

UNIVERSIDADE FEDERAL DE SANTA CATARINA
CURSO DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA

**Métodos de Krylov-Newton Aplicados à Simulação Numérica de
Reservatórios de Petróleo**

DISSERTAÇÃO SUBMETIDA À UNIVERSIDADE FEDERAL DE SANTA CATARINA
PARA OBTENÇÃO DO GRAU DE MESTRE EM ENGENHARIA

RODRIGO MACHADO LUCIANETTI

FLORIANÓPOLIS, SETEMBRO DE 2000

**MÉTODOS DE KRYLOV-NEWTON APLICADOS À SIMULAÇÃO NUMÉRICA DE
RESERVATÓRIOS DE PETRÓLEO**

RODRIGO MACHADO LUCIANETTI

ESTA DISSERTAÇÃO FOI JULGADA ADEQUADA PARA A OBTENÇÃO DO TÍTULO DE MESTRE EM ENGENHARIA ESPECIALIDADE ENGENHARIA MECÂNICA, ÁREA DE CONCENTRAÇÃO ENGENHARIA E CIÊNCIAS TÉRMICAS, APROVADA EM SUA FORMA FINAL PELO CURSO DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA.

Prof. CLOVIS RAIMUNDO MALISKA, Ph.D.

ORIENTADOR

Prof. JÚLIO CÉSAR PASSOS, Dr. Eng. Mec.

COORDENADOR DO CURSO DE PÓS-GRADUAÇÃO

BANCA EXAMINADORA

Prof. ANTÔNIO FÁBIO CARVALHO DA SILVA, Dr. Eng. Mec.

Prof. JÁUBER CAVALCANTE OLIVEIRA, Ph.D.

Prof. MÁRIO CÉSAR ZAMBALDI, Dr.

Aos meus pais, Vera Lúcia e Paulino, aos meus sogros, Maria da Glória e Roberto, e
especialmente a minha esposa Francini.

AGRADECIMENTOS

Ao meu orientador e amigo Prof. Clovis R. Maliska pelos conhecimentos transmitidos, pela confiança em mim depositada e, acima de tudo, pelo seu exemplo de motivação e amor pela ciência, que influenciaram minha decisão de trilhar o mesmo caminho.

Ao meu grande amigo Alexandre de O. Czesnat pelas valiosas discussões e inúmeras contribuições ao trabalho.

Ao amigo Clovis Maliska Jr. pela motivação e imprescindível suporte na programação do simulador por mim utilizado neste trabalho.

Ao amigo Axel Dihlman pelo estímulo constante e pelo prazer de durante este trabalho poder contar com a sua companhia.

Aos demais colegas do SINMEC pelo excelente ambiente de trabalho proporcionado.

A ESSS que, conciente da importância da formação acadêmica, me proporcionou o tempo e os recursos necessários para terminar este trabalho.

Aos contribuintes brasileiros que, através do CNPq, financiaram este trabalho.

A todos os meus professores do curso de Pós-Graduação do Depto. de Engenharia Mecânica da UFSC.

SUMÁRIO

1	<u>INTRODUÇÃO</u>	1
1.1	PRELIMINARES	1
1.2	REVISÃO BIBLIOGRÁFICA	4
1.3	OBJETIVOS E CONTRIBUIÇÕES	9
1.4	ESCOPO DO TRABALHO	10
2	<u>FORMULAÇÃO MATEMÁTICA</u>	11
2.1	LEI DE CONSERVAÇÃO DA MASSA	11
2.2	FORMULAÇÃO EM FRAÇÕES MÁSSICAS GLOBAIS	13
2.3	LEI DE DARCY	14
2.4	MODELO <i>BLACK-OIL</i> PADRÃO	15
2.5	CONDIÇÕES DE CONTORNO E INICIAIS	19
3	<u>FORMULAÇÃO NUMÉRICA</u>	21
3.1	TRANSFORMAÇÃO DAS EQUAÇÕES GOVERNANTES PARA O SISTEMA COORDENADO CURVILÍNEO	21
3.2	DISCRETIZAÇÃO DAS EQUAÇÕES GOVERNANTES	23
3.3	SISTEMA DE EQUAÇÕES NÃO-LINEARES	27
4	<u>MÉTODOS DE SOLUÇÃO DE SISTEMAS DE EQUAÇÕES NÃO-LINEARES</u>	29
4.1	VISÃO GERAL DO PROBLEMA	29
4.2	MÉTODOS DE NEWTON	31
4.2.1	O CONCEITO DE LINEARIZAÇÃO	32
4.2.2	FORMA GERAL DOS MÉTODOS DE NEWTON	33
4.2.3	MÉTODOS DE NEWTON INEXATOS	33
4.2.4	MÉTODOS DE KRYLOV-NEWTON	38
4.2.5	MONTAGEM DA MATRIZ JACOBIANA	41
5	<u>MÉTODOS DE KRYLOV-NEWTON APLICADOS À SOLUÇÃO DAS- EQUAÇÕES DO MODELO <i>BLACK-OIL</i> PADRÃO</u>	43
5.1	LINEARIZAÇÃO DAS EQUAÇÕES DISCRETIZADAS	43
5.2	OS MÉTODOS DE KRYLOV-NEWTON	48
5.3	CONTROLE AUTOMÁTICO DO PASSO DE TEMPO	51
5.4	OUTROS DETALHES RELATIVOS À IMPLEMENTAÇÃO	53
6	<u>RESULTADOS E DISCUSSÕES</u>	55
6.1	VALIDAÇÃO DA IMPLEMENTAÇÃO	55
6.2	PARALELEPÍPEDO DEFORMADO	59
6.2.1	<i>BACKTRACKING</i> E <i>FORCING TERMS</i>	63
6.2.2	<i>SOLVERS</i> E PRECONDICIONADORES	74

6.2.3	AVALIAÇÃO DOS MÉTODOS DE KRYLOV-NEWTON	79
6.3	RESERVATÓRIO REAL	86
7	CONCLUSÕES E RECOMENDAÇÕES	89
7.1	CONCLUSÕES	89
7.2	RECOMENDAÇÕES	91
	APÊNDICE A - AVALIAÇÃO DAS DERIVADAS DOS POTENCIAIS	93
	REFERÊNCIAS BIBLIOGRÁFICAS	95

RESUMO

A simulação do processo de extração de óleo em um reservatório de petróleo é um problema complexo que envolve o escoamento multifásico em um meio poroso de geometria arbitrária, podendo conter falhas geológicas e outras irregularidades. A modelagem numérica implícita deste problema requer a solução de um complexo sistema de equações não-lineares que pode ser resolvido de diversas formas, mais ou menos eficientes. A forma mais comum de solução deste sistema emprega o método de linearização de Newton ou alguma de suas variantes.

O foco deste trabalho está na análise de uma extensa classe de métodos de Newton conhecidos como métodos de Newton inexatos. Estes métodos empregam um segundo método iterativo na solução da equação linearizada resolvendo-a até uma certa tolerância; daí serem chamados de inexatos. Os métodos inexatos abordados neste trabalho concentram-se na escolha de processos iterativos baseados em métodos de projeção no espaço de Krylov, tais como: GMRES, BICGSTAB, TFQMR. Estes métodos, que formam uma subclasse dos métodos de Newton inexatos, são conhecidos como métodos de Krylov-Newton. São abordados também métodos quase-newtonianos baseados na aproximação da matriz Jacobiana considerando apenas os termos ortogonais da discretização em sistemas curvilíneos não-ortogonais.

Vários problemas com diferentes graus de não-ortogonalidade da malha e não-linearidades são resolvidos e comparados usando os diferentes métodos implementados.

ABSTRACT

The petroleum reservoir simulation involves multiphase flows in porous media with a complex geometry, containing geological faults and other irregularities. The implicit solution of the problem requires the simultaneous solution of a large number of complex non-linear equations. This system of non-linear equations can be solved by several methods with different performances. The standard way employs some linearization method of Newton form.

This work concentrates in the analysis of a set of Newton form's methods called inexact Newton methods. These methods employ a secondary linear iterative process to find the solution to the linearized equation. The methods are called inexact because the convergence criterion for the solution of the linearized equation can be fairly relaxed, while the criterion for the convergence of the overall scheme to find the solution to the problem is still reasonably tight. The inexact methods used in this work apply Krylov solvers as the inner iterative process. These methods are usually called Krylov-Newton methods. Other two schemes based on the approximation of the Jacobian matrix are also introduced. This approximation is built considering only the orthogonal terms in a generalized discretization technique.

Several cases, with different levels of non-orthogonality and non-linearity, are solved and compared using all implemented Newton-like methods.

LISTA DE FIGURAS

Figura 2.1 – Fluxos de massa através do volume de controle	11
Figura 3.1 – Domínio físico e transformado	21
Figura 3.2 – Volume de controle elementar no espaço transformado	24
Figura 3.3 – Volumes presentes no esquema numérico	25
Figura 5.1 – Matriz de 19 diagonais (Ex.: malha 4x4x3)	44
Figura 5.2 – Matriz de 7 diagonais (Ex.: malha 4x4x3)	50
Figura 6.1 – Distribuição e configuração dos poços	56
Figura 6.2 – Malha utilizada para o caso dos 3 poços (24x10)	56
Figura 6.3 – Gráfico do corte de água nos poços produtores	58
Figura 6.4 - Geometria e malha utilizadas nos testes	59
Figura 6.5 – $\mathbf{a} = 45^\circ$ e $n = 2,0$, sem <i>backtracking</i> (19 diag., GMRES(25), BILU 0)	65
Figura 6.6 – $\mathbf{a} = 45^\circ$ e $n = 2,0$, sem <i>backtracking</i> (19 diag., GMRES(25), BILU 0)	66
Figura 6.7 – $VPI = 1,5$, $\mathbf{a} = 45^\circ$ e $n = 2,0$, sem <i>backtracking</i> (19 diag., GMRES(25), BILU 0)	66
Figura 6.8 – $VPI = 0,2$, $\mathbf{a} = 45^\circ$ e $n = 2,0$, sem <i>backtracking</i> (19 diag., GMRES(25), BILU 0)	67
Figura 6.9 – Influência da não-ortogonalidade nos métodos de Krylov-Newton para $n = 2,0$	82
Figura 6.10 – Influência da não-linearidade nos métodos de Krylov-Newton para $\mathbf{a} = 45^\circ$	83
Figura 6.11 – $VPI = 0,2$ para $\mathbf{a} = 45^\circ$ e $n = 2,0$	84
Figura 6.12 – $VPI = 1,0$ para $\mathbf{a} = 45^\circ$ e $n = 2,0$	85
Figura 6.13 – $VPI = 1,5$ para $\mathbf{a} = 45^\circ$ e $n = 2,0$	85
Figura 6.14 – Malha 10x10x4 (\mathbf{x} , \mathbf{h} , \mathbf{g}) e disposição dos poços	86

LISTA DE TABELAS

Tabela 6.1 – Dados do problema dos 3 poços	56
Tabela 6.2 – Permeabilidade relativa do problema dos 3 poços	57
Tabela 6.3 – Dados do modelo físico (Paralelepípedos Deformados)	60
Tabela 6.4 – Dados do modelo numérico (Paralelepípedos Deformados)	62
Tabela 6.5 – $\mathbf{a} = 45^\circ$, com <i>backtracking</i> e método 1 (19 diag., GMRES(25), BILU 0)	63
Tabela 6.6 – $\mathbf{a} = 45^\circ$, sem <i>backtracking</i> e método 1 (19 diag., GMRES(25), BILU 0)	63
Tabela 6.7 – $\mathbf{a} = 60^\circ$, sem <i>backtracking</i> (19 diag., GMRES(25), BILU 0)	68
Tabela 6.8 – $\mathbf{a} = 45^\circ$, sem <i>backtracking</i> (19 diag., GMRES(25), BILU 0)	69
Tabela 6.9 – $\mathbf{a} = 30^\circ$, sem <i>backtracking</i> (19 diag., GMRES(25), BILU 0)	70
Tabela 6.10 – $\mathbf{a} = 90^\circ$, sem <i>backtracking</i> (7 diag., GMRES(25), BILU 1)	71
Tabela 6.11 – $\mathbf{a} = 45^\circ$, sem <i>backtracking</i> (7 diag., GMRES(25), BILU 1)	72
Tabela 6.12 – $\mathbf{a} = 30^\circ$, sem <i>backtracking</i> (7 diag., GMRES(25), BILU 1)	72
Tabela 6.13 – $\mathbf{a} = 60^\circ$, sem <i>backtracking</i> (<i>Matrix-free</i> , GMRES(25), BILU 2)	72
Tabela 6.14 – $\mathbf{a} = 45^\circ$, sem <i>backtracking</i> (<i>Matrix-free</i> , GMRES(25), BILU 2)	73
Tabela 6.15 – $\mathbf{a} = 30^\circ$, sem <i>backtracking</i> (<i>Matrix-free</i> , GMRES(25), BILU 2)	74
Tabela 6.16 – $\mathbf{a} = 45^\circ$ e $n = 1,5$ (19 diag.)	75
Tabela 6.17 – $\mathbf{a} = 45^\circ$ e $n = 2,0$ (19 diag.)	76
Tabela 6.18 – $\mathbf{a} = 45^\circ$ e $n = 1,5$ (7 diag.)	76
Tabela 6.19 – $\mathbf{a} = 45^\circ$ e $n = 2,0$ (7 diag.)	77
Tabela 6.20 – $\mathbf{a} = 45^\circ$ e $n = 1,5$ (<i>Matrix-free</i>)	77
Tabela 6.21 – $\mathbf{a} = 45^\circ$ e $n = 2,0$ (<i>Matrix-free</i>)	78
Tabela 6.22 – Memória consumida pelos condicionadores (em Kbytes) em função da estrutura da matriz Jacobiana	78
Tabela 6.23 – Distribuição do tempo de processamento do problema linear em % do tempo total de CPU (GMRES, $\mathbf{a} = 45^\circ$ e $n = 1,5$)	79
Tabela 6.24 – Parâmetros numéricos usados em cada método de Krylov-Newton	80
Tabela 6.25 – Comparação entre os métodos de Krylov-Newton para vários valores de \mathbf{a} e n	80
Tabela 6.26 – Razão $\text{CPU}^7 / \text{CPU}^{19}$	83
Tabela 6.27 – Razão $\text{CPU}^7 / \text{CPU}^{\text{mf}}$	83
Tabela 6.28 – Razão $\text{CPU}^{\text{mf}} / \text{CPU}^{19}$	84
Tabela 6.29 – Parâmetros numéricos usados em cada método de Krylov-Newton	87
Tabela 6.30 – Comparação entre os métodos de Krylov-Newton na simulação de um reservatório real	87
Tabela 6.31 – Razão entre tempos de CPU	88

SIMBOLOGIA

A	Área transversal	$[\text{m}^2]$
A	Matriz da equação linear	
B_p	Fator volume de formação da fase p	$[\text{m}^3/\text{m}^3]$
c_p	Compressibilidade da fase p	$[\text{Pa}^{-1}]$
D	Profundidade	$[\text{m}]$
\mathbf{D}_{cp}	Coeficiente efetivo de dispersão do componente c na fase p	$[\text{m}^2/\text{s}]$
F	Função resíduo	
J	Jacobiano da transformação de coordenadas	
\mathbf{J}	Matriz Jacobiana	
k_{rp}	Permeabilidade relativa da fase p	[adimensional]
\mathbf{K}	Permeabilidade absoluta	$[\text{m}^2]$
\mathbf{K}_m	Subespaço de Krylov de dimensão m	
$\dot{\mathbf{m}}_{cp}$	Fluxo de massa do componente c na fase p	$[\text{kg}/\text{s}]$
m_p	Vazão mássica da fase p por unidade de volume do reservatório	$[\text{kg}/\text{m}^3 \cdot \text{s}]$
M_k	Bloco de memória na iteração k	
M_p	Massa da fase p	$[\text{kg}]$
M_{cp}	Massa do componente c na fase p	$[\text{kg}]$
M_E	Matriz de condicionamento à esquerda	
M_D	Matriz de condicionamento à direita	
\mathbf{n}	Vetor normal	

P_p	Pressão da fase p	[Pa]
P_{C_s}	Pressão capilar no sistema bifásico s	[Pa]
q_p	Vazão da fase p por unidade de volume	[m ³ /s.m ³]
r_k	Resíduo da equação linear na iteração k	
r_{cp}	Taxa de massa do componente c na fase p gerada por reações químicas	[kg/s]
R_S	Razão de solubilidade do gás na fase óleo	[m ³ /m ³]
\hat{s}^k	Passo do método de Newton inexato	
S_p	Saturação da fase p	[m ³ /m ³]
t	Tempo	[s]
\mathbf{u}_p	Velocidade da fase p	[m/s]
V_p	Volume da fase p	[m ³]
V_{cp}	Volume ocupado por uma quantidade fixa de massa do componente c na fase p	[m ³]
x, y, z	Coordenadas cartesianas	[m]
x^k	Vetor solução na iteração k	
X_{cp}	Fração mássica do componente c na fase p	[kg/kg]
Z_c	Fração mássica global do componente c	[kg/kg]
Gregos		
l_p	Mobilidade da fase p	[m ⁴ /N.s]
f	Porosidade	[m ³ /m ³]

Φ_p	Potencial da fase p	[Pa]
\mathbf{m}_p	Viscosidade da fase p	[N.s/m ²]
ξ_p	Peso específico da fase p	[Pa/m ³]
\mathbf{r}_m	Densidade média da mistura	[kg/m ³]
$\mathbf{x}, \mathbf{h}, \boldsymbol{\xi}$	Coordenadas curvilíneas	
\mathbf{h}_k	<i>Forcing term</i> na iteração k	
w_k	Fator de relaxação na iteração k	

Subíndices e Superíndices

w	Água
o	Óleo
g	Gás
ow	Sistema óleo/água
og	Sistema óleo/gás
c	Componente
e, w, n, s, t, b	Interfaces do volume de controle
E, W, N, S, etc	Volumes vizinhos ao volume P
k	Nível iterativo
p	Fase
R	Condições de reservatório
STD	Condições do tanque de armazenamento na superfície (condições <i>standard</i>)

1 Introdução

1.1 Preliminares

O principal objetivo em um estudo de reservatório de petróleo é a predição da sua performance futura e a procura de métodos e meios capazes de extrair o maior volume de óleo possível. A simulação de reservatórios vem ao longo dos anos se firmando como uma ferramenta indispensável ao engenheiro de petróleo na realização deste estudo; sobretudo a simulação numérica, já que a simulação completa em laboratório é impossível de ser realizada.

Apesar de sua reconhecida utilidade, a simulação numérica de reservatórios é uma tarefa complexa e muitas vezes até frustrante para o engenheiro, que tem de lidar com uma série de questões que vão, desde a escolha do modelo matemático aplicável ao reservatório e ao fluido em estudo, até a análise dos resultados e o conseqüente ajuste de dados visando aproximar o histórico de recuperação de óleo real.

Este trabalho, entretanto, concentra-se apenas em determinados aspectos numéricos de um simulador baseado no modelo *Black-Oil*. Este modelo é o mais simples de todos os modelos de reservatório, porém apresenta todas as principais características também presentes em modelos mais complexos, sendo, por isso, ideal para o estudo acadêmico. Como é mostrado ao longo do texto, as análises e conclusões feitas neste trabalho também podem ser estendidas a simuladores que empregam modelos matemáticos mais complexos.

Desde a introdução dos métodos numéricos na engenharia de petróleo e ainda hoje, a discretização baseada em coordenadas cartesianas tem sido extensivamente usada devido a sua fácil construção, simplicidade das equações discretizadas e rápida implementação do código computacional. Entretanto, a discretização cartesiana apresenta consideráveis desvantagens para modelar satisfatoriamente reservatórios com geometrias complexas, falhas geológicas e/ou configurações complexas de poços. Visando contornar estes problemas, surgiram modelos mais eficientes no tratamento da geometria de reservatórios e, para isso, tanto malhas estruturadas, quanto não-estruturadas, têm sido exploradas.

No campo das malhas estruturadas cada vez mais se amplia o uso de sistemas curvilíneos não-ortogonais ou generalizados permitindo uma maior versatilidade na discretização de complexos reservatórios. Todavia, este tipo de discretização requer um esquema numérico envolvendo mais pontos do que somente os sete necessários para uma discretização ortogonal tridimensional, por exemplo. As implicações decorrentes da introdução de mais pontos no esquema numérico pela discretização não-ortogonal serão analisadas em detalhe ao longo deste trabalho.

O modelo *Black-Oil* padrão é representado matematicamente por um conjunto de equações diferenciais parciais não-lineares obtidas do balanço de massa para cada componente do escoamento multifásico e do uso da aproximação semi-empírica de Darcy no lugar das equações de conservação da quantidade de movimento, tendo como incógnitas a pressão do óleo e as saturações de cada fase ou a fração mássica de cada componente. Atualmente duas metodologias de solução destas equações são largamente empregadas em diferentes simuladores de petróleo: o método IMPES (*Implicit Pressure - Explicit Saturation*) e o método Totalmente Implícito.

No método IMPES os coeficientes das equações são avaliados no nível de tempo anterior, de maneira que é possível desacoplar a pressão da saturação obtendo-se uma equação envolvendo apenas a pressão. Resolvendo o campo de pressão, obtém-se as saturações de forma explícita, ou seja, resolve-se apenas um conjunto de equações algébricas para as saturações. Apesar do menor sistema linear a ser resolvido (envolve apenas a pressão) e dá atrativa facilidade de vetorização, este método exige o uso de pequenos intervalos de tempo para manter a estabilidade da solução, principalmente quando aplicado à solução de problemas envolvendo regiões de alta vazão e/ou elevados gradientes de saturação. Já, no método Totalmente Implícito, os coeficientes presentes nas equações discretizadas são avaliados no nível de tempo mais atual e todo o conjunto de equações é resolvido simultaneamente, resultando num maior custo computacional por passo de tempo. Em compensação, este método é muito estável, permitindo o uso de intervalos de tempo maiores. O método Totalmente Implícito permite ainda que uma série de estratégias numéricas sejam empregadas para a solução do conjunto de equações não-lineares acopladas; dentre as quais destacam-se o método de Newton e suas variantes.

Atualmente, há um grupo de variantes do método de Newton clássico que vêm recebendo cada vez mais atenção da academia e da indústria. Estes métodos, denominados de

métodos de Newton inexatos, empregam um segundo processo iterativo para achar uma solução aproximada para a equação linearizada; em contraste com o método de Newton tradicional que resolve esta equação através de um método direto. Isto é particularmente atrativo para problemas derivados de processos de discretização, pois normalmente geram uma matriz Jacobiana esparsa.

Na verdade, a linearização de Newton gera uma equação linear envolvendo a matriz Jacobiana, um vetor solução que representa a variação do vetor incógnita (que deve tender a zero) e um vetor de resíduos que também deve ser levado a zero quando o processo se aproximar da solução. É possível substituir a matriz Jacobiana real por uma aproximação desde que a equação resultante continue zerando os vetores resíduo e variação. Esta técnica nos remete a uma outra classe de métodos de Newton, chamados de quase-newtonianos. A idéia é substituir uma iteração de Newton completa por uma mais simples e barata computacionalmente pelo preço de uma taxa de convergência mais lenta, ou seja, por uma maior quantidade de iterações. Mais recentemente, foram desenvolvidos métodos que nem sequer constroem a matriz Jacobiana, trabalhando diretamente com o produto Jacobiana-vetor. São os chamados métodos *matrix-free*.

