iterativos. Os métodos diretos utilizam algorítmos de fatoração para encontrar a solução exata do sistema, enquanto os métodos iterativos utilizam um processo iterativo para obter uma solução aproximada para o sistema de equações.

Em muitos casos, métodos iterativos são mais adequados do que os métodos diretos para resolver problemas de larga escala em computadores paralelos, pois, além da facilidade de paralelização e implementação dos métodos iterativos nestas arquiteturas, estes são preferíveis por utilizarem menos memória computacional e ainda poderem resolver o sistema de forma aproximada, e com uma precisão adequada, em um tempo muito inferior ao tempo gasto utilizando métodos diretos.

Este capítulo trata, portanto, dos conceitos básicos relacionados aos métodos iterativos para solução de sistemas de equações lineares e técnicas de precondicionamento. Técnicas de precondicionamento são essenciais para acelerar o processo de convergência dos métodos iterativos, sendo discutidos também alguns aspectos sobre o uso de precondicionadores associados a estes métodos.

4.1 Métodos Iterativos e Subespaços de Krylov

Métodos iterativos (BARRET et al, 1994; SAAD, 1996; TREFETHEN e BAU, 1997; DEMMEL, 1997) são técnicas de solução de sistemas de equações lineares, baseados na obtenção de soluções aproximadas para o sistema através de iterações sucessivas, até atingir um critério de convergência determinado. Podem ser divididos em duas categorias: métodos estacionários e métodos não-estacionários.

Métodos estacionários são mais antigos e mais bem compreendidos. Atualmente, são pouco utilizados devido ao surgimento de algorítmos mais e...cientes, porém são ainda bastante úteis em alguns problemas, principalmente quando associados a outros métodos de solução. Alguns métodos estacionários discutidos a seguir são: Jacobi, Gauss-Seidel, SOR (Sucessive Over Relaxation) e Block-Jacobi.

Os métodos iterativos não-estacionários são mais recentes e, embora a teoria por trás destes algorítmos seja mais complexa, são normalmente mais e...cientes para a solução da grande maioria dos problemas práticos. Alguns métodos não-estacionários discutidos neste capítulo são: Generalized Minimum Residual (GMRES), Biconjugate Gradient (BiCG), Conjugate Gradient Squared (CGS) e Biconjugate Gradient Stabilized (BiCGSTAB).

4.1.1 Métodos Estacionários

Considere o sistema linear

$$Ax = b \tag{4.1}$$

onde A é uma matriz real $n \times n$, b e x são vetores coluna com n elementos. Admita que a matriz A possa ser escrita por $A = D - E - F$, onde D é a matriz formada unicamente pela diagonal principal de A, e as matrizes E e F são compostas pelos elementos localizados estritamente abaixo e estritamente acima da diagonal principal da matriz A, respectivamente.

Método de Jacobi

A i-ésima linha do sistema (4.1) pode ser escrita da seguinte forma:

$$\sum_{j=1}^{n} a_{i,j} x_j = b_i \tag{4.2}$$

No método iterativo de Jacobi, encontra-se uma solução aproximada para o componente x_i^{k+1} na iteração $k+1$, escrevendo-o em função dos demais elementos da iteração

anterior, k, chegando-se a:

$$x_i^{k+1} = \frac{1}{a_{i;i}} \left(b_i - \sum_{\substack{j=1 \\ j \neq i}}^{n} a_{i;j} x_j^k \right); \qquad i = 1::n \tag{4.3}$$

A eq. (4.3) pode ser escrita na forma matricial por:

$$x^{k+1} = D^{-1}(E + F)x^k + D^{-1}b \tag{4.4}$$

Método de Gauss-Seidel

O método de Gauss-Seidel é obtido a partir do método de Jacobi, onde agora os componentes do vetor solução que já foram calculados na iteração $k + 1$ são imediatamente utilizados no cálculo dos demais componentes desta mesma iteração, resultando na seguinte expressão:

$$x_i^{k+1} = \frac{1}{a_{i;i}} \left(b_i - \sum_{j=1}^{i-1} a_{i;j} x_j^{k+1} - \sum_{j=i+1}^{n} a_{i;j} x_j^k \right); \qquad i = 1::n \tag{4.5}$$

No método de Jacobi, observa-se que os elementos x_i^{k+1} podem ser calculados independentes da ordem em que cada elemento é calculado. No método de Gauss-Seidel a ordem em que o elemento x_i^{k+1} é calculado, segundo a eq (4.5), é importante pois seu valor depende dos elementos já calculados nesta mesma iteração. Normalmente o número de iterações no método de Gauss-Seidel é menor que no método de Jacobi.

Usando notação matricial, da eq. (4.5), tem-se:

$$Dx^{k+1} = Ex^{k+1} + Fx^k + b; \tag{4.6}$$

$$x^{k+1} = (D - E)^{-1} Fx^k + (D - E)^{-1} b. \tag{4.7}$$

Método SOR

O método SOR - Sucessive Over Relaxation é obtido a partir de uma sobre-relaxação do método de Gauss-Seidel. A idéia é obter uma média ponderada entre a iteração de Gauss-Seidel e a iteração anterior para cada componente, resultando em:

$$x_i^{k+1} = \omega \bar{x}_i^{k+1} + (1 - \omega) x_i^k \tag{4.8}$$

onde $\bar{x}_i^{k+1}$ é o componente da iteração de Gauss-Seidel, obtido pela eq. (4.5), e ω é o fator de sobre-relaxação. A escolha do fator ω adequado deve acelerar o processo iterativo. Se ω for igual a 1; a método SOR é igual ao método de Gauss-Seidel.

Escrevendo na forma matricial, tem-se:

$$x^{k+1} = (D - \omega E)^{-1} [\omega F + (1 - \omega) D] x^k + (D - \omega E)^{-1} \omega b \tag{4.9}$$

Método Block-Jacobi

O método Block-Jacobi é uma generalização do método de Jacobi, onde o esquema iterativo é aplicado a um bloco de elementos ao invés de um único elemento. As eqs. (4.3) e (4.4), podem ser escritas na forma blocada, simplesmente considerando que cada elemento do vetor agora é representado por um sub-vetor e cada elemento da matriz corresponde a uma sub-matriz. Os métodos de Gauss-Seidel e SOR também apresentam sua forma blocada.

Métodos estacionários blocados podem ser utilizados em problemas que envolvem mais de um grau de liberdade por bloco da malha, aumentando a e...ciência do processo iterativo em muitas aplicações práticas.

4.1.2 Métodos Não-Estacionários

Os métodos não-estacionários discutidos a seguir baseiam-se em técnicas de projeção sobre subespaços de Krylov. Antes de falar sobre os métodos não-estacionários, são introduzidas algumas noções sobre métodos de projeção e métodos de subespaços de Krylov.

Métodos de Projeção

Seja o sistema de equações lineares

$$Ax = b; \tag{4.10}$$

onde A é uma matriz real $n \times n$: A idéia dos métodos de projeção é extrair uma solução aproximada para o sistema (4.10), a partir de um subespaço de R^n com dimensão m, chamado subespaço de busca e representado por K. Para obter esta solução aproximada são necessárias em geral m restrições. Estas restrições são representadas através de condições de ortogonalidade impostas ao vetor de resíduos $r = b - Ax$ em relação a m vetores linearmente independentes, de...nindo assim o subespaço de restrições L, com dimensão m:

Sendo x^* o vetor solução aproximada para o sistema (4.10), um método de projeção estabelece o seguinte:

$$\text{Encontre } x^* \in K, \quad \text{tal que } b - Ax^* \perp L: \tag{4.11}$$

Sendo x_0 o vetor solução inicial do processo iterativo, o vetor x^* pode ser obtido a partir do subespaço a...m $x_0 + K$; tendo-se portanto:

$$\text{Encontre } x^* \in x_0 + K, \quad \text{tal que } b - Ax^* \perp L: \tag{4.12}$$

Escrevendo x^* em função da solução inicial como sendo $x^* = x_0 + \delta$, e definindo o vetor de resíduo inicial por $r_0 = b - Ax_0$; a condição da equação acima pode ser escrita por $b - A(x_0 + \delta) \perp L$. Definindo o novo resíduo por $r = r_0 - A\delta \perp L$, o método de projeção resulta em encontrar uma solução que atenda às seguintes condições:

$$x^* = x_0 + \delta; \quad \delta \in K \tag{4.13}$$

$$(r_0 - A\delta, w) = 0; \qquad \forall w \in L \tag{4.14}$$

onde (x, y) é a notação usada para o produto escalar entre dois vetores x e y:

O processo iterativo nos métodos de projeção prossegue a partir deste passo inicial, usando uma sucessão de projeções definidas por (4.13) e (4.14), gerando novos subespaços K e L, e fazendo x_0 igual ao vetor de solução aproximada mais recente.

Os métodos de projeção podem ser divididos em duas classes: métodos de projeção ortogonal e métodos de projeção oblíqua. A diferença entre estes métodos está na forma como o subespaço L é obtido. Em métodos de projeção ortogonal, L é igual a K, enquanto que nos métodos de projeção oblíqua L é diferente de K.

Métodos de subespaços de Krylov, descritos a seguir, utilizam conceitos de métodos de projeção para estabelecer os algorítmos iterativos de busca para a solução aproximada.

Métodos de Subespaços de Krylov

Considere o sistema de equações representado por $Ax = b$: Seja x_0 o valor inicial para o vetor solução do processo iterativo, e $r_0 = b - Ax_0$ o vetor de resíduo inicial do sistema. Um método de projeção, como visto anteriormente, é um método que busca uma solução aproximada x_m a partir de um subespaço afim $x_0 + K_m$ com dimensão m, através da imposição da condição de ortogonalidade $b - Ax_m \perp L_m$; sendo L_m outro subespaço de dimensão m.

Um método de subespaço de Krylov é um método onde K_m é o subespaço de Krylov formado por

$$K_m = \left\{ r_0; Ar_0; A^2 r_0; \ldots; A^{m-1} r_0 \right\}. \tag{4.15}$$

A dimensão do subespaço K_m cresce à medida que o processo iterativo avança.

Os diferentes métodos de subespaços de Krylov surgem a partir da escolha do subespaço L_m, como visto nos métodos de projeção. Os principais métodos iterativos beseados neste subespaços usam duas aproximações para o subespaço L_m; a primeira fazendo $L_m = K_m$ ou uma variação baseada em minimização residual $L_m = AK_m$, e

a outra fazendo $L_m = K_m^T$, onde K_m^T é o subespaço de Krylov relacionado à matriz transposta A^T.

Alguns métodos de subespaços de Krylov são descritos a seguir.

Generalized Minimum Residual - GMRES

O GMRES (SAAD e SCHULTZ, 1986) aplica-se na solução de sistemas de equações lineares não-simétricos. Este método consiste em um método de projeção baseado em tomar $L_m = AK_m$, onde K_m é o subespaço de Krylov de dimensão m, dado por:

$$K_m = \{v_1; Av_1; A^2v_1; \ldots; A^{m-1}v_1\}; \tag{4.16}$$

onde $v_1 = r_0/\|r_0\|_2$. Este método encontra, a cada iteração, o único vetor $x_m \in x_0 + K_m$ que minimiza a norma do resíduo, $\|b - Ax_m\|_2$.

No GMRES, é formada explicitamente uma base ortogonal para o subespaço K_m em função dos resíduos, usando a técnica de ortogonalização de Gram-Schidt, que quando aplicada a uma sequência de Krylov $\{A^k r_0\}$ é chamada método de Arnoldi (SAAD, 1996).

Como o processo de ortogonalização envolve longas recorrências, utiliza-se normalmente uma versão com reinício após m iterações, conhecida por GMRES(m). Isto reduz signi...cativamente a memória computacional utilizada no método, às custas do aumento do número de iterações.

O algorítmo do GMRES(m), está mostrado abaixo.

Algorítmo do GMRES

1. $r_0 = b - Ax_0$; $\beta = \|r_0\|_2$; $v_1 = r_0/\beta$;
2. De...na a matriz $(m+1) \times m$ por $\bar{H}_m = \{h_{i,j}\}_{(1 \le i \le m+1);(1 \le j \le m)}$; $\bar{H}_m = 0$;
3. For $j = 1; 2; \ldots; m$
4. $\quad w_j = Av_j$;
5. $\quad$ For $i = 1; \ldots; j$
6. $\quad\quad h_{i,j} = (w_j; v_i)$;
7. $\quad\quad w_j = w_j - h_{i,j}v_i$;
8. $\quad$ end;
9. $\quad h_{j+1,j} = \|w_j\|_2$; Se $h_{j+1,j} = 0$; $m = j$; (vá para 12.);
10. $\quad v_{j+1} = w_j/h_{j+1,j}$;
11. end;
12. $y_m = \arg\min_y \|\beta e_1 - \bar{H}_m y\|_2$; $x_m = x_0 + V_m y_m$;

Biconjugate Gradient - BiCG

Como o método tradicional dos Gradientes Conjugados é inadequado para sistemas não-simétricos, devido à impossibilidade de formar vetores de resíduos ortogonais com pequena recorrência, a idéia do BiCG é substituir a sequência de resíduos ortogonais por duas sequências mutuamente ortogonais.

O BiCG consiste em um método de projeção sobre o subespaço de Krylov

$$K_m = \left\{ v_1; Av_1; A^2v_1; \ldots; A^{m-1}v_1 \right\}; \tag{4.17}$$

onde o subespaço L_m é ortogonal a K_m e é escolhido como sendo igual a K_m^T; ou seja

$$L_m = K_m^T = \left\{ w_1; A^T w_1; \left(A^T\right)^2 w_1; \ldots; \left(A^T\right)^{m-1} w_1 \right\}. \tag{4.18}$$

Neste caso $v_1 = r_0/\|r_0\|_2$ e o vetor w_1 pode ser um vetor arbitrário qualquer, desde que $(v_1, w_1) \neq 0$. Portanto, a formulação deste método resulta implicitamente na solução de um outro sistema linear dual $A^T x^* = b^*$; onde o resíduo é dado por $r^* = b^* - A^T x^*$.

Utilizando o procedimento de biortogonalização de Lanczos (SAAD, 1996), que consiste na obtenção de bases biortogonais para os subespaços K_m e K_m^T, é possível escrever o algoritmo para o método BiCG, indicado abaixo.

Algorítmo do BiCG

1. $r_0 = b - Ax_0$; Escolha r_0^* tal que $(r_0, r_0^*) \neq 0$;
2. $p_0 = r_0$; $p_0^* = r_0^*$;
3. For $j = 0, 1, 2, \ldots$
4. $\quad \alpha_j = \langle r_j, r_j^* \rangle / \langle Ap_j, p_j^* \rangle$;
5. $\quad x_{j+1} = x_j + \alpha_j p_j$;
6. $\quad r_{j+1} = r_j + \alpha_j A p_j$;
7. $\quad r_{j+1}^* = r_j^* + \alpha_j A^T p_j^*$;
8. $\quad \beta_j = \langle r_{j+1}, r_{j+1}^* \rangle / \langle r_j, r_j^* \rangle$;
9. $\quad p_{j+1} = r_{j+1} + \beta_j p_j$;
10. $\quad p_{j+1}^* = r_{j+1}^* + \beta_j p_j^*$;
11. end;

Conjugate Gradient Squared - CGS

Este método foi desenvolvido para evitar o uso da matriz transposta A^T necessário no algorítmo do BiCG, buscando acelerar o processo de convergência sob mesmo custo

computacional. O idéia baseia-se em utilizar o quadrado dos polinômios da sequência de Krylov, $\{P_j^2(A)\, r_0\}$:

O algoritmo do CGS está indicado a seguir.

Algorítmo do CGS

1. $r_0 = b - Ax_0$; Escolha r_0^* qualquer;
2. $p_0 = u_0 = r_0$;
3. For $j = 0; 1; 2; \ldots$
4. $\quad \alpha_j = (r_j, r_0^*) / (Ap_j, r_0^*)$;
5. $\quad q_j = u_j - \alpha_j Ap_j$;
6. $\quad x_{j+1} = x_j + \alpha_j (u_j + q_j)$;
7. $\quad r_{j+1} = r_j - \alpha_j A(u_j + q_j)$;
8. $\quad \beta_j = (r_{j+1}, r_0^*) / (r_j, r_0^*)$;
9. $\quad u_{j+1} = r_{j+1} + \beta_j q_j$;
10. $\quad p_{j+1} = u_{j+1} + \beta_j (q_j + \beta_j p_j)$;
11. end;

O fato deste método calcular o quadrado dos polinômios residuais, resulta em maiores erros de truncamento, observando-se problemas de oscilação na norma do resíduo.

Biconjugate Gradient Stabilized - BiCGSTAB

O método BiCGSTAB é adequado para sistema não-simétricos e busca eliminar os problemas de convergência existentes no método CGS. Portanto, a sequência $\{P_j^2(A)\, r_0\}$ é substituída por $\{Q_j(A) P_j(A)\, r_0\}$, onde $P_j(A)$ é o polinômio residual associado ao algoritmo CGS e $Q_j(A)$ é um novo polinômio cujo objetivo é minimizar tais problemas de convergência.

Algorítmo do BiCGSTAB

1. $r_0 = b - Ax_0$; Escolha r_0^* qualquer;
2. $p_0 = r_0$;
3. For $j = 0; 1; 2; \ldots$
4. $\quad \alpha_j = (r_j, r_0^*) / (Ap_j, r_0^*)$;
5. $\quad s_j = r_j - \alpha_j Ap_j$;
6. $\quad \omega_j = (As_j, s_j) / (As_j, As_j)$;

7. $x_{j+1} = x_j + \alpha_j p_j + \omega_j s_j;$
8. $r_{j+1} = s_j - \omega_j A s_j;$
9. $\beta_j = \frac{(r_{j+1}, r_0^*)}{(r_j, r_0^*)} \frac{\alpha_j}{\omega_j};$
10. $p_{j+1} = r_{j+1} + \beta_j (p_j - \omega_j A p_j);$
11. end;

4.2 Técnicas de Precondicionamento

O processo de convergência de um método iterativo depende das propriedades do espectro da matriz do sistema. Transformando o sistema em outro equivalente e que apresente a mesma solução, porém apresentando propriedades espectrais mais adequadas, é possível acelerar a convergência do processo de solução. Esta transformação pode ser efetuada através de uma matriz chamada precondicionador.

4.2.1 Introdução

Voltando aos métodos estacionários, é possível escrever a seguinte expressão geral para o processo iterativo:

$$x^{k+1} = Gx^k + f \tag{4.19}$$

onde G e f correspondem à matriz e ao vetor de iteração, respectivamente.

Escrevendo a matriz A como $A = D - E - F$, para os métodos de Jacobi, Gauss-Seidel e SOR, tem-se:

$$G_J = D^{-1}(E + F); \qquad f_J = D^{-1}b$$

$$G_{GS} = (D - E)^{-1}F; \qquad f_{GS} = (D - E)^{-1}b$$

$$G_{SOR} = (D - \omega E)^{-1}[\omega F + (1 - \omega) D]; \qquad f_{SOR} = (D - \omega E)^{-1}\omega b.$$

Definindo uma nova partição para a matriz A; dada por $A = M - N$, verifica-se que as iterações de Jacobi, Gauss-Seidel e SOR apresentam a seguinte forma geral:

$$\begin{aligned} Mx^{k+1} &= Nx^k + b \\ &= (M - A)x^k + b \end{aligned} \tag{4.20}$$

$$x^{k+1} = M^{-1}(M - A)x^k + M^{-1}b \tag{4.21}$$

Relacionando as eqs. (4.19) e (4.21), tem-se:

$$G = I - M^{-1}A \tag{4.22}$$

$$f = M^{-1}b. \tag{4.23}$$

A iteração da eq. (4.19) busca resolver o sistema

$$(I - G)x = f; \tag{4.24}$$

que pode ser representado, usando as eqs. (4.22) e (4.23), por

$$M^{-1}Ax = M^{-1}b. \tag{4.25}$$

O sistema acima possui a mesma solução do sistema $Ax = b$, e é chamado sistema precondicionado associado à partição $A = M - N$, onde M é a matriz de precondicionamento ou precondicionador.

Aplicando-se os métodos de subespaços de Krylov para resolver o sistema (4.25), resulta na solução de um sistema precondicionado, cujo precondicionador pode ser obtido, por exemplo, a partir de um dos métodos estacionários. Da eq. (4.23), tem-se os seguintes precondionadores de Jacobi, Gauss-Seidel e SOR:

$$M_J = D \tag{4.26}$$

$$M_{GS} = D - E \tag{4.27}$$

$$M_{SOR} = \omega^{-1}(D - \omega E): \tag{4.28}$$

4.2.2 Formas de Precondicionamento

Existem duas formas de precondicionamento mais comuns: precondicionamento a esquerda e precondicionamento a direita. Na seção anterior foi mostrada a forma de precondicionamento a esquerda, representada pelo sistema $M^{-1}Ax = M^{-1}b$.

O sistema original, $Ax = b$; pode ser escrito incluindo os dois precondicionadores, através da seguinte forma geral:

$$M_L A M_R^{-1}(M_R x) = M_L^{-1}b \tag{4.29}$$

onde as matrizes M_L e M_R são chamadas precondicionador esquerdo e precondicionador direito, respectivamente. A solução do sistema original pode ser obtida aplicando-se alguns dos métodos iterativos vistos anteriormente ao sistema precondicionado (4.29), sendo necessárias, porém, algumas operaçõers adicionais nos algorítmos destes métodos, consistindo basicamente em fazer $r_0 = M_L^{-1}r_0$ no início do processo, e $x = M_R^{-1}x$ no …nal da cada iteração.

Normalmente o sistema precondicionado é resolvido escolhendo-se uma das formas de precondicionamento. Detalhes sobre vários precondicionadores e sobre sua implementação junto aos algorítmos dos métodos iterativos são discutidos por BARRET et al. (1994). A seguir são mostrados alguns destes precondicionadores.