Como já dito anteriormente, o emprego de coordenadas generalizadas gera um esquema numérico que envolve um maior número de volumes. Sendo assim, a matriz de coeficientes para o método de Newton terá uma maior largura de banda e um maior número de condição (*condition number*); o que, em tese, acarretará em um maior espaço de memória requerido, um maior tempo de computação e dificuldades de convergência em malhas fortemente não-ortogonais. Uma aproximação da matriz Jacobiana utilizando apenas os termos ortogonais pode então ser usada, gerando um método quase-newtoniano. Ou ainda, pode-se lançar mão de métodos *matrix-free* para se ter uma linearização aproximada que leva em conta todos os termos e utilizar a parte ortogonal da matriz Jacobiana apenas como matriz de condicionamento. Os métodos quase-newtonianos podem ainda empregar *solvers* iterativos para a solução da equação linearizada, tornando-se também inexatos.

Dos *solvers* lineares conhecidos, os mais difundidos para o uso em métodos de Newton inexatos são os métodos da família de Krylov, normalmente empregados junto com condicionadores. Por sua vasta popularidade na solução de sistema de equações não-lineares, estes métodos hoje são mais conhecidos na literatura por métodos de Krylov-Newton.

A seguir, é apresentada a evolução dos diferentes aspectos numéricos abordados na construção do simulador *Black-Oil* adotado neste estudo.

1.2 Revisão Bibliográfica

Segundo Aziz e Settari (1979), há basicamente dois importantes modelos matemáticos para o escoamento em reservatórios de petróleo: o escoamento mono ou multifásico onde mais de dois componentes hidrocarbonetos são considerados e o escoamento multifásico em que a composição de hidrocarbonetos pode ser aproximada por apenas dois componentes: um componente não volátil (*black-oil*) e um componente volátil (gás) solúvel na fase óleo. Este trabalho aborda apenas questões numéricas referentes ao segundo modelo matemático, também conhecido como Modelo- β ou Modelo *Black-Oil*.

Neste modelo considera-se o escoamento simultâneo de três fases – gás, óleo e água – a três componentes – gás, óleo e água. Normalmente a fase água é a fase molhante, a fase óleo tem molhabilidade intermediária e a fase gás é a fase não-molhante. Assume-se também que água e óleo são imiscíveis, não trocam massa entre si e não mudam de fase. Todavia, o componente gás pode estar dissolvido em óleo, mas não em água. Considera-se também que os fluidos estão a temperatura constante e em equilíbrio termodinâmico no reservatório. Segundo Cunha (1996), o modelo *Black-Oil* é amplamente usado na indústria de petróleo e aplica-se, especialmente, na modelagem de reservatórios contendo óleos pesados ou de baixa volatilidade.

Para formular matematicamente o modelo *Black-Oil*, normalmente utiliza-se como variáveis primárias a pressão do óleo e as saturações de cada fase. Exemplos de modelos numéricos que utilizam este tipo de formulação podem ser encontrados em Peaceman (1977), Aziz e Settari (1979), Prais e Campagnolo (1991) e Cunha (1996). Todavia, este tipo de formulação apresenta sérios problemas em escoamentos envolvendo uma fase gasosa. Isto ocorre porque, em algumas regiões do reservatório, devido às condições termodinâmicas, a fase gás pode desaparecer, pois o componente gás encontra-se completamente dissolvido na fase óleo. Nestas condições, a equação da fase gás da formulação baseada em saturações torna-se irrelevante. Como é mostrado nos estudos realizados por Abou-Kassem e Aziz (1985) e Forsyth e Sammon (1984), isso implica na utilização de esquemas envolvendo a substituição

de variável e equação ou que impeçam que a fase gás desapareça, forçando a existência de um valor residual especificado para a saturação de gás. Visando contornar estas dificuldades, desenvolveu-se a formulação que utiliza as frações mássicas globais de cada componente em lugar das saturações das fases. Esta técnica, reportada por Prais e Campagnolo (1991) e Maliska *et al.* (1997), é usada neste trabalho e, portanto, será revista em detalhes mais adiante.

Uma vez definido o modelo matemático, parte-se para a modelagem numérica do problema, onde a principal decisão recai sobre o método de discretização das equações. Este trabalho concentra-se no método dos volumes finitos em coordenadas generalizadas.

O uso de coordenadas generalizadas, também conhecida como coordenadas coincidentes com a fronteira (*boundary-fitted coordinates*), na solução de problemas envolvendo o escoamento de fluidos, experimentou um grande avanço depois da metade da década de 70 com os trabalhos de Thompson *et al.* (1976), sendo, a princípio, aplicadas na solução de problemas de aerodinâmica empregando diferenças finitas. Hoje em dia, esta técnica é largamente difundida e usada para resolver diferentes problemas, entre eles a simulação numérica de reservatórios de petróleo.

Neste campo, os primeiros trabalhos envolvendo sistemas curvilíneos restringiam-se apenas a sistemas ortogonais, que, se por um lado permitem um tratamento eficiente da geometria aliado a um esquema numérico simples, por outro reduzem drasticamente os tipos de geometrias capazes de serem discretizadas. Nesta linha, pode-se citar os trabalhos de Hirasaki e O'Dell (1970), Sonier e Chaumet (1974), Robertson e Woo (1978), Fleming (1984) e Sharpe e Anderson (1990;1991). Neste último, os autores propõem um interessante método alternativo para a geração de malhas curvilíneas ajustadas a quaisquer contornos internos e externos. Estas malhas são geradas resolvendo-se um conjunto de equações diferenciais parciais parabólicas quase-lineares, em contraste com a geração elíptica normalmente usada na criação de malhas curvilíneas, podendo gerar malhas aproximadamente ortogonais mesmo em reservatórios complexos. Entretanto, ao tentar satisfazer a condição de ortogonalidade, acaba por criar grande adensamento de volumes até em regiões onde não haveria necessidade. Só mais recentemente, surgiram estudos da aplicação de sistemas não-ortogonais em reservatórios, dos quais destacam-se os de Britto *et al.* (1991), Aavatsmark *et al.* (1994) e Maliska *et al.* (1994, 1997). Estes métodos, apesar de trazerem complexidades adicionais ao esquema numérico – já agora as conexões entre volumes deixam de ser ortogonais as faces dos mesmos – permitem a discretização de domínios bem mais complexos, especialmente

quando a geometria do problema é tridimensional.

Repetindo o que já foi dito nas preliminares, o modelo *Black-Oil* é representado por um sistema de equações diferenciais parciais. Este pode ser resolvido simultaneamente – ou seja, todas as equações acopladas – ou, como é mais comum, desacoplando a pressão das saturações ou frações mássicas. Dos métodos segregados, que desacoplam as equações, sem dúvida nenhuma o mais famoso e empregado na indústria é o método IMPES – *Implicit Pressure / Explicit Saturation* – que teve origem nos trabalhos de Sheldon et al. (1959) e Stone e Garder Jr. (1961). Este método é similar ao método PRIME (Maliska, 1981) empregado na modelagem das equações de Navier-Stokes. Apesar de largamente utilizado, o método IMPES exige um passo de tempo limitado por questões de estabilidade devido ao tratamento explícito das saturações. Visando melhorar a estabilidade, MacDonald e Coats (1970) sugerem a utilização de um tratamento implícito também às saturações, sem, no entanto, acoplar todas as equações, dando origem ao método sequencial ou SEQ, reportado mais tarde nos trabalhos de Spillete et al. (1973), Coats et al. (1974) e Coats (1976). O método de solução simultânea ou SS, originou-se dos estudos de Douglas Jr. et al. (1959), sendo em seguida explorado por vários autores (Coats et al., 1967; Coats, 1968; Sheffield, 1969). Mais recentemente, surgiram métodos híbridos, mais conhecidos como métodos Adaptativos, que tratam implicitamente apenas os volumes que violam algum critério de estabilidade, gerando um processo iterativo mais eficiente. Destes, pode-se citar o método proposto por Thomas e Thurnao (1983), o método AIN (Bertiger e Kelsey, 1985) – *Adaptative Inexact Newton Method* – e o método AIM – *Adaptative Implicit Method* – proposto por Forsyth e Sammon (1986). Entretanto, como é esclarecido por Russel (1989), apenas o último método pode realmente ser considerado como uma alternativa aos métodos SS e IMPES, sendo, os dois primeiros, métodos simultâneos onde um algoritmo quase-newtoniano é empregado na solução das não-linearidades. Isto pode ser verificado com o fato de ambos os métodos convergirem para a solução simultânea. Detalhes a respeito dos métodos IMPES e SEQ são encontrados em Aziz e Settari (1979) e maiores informações sobre os métodos Adaptativos podem ser achadas em Marcondes (1996). O método SS é o método empregado neste trabalho.

Segundo Aziz e Settari (1979), as não linearidades encontradas nas equações do modelo *Black-Oil* podem ser classificadas em dois tipos: fracas e fortes. As fracas estão presentes nas propriedades do fluido e reservatório que são funções da pressão e não requerem tratamento

especial. Já as não-linearidades fortes encontram-se nos coeficientes que dependem da saturação ou da pressão capilar. Do método usado para resolvê-las, dependerá a robustez e precisão do simulador. As não-linearidades devem ser aproximadas no espaço e no tempo. O método usado para aproximar as não-linearidades fortes no espaço também é chamado de função interpolação e constitui, por si só, num dos principais ramos de pesquisa em simulação numérica. Inúmeros trabalhos foram e continuam sendo devotados ao tema e muitos avanços têm sido obtidos na área. Uma discussão mais detalhada do tema é encontrada no trabalho de Czesnat (1999). Neste trabalho optou-se por empregar o método UDS – *Upstream Differencing Scheme* – devido a sua robustez e por ser extensivamente usado nos simuladores comerciais, a despeito de sua alta dispersão numérica decorrente da aproximação ser de primeira ordem. Com relação a solução temporal das não-linearidades, também há várias técnicas que podem ser empregadas, que variam desde o tratamento explícito das transmissividades até o tratamento totalmente implícito usando o método de Newton. Uma discussão detalhada de várias destas técnicas é encontrada no texto de Aziz e Settari (1979), do qual é reproduzido aqui um trecho que resume a principal característica destas técnicas: “Conforme a ‘implicitude’ do método aumenta, sua estabilidade melhora, porém os erros de truncamento também aumentam”.

No âmbito da solução simultânea das equações com tratamento implícito das não-linearidades, como já foi mencionado acima, é comum o uso do método de Newton e suas variações, das quais destacam-se os métodos de Newton inexatos, os métodos quase-newtonianos e o método de Newton modificado. Os métodos de Newton inexatos diferenciam-se do método de Newton clássico por utilizarem um segundo processo iterativo para resolver a equação linearizada. Desta forma, a solução do modelo linear resolvido a cada iteração newtoniana é calculada de forma inexata, dando origem ao nome do método (Dembo, Eisenstat e Steihaug, 1982).

Destes métodos, há uma sub-classe que vem ganhando popularidade na solução de sistema de equações diferenciais não-lineares derivadas de problemas de engenharia. São os chamados métodos de Krylov-Newton. Nestes métodos, emprega-se particularmente um *solver* linear baseado em um método de projeção em espaços de Krylov (Saad, 1996), dos quais destacam-se: o GMRES (Saad e Schultz, 1986), o TFQMR (Freund, 1993), o CGS (Sonneveld, 1989), o BICGSTAB (van der Vorst, 1992), entre outros. Vários trabalhos de diferentes áreas têm sido publicados tendo a aplicação de métodos de Newton inexatos como

alvo; dentre os quais podemos citar: Choquet e Erhel (1996), Johnson, McHugh e Knoll (1997), Shadid, Tuminaro e Walker (1997), Pueyo e Zingg (1997,1998) e Kim e Deo (1999). Neste último trabalho, os autores aplicam a um simulador *Black-Oil* um interessante procedimento adaptativo para estimar o grau de convergência necessário para o *solver* iterativo em cada iteração newtoniana. Este método é atribuído a Eisenstat e Walker (1996) e será discutido em detalhes em capítulos posteriores.

Os métodos quase-newtonianos, por sua vez, diferenciam-se do método de Newton tradicional por empregarem uma matriz Jacobiana aproximada. Inúmeros métodos aplicados a soluções de problemas de engenharia possuem as características destes métodos. No âmbito da simulação de reservatórios, Maliska *et al.* (1997) propõem um método quase-newtoniano, aplicado à solução simultânea totalmente implícita utilizando coordenadas generalizadas, em que a matriz Jacobiana é aproximada apenas pelos termos ortogonais. Há ainda os trabalhos de Nghiem (1983) e Barua e Horne (1989), que baseiam-se nos métodos quase-newtonianos da família de Broyden (1965). Estes métodos são uma versão multidimensional do tradicional método 1D da secante. Nghiem introduz uma variação, chamada QNSS – Quasi-Newton Successive Substitution – evitando o cálculo da inversa da Jacobiana, sendo, portanto, ideal para sistemas esparsos. Já Barua e Horne empregam uma outra versão, derivada do trabalho de Mattheis e Strang (1979), na implementação de um simulador paralelo. Na área de escoamento de fluidos, métodos quase-newtonianos são muito utilizados em um procedimento similar ao proposto por Maliska *et al.*, onde a matriz Jacobiana é calculada com uma aproximação de primeira ordem das equações de Navier-Stokes, enquanto as funções resíduo são calculadas com uma aproximação de alta ordem. Segundo Johnson, McHugh e Knoll (1997), este procedimento é conhecido na literatura por *defect correction* e é empregado por diversos autores (Johnson *et al.*, 1994; Gropp e Keyes, 1992; Chin *et al.*, 1992). Além disso, lembrando o que já foi comentado anteriormente, alguns dos métodos adaptativos são, na realidade, métodos quase-newtonianos.

Em 1990, Brown e Saad introduziram métodos híbridos, que combinam a aproximação da matriz Jacobiana com a solução inexata da equação linearizada. São os chamados métodos *matrix-free*. Este nome deriva do fato destes métodos não construírem a matriz Jacobiana diretamente. Ao invés disso, faz-se uma aproximação em diferenças finitas do produto Jacobiana-vetor para ser usado nos *solvers* de Krylov. Johnson, McHugh e Knoll (1997) fazem uma comparação entre o método *matrix-free* usando GMRES preconditionado, o

método de Newton clássico e *defect correction* em escoamentos internos incompressíveis com diferentes números de Reynolds, obtendo melhores resultados com o primeiro método. Pueyo e Zingg (1997, 1997, 1998) aplicam praticamente o mesmo procedimento – *matrix-free* e GMRES preconditionado – em problemas de aerodinâmica, obtendo resultados excelentes, chegando inclusive a superar alguns simuladores que empregam *multigrid* e são referência na área. Trabalhos reportando a aplicação de métodos *matrix-free* em simulação de reservatórios não foram encontrados na literatura disponível.

1.3 Objetivos e Contribuições

Este trabalho tem por objetivo a implementação e análise de diversos métodos de Krylov-Newton aplicados a um simulador *Black-Oil*, baseado na formulação em frações mássicas e na discretização usando volumes finitos em coordenadas curvilíneas generalizadas.

Estes métodos foram divididos em três grupos principais:

- Métodos de Krylov-Newton baseados na proposta de Maliska *et al.* (1997), em que a matriz Jacobiana é aproximada apenas pelos termos ortogonais da discretização.
- Métodos de Krylov-Newton que utilizam a matriz Jacobiana completa, com suas 19 diagonais.
- Métodos *matrix-free*, que utilizam a aproximação da matriz Jacobiana do primeiro método apenas como matriz de condicionamento.

Os métodos pertencentes ao primeiro e ao terceiro grupo também podem ser classificados como métodos quase-newtonianos.

Em cada grupo, os métodos diferem apenas pela escolha do *solver* de Krylov, do preconditionador e da utilização de alguns procedimentos adaptativos para a escolha da tolerância usada no processo iterativo interno (também conhecida como *forcing term*).

Um dos principais aspectos abordados é a influência da não-ortogonalidade da malha no processo iterativo destes métodos, que passa a ter uma importância decisiva devido ao tipo de aproximação adotada nos métodos quase-newtonianos. Deseja-se com isso, dar uma indicação de quanto, maiores ou menores distorções da malha, podem afetar a taxa de convergência destes métodos e qual deles se sai melhor em diferentes graus de não-ortogonalidade.

1.4 Escopo do Trabalho

No capítulo seguinte, obtém-se o conjunto de equações diferenciais parciais que descrevem o modelo *Black-Oil* padrão, a partir da formulação matemática geral do fenômeno de extração de hidrocarbonetos em um reservatório natural. Neste capítulo também é introduzida a formulação em frações mássicas.

No terceiro capítulo, as equações diferenciais do modelo *Black-Oil* são transformadas para o sistema de coordenadas coincidente com a fronteira e, então, são integradas pelo método dos volumes finitos, até a obtenção do sistema de equações discretizadas não-lineares.

O quarto capítulo faz um breve panorama do problema matemático envolvido na solução de sistemas de equações não-lineares e apresenta alguns dos métodos de linearização mais empregados na solução de sistemas provenientes de problemas de engenharia.

No quinto capítulo são apresentados em detalhes os três grupos de métodos de Krylov-Newton mencionados anteriormente, baseados em características particulares da discretização não-ortogonal.

O sexto capítulo é reservado para a apresentação e análise dos resultados numéricos provenientes da aplicação de cada um dos grupos de métodos apresentados no quinto capítulo. Neste capítulo também é feita uma validação da implementação correta dos métodos, através da comparação com um caso descrito na literatura.

Por fim, no sétimo capítulo são apresentadas as conclusões do presente trabalho, bem como as recomendações para trabalhos futuros.

2 Formulação Matemática

Neste capítulo será apresentada a formulação matemática do modelo *Black-Oil* padrão utilizando as frações mássicas de cada componente e a pressão da fase óleo como variáveis primárias.

2.1 Lei de Conservação da Massa

O fenômeno da recuperação de hidrocarbonetos em reservatórios naturais envolve o escoamento multifásico de várias espécies químicas em um meio poroso heterogêneo. Este processo físico leva em conta trocas de calor e massa por convecção e difusão e pode também envolver reações químicas.

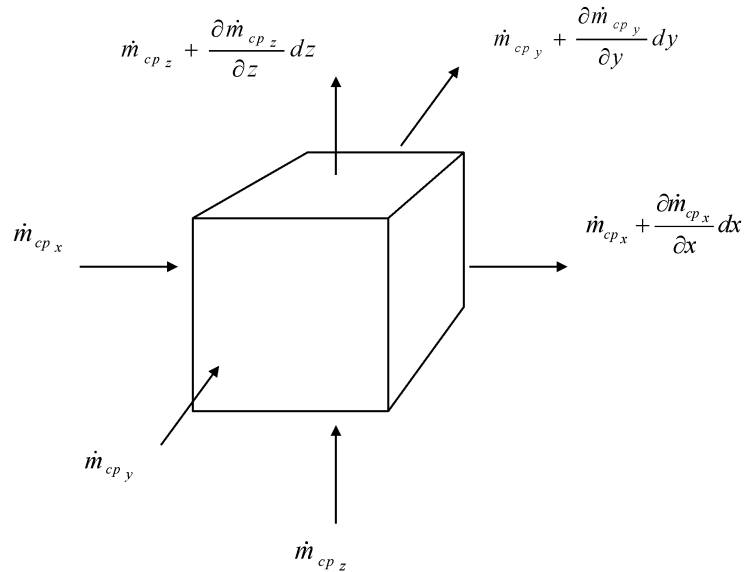


Figura 2.1 – Fluxos de massa através do volume de controle

Considerando o volume de controle mostrado na Fig. 2.1, o fluxo de massa do componente c na fase p pode ser dado por

$$\dot{\mathbf{m}}_{cp} = (X_{cp} \mathbf{r}_p \mathbf{u}_p + \mathbf{D}_{cp} \nabla X_{cp}) A \quad (2.1)$$

onde, X_{cp} é a fração mássica do componente c na fase p;

r_p é a densidade da fase p;

u_p é o vetor velocidade da fase p;

D_{cp} é o tensor coeficiente efetivo de dispersão do componente c na fase p;

A é a área transversal.

O primeiro termo da Eq. (2.1) descreve o transporte convectivo de fluido através do meio poroso e o segundo, o efeito combinado do transporte de massa por dispersão e difusão molecular. O coeficiente efetivo de dispersão é afetado pela porosidade, tortuosidade, distribuição de tamanho de poro, heterogeneidade da rocha e velocidade do fluido. O transporte por convecção está sempre presente no processo de recuperação de petróleo, já o transporte por dispersão/difusão pode ser ignorado na maioria dos casos.

Aplicando o Princípio da Conservação da Massa ao volume de controle da Fig. 2.1, obtém-se

$$\left| \begin{array}{c} \text{taxa líquida de fluxo de massa} \\ \text{do componente c} \\ \text{em todas as fases} \end{array} \right| = \left| \begin{array}{c} \text{taxa de acúmulo de massa} \\ \text{do componente c} \\ \text{em todas as fases} \end{array} \right| + \left| \begin{array}{c} \text{taxa de produção/injeção} \\ \text{de massa do componente c} \\ \text{em todas as fases} \end{array} \right| + \left| \begin{array}{c} \text{taxa de massa do componente c} \\ \text{gerada por reação química} \\ \text{em todas as fases} \end{array} \right| \quad (2.2)$$

A taxa líquida de fluxo de massa do componente c na fase p através das fronteiras do volume de controle é dada por

$$\begin{aligned} -\frac{\partial \dot{m}_{cp_x}}{\partial x} dx - \frac{\partial \dot{m}_{cp_y}}{\partial y} dy - \frac{\partial \dot{m}_{cp_z}}{\partial z} dz = \\ -\frac{\partial}{\partial x} \left(X_{cp} r_p u_{p_x} + D_{cp_x} \frac{\partial X_{cp}}{\partial x} \right) (dydz) dx - \\ -\frac{\partial}{\partial y} \left(X_{cp} r_p u_{p_y} + D_{cp_y} \frac{\partial X_{cp}}{\partial y} \right) (dxdz) dy - \\ -\frac{\partial}{\partial z} \left(X_{cp} r_p u_{p_z} + D_{cp_z} \frac{\partial X_{cp}}{\partial z} \right) (dxdy) dz \end{aligned} \quad (2.3)$$

que, dividindo pelo volume, é o divergente do fluxo de massa:

$$-\nabla \cdot (X_{cp} \mathbf{r}_p \mathbf{u}_p + \mathbf{D}_{cp} \nabla X_{cp}) \quad (2.4)$$

A taxa de acúmulo de massa do componente c na fase p por unidade de volume é expressa por

$$\frac{\partial}{\partial t} (f X_{cp} \mathbf{r}_p S_p) \quad (2.5)$$

onde, f é a porosidade;

S_p é a saturação da fase p .

Os termos fonte/sumidouro de massa do componente c na fase p são dados por

$$X_{cp} \mathbf{r}_p q_p + r_{cp} \quad (2.6)$$

onde, q_p é a taxa volumétrica de produção/injeção da fase p por unidade de volume;

r_{cp} é a taxa de massa do componente c na fase p gerada por reações químicas.