Jacobi

O precondicionador Jacobi é o mais simples e é formado somente pela diagonal principal da matriz A, tendo-se: $M = \text{diag}(a_{11}; \ldots; a_{nn})$:

Block-Jacobi

O precondicionador block-Jacobi é uma generalização do precondicionador de Jacobi, onde agora os elementos da matriz A são submatrizes e a matriz M é uma matriz bloco diagonal formada pelos blocos da diagonal de A: $M = \text{diag}(A_{11}; \ldots; A_{nn})$: Este precondicionador é bastante adequado para soluções em computadores paralelos.

LU Incompleto

Uma grande variedade de precondicionadores baseia-se em técnicas de fatoração incompleta da matriz. Uma fatoração é chamada incompleta se, durante o processo de fatoração, alguns elementos diferentes de zero, localizados em posições onde antes havia um zero na matriz original, são ignorados. O precondicionador ILU é escrito na forma fatorada por M = LU, onde L e U são matrizes triangular inferior e superior, respectivamente.

O tipo de fatoração incompleta mais comum baseia-se em escolher um conjunto P contendo algumas posições da matriz, e manter todas as posições que não pertencem a este conjunto com o valor zero, durante a fatoração.

O conjunto P é normalmente escolhido de modo a envolver todas as posições $(i; j)$; cujos elementos $a_{i;j}$ sejam diferentes de zero. Uma posição é chamada posição de enchimento quando possui valor zero na matriz A e recebe um valor diferente de zero durante uma fatoração completa. Caso esta posição não pertença ao conjunto P, o valor calculado é descartado. Geralmente, P é escolhido de modo a coincidir com o conjunto das posições dos valores diferentes de zero na matriz A, descartando todo enchimento possível. Este tipo de fatoração é chamado ILU(0).

Variações sobre o tratamento do nível de enchimento da fatoração resultam em outros precondicionadores baseados nesta técnica, sendo descritos com detalhes por BARRET et al. (1994) e SAAD (1996).

Multigrid

A técnica usada para formar os precondicionadores descritos anteriormente independe do tipo de problema que origina o sistema de equações. Alguns precondicionadores, porém, são derivados a partir de informações sobre a estrutura do problema, e a técnica de multigrid utiliza esta estratégia. A idéia básica do precondicionador multigrid é usar informações da matriz referente a uma malha menos discretizada, para formar o precondicionador, transferindo as soluções da malha menos discretizada para a mais discretizada usando métodos de interpolação.

Decomposição de domínio

Métodos de decomposição de domínio também são utilizados para a obtenção de precondicionadores e…cientes que usam informações do problema analisado. Um precondicionador bastante conhecido é o Additive Schwarz (SAAD, 1996).

Capítulo 5

Noções sobre Computação Paralela

Este capítulo descreve alguns aspectos relevantes quando se trata de computação paralela, enfatizando o ambiente computacional destinado à aplicação do simulador desenvolvido neste trabalho. Mais detalhes a respeito de computação de alto desempenho são encontrados em FOSTER (1994), MINTY et al. (1996) e TOLEDO et al. (1997).

5.1 Introdução

A necessidade de resolver modelos cada vez mais complexos e realistas tem levado cientistas e pesquisadores por todo o mundo a buscarem técnicas de computação mais avançadas. O uso de computadores de alto desempenho tem possibilitado resolver problemas reais em diversas áreas das ciências e engenharias com um nível de detalhamento bastante elevado (MINTY et al., 1996).

Inicialmente, é necessário de...nir as principais características dos computadores sequenciais e paralelos (CSEP-Computer Archtecture, 1995).

5.1.1 Computadores Sequenciais

Os computadores sequenciais caracterizam-se por uma arquitetura simples, composta basicamente por uma unidade de processamento - processador - e pela memória, onde o processador executa sequencialmente um conjunto de instruções sobre os dados armazenados na memória. Alguns microcomputadores usados atualmente são exemplos de computadores sequenciais.

Pode-se aumentar o desempenho das máquinas sequenciais de duas formas: aumentando-se a velocidade do processador ou aumentando-se a taxa de transferência

(largura de banda) para o acesso à memória.

A técnica denominada pipelining foi uma das primeiras empregadas para aumentar o desempenho de um sistema computacional, e seu funcionamento assemelha-se a uma linha de montagem industrial, onde uma tarefa de produção é dividida em uma sequência de subtarefas, cada uma podendo ser executada simultaneamente com outras subtarefas que se encontram em execução em outros estágios desta linha de produção.

Atualmente, as memórias têm evoluído bastante criando uma verdadeira hierarquia. A hierarquia de memória está relacionada ao tempo de acesso que elas apresentam, aumentando-se este tempo à medida que cresce o tamanho da memória. Em ordem crescente de tempo de acesso, os diferentes tipos de memória encontrados nos computadores sequenciais são: registradores (na cpu), memória cache, memória principal (RAM), discos e …tas magnéticas. A criação destas hierarquias têm possibilitado reduzir o tempo de acesso à memória em códigos computacionais, com a otimização da distribuição dos dados entre os diferentes níveis de memória, durante a execução.

5.1.2 Computadores Paralelos

Computadores paralelos são máquinas compostas por um conjunto de processadores capazes de trabalhar em paralelo na solução de um problema computacional. O problema a ser resolvido é dividido entre os vários processadores, esperando–se que forneçam o mesmo resultado em um tempo de execução menor, de acordo com o número de processadores utilizados. Além da vantagem de reduzir o tempo de processamento em relação a um computador sequencial, o processamento paralelo também permite resolver problemas que necessitam muita memória para armazenamento.

A forma como determinado problema é decomposto em uma máquina paralela é fundamental para se alcançar um bom desempenho na sua solução. Existem várias formas para se paralelizar um código computacional, e a escolha entre uma delas depende das características do problema e do tipo de arquitetura da máquina utilizada. Além disso, podem ser empregados diferentes modelos de programação de acordo com a arquitetura escolhida.

Nas seções seguintes são mostradas algumas arquiteturas de computadores paralelos, os detalhes envolvidos na paralelização de códigos computacionais e uma descrição mais detalhada do modelo de programação especí…co para o ambiente computacional utilizado no desenvolvimento do simulador.

5.2 Arquiteturas de Computadores Paralelos

5.2.1 Taxonomia de Flynn

As máquinas paralelas atuais apresentam arquiteturas bastante variadas, podendo ser classi…cadas sob diferentes aspectos. Um esquema de classi…cação bastante usado para as arquiteturas paralelas é a conhecida Taxonomia de Flynn, que se baseia na noção de ‡uxo de informações. Segundo esta classi…cação, podem existir dois tipos de ‡uxos de informações nos processadores: ‡uxo de instruções e ‡uxo de dados. As máquinas são classi…cadas, segundo possuam uma ou mais linhas de ‡uxo de instruções ou de dados, da seguinte forma:

- SISD (única linha de instruções, única linha de dados): neste tipo de arquitetura o processador executa uma única instrução em uma única parte dos dados a cada instante. Os computadores sequenciais convencionais encontram-se nesta classi…cação.
- SIMD (única linha de instruções, múltiplas linhas de dados): vários processadores simples executam a mesma instrução sobre os dados que cada processador possui em sua memória local.
- MISD (múltiplas linhas de instruções, única linha de dados): este tipo de arquitetura é pouco utilizado em problemas de simulação computacional.
- MIMD (múltiplas linhas de instruções, múltiplas linhas de dados): estas máquinas possuem vários processadores independentes e interconectados, cada um capaz de executar individualmente suas próprias linhas de instruções ou seus próprios programas, que podem ser diferentes para cada um dos processadores. Esta classi…cação é a que apresenta uma maior variedade de arquiteturas, podendo ser subdividida, de acordo com a organização da memória, em arquiteturas com memória compartilhada e arquiteturas com memória distribuída, descritas a seguir.

Memória Compartilhada

Em arquiteturas com memória compartilhada, Figura 5.1(a), cada processador tem acesso a uma memória global única através de algum sistema de interconexão. Nestes computadores, os processadores acessam os dados diretamente através do endereço na memória global, não havendo necessidade de comunicação explícita entre eles. Um dos problemas relacionados a e…ciência nestas máquinas deve-se à limitação da velocidade

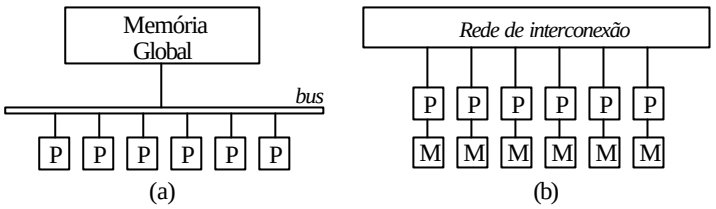

Figura 5.1: Arquiteturas MIMD. (a) memória compartilhada; (b) memória distribuída.

de acesso à memória global. Há também o problema de contenção, veri...cado à medida que vários processadores tentam acessar o mesmo endereço na memória global e somente um pode acessá-lo em cada instante, o que reduz a e...ciência da execução do código.

Memória Distribuída

Neste tipo de arquitetura paralela, Figura 5.1(b), cada processador tem acesso direto somente a sua memória local, não existindo uma memória global comum. A troca de informações entre processadores é realizada explicitamente usando o modelo de troca de mensagem, descrito adiante. Os processadores são interligados através de uma rede por onde ocorre toda a comunicação. A arquitetura de memória distribuída elimina os problemas com o acesso à memória global observados nas arquiteturas com memória compartilhada, porém surge outro problema referente ao acesso à memória remota, que é mais lento do que o acesso à memória local e depende ainda da distância lógica entre os processadores que trocam as mensagens (problema de comunicação).

5.2.2 Outras Classi...cações

Quanto ao número de processadores, as máquinas podem ser classi...cadas como máquinas de granularidade ...na, quando possuem grande número de processadores e o tempo de comunicação é alto em relação ao tempo de processamento, e máquinas de granularidade grossa, quando o número de processadores é menor, aumentando a relação tempo de processamento por tempo de comunicação.

As máquinas paralelas também são classi...cadas em função da topologia de conexão entre processadores. Algumas formas de conexão bastante comuns são: anel, malha 2D, crossbar, ligação por barramento e hipercubo. A Figura 5.2 ilustra estas topologias.

As máquinas podem ser classi...cadas ainda quanto à homogeneidade, sendo comum

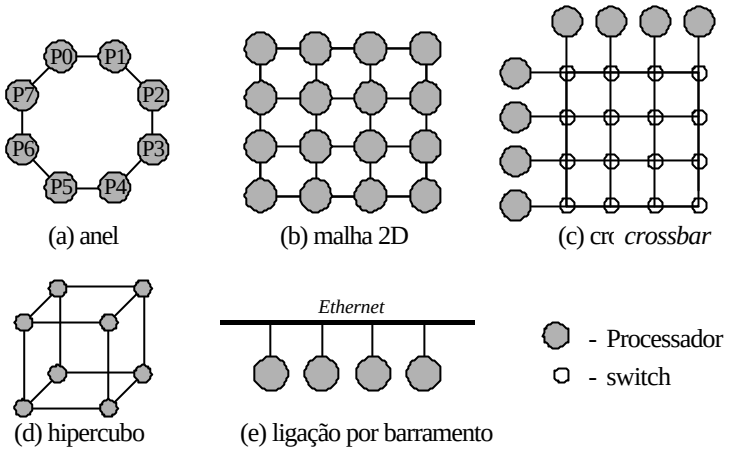

Figura 5.2: Algumas topologias de conexão entre processadores.

na computação paralela distribuída a associação de máquinas com diferentes capacidades computacionais, classi...cadas como máquinas heterogêneas.

5.3 Métodos para Paralelização de um Programa

Todo programa possui uma parte sequencial e uma parte que pode ser paralelizada. Um exemplo de um trecho sequencial em um código consiste nas etapas de leitura e impressão dos dados. Existem ainda algumas operações que são naturalmente sequenciais e não podem ser paralelizadas.

Existe uma variedade de formas através das quais pode-se decompor um código paralelizável em tarefas separadas para cada processador, visando reduzir o tempo de execução do programa. Os principais aspectos que devem ser considerados quando se deseja paralelizar um programa em arquiteturas de computadores com memória distribuída são descritos a seguir.

5.3.1 Partição do Programa

A partição de um programa está relacionada à forma como as etapas de um programa são distribuídas entre processadores, identi...cando-se as principais tarefas com potenciais de paralelismo e distribuindo-as para tentar reduzir o tempo de processamento. Um programa pode ser particionado usando os seguintes métodos: decomposição trivial, decomposição funcional e decomposição de domínio.

Decomposição Trivial

A decomposição trivial é a técnica mais simples e, ao invés de paralelizar o código, a idéia consiste em executar o programa em cada processador usando diferentes dados de entrada para cada um. Considerando que uma execução não dependa dos resultados de outra, cada processador pode trabalhar independentemente, não havendo necessidade de comunicação entre eles para produzir os resultados do programa.

Decomposição Funcional

A decomposição funcional consiste em dividir as tarefas de um programa em diversas subtarefas, cada uma sendo executada em diferentes processadores. Um exemplo deste tipo de decomposição é o pipeline. O paralelismo é introduzido à medida que os subprogramas percorrem o pipeline, funcionando do mesmo modo como uma de linha de montagem.

Decomposição de Domínio

Uma técnica de paralelização bastante comum é a técnica de decomposição do domínio (MINTY et al., 1996), cuja idéia está relacionada à distribuição dos dados de um programa entre os diversos processadores, de modo que cada processador execute operações somente sobre os dados que possui. O método de decomposição de domínio é um dos mais empregados na paralelização de códigos numéricos.

Considere, por exemplo, um problema numérico aplicado sobre uma malha regular. A aplicação da decomposição de domínio consiste em subdividir a malha em vários subdomínios, de modo que os dados referentes a cada subdomínio sejam distribuídos para cada processador, como ilustra a Figura 5.3. Cada processador então é encarregado de executar operações sobre os dados referentes a seus próprios subdomínios. É importante que a quantidade de dados seja distribuída da maneira mais uniforme possível, visando balancear a carga de processamento entre processadores.

5.3.2 Comunicação entre Processadores

A comunicação entre processadores está intrinsecamente relacionada a arquiteturas com memória distribuída. Nas decomposições trivial e funcional as tarefas são praticamente independentes, não havendo necessidade de troca de informações entre processadores durante a execução. Na decomposição de domínio, normalmente os processadores podem precisar de dados de outros processadores para realizar algumas operações, como

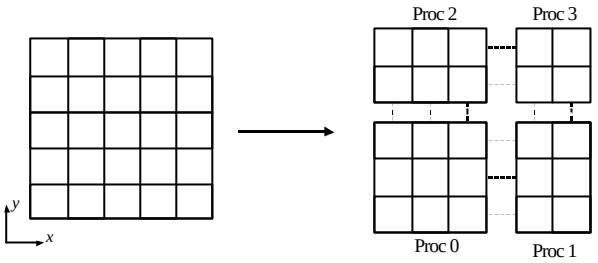

Figura 5.3: Malha regular 5x5 particionada entre 4 processadores - decomposição regular 2D.

no caso de problemas usando discretização por diferenças …nitas. Nestas situações, além dos dados locais, cada processador deve possuir uma cópia dos dados do contorno pertencentes a processadores vizinhos, chamados valores ocultos. A Figura 5.4 ilustra os dados locais do processador 0, incluindo os valores ocultos referentes ao problema de diferenças …nitas com um esquema de 5 pontos.

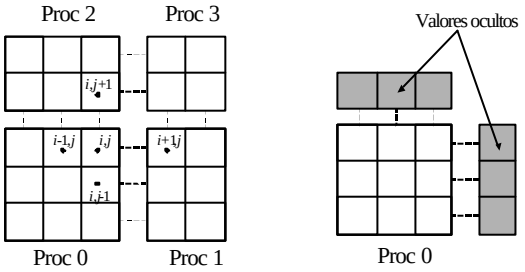

Figura 5.4: Malha global particionada e malha local do processador 0, incluindo valores ocultos referentes ao esquema de diferenças …nitas com 5 pontos.

Como os valores ocultos são cópias dos dados de outros processadores, é necessário comunicação para enviar estes dados de um processador a outro. O tempo gasto com esta comunicação reduz a e…ciência do código, sendo necessário estudar alternativas para minimizar este tempo.

Existem várias formas de comunicação entre processadores, podendo ser classi…-cadas como:

a) Comunicação local e global

A comunicação local ocorre quando um processador comunica-se diretamente com outro único processador. A comunicação local também é usada quando um processador

comunica-se somente com os processadores vizinhos. A comunicação global ocorre quando um processador comunica-se com vários outros processadores para realizar uma operação.

b) Comunicação estruturada e não-estruturada

A comunicação estruturada ocorre quando os processadores estão conectados regularmente seguindo uma orientação organizada, enquanto que na comunicação não-estruturada a rede formada pelos processadores pode apresentar uma estrutura qualquer, sem seguir uma ordem lógica.

c) Comunicação estática e dinâmica

Na comunicação estática o esquema de comunicação não se altera durante a execução do programa, enquanto que na comunicação dinâmica este esquema pode variar, como no caso de problemas com malhas adaptativas.

c) Comunicação sincronizada e não-sincronizada

A comunicação sincronizada ocorre de forma organizada, onde cada processador que participa do processo de envio e recebimento da mensagem só prossegue na execução do programa após a comunicação ter sido totalmente concluída. Na comunicação não-sincronizada esta organização não existe.

5.3.3 Efeito da Granularidade

Considere o problema da malha particionada usando decomposição de domínio, mostrada na Figura 5.4. A quantidade de valores ocultos que cada processador necessita depende da formulação das equações do problema e também da forma como os dados da malha são distribuídos entre os processadores. Em malhas com granularidade …na a relação entre o número de valores ocultos e o número de valores locais em um processador (comunicação/processamento) é elevada, necessitando portanto mais tempo de comunicação para atualização dos valores ocultos. Em malhas com granularidade grossa esta relação é menor e o código torna-se mais e…ciente.

O número de processadores utilizados no processamento de um programa com decomposição de domínio deve ser analisado do ponto de vista do desempenho da execução. Existe, um limite para o número de processadores utilizados onde o tempo gasto com comunicação se aproxima do tempo gasto com processamento. Apartir daí, o aumento do número de processadores reduz cada vez mais o desempenho da execução.

5.3.4 Mapeamento e Balanceamento de Carga

Programas que utilizam a técnica de decomposição do domínio precisam de uma estratégia para mapear e distribuir seus dados entre processadores, visando minimizar o tempo de execução do código. Em computadores com memória distribuída esta etapa é fundamental pois o mapeamento deve ser controlado pelo programador. O problema do mapeamento torna-se mais complexo em problemas com comunicação não-estruturada e dinâmica, devendo-se empregar técnicas específicas para o balanceamento de carga.

5.4 Medidas de Desempenho

Para medir o desempenho de programas em computadores paralelos (TOLEDO e SILVA, 1997) existem algumas medidas bastante utilizadas, cuja definição é dada a seguir.

- speedup (S_P): é a razão entre o tempo de execução do programa paralelo usando apenas um processador, T_1; e o tempo de execução do programa paralelo usando P processadores, T_P. Alguns autores definem speedup utilizando o tempo de execução T_1 do melhor algorítmo sequencial disponível para o problema.

$$S_P = \frac{T_1}{T_P} \tag{5.1}$$

Como a parte sequencial de um programa não pode ser paralelizada, seu efeito no speedup do algorítmo pode ser explicado pela Lei de Amdahl. Sendo ® a fração de operações correspondente à parte sequencial de um programa paralelo, a Lei de Amdahl estabelece a seguinte relação para o speedup:

$$S_P \cdot \frac{1}{® + (1 + ®)/P} \cdot \frac{1}{®} \tag{5.2}$$

Em muitos problemas computacionais, a fração ® tende a diminuir bastante à medida que o tamanho do problema aumenta. Nestes casos, a fórmula (5.2) perde seu significado e uma forma alternativa para esta expressão pode ser escrita por:

$$S_P \cdot (1 - ®)P \cdot ® \tag{5.3}$$

onde ® representa a fração do tempo da parte sequencial do programa paralelo quando usados P processadores.

- eficiência (E_P): é a razão entre o speedup e o número de processadores usados para obter o speedup.

$$E_P = \frac{S_P}{P} = \frac{T_1}{P\,T_P} \tag{5.4}$$

5.5 Paralelização de Algumas Operações Básicas

As principais operações envolvidas nos métodos iterativos, discutivos no Capítulo 4, e que consomem maior parte do tempo de processamento são: produto escalar, atualização de vetores, produto de matriz por vetor, aplicação de precondiciondores.

Para reduzir o tempo gasto com estas operações, foram implementadas subrotinas de algebra linear básica, conhecidas por BLAS (Basic Linear Algebra Subroutines). Estas subrotinas são encontradas em bibliotecas numéricas como o LAPACK (CSEP-Numerical Linear Algebra, 1995).

Visando tirar proveito da hierarquia de memória dos sistemas computacionais, através do armazenamento otimizado dos dados nos diferentes níveis de memória, foram desenvolvidos BLAS especí...cos para cada arquitetura de computador.

A seguir são discutidos alguns detalhes sobre a forma como algumas destas operações são tratadas visando obter mais e...ência no contexto dos métodos iterativos em computadores com memória distribuída.

Produto Escalar

O produto escalar $a = x:y$ entre dois vetores x e y pode ser realizado em um computador paralelo com memória distribuída, simplesmente fazendo com que cada processador efetue o produto escalar localmente sobre suas porções locais dos vetores. Em seguida, através de uma operação de comunicação global, estes produtos escalares locais podem ser somados fornecendo o resultado ...nal.

A última fase da operação de produto escalar requer comunicação global entre processadores, podendo gerar problemas de sincronização visto que os processadores não podem prosseguir até que a comunicação tenha sido concluída. Em alguns algorítmos de métodos iterativos são necessárias alterações visando minimizar este problema de sincronia. BARRET et al. (1994) discutem alternativas para reduzir tais problemas.

Atualização de Vetores

A atualização de vetores envolve operações simples entre vetores, por exemplo $y = ax + y$ ou $w = ax + by$, onde $x; y$ e w são vetores paralelos, a e b são escalares. Estas operações são facilmente paralelizáveis, visto que cada processador opera sobre sua própria porção local, não havendo necessidade de comunicação entre processadores.

Produto de Matriz por Vetor

A operação $y = Ax$ depende da estrutura da matriz, esparsa ou densa, e da forma como é armazenada. Considerando matrizes esparsas usando o esquema de armazenamento CSR - compressed sparse rows (SAAD, 1996), cada componente do vetor y é calculado através do produto escalar da i-ésima linha da matriz pelo vetor x:

Em computadores com memória distribuída, a matriz normalmente é particionada por linhas de modo que cada processador possua armazenado em sua memória local somente os elementos não-nulos das linhas contínuas que possui. Os elementos dos vetores são distribuídos seguindo o mesmo esquema de partição da matriz. Dependendo da largura de banda da matriz e do número de processadores utilizados, ao efetuar o produto escalar entre as linhas da matriz A e o vetor x, alguns processadores podem precisar da cópia de alguns elementos do vetor x que não pertencem a sua porção local, exigindo comunicação entre processadores.

5.6 Modelos de Programação Paralela

A escolha do modelo utilizado para o desenvolvimento de códigos computacionais em arquiteturas paralelas deve ser feita de acordo com o tipo de arquitetura utilizada. Os dois principais modelos de programação paralela são o modelo de message-passing e o modelo data-parallel (MINTY et al., 1996).

O modelo de programação data-parallel utiliza uma memória global que pode ser acessada diretamente por todos os processadores. Este modelo opera sobre os dados através de procedimentos de paralelização intrínsecos à linguagem de programação, fornecendo ao programador a responsabilidade de coordenar os processos e a troca de dados através destes procedimentos.

No modelo de programação com message-passing (troca de mensagem) cada processador só possui acesso direto a sua memória local, sendo utilizado em computadores com memória distribuída. Neste modelo o programador deve de…nir que tarefas serão executadas por cada processador e estabeler a sincronização e a troca de mensagens entre eles.

O modelo de programação com troca de mensagem foi o modelo utilizado para o desenvolvimento do simulador descrito neste trabalho e é descrito na seção seguinte, que mostra o funcionamento de bibliotecas paralelas para este modelo de programação.

5.7 Bibliotecas Paralelas

5.7.1 MPI - Message Passing Interface

O MPI (GROPP et al, 1999) é uma biblioteca de programação paralela desenvolvida para atender àqueles programas que buscam explorar a existência de vários processadores, realizando a comunicação entre eles através do modelo de troca de mensagem. Seu principal objetivo é fornecer uma interface de comunicação padrão para ser utilizada em aplicações que possam ser executadas em computadores paralelos com arquiteturas de memória distribuída, MIMD, e redes de estações de trabalho, constituindo-se numa ferramenta e…ciente e portável entre diferentes arquiteturas.

Esta biblioteca foi desenvolvida a partir de 1994, por um comitê formado por cientístas e pesquisadores de universidades e da indústria, que fundaram o MPI Forum (MPI Forum, 1994). O MPI pode ser utilizado em programas escritos em C, C++ ou Fortran.

São descritos a seguir alguns conceitos básicos sobre o modelo de programação com troca de mensagem usando o MPI, introduzindo-se algumas noções básicas sobre seu funcionamento (MacDONALD et al., 1999).

Conceitos Básicos

Um processo pode ser de…nido como uma entidade que se encarrega de realizar alguma tarefa computacional. No modelo do MPI, cada processo está associado a um processador e a uma única memória local, podendo trocar informações entre si.

Uma mensagem compreende os dados que são enviados na comunicação, sendo constituída pelo corpo e por outros dados adicionais contidos no envelope. A mensagem carrega as informações necessárias para que seja possível realizar uma comunicação, que é a troca de mensagens entre dois ou mais processos. A comunicação é realizada através de funções ou subrotinas desenvolvidas para este propósito especí…co.

Um grupo de processos corresponde a um conjunto de processos envolvidos em determinado trabalho, onde cada processo é identi…cado por um número iniciando por 0, chamado rank.

Um contexto de comunicação permite a criação de diferentes ‡uxos de mensagens independentes entre processos, cada um possuindo um contexto diferente. Um contexto garante que as mensagens enviadas em uma determinada fase de um programa não seja interceptada incorretamente em outra fase do programa.

O MPI fornece suporte para uma variedade de topologias, que compreende a forma

como os processos são arranjados buscando seguir a forma de conexão entre processadores. Uma topologia conveniente pode tornar o código mais e...ciente.

Normalmente, as comunicações com o MPI envolvem grupos de processos, contextos e topologias de comunicação. Os comunicadores são usados para de...nir as interfaces de comunicação. Toda comunicação no MPI envolve pelo menos um comunicador e todos os processos envolvidos numa comunicação devem compartilhar o mesmo comunicador.

Tipos de Dados no MPI

Sempre que forem efetuadas trocas de mensagens entre processos, o tipo de dado contido nas mensagens deve ser declarado como argumento da subrotina de comunicação. Os tipos de dados básicos do MPI estão relacionados aos mesmos tipos de dados básicos do C e do Fortran. A Tabela 5.1 mostra os tipos de dados básicos do MPI, usados em códigos Fortran.

MPI	Fortran
MPI_INTEGER	INTEGER
MPI_REAL	REAL
MPI_DOUBLE_PRECISION	DOUBLE_PRECISION
MPI_COMPLEX	COMPLEX
MPI_LOGICAL	LOGICAL
MPI_CHARACTER	CHARACTER(1)
MPI_BYTE	
MPI_PACKED	

Tabela 5.1: Relação entre tipos de dados do MPI com o Fortran

O MPI permite ainda que sejam formados tipos de dados derivados, ou seja, pode-se agrupar dados de diferentes tipos em um novo tipo de dado criado pelo programador. Com isso, é possível agrupar vários dados de diferentes tipos e enviá-los de uma só vez através de uma única mensagem, reduzindo o tempo de comunicação evitando o envio de várias mensagens para cada tipo de dado independente.

Existem dois tipos de comunicação no modelo de passagem de mensagem: a comunicação ponto a ponto (local) e a comunicação coletiva (global).

Comunicação Ponto a Ponto

Uma comunicação ponto a ponto envolve exatamente dois processos, um que envia a mensagem e o outro que a recebe. Os vários modos de comunicação ponto a ponto existentes são: standard, synchronous, bu¤ered e ready. A diferença entre estes modos está na forma como cada mensagem é enviada por um processo. A Tabela 5.2 mostra as principais rotinas de envio e recebimento de mensagens.

subrotinas	Modo
MPI_SEND()	standard
MPI_SSEND()	synchronous
MPI_BSEND()	bu¤ered
MPI_RSEND()	ready
MPI_RECV()	

Tabela 5.2: Subrotinas de comunicação ponto a ponto

Quando uma mensagem é enviada no modo standard, signi…ca que o envio foi concluído porém nada se pode a…rmar se a mensagem foi recebida ou não pelo processo de destino.

No modo synchronous o processador que envia a mensagem só retorna quando tiver certeza de que o processo que irá recebê-la tenha recebido a mensagem.

Quando a mensagem é enviada no modo bu¤ered, o envio da mensagem é concluído imediatamente, enviando a mensagem ao bu¤er para uma futura transmissão se necessário.

Da mesma forma que no modo bu¤ered, no modo ready o envio da mensagem é concluído imediatamente, mas agora a mensagem é simplesmente lançada na rede podendo ser recebida ou não pelo processo de destino.

A subrotina de recebimento das mensagens MPI_RECV() é a mesma para todos os modos descritos, e só é concluída quando a mensagem for totalmente recebida.

Os modos de comunicação descritos acima são chamados blocados. Para cada um desses modos existe a versão não-blocada. Uma subrotina de envio de mensagem blocada não retorna até que os dados especi…cados no envio possam ser reutilizados sem que seja causado nenhum dano à mensagem. No modo não-blocado a subrotina de envio da mensagem não espera pela ocorrência de nenhum evento até o retorno da mensagem. O recebimento das mensagens também pode ser blocado ou não-blocado.

Comunicação Coletiva

Subrotinas de comunicação coletiva destinam-se a comunicações que envolvem um grupo de processos. Todas as comunicações coletivas no MPI são consideradas blocadas. As principais subrotinas de comunicação coletiva incluídas no MPI são descritas a seguir.

A subrotina coletiva mais simples é `MPI_barrier()`, utilizada para sincronizar os processos em um comunicador. Esta subrotina faz com que o processo seja interrompido até que os demais processos do comunicador tenham chamado esta subrotina.

Algumas subrotinas básicas usadas para a distribuição de dados entre os processos de um comunicador, sem realizar nenhuma operação sobre estes dados, são: `MPI_Bcast()`, `MPI_Scatter()`, `MPI_Gather()`, `MPI_AllGather()` e `MPI_AllToAll()`, cujo funcionamento está ilustrado na Figura 5.5.

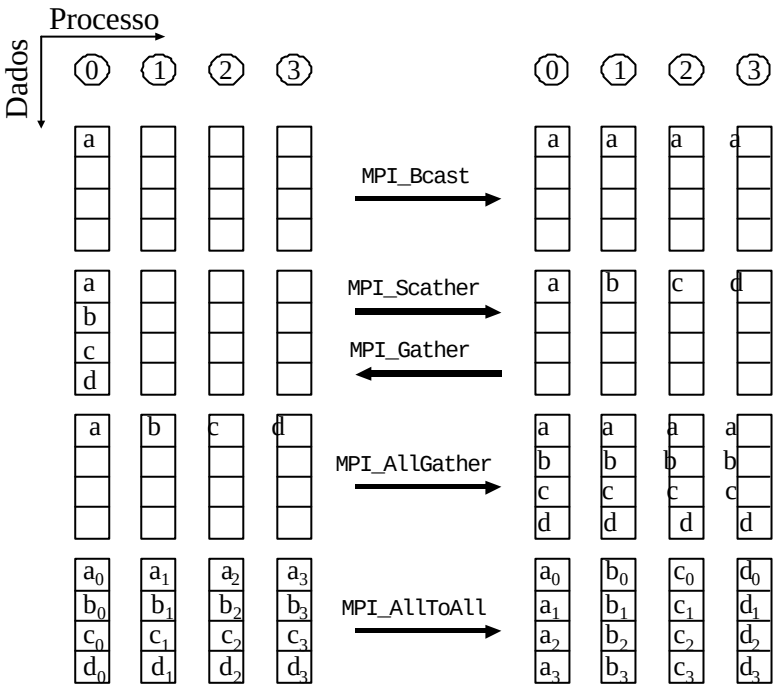

Figura 5.5: Esquema de comunicação das subrotinas coletivas.

Segundo esta …gura, a subrotina `MPI_Bcast()` envia os dados de um único processo para todos os demais processos do comunicador.

Subrotinas que realizam determinadas operações sobre os dados e os distribui para outros processos são chamadas subrotinas de redução global. As subrotinas `MPI_Reduce()` e `MPI_AllReduce()` são responsáveis por efetuar estas operações de redução. A Tabela 5.3 mostra alguns operadores utilizados por estas subrotinas.

Operador	Descrição
MPI_MAX	máximo
MPI_MIN	mínimo
MPI_SUM	somatório
MPI_PROD	produtório
MPI_LAND	AND lógico
MPI_BAND	AND em termos de bits
MPI_LOR	OR lógico
MPI_BOR	OR em termos de bits
MPI_MAXLOC	máximo e posição do máximo
MPI_MINLOC	mínimo e posição do mínimo

Tabela 5.3: Operações de redução global

5.7.2 PETSc - Portable, Extensible Toolkit for Scientific Computing

O PETSc (BALAY et al., 1997) é uma biblioteca de subrotinas numéricas paralelas para o desenvolvimento de códigos computacionais escaláveis, destinados à solução de sistemas de equações diferenciais parciais lineares ou não-lineares, dependentes do tempo. O PETSc foi desenvolvido para auxiliar no desenvolvimento de códigos paralelos para aplicação em computadores com arquiteturas de memória distribuída.

Este software busca tratar de forma eficiente, através de uma interface simples e uniforme, os detalhes de baixo nível envolvidos nas hierarquias de memória distribuída, utilizando outras bibliotecas como MPI, BLAS e LAPACK como base para fornecer uma interface portável entre diversas máquinas paralelas. Programas escritos em C, C++ ou Fortran 77/90 podem usar esta biblioteca.

Algumas das principais funcionalidades do PETSc são: manipulação e operações com vetores e matrizes paralelos, diferentes formatos para o armazenamento de matrizes esparsas, ferramentas de partição e gerenciamento de malhas regulares, solvers lineares, não-lineares e dependentes do tempo, vários métodos de subespaços de Krylov e precondicionadores implementados, ferramentas de visualização, entre outras.

O PETSc foi desenvolvido na linguagem C usando técnicas de programação orientada a objetos. Seus componentes estão organizados hierarquicamente, como mostra a Figura 5.6. Cada componente é representado por um objeto, sendo mostrados na Figura 5.7 alguns detalhes destes objetos.

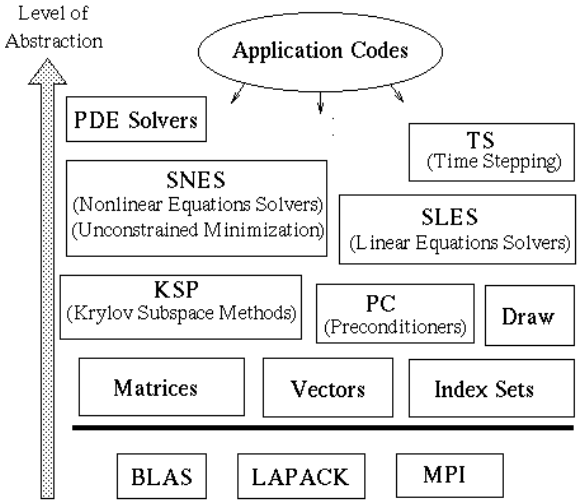

Figura 5.6: Organização dos componentes do PETSc (BALAY et al., 2001).

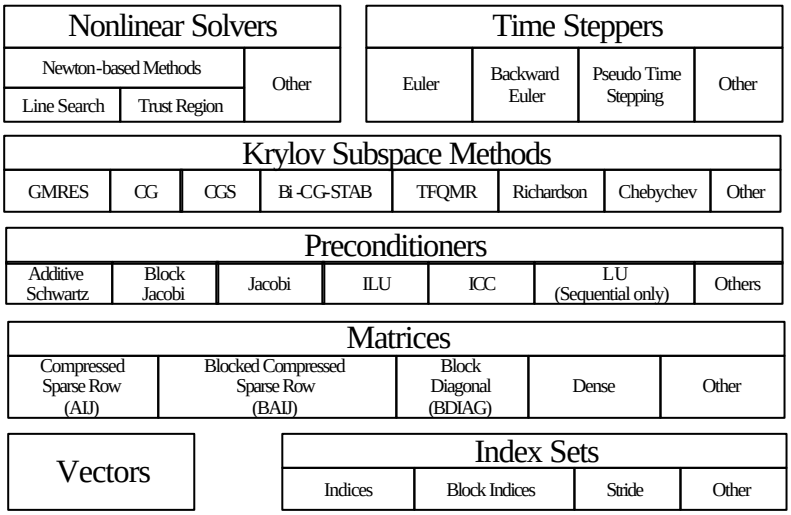

Figura 5.7: Detalhes dos objetos do PETSc (BALAY et al., 2001).

Para o desenvolvimento do simulador foram usados, basicamente, os seguintes componentes: vetores, matrizes, visualizadores, ferramentas de particionamento para malhas regulares e solvers lineares (métodos de subespaços de Krylov e precondicionadores) e não-lineares (método de Newton Inexato).

Capítulo 6

Implementação do Simulador

Neste capítulo são mostradas as características do simulador Black-Oil bifásico óleo/água, desenvolvido visando sua aplicação em computadores sequenciais e paralelos com memória distribuída. As formulações matemática e numérica descritas nos Capítulos 2 e 3 foram utilizadas na implementação computacional.

6.1 Características do Simulador

O simulador permite resolver problemas de fluxo bidimensional em reservatórios de petróleo usando o modelo Black-Oil bifásico óleo/água. O sistema de equações diferenciais parciais não-lineares que regem o problema é discretizado usando o método das diferenças finitas aplicado a uma malha de blocos centrados. Foram implementadas as formulações totalmente implícita e IMPES, podendo-se optar por uma das duas para a solução do problema.

6.1.1 Malha de Blocos Centrados

A geometria do reservatório é aproximada por uma malha retangular bidimensional, segundo as direções x e y, onde os blocos são numerados usando um esquema de numeração natural por linhas. Reservatórios com contorno irregular podem ter sua geometria aproximada como mostra a Figura 6.1.

No reservatório da Figura 6.1, o contorno irregular é aproximado pelo contorno definido pelas linhas escuras, separando os blocos ativos - que são utilizados efetivamente na modelagem - dos blocos inativos. Os blocos inativos funcionam como barreiras para o fluxo e são levados em conta fornecendo-se porosidades nulas a estes blocos.

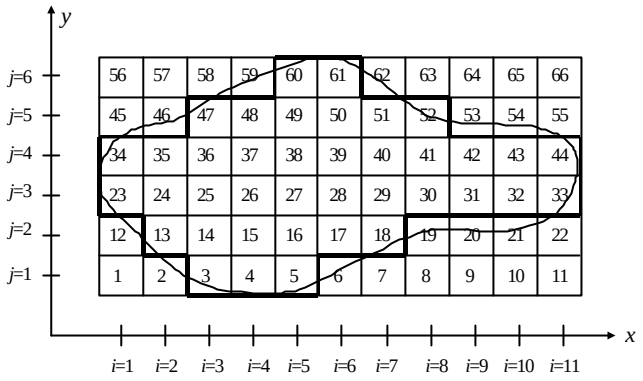

Figura 6.1: Malha retangular de blocos centrados utilizada na aproximação da geometria de um reservatório com contorno irregular.

Todos os blocos da malha retangular, incluindo blocos inativos, são numerados sequencialmente como mostra a Figura 6.1. Uma consequência desse esquema de numeração é que, em relação à montagem, a matriz do sistema possui sempre a mesma estrutura pentadiagonal ou bloco-pentadiagonal, sendo atribuído valor nulo aos elementos referentes aos blocos inativos localizados fora da diagonal principal, e valor unitário aos elementos da diagonal principal da matriz. Este esquema de numeração simpli…ca a implementação da subrotina de montagem da matriz em paralelo, mas por outro lado, elimina a possibilidade de redução do número de equações do problema, como seria o caso se fossem numerados apenas os blocos ativos da malha (ERTEKIN et al., 2001).

6.1.2 Dados de Entrada

Os dados de entrada do modelo são lidos a partir de um arquivo de entrada, cujo nome é fornecido no momento da execução do programa. Os principais dados fornecidos neste arquivo são:

Geometria:

- número de blocos na duas direções, n_x e n_y;
- dimensões dos blocos, $¢x_{i;j}$; $¢y_{i;j}$ e $¢z_{i;j}$;
- profundidade da face superior dos blocos, $Z_{i;j}$;

Propriedades da rocha:

- compressibilidade da rocha, c_r;
- permeabilidades absolutas, $k_{x_{i;j}}$ e $k_{y_{i;j}}$;
- porosidade da rocha, $Á_{i;j}$;

Propriedades dos ‡uidos:

- compressibilidade do óleo e da água, c_o e c_w;
- valores de referência de densidade, fator volume de formação e viscosidade do óleo e da água: $½_o^{ref}$, $½_w^{ref}$, B_o^{ref}, B_w^{ref}, $¹_o^{ref}$, $¹_w^{ref}$;
- pressão de referência, p^{ref};

Tabelas:

- propriedades dos ‡uidos: p, $½_o(p)$, $½_w(p)$, $B_o(p)$, $B_w(p)$, $¹_o(p)$, $¹_w(p)$;
- propriedades rocha-‡uido: S_w, $k_{ro}(S_w)$, $k_{rw}(S_w)$, $P_{cow}(S_w)$;

Condições iniciais:

- profundidade de referência para inicialização de pressões e saturações, Z^{ref};
- pressão de óleo inicial p_o^0 no reservatório ou pressão de óleo na profundidade de referência, p_o^{ref};
- saturação de água inicial S_w^0 no reservatório ou saturação de água na profundidade de referência, S_w^{ref};

Tempo de simulação:

- tempo de simulação máximo, t_{max};
- intervalo de tempo inicial, $¢t$;

Dados de poços:

- número de poços, n_p;
- índices dos poços, tipo de condição de contorno, valor prescrito, raio do poço, fator de skin, fator geométrico do poço: $i_p; j_p; tipo; val; s; G_w$:

O limite do número de blocos nas direções x e y depende somente da memória computacional disponível. Os blocos podem apresentar dimensões variadas nas direções x, y e z, podendo-se fornecer ainda diferentes profundidades para cada um. As profundidades lidas no arquivo de entrada, $Z_{i;j}$; são acrescidas do valor $¢z_{i;j}=2$ pelo simulador, para levar em conta a profundidade em relação ao centro do bloco.

Propriedades da rocha, como porosidade e permeabilidades absolutas, podem assumir diferentes valores em cada bloco da malha. As porosidades nos blocos variam em função das pressões de óleo no reservatório, de acordo com a eq. (2.1), e as permeabilidades absolutas permanecem constantes durante a simulação.

Os valores das propriedades dos fluidos podem ser calculados em função dos dados tabelados ou usando as equações de estado descritas na seção 2.1.1. Permeabilidades relativas e pressões de capilaridade são fornecidas em formas de tabela no arquivo de entrada. Todos os dados tabelados são calculados, durante a simulação, usando interpolação linear e algorítmo de busca sequencial.

As condições iniciais no reservatório são introduzidas fornecendo-se valores iniciais constantes para pressões de óleo e saturações de água em todo reservatório, ou fornecendo-se as pressões de óleo e saturações de água a uma profundidade de referência, utilizando o algorítmo de inicialização descrito na seção 3.3.1. A escolha da forma de inicialização é feita no arquivo de entrada.

As condições de poço implementadas estão descritas na seção 3.2.3. O fator geométrico de cada poço, G_w, é calculado utilizando a fórmula (3.28), ou pode ser fornecido diretamente no arquivo de entrada.