Reagrupando os termos na forma da Eq. (2.2), a equação da conservação da massa fica

$$-\sum_p \nabla \cdot (X_{cp} \mathbf{r}_p \mathbf{u}_p + \mathbf{D}_{cp} \nabla X_{cp}) = \sum_p \frac{\partial}{\partial t} (f X_{cp} \mathbf{r}_p S_p) + \sum_p (X_{cp} \mathbf{r}_p q_p + r_{cp}) \quad (2.7)$$

para $p = 1, 2, \dots, n_p$ e $c = 1, 2, \dots, n_c$.

2.2 Formulação em Frações Mássicas Globais

A Eq. (2.7) é também chamada de formulação em saturações das equações do petróleo, pois o termo de acumulação de massa está escrito em função da variável saturação. Como já foi comentado no capítulo anterior, em problemas envolvendo alguma fase gasosa este tipo de formulação traz complexidades para o modelo computacional que tem de lidar com o desaparecimento de fase. Visando contornar este problema, introduziu-se a formulação em frações mássicas globais. Neste caso, o termo de acumulação fica

$$\frac{\partial}{\partial t} (f \mathbf{r}_m Z_c) \quad (2.8)$$

onde, \mathbf{r}_m é a densidade média da mistura, dada por

$$\mathbf{r}_m = \sum_p \mathbf{r}_p S_p \quad (2.9)$$

e Z_c é a fração mássica global do componente c .

Substituindo na Eq. (2.7), obtém-se

$$\frac{\partial}{\partial t}(\mathbf{f}\mathbf{r}_m Z_c) = -\sum_p \nabla \cdot (X_{cp} \mathbf{r}_p \mathbf{u}_p + \mathbf{D}_{cp} \nabla X_{cp}) + \sum_p (X_{cp} \mathbf{r}_p q_p + r_{cp}) \quad (2.10)$$

para $p = 1, 2, \dots, n_p$ e $c = 1, 2, \dots, n_c$.

2.3 Lei de Darcy

Para descrever matematicamente o escoamento, além da equação da continuidade desenvolvida nas seções anteriores, é necessária uma relação entre a velocidade e o gradiente de pressão em cada fase. Esta relação, que substitui a equação de conservação da quantidade de movimento em escala macroscópica para um meio poroso, foi descoberta por Darcy (1856) para o escoamento monofásico e é chamada de Lei de Darcy. Embora esta lei tenha sido originalmente baseada em observações experimentais, ela também pode ser obtida através de médias volumétricas das equações do movimento.

Para um escoamento multifásico, sua forma é:

$$\mathbf{u}_p = -\mathbf{K} \frac{k_{rp}}{\mathbf{m}_p} (\nabla P_p - \boldsymbol{\xi}_p \nabla D) = -\mathbf{I}_p \nabla \Phi_p \quad (2.11)$$

onde, \mathbf{K} é o tensor permeabilidade absoluta;

k_{rp} é a permeabilidade relativa da fase p ;

\mathbf{m}_p é a viscosidade da fase p ;

P_p é a pressão da fase p ;

$\boldsymbol{\xi}_p$ é o peso específico da fase p ;

D é a profundidade com relação a algum ponto de referência;

$I_p = \mathbf{K} \frac{k_{rp}}{m_p}$ é a mobilidade da fase p;

$\Phi_p = P_p - \epsilon_p D$ é o potencial da fase p.

O tensor permeabilidade usado na Eq. (2.11) é definido pela própria equação e, portanto, deve ser determinado experimentalmente. Na maioria dos problemas encontrados na prática é possível (ou necessário) assumir que \mathbf{K} é um tensor diagonal, dado por

$$\mathbf{K} = \begin{bmatrix} k_x & & \\ & k_y & \\ & & k_z \end{bmatrix} \quad (2.12)$$

Se $k_x = k_y = k_z$, então o meio poroso é chamado de isotrópico, se não, o meio é anisotrópico.

Discussões detalhadas a respeito das limitações inerentes à Lei de Darcy são encontradas nos trabalhos de Hubbert (1956), Scheidegger (1974), Collins (1961) e Whitaker (1966, 1969).

2.4 Modelo *Black-Oil* Padrão

O modelo *Black-Oil* padrão é largamente usado na prática da simulação de reservatórios, sendo especialmente empregado na modelagem de reservatórios contendo óleos pesados ou de baixa volatilidade. Nesta seção, será mostrado como obter as equações deste modelo a partir do modelo genérico da Eq. (2.10), bem como obter as propriedades de cada fase através das relações **Pressão-Volume-Temperatura** (PVT).

O modelo *Black-Oil* padrão possui as seguintes características:

- Sistema com três componentes ($c = w, o, g$)

w , componente água

o , componente hidrocarboneto líquido (óleo)

g , componente hidrocarboneto gasoso (gás)

- Sistema com três fases ($c = w, o, g$)

w , fase água

o , fase óleo

g , fase gás

A água geralmente é a fase molhante, o óleo tem molhabilidade intermediária e o gás é a fase não-molhante.

- Óleo e água são imiscíveis e não trocam massa, ou seja, $X_{wo} = X_{ow} = 0$
- Os componentes água e óleo não vaporizam na fase gás ($X_{wg} = X_{og} = 0$)
- O componente gás pode estar dissolvido na fase óleo, mas não na fase água ($X_{gw} = 0$)
- Escoamento isotérmico
- Os fluidos estão em equilíbrio termodinâmico no reservatório
- Transporte convectivo dominante
- Não há reações químicas ocorrendo no reservatório

Sob estas condições, o comportamento PVT do sistema pode ser representado pelos fatores volume de formação da seguinte maneira:

$$B_o = \frac{(V_{oo} + V_{go})_R}{(V_{oo})_{STD}} = f(P_o) \quad (2.13)$$

$$B_w = \frac{(V_{ww})_R}{(V_{ww})_{STD}} = f(P_w) \quad (2.14)$$

$$B_g = \frac{(V_{gg})_R}{(V_{gg})_{STD}} = f(P_g) \quad (2.15)$$

Nas equações acima, $(V_{cp})_R$ representa o volume ocupado por uma quantidade fixa de massa do componente c (o , w ou g) na fase p (o , w ou g) em condições do reservatório e $(V_{cp})_{STD}$, o volume ocupado pelo mesmo componente nas condições do tanque de armazenamento na superfície (condição *standard*). Os fatores volume de formação são funções da pressão de sua respectiva fase.

Normalmente, esses fatores são fornecidos via uma tabela de pontos experimentais ou são definidos com base na compressibilidade da fase, dada por

$$c_p = -\frac{1}{V_p} \left(\frac{\partial V_p}{\partial P} \right)_T \quad (2.16)$$

Líquidos têm fatores de forma da ordem da unidade, já, para os gases, este fator é muito menor.

A troca de massa entre as fases óleo e gás é descrita pela solubilidade do gás na fase óleo, que é dada pela razão

$$R_S = \left(\frac{V_{go}}{V_{oo}} \right)_{STD} = f(P_o) \quad (2.17)$$

Esta propriedade relaciona a quantidade de gás dissolvido em óleo com a pressão da fase óleo.

A razão de solubilidade gás/óleo está relacionada com a respectiva fração mássica da seguinte maneira

$$R_S = \frac{(V_{go})_{STD}}{(V_{oo})_{STD}} = \frac{\frac{M_{go}}{\mathbf{r}_g^{STD}}}{\frac{V_o}{B_o}} = \frac{\frac{M_{go}}{\mathbf{r}_g^{STD}}}{\frac{M_o}{\mathbf{r}_o B_o}} = \frac{M_{go}}{M_o} \frac{\mathbf{r}_o}{\mathbf{r}_g^{STD}} B_o = X_{go} \frac{\mathbf{r}_o}{\mathbf{r}_g^{STD}} B_o \quad (2.18)$$

e, resolvendo para a fração mássica do gás na fase óleo, tem-se

$$X_{go} = \frac{R_S}{\mathbf{r}_o B_o} \mathbf{r}_g^{STD} \quad (2.19)$$

As frações mássicas dos componentes água, óleo e gás em suas respectivas fases são então facilmente derivadas das hipóteses acima, obtendo-se

$$\begin{aligned} X_{ww} &= 1 \\ X_{oo} &= 1 - X_{go} \\ X_{gg} &= 1 \end{aligned} \quad (2.20)$$

Introduzindo as definições e hipóteses simplificativas do modelo *Black-Oil* e a Lei de Darcy na Eq. (2.10), tem-se as equações de conservação da massa dos componentes água, óleo e gás para o modelo *Black-Oil* padrão:

$$\frac{\partial}{\partial t} [\mathbf{f} \mathbf{r}_m Z_w] = \nabla \cdot [\mathbf{I}_w \nabla \Phi_w] + m_w \quad (2.21)$$

$$\frac{\partial}{\partial t}[\mathbf{f}\mathbf{r}_m Z_o] = \nabla \cdot [X_{oo} \mathbf{I}_o \nabla \Phi_o] + X_{oo} m_o \quad (2.22)$$

$$\frac{\partial}{\partial t}[\mathbf{f}\mathbf{r}_m Z_g] = \nabla \cdot [(1 - X_{oo}) \mathbf{I}_o \nabla \Phi_o + \mathbf{I}_g \nabla \Phi_g] + (1 - X_{oo}) m_o + m_g \quad (2.23)$$

onde, $m_p = \mathbf{r}_p q_p$ é fluxo mássico de produção/injeção da fase p por unidade de volume, sendo positivo para injeção e negativo para a produção, e a mobilidade da fase p foi redefinida para

$$\mathbf{I}_p = \mathbf{K} \frac{\mathbf{r}_p k_{rp}}{\mathbf{m}_p} \quad (2.24)$$

As incógnitas do sistema formado pelas equações (2.21), (2.22) e (2.23) são P_w , P_o , P_g , Z_w , Z_o e Z_g . Logo, necessita-se de mais três equações para o fechamento do problema. Uma delas é a equação de restrição imposta às frações mássicas globais, que permite escrever a fração mássica do gás em função das frações mássicas do óleo e da água como é mostrado na equação abaixo

$$Z_g = 1 - Z_w - Z_o \quad (2.25)$$

As duas equações restantes relacionam as pressões das fases através do efeito de capilaridade como é mostrado abaixo

$$P_{C_{ow}} = P_o - P_w = f(S_w) \quad (2.26)$$

$$P_{C_{og}} = P_g - P_o = f(S_o) \quad (2.27)$$

As pressões capilares $P_{C_{ow}}$ e $P_{C_{og}}$ são funções das saturações das fases molhantes em cada sistema (água/óleo e óleo/gás) e, na maioria dos casos, são determinadas empiricamente. Com as Eqs. (2.25), (2.26) e (2.27) introduzidas no sistema, as variáveis primárias do modelo *Black-Oil* passam a ser a pressão da fase óleo, P_o , e as frações mássicas globais da água, Z_w , e do óleo, Z_o . Além disso, é comum substituir a equação de conservação da massa do componente gás pela equação de conservação da massa total, formada pela soma das Eqs. (2.21), (2.22) e (2.23). Com isso o sistema a ser resolvido torna-se

$$\frac{\partial}{\partial t}[\mathbf{fr}_m Z_w] = \nabla \cdot [\mathbf{I}_w \nabla \Phi_w] + m_w \quad (2.28)$$

$$\frac{\partial}{\partial t}[\mathbf{fr}_m Z_o] = \nabla \cdot [X_{oo} \mathbf{I}_o \nabla \Phi_o] + X_{oo} m_o \quad (2.29)$$

$$\frac{\partial}{\partial t}[\mathbf{fr}_m] = \nabla \cdot [\mathbf{I}_w \nabla \Phi_w + \mathbf{I}_o \nabla \Phi_o + \mathbf{I}_g \nabla \Phi_g] + m_w + m_o + m_g \quad (2.30)$$

$$Z_g = 1 - Z_w - Z_o \quad (2.31)$$

onde,

$$\Phi_w = P_o - P_{C_{ow}} + \xi_w D \quad (2.32)$$

$$\Phi_o = P_o + \xi_o D \quad (2.33)$$

$$\Phi_g = P_o + P_{C_{og}} + \xi_g D \quad (2.34)$$

No restante do texto, a pressão da fase óleo, P_o , será designada, por simplicidade, por apenas P . Além disso, a equação de conservação da massa do componente água, será chamada apenas de equação da água; a da conservação da massa do componente óleo, de equação do óleo; e a equação de conservação da massa total, de equação da pressão. O nome “equação da pressão” advém desta equação ter sido originalmente moldada para o avanço da variável pressão em simuladores que utilizam a técnica IMPES.

2.5 Condições de Contorno e Iniciais

A interação entre o reservatório e sua vizinhança é levada em conta na simulação através das condições especificadas em suas fronteiras. As condições de contorno (ou de fronteira) são necessárias nos poços e nas fronteiras exteriores do reservatório.

Há basicamente três tipos de condições de contorno:

- Fronteira impermeável:

$$\mathbf{u}_p \cdot \mathbf{n} = 0 \quad \text{em } \Gamma \quad (2.35)$$

- Vazão prescrita:

$$\mathbf{u}_p \cdot \mathbf{n} = q_p(\Gamma) \quad (2.36)$$

- Pressão prescrita:

$$P_p(x, y, z, t) = P_p(\Gamma) \quad (2.37)$$

A prescrição da vazão ou pressão nos poços e nas fronteiras entre o reservatório e um aquífero são geralmente feitas através de termos fontes. Em alguns problemas, onde não é possível utilizar as formulações acima para especificar as condições de contorno, deve-se lançar mão de modelos adicionais que não serão aqui abordados.

As distribuições iniciais de pressão e saturações são normalmente conhecidas e dadas como funções das coordenadas espaciais:

$$\begin{aligned} P_p(x, y, z, 0) &= P_p(x, y, z) \\ S_p(x, y, z, 0) &= S_p(x, y, z) \end{aligned} \quad (2.38)$$

Na maioria dos problemas a distribuição inicial da pressão é determinada assumindo o equilíbrio entre os efeitos gravitacionais e os efeitos de capilaridade no reservatório. Isso torna a pressão uma função apenas da coordenada vertical:

$$P_p(x, y, z, 0) = P_p(z) \quad (2.39)$$

Além disso, para a formulação em frações mássicas, é necessário determinar as frações mássicas globais iniciais através dos campos de pressão e saturação iniciais. Isso é feito através das rotinas de “flash” (Cunha, 1996), assumindo o equilíbrio instantâneo no reservatório.

3 Formulação Numérica

Neste capítulo, as equações diferenciais do modelo *Black-Oil* padrão serão transformadas para o sistema de coordenadas curvilíneas e discretizadas através do método dos volumes finitos. Utiliza-se o método UDS para a avaliação das mobilidades nas faces do volume de controle.

3.1 Transformação das Equações Governantes para o Sistema Coordenado Curvilíneo

Visando obter malhas que se adaptam melhor às fronteiras e a outras irregularidades geométricas do reservatório, uma alternativa eficiente e que vem ganhando adeptos entre os analistas numéricos especializados em simulação de reservatório é o uso de coordenadas generalizadas ou coincidentes com a fronteira. Neste método, popularizado com os trabalhos de Thompson *et al.* (1976,1977), é possível encontrar a transformação, mesmo que numérica, entre o sistema coordenado cartesiano original e o sistema de coordenadas generalizadas coincidente com a geometria irregular. Isso permite que seja feito um mapeamento da geometria complexa, escrita no sistema (x, y, z) , em uma geometria regular no sistema curvilíneo $(\mathbf{x}, \mathbf{h}, \mathbf{g})$, conforme é mostrado na Fig. 3.1.

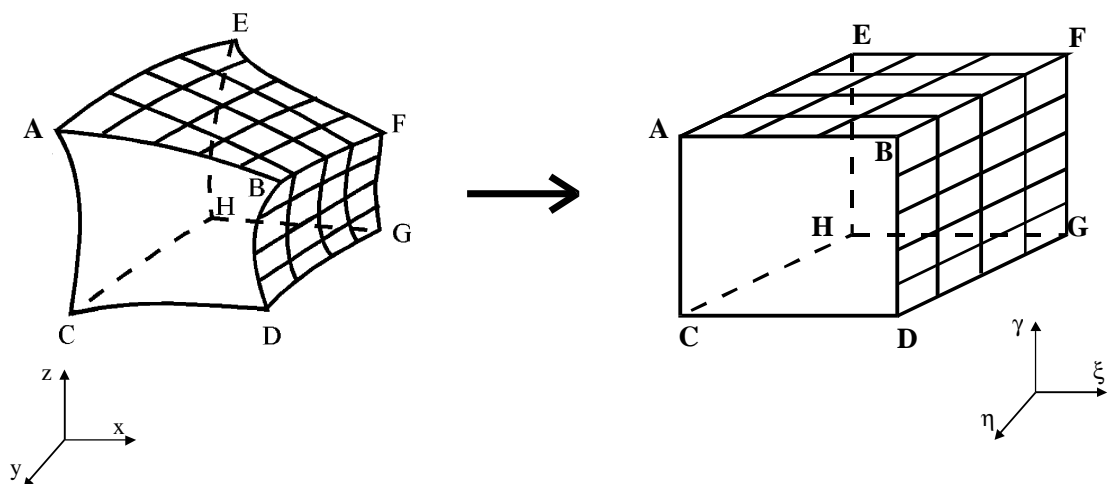


Figura 3.1 – Domínio físico e transformado

Obtida a transformação – cujos detalhes não serão aqui abordados e podem ser encontrados em Maliska (1995) – o sistema de equações governantes deve ser reescrito no sistema curvilíneo. Logo, no espaço transformado, as Eqs. (2.28), (2.29) e (2.30) assumem as seguintes formas:

- Equação da conservação da massa total (Equação da pressão)

$$\begin{aligned}
\frac{1}{J} \frac{\partial}{\partial t} (\mathbf{f} \mathbf{r}_m) - \frac{(m_w + m_o + m_g)}{J} &= \frac{\partial}{\partial \eta} \left[D_{11}^w \frac{\partial \Phi_w}{\partial \eta} + D_{12}^w \frac{\partial \Phi_w}{\partial \eta} + D_{13}^w \frac{\partial \Phi_w}{\partial \eta} \right] + \\
\frac{\partial}{\partial \eta} \left[D_{21}^w \frac{\partial \Phi_w}{\partial \eta} + D_{22}^w \frac{\partial \Phi_w}{\partial \eta} + D_{23}^w \frac{\partial \Phi_w}{\partial \eta} \right] &+ \frac{\partial}{\partial \eta} \left[D_{31}^w \frac{\partial \Phi_w}{\partial \eta} + D_{32}^w \frac{\partial \Phi_w}{\partial \eta} + D_{33}^w \frac{\partial \Phi_w}{\partial \eta} \right] + \\
\frac{\partial}{\partial \eta} \left[D_{11}^o \frac{\partial \Phi_o}{\partial \eta} + D_{12}^o \frac{\partial \Phi_o}{\partial \eta} + D_{13}^o \frac{\partial \Phi_o}{\partial \eta} \right] &+ \frac{\partial}{\partial \eta} \left[D_{21}^o \frac{\partial \Phi_o}{\partial \eta} + D_{22}^o \frac{\partial \Phi_o}{\partial \eta} + D_{23}^o \frac{\partial \Phi_o}{\partial \eta} \right] + \\
\frac{\partial}{\partial \eta} \left[D_{31}^o \frac{\partial \Phi_o}{\partial \eta} + D_{32}^o \frac{\partial \Phi_o}{\partial \eta} + D_{33}^o \frac{\partial \Phi_o}{\partial \eta} \right] &+ \frac{\partial}{\partial \eta} \left[D_{11}^g \frac{\partial \Phi_g}{\partial \eta} + D_{12}^g \frac{\partial \Phi_g}{\partial \eta} + D_{13}^g \frac{\partial \Phi_g}{\partial \eta} \right] + \\
\frac{\partial}{\partial \eta} \left[D_{21}^g \frac{\partial \Phi_g}{\partial \eta} + D_{22}^g \frac{\partial \Phi_g}{\partial \eta} + D_{23}^g \frac{\partial \Phi_g}{\partial \eta} \right] &+ \frac{\partial}{\partial \eta} \left[D_{31}^g \frac{\partial \Phi_g}{\partial \eta} + D_{32}^g \frac{\partial \Phi_g}{\partial \eta} + D_{33}^g \frac{\partial \Phi_g}{\partial \eta} \right].
\end{aligned} \tag{3.1}$$

- Equação da conservação da massa do componente água (Equação da água)

$$\begin{aligned}
\frac{1}{J} \frac{\partial}{\partial t} (\mathbf{f} \mathbf{r}_m Z_w) - \frac{m_w}{J} &= \\
\frac{\partial}{\partial \eta} \left[D_{11}^w \frac{\partial \Phi_w}{\partial \eta} + D_{12}^w \frac{\partial \Phi_w}{\partial \eta} + D_{13}^w \frac{\partial \Phi_w}{\partial \eta} \right] &+ \\
\frac{\partial}{\partial \eta} \left[D_{21}^w \frac{\partial \Phi_w}{\partial \eta} + D_{22}^w \frac{\partial \Phi_w}{\partial \eta} + D_{23}^w \frac{\partial \Phi_w}{\partial \eta} \right] &+ \\
\frac{\partial}{\partial \eta} \left[D_{31}^w \frac{\partial \Phi_w}{\partial \eta} + D_{32}^w \frac{\partial \Phi_w}{\partial \eta} + D_{33}^w \frac{\partial \Phi_w}{\partial \eta} \right]. &
\end{aligned} \tag{3.2}$$

- Equação da conservação da massa do componente óleo (Equação do óleo)

$$\begin{aligned}
\frac{1}{J} \frac{\partial}{\partial t} (\mathbf{f} \mathbf{r}_m Z_o) - \frac{X_{oo} m_o}{J} &= \\
\frac{\partial}{\partial \eta} \left[X_{oo} \left(D_{11}^o \frac{\partial \Phi_o}{\partial \eta} + D_{12}^o \frac{\partial \Phi_o}{\partial \eta} + D_{13}^o \frac{\partial \Phi_o}{\partial \eta} \right) \right] &+ \\
\frac{\partial}{\partial \eta} \left[X_{oo} \left(D_{21}^o \frac{\partial \Phi_o}{\partial \eta} + D_{22}^o \frac{\partial \Phi_o}{\partial \eta} + D_{23}^o \frac{\partial \Phi_o}{\partial \eta} \right) \right] &+ \\
\frac{\partial}{\partial \eta} \left[X_{oo} \left(D_{31}^o \frac{\partial \Phi_o}{\partial \eta} + D_{32}^o \frac{\partial \Phi_o}{\partial \eta} + D_{33}^o \frac{\partial \Phi_o}{\partial \eta} \right) \right]. &
\end{aligned} \tag{3.3}$$

onde,

$$D_{ij}^p = \frac{I_p}{J} \left[\frac{\partial x^i}{\partial x} \frac{\partial x^j}{\partial x} + \frac{\partial x^i}{\partial y} \frac{\partial x^j}{\partial y} + \frac{\partial x^i}{\partial z} \frac{\partial x^j}{\partial z} \right] \quad (3.4)$$

e $x^{i,j}$, com $i, j=1...3$, correspondem a \mathbf{x} , \mathbf{h} e $\underline{\mathbf{g}}$, respectivamente.