6.1.3 Impressão e Visualização dos Resultados

Os resultados da simulação podem ser impressos em arquivos de saída ou visualizados graficamente, durante a execução. Os resultados são impressos nos arquivos de saída a cada de passo de tempo especificado. Existem três arquivos de saída: arquivo de resultados nos blocos da malha, arquivo de resultados nos poços e arquivo contendo informações gerais da simulação. Os dados fornecidos em cada arquivo são:

- Resultados nos blocos: pressões de óleo e água, saturações de óleo e água, pressões de capilaridade, permeabilidades relativas, porosidade e propriedades dos fluidos;
- Resultados nos poços: vazões de óleo e água, pressão no poço, pressões e saturações no bloco do poço;
- Informações gerais: volume total e volume poroso do reservatório, porosidade média, permeabilidades absolutas médias, espessura média dos blocos, balanço de massa de óleo e água incrementais, segundo a seção 3.5.2, volumes totais de óleo e água produzidos, pressões de óleo e saturações de água máximas, mínimas e médias no reservatório.

É possível ainda visualizar gra…camente os campos de pressões de óleo e saturações de água, a cada passo de tempo. Para isto, é utilizada a ferramenta de visualização do PETSc.

6.1.4 Paralelização do Simulador

A paralelização do simulador baseou-se no método de decomposição de domínio. Todos os dados referentes aos blocos da malha são distribuídos entre processadores, sendo empregado o modelo de programação com troca de mensagem para realizar toda comunicação entre processos. Para isto, foram utilizados no desenvolvimento do simulador os softwares MPICH-1.2.3 (GROPP et al., 1996) e PETSc-2.1.0 (BALAY et al., 2001).

Para executar o simulador é necessário usar o comando padrão do MPI, `mpirun`, descrito abaixo:

```
mpirun -np <> oa2D [opções]
```

onde

`mpirun` é o comando de execução do MPI;

`oa2D` é o nome do arquivo exetutável do simulador;

`-np <>` fornece o número total de processadores envolvidos;

[opções] abrange todas as opções disponíveis pelo PETSc, e algumas opções adicionais implementadas no simulador, que são basicamente:

`-arq <>` fornece o nome do arquivo de entrada;

`-fully` utiliza a formulação totalmente implícita para solução do problema (a opção padrão é a IMPES);

`-result` ativa a opção de impressão de resultados em arquivo;

`-visual` ativa a opção de visualização grá…ca dos campos de pressões e saturações;

`-tempo` ativa a subrotina de controle do passo de tempo;

`-npx <>` fornece o número de processadores na direção x; para particionamento da malha;

`-npy <>` fornece o número de processadores na direção y; para particionamento da malha;

`-tmax <>` fornece o tempo de simulação total, substituindo o valor fornecido no arquivo de entrada.

6.2 Etapas da Simulação

Para o desenvolvimento do simulador paralelo foi necessário identificar as etapas paralelizáveis. A Figura 6.2 ilustra o fluxograma principal do simulador, e as Figuras 6.3 e 6.4 mostram os detalhes da implementação de cada formulação.

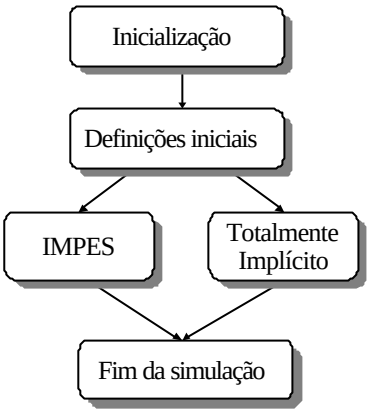

Figura 6.2: Fluxograma principal do simulador.

Os detalhes de cada uma destas etapas são descritas a seguir, mostrando-se como foram implementadas e as subrotinas mais importantes envolvidas (BALAY et al., 2001).

6.2.1 Inicialização do Programa

Esta etapa envolve a inicialização das bibliotecas paralelas MPI e PETSc através da subrotina `PetscInitialize()`. Com isto, é possível utilizar as subrotinas destes softwares.

6.2.2 Definições Iniciais

Inicialmente, é definido o comunicador padrão do PETSc, chamado `PETSC_COMM_WORLD`. Este comunicador refere-se ao comunicador global do MPI, chamado `MPI_COMM_WORLD`, que inclui o grupo com todos os processos envolvidos. O número total de processadores, `size`, e o `rank` de cada um dentro do comunicador são obtidos com as subrotinas `MPI_Comm_Size()` e `MPI_Comm_Rank()`.

O processo master é identificado pelo `rank = 0`, e também participa do processamento sobre os dados da malha da mesma forma que os demais processos.

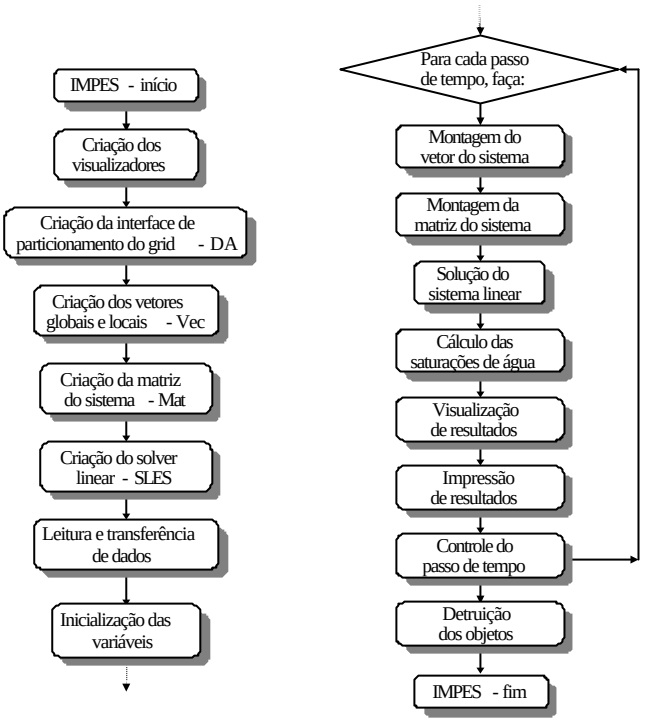

Figura 6.3: Fluxograma da implementação da formulação IMPES.

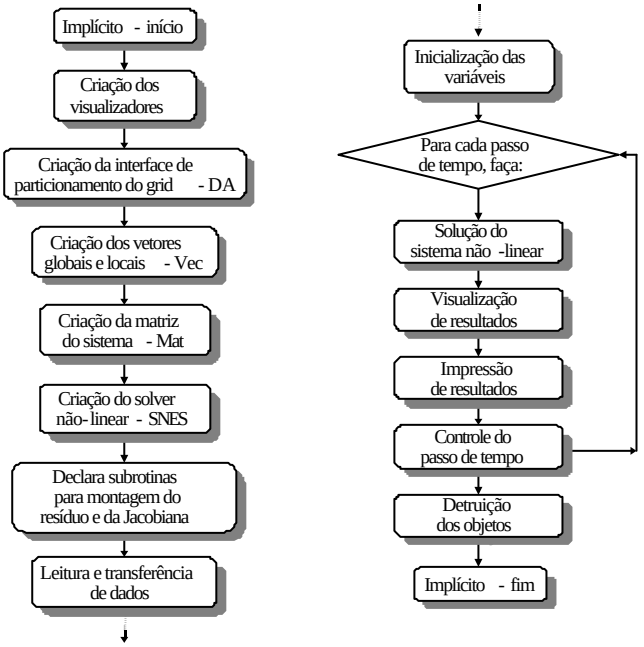

Figura 6.4: Fluxograma da implementação da formulação totalmente implícita.

Nesta etapa, o arquivo de entrada de dados é acessado pelo processo master e o número de blocos da malha nas duas direções, n_x e n_y; são lidos e transferidos para os demais processos usando MPI_Bcast(). A leitura destes dados é necessária pois serão utilizados nas etapas seguintes para alocar espaço na memória para vetores e matriz do sistema.

São definidos ainda os parâmetros padrões iniciais do programa, que são: tolerâncias e limites de convergência usados nos solvers lineares e não-lineares, fatores de conversão de unidades, entre outros.

Alguns destes parâmetros iniciais podem ser modificados usando opções de linha de comando, como as mostradas na seção 6.1.4. Para implementar estas opções foram usadas as seguintes subrotinas: PetscOptionsHasName(), PetscOptionsGetString(), PetscOptionsGetInt() e PetscOptionsGetDouble().

6.2.3 Criação da Interface de Visualização Gráfica

Esta etapa só é realizada se a opção de visualização gráfica dos resultados tiver sido ativada com a opção de linha de comando. São criados objetos do PETSc chamados visualizadores, através dos quais é possível visualizar os dados dos vetores paralelos em janelas gráficas. As subrotinas básicas necessárias para a criação dos visualizadores são: PetscViewerDrawOpen() e PetscViewerDrawGetDraw().

6.2.4 Criação da Interface de Particionamento da Malha

O particionamento da malha regular consiste em obter todas as informações necessárias para distribuição de dados e gerenciamento de comunicação entre processos. O PETSc possui um objeto para realizar estas tarefas, chamado distributed array - DA, específico para problemas que usam malhas retangulares logicamente regulares, onde é necessário comunicação de dados antes de ser efetuado algum processamento sobre os dados da malha.

Os DAs utilizam o método de decomposição de domínio para partionar a malha, obtendo informações sobre a forma de distribuição dos dados e comunicação entre processos, levando em conta a localização dos valores ocultos referentes a cada malha local.

Para a malha bidimensional tratada neste problema, a interface do DA é criada com a subrotina DACreate2d(). Entre as informações fornecidas para criar esta estrutura de dados, tem-se: nome do comunicador, dimensões da malha, número de processadores

para partionamento da malha em cada direção, esquema de diferenças ...nitas utilizado e número de graus de liberdade por bloco.

Para a formulação IMPES foi necessário criar um único objeto DA, considerando a malha com apenas 1 grau de liberdade por bloco. Na formulação totalmente implícita foram criados dois objetos DA, com 1 e 2 graus de liberdade. Estas informações são usadas para a criação dos vetores e matriz paralelos.

6.2.5 Criação dos Vetores Globais e Locais

Considerando uma malha com $N = n_x n_y$ blocos, os vetores necessários para armazenar dimensões, profundidade e propriedades da rocha nos blocos possuem todos dimensão N.

Além destes vetores, são necessários vetores para armazenar as pressões de óleo e saturações de água, referentes aos passos de tempos atual, $n + 1$; e anterior, n, e para armazenar os elementos dos termos independentes do sistema linear ou não-linear. A dimensão destes vetores variam de acordo com a formulação considerada.

Na formulação IMPES os vetores utilizados são p_o, p_o^n, S_w, S_w^n e b, sendo omitido o índice do passo de tempo atual. Estes vetores possuem dimensão N. Na formulação totalmente implícita os vetores utilizados são X, X^n e R, e possuem dimensão $2N$:

A Tabela 6.1 resume os vetores necessários em cada formulação e o número de elementos de cada um.

Dimensão	IMPES	Totalmente Implícita
N	$p_o; p_o^n; S_w; S_w^n; b$ ¢x; ¢y; ¢z; Z Á; k_x; k_y	¢x; ¢y; ¢z; Z Á; k_x; k_y
$2N$		$X; X^n; R$

Tabela 6.1: Vetores criados e suas respectivas dimensões

As informações obtidas com os objetos DAs permitem criar dois tipos de vetores: um vetor global paralelo, cujos dados são distribuídos entre processos, e um vetor local sequencial, que inclui espaço para os valores ocultos. A criação destes vetores e a alocação de espaço em memória para armazenar seus elementos é realizada com as subrotinas `DACreateGlobalVector()`, para os vetores globais, e `DACreateLocalVector()`, para os vetores locais.

A Figura 6.5 ilustra uma malha regular 5x5 particionada entre 4 processos, 2 em

cada direção, sendo mostrados dois esquemas de numeração global: uma numeração natural e uma numeração usada pelo PETSc para gerenciar os dados da malha.

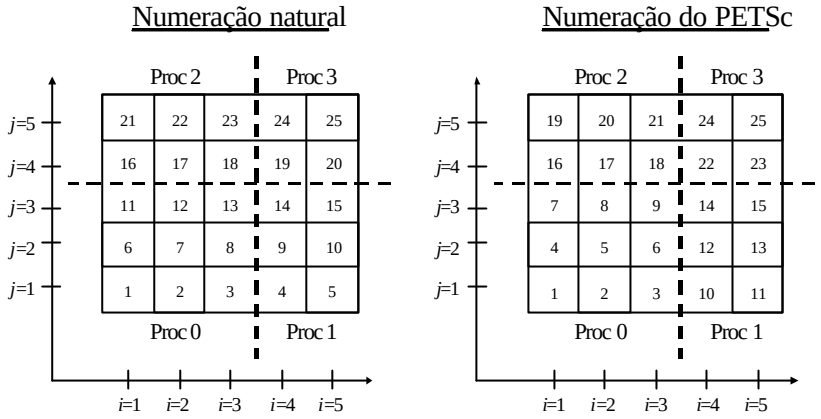

Figura 6.5: Malha 5x5 particionada entre 4 processos - numeração natural e numeração do PETSc.

O vetor global e os vetores locais a cada processo, incluindo valores ocultos, estão ilustrados na Figura 6.6. O acesso aos vetores locais permite que cada processo realize operações sobre seus dados locais, percorrendo os índices globais de seu subdomínios.

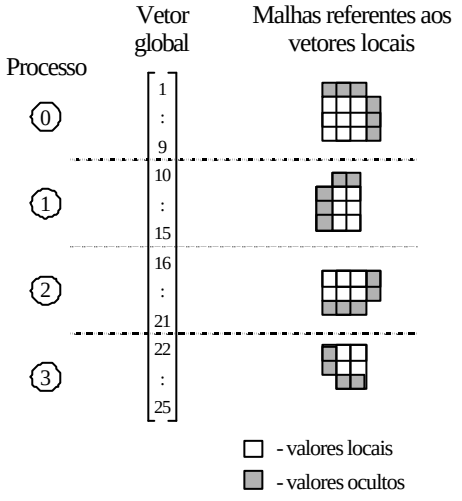

Figura 6.6: Vetor global paralelo e vetores locais incluindo valores ocultos.

6.2.6 Criação da Matriz

A matriz é particionada entre processos de modo que cada um armazene somente os elementos contidos no conjunto de linhas contínuas a que lhe pertence. O número de linhas de cada processo deve ser igual ao número de elementos da porção local dos vetores globais. O comando VecGetLocalSize() obtém o tamanho da porção local dos vetores. A Figura 6.7 ilustra a matriz particionada referente à malha da Figura 6.5.

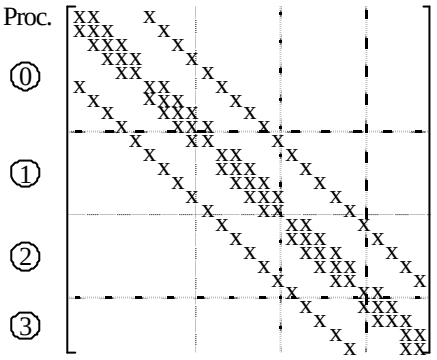

Figura 6.7: Matriz partionada entre 4 processadores.

A criação da matriz envolve a alocação de espaço na memória, e é realizada usando as seguintes subrotinas:

MatCreateSeqAIJ(), em problemas com 1 processador, e

MatCreateMPIAIJ(), em problemas com mais de um processador.

O PETSc possui vários formatos para armazenamento da matriz esparsa (BALAY et al., 2001), tendo sido utilizado o formato CSR (compressed sparse row). Neste formato, são armazenados somente os elementos diferentes de zero, sendo necessários três vetores para guardar as informações: um vetor contendo os elementos diferentes de zero ordenados por linha, um com os respectivos índices das colunas destes elementos e outro contendo apontadores para o início de cada linha no primeiro vetor. SAAD (1996) descreve outros formatos de armazenamento para matrizes esparsas.

As informações necessárias para criação da matriz esparsa são: numero de linhas e colunas da matriz global, numero de linhas e colunas da porção local da matriz e número de elementos diferentes de zero por linha.

6.2.7 Criação do Solver

A formulação IMPES utiliza apenas o solver linear do PETSc, que resolve o sistema usando diferentes métodos de subespaços de Krylov e vários precondicionadores disponíveis no software. A formulação totalmente implícita utiliza o solver não-linear, onde é utilizado variações do método de Newton - Newton Inexato (KELLEY, 1996) - para a solução do sistema.

A criação do solver linear é realizada com a subrotina SLESCreate(). Para acessar as subrotinas referentes aos métodos de subespaços de Krylov e precondicionadores são utilizadas as subrotinas SLESGetKSP() e SLESGetPC(). Desse modo é possível, por exemplo, alterar as tolerâncias padrões do solver linear usando KSPSetTolerances(), onde são fornecidas as tolerâncias residuais baseadas na norma l_2 do resíduo, além do máximo número de iterações permitido.

Os critérios de convergência dos métodos iterativos do solver linear utilizam as seguintes tolerâncias: redução relativa na norma do resíduo rtol, valor absoluto da norma do resíduo atol e aumento relativo na norma do resíduo dtol. O processo iterativo converge se for atendida a condição abaixo,

$$\|r^k\|_2 < \max\left[\text{rtol} \cdot \|r^0\|_2 ; \text{atol}\right] \qquad \text{onde} \qquad r^k = b - Ax^k; \tag{6.1}$$

sendo interrompido se

$$\|r^k\|_2 > \text{dtol} \cdot \|r^0\|_2 : \tag{6.2}$$

O solver não-linear é criado com a função SNESCreate(), que usa, por sua vez, todos os recursos do solver linear já descritos. Se forem utilizados os métodos de subespaços de Krylov nos ciclos mais internos - sistemas lineares - do método de Newton, tem-se o método de Newton Inexato (BALAY et al., 2001; KELLEY, 1996), que exige menos memória computacional que o método de Newton tradicional. As tolerâncias do solver não-linear são controladas através da subrotina SNESSetTolerances(), fornecendo-se os seguintes parâmetros: norma do incremento na solução entre duas iterações consecutivas stol, norma da função resíduo atol; variação da norma da função resíduo em relação a um valor inicial rtol e número máximo admissível para as iterações não-lineares e para cálculos da função resíduo.

Na formulação totalmente implícita, são fornecidos ainda os nomes das rotinas para montagem do vetor de resíduos e da matriz Jacobiana , usando SNESSetFunction() e SNESSetJacobian(), visto que o solver não-linear encarrega-se de chamar estas rotinas durante o processo iterativo.

6.2.8 Leitura e Transferência dos Dados de Entrada

Os dados de entrada são lidos no arquivo de entrada somente pelo processo master. Alguns dados são armazenados ou em variáveis escalares ou em pequenos arrays de dados, enquanto outros são armazenados nos vetores globais paralelos.

Os dados escalares ou pequenos arrays - dados tabelados - são lidos pelo processo master e, posteriormente, transferidos para os demais processos usando a subrotina de comunicação global `MPI_Bcast()`.

A leitura dos dados a serem armazenados em vetores globais paralelos pode ser realizada de duas formas, dependendo se os dados fornecidos são constantes ou variáveis para cada bloco da malha. Nos dois casos os valores são lidos pelo processo master e em seguida distribuídos para os demais processos.

O esquema da Figura 6.8 mostra como funciona o processo de leitura dos dados dos vetores paralelos, que são ¢x; ¢y; ¢z; Z; Á; k_x e k_y:

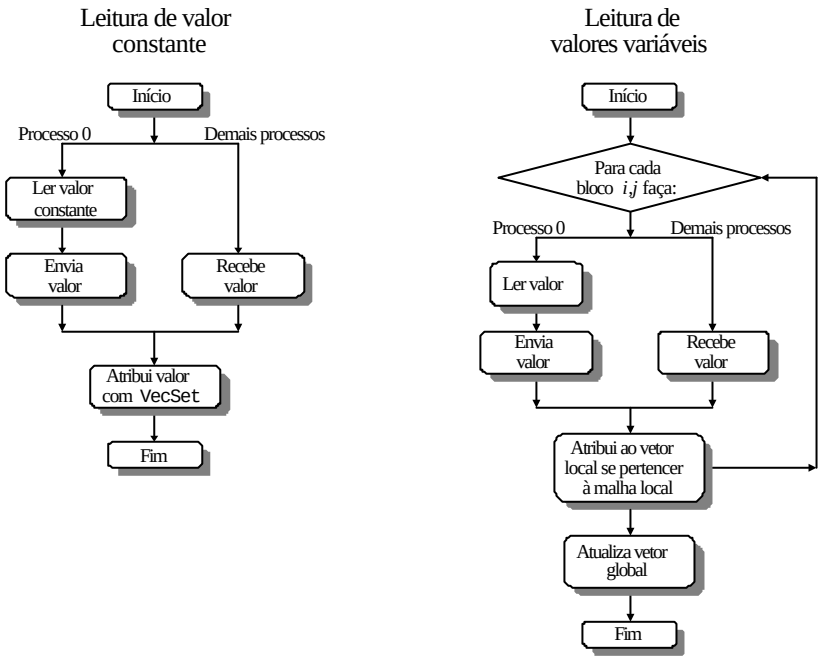

Figura 6.8: Processo de leitura de dados.

Quando o dado é constante para toda a malha, apenas um valor é lido pelo processo 0 e transferido imediatamente para os demais usando `MPI_Bcast()`. Em seguida cada processo atribui este valor ao vetor global utilizando a subrotina `VecSet()`, não havendo nenhuma comunicação nesta operação, visto que cada processo atribui o valor a sua porção local do vetor global.

Quando o dado é variável, sendo fornecidos diferentes valores para cada bloco da malha no arquivo de entrada, é necessário fazer com que cada processo atribua somente os valores referentes aos blocos pertencentes a sua malha local. Isto é feito utilizando os vetores locais, cujos endereços das memórias locais são acessados através de apontadores, obtidos com a subrotina `VecGetArray()`. Para cada bloco da malha, o processo master ler o dado e envia-o para os demais usando `MPI_Bcast()`. O processo, cujo bloco referente ao dado lido pertença a malha local, atribui este valor a seu vetor local. Quando todos os dados forem lidos e atribuídos aos vetores locais por cada processo, os apontadores são guardados usando a subrotina `VecRestoreArray()`, sendo necessário, em seguida, copiar os dados dos vetores locais para os vetores globais paralelos usando `DALocalToGlobal()`.

6.2.9 Inicialização das Pressões e Saturações

Nesta etapa são atribuídos valores iniciais aos vetores de pressões e saturações, representados por p_o e S_w na formulação IMPES, e X na formulação totalmente implícita.

A Figura 6.9 ilustra o funcionamento da subrotina de inicialização das variáveis.

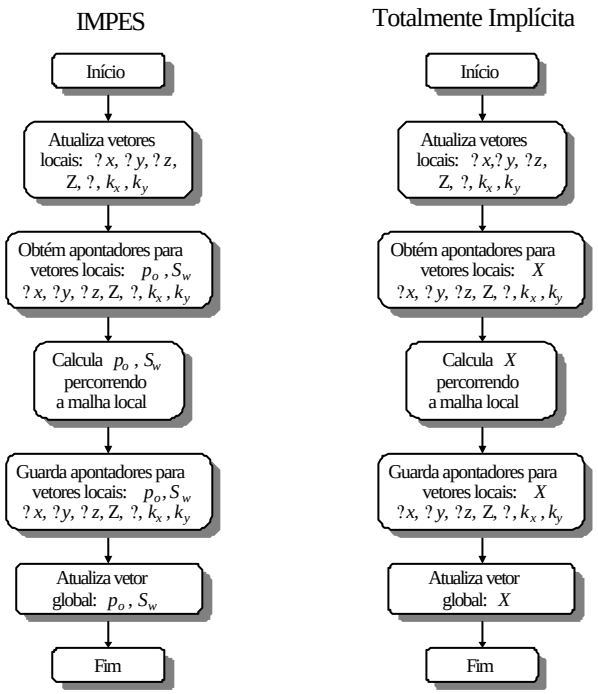

Figura 6.9: Fluxograma da rotina de inicialização das variáveis primárias.

Inicialmente, os elementos dos vetores globais já montados - vetores com dados geométricos e propriedades da rocha - são copiados para os vetores locais, que recebem também uma cópia dos valores ocultos. Para enviar cópias dos valores ocultos de um processo a outro é necessário comunicação, e esta operação é realizada usando as subrotinas

DAGlobalToLocalBegin()

DAGlobalToLocalEnd().

Esta operação envolve apenas comunicação entre processos sendo permitido adicionar linhas de código entre elas para sobrepor comunicação e computação, aumentando, eventualmente, a e...ciência do código.

Para acessar os elementos dos vetores locais, é necessário usar a subrotina VecGetArray(), com os quais são realizados os cálculos percorrendo cada bloco da malha local através dos índices globais $i; j$, sendo obtidos os valores iniciais de $p_{o_{i;j}}$ e $S_{w_{i;j}}$ na formulação IMPES, ou $X_{2i\ i\ 1;j}$ e $X_{2i;j}$ na formulação totalmente implícita. Os vetores são tratados como arrays bidimensionais, usando a convenção padrão da linguagem Fortran.

Tendo concluído as operações sobre os elementos dos vetores locais, é necessário utilizar a subrotina VecRestoreArray() para guardar os apontadores destes vetores. Finalmente, os dados dos vetores locais das variáveis primárias presisam ser transferidos para os vetores globais usando a subrotina DALocalToGlobal().

6.2.10 Montagem do Vetor do Sistema

Na formulação IMPES, esta etapa envolve a montagem do vetor de termos independentes do sistema, b; sendo realizada uma ou mais vezes a cada passo de tempo. Na formulação totalmente implícita, esta etapa é realizada a cada iteração de Newton, envolvendo a montagem do vetor de resíduos R:

O processo de montagem dos vetores segue a mesma idéia empregada na etapa de inicialização das variáveis, onde cada processo percorre sua malha local realizando operações e calculando os elementos dos vetores locais, $b_{i;j}$, ou $R_{2i\ i\ 1;j}$ e $R_{2i;j}$. A diferença básica em relação ao procedimento da etapa anterior está na fase inicial de transferência de dados dos vetores globais para os vetores locais, onde somente os vetores de pressões de óleo e saturações de água é que precisam ser atualizados. A Figura 6.10 mostra o funcionamento das rotinas de montagem destes vetores.

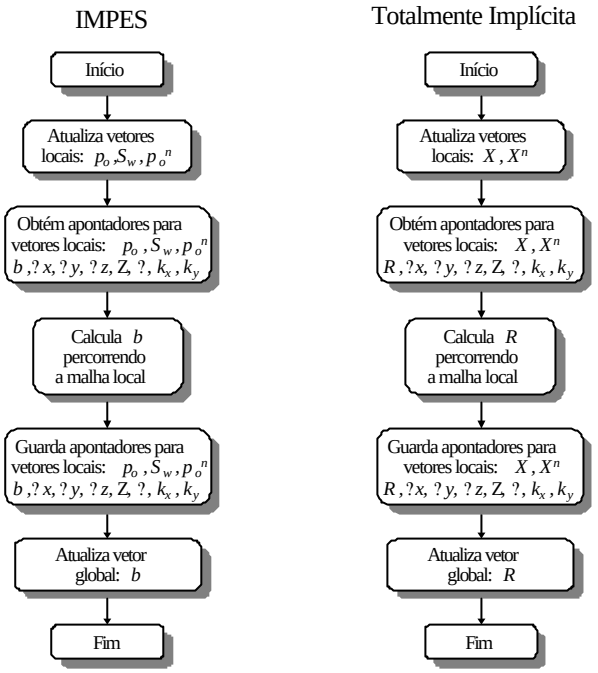

Figura 6.10: Fluxograma das rotinas para montagem do vetor b e vetor R:

6.2.11 Montagem da Matriz do Sistema

Foram implementadas as rotinas para montagem da matriz do sistema linear, A, e matriz Jacobiana, J, usando basicamente o mesmo procedimento já descrito para a montagem dos vetores do sistema, com cada processo percorrendo os blocos de sua malha local e efetuando as operações sobre os elementos dos vetores locais, calculando os elementos da matriz referentes a cada bloco. A Figura 6.11 ilustra o processo de montagem da matriz.

Cada processo calcula somente os elementos referentes aos blocos pertencentes a sua malha local, de acordo com o particionamento da malha usando o DA, atribuindo os valores calculados através da função `MatSetValues()`. No processo de montagem, cada elemento da matriz é distribuído para o processo que irá armazenar este elemento, e esta operação requer comunicação. Para isto, são utilizadas as seguintes subrotinas: `MatAssemblyBegin()` e `MatAssemblyEnd()`.

É possível adicionar linhas de código para sobrepor comunicação e computação, como foi feito no simulador.

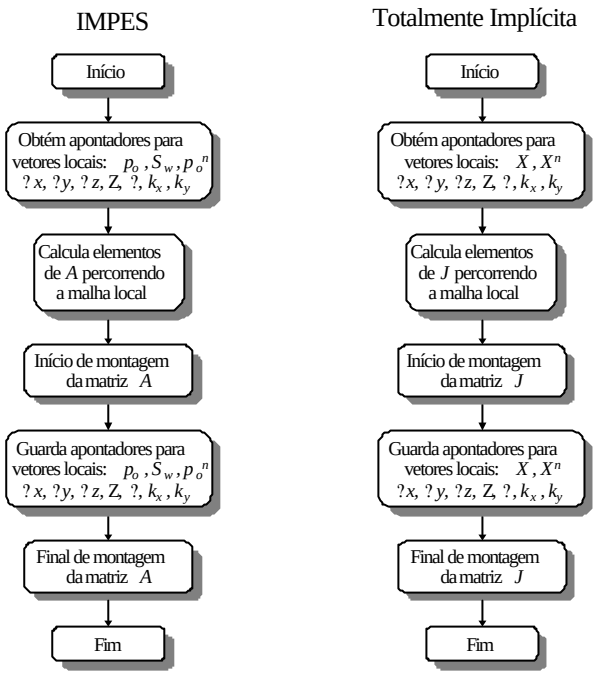

Figura 6.11: Fluxograma das rotinas para montagem das matrizes.

6.2.12 Solução do Sistema de Equações

ões

O solver linear do PETSc é acionado, na formulação IMPES, através da subrotina SLESSolve(), fornecendo como resultado o número de iterações lineares e o vetor de pressões de óleo p_o. Na formulação totalmente implícita, o sistema não-linear é resolvido iterativamente usando a subrotina SNESSolve(), cuja solução é o vetor multicomponente, X; contendo pressões de óleo e saturações de água.

Os sistemas lineares em ambos os solvers são resolvidos com um dos métodos de subespaços de Krylov e precondicionadores do PETSc (BALAY et al., 2001).

6.2.13 Cálculo Explícito das Saturações de Água

Na formulação IMPES, após a solução do sistema linear para pressões de óleo, calcula-se as saturações de água explicitamente, segundo a eq. (3.94). O ‡uxograma da rotina para o cálculo destas saturações é mostrado na Figura 6.12.

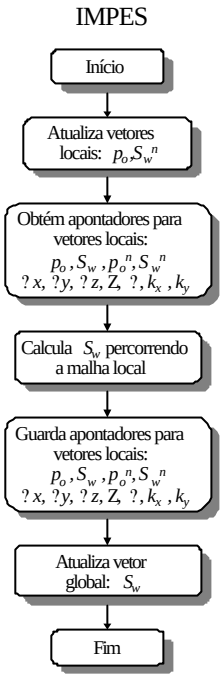

Figura 6.12: Fluxograma da rotina para o cálculo explícito das saturações de água usando a formulação IMPES.

6.2.14 Visualização dos Resultados

Tendo obtido os resultados do problema a cada passo de tempo, pode-se visualizar os dados dos vetores de pressões de óleo e saturações de água através dos visualizadores, usando-se a subrotina `VecView()`. Esta subrotina faz com que todos os processos enviem seus dados locais, dos vetores globais, para o processo master, que se encarrega de plotar os resultados na tela.

A visualização dos resultados requer comunicação e é ativada, opcionalmente, através da opção de linha de comando `-visual`.

6.2.15 Impressão dos Resultados

A impressão dos resultados segue a mesma idéia da etapa de visualização, onde todos os processos enviam seus resultados para o processo master, que imprime-os nos arquivos de saída. A implementação da rotina de impressão dos resultados foi realizada usando subrotinas específicas do MPI, descritas na seção 5.7.1. A impressão dos resultados também é opcional, sendo ativada com a opção `-result`. As Figuras 6.13 e 6.14 ilustram o esquema de impressão dos resultados.

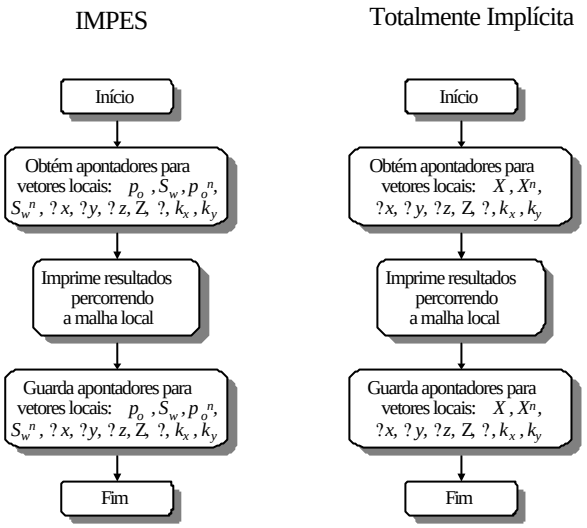

Figura 6.13: Fluxograma da rotina de impressão de resultados.

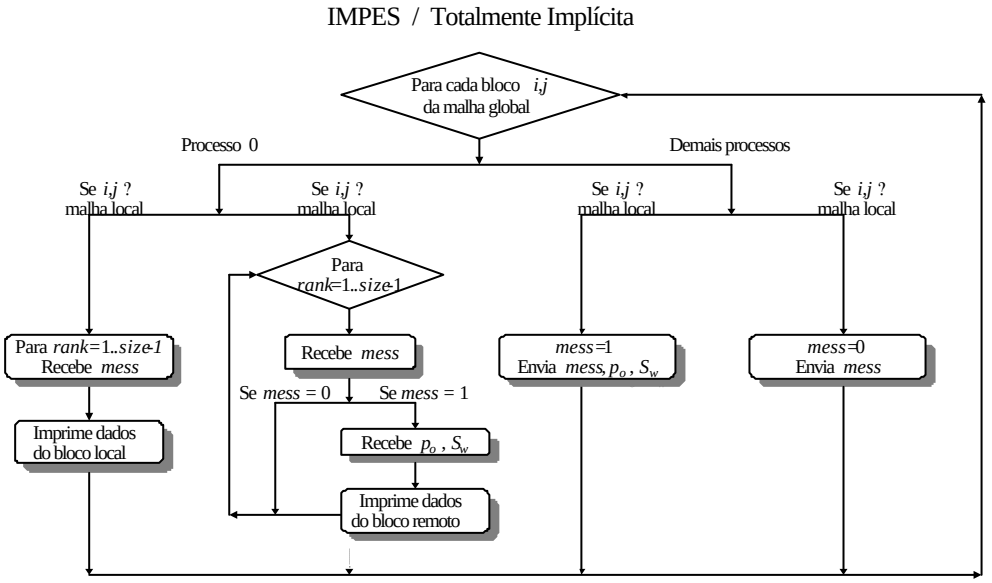

Figura 6.14: Processo de impressão dos resultados para cada bloco da malha global.

Para o cálculo das médias volumétricas das propriedades, e valores máximos e mínimos das variáveis primárias, foram utilizadas as subrotinas VecPointwiseMult(), VecSum(), VecMax() e VecMin().

6.2.16 Controle do Passo de Tempo

A rotina para controle do passo de tempo foi implementada de acordo com o algorítmo descrito na seção 3.5.1. São utilizadas as subrotinas VecWAXPY(), VecAbs() e VecMax() para o cálculo dos máximos incrementos de pressão e saturação entre passos de tempo consecutivos. Como cada processo fica com uma cópia destes máximos incrementos, todos calculam o mesmo valor para o novo intervalo de tempo.

O controle do passo de tempo é opcional, sendo ativado com a opção -tempo.

6.2.17 Destruição dos Objetos e Finalização do Programa

Após o último passo de tempo, a simulação é concluída e todos os objetos do PETSc criados são destruídos para liberar espaço na memória. Para isto são usadas as subrotinas: PetscViewerDestroy(), DADestroy(), VecDestroy(), MatDestroy(), SLESDestroy(), SNESDestroy().

Em seguida, as biliotecas, MPI e PETSc, são finalizadas usando a subrotina PetscFinalize().

Capítulo 7

Aplicações e Resultados Numéricos

Os capítulos anteriores mostraram o equacionamento do problema, as formulações numéricas para sua solução, o ambiente computacional e os detalhes da implementação do simulador. Neste capítulo são mostradas aplicações do simulador na solução de alguns problemas de ‡uxo em reservatórios.

As etapas necessárias para validar a implementação do simulador são: comparação dos resultados do simulador desenvolvido com os de outro simulador existente em um computador sequencial; análise de desempenho do simulador usando um computador paralelo.

Para comparação dos resultados foi utilizado o simulador comercial BOAST-98 (FANCHI, 1982), que é um simulador Black-Oil convencional e admite malhas de discretização tridimensionais, empregando a formulação IMPES para resolver o sistema de equações. São mostrados os resultados obtidos em um único microcomputador para três problemas de injeção de água em reservatórios, com diferentes graus de complexidade.

Em seguida, é mostrada a análise de desempenho do simulador em um cluster de PCs com 4 nós, sendo medidos os tempos de execução, speedup e e…ciências para problemas de ‡uxo em modelos 1/4 de 5-spot, admitindo diferentes malhas e utilizando diferentes opções para os solvers linear e não-linear.

7.1 Comparação dos Resultados do Simulador

Os três problemas utilizados para esta comparação são descritos a seguir, onde os dois primeiros casos representam reservatórios mais simples e homogêneos, e o terceiro é um reservatório heterogêneo, cujos dados são obtidos de um reservatório real descrito

por ERTEKIN et al. (2001).

7.1.1 Caso 1: Reservatório 1D

O primeiro caso envolve a simulação do ‡uxo unidimensional de óleo e água em um reservatório horizontal e linear, com um poço de injeção de água e um poço de produção em cada extremo, como ilustra a Figura 7.1. As propriedades da rocha e dimensões dos blocos são constantes em toda malha. O poço 1 injeta água com uma vazão constante em condições padrão, e o poço 2 produz óleo e água a uma pressão de poço constante. As forças de gravidade e de capilaridade são nulas. Alguns dados utilizados no problema estão indicados nas Tabelas 7.1 a 7.4, não sendo usadas as pressões de capilaridade da Tabela 7.3 neste caso.

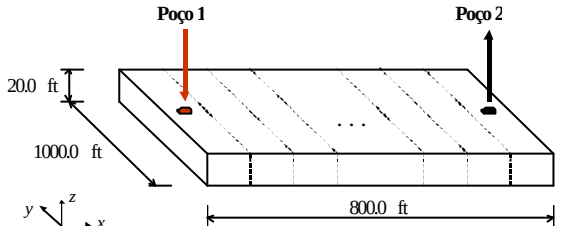

Figura 7.1: Caso 1 - Modelo do reservatório.

Propriedades da rocha		
Permeabilidades Absolutas (mD):	$k_x = 200$	$k_y = 200$
Porosidade:	$Á = 0{:}25$	
Compressibilidade da rocha (psi^{-1}):	$c_r = 3{:}0 \pounds 10^{-6}$	
Condições iniciais		
Pressão de óleo inicial (psi):	$p_o^0 = 3000$	
Saturação de água inicial:	$S_w^0 = 0{:}2$	
Tempo de simulação total (dias):	$t_{max} = 720$	

Tabela 7.1: Dados do reservatório - Caso 1

p (psi)	$½_o$ (lbm/ft^3)	$½_w$ (lbm/ft^3)	B_o (rb/stb)	B_w (rb/stb)	$¹_o$ (cp)	$¹_w$ (cp)
-	46.244	62.238	1.50	1.00	2.00	1.00

Tabela 7.2: Propriedades dos ‡uidos - Caso 1

S_w	k_{ro}	k_{rw}	P_{cow} (psi)
0.0	1.00000	0.0000	698.0
0.1	1.00000	0.0000	698.0
0.2	0.60400	0.0000	698.0
0.3	0.20700	0.0122	363.0
0.4	0.13400	0.0244	237.0
0.5	0.05710	0.0336	161.0
0.6	0.03700	0.0672	98.0
0.7	0.00228	0.1344	42.0
0.8	0.00147	0.2688	0.0
0.9	0.00000	0.4704	0.0
1.0	0.00000	0.5000	0.0

Tabela 7.3: Permeabilidade relativa e pressão de capilaridade - Casos 1 e 2

Poço	Tipo	Valor prescrito	Fator Geométrico - G_w
1	Injeção	q_{wsc} = 1000:0 stb/dia	1:0
2	Produção	p_{wf} = 2500:0 psi	1:0

Tabela 7.4: Dados de poços - Caso 1

Foram realizadas várias análises variando-se os intervalos de tempo, admitidos constantes durante toda a simulação, e também o número de blocos da malha. Os intervalos de tempo e os números de blocos utilizados são: $¢t$ = 1; 10 e 20 dias e n_x = 10; 20 e 40: As simulações foram realizadas usando as formulações IMPES e totalmente implícita.

As Figuras 7.2, 7.3 e 7.4 mostram os grá...cos de vazão de óleo no poço produtor, pressão no poço injetor e pressão média no reservatório, respectivamente, ao longo do tempo de simulação, referentes a $¢t$ = 1 dias e n_x = 20. Não veri...ca-se diferenças signi...cativas entre os resultados obtidos com as formulações IMPES e totalmente implícita (praticamente iguais), e com o simulador BOAST-98.

À medida que injeta-se água no reservatório, a pressão no reservatório tende a crescer aumentando também a vazão de produção de óleo. Quando chega um instante onde a vazão de óleo atinge um valor máximo, t = 30 dias; a pressão no reservatório tende a estabilizar. A água injetada atinge o poço produtor no tempo t = 300 dias, o que reduz a produção de óleo e gera um aumento na pressão do reservatório.

Os mesmos resultados foram obtidos considerando agora malhas com n_x = 10 e

n_x = 40 blocos, mostrados nas Figuras 7.5 e 7.6, onde se observa o efeito da dispersão numérica (MATTAX e DALTON, 1990), bastante comum em problemas de simulação de reservatórios. Este efeito in‡uencia bastante na seleção do número de blocos a ser utilizado nas simulações, e está relacionado ao fato de que a cada passo de tempo a frente de saturação de água avança de um bloco para outro, independente do tamanho dos blocos ou do intervalo de tempo usado. MATTAX e DALTON (1990) discutem alternativas para reduzir o efeito da dispersão numérica.

A Figura 7.7 ilustra o grá...co com as frentes de saturação de água nos tempos 90, 180 e 360 dias, para os diferentes níveis de discretização analisados, observando-se maiores efeitos nas curvas de saturação mais inclinadas - com menor número de bloco. Aumentando-se o número de blocos a frente de saturação torna-se mais vertical. Nos grá..cos de vazões, Figuras 7.5 e 7.6, este efeito é explicado pela redução precoce da vazão de óleo no poço produtor quando utilizado um menor número de blocos, sendo observado nas formulações IMPES e totalmente implícita implementadas, e no BOAST-98.

O simulador BOAST-98, que utiliza a formulação IMPES para a solução do problema de ‡uxo, apresentou sinais de problemas de instabilidade numérica com n_x = 40 blocos, comum em simuladores que usam a formulação IMPES, como visto na seção 3.4.2 (MATTAX e DALTON, 1990; e ERTEKIN et al., 2001). Este mesmo problema foi veri...cado também no simulador desenvolvido, quando empregada a formulação IMPES com um número de blocos pouco maior que 40: Já a formulação totalmente implícita não apresentou estes problemas, visto que é incondicionalmente estável.