Exemplificando, para D_{12}^w , tem-se

$$D_{12}^w = \frac{I_w}{J} \left[\frac{\partial \mathbf{x}}{\partial x} \frac{\partial \mathbf{h}}{\partial x} + \frac{\partial \mathbf{x}}{\partial y} \frac{\partial \mathbf{h}}{\partial y} + \frac{\partial \mathbf{x}}{\partial z} \frac{\partial \mathbf{h}}{\partial z} \right] \quad (3.5)$$

E J é o jacobiano da transformação, dado por

$$J = \begin{vmatrix} \frac{\partial \mathbf{x}}{\partial x} & \frac{\partial \mathbf{x}}{\partial y} & \frac{\partial \mathbf{x}}{\partial z} \\ \frac{\partial \mathbf{h}}{\partial x} & \frac{\partial \mathbf{h}}{\partial y} & \frac{\partial \mathbf{h}}{\partial z} \\ \frac{\partial \underline{\mathbf{g}}}{\partial x} & \frac{\partial \underline{\mathbf{g}}}{\partial y} & \frac{\partial \underline{\mathbf{g}}}{\partial z} \end{vmatrix} \quad (3.6)$$

Neste ponto é interessante notar que, para cada fase, devido as derivadas cruzadas oriundas da não-ortogonalidade da malha, tem-se nove termos possíveis de derivada dos potenciais. Este fato também pode ser evidenciado pelo aparecimento das nove componentes do tensor métrico através dos D_{ij}^p . Fazendo um paralelo com sistemas ortogonais, neste caso ter-se-ia apenas três termos, a saber:

$$D_{11}^p \frac{\partial \Phi_p}{\partial \mathbf{x}}, \quad D_{22}^p \frac{\partial \Phi_p}{\partial \mathbf{h}}, \quad D_{33}^p \frac{\partial \Phi_p}{\partial \underline{\mathbf{g}}} \quad (3.7)$$

3.2 Discretização das Equações Governantes

A transformação das equações para um sistema coordenado generalizado permite que o método dos volumes finitos, baseado na integração das equações de balanço escritas na forma conservativa, seja facilmente aplicado. A partir deste ponto, por simplicidade, apenas a equação discretizada da água será abordada em detalhes, sendo que as demais podem ser tranquilamente obtidas repetindo-se o mesmo procedimento. Sendo assim, efetuando-se a integração da equação da água no tempo e no volume elementar do espaço transformado da Fig. 3.2, tem-se:

$$\begin{aligned}
 & \frac{1}{J} \left[(\mathbf{f} \mathbf{r}_m \mathbf{Z}_w)_P - (\mathbf{f} \mathbf{r}_m \mathbf{Z}_w)_P^o \right] \mathbf{V} - \frac{m_w}{J} \mathbf{V} \mathbf{t} = \\
 & \left[D_{11}^w \frac{\mathcal{F}_w}{\mathcal{F}} + D_{12}^w \frac{\mathcal{F}_w}{\mathcal{F}} + D_{13}^w \frac{\mathcal{F}_w}{\mathcal{F}} \right]_{e^-} \mathbf{t} - \\
 & \left[D_{11}^w \frac{\mathcal{F}_w}{\mathcal{F}} + D_{12}^w \frac{\mathcal{F}_w}{\mathcal{F}} + D_{13}^w \frac{\mathcal{F}_w}{\mathcal{F}} \right]_{w^+} \mathbf{t} + \\
 & \left[D_{21}^w \frac{\mathcal{F}_w}{\mathcal{F}} + D_{22}^w \frac{\mathcal{F}_w}{\mathcal{F}} + D_{23}^w \frac{\mathcal{F}_w}{\mathcal{F}} \right]_{n^-} \mathbf{t} - \\
 & \left[D_{21}^w \frac{\mathcal{F}_w}{\mathcal{F}} + D_{22}^w \frac{\mathcal{F}_w}{\mathcal{F}} + D_{23}^w \frac{\mathcal{F}_w}{\mathcal{F}} \right]_{s^+} \mathbf{t} + \\
 & \left[D_{31}^w \frac{\mathcal{F}_w}{\mathcal{F}} + D_{32}^w \frac{\mathcal{F}_w}{\mathcal{F}} + D_{33}^w \frac{\mathcal{F}_w}{\mathcal{F}} \right]_{t^-} \mathbf{t} - \\
 & \left[D_{31}^w \frac{\mathcal{F}_w}{\mathcal{F}} + D_{32}^w \frac{\mathcal{F}_w}{\mathcal{F}} + D_{33}^w \frac{\mathcal{F}_w}{\mathcal{F}} \right]_{b^+} \mathbf{t}
 \end{aligned} \tag{3.8}$$

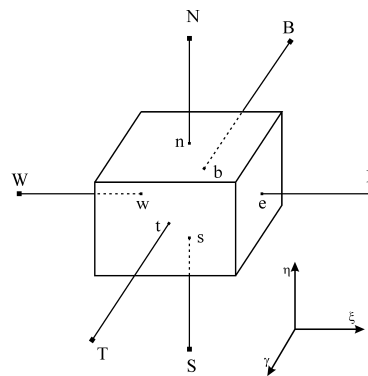


Figura 3.2 – Volume de controle elementar no espaço transformado

A equação acima foi obtida utilizando-se o método totalmente implícito, que avalia as derivadas espaciais no tempo mais atual. É necessário ainda avaliar as derivadas dos potenciais nas faces do volume de controle (e, w, n, s, t, b), bem como as mobilidades presentes nos coeficientes D_{ij}^w .

As derivadas dos potenciais são aproximadas por diferenças centrais; o que, devido ao sistema curvilíneo não-ortogonal, gera um esquema numérico envolvendo dezenove pontos (ver Fig. 3.3): o volume P, seis vizinhos diretos (E, W, N, S, T, B) e doze diagonais (BS, BW, BE, BN, SW, SE, NW, NE, TS, TW, TN, TE). Os termos diagonais aparecem na avaliação das derivadas cruzadas do tipo $(\partial F_w / \partial \mathbf{h})_e$ ou $(\partial F_w / \partial \boldsymbol{\xi})_e$.

Exemplificando, as derivadas do potencial da fase água na face leste são dadas por

$$\frac{\mathcal{I}F_w}{\mathcal{I}^?} \Big|_e = F_e \left[\frac{F_w|_E - F_w|_P}{??} \right] \tag{3.9}$$

$$\frac{\mathcal{I}F_w}{\mathcal{I}^?} \Big|_e = F_e \frac{(F_{\gamma_{SE}}^E F_w|_{SE} + F_{\gamma_E}^E F_w|_E + F_{\gamma_{NE}}^E F_w|_{NE} + F_{\gamma_S}^P F_w|_S + F_{\gamma_P}^P F_w|_P + F_{\gamma_N}^P F_w|_N)}{??} \tag{3.10}$$

$$\frac{\mathcal{I}F_w}{\mathcal{I}^?} \Big|_e = F_e \frac{(F_{\gamma_{BE}}^E F_w|_{BE} + F_{\gamma_E}^E F_w|_E + F_{\gamma_{TE}}^E F_w|_{TE} + F_{\gamma_B}^P F_w|_B + F_{\gamma_P}^P F_w|_P + F_{\gamma_T}^P F_w|_T)}{??} \tag{3.11}$$

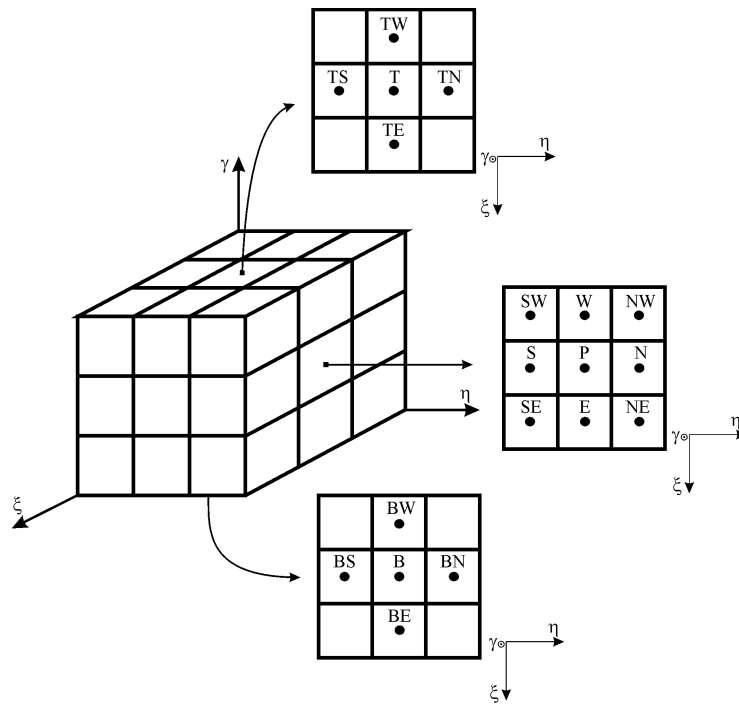


Figura 3.3 – Volumes presentes no esquema numérico

Os coeficientes F são chamados de coeficientes de falha e foram introduzidos por Cunha (1996) com o objetivo de criar uma única equação para todos os volumes da malha; seja este interno, de fronteira ou vizinho a uma falha geológica. Na verdade, estes coeficientes são apenas artifícios computacionais usados para simplificar a codificação. Dadas as condições de contorno e a posição das falhas (se houverem), o programa gera estes coeficientes automaticamente, que depois serão utilizados na montagem das equações. Detalhes do cálculo destes coeficientes são encontrados em Czesnat (1999) e as aproximações das demais derivadas do potencial da fase água encontram-se no Apêndice A.

Como já foi visto anteriormente, as mobilidades das fases são funções das saturações. Estas, por sua vez, são armazenadas nos centros dos volumes de controle. Logo, as

mobilidades também estão definidas apenas nos centros e necessitam de um método de interpolação para determinar seus valores nas faces do volume, onde são requeridas pelo método numérico. Neste trabalho, optou-se pelo método UDS – *Upstream Differencing Scheme* – devido as suas propriedades de estabilidade e fácil implementação, além de ser largamente empregado nos simuladores comerciais. No entanto, como desvantagem, este método apresenta uma considerável difusão numérica. Exemplos e resultados da aplicação de outras funções interpolação mais modernas e precisas são encontrados no trabalho de Czesnat (1999).

No esquema UDS, a mobilidade na face assume o valor da mobilidade calculada no centro do volume de controle que está atrás da face em relação ao sentido do escoamento. Sendo assim, supondo que o escoamento dirige-se do volume W para o volume P, a mobilidade na interface w será avaliada em W e a da interface e será avaliada em P. É interessante salientar ainda que, para o escoamento trifásico, deve-se aplicar o esquema para cada fase, pois estas possuem movimentos independentes. Além disso, em problemas tridimensionais o método UDS utiliza uma aproximação unidimensional em cada um dos eixos coordenados.

Desta forma, para cada direção, tem-se as seguintes aproximações:

- Se $(\partial F_p / \partial \mathbf{n})_p < 0$, então

$$\begin{aligned} I_p|_e &= I_p|_P \\ I_p|_w &= I_p|_W \end{aligned} \quad (3.12)$$

caso contrário

$$\begin{aligned} I_p|_e &= I_p|_E \\ I_p|_w &= I_p|_P \end{aligned} \quad (3.13)$$

- Se $(\partial F_p / \partial \mathbf{n})_p < 0$, então

$$\begin{aligned} I_p|_n &= I_p|_P \\ I_p|_s &= I_p|_S \end{aligned} \quad (3.14)$$

caso contrário

$$\begin{aligned} \mathbf{I}_p|_n &= \mathbf{I}_p|_N \\ \mathbf{I}_p|_s &= \mathbf{I}_p|_P \end{aligned} \quad (3.15)$$

- Se $(\partial F_p / \partial \mathbf{n})_p < 0$, então

$$\begin{aligned} \mathbf{I}_p|_t &= \mathbf{I}_p|_P \\ \mathbf{I}_p|_b &= \mathbf{I}_p|_B \end{aligned} \quad (3.16)$$

caso contrário

$$\begin{aligned} \mathbf{I}_p|_t &= \mathbf{I}_p|_T \\ \mathbf{I}_p|_b &= \mathbf{I}_p|_P \end{aligned} \quad (3.17)$$

Na verdade, para a equação do óleo deve-se aproximar $X_{oo}\mathbf{I}_o$ ao invés de apenas \mathbf{I}_o .

3.3 Sistema de Equações Não-Lineares

Uma vez discretizadas as equações, é possível agrupá-las gerando um sistema de equações não-lineares que pode ser escrito na forma

$$\mathbf{F}(\mathbf{x}) = [f_1(\mathbf{x}) \quad f_2(\mathbf{x}) \quad \dots \quad f_{3n}(\mathbf{x})]^T = \mathbf{0} \quad (3.18)$$

onde n é o número de volumes na malha computacional. O vetor incógnita \mathbf{x} é composto de todas as variáveis dependentes ou incógnitas do problema:

$$\mathbf{x} = [P_1 \quad Z_{w1} \quad Z_{o1} \quad P_2 \quad Z_{w2} \quad Z_{o2} \quad \dots \quad P_n \quad Z_{wn} \quad Z_{on}]^T \quad (3.19)$$

As não-linearidades presentes na Eq. (3.18) podem ser divididas em dois grupos:

- Fracas. Todas as variáveis que são funções apenas da pressão de uma fase podem ser consideradas não-linearidades fracas. Isto inclui os fatores volume de formação, a razão de solubilidade e as viscosidades.
- Fortes. Estão presentes nas variáveis que dependem da saturação, ou seja, as permeabilidades relativas e as pressões capilares.

Devido ao tratamento totalmente implícito empregado na obtenção das equações, todas as não-linearidades são avaliadas no mesmo nível de tempo.

No próximo capítulo, serão abordados detalhes relativos à solução de equações do tipo da Eq. (3.18).

4 Métodos de Solução de Sistemas de Equações Não-Lineares

Neste capítulo, inicialmente será apresentado um breve panorama das várias questões que podem ser levantadas a respeito da obtenção da solução computacional de um sistema de equações não-lineares, para depois introduzir os métodos que serão, no capítulo seguinte, empregados na solução específica da Eq. (3.18).

4.1 Visão Geral do Problema

De maneira geral, um sistema de equações não-lineares pode ser descrito como a aplicação de uma transformação não-linear, $F : R^n \mapsto R^n$, do espaço linear real n-dimensional nele mesmo, de tal forma que

$$F(x) = [f_1(x), \dots, f_n(x)]^T = 0, \quad \forall x \in R^n \quad (4.1)$$

A obtenção das soluções da equação acima envolve, ao menos, duas questões de certa forma entrelaçadas: a análise das propriedades de resolubilidade e o desenvolvimento e estudo de métodos numéricos adequados. É óbvio que os resultados de ambas as questões acima dependerão sobretudo das propriedades da função F .

Com relação às propriedades de resolubilidade do sistema, há no mínimo outras três questões que devem ser consideradas

- a) Existe solução em um subespaço específico do domínio de F ?
- b) Quantas soluções há neste subespaço?
- c) Qual é a estabilidade das soluções mediante uma pequena variação nas condições do problema?

Mesmo para casos simples, unidimensionais, exemplos mostram que a Eq. (4.1) pode ter finitas, infinitas ou nenhuma solução, o que dá uma idéia da importância de se ter respostas às questões levantadas acima. No entanto, para a maioria dos casos de interesse da engenharia, obter essas respostas ainda é uma tarefa extremamente difícil e muitas vezes até impossível.

Na maioria dos casos, a confiança em se encontrar a solução para um conjunto de equações não-lineares através da aplicação de um determinado método tem base empírica, carecendo de algum fundamento teórico.

Uma vez sabido que (4.1) tem solução (ou soluções), deve-se concentrar atenção nos métodos para aproximá-las. Estes podem ser classificados, quanto ao seu objetivo, em:

- a) Métodos de localização, que constroem subespaços que contém as soluções
- b) Métodos que aproximam uma solução
- c) Métodos para achar todas as soluções

Este trabalho se concentra num subgrupo de b). Métodos da classe c) são disponíveis apenas para problemas de minimização e os da classe a) são baseados no uso de matemática intervalar.

Normalmente, métodos diretos não são aplicáveis na solução de (4.1), restando apenas a utilização de métodos iterativos. Segundo Rheinboldt (1998), estes métodos podem ser, de forma geral, classificados em:

- a) Métodos de linearização
- b) Redução a equações não-lineares mais simples
- c) Combinação de métodos
- d) Métodos de Continuação
- e) Métodos de Minimização

Para o problema específico abordado neste trabalho, os métodos das classes a) e c) são mais apropriados e alguns serão vistos em detalhe mais adiante.

Ainda segundo Rheinboldt, ao se analisar um determinado método iterativo, ao menos quatro questões de cunho geral devem ser observadas. São elas:

- a) Sob quais condições o processo iterativo está bem definido?
- b) Quando as iterações convergem?
- c) No caso de convergência, o limite é realmente uma solução de (4.1)?
- d) Quão eficiente é o processo?

Rheinboldt diz que, na maioria dos casos, apenas respostas parciais são possíveis e, em particular, o conjunto exato de condições iniciais, para as quais um processo iterativo está bem definido ou é convergente, é muito raro de ser computado. Muitos exemplos mostram, que mesmo existindo um limite para o processo iterativo, este pode não resolver o sistema.

Com relação a questão d), sobre a economia ou complexidade computacional do processo iterativo, pode-se ainda desenvolver outras questões relacionadas:

- a) Qual o “custo” de uma única iteração?
- b) Quão “rápido” o processo converge?
- c) Qual a “sensibilidade” do processo a mudanças na transformação não-linear, nas condições iniciais ou na dimensão do espaço?

Qualquer uma destas questões requer antes de tudo uma definição mais precisa para os termos “custo”, “rápido” e “sensibilidade”, o que dependerá sobretudo da definição minuciosa dos parâmetros usados pelos processos iterativos. Geralmente, questões sobre a economia dos métodos são tratadas apenas de forma relativa, através da comparação direta entre os métodos; o que nem sempre é tão simples, pois cada método exige uma série de parâmetros com efeitos distintos na taxa de convergência. A questão c) é a que possui menor chance de ser respondida de uma forma abrangente, embora, para certos tipos de problema, vários algoritmos certamente apresentarão alta sensibilidade a mudanças no operador ou nas condições iniciais. Isso normalmente acontece devido a presença de singularidades; o que, por si só, constitui-se um importante campo de estudos da matemática.

4.2 Métodos de Newton

Nesta seção serão introduzidos vários métodos iterativos desenvolvidos a partir da linearização da Eq. (4.1), entre eles, o método de Newton tradicional e várias de suas derivações. Alguns métodos, de acordo com a classificação de Rheinboldt apresentada anteriormente, recaem no caso da combinação de métodos iterativos, onde, a cada passo de um primeiro processo iterativo, um segundo processo é aplicado.

4.2.1 O Conceito de Linearização

Seja $x^k \in R^n$ o vetor solução na iteração k do processo iterativo para resolver $F(x)=0$.

A idéia do método de linearização é construir uma aproximação $L_k : R^n \mapsto R^n$ tal que

$$L_k(x) = A_k(x - x^k) + F(x^k), \quad A_k \in L(R^n) \quad (4.2)$$

que coincida com F em x^k , e usar o resultado de $L_k(x)=0$ como o valor mais recente de x .

Considerando apenas o caso onde todas as matrizes $A_k, k \geq 0$ em (4.2) são inversíveis, o método de linearização resulta em

$$x^{k+1} = x^k - A_k^{-1}F(x^k), \quad k = 0, 1, \dots \quad (4.3)$$

que, em termos de algoritmo, traduz-se para a seguinte sequência

1. **entra:** $\{k, x^k, M_k\}$
 2. avaliar $F(x^k)$;
 3. construir a matriz A_k ;
 4. resolver $A_k y = F(x^k)$ para y ;
 5. **se** solução falha **então retorna:** falha;
 6. $x^{k+1} = x^k - y$;
 7. **retorna:** $\{x^{k+1}, M_{k+1}\}$
- (4.4)

onde, M representa um bloco de memória contendo todo e qualquer dado necessário ao processo que não seja k e x .

O algoritmo (4.4) representa uma única iteração do método de linearização e, como nos processos iterativos empregados na solução de sistemas lineares, pode ser necessário introduzir um fator de relaxação (*damping factor*), ou seja, aumentar ou diminuir o passo entre duas iterações sucessivas, y , por um fator $w_k > 0$. Isto significa alterar (4.3) para

$$x^{k+1} = x^k - w_k A_k^{-1} F(x^k), \quad k = 0, 1, \dots \quad (4.5)$$

ou, simplesmente, alterar a sexta linha do algoritmo (4.4) para $x^{k+1} = x^k - w_k y$.

4.2.2 Forma Geral dos Métodos de Newton

A definição do processo de linearização dada na seção anterior não inclui nenhuma informação a respeito do mecanismo de construção das matrizes A_k em (4.3). Uma forma natural e simples de gerar estas matrizes é avaliá-las, em cada iteração k , através de x^k , ou seja, fazendo $A_k = A(x^k)$. Estes métodos têm a forma

$$x^{k+1} = x^k - A^{-1}(x^k)F(x^k), \quad k = 0, 1, \dots \quad (4.6)$$

também conhecida como forma de Newton e, cujo membro mais famoso, é o Método de Newton clássico

$$x^{k+1} = x^k - J^{-1}(x^k)F(x^k), \quad k = 0, 1, \dots \quad (4.7)$$

onde a aproximação L_k de (4.2) é obtida truncando-se a expansão em série de Taylor de F , ao redor de x^k , depois do termo linear, de maneira que a matriz $J(x^k)$ é a matriz Jacobiana

$$J(x) = \begin{bmatrix} \frac{\partial}{\partial x_1} f_1(x) & \cdots & \frac{\partial}{\partial x_n} f_1(x) \\ \cdots & \cdots & \cdots \\ \frac{\partial}{\partial x_1} f_n(x) & \cdots & \frac{\partial}{\partial x_n} f_n(x) \end{bmatrix} \quad (4.8)$$

Uma forma mais simples do método de Newton, conhecida como método de Newton modificado ou paralelo, pode ser obtida forçando-se que todas as matrizes A_k sejam idênticas a matriz Jacobiana obtida na primeira iteração, $J(x^o)$. Em outras palavras, este método possui a forma

$$x^{k+1} = x^k - J^{-1}(x^o)F(x^k), \quad k = 0, 1, \dots \quad (4.9)$$

Apesar do menor esforço computacional despendido no cálculo das Jacobianas, o método de Newton modificado possui uma taxa de convergência muito pobre.

4.2.3 Métodos de Newton Inexatos

Quando o processo de linearização é aplicado a um problema com um número muito grande de equações, normalmente provenientes da discretização de um conjunto de equações

diferenciais parciais, torna-se necessário lançar mão de um segundo processo iterativo para resolver a equação linear resultante em cada iteração de maneira eficiente. Um problema que surge desta combinação de processos iterativos é escolher como controlar a convergência do segundo método. Pode-se especificar a priori algum critério ou condição fixa ou criar alguma estratégia adaptativa baseada na performance do algoritmo. De toda forma, o mérito destes métodos está na relaxação da equação de Newton clássica para uma condição inexata do tipo

$$\|F(x^k) + J(x^k)\hat{s}^k\| \leq h_k \|F(x^k)\| \quad (4.10)$$

onde $h_k \in [0,1)$ e $\|\cdot\|$ é uma norma qualquer. Daí o fato destes métodos serem chamados de métodos de Newton inexatos.

A idéia é escolher algum h_k e então aplicar um *solver* iterativo na solução da equação linear

$$J(x^k)\hat{s}^k = -F(x^k) \quad (4.11)$$

até que a solução aproximada \hat{s}^k satisfaça (4.10). Neste contexto, h_k é frequentemente chamado de *forcing term* (Eisenstat e Walker, 1996), já que seu papel é forçar que o resíduo da Eq. (4.11) seja pequeno o suficiente.

É importante notar que $F(x^k) + J(x^k)\hat{s}^k$ é tanto o resíduo da Eq. (4.11), quanto a aproximação linear para $F(x^{k+1})$ (com $x^{k+1} = x^k + \hat{s}^k$) através da expansão em série de Taylor, truncada no termo linear, de F ao redor de x^k . Logo, reduzindo o resíduo linear para satisfazer (4.11), se reduz também o resíduo não-linear, desde que haja concordância suficiente entre F e sua aproximação linear local.

Dembo, Eisenstat e Steihaug (1982) mostram que, próximo à solução do sistema de equações não-lineares, a convergência local dos métodos de Newton inexatos é controlada pelos *forcing terms*. Mais precisamente, próximo à solução não-linear, fazendo-se $h_k \rightarrow 0$, a taxa de convergência local é q-superlinear (Rheinboldt, 1998). Como se verá no capítulo de resultados, isso dependerá essencialmente do quão próximo a matriz Jacobiana é aproximada.

Além da garantia de uma boa taxa de convergência local próxima à solução, estes métodos requerem ainda procedimentos que aumentem a probabilidade de convergência quando x^o não está perto da solução. Estes procedimentos são chamados de globalização

(Eisenstat e Walker, 1994), dentre os quais, o mais difundido e simples de implementar, é o procedimento de *backtracking*, também conhecido como *linesearch* ou *damping*. Neste procedimento, o passo do método de Newton, \hat{s}^k , é encurtado até que seja considerado aceitável, satisfazendo uma determinada condição. O algoritmo de *backtracking* usado neste trabalho foi proposto por Eisenstat e Walker (1996), sendo empregado em diversos outros simuladores (Shadid, Tuminaro e Walker, 1997), inclusive em um pacote para solução de quaisquer sistemas de equações não-lineares, o NITSOL (Pernice e Walker, 1996). O algoritmo, aplicado a uma iteração do método de Newton inexato, é mostrado abaixo.

1. **entra:** $\{k, x^k, M_k\}$
2. determinar \hat{s}^k , com $\mathbf{h}_k \in (0, \mathbf{h}_{\max})$,
tal que $\|F(x^k) + J(x^k)\hat{s}^k\| \leq \mathbf{h}_k \|F(x^k)\|$;
3. $\mathbf{h} := \mathbf{h}_k$; $s := \hat{s}^k$; $j := 0$;
4. **enquanto** $\|F(x^k + s)\| > [1 - \mathbf{t}(1 - \mathbf{h})] \cdot \|F(x^k)\|$ e
 $j \leq j_{\max}$ **faça:**
5. escolher $\mathbf{q} \in [\mathbf{q}_{\min}, \mathbf{q}_{\max}]$;
6. $\mathbf{h} := (1 - \mathbf{q}) + \mathbf{q}\mathbf{h}$;
7. $s := \mathbf{q}s$;
8. **se** $j > j_{\max}$ **então retorna:** falha;
9. $x^{k+1} := x^k + s$;
10. **retorna:** $\{x^{k+1}, M_{k+1}\}$

Note que, para um dado \mathbf{h}_k inicial, existirá um \hat{s}^k inicial se a Eq. (4.11) for consistente, particularmente, se $J(x^k)$ for inversível. Se este \hat{s}^k inicial for satisfatório e o j_{\max} escolhido não for demasiadamente pequeno, espera-se que o algoritmo não falhe. Além disso, pode-se facilmente notar que a condição ditada pela Eq. (4.10) é obedecida por \mathbf{h} e s definidos no interior do laço. Logo, o passo final, satisfaz tanto (4.10), quanto

$$\|F(x^k + s)\| \leq [1 - \mathbf{t}(1 - \mathbf{h})] \cdot \|F(x^k)\| \quad (4.13)$$

Tomando-se a redução final de $\|F\|$ promovida pelo algoritmo (4.12)

$$\text{red}_{\text{backtracking}} = \|F(x^k)\| - \|F(x^k + s)\| \quad (4.14)$$

e a redução prevista pela aproximação linear

$$\text{red}_{\text{linearização}} = \|F(x^k)\| - \|F(x^k) + J(x^k)\hat{s}_k\| \quad (4.15)$$

e aplicando as condições (4.10) e (4.13), vem que

$$\begin{aligned} \text{red}_{\text{backtracking}} &\geq t(1+\mathbf{h})\|F(x^k)\| \\ \text{red}_{\text{linearização}} &\geq (1+\mathbf{h})\|F(x^k)\| \end{aligned} \quad (4.16)$$

Ou seja, a redução em $\|F\|$ promovida pelo procedimento de *backtracking* é ao menos uma fração t da redução prevista pelo modelo linear de $\|F\|$. Isto agrega a propriedade de convergência global combinada a possivelmente rápida convergência local inerente aos métodos de Newton inexatos.

Um resultado importante apontado por Eisenstat e Walker (1994) é decorrente do seguinte teorema:

Teorema 4.1. *Assumindo que F é continuamente diferenciável, se a sequência $\{x^k\}$, produzida pelo algoritmo (4.12), tem um limite x^* em que $J(x^*)$ é inversível, então $F(x^*)=0$ e $x^k \rightarrow x^*$. Além disso, \hat{s}_k e \mathbf{h}_k iniciais são aceitos sem modificação pelo laço para um k suficientemente grande.*

Logo, se a sequência de iterações tiver um limite em que $J(x^*)$ é inversível, então este deve ser a solução do sistema de equações não-lineares e as iterações devem convergir para ele. Além disso, próximo à solução, o algoritmo não necessitará de *backtracking*, pois a convergência é governada pelo \mathbf{h}_k inicial. Ou seja, perto da solução a convergência é governada pela análise local apresentada por Dembo, Eisenstat e Steihaug (1982) e, portanto, pode ser acelerada escolhendo-se \mathbf{h}_k tão pequeno quanto necessário. A convergência próxima à solução do sistema (4.1), também é chamada de convergência assintótica.

De acordo com o que foi visto acima, os termos \mathbf{h}_k (ou *forcing terms*) não apenas determinam a velocidade da convergência assintótica, mas também afetam a eficiência e a robustez do algoritmo longe da solução. Na verdade, longe da solução, escolhendo-se \mathbf{h}_k muito pequeno e resolvendo a Eq. (4.11) até este nível de tolerância pode resultar num passo \hat{s}_k tão grande que $F(x^k + \hat{s}_k)$ difere em muito de sua aproximação linear $F(x^k) + J(x^k)\hat{s}_k$. Este problema é conhecido como *oversolving* (Eisenstat e Walker, 1996). Com isso, pouco ou

nenhum progresso na redução de $\|F\|$ é alcançado, o que, conseqüentemente, levará à execução do procedimento de *backtracking*. Mesmo que uma redução razoável de $\|F\|$ seja alcançada, resolver (4.11) com tamanha acurácia no início do processo iterativo pode ser relativamente oneroso. Uma solução mais pobre, além de ser obtida mais rapidamente, pode gerar maior redução em $\|F\|$, além de usar um menor número de iterações do procedimento de *backtracking*.

Neste trabalho, dois métodos adaptativos para a escolha dos *forcing terms* foram implementados e testados com o objetivo de minimizar o *oversolving* mantendo uma rápida convergência assintótica. Ambos foram propostos por Eisenstat e Walker (1996) e são assim definidos:

- Método 1: Selecionar algum $\mathbf{h}_0 \in (0,1]$ e escolher

$$\mathbf{h}_k = \frac{\|F(x^k)\| - \|F(x^{k-1}) + J(x^{k-1})s_{k-1}\|}{\|F(x^{k-1})\|}, \quad k = 1, 2, \dots \quad (4.17)$$

- Método 2: Dado um $\underline{\xi} \in [0,1]$ e $\mathbf{a} \in (1,2]$, selecionar algum $\mathbf{h}_0 \in (0,1]$ e escolher

$$\mathbf{h}_k = \underline{\xi} \left(\frac{\|F(x^k)\|}{\|F(x^{k-1})\|} \right)^{\mathbf{a}}, \quad k = 1, 2, \dots \quad (4.18)$$

Além disso, para assegurar que $\mathbf{h}_k \leq \mathbf{h}_{\max}$ de acordo com o algoritmo (4.12), após as escolhas (4.17) e (4.18) aplica-se a proteção

$$\mathbf{h}_k \leftarrow \min\{\mathbf{h}_k, \mathbf{h}_{\max}\} \quad (4.19)$$

Pernice e Walker (1996) mostram que, combinando resultados da análise da convergência local ao **Teorema 4.1**, obtém-se, para o método 1, convergência q-superlinear e, para o método 2 com $\underline{\xi} < 1$, convergência da ordem q- \mathbf{a} e, com $\underline{\xi} = 1$, da ordem q- p , onde $p \in [1, \mathbf{a})$.

Ainda de acordo com os trabalhos de Eisenstat e Walker (1996) e Pernice e Walker (1996), implementou-se as proteções abaixo, que são aplicadas respectivamente após (4.17) e

(4.18) e antes de (4.19).

- Proteção para o método 1:

$$\mathbf{h}_k \leftarrow \max\{\mathbf{h}_k, \mathbf{h}_{k-1}^{(1+\sqrt{5})/2}\}, \quad \text{se } \mathbf{h}_{k-1}^{(1+\sqrt{5})/2} > 0.1 \quad (4.20)$$

- Proteção para o método 2:

$$\mathbf{h}_k \leftarrow \max\{\mathbf{h}_k, \mathbf{g}_{k-1}^a\}, \quad \text{se } \mathbf{g}_{k-1}^a > 0.1 \quad (4.21)$$

Estas proteções têm o propósito de prevenir que os \mathbf{h}_k iniciais tornem-se muito pequenos longe da solução. Note que, próximo à solução, $\mathbf{h}_k \rightarrow 0$ por ambos os métodos e, portanto, (4.20) e (4.21) ficam inativas não afetando a convergência assintótica.

4.2.4 Métodos de Krylov-Newton

Os métodos de Newton inexatos podem, em princípio, empregar qualquer processo iterativo na solução da equação linear (4.11), todavia há uma classe de métodos iterativos que atualmente vêm recebendo maior atenção. São métodos baseados em técnicas de projeção em espaços de Krylov, que são subespaços cujos vetores de base têm a forma $p(A)v$, onde p é um polinômio (A é a matriz do sistema linear a ser resolvido). Métodos que se enquadram nesta categoria são o GMRES – *Generalized Minimum Residual* – de Saad e Schultz (1986), o TFQMR – *Transpose-Free Quasi-Minimal Residual* – de Freund (1993), o BICGSTAB – *Biconjugate Gradient Stabilized* – de van der Vorst (1992), entre outros.

Um método de projeção qualquer para resolver o sistema linear

$$Ax = b \quad (4.22)$$

é um método que procura por uma solução x_m no subespaço $x_0 + K_m$ de dimensão m através da imposição da condição de Petrov-Galerkin

$$b - Ax_m \perp \Gamma_m \quad (4.23)$$

onde Γ_m é outro subespaço de dimensão m e x_0 é uma estimativa inicial arbitrária. Um método de Krylov é um método cujo subespaço K_m é o subespaço de Krylov

$$K_m(A, r_0) = \text{span}\{r_0, Ar_0, A^2r_0, \dots, A^{m-1}r_0\} \quad (4.24)$$

onde $r_0 = b - Ax_0$. A diferença entre os vários métodos de Krylov está na escolha do subespaço Γ_m e na maneira como o sistema (4.22) é preconditionado.

Embora todas as variantes do método de Krylov estejam baseadas no mesmo tipo de aproximação polinomial da solução, a escolha do subespaço Γ_m , ou seja, a restrição imposta a esta aproximação, terá um efeito decisivo no processo iterativo. Dois tipos ou classes de Γ_m são encontradas na maioria dos métodos. A primeira é tomar simplesmente $\Gamma_m = K_m$ ou sua variante de mínimo resíduo $\Gamma_m = AK_m$. A segunda é criada fazendo Γ_m ser um subespaço de Krylov associado a A^T , $\Gamma_m = K_m(A^T, r_0)$. É importante salientar que um único método de projeção pode ter uma série de implementações diferentes, dando origem a diferentes algoritmos que são matematicamente equivalentes, porém com características numéricas distintas.

Embora a convergência dos métodos de Krylov seja bem fundamentada teoricamente (Saad, 1996), na prática todos sofrem com uma baixa taxa de convergência na solução de sistemas oriundos de problemas de engenharia. O preconditionamento é o mecanismo chave que permite o sucesso destes métodos em aplicações deste tipo, aumentando tanto a eficiência, quanto a robustez dos processos iterativos. O preconditionamento é simplesmente um meio de transformar o sistema linear original, Eq. (4.22), em um outro sistema que possui a mesma solução, porém é mais fácil de resolver com um determinado *solver*. Para se ter uma idéia do poder dos preconditionadores, normalmente a confiabilidade de um determinado *solver* está muito mais ligada ao bom preconditionamento do sistema do que ao processo iterativo de Krylov usado. Geralmente o preconditionador é uma matriz M que aproxima de alguma forma a matriz original A , com a importante exigência de que a solução de sistemas do tipo $Mv = w$ seja simples e barata. Isso porque os algoritmos que empregam preconditionamento têm que resolver sistemas deste tipo a cada iteração. Para o caso geral, o preconditionamento é aplicado ao sistema (4.22) na forma

$$\begin{aligned} M_E^{-1} A M_D^{-1} u &= M_E^{-1} b \\ x &= M_D^{-1} u \end{aligned} \quad (4.25)$$

onde M_E e M_D são, respectivamente, as matrizes de preconditionamento à esquerda e à

direita. Se $M_E = I$, tem-se o condicionamento à direita e o resíduo fica inalterado, todavia, se $M_D = I$, tem-se o condicionamento à esquerda e resíduo altera-se para

$$r_E = M_E^{-1}b - M_E^{-1}Ax = M_E^{-1}r \quad (4.26)$$

Este resíduo, que também aplica-se ao caso geral da Eq. (4.25), é chamado de resíduo condicionado. Neste caso, geralmente controla-se a convergência do processo iterativo através da norma do resíduo condicionado, ao invés da norma do resíduo propriamente dito, devido o cálculo deste último necessitar da multiplicação pela matriz M_E , encarecendo o algoritmo.

Há vários bons condicionadores disponíveis, cada um com propriedades e aplicações distintas. Uma das formas mais simples e eficiente de montar um condicionador é através da fatoração incompleta da matriz A . Isto pode ser feito decompondo-se A na forma $A = LU - R$, onde L e U têm a mesma estrutura de não-zeros das partes inferior e superior, respectivamente, de A ; R é a matriz resíduo ou erro da fatoração. Este tipo de fatoração é conhecida como ILU(0) ou fatoração incompleta de nível zero e tem como vantagens ser de construção simples e rápida. Por outro lado, pode ser uma aproximação pobre, fazendo com que o método de Krylov associado necessite de várias iterações para convergir. Visando evitar este problema, várias alternativas de fatoração LU incompleta foram desenvolvidas através da introdução de mais elementos não-nulos – ou *fill-in's* – em L e U . Em geral, fatorações mais precisas requerem um menor número de iterações do *solver*, porém a montagem do condicionador tem um custo computacional mais elevado. Neste trabalho utilizou-se o algoritmo denominado BILU(l), que é uma fatoração adaptada a matrizes de blocos (ver próximo capítulo) baseada no conceito de nível (Saad, 1996). Para dar uma idéia simplificada do conceito de nível, quanto maior o valor de l , maior é o número de *fill-in's* em L e U e conseqüentemente mais precisa será a fatoração.

Tanto o condicionador, quanto os métodos de Krylov usados neste trabalho são fornecidos pela biblioteca de domínio público PETSc (Balay *et al.*, 1997, 2000) desenvolvida e mantida pela Divisão de Matemáticas e Ciências da Computação do Laboratório Nacional de Argonne nos E.U.A..

4.2.5 Montagem da Matriz Jacobiana

A matriz Jacobiana é formada pelas derivadas parciais da função resíduo com relação às variáveis do problema (ver Eq. (4.8)). Sendo n a dimensão do problema, para o caso geral, tem-se n^2 derivadas a serem avaliadas. Problemas envolvendo a discretização de equações de conservação, por exemplo, geralmente possuem dimensão elevada, porém apresentam a matriz Jacobiana esparsa; sobretudo se malhas estruturadas são empregadas. Nestes casos, devido à complexidade das equações, pode ser vantajoso aproximar a matriz Jacobiana numericamente usando diferenças finitas. Este procedimento é característica dos métodos denominados de métodos de Newton discretizados ou discretos (Rheinboldt, 1996). Casos particulares destes métodos, muito populares atualmente, são os métodos *matrix-free*, assim chamados por não criarem a matriz Jacobiana diretamente. Ao invés disso, estes métodos aproximam o produto Jacobiana-vetor necessário aos métodos de Krylov.

Usando diferenças finitas de primeira ordem, tem-se:

$$J(x)v = \frac{F(x+hv) - F(x)}{h} \quad (4.27)$$

onde h é uma pequena perturbação e v é um vetor de base qualquer do espaço de Krylov.

Neste trabalho optou-se por calcular a perturbação h através do método proposto por Pernice e Walker (1998), onde

$$h = \frac{\sqrt{1 + \|x\|}}{\|v\|} e_{rel} \quad (4.28)$$

e e_{rel} é a raiz quadrada do erro relativo presente no cálculo da função resíduo F , cujo valor utilizado neste trabalho foi 10^{-8} , devido à função ser avaliada com dupla precisão.

Os métodos *matrix-free* têm a vantagem de não precisarem guardar a matriz Jacobiana. Por outro lado, há o problema da limitação do tipo de preconditionador que pode ser usado nestes métodos, já que não há matriz a ser preconditionada. No capítulo seguinte, será apresentado um método *matrix-free* com preconditionamento da equação linear com base em uma aproximação da matriz Jacobiana calculada apenas para esse fim.

Há várias outras maneiras possíveis de se aproximar a matriz Jacobiana, muitas delas envolvendo características específicas do problema numérico abordado. Métodos deste tipo são comumente chamados de quase-newtonianos (Pueyo e Zingg, 1997, 1998; Blanco e

Zingg, 1997). No próximo capítulo, será apresentado um método quase-newtoniano reportado por Maliska *et al.* (1997), em que a matriz Jacobiana gerada por um modelo *Black-Oil* padrão, discretizado por volumes finitos em coordenadas generalizadas, é aproximada apenas pelos termos ortogonais.

5 Métodos de Krylov-Newton Aplicados à Solução das Equações do Modelo *Black-Oil* Padrão

Neste capítulo serão apresentadas três versões do método de Krylov-Newton, que utilizam-se de determinadas características específicas da Eq. (3.18). Serão também abordados alguns aspectos de implementação e outros detalhes numéricos, tais como: obtenção das derivadas das funções residuais, controles de convergência e detalhes relativos ao problema transiente.

5.1 Linearização das Equações Discretizadas

Aplicando o método de linearização de Newton, Eq. (4.7), ao sistema não-linear (3.18), formado pelas equações discretizadas implícitas do modelo *Black-Oil* padrão, obtém-se o seguinte sistema linear $n \times n$ (n é número de volumes na malha)

$$\mathbf{J}\mathbf{s} = -\mathbf{F} \quad (5.1)$$

onde,

$$J_{ij} = \begin{bmatrix} \frac{\partial F_i^P}{\partial P_j} & \frac{\partial F_i^P}{\partial Z_{wj}} & \frac{\partial F_i^P}{\partial Z_{oj}} \\ \frac{\partial F_i^w}{\partial P_j} & \frac{\partial F_i^w}{\partial Z_{wj}} & \frac{\partial F_i^w}{\partial Z_{oj}} \\ \frac{\partial F_i^o}{\partial P_j} & \frac{\partial F_i^o}{\partial Z_{wj}} & \frac{\partial F_i^o}{\partial Z_{oj}} \end{bmatrix} \quad (5.2)$$

$$s_i = \begin{bmatrix} P_i - P_i^{k-1} \\ Z_{wi} - Z_{wi}^{k-1} \\ Z_{oi} - Z_{oi}^{k-1} \end{bmatrix} \quad (5.3)$$

$$F_i = \begin{bmatrix} F_i^P \\ F_i^w \\ F_i^o \end{bmatrix} \quad (5.4)$$

Os superescritos p , w e o de F referem-se, respectivamente, às funções resíduo das equações da pressão, da água e do óleo.

O sistema (5.1) tem a característica de ser um sistema esparso formado por blocos de matrizes densas 3x3. Outra característica importante é que cada equação de blocos envolve apenas os blocos correspondentes ao esquema numérico de 19 volumes da malha curvilínea estruturada, gerando uma estrutura de matriz de banda com 19 diagonais (ver Fig. 5.1). Ambos *solver* e preconditionador podem tirar proveito destas características visando criar algoritmos mais eficientes. O armazenamento da matriz é feito através do formato BCRS – ou *Block Compressed Row Storage* (Barret *et al.*, 1994) – onde os blocos não-zeros são compactados em cada linha e agrupados em sequência, utilizando dois conjuntos de índices que indicam a linha e coluna do primeiro elemento de cada bloco.