Mantendo-se o número de blocos ...xo, n_x = 10; e fazendo-se variar o intervalo de tempo de simulação, ¢t = 1; 10 e 20 dias, tem-se os grá...cos mostrados nas Figuras 7.8 a 7.10. Usando a formulação totalmente implícita os resultados das análises para os diferentes intervalos de tempo não apresentaram variações signi...cativas, enquanto as formulações IMPES do simulador desenvolvido e do BOAST-98 apresentaram maiores variações. De fato, embora a formulação totalmente implícita exija maior esforço computacional, para resolver o problema, do que a formulação IMPES, o fato de usar maiores intervalos de tempo pode viabilizar sua aplicação, principalmente em modelos com elevadas não-linearidades, modelos que usam malhas irregulares ou com malhas bastante re...nadas, onde o intervalo de tempo necessário na formulação IMPES torna-se muito limitado.

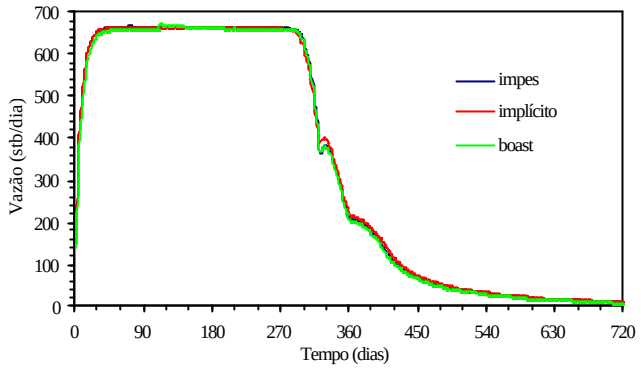

Figura 7.2: Caso 1 - vazão de óleo no poço produtor, q_{osc}, para $\Delta t = 1$ dia e $n_x = 20$.

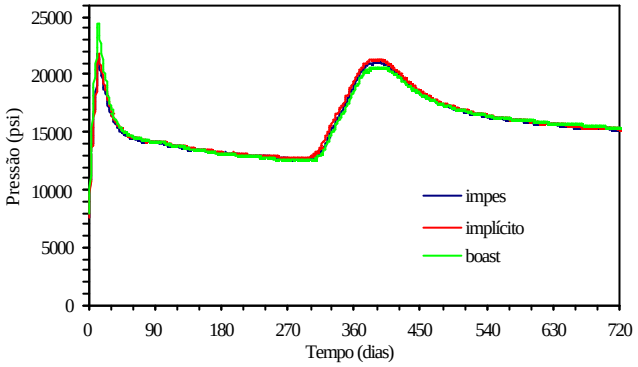

Figura 7.3: Caso 1 - pressão no poço injetor, p_{wf}, para $\Delta t = 1$ dia e $n_x = 20$.

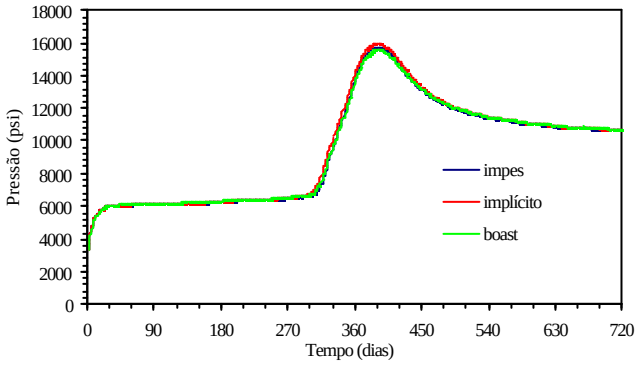

Figura 7.4: Caso 1 - pressão média no reservatório, para $\Delta t = 1$ dia e $n_x = 20$.

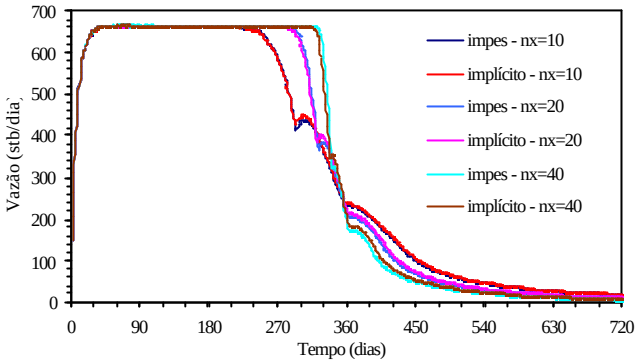

Figura 7.5: Caso 1 - vazão de óleo no poço produtor, q_{osc}, para $\Delta t = 1$ dia e $n_x = 10; 20; 40$ - IMPES e totalmente implícito.

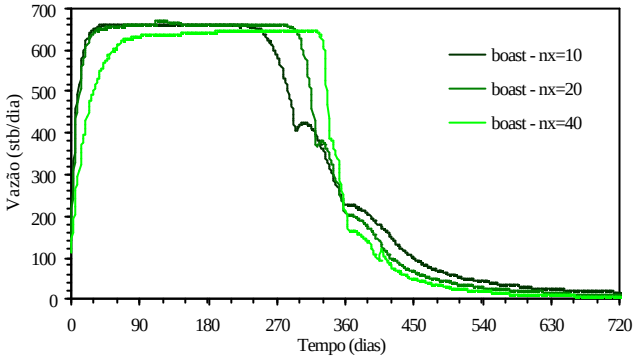

Figura 7.6: Caso 1 - vazão de óleo no poço produtor, q_{osc}, para $\Delta t = 1$ dia e $n_x = 10; 20; 40$ - BOAST-98.

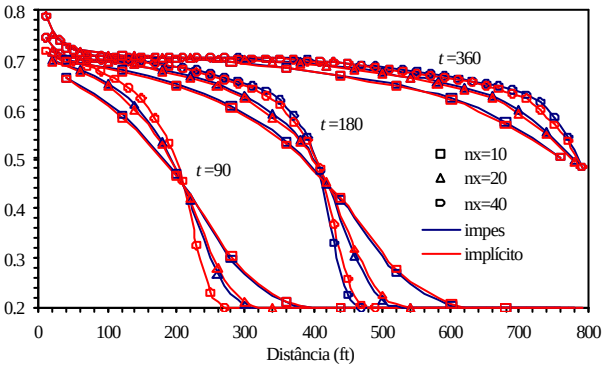

Figura 7.7: Caso 1 - curvas da frente de saturação de água nos tempos de simulação 90, 180 e 360 dias.

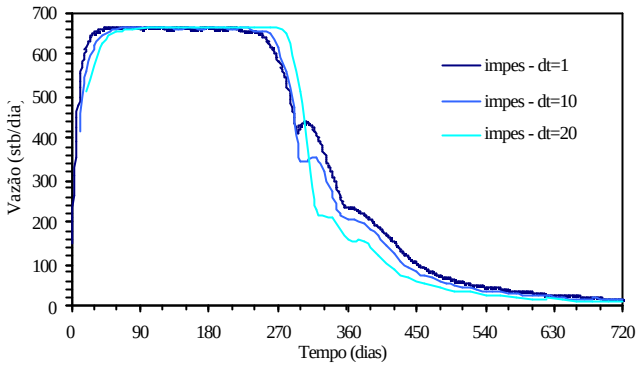

Figura 7.8: Caso 1 - vazão de óleo no poço produtor, q_{osc}, para $n_x = 10$ e $¢t = 1; 10$ e 20 dias - IMPES.

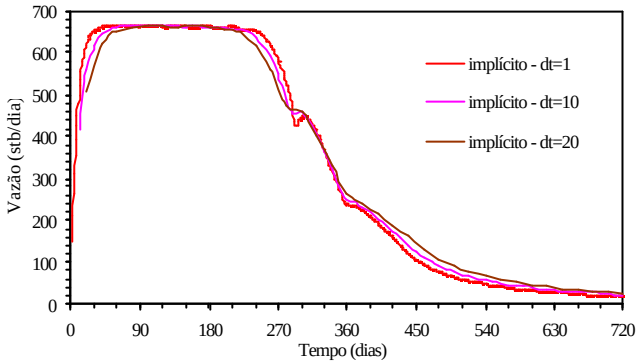

Figura 7.9: Caso 1 - vazão de óleo no poço produtor, q_{osc}, para $n_x = 10$ e $¢t = 1; 10$ e 20 dias - totalmente implícito.

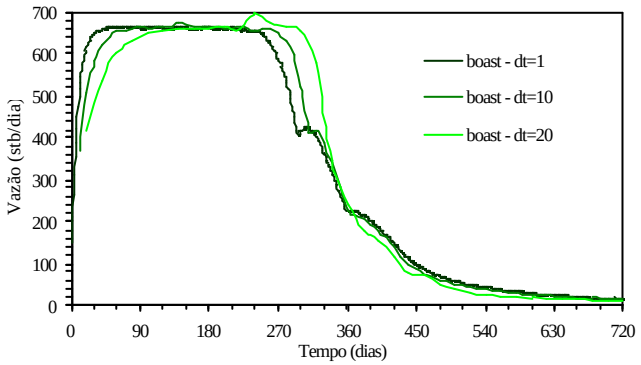

Figura 7.10: Caso 1 - vazão de óleo no poço produtor, q_{osc}, para $n_x = 10$ e $¢t = 1; 10$ e 20 dias - BOAST-98.

7.1.2 Caso 2: Reservatório 2D com modelo 5-spot

Neste caso, o reservatório analisado é homogêneo e horizontal, sendo discretizado com uma malha retangular bidimensional para representar o ‡uxo segundo as direções x e y: É modelado o sistema de produção 5-spot, como ilustra a Figura 7.11, onde são considerados 4 poços de injeção de água, um em cada extremo da malha, e um poço de produção no centro da malha. É utilizada uma malha 9x9, com blocos de dimensões constantes, e um intervalo de tempo constante, ¢t = 5 dias. Os poços de injeção operam com uma vazão constante e o poço de produção com uma pressão de poço constante durante toda simulação.

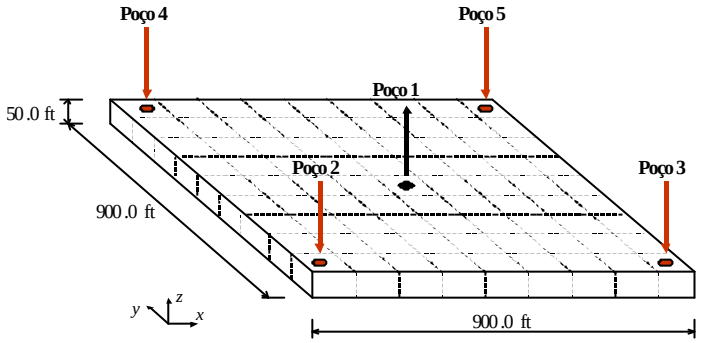

Figura 7.11: Caso 2 - Modelo do reservatório.

São utilizados alguns dados do Caso 1, indicados nas Tabelas 7.1 a 7.3, modi…cando apenas a pressão de óleo inicial no reservatório, que neste caso é 2000:0 psi. As forças de gravidade são nulas e as forças capilares são consideradas através das pressões de capilaridade fornecidas na Tabela 7.3. Os dados especí…cos de poços usados neste caso estão indicados na Tabela 7.5.

Poço	Tipo	Valor prescrito	Fator Geométrico - G_w
1	Produção	p_{wf} = 1500:0 psi	1:0
2,3,4,5	Injeção	q_{wsc} = 200:0 stb/dia	1:0

Tabela 7.5: Dados de poços - Caso 2

Os grá…cos das Figuras 7.12 a 7.15 mostram que os resultados obtidos com os simuladores são muito próximos aos resultados fornecidos pelo simulador BOAST-98, validando a aplicação das formulações IMPES e totalmente implícita implementadas para a solução de problemas com estas características.

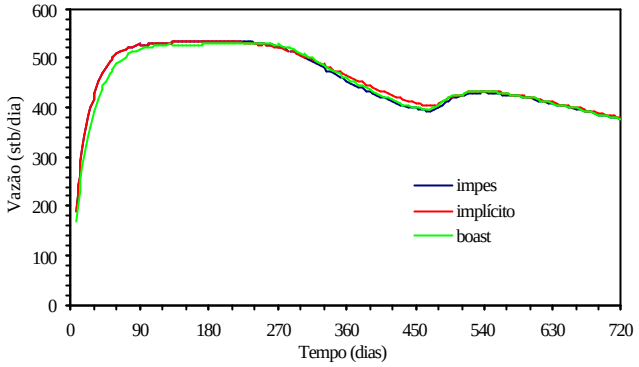

Figura 7.12: Caso 2 - vazão de óleo no poço produtor, q_{osc}.

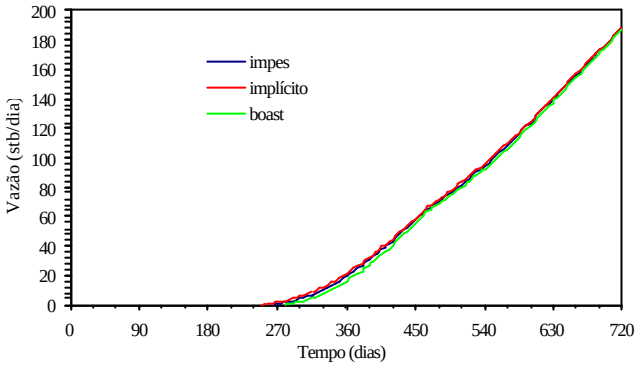

Figura 7.13: Caso 2 - vazão de água no poço produtor, q_{wsc}.

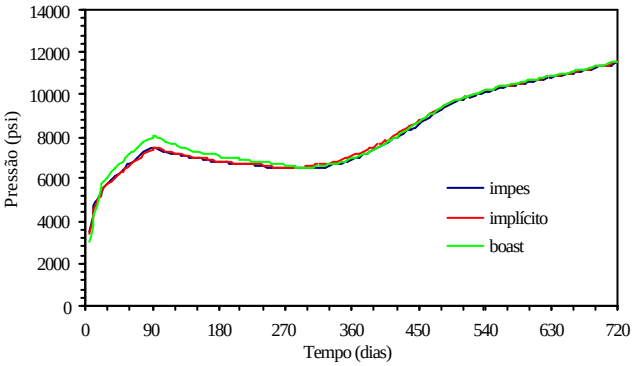

Figura 7.14: Caso 2 - pressão no poço injetor, p_{wf}:

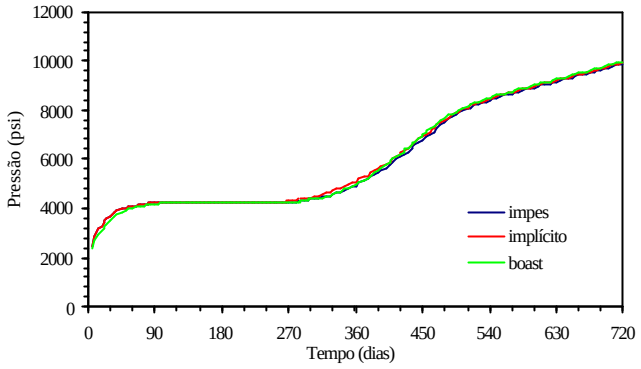

Figura 7.15: Caso 2 - pressão média no reservatório.

7.1.3 Caso 3: Reservatório Heterogêneo

O terceiro caso analisado consiste na simulação do ‡uxo bifásico óleo/água em um reservatório heterogêneo com contorno irregular. ERTEKIN et al. (2001) fornecem com detalhes a caracterização de um reservatório real, apresentado como reservatório A-1. Este reservatório possui uma geometria irregular, e é discretizado com a malha retangular 12 £ 9 mostrada na Figura 7.16, onde o contorno irregular é aproximado - linhas escuras - admitindo a existência de blocos inativos, representados por blocos com porosidades nulas, tanto no simulador desenvolvido neste trabalho como no simulador BOAST-98. As dimensões, profundidades, permeabilidades absolutas e porosidades dos blocos estão representadas nas Figuras 7.17 a 7.20, respectivamente. Outros dados referentes a propriedades da rocha, propriedades dos ‡uidos, permeabilidades relativas e pressões de capilaridade são fonecidos nas Tabelas 7.6 a 7.8.

São considerados três poços no modelo, localizados de acordo com a Figura 7.16, sendo dois poços produtores, operando com pressão constante, e um poço injetor com vazão de água constante. Os dados dos poços encontram-se na Tabela 7.9.

As Figuras 7.21 a 7.25 ilustram os grá…cos de vazões e pressões nos poços obtidos com o simulador desenvolvido, com o BOAST-98 e por ERTEKAN et al. (2001). Estes grá…cos apresentam resultados satisfatórios, mostrando também a aplicabilidade do simulador na solução de modelos de reservatórios heterogêneos e com geometria irregular.

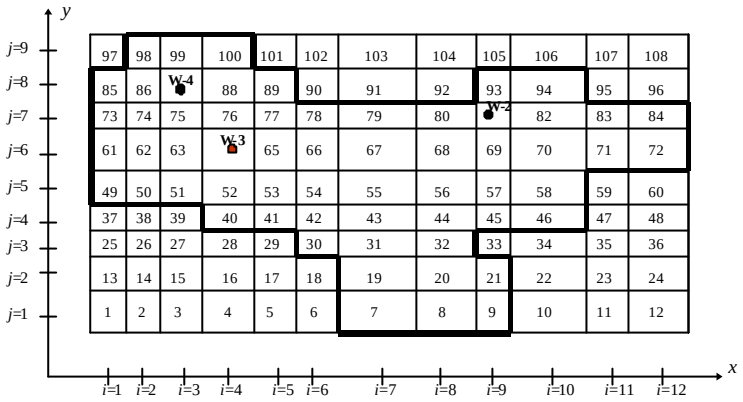

Figura 7.16: Caso 3 - Modelo do reservatório.

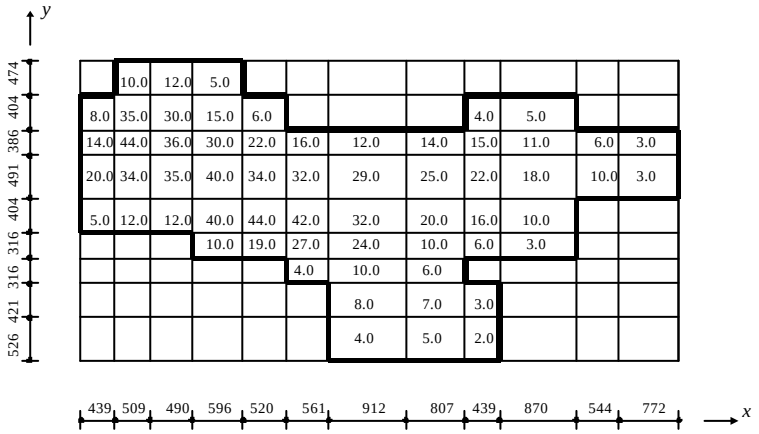

Figura 7.17: Dimensões, ¢x e ¢y, e espessura dos blocos, ¢z, em ft - Caso 3.

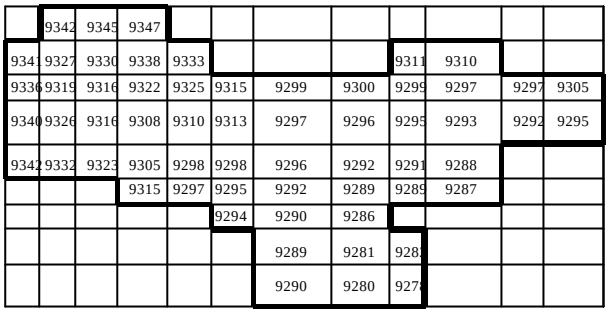

Figura 7.18: Profundidade das faces superiores dos blocos, ft - Caso 3.

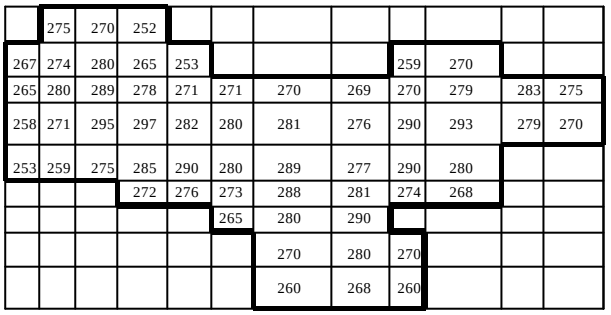

Figura 7.19: Permeabilidade absoluta na direção x, mD - Caso 3.

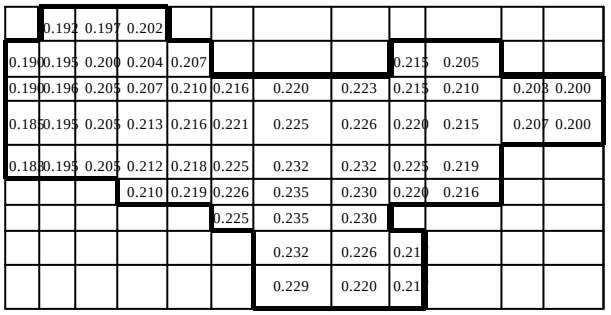

Figura 7.20: Porosidade da rocha - Caso 3.