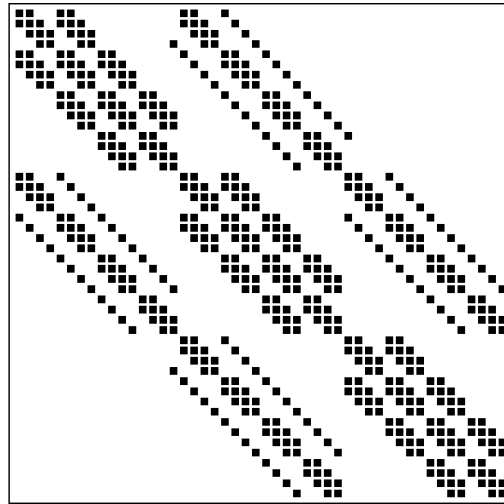


Figura 5.1 – Matriz de 19 diagonais (Ex.: malha 4x4x3)

Analisando mais de perto a equação linearizada da água no ponto P , tem-se:

$$\begin{aligned} & \left(\frac{\partial F_P^w}{\partial P_P} \right) \Delta P_P + \left(\frac{\partial F_P^w}{\partial Z_{wP}} \right) \Delta Z_{wP} + \left(\frac{\partial F_P^w}{\partial Z_{oP}} \right) \Delta Z_{oP} + \\ & \sum_{NB} \left[\left(\frac{\partial F_P^w}{\partial P_{NB}} \right) \Delta P_{NB} + \left(\frac{\partial F_P^w}{\partial Z_{wNB}} \right) \Delta Z_{wNB} + \left(\frac{\partial F_P^w}{\partial Z_{oNB}} \right) \Delta Z_{oNB} \right] = -F_P^w \end{aligned} \quad (5.5)$$

onde NB representa todos os vizinhos considerados no esquema numérico (W, E, N, S, B, T, SW, SE, NW, NE, BS, BW, BE, BN, TS, TW, TE, TN) e $F_P^w = F_P^w(P, Z_w, Z_o)$ é dada por

$$\begin{aligned}
 F_p^w = & -\frac{1}{J_p \Delta t} \left[(f_{r_m} Z_w)_p - (f_{r_m} Z_w)_p^o \right] + \frac{m_w}{J_p} \Big|_p \\
 & \left[D_{11}^w \frac{\mathcal{J}F_w}{\mathcal{J}^?} + D_{12}^w \frac{\mathcal{J}F_w}{\mathcal{J}^?} + D_{13}^w \frac{\mathcal{J}F_w}{\mathcal{J}^?} \right]_e - \\
 & \left[D_{11}^w \frac{\mathcal{J}F_w}{\mathcal{J}^?} + D_{12}^w \frac{\mathcal{J}F_w}{\mathcal{J}^?} + D_{13}^w \frac{\mathcal{J}F_w}{\mathcal{J}^?} \right]_w + \\
 & \left[D_{21}^w \frac{\mathcal{J}F_w}{\mathcal{J}^?} + D_{22}^w \frac{\mathcal{J}F_w}{\mathcal{J}^?} + D_{23}^w \frac{\mathcal{J}F_w}{\mathcal{J}^?} \right]_n - \\
 & \left[D_{21}^w \frac{\mathcal{J}F_w}{\mathcal{J}^?} + D_{22}^w \frac{\mathcal{J}F_w}{\mathcal{J}^?} + D_{23}^w \frac{\mathcal{J}F_w}{\mathcal{J}^?} \right]_s + \\
 & \left[D_{31}^w \frac{\mathcal{J}F_w}{\mathcal{J}^?} + D_{32}^w \frac{\mathcal{J}F_w}{\mathcal{J}^?} + D_{33}^w \frac{\mathcal{J}F_w}{\mathcal{J}^?} \right]_t - \\
 & \left[D_{31}^w \frac{\mathcal{J}F_w}{\mathcal{J}^?} + D_{32}^w \frac{\mathcal{J}F_w}{\mathcal{J}^?} + D_{33}^w \frac{\mathcal{J}F_w}{\mathcal{J}^?} \right]_b
 \end{aligned} \tag{5.6}$$

Observe que na equação acima assumiu-se um volume computacional com $\Delta \mathbf{x} = \Delta \mathbf{h} = \Delta \mathbf{g} = 1$; suposição esta que deve ser levada em consideração na geração da malha generalizada, todavia não representa nenhum tipo de restrição quanto a generalidade da mesma.

Visando tornar as equações das derivadas desta função mais compactas, define-se as derivadas normais dos potenciais nas interfaces do volume de controle

$$\frac{\partial F_p}{\partial \mathbf{n}} \Big|_e = \left[G_{11} \frac{\partial F_p}{\partial ?} + G_{12} \frac{\partial F_p}{\partial ?} + G_{13} \frac{\partial F_p}{\partial ?} \right]_e \tag{5.7}$$

$$\frac{\partial F_p}{\partial \mathbf{n}} \Big|_w = \left[G_{11} \frac{\partial F_p}{\partial ?} + G_{12} \frac{\partial F_p}{\partial ?} + G_{13} \frac{\partial F_p}{\partial ?} \right]_w \tag{5.8}$$

$$\frac{\partial F_p}{\partial \mathbf{n}} \Big|_n = \left[G_{21} \frac{\partial F_p}{\partial ?} + G_{22} \frac{\partial F_p}{\partial ?} + G_{23} \frac{\partial F_p}{\partial ?} \right]_n \tag{5.9}$$

$$\frac{\partial F_p}{\partial \mathbf{n}} \Big|_s = \left[G_{21} \frac{\partial F_p}{\partial ?} + G_{22} \frac{\partial F_p}{\partial ?} + G_{23} \frac{\partial F_p}{\partial ?} \right]_s \tag{5.10}$$

$$\frac{\partial F_p}{\partial \mathbf{n}} \Big|_t = \left[G_{31} \frac{\partial F_p}{\partial ?} + G_{32} \frac{\partial F_p}{\partial ?} + G_{33} \frac{\partial F_p}{\partial ?} \right]_t \tag{5.11}$$

$$\left. \frac{\partial F_p}{\partial \mathbf{n}} \right|_b = \left[G_{31} \frac{\partial F_p}{\partial ?} + G_{32} \frac{\partial F_p}{\partial ?} + G_{33} \frac{\partial F_p}{\partial ?} \right]_b \quad (5.12)$$

onde,

$$G_{ij} = \frac{D_{ij}^p}{I_p} \quad (5.13)$$

e \mathbf{n} é o vetor ortogonal à face não normalizado.

Como exemplo de derivadas da função resíduo da água, serão apresentadas aqui apenas as derivadas com relação as variáveis primárias P , Z_w e Z_o nos pontos P, E e NE. São elas:

- Derivadas da função resíduo da água com relação às variáveis primárias no ponto P :

$$\begin{aligned} \frac{\partial F_p^w}{\partial P_p} = & -\frac{Z_{wp}}{J_p \Delta t} \frac{\partial(\mathbf{f}r_m)_p}{\partial P_p} + \frac{1}{J_p} \frac{\partial m_w|_p}{\partial P_p} + \left. \frac{\partial F_w}{\partial \mathbf{n}} \right|_e \frac{\partial I_w|_e}{\partial P_p} - \left. \frac{\partial F_w}{\partial \mathbf{n}} \right|_w \frac{\partial I_w|_w}{\partial P_p} + \\ & \left. \frac{\partial F_w}{\partial \mathbf{n}} \right|_n \frac{\partial I_w|_n}{\partial P_p} - \left. \frac{\partial F_w}{\partial \mathbf{n}} \right|_s \frac{\partial I_w|_s}{\partial P_p} + \left. \frac{\partial F_w}{\partial \mathbf{n}} \right|_t \frac{\partial I_w|_t}{\partial P_p} - \left. \frac{\partial F_w}{\partial \mathbf{n}} \right|_b \frac{\partial I_w|_b}{\partial P_p} + \\ & \left[\begin{aligned} & F_e \left(-D_{11}^w|_e + D_{12}^w|_e F_{?P}^P + D_{13}^w|_e F_{?P}^P \right) - F_w \left(D_{11}^w|_w + D_{12}^w|_w F_{?P}^P + D_{13}^w|_w F_{?P}^P \right) + \\ & F_n \left(D_{21}^w|_n F_{?P}^P - D_{22}^w|_n + D_{23}^w|_n F_{?P}^P \right) - F_s \left(D_{21}^w|_s F_{?P}^P + D_{22}^w|_s + D_{23}^w|_s F_{?P}^P \right) + \\ & F_t \left(D_{31}^w|_t F_{?P}^P + D_{32}^w|_t F_{?P}^P - D_{33}^w|_t \right) - F_b \left(D_{31}^w|_b F_{?P}^P + D_{32}^w|_b F_{?P}^P + D_{33}^w|_b \right) \end{aligned} \right] \frac{\partial F_w|_p}{\partial P_p} \end{aligned} \quad (5.14)$$

$$\begin{aligned} \frac{\partial F_p^w}{\partial Z_{wp}} = & -\frac{\mathbf{f}_p}{J_p \Delta t} \left(\mathbf{r}_m|_p + Z_{wp} \frac{\partial \mathbf{r}_m|_p}{\partial Z_{wp}} \right) + \frac{1}{J_p} \frac{\partial m_w|_p}{\partial Z_{wp}} + \left. \frac{\partial F_w}{\partial \mathbf{n}} \right|_e \frac{\partial I_w|_e}{\partial Z_{wp}} - \left. \frac{\partial F_w}{\partial \mathbf{n}} \right|_w \frac{\partial I_w|_w}{\partial Z_{wp}} + \\ & \left. \frac{\partial F_w}{\partial \mathbf{n}} \right|_n \frac{\partial I_w|_n}{\partial Z_{wp}} - \left. \frac{\partial F_w}{\partial \mathbf{n}} \right|_s \frac{\partial I_w|_s}{\partial Z_{wp}} + \left. \frac{\partial F_w}{\partial \mathbf{n}} \right|_t \frac{\partial I_w|_t}{\partial Z_{wp}} - \left. \frac{\partial F_w}{\partial \mathbf{n}} \right|_b \frac{\partial I_w|_b}{\partial Z_{wp}} + \\ & \left[\begin{aligned} & F_e \left(-D_{11}^w|_e + D_{12}^w|_e F_{?P}^P + D_{13}^w|_e F_{?P}^P \right) - F_w \left(D_{11}^w|_w + D_{12}^w|_w F_{?P}^P + D_{13}^w|_w F_{?P}^P \right) + \\ & F_n \left(D_{21}^w|_n F_{?P}^P - D_{22}^w|_n + D_{23}^w|_n F_{?P}^P \right) - F_s \left(D_{21}^w|_s F_{?P}^P + D_{22}^w|_s + D_{23}^w|_s F_{?P}^P \right) + \\ & F_t \left(D_{31}^w|_t F_{?P}^P + D_{32}^w|_t F_{?P}^P - D_{33}^w|_t \right) - F_b \left(D_{31}^w|_b F_{?P}^P + D_{32}^w|_b F_{?P}^P + D_{33}^w|_b \right) \end{aligned} \right] \frac{\partial F_w|_p}{\partial Z_{wp}} \end{aligned} \quad (5.15)$$

$$\begin{aligned}
 \frac{\partial F_P^w}{\partial Z_{oP}} = & -\frac{(\mathbf{f}_P Z_w)_P}{J_P \Delta t} \frac{\partial \mathbf{r}_m|_P}{\partial Z_{oP}} + \frac{1}{J_P} \frac{\partial m_w|_P}{\partial Z_{oP}} + \frac{\partial F_w}{\partial \mathbf{n}} \Big|_e \frac{\partial \mathbf{I}_w|_e}{\partial Z_{oP}} - \frac{\partial F_w}{\partial \mathbf{n}} \Big|_w \frac{\partial \mathbf{I}_w|_w}{\partial Z_{oP}} + \\
 & \frac{\partial F_w}{\partial \mathbf{n}} \Big|_n \frac{\partial \mathbf{I}_w|_n}{\partial Z_{oP}} - \frac{\partial F_w}{\partial \mathbf{n}} \Big|_s \frac{\partial \mathbf{I}_w|_s}{\partial Z_{oP}} + \frac{\partial F_w}{\partial \mathbf{n}} \Big|_t \frac{\partial \mathbf{I}_w|_t}{\partial Z_{oP}} - \frac{\partial F_w}{\partial \mathbf{n}} \Big|_b \frac{\partial \mathbf{I}_w|_b}{\partial Z_{oP}} + \\
 & \left[\begin{aligned} & F_e \left(-D_{11}^w|_e + D_{12}^w|_e F_{?P}^P + D_{13}^w|_e F_{?P}^P \right) - F_w \left(D_{11}^w|_w + D_{12}^w|_w F_{?P}^P + D_{13}^w|_w F_{?P}^P \right) + \\ & F_n \left(D_{21}^w|_n F_{?P}^P - D_{22}^w|_n + D_{23}^w|_n F_{?P}^P \right) - F_s \left(D_{21}^w|_s F_{?P}^P + D_{22}^w|_s + D_{23}^w|_s F_{?P}^P \right) + \\ & F_t \left(D_{31}^w|_t F_{?P}^P + D_{32}^w|_t F_{?P}^P - D_{33}^w|_t \right) - F_b \left(D_{31}^w|_b F_{?P}^P + D_{32}^w|_b F_{?P}^P + D_{33}^w|_b \right) \end{aligned} \right] \frac{\partial F_w|_P}{\partial Z_{oP}}
 \end{aligned} \tag{5.16}$$

- Derivadas da função resíduo da água com relação às variáveis primárias no ponto E :

$$\begin{aligned}
 \frac{\partial F_P^w}{\partial P_E} = & \frac{\partial F_w}{\partial \mathbf{n}} \Big|_e \frac{\partial \mathbf{I}_w|_e}{\partial P_E} + \\
 & \left[\begin{aligned} & F_e \left(D_{11}^w|_e + D_{12}^w|_e F_{?E}^E + D_{13}^w|_e F_{?E}^E \right) + \\ & F_{?E}^P \left(F_n D_{21}^w|_n - F_s D_{21}^w|_s + F_t D_{31}^w|_t - F_b D_{31}^w|_b \right) \end{aligned} \right] \frac{\partial F_w|_E}{\partial P_E}
 \end{aligned} \tag{5.17}$$

$$\begin{aligned}
 \frac{\partial F_P^w}{\partial Z_{wE}} = & \frac{\partial F_w}{\partial \mathbf{n}} \Big|_e \frac{\partial \mathbf{I}_w|_e}{\partial Z_{wE}} + \\
 & \left[\begin{aligned} & F_e \left(D_{11}^w|_e + D_{12}^w|_e F_{?E}^E + D_{13}^w|_e F_{?E}^E \right) + \\ & F_{?E}^P \left(F_n D_{21}^w|_n - F_s D_{21}^w|_s + F_t D_{31}^w|_t - F_b D_{31}^w|_b \right) \end{aligned} \right] \frac{\partial F_w|_E}{\partial Z_{wE}}
 \end{aligned} \tag{5.18}$$

$$\begin{aligned}
 \frac{\partial F_P^w}{\partial Z_{oE}} = & \frac{\partial F_w}{\partial \mathbf{n}} \Big|_e \frac{\partial \mathbf{I}_w|_e}{\partial Z_{oE}} + \\
 & \left[\begin{aligned} & F_e \left(D_{11}^w|_e + D_{12}^w|_e F_{?E}^E + D_{13}^w|_e F_{?E}^E \right) + \\ & F_{?E}^P \left(F_n D_{21}^w|_n - F_s D_{21}^w|_s + F_t D_{31}^w|_t - F_b D_{31}^w|_b \right) \end{aligned} \right] \frac{\partial F_w|_E}{\partial Z_{oE}}
 \end{aligned} \tag{5.19}$$

- Derivadas da função resíduo da água com relação às variáveis primárias no ponto NE :

$$\frac{\partial F_P^w}{\partial P_{NE}} = \left[D_{12}^w|_e F_e F_{?NE}^E + D_{21}^w|_n F_n F_{?NE}^N \right] \frac{\partial F_w|_{NE}}{\partial P_{NE}} \tag{5.20}$$

$$\frac{\partial F_P^w}{\partial Z_{wNE}} = \left[D_{12}^w \Big|_e F_e F_{?NE}^E + D_{21}^w \Big|_n F_n F_{?NE}^N \right] \frac{\partial F_w \Big|_{NE}}{\partial Z_{wNE}} \quad (5.21)$$

$$\frac{\partial F_P^w}{\partial Z_{oNE}} = \left[D_{12}^w \Big|_e F_e F_{?NE}^E + D_{21}^w \Big|_n F_n F_{?NE}^N \right] \frac{\partial F_w \Big|_{NE}}{\partial Z_{oNE}} \quad (5.22)$$

Resta ainda avaliar as derivadas de I_p , $X_{oo}I_o$, Φ , $f\mathbf{r}_m$ e \mathbf{r}_m . Devido a complexidade de se obter estas derivadas analiticamente (equações muito grandes), optou-se por avaliá-las numericamente através de diferenças finitas de primeira ordem. Para tanto, utilizou-se a sugestão de Cunha (1996), em que perturbações iguais a $0,001P$ e $0,01Z_p$ (10^{-5} , se $Z_p < 0,001$) são usadas para a aproximação das derivadas com relação, respectivamente, à pressão e às frações mássicas.

5.2 Os métodos de Krylov-Newton

O primeiro método de Krylov-Newton proposto surge da aplicação direta do algoritmo (4.12) em conjunto com algum processo iterativo de Krylov (GMRES, BICGSTAB, TFQMR, etc) para resolver a equação linearizada (5.1). Neste método a matriz Jacobiana é montada como foi mostrado na seção 5.1. Todos os termos das equações resíduo são considerados implícitos (mesmo nível iterativo), gerando uma matriz com 19 diagonais.

Um segundo método, proposto e testado por Maliska *et al.* (1997) e aplicado também por Cunha (1996) e Czesnat (1999), é derivado de uma aproximação da matriz Jacobiana em que apenas os termos ortogonais (P, W, E, S, N, B, T) são considerados. Isto é feito tornando todas as derivadas cruzadas dos potenciais em relação a \mathbf{x} , \mathbf{h} e $\underline{\mathbf{g}}$ explícitas, ou seja, avaliando-as no nível iterativo anterior. Desta forma, derivadas do tipo

$$\frac{\partial}{\partial P_E} \left(\frac{\partial F}{\partial \underline{\mathbf{g}}_e} \right) \quad (5.23)$$

serão consideradas nulas.

Neste caso, as equações (5.14) a (5.22), ficam da seguinte forma:

- Derivadas da função resíduo da água com relação às variáveis primárias no ponto P :

$$\begin{aligned} \frac{\partial F_P^w}{\partial P_P} &= -\frac{Z_{wP}}{J_P \Delta t} \frac{\partial(\mathbf{f}r_m)_P}{\partial P_P} + \frac{1}{J_P} \frac{\partial m_w|_P}{\partial P_P} + \frac{\partial F_w}{\partial \mathbf{n}} \Big|_e \frac{\partial \mathbf{I}_w|_e}{\partial P_P} - \frac{\partial F_w}{\partial \mathbf{n}} \Big|_w \frac{\partial \mathbf{I}_w|_w}{\partial P_P} + \\ &\frac{\partial F_w}{\partial \mathbf{n}} \Big|_n \frac{\partial \mathbf{I}_w|_n}{\partial P_P} - \frac{\partial F_w}{\partial \mathbf{n}} \Big|_s \frac{\partial \mathbf{I}_w|_s}{\partial P_P} + \frac{\partial F_w}{\partial \mathbf{n}} \Big|_t \frac{\partial \mathbf{I}_w|_t}{\partial P_P} - \frac{\partial F_w}{\partial \mathbf{n}} \Big|_b \frac{\partial \mathbf{I}_w|_b}{\partial P_P} - \\ &\left(F_e D_{11}^w \Big|_e + F_w D_{11}^w \Big|_w + F_n D_{22}^w \Big|_n + F_s D_{22}^w \Big|_s + F_t D_{33}^w \Big|_t + F_b D_{33}^w \Big|_b \right) \frac{\partial F_w|_P}{\partial P_P} \end{aligned} \quad (5.24)$$

$$\begin{aligned} \frac{\partial F_P^w}{\partial Z_{wP}} &= -\frac{\mathbf{f}_P}{J_P \Delta t} \left(\mathbf{r}_m|_P + Z_{wP} \frac{\partial \mathbf{r}_m|_P}{\partial Z_{wP}} \right) + \frac{1}{J_P} \frac{\partial m_w|_P}{\partial Z_{wP}} + \frac{\partial F_w}{\partial \mathbf{n}} \Big|_e \frac{\partial \mathbf{I}_w|_e}{\partial Z_{wP}} - \frac{\partial F_w}{\partial \mathbf{n}} \Big|_w \frac{\partial \mathbf{I}_w|_w}{\partial Z_{wP}} + \\ &\frac{\partial F_w}{\partial \mathbf{n}} \Big|_n \frac{\partial \mathbf{I}_w|_n}{\partial Z_{wP}} - \frac{\partial F_w}{\partial \mathbf{n}} \Big|_s \frac{\partial \mathbf{I}_w|_s}{\partial Z_{wP}} + \frac{\partial F_w}{\partial \mathbf{n}} \Big|_t \frac{\partial \mathbf{I}_w|_t}{\partial Z_{wP}} - \frac{\partial F_w}{\partial \mathbf{n}} \Big|_b \frac{\partial \mathbf{I}_w|_b}{\partial Z_{wP}} - \\ &\left(F_e D_{11}^w \Big|_e + F_w D_{11}^w \Big|_w + F_n D_{22}^w \Big|_n + F_s D_{22}^w \Big|_s + F_t D_{33}^w \Big|_t + F_b D_{33}^w \Big|_b \right) \frac{\partial F_w|_P}{\partial Z_{wP}} \end{aligned} \quad (5.25)$$

$$\begin{aligned} \frac{\partial F_P^w}{\partial Z_{oP}} &= -\frac{(\mathbf{f}_P Z_w)_P}{J_P \Delta t} \frac{\partial \mathbf{r}_m|_P}{\partial Z_{oP}} + \frac{1}{J_P} \frac{\partial m_w|_P}{\partial Z_{oP}} + \frac{\partial F_w}{\partial \mathbf{n}} \Big|_e \frac{\partial \mathbf{I}_w|_e}{\partial Z_{oP}} - \frac{\partial F_w}{\partial \mathbf{n}} \Big|_w \frac{\partial \mathbf{I}_w|_w}{\partial Z_{oP}} + \\ &\frac{\partial F_w}{\partial \mathbf{n}} \Big|_n \frac{\partial \mathbf{I}_w|_n}{\partial Z_{oP}} - \frac{\partial F_w}{\partial \mathbf{n}} \Big|_s \frac{\partial \mathbf{I}_w|_s}{\partial Z_{oP}} + \frac{\partial F_w}{\partial \mathbf{n}} \Big|_t \frac{\partial \mathbf{I}_w|_t}{\partial Z_{oP}} - \frac{\partial F_w}{\partial \mathbf{n}} \Big|_b \frac{\partial \mathbf{I}_w|_b}{\partial Z_{oP}} - \\ &\left(F_e D_{11}^w \Big|_e + F_w D_{11}^w \Big|_w + F_n D_{22}^w \Big|_n + F_s D_{22}^w \Big|_s + F_t D_{33}^w \Big|_t + F_b D_{33}^w \Big|_b \right) \frac{\partial F_w|_P}{\partial Z_{oP}} \end{aligned} \quad (5.26)$$

- Derivadas da função resíduo da água com relação às variáveis primárias no ponto E :

$$\frac{\partial F_P^w}{\partial P_E} = \frac{\partial F_w}{\partial \mathbf{n}} \Big|_e \frac{\partial \mathbf{I}_w|_e}{\partial P_E} + F_e D_{11}^w \Big|_e \frac{\partial F_w|_E}{\partial P_E} \quad (5.27)$$

$$\frac{\partial F_P^w}{\partial Z_{wE}} = \frac{\partial F_w}{\partial \mathbf{n}} \Big|_e \frac{\partial \mathbf{I}_w|_e}{\partial Z_{wE}} + F_e D_{11}^w \Big|_e \frac{\partial F_w|_E}{\partial Z_{wE}} \quad (5.28)$$

$$\frac{\partial F_P^w}{\partial Z_{oE}} = \frac{\partial F_w}{\partial \mathbf{n}} \Big|_e \frac{\partial I_w}{\partial Z_{oE}} + F_e D_{11}^w \Big|_e \frac{\partial F_w}{\partial Z_{oE}} \quad (5.29)$$

- Derivadas da função resíduo da água com relação às variáveis primárias no ponto NE :

$$\frac{\partial F_P^w}{\partial P_{NE}} = \frac{\partial F_P^w}{\partial Z_{wNE}} = \frac{\partial F_P^w}{\partial Z_{oNE}} = 0 \quad (5.30)$$

A Fig. 5.2 mostra um esquema da estrutura hepta-diagonal da matriz Jacobiana. É importante ter mente que, para malhas ortogonais, a matriz Jacobiana aproximada torna-se exata.

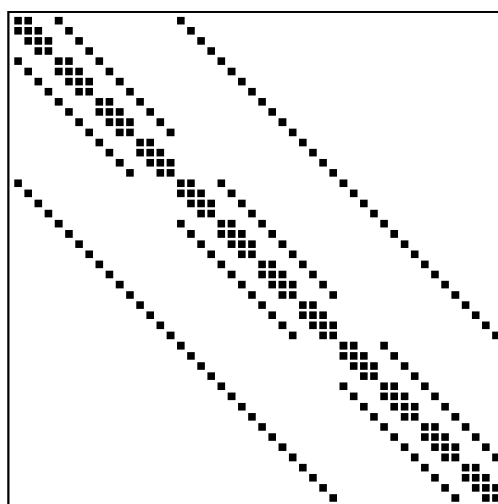


Figura 5.2 – Matriz de 7 diagonais (Ex.: malha 4x4x3)

Maliska *et al.* (1997) sugerem este método de Newton aproximado, ou quase-newtoniano, para malhas quase-ortogonais, indicando que, com o aumento da não-ortogonalidade e, conseqüentemente, o empobrecimento da aproximação, provavelmente haveria dificuldade de convergência. No entanto, carecia-se de um estudo mais aprofundado do real efeito da não-ortogonalidade neste método. Questões do tipo:

- Há realmente uma diferença significativa entre o método Quase-Newtoniano de Maliska *et al.* e o método de Newton com a Jacobiana completa do ponto de vista da performance para malhas razoavelmente não-ortogonais?
- Se há diferença, até que grau de não-ortogonalidade é mais interessante usar um ou

outro método?

Este trabalho visa lançar alguma luz sobre estas indagações e o resultado encontra-se no capítulo seguinte.

Uma contribuição importante deste trabalho, ainda não mencionada, é a introdução de um novo método baseado na mesma proposta de Maliska *et al.* para uma aproximação da matriz Jacobiana. Este método é uma variante dos métodos *matrix-free*, introduzidos no capítulo 4, onde a matriz Jacobiana completa (19 diagonais) é aproximada por diferenças finitas de primeira ordem (ou melhor, aproxima-se o produto Jacobiana-vetor), enquanto a matriz aproximada (7 diagonais) é utilizada apenas como matriz de condicionamento. A grande virtude deste método é combinar características, tanto do método de Newton Inexato completo, quanto do método de Maliska *et al.*. Através do algoritmo *matrix-free* tem-se o efeito da Jacobiana completa, sem, no entanto, a necessidade de criá-la e armazená-la, mantendo a Jacobiana aproximada, mais simples, para se ter um bom condicionamento. Esta técnica, adaptada aqui ao problema da não-ortogonalidade, foi desenvolvida no contexto das funções interpolações (Gropp e Keyes, 1992; Chin *et al.*, 1992; Johnson, McHugh e Knoll, 1992, 1996; Blanco e Zingg, 1997; Pueyo e Zingg, 1998). Neste caso, aproxima-se a matriz Jacobiana por um método de primeira ordem, enquanto as funções resíduo são discretizadas por métodos de alta ordem. Esta técnica é denominada *defect correction*.

5.3 Controle Automático do Passo de Tempo

Um aspecto importantíssimo da simulação de reservatórios é que o problema é essencialmente transiente e o histórico da extração de óleo é um resultado chave para o estudo do reservatório. Sendo assim, a solução precisa das variáveis em cada passo de tempo é a norma neste tipo de problema e sua obtenção está diretamente associada à magnitude deste passo de tempo.

Uma boa escolha para o passo de tempo deve assegurar estabilidade e erros de truncamento dentro do aceitável. Ambos os requisitos são, ao menos qualitativamente, relacionados à taxa de variação da pressão e das saturações no reservatório. Quanto mais rápidas são essas variações, menor deve ser o passo de tempo usado. Por exemplo, o passo de tempo usado na simulação de um reservatório contendo um único poço pode variar de 0,01

dias logo após uma mudança drástica de vazão até vários meses quando o reservatório está próximo ao esgotamento (Aziz e Settari, 1979).

O problema da estabilidade é em grande parte resolvido pela solução implícita das equações, todavia a correta escolha do passo de tempo visando amenizar os erros de truncamento é uma tarefa difícil de ser realizada manualmente. Neste trabalho, utilizou-se um esquema para a escolha automática do passo de tempo baseado em controles empíricos como sugerido por Aziz e Settari (1979). Neste esquema, o Δt escolhido para a próxima iteração será

$$\Delta t^{k+1} = \min \left\{ \Delta t_{\max}, \max \left\{ \Delta t_{\min}, 0, 5\Delta t^k, \min \left\{ \Delta t_S, \Delta t_P, 2\Delta t^k \right\} \right\} \right\} \quad (5.31)$$

com

$$\Delta t_S = \frac{\Delta t^k \Delta S_{\text{lim}}}{\Delta S_{\max}^k}, \quad \Delta S_{\max}^k = \max_{ijk,p} \left\{ S_p^k - S_p^{k-1} \right\} \quad (5.32)$$

$$\Delta t_P = \frac{\Delta t^k \Delta P_{\text{lim}}}{\Delta P_{\max}^k}, \quad \Delta P_{\max}^k = \max_{ijk} \left\{ P^k - P^{k-1} \right\} \quad (5.33)$$

e Δt_{\max} , Δt_{\min} , ΔS_{lim} e ΔP_{lim} são parâmetros fornecidos pelo usuário.

Colocando de outra forma, a relação acima quer dizer que o mínimo passo de tempo entre os valores permitidos pela variação da pressão e saturação, Δt_P e Δt_S (respectivamente), deve respeitar a seguinte relação

$$\Delta t_{\min} \leq \Delta t^k / 2 \leq \min \{ \Delta t_P, \Delta t_S \} \leq 2\Delta t^k \leq \Delta t_{\max} \quad (5.34)$$

Se ΔP_{\max}^k ou ΔS_{\max}^k , ao final de um *time step*, forem maiores do que os valores limites estipulados o passo de tempo é rejeitado e reinicia-se a simulação com um novo passo de tempo menor dado pela Eq. (5.31).

Como será evidenciado no capítulo seguinte, o fato da simulação ser transiente traz características novas ao problema não-linear. Uma característica óbvia é que agora passam a ser vários problemas não-lineares resolvidos em sequência e não apenas um. Outra característica é que a estimativa inicial de cada método de Newton é, na realidade, a solução do passo de tempo anterior e, dependendo do controle imposto ao passo de tempo (mais ou menos restritivo) esta estimativa pode estar razoavelmente próxima à solução buscada. Ou

seja, com controles restritivos do passo de tempo, a convergência dos métodos de Newton inexatos está sempre muito próxima do regime assintótico, dispensando o *backtracking*.

5.4 Outros Detalhes Relativos à Implementação

Esta seção trata de alguns detalhes numéricos importantes ainda não abordados.

O uso de q_{\min} e q_{\max} na linha 5 do algoritmo (4.12) para determinar, respectivamente, a menor e a maior redução permitida para o passo do método de Newton inexato é uma técnica conhecida na área de otimização como *safeguarded backtracking*. Normalmente utiliza-se os valores de 0,1 e 0,5 e $q \in [q_{\min}, q_{\max}]$ é determinado de forma a minimizar um polinômio de segundo grau $p(y)$, que satisfaz

$$\begin{aligned} p(0) &= \frac{1}{2} \|F(x^k)\|^2 \\ p(1) &= \frac{1}{2} \|F(x^k + s)\|^2 \\ p'(0) &= \left. \frac{d}{dy} \frac{1}{2} \|F(x^k + ys)\|^2 \right|_{y=0} \end{aligned} \tag{5.35}$$

ou seja,

$$q = - \frac{\langle J(x^k)s, F(x^k) \rangle}{\left(\|F(x^k + s)\|^2 - \|F(x^k)\|^2 - 2\langle J(x^k)s, F(x^k) \rangle \right)} \tag{5.36}$$

Além disso, utilizou-se $h_{\max} = 0,9$, que é um valor razoavelmente grande, permitindo que os h_k 's possam tornar-se elevados, se necessário, evitando o *oversolving*. Utilizou-se ainda $t = 10^{-4}$, permitindo que praticamente todos os passos que promovam alguma redução em $\|F\|$ sejam aceitos sem *backtracking*.

A convergência do método newtoniano é declarada quando

$$\text{Max}_i \left[\frac{|s_i^k|}{\max[|x_i^k|, 1]} \right] < 10^{-p} \quad \text{e} \quad \|F\| < 10^{-p} \tag{5.37}$$

O primeiro critério provê p dígitos de acurácia quando a magnitude da variável

primária é maior do que 1 (pressão) e p décimos de acurácia quando a variável é menor do que 1 (fração mássica) (Dennis Jr. e Schnabel, 1983), onde p é o expoente da tolerância em (5.37). Foi usado um valor de 6 para p .

O número máximo de passos permitidos no procedimento de *backtracking*, j_{\max} , é 8.

6 Resultados e Discussões

Neste capítulo, os diferentes métodos de Newton-Krylov apresentados anteriormente são testados, analisados e comparados em diversos problemas com diferentes níveis de não-ortogonalidade da malha.

Estes problemas foram divididos em três seções:

- Validação da implementação. Aborda a comparação entre as soluções obtidas com os métodos de Krylov-Newton implementados e a reportada em um caso encontrado na literatura envolvendo uma malha não-ortogonal.
- Paralelepípedo deformado. Analisa e apresenta os resultados de uma série de problemas baseados na deformação tridimensional gradativa de um reservatório cartesiano, envolvendo o escoamento bifásico imiscível água-óleo com diferentes curvas de permeabilidade relativa.
- Reservatório real. Apresenta os resultados de cada método de Newton inexato para um escoamento semelhante ao anterior em um reservatório dotado de uma geometria mais realística com uma distribuição heterogênea de não-ortogonalidade na malha gerada.

6.1 Validação da Implementação

Esta seção tem por objetivo apresentar a validação da implementação dos três métodos de Krylov-Newton introduzidos nos capítulos anteriores com base em um caso colhido na literatura, conhecido como problema dos 3 poços.

Este problema, originalmente proposto por Hegre *et al.* (1986), foi utilizado por Czesnat (1999) para estudar o efeito de orientação de malha. Sua principal característica consiste no posicionamento simétrico dos poços produtores com relação ao poço injetor como mostra a Fig. 6.1, sendo que a malha utilizada cria um caminho preferencial para o avanço da frente de água entre o poço produtor 2 e o poço injetor (ver Fig. 6.2).

Como é mostrado na Fig. 6.2, a malha utilizada neste caso possui um ângulo constante

de 45° entre as linhas ξ e η , sendo, portanto, não-ortogonal. Devido a esta característica, optou-se por adotar os resultados obtidos por Czesnat com o método UDS como parâmetro de validação dos métodos de Krylov-Newton implementados.

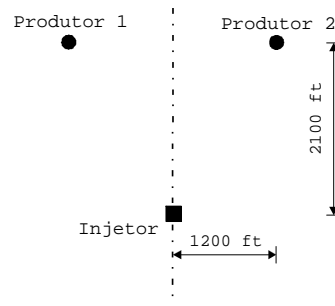


Figura 6.1 – Distribuição e configuração dos poços

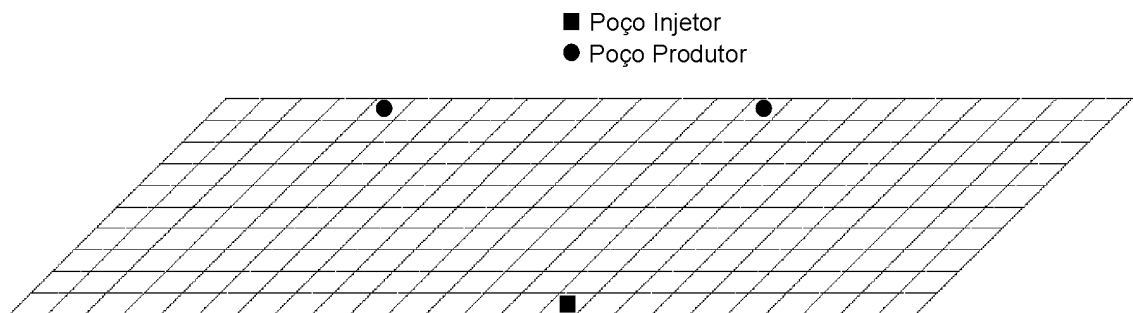


Figura 6.2 – Malha utilizada para o caso dos 3 poços (24x10)

As tabelas 6.1 e 6.2 mostram os dados do problema.

Tabela 6.1 – Dados do problema dos 3 poços

Dados	Valor
Pressão inicial	$P^{ini} = 2,725 \times 10^7$ Pa (3952 psi)
Saturação inicial de água	$S_w^{ini} = 0,0$
Porosidade	$f = f_{ref} [1 + c_r (P - P_{ref})]$
Porosidade de referência	$f_{ref} = 0,19$
Compressibilidade da rocha	$c_r = 4,35 \times 10^{-10}$ 1/Pa (3×10^{-6} 1/psi)

Pressão de referência	$P_{ref} = 1,0135 \times 10^5 \text{ Pa (14,7 psi)}$
Densidade da água	$\rho_w^{STD} = 996,96 \text{ kg/m}^3$
Densidade do óleo	$\rho_o^{STD} = 740,76 \text{ kg/m}^3$
Viscosidade da água	$\mu_w = 5 \times 10^{-4} \text{ Ns/m}^2 \text{ (0,5cp)}$
Viscosidade do óleo	$\mu_o = 2 \times 10^{-3} \text{ Ns/m}^2 \text{ (2,0cp)}$
Fator volume de formação (FVF)	$B_p = \frac{B_p^{ref}}{1 + c_p(P - P_{ref})}$
FVF de referência da água	$B_w^{ref} = 1,0$
FVF de referência do óleo	$B_o^{ref} = 0,96$
Compressibilidade da água	$c_w = 4,35 \times 10^{-10} \text{ 1/Pa (} 3 \times 10^{-6} \text{ 1/psi)}$
Compressibilidade do óleo	$c_o = 1,45 \times 10^{-9} \text{ 1/Pa (} 1 \times 10^{-5} \text{ 1/psi)}$
Permeabilidade absoluta	$k_x = k_y = k_z = 4,935 \times 10^{-14} \text{ m}^2 \text{ (50 mD)}$
Vazão do injetor	$3,53 \times 10^{-3} \text{ m}^3/\text{s (1920 STB/dias)}$
Vazão nos produtores	$1,84 \times 10^{-3} \text{ m}^3/\text{s (1000 STB/dias)}$
Δt	$129600 \text{ s (1.5 dias)}$

Tabela 6.2 – Permeabilidade relativa do problema dos 3 poços

S^w	K^{rw}	K^{ro}
0.25	0	0.92
0.3	0.02	0.705
0.4	0.055	0.42
0.5	0.1	0.24
0.6	0.145	0.11
0.7	0	0

Para cada um dos três grupos de métodos de Krylov-Newton, foram usados os seguintes solvers de Krylov: GMRES(r), TFQMR e BICGSTAB. O GMRES(r) é uma variação do

GMRES tradicional onde a base de Krylov construída pelo algoritmo tem dimensão máxima r , uma vez atingida a dimensão máxima o processo é reiniciado. Todos os algoritmos são preconditionados com o BILU(n) e três valores de n foram testados: 0, 1 e 2.

É interessante observar que, como este caso é bidimensional, a matriz Jacobiana completa tem na realidade 9 diagonais e a matriz aproximada apenas 5.

Todos os casos apresentaram exatamente os mesmos resultados. A Fig. 6.3 mostra os gráficos do corte de água (razão entre a vazão de água e a vazão de líquido – água+óleo – que passa por um poço) nos poços produtores para os três tipos de métodos de Krylov-Newton discutidos no capítulo anterior e também os obtidos por Czesnat (1999). Todos os métodos de Krylov-Newton mostrados na figura utilizam o GMRES(30) com BILU de nível 1 como preconditionador. Como era esperado, simplesmente não há diferença entre as soluções, mostrando que a implementação dos métodos está correta.

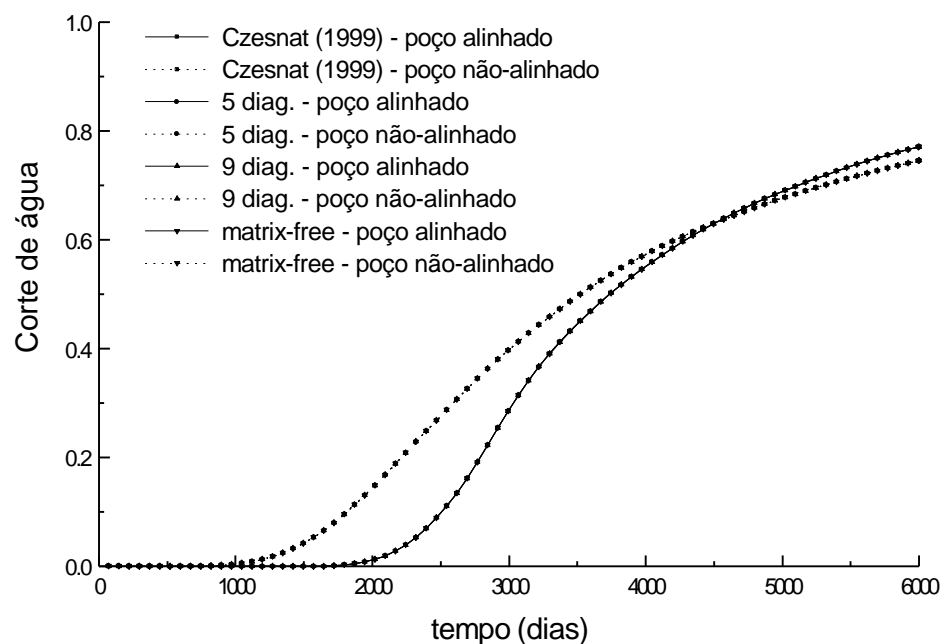


Figura 6.3 – Gráfico do corte de água nos poços produtores

6.2 Paralelepípedo Deformado

Nesta seção, serão apresentados os resultados referentes à aplicação dos métodos de Krylov-Newton, vistos no capítulo 5, em uma série de casos baseados em reservatórios em forma de paralelepípedo deformado por um ângulo α como é mostrado na Fig. 6.4. α é o ângulo entre \mathbf{x} e \mathbf{h} e também entre $\underline{\xi}$ e o plano $\mathbf{x}-\mathbf{h}$, que coincide com o plano x-y (não mostrado na figura). Com isso todos os G_{ij} (Eq. (5.13)) são não-nulos para $\alpha < \pi/2$ e consequentemente a matriz Jacobiana completa possui 19 diagonais não-nulas.

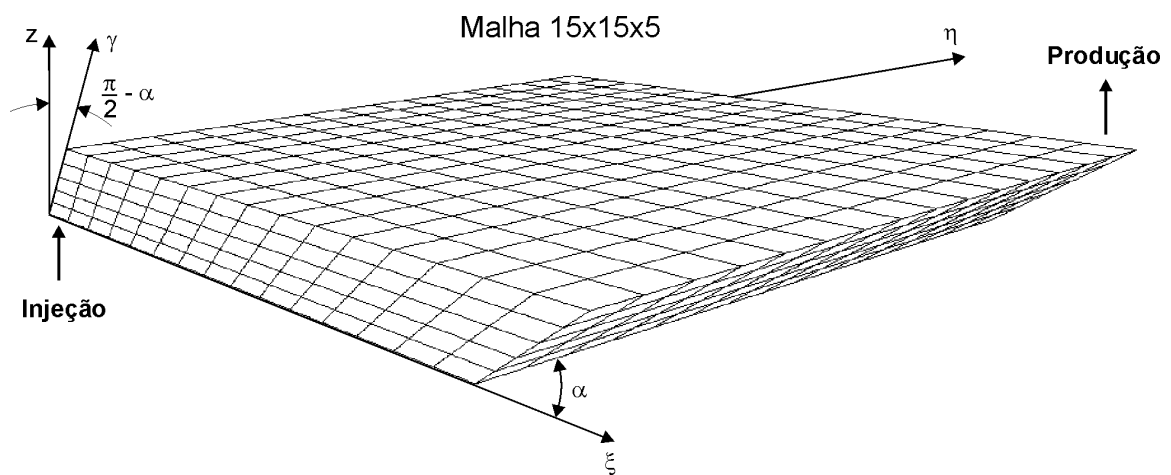


Figura 6.4 - Geometria e malha utilizadas nos testes

O comprimento das arestas nas direções \mathbf{x} e \mathbf{h} é de 1 km e o comprimento na direção $\underline{\xi}$ varia de acordo com o ângulo α de maneira que todos os reservatórios simulados tenham o mesmo volume igual a 10^7 m^3 . Conforme é ilustrado na Fig. 6.4, há dois poços localizados em cantos opostos da maior diagonal do paralelepípedo: um injetor e um produtor. O poço injetor introduz água a uma vazão constante de 1920 barris/dia ($3,53 \times 10^{-3} \text{ m}^3/\text{s}$) no reservatório, enquanto o poço produtor extrai a mesma vazão de líquido, que pode ser apenas óleo, água+óleo ou apenas água dependendo do tempo da simulação. Esta configuração caracteriza a simulação de um método de recuperação denominado secundário (Mezzomo, 1992), onde água (ou gás) é injetada no interior do reservatório com o objetivo de manter a pressão elevada e deslocar o petróleo para os poços produtores.

Utilizou-se uma malha estruturada generalizada do tipo “cartesiana uniforme

deformada” (ver Fig. 6.4) com $15 \times 15 \times 5$ ($x \times h \times z$) volumes. O poço injetor está localizado no volume (1, 1, 1) e o produtor, no volume (15, 15, 5). Todas as fronteiras são consideradas impermeáveis e as vazões mássicas nos poços são introduzidas no modelo matemático diretamente através dos termos fonte/sumidouro. O escoamento é bifásico, envolvendo somente água e óleo. Optou-se por não introduzir o componente gás no modelo, já que isso não iria acrescentar muito à análise feita e, por outro lado, aumentaria em muito os tempos de computação. Os efeitos gravitacionais e de capilaridade foram negligenciados.

As curvas de permeabilidade relativa do sistema água-óleo são dadas pelas seguintes equações

$$k_{rw}(S^w) = k_{rw}^o \left(\frac{S_w - S_{wc}}{1 - S_{wc} - S_{or}} \right)^{n_w} \quad (6.1)$$

$$k_{ro}(S^w) = k_{ro}^o \left(\frac{1 - S_w - S_{or}}{1 - S_{wc} - S_{or}} \right)^{n_o} \quad (6.2)$$

onde, k_{rw}^o é a permeabilidade relativa da água para a saturação de $1 - S_{or}$;

k_{ro}^o é a permeabilidade relativa do óleo para a saturação de água irreduzível;

S_{wc} é a saturação de água irreduzível;

S_{or} é a saturação de óleo residual;

n_w e n_o são expoentes.

Estes e outros dados relativos ao modelo físico do problema estão condensados na tabela 6.3.