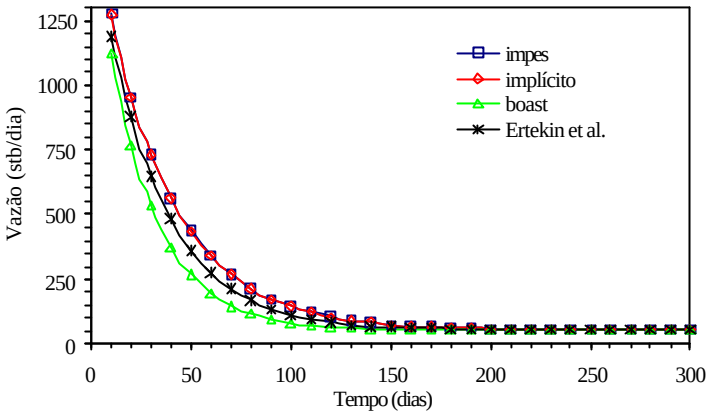

Figura 7.21: Caso 3 - vazão de óleo no poço produtor W-2, q_{osc}:

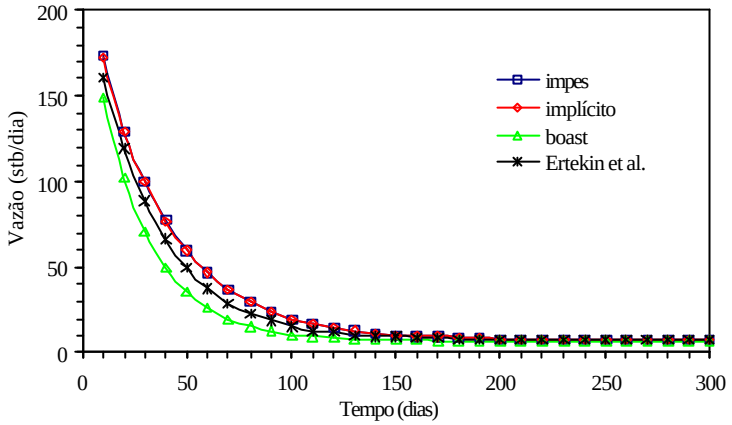

Figura 7.22: Caso 3 - vazão de água no poço produtor W-2, q_{wsc}:

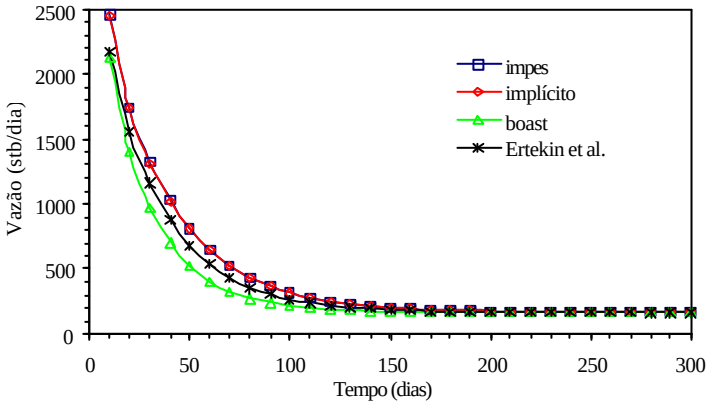

Figura 7.23: Caso 3 - vazão de óleo no poço produtor W-4, q_{osc}:

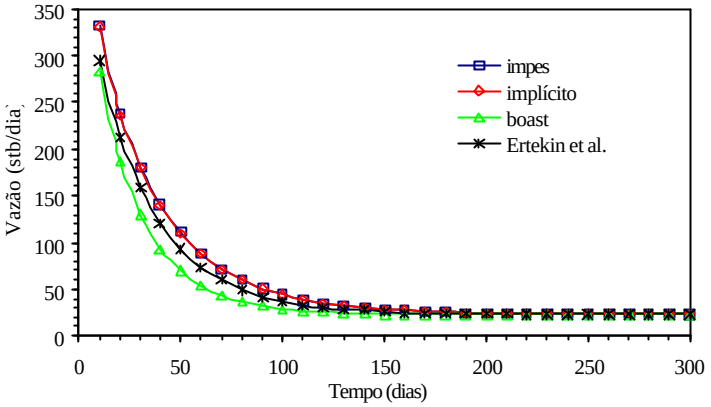

Figura 7.24: Caso 3 - vazão de água no poço produtor W-4, q_{wsc}:

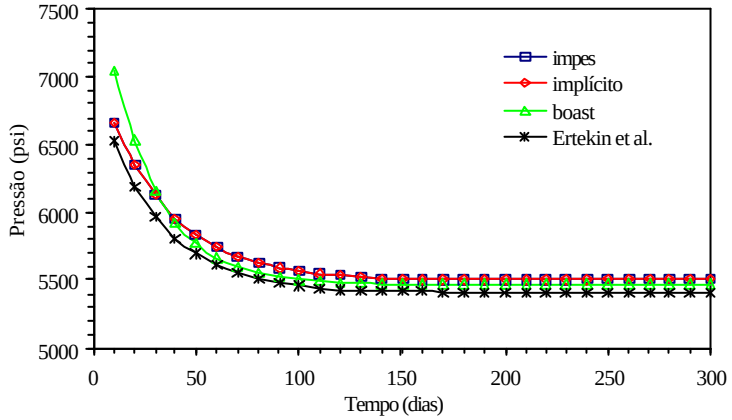

Figura 7.25: Caso 3 - pressão no poço injetor W-3, p_{wf}:

Propriedades da rocha	
Permeabilidade absoluta na direção y:	$0{:}8k_x$
Compressibilidade da rocha (psi^{-1}):	$c_r = 5{:}9 \pounds 10^{-6}$
Propriedades dos ‡uidos	
Compressibilidade do óleo (psi^{-1}):	$c_o = 5{:}0 \pounds 10^{-6}$
Compressibilidade da água (psi^{-1}):	$c_w = 1{:}0 \pounds 10^{-6}$
Valores de referência: $B_o^{ref} = 1{:}0$ rb/stb $B_w^{ref} = 1{:}0$ rb/stb	$p^{ref} = 14{:}7$ psi
Condições Iniciais	
Pressão de óleo inicial em todo reservatório (psi):	$p_o^0 = 7000{:}0$
Saturação de água inicial em todo reservatório (psi):	$S_w^0 = 0{:}50$
Tempo de simulação total (dias):	$t_{max} = 300$
Intervalo de tempo (dias):	$¢t = 10$

Tabela 7.6: Dados do reservatório - Caso 3

p (psi)	$½_o$ (lbm/ft^3)	$½_w$ (lbm/ft^3)	$¹_o$ (cp)	$¹_w$ (cp)
5000.0	45.1925	63.5225	0.9200	0.52
5500.0	45.4413	63.7077	0.9243	0.52
6000.0	45.7426	63.8928	0.9372	0.52
6500.0	45.7426	63.8928	0.9494	0.52
7000.0	45.7426	63.8928	0.9650	0.52
7500.0	45.7426	63.8928	0.9812	0.52
8000.0	45.7426	63.8928	1.0019	0.52

Tabela 7.7: Propriedades dos ‡uidos - Caso 3

S_w	k_{ro}	k_{rw}
0.18	1.00000	0.00000
0.21	0.92692	0.00000
0.24	0.85441	0.00002
0.27	0.79288	0.00014
0.30	0.71312	0.00045
0.33	0.64526	0.00111
0.36	0.57980	0.00232
0.39	0.51709	0.00430
0.42	0.45744	0.00733
0.45	0.40110	0.01175
0.48	0.34831	0.01791
0.51	0.29924	0.02623
0.54	0.25403	0.03714
0.57	0.21278	0.05160
0.60	0.17552	0.06882
0.63	0.14228	0.09069
0.66	0.11301	0.11741
0.69	0.08763	0.14963
0.72	0.06603	0.18807
0.75	0.04803	0.23347
0.78	0.03344	0.28664
0.81	0.02199	0.34842
0.84	0.01340	0.41968
0.87	0.00733	0.50135
0.90	0.00340	0.59439

S_w	P_{cow} (psi)
0.20	8.00
0.25	4.30
0.30	3.00
0.40	1.78
0.50	1.21
0.60	0.79
0.70	0.43
0.80	0.10
0.90	0.0

Tabela 7.8: Permeabilidades relativas e pressão de capilaridade - Caso 3

Poço	Tipo	Bloco	Valor prescrito	Raio do poço	Fator de skin
W-2	Produção	(9; 7)	p_{wf} = 5300:0 psi	0.25 ft	0.0
W-3	Injeção	(4; 6)	q_{wsc} = 250:0 stb/dia	0.25 ft	0.0
W-4	Produção	(3; 8)	p_{wf} = 5300:0 psi	0.25 ft	0.0

Tabela 7.9: Dados de poços - Caso 3

7.2 Análise de Desempenho do Simulador

Para analisar a aplicabilidade e o desempenho do simulador em um computador paralelo com memória distribuída, foi utilizado um cluster de PCs, montado no Departamento de Eletrônica e Sistemas, do CTG - UFPE. Cluster de PCs (STERLING et al., 1995) é um computador paralelo formado por um conjunto de microcomputadores interligados através de uma rede de conexão, capaz de executar programas em paralelo utilizando o modelo de programação com memória distribuída. Esta máquina vem sendo utilizada com sucesso na solução de problemas de larga escala, em simulações de reservatórios, (ABATE et al., 1999; WANG et al., 1999), surgindo como uma alternativa de baixo custo para a aplicação do processamento paralelo, em lugar dos supercomputadores paralelos proprietários.

7.2.1 Descrição do Cluster de PCs

O cluster utilizado nestas análises é do tipo NOW - network of workstations, homogêneo, sendo composto por 4 nós, cada um com processador Pentium II de 350 MHz, memória RAM de 128 Mb, disco rígido com 4 Gb de memória e placa de rede 3Com Fast EtherLink XL - PCI 10/100BASE-TX. Cada nó está conectado a um hub Intel Express 140T Standalone, operando a uma velocidade de 100 Mbps. Os 4 nós funcionam com o sistema operacional Linux Red Hat 6.2, onde foram instalados os softwares MPICH-1.2.3 e PETSc-2.1.0.

7.2.2 Modelo Analisado

Para veri...car o desempenho do simulador no cluster, foi analisado um problema de injeção de água em um reservatório homogêneo, usando o modelo 1/4 de 5-spot, sendo analisado apenas 2 dias de simulação com um intervalo de tempo constante igual a 0.5 dia. Foram medidos os tempos de processamento usando 1, 2 e 4 processadores, admitindo diferentes malhas de discretização, correspondentes a 200x200, 300x300 e 400x400, para a formulação IMPES, e 150x150, 200x200 e 240x240, para a formulação totalmente implícita. As dimensões dos blocos são mantidas constantes em todas as análises, igual a 100x100x20 ft.

As malhas foram particionadas utilizando uma decomposição unidimensional segundo a direção y, como ilustra a Figura 7.26.

Foram utilizados diferentes métodos iterativos para a solução dos sistemas lineares, sendo medidos os tempos de execução com cada método nas várias análises. Os métodos

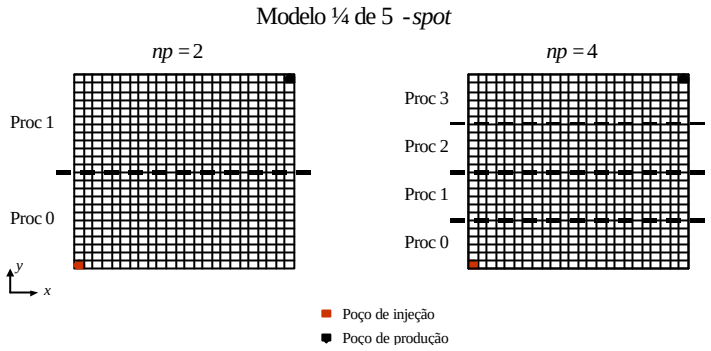

Figura 7.26: Esquema de partição das malhas do modelo 1/4 de 5-spot.

usados foram: Biconjugate Gradient (bicg), Biconjugate Gradient Stabilized (bcgs) e Generalized Minimum Residual (gmres) - com reinício a cada 10 iterações, gmres(10). Como precondicionador, foi empregado o precondicionador Block-Jacobi, considerando um bloco por processador, onde em cada bloco o sistema é resolvido com o método ILU(0).

As tolerâncias usadas para os solvers linear e não-linear estão indicadas na Tabela 7.10.

Formulação	Solver Linear	Solver Não-Linear
IMPES	atol = $1 \pounds 10^{-10}$ rtol = $1 \pounds 10^{-5}$ dtol = $1 \pounds 10^{3}$	
Totalmente implícito	atol = $1 \pounds 10^{-10}$ rtol = $1 \pounds 10^{-5}$ dtol = $1 \pounds 10^{3}$	stol = $1 \pounds 10^{-5}$ atol = $1 \pounds 10^{-7}$ rtol = $1 \pounds 10^{-7}$

Tabela 7.10: Tolerâncias utilizadas nos solvers

Nas tabelas com medidas de desempenho, é adotada a seguinte notação: t = tempo total de execução, em segundos; it = numero médio de iterações do solver linear, por passo de tempo; sp = speedup; efi = e...ciência.

As medidas de speedup e e...ciência apresentadas a seguir foram obtidas em relação ao tempo de execução do código paralelo usando um único processador.

7.2.3 Resultados com a Formulação IMPES

A Tabela 7.11 mostra os resultados obtidos na análise de desempenho do problema descrito, usando a formulação IMPES. Os grá...cos ilustrando o speedup alcançado estão representados nas Figuras 7.27 a 7.29, referentes a cada malha especi...cada. Observa-se speedup quase linear em todos os casos, sendo alcançadas e...ciências entre 92 e 97%, quando utilizadas apenas 2 cpus, e e...ciências entre 86 e 93% utilizando as 4 cpus. Estes resultados são satisfatórios, tendo em vista que a escolha de um precondicionador adequado pode elevar ainda mais estas e...ciências. O método bicg foi o que apresentou maiores speedups e e...ciências, porém foi o que levou mais tempo para resolver os problemas, enquanto o método bcgs foi o que apresentou os menores tempos de execução.

		Número de Processadores									
		1		2				4			
Método	Malha	t	it	t	it	sp	efi	t	it	sp	efi
bicg	200x200	80	25	43	27	1.86	0.93	22	28	3.64	0.91
bcgs		65	18	35	19	1.86	0.93	19	19	3.42	0.86
gmres(10)		68	26	37	28	1.84	0.92	19	29	3.58	0.89
bicg	300x300	186	28	96	27	1.94	0.97	50	28	3.72	0.93
bcgs		151	19	79	19	1.91	0.96	41	19	3.68	0.92
gmres(10)		155	26	83	28	1.87	0.93	44	28	3.52	0.88
bicg	400x400	322	25	169	27	1.91	0.95	90	27	3.58	0.89
bcgs		258	18	140	19	1.84	0.92	74	19	3.49	0.87
gmres(10)		274	26	148	28	1.85	0.93	78	28	3.51	0.88

Tabela 7.11: Medidas de desempenho usando a formulação IMPES.

7.2.4 Resultados com a Formulação Totalmente Implícita

A análise de desempenho realizada para a formulação totalmente implícita também apresentou bons resultados, como mostra a Tabela 7.12. O número de iterações de Newton por passo de tempo variou entre 2 e 3, e os speedups obtidos estão indicados nos grá...cos das Figuras 7.30 a 7.32 para as diferentes malhas analisadas. Veri...ca-se e...ciências entre 89 e 96%, com 2 cpus, e entre 84 e 92% com 4 cpus, sendo o método bcgs o que apresentou os menores tempos de execução e também os maiores speedups e e...ciências. Foi veri...cado ainda que o solver não-linear é responsável por 95 a 98% do tempo total de execução do simulador.

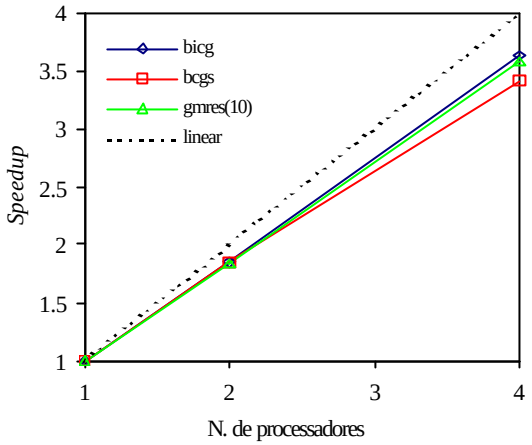

Figura 7.27: Speedup para a malha 200x200 (40.000 eqs.) - IMPES.

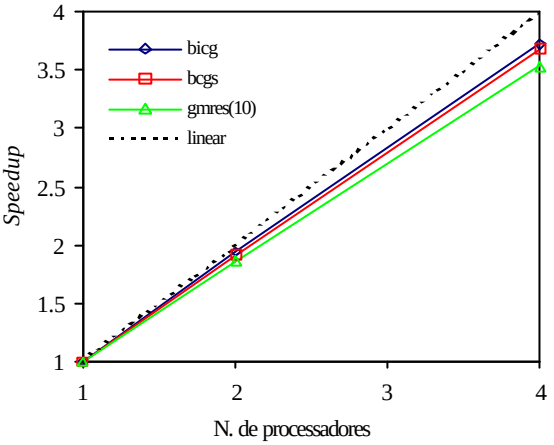

Figura 7.28: Speedup para a malha 300x300 (90.000 eqs.) - IMPES.

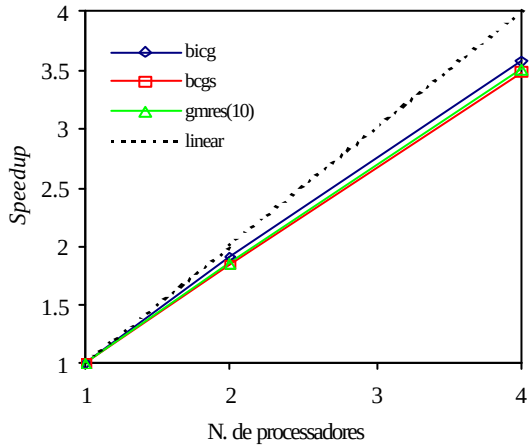

Figura 7.29: Speedup para a malha 400x400 (160.000 eqs.) - IMPES.

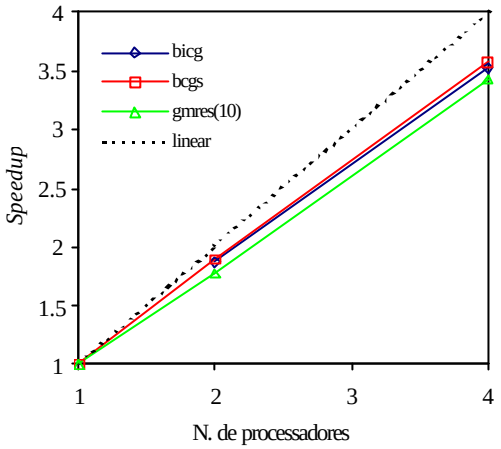

Figura 7.30: Speedup para a malha 150x150 (45.000 eqs.) - totalmente implícita.

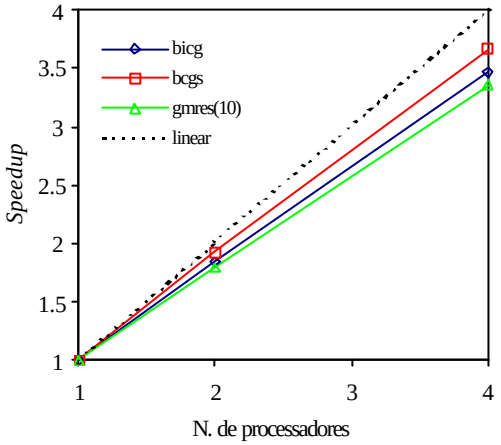

Figura 7.31: Speedup para a malha 200x200 (80.000 eqs.) - totalmente implícita.

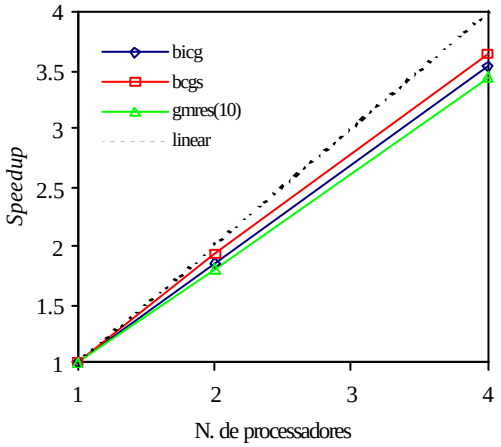

Figura 7.32: Speedup para a malha 240x240 (115.200 eqs.) - totalmente implícita.

		Número de Processadores						
		1	2			4		
Método	Malha	t	t	sp	efi	t	sp	efi
bicg	150x150	116	62	1.87	0.94	33	3.52	0.88
bcgs		93	49	1.90	0.95	26	3.58	0.89
gmres(10)		89	50	1.78	0.89	26	3.42	0.86
bicg	200x200	204	111	1.84	0.92	59	3.46	0.86
bcgs		165	86	1.92	0.96	45	3.67	0.92
gmres(10)		158	88	1.80	0.90	47	3.36	0.84
bicg	240x240	293	158	1.85	0.93	83	3.53	0.88
bcgs		237	123	1.93	0.96	65	3.65	0.91
gmres(10)		227	126	1.80	0.90	66	3.44	0.86

Tabela 7.12: Medidas de desempenho usando a formulação totalmente implícita.

7.2.5 Speedup Escalável

Do ponto de vista prático, uma medida de desempenho mais interessante, e que elimina o efeito de redução da e...ciência devido à parte sequencial do código, consiste no speedup escalável. O que se deseja, agora, é analisar problemas maiores à medida que o número de processadores aumenta, mantendo-se constante o número de equações do sistema em cada processador.

Foram medidos os tempos de execução usando 1, 2 e 4 processadores, e em seguida as medidas de e...ciência e speedup foram obtidas usando as seguintes fórmulas:

$$S_P = \frac{T_1}{T_P} P \qquad \text{e} \qquad E_P = \frac{S_P}{P} \tag{7.1}$$

Neste caso, são utilizadas as malhas 400x400, 566x566 e 800x800, referentes a simulações com 1, 2 e 4 nós, respectivamente, para a formulação IMPES, e 240x240, 340x340 e 480x480, da mesma forma para a formulação totalmente implícita. Com estas malhas, a simulação ocupa praticamente toda a memória RAM disponível dos nós do cluster, permitindo assim resolver problemas com até 0.5 milhão de equações.

A Tabela 7.13 mostra as medidas de desempenho obtidas, conseguindo-se e...ciências entre 87 e 96% em todas as simulações. As Figuras 7.33 e 7.34 mostram grá...cos com speedup e o tempo de execução para os diferentes métodos.

Os grá...cos com tempos de execução mostram que apesar dos tempos terem aumentado com o aumento do problema, as diferenças entre as medidas de tempo são

		Número de Processadores						
		1	2			4		
Formulação	Método	t	t	sp	efi	t	sp	efi
IMPES	bicg	322	341	1.89	0.94	353	3.65	0.91
	bcgs	258	280	1.84	0.92	298	3.46	0.87
	gmres(10)	274	302	1.81	0.91	314	3.49	0.87
Totalmente	bicg	293	313	1.87	0.94	335	3.50	0.87
implícita	bcgs	237	246	1.93	0.96	251	3.78	0.94
	gmres(10)	227	252	1.80	0.90	262	3.47	0.87

Tabela 7.13: Medidas de desempenho obtidas variando o tamanho da malha em função do número de processadores

relativamente pequenas, aproximando-se ao comportamento ideal que seria correspondente a uma reta horizontal. Na formulação IMPES, o método cuja reta apresentou comportamento mais horizontal foi o bicg, enquanto na formulação totalmente implícita foi o bcgs.

Observa-se que o desempenho foi satisfatório em todas as análises realizadas, mostrando que o simulador é aplicável na solução de problemas de larga escala e comprovando a e...ciência do modelo de programação paralela utilizado no seu desenvolvimento.

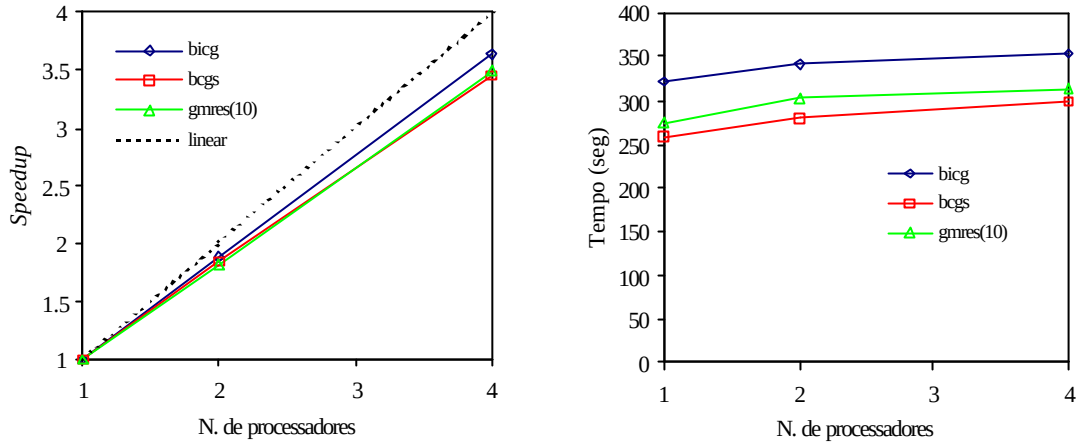

Figura 7.33: Speedup escalável e tempo de execução com malha variável - 400x400 (160.000 eqs.), 566x566 (320.356 eqs.) e 800x800 (640.000 eqs.) - IMPES.

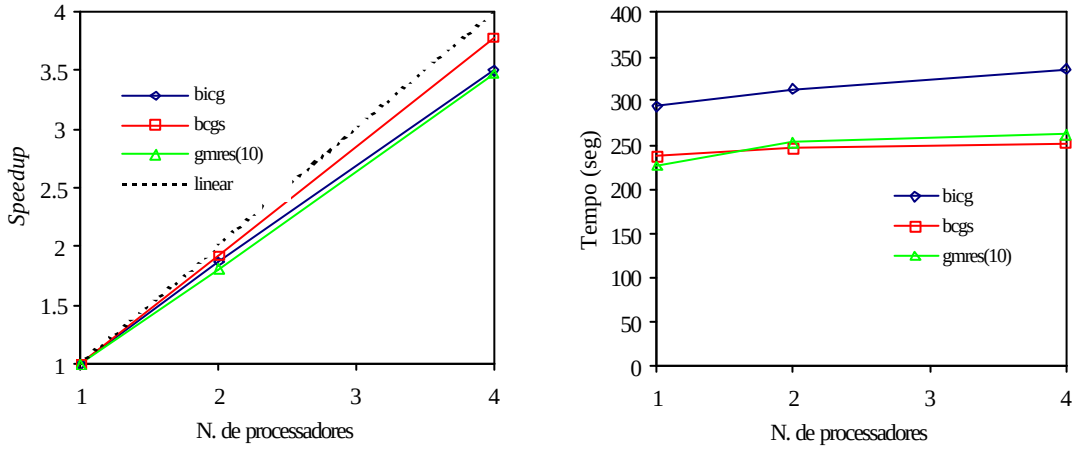

Figura 7.34: Speedup escalável e tempo de execução com malha variável - 240x240 (115.200 eqs.), 340x340 (231.200 eqs.), 480x480 (460.800 eqs.) - totalmente implícita.

Capítulo 8

Conclusões e Trabalhos Futuros

8.1 Conclusões

- O fato de não ter levado em consideração a fase gás, nas equações do modelo Black-Oil, simpli...cou de certo modo o desenvolvimento do simulador pois possibilitou concentrar mais tempo no estudo de outros assuntos envolvidos neste trabalho, entre eles: funcionamento e utilização do sistema operacional Unix e Linux, processamento de alto desempenho, álgebra linear numérica básica, métodos iterativos e técnicas de precondicionamento para solução de sistemas lineares, métodos para solução de sistemas não-lineares, softwares para aplicação em arquiteturas paralelas com memória distribuída (MPI, PETSc, etc) e clusters de PCs.

- A aplicação de computadores paralelos em simulações de reservatórios de larga escala tem se tornado bastante comum nos últimos anos, sendo necessário, portanto, que o engenheiro de reservatórios conheça mais sobre o funcionamento destas arquiteturas e os modelos de programação especí...cos de cada uma, visando obter o máximo desempenho em suas aplicações.

- Do ponto de vista de paralelização dos métodos numéricos, é importante estudar a e...ciência do método dos elementos ...nitos para simulações de reservatórios, visto que o emprego desta técnica vem crescendo cada vez mais nesta área.

- O processo de solução dos sistemas lineares em simulações de reservatórios são responsáveis por 60 a 80% do tempo da simulação, sendo fundamental empregar métodos de solução e...cientes e escaláveis em computadores paralelos.

- A utilização dos solvers do PETSc possibilitou economizar tempo com o desenvolvimento de subrotinas numéricas paralelas básicas para a solução de sistemas de equações, permitindo dedicar mais tempo aos detalhes da paralelização do problema físico.

- Os diversos métodos iterativos de subespaços de Krylov e precondicionadores, disponíveis nos solvers do PETSc, fornecem um ambiente bastante abrangente, possibilitando diferentes análises para encontrar os métodos mais adequados para a solução de um determinado problema, visando assim reduzir seu tempo de execução.

- O emprego do MPI fornece portabiliade ao simulador, permitindo sua aplicação em várias outras arquiteturas de memória distribuída.

- O modelo de programação usado no desenvolvimento do simulador pode ser utilizado na implementação de códigos mais so...sticados, funcionando como um ambiente básico para o desenvolvimento de códigos para aplicações numéricas.

- A comparação dos resultados do simulador desenvolvido com os do simulador BOAST-98, mostrou que as duas formulações implementadas podem ser aplicadas em simulações de reservatórios homogêneos ou heterogêneos, com contorno e geometria irregulares quaisquer.

- A análise de desempenho do simulador no cluster de PCs com até 4 cpus apresentou e...ciências de até 97%, atendendo, portanto, às espectativas de desempenho esperadas, visto que foi realizado pouco esforço - tuning - visando otimizar a e...ciência do código. Como o desempenho do simulador só foi analisado para um problema de reservatório homogêneo, é necessário estudar sua e...ciência na simulação de reservatórios heterogêneos e com maior número de poços.

- O uso de clusters de computadores têm se tornado cada dia mais comum e vem conquistando espaço no lugar dos supercomputadores proprietários. A tendência para os próximos anos é que esta tecnologia se espalhe rapidamente entre as grandes e pequenas empresas, e centros de pesquisa, que tratam problemas de larga escala, principalmente devido a sua baixa relação custo-desempenho comparado a outros computadores paralelos proprietários.

- É necessário avaliar o desempenho do simulador usando mais cpus para resolver problemas com milhões ou dezenas de milhões de equações, que é o objetivo prin-

cipal destes simuladores paralelos. De fato, o problema de comunicação cresce à medida que são adicionados mais nós ao cluster, exigindo mais ajustes nos softwares de comunicação - con…gurações de memória do bu¤er, análise da velocidade de comunicação efetiva entre nós - de modo a conseguir medidas de desempenho adequadas. Como só foi possível utilizar 4 nós nas análises, tais ajustes não foram necessários visto que as medidas de desempenho foram satisfatórias. Contudo, espera-se que com a adição de mais nós no cluster a e…ciência do simulador reduza, sendo necessário identi…car os possíveis problemas responsáveis por esta redução.

8.1.1 Trabalhos Futuros

Visando dar continuidade à linha de pesquisa deste trabalho, cita-se algumas propostas para desenvolvimentos futuros:

- Adicionar a discretização segundo a direção z; admitindo a modelagem de vários layers e possibilitando a utilização de malhas tridimensionais.
- Realizar análises de desempenho em clusters de PCs dedicados com maior número de nós envolvidos.
- Aperfeiçoar o modelo físico e matemático do simulador, implementando o modelo Black-Oil convencional incluindo a fase gasosa.
- Usar o mesmo ambiente computacional para implementar outros modelos físicos: modelo composicional, modelo térmico e modelo de ‡uxo miscível. Isto permite ampliar o campo de aplicação do simulador paralelo.
- Testar e comparar outros métodos iterativos e precondicionadores na solução de problemas de reservatórios, analisando os métodos mais e…cientes para a solução dos sistemas de larga escala resultantes.
- Estudar a aplicação de outros modelos numéricos de discretização como método das streamlines, elementos …nitos e volumes …nitos, visando comparar o desempenho desses modelos em arquiteturas paralelas com memória distribuída.
- Introduzir novas condições de operação nos poços, permitindo, por exemplo, o fornecimento de vazões ou pressões prescritas variáveis durante o processo de simulação, ampliando as opções de aplicação do simulador.

- Estudar a necessidade de utilizar técnicas de computação paralela ou computação distribuída, identi…cando em que situações é preferível paralelizar o código ou distribuir as tarefas entre vários processos.
- Estudar métodos que utilizam o conceito de conexão para o tratamento numérico e montagem dos sistemas de equações de ‡uxo.
- Analisar a questão do balanceamento de carga, admitindo que o modelo possa ser divido em diferentes subdomínios, cada um responsável por um grupo de processos, de modo que a convergência do método numérico de solução seja realizada a nível local em cada subdomínio.
- Implementar diferentes metodologias para leitura e impressão dos dados (leitura e impressão em vários processadores, pré-processadores e pós-processadores).

Bibliografia

ABATE, J.; WANG, P.; SEPEHRNOORI, K.; Parallel Compositional Reservoir Simulation on a Cluster of PCs; Comm. In Numm. Meth. In Eng.; 1999.

ABOU-KASSEM, J. H.; AZIZ, K.; Analytical Well Models for Reservoir Simulation; SPEJ, 573; 1985.

AZIZ, K.; SETTARI, A.; Petroleum Reservoir Simulation; Applied Science Publishers LTd, London; 1979.

BABU, D. K.; ODEH, A. S.; Produtivity of a Horizontal Well; SPERE, 417; 1989.

BALAY, S.; GROPP, W. D.; McINNES, L. C.; SMITH, B. F.; Efficient managment of parallelism in object-oriented numerical libraries, In E. Arge, A. M. Bruaset, and H. P. Langtange, editors, Modern Software Tools in Scientific Computing, pages 163-202. Birkhauser Press; 1997.

BALAY, S.; GROPP, W. D.; McINNES, L. C.; SMITH, B. F.; PETSc Web page, http://www.mcs.anl.gov/petsc.

BALAY, S.; GROPP, W. D.; McINNES, L. C.; SMITH, B. F.; PETSc 2.1.0 Users Manual, Argonne National Laboratory, ANL-95/11 - 2.1.0; 2001.

BARRA, L. P. S.; TOLEDO, E. M.; TELLES, J. C. F.; Parallel Boundary Element Method in Clusters of PC's; Computational Methods in Engineering; 1999:

BARRET, R.; BERRY, M.; CHAN, T. F.; DEMMEL, J.; DONATO, J. M.; DONGARRA, J.; EIJKHOUT, V.; POZO, R.; ROMINE, C.; VORST, H. V. der; Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods, SIAM, http://netlib.org/templates/Templates.html; 1994.

BEAR, J.; Dynamics of Fluids in Porous Media; Dover Publications, New York; 1972.

BECKER, D. J.; STERLING, T.; SAVARESE, D.; FRYXELL, B.; OLSON, K.; Communication Overhead for Space Science Applications on the Beowulf Parallel Workstation, Proceedings, High Performance and Distributed Computing, http://www.beowulf.org/papers; 1995.

CAI, X.; ODEGARD, A.; Parallel Simulation of 3D Nonlinear Acoustic Fields on a Linux-cluster, Proceedings of the Cluster 2000 Conference; 2000.

CAI, X.; ODEGARD, A.; On the Performance of PC Clusters in Solving Partial Di¤erential Equations; Proceedings of the Tenth SIAM Conference on Parallel Processing for Scienti...c Computing; 2001.

CAMERON, K.; Harnessing the Power of PC Clusters, Edinburg Parallel Computing Centre, http://www.epcc.ed.ac.uk/epcc-tec/documents/, Version 1.0; 1996.

COATS, K. H.; et al.; Three-Dimensional Simulation of Steam‡ooding; SPEJ, 573, Trans., AIME, 253; 1974.

CSEP - Computational Science Educational Project; Numerical Linear Algebra; 1995; http://compsci.cas.vanderbilt.edu/csep.html/.

CSEP - Computational Science Educational Project; Computer Archtecture; 1995; http://compsci.cas.vanderbilt.edu/csep.html/.

DAKE, L. P.; Fundamentals of Reservoir Engineering; Elsevier, Amsterdam; 1978.

DEMMEL, J. W.; Applied Numerical Linear Algebra; SIAM, Philadelphia; 1997.

ERTEKIN, T.; ABOU-KASSEM, J. H.; KING, G. R.; Basic Applied Reservoir Simulation; SPE Textbgok Series, Vol. 7; Richardson, TX; 2001.

FANCHI, J. R.; HARPOLE, K. J.; BUJNOWSKI, S. W.; BOAST: A Three-Dimensional, Three-Phase Black Oil Applied Simulation Tool; Vol 1, DOE/BC/10033-3; Oklahoma; 1982.

FOSTER, I. T.; Designing and Building Parallel Programs; Addison-Wesley Publishing Company; 1994.

GROPP, W.; LUSK, E.; MPICH Web page, http://www.mcs.anl.gov/mpi/mpich.

GROPP, W.; LUSK, E.; DOSS, N.; SKJELLUM, A.; A high-performance, portable implementation of the MPI message passing interface standard; Parallel Computing 22, 789-828; 1996.

GROPP, W.; LUSK, E.; SKJELLUM, A.; Using MPI: Portable Parallel Programming with the Message Passing Interface, 2nd Edition, MIT Press; 1999.

HAYDER, M. E.; IEROTHEOU, C. S.; KEYES, D. E.; Three Parallel Programming Paradigms: Comparisons on an Archetypal PDE Computation; CRPC-TR99813; 1999.

HOVLAND, P. D.; McINNES, L. C.; Parallel simulation of compressible ‡ow using automatic di¤erentiation and PETSc, Parallel Computing 27, 503-519; 2001.

JOUBERT, W.; Next Generation Oil Reservoir Simulations, Los Alamos National Laboratory, CUG; 1996.

JOUBERT, W.; JANARDHAN, R.; Evaluation of Linear Solvers for Oil Reservoir Simulation Problems, Part 2: The Fully Implicit Case; 1997.

JOUBERT, W.; BISWAS, D.; Evaluation of Linear Solvers for Oil Reservoir Simulation Problems, Part 1: The IMPES Case; 1997.

KELLEY, C. T.; Iterative Methods for Linear and Nonlinear Equations, Frontiers in Applied Methematics, SIAM; 1996.

KILLOUGH, J. E.; The Application of Parallel Computing to the Flow of Fluids in Porous Media, Computers Chem. Eng., Vol 19, No 6/7, 775-786; 1995.

KIM, J. G.; Advanced Techniques for Oil Reservoir Simulation: Discrete Fracture Model and Parallel Implementation; Dissertação de Doutorado, University of Utah - Department of Chemical and Fuels Engineering, Dezembro; 1999.

LU, Q.; PESZYNSKA, M.; GAI, X.; Implicit Black-oil Model in Ipars Framework; 2001.

MATTAX, C. C.; DALTON, R. L.; Reservoir Simulation; Monograph Series; SPE; Richardson, TX; 1990.

MacDONALD, N., MINTY, E., MALARD, J.; et al.; Writing Message Passing Parallel Programs with MPI; Edinburg Parallel Computing Centre, http://www.epcc.ed.ac.uk/epcc-tec/documents/, Version 2.0; 1999.

MINTY, E.; DAVEY, R.; SIMPSON, A.; HENTY, D.; Decomposing the Potentially Parallel; Edinburg Parallel Computing Centre, http://www.epcc.ed.ac.uk/epcc-tec/documents/, Version 2.0; 1996.

MPI Forum; MPI: A Message Passing Interface Standard, http://www.mcs.anl.gov/mpi/standard.html; 1994.

PARASHAR, M.; WHEELER, J. A.; POPE, G.; WANG, K.; WANG, P.; A New Generation EOS Compositional Reservoir Simulator: Part II - Framework and Multiprocessing; SPE 37977, Symposium on Reservoir Simulation, Austin,TX; 1997.

PARASHAR, M.; YOTOV, I.; An Environment for Parallel Multi-block, Multi-Resolution Reservoir Simulation; Proceedings of the ISCA 11th International Conference on Parallel and Distributed Computing Systems (PDCS 98), Chicago, pp 230-235; 1998.

PEACEMAN, D. W.; Fundamentals of numerical reservoir simulation; Elsevier, Amsterdam; 1977.

RADAJEWSKI, J.; EADLINE, D.; Beowulf HOWTO, Version 1.1.1; 1998.

RAMÉ, M.; DELSHAD, M.; A Compositional Reservoir Simulator on Distributed Memory Parallel Computers; SPE 29103, Symposium on Reservoir Simulation, San Antonio,TX; 1995.

RIDGE, D.; BECKER, D. J.; MERKEY, P.; STERLING, T.; Beowulf: Harnessing the Power of Parallelism in a Pile-of-PCs; Proceedings, IEEE Aerospace, http://www.beowulf.org/papers; 1997.

SAAD, Y.; SCHULTZ, M.; GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymetric Linear Systems, SIAM J. Sci. Stat. Comput., Vol. 7, No 3; 1986.

SAAD, Y.; Iterative Methods for Sparse Linear Systems; PWS Publish Company, Boston; 1996.

SHIRALKAR, G. S.; STEFHENSON, R. E.; JOUBERT, W.; LUBECK, O.; WAANDERS, B. van B.; Falcon: A Production Quality Distributed Memory Reservoir Simulator, SPE 37975, Symposium on Reservoir Simulation , Dallas, TX; 1997.

SILVA, R. S.; RIVELLO, M. F. P.; Using Cluster od PCs to Solve Convection Difusion Problems, Parallel CFD; 2000.

STERLING, T.; BECKER, D. J.; SAVARESE, D.; et al.; Beowulf: A Parallel Workstation for Scienti…c Computation, Proceedings, International Conference on Parallel Processing, http://www.beowulf.org/papers; 1995.

STERLING, T.; SAVARESE, D.; BECKER, D. J.; et al.; Comunication Overhead for Space Science Applications on the Beowulf Parallel Workstation, Proceedings of the Fourth IEEE Symposium on High performance Distributed Computing, (HPDC) http://www.beowulf.org/papers; 1995.

TCHELEPI, H. A.; DURLOFSKY, L. J.; CHEN, W. H.; BERNATH, A.; CHIEN, M. C. H.; Pratical use of scale up and parallel reservoir simulation technologies in ...eld studies; SPE Reservoir Evaluation & Engineering, 2: (4), 368-376; 1999.

TOLEDO, E. M.; SILVA, R. S.; Introdução à Computação Paralela; 2ª Escola de Verão em Computação Cientí...ca - LNCC/CNPq, Rio de Janeiro; 1997.

TREFETHEN, L., N.; BAU III, D.; Numerical Linear Algebra; SIAM, Philadelphia; 1997.

WANG, P.; YOTOV, M.; ARBOGAST, T.; DAWSON, C.; PARASHAR, M.; SEPEHRNOORI, K.; A New Generation EOS Compositional Reservoir Simulator: Part I - Formulation and Discretization; SPE 37979, Symposium on Reservoir Simulation, Austin,TX; 1997.

WANG, P.; BALAY, S.; SEPEHRNOORI, K.; WHEELER, J.; ABATE, J.; SMITH, B.; POPE, G. A.; A Fully Implicit Parallel EOS Compositional Simulator for Large Scale Reservoir Simulation; SPE 51885, 5th Symposium on Reservoir Simulation, Houston, TX; 1999.

WHEELER, M.; ARBOGAST, T.; BRYANT, S.; EATON, J.; LU, Q.; PESZYNSKA, M.; A Parallel Multiblock/Multidomain Aproach for Reservoir Simulation, SPE 51884, Symposium on Reservoir Simulation , Houston, TX; 1999.

ZHANG, K.; WU, Y. S.; DING, C.; PRUESS, K.; ELMROTH, E.; Parallel Computing Techinics for Large-Scale Reservoir Simulation of Multi-Component Fluid Flow, SPE 66343, Symposium on Reservoir Simulation , Houston, TX; 2001.

ZIENKIEWICS, O. C.; MORGAN, K.; Finite Elements and Approximation; Wiley-Interscience Publication; 1982.