Tabela 6.3 – Dados do modelo físico (Paralelepípedos Deformados)

Dados	Valor
Pressão inicial	$P^{ini} = 2,725 \times 10^7$ Pa (3952 psi)
Saturação inicial de água	$S_w^{ini} = 0,0$
Porosidade	$f = f_{ref} [1 + c_r (P - P_{ref})]$
Porosidade de referência	$f_{ref} = 0,19$

Compressibilidade da rocha	$c_r = 4,35 \times 10^{-10} \text{ 1/Pa } (3 \times 10^{-6} \text{ 1/psi})$
Pressão de referência	$P_{ref} = 1,0135 \times 10^5 \text{ Pa } (14,7 \text{ psi})$
Densidade da água	$\rho_w^{STD} = 996,96 \text{ kg/m}^3$
Densidade do óleo	$\rho_o^{STD} = 740,76 \text{ kg/m}^3$
Viscosidade da água	$\mu_w = 5 \times 10^{-4} \text{ Ns/m}^2 (0,5 \text{ cp})$
Viscosidade do óleo	$\mu_o = 2 \times 10^{-3} \text{ Ns/m}^2 (2,0 \text{ cp})$
Fator volume de formação (FVF)	$B_p = \frac{B_p^{ref}}{1 + c_p (P - P_{ref})}$
FVF de referência da água	$B_w^{ref} = 1,0$
FVF de referência do óleo	$B_o^{ref} = 0,96$
Compressibilidade da água	$c_w = 4,35 \times 10^{-10} \text{ 1/Pa } (3 \times 10^{-6} \text{ 1/psi})$
Compressibilidade do óleo	$c_o = 1,45 \times 10^{-9} \text{ 1/Pa } (1 \times 10^{-5} \text{ 1/psi})$
Permeabilidade absoluta	$k_x = k_y = k_z = 4,935 \times 10^{-14} \text{ m}^2 (50 \text{ mD})$
Aceleração da gravidade	$g = 0,0 \text{ m}^2/\text{s}$
Pressão capilar do sistema água-óleo	$P_{C_{ow}} = 0,0$
Saturação de água irreduzível	$S_{wc} = 0,0$
Saturação de óleo residual	$S_{or} = 0,0$
k_{rw}^o, k_{ro}^o	$k_{rw}^o = k_{ro}^o = 1,0$
Vazão de injeção/produção	$q = 3,53 \times 10^{-3} \text{ m}^3/\text{s} (1920 \text{ STB/dias})$

A simulação é encerrada após atingir 2 VPI (Volume Poroso Injetado), ou seja, após a injeção de um volume de água equivalente a duas vezes o volume poroso do reservatório. Foi usado um passo de tempo variável de acordo com o método descrito na seção 5.3 com $\Delta t_{inicial} = 1 \text{ dia}$ e $\Delta t_{max} = 20 \text{ dias}$.

Os principais parâmetros físicos que foram alterados de problema em problema são:

- Ângulo de deformação do reservatório, \mathbf{a} : Visa avaliar o peso das diagonais extras na matriz Jacobiana em problemas não-ortogonais. Quanto menor o valor de \mathbf{a} , maior é a não-ortogonalidade da malha. Foram testados casos com valores de \mathbf{a} iguais a 30° , 45° , 60° , 80° e 90° . Observe que, para $\mathbf{a} = 90^\circ$, tem-se uma malha cartesiana ortogonal e, portanto, a mesma matriz Jacobiana de apenas 7 diagonais será usada por todos os métodos.
- Expoentes n_w e n_o das curvas de permeabilidade relativa: Visa avaliar o peso da não-linearidade das equações. Neste caso, foi usado apenas um único expoente $n = n_w = n_o$. Quanto maior o valor de n , maior é a não-linearidade do sistema de equações. Foram testados casos com valores de n iguais a 1,5, 2,0, 2,5, 3,0, 3,5 e 4,0.

Os parâmetros numéricos, que são fixados para todos os casos testados, estão resumidos na tabela 6.4. As demais opções, tais como o *solver* do sistema linear, nível de *fill-in* do BILU, *backtracking* e *forcing terms*, serão analisados individualmente nas próximas seções. Observe que as condições de convergência impostas tanto ao problema não-linear a cada passo de tempo, quanto ao problema transiente podem ser consideradas razoavelmente restritivas.

Tabela 6.4 – Dados do modelo numérico (Paralelepípedos Deformados)

Dado	Valor
Máx. número de iterações do método de Krylov-Newton	500
Máx. número de iterações do <i>solver</i> linear	100
Expoente das tolerâncias (Eq. (5.37))	$p = 6$
Varição limite aceitável para a pressão	$\Delta P_{\text{lim}} = 5,0 \times 10^4 \text{ Pa (72,52 psi)}$
Varição limite aceitável para a saturação	$\Delta S_{\text{lim}} = 0,005$
Passo de tempo inicial	$\Delta t_{\text{inicial}} = 1 \text{ dia}$
Passo de tempo mínimo	$\Delta t_{\text{inicial}} = 0,01 \text{ dia}$
Passo de tempo máximo	$\Delta t_{\text{max}} = 20 \text{ dias}$

6.2.1 *Backtracking e Forcing terms*

Nesta seção, procurou-se, para cada método de Krylov-Newton, avaliar as vantagens, se houverem, da introdução de *backtracking* e dos métodos de escolha dos *forcing terms* sugeridos por Eisenstat e Walker (1994, 1996) e apresentados no capítulo 4.

As tabelas 6.5 e 6.6 apresentam os resultados, respectivamente, com e sem *backtracking* para um reservatório deformado de 45° e com vários valores do expoente das saturações. Em ambas foram utilizados o método de Krylov-Newton com a matriz Jacobiana completa com 19 diagonais, GMRES com *restart* (base contendo no máximo 25 vetores), BILU de nível 0 e o método 1 para escolha dos *forcing terms*.

Como já era esperado, a condição necessária ao início do *backtracking* praticamente não é atingida e, mesmo quando é, o efeito do *backtracking* é tênue, podendo ser desprezado. Isso pode ser explicado pelo fato do problema ser transiente. Neste tipo de problema a limitação das variações de pressão e saturação é ditada pelo passo de tempo e a estimativa inicial do método de Newton é a própria solução convergida do passo de tempo anterior. Isso mantém a taxa de convergência sempre próxima do regime assintótico.

**Tabela 6.5 – $\alpha = 45^\circ$, com *backtracking* e método 1
(19 diag., GMRES(25), BILU 0)**

Caso	Iter. Newton	Média	Iter. Solver	Média	Bkt	CPU
$n = 1,5$	3959	3	51265	44	2	33,0
$n = 2,0$	6324	3	58737	33	0	55,4
$n = 2,5$	9323	4	72433	31	3	90,8
$n = 3,0$	11557	4	82788	30	3	118,5
$n = 3,5$	13213	4	88724	29	1	143,2
$n = 4,0$	14657	4	91519	27	5	153,3

**Tabela 6.6 – $\alpha = 45^\circ$, sem *backtracking* e método 1
(19 diag., GMRES(25), BILU 0)**

Caso	Iter. Newton	Média	Iter. Solver	Média	Bkt	CPU
$n = 1,5$	3966	3	51691	44	0	33,2
$n = 2,0$	6324	3	58737	33	0	55,7

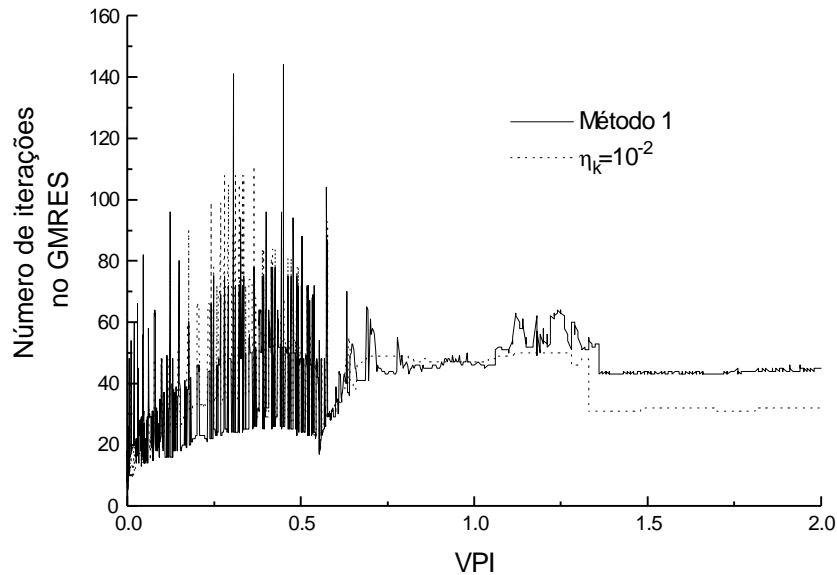
$n=2,5$	9364	4	72503	31	0	92,4
$n=3,0$	11558	4	82906	30	0	120,5
$n=3,5$	13211	4	88718	29	0	148,1
$n=4,0$	14638	4	91424	27	0	160,6

O algoritmo de *backtracking* também foi testado nos métodos que utilizam a matriz Jacobiana aproximada com 7 diagonais, todavia os resultados obtidos foram muito ruins. Neste caso, o problema foi de excesso de *backtracking*, levando a tempos de CPU proibitivos. Analisando o processo de convergência chegou-se a conclusão de que o problema é intrínseco ao algoritmo proposto por Eisenstat e Walker, já que este é embasado teoricamente na linearização exata promovida pela matriz Jacobiana completa.

Nas tabelas 6.7 a 6.15 é feita uma comparação entre os métodos de escolha dos *forcing terms* para vários casos com diferentes ângulos \mathbf{a} e valores de n . Testou-se os dois métodos propostos por Eisenstat e Walker (1996), sendo que, para o método 2, empregou-se $\underline{\epsilon} = 0,9$ e $\mathbf{a} = 1,5$, que provê uma convergência assintótica de ordem- q igual a 1,5. Além disso, testou-se também duas escolhas fixas para os *forcing terms*: $\mathbf{h}_k = 10^{-2}$, que provê apenas uma acurácia moderada na solução da equação linear em cada passo do Newton inexato e $\mathbf{h}_k = 10^{-4}$, que dá uma solução bem mais apurada para a equação linear e uma rápida convergência assintótica próximo à solução.

As tabelas 6.7 a 6.9 mostram os resultados obtidos para o método de Krylov-Newton baseado na matriz Jacobiana completa. Todos empregam o GMRES(25) com BILU 0 na solução da equação linear. Com exceção dos casos onde $n=1,5$, a melhor escolha para os *forcing terms* foi $\mathbf{h}_k = 10^{-2}$. Pela observação do número de iterações newtonianas e do *solver* linear, fica evidente que os outros métodos sofrem com o problema de *oversolving*. Note que, para $\mathbf{h}_k = 10^{-2}$, apesar do maior número de iterações newtonianas, o número de iterações na solução da equação linear é sensivelmente menor. A questão que surge é: por que, neste caso, os métodos adaptativos, que deveriam evitar o *oversolving*, sofrem mais com este problema do que o método fixo? A resposta a esta questão novamente passa pela análise do problema transiente. A Fig. 6.5 mostra a distribuição de iterações no *solver* linear ao longo da simulação para o método 1 e para $\mathbf{h}_k = 10^{-2}$. Inicialmente, até aproximadamente VPI 0,6, há uma

oscilação do número de iterações no *solver*, que depois fica relativamente estabilizada. Principalmente nesta última fase da simulação, o método 1 faz mais iterações no *solver*. O gráfico da Fig. 6.6 apresenta a evolução do passo de tempo ao longo da simulação. Note que a região de oscilação coincide com o período onde há uma maior restrição ao passo de tempo.



**Figura 6.5 – $\alpha = 45^\circ$ e $n = 2,0$, sem *backtracking*
(19 diag., GMRES(25), BILU 0)**

A Fig. 6.7 compara o histórico de convergência em VPI 1,5 (região de elevado passo de tempo) do método adaptativo 1 e $h_k = 10^{-2}$. Optou-se por plotar o log dos resíduos a cada iteração do GMRES, expandindo cada passo do Newton inexatos, de forma a comparar a norma do resíduo não-linear, $\|F\|$, com a do resíduo linear local, $\|F_{x^k} + DF(x^k)s\|$. Note que ambos os métodos apresentam a mesma convergência até 33 iterações do GMRES. Até aí também há plena concordância entre a taxa de convergência dos resíduos linear e não-linear. A partir deste ponto, enquanto o método fixo atinge a convergência, o método adaptativo continua convergindo o resíduo linear, porém avançando muito pouco o resíduo não-linear. Esta diferença entre as curvas caracteriza o fenômeno do *oversolving*.

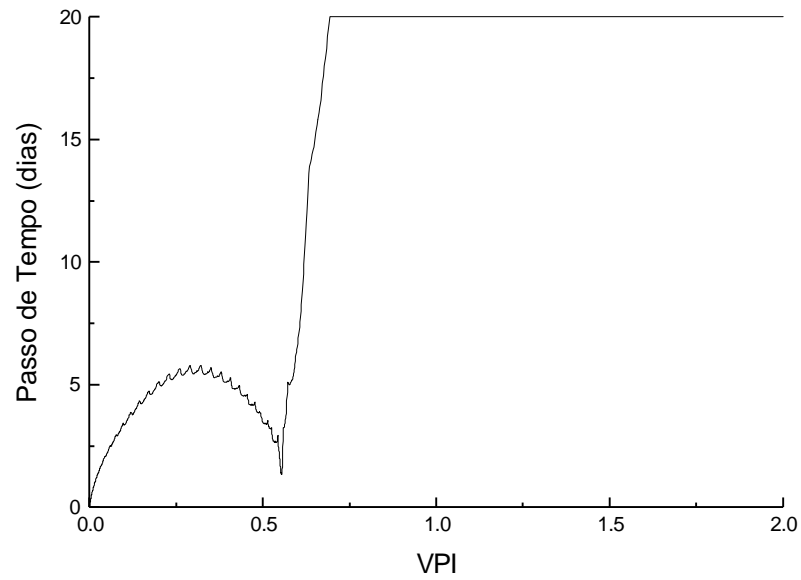


Figura 6.6 – $\alpha = 45^\circ$ e $n = 2,0$, sem *backtracking*
(19 diag., GMRES(25), BILU 0)

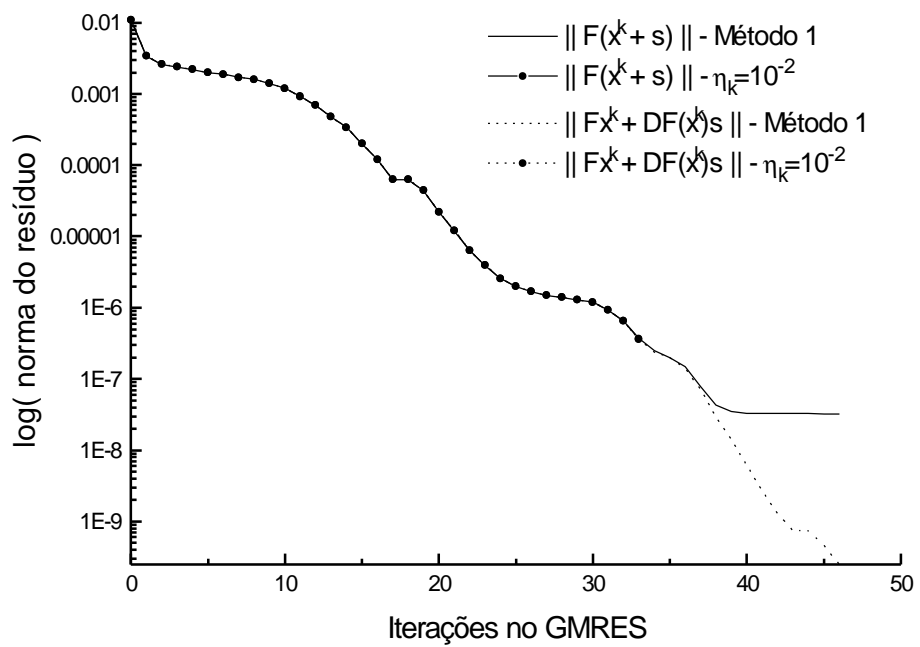


Figura 6.7 – VPI = 1,5, $\alpha = 45^\circ$ e $n = 2,0$, sem *backtracking*
(19 diag., GMRES(25), BILU 0)

A Fig. 6.8 mostra o mesmo gráfico para um VPI de 0,2. Há várias informações interessantes neste gráfico. Uma delas são os picos nas normas do resíduo. Isto ocorre devido a existência de um corte no passo de tempo em virtude de uma variação de saturação ou pressão acima do permitido. Ao detectar variações extremas, o simulador corrige o passo de tempo como mostrado na seção 5.3 e reinicia o método de Newton inexato. Como é ilustrado na figura, em ambos os métodos ocorre o *oversolving*, sendo muito mais pronunciado no método adaptativo. Em compensação, após o *oversolving* este método apresenta uma taxa de convergência muito elevada, atingindo a tolerância em um menor número de iterações, ou seja, neste *time step* o método 1 foi melhor. Este fato dá uma idéia de que em problemas transientes adaptativos o importante é manter uma alta “eficiência média” dos métodos de Newton inexatos ao longo da simulação; o que, na maioria dos testes, foi atingido pelo método fixo com $h_k = 10^{-2}$.

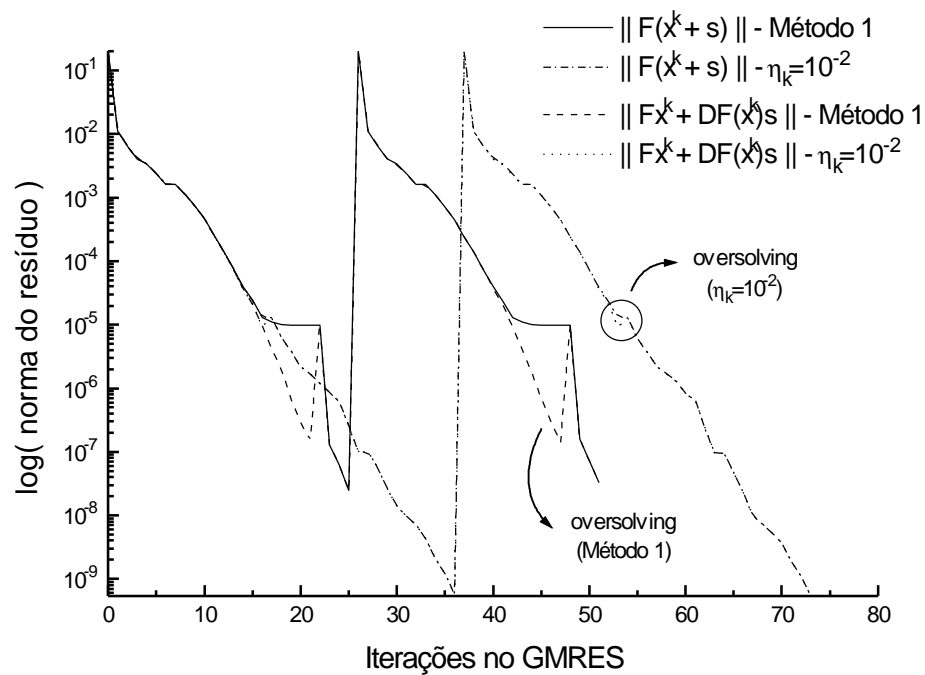


Figura 6.8 – VPI = 0,2, $\alpha = 45^\circ$ e $n = 2,0$, sem *backtracking* (19 diag., GMRES(25), BILU 0)

É interessante observar ainda que, para $n = 1,5$, o método que se sobressai é o método fixo com $h_k = 10^{-4}$. Isso pode ser explicado pela pequena não-linearidade presente nas equações somado ao regime transiente que faz com que a taxa de convergência permaneça no

regime assintótico, privilegiando baixos valores de h_k .

**Tabela 6.7 – $\alpha = 60^\circ$, sem backtracking
(19 diag., GMRES(25), BILU 0)**

h_k	Iter. Newton	Média	Iter. Solver	Média	CPU
$n = 1,5$					
Método 1	3952	3	36943	31	32,4
Método 2	3944	3	42404	36	33,2
$h_k = 10^{-2}$	4233	3	32354	27	32,5
$h_k = 10^{-4}$	3469	3	38909	33	27,8
$n = 2,0$					
Método 1	6496	3	46124	26	60,3
Método 2	6527	3	53778	30	62,1
$h_k = 10^{-2}$	7003	3	43436	24	57,9
$h_k = 10^{-4}$	6072	3	57802	32	59,5
$n = 3,0$					
Método 1	11396	4	58730	21	134,9
Método 2	11392	4	67738	24	137,6
$h_k = 10^{-2}$	12505	4	52951	19	120,7
$h_k = 10^{-4}$	10972	4	77318	28	136,6
$n = 4,0$					
Método 1	14525	4	68482	20	150,6
Método 2	14548	4	74784	22	151,6
$h_k = 10^{-2}$	15868	4	53971	16	150,4
$h_k = 10^{-4}$	14279	4	85017	25	151,9

Tabela 6.8 – $\alpha = 45^\circ$, sem *backtracking*
(19 diag., GMRES(25), BILU 0)

h_k	Iter. Newton	Média	Iter. Solver	Média	CPU
$n = 1,5$					
Método 1	3966	3	51691	44	33,2
Método 2	4017	3	56539	49	34,3
$h_k = 10^{-2}$	4377	3	45376	39	33,4
$h_k = 10^{-4}$	3440	2	51386	44	30,4
$n = 2,0$					
Método 1	6324	3	58737	33	55,7
Método 2	6367	3	64705	36	56,8
$h_k = 10^{-2}$	7034	3	56179	31	52,6
$h_k = 10^{-4}$	6072	3	76858	43	57,5
$n = 3,0$					
Método 1	9364	4	72503	31	92,4
Método 2	9371	4	82107	35	93,8
$h_k = 10^{-2}$	10212	4	67009	29	83,8
$h_k = 10^{-4}$	9004	3	94677	41	94,8
$n = 4,0$					
Método 1	14638	4	91424	27	160,6
Método 2	14819	4	101650	30	164,8
$h_k = 10^{-2}$	15996	4	71867	21	153,3
$h_k = 10^{-4}$	14435	4	111973	33	163,3

Tabela 6.9 – $\alpha = 30^\circ$, sem *backtracking*
(19 diag., GMRES(25), BILU 0)

h_k	Iter. Newton	Média	Iter. Solver	Média	CPU
$n = 1,5$					
Método 1	3958	3	98564	86	42,9
Método 2	4028	3	104166	90	44,2
$h_k = 10^{-2}$	4542	3	76083	66	41,6
$h_k = 10^{-4}$	3471	3	96206	83	39,7
$n = 2,0$					
Método 1	6890	3	128031	72	73,0
Método 2	7028	3	129225	73	72,9
$h_k = 10^{-2}$	8000	4	106571	60	66,4
$h_k = 10^{-4}$	6369	3	134941	76	71,5
$n = 3,0$					
Método 1	11968	4	147659	53	141,7
Método 2	12372	4	158490	57	142,6
$h_k = 10^{-2}$	13562	4	118077	42	124,5
$h_k = 10^{-4}$	11365	4	168858	61	147,4
$n = 4,0$					
Método 1	15524	4	161413	47	195,0
Método 2	16301	4	179960	53	204,7
$h_k = 10^{-2}$	19501	5	133982	39	200,3
$h_k = 10^{-4}$	15175	4	185864	54	196,7

As tabelas 6.10 a 6.15 mostram os resultados dos testes descritos acima para os métodos de Krylov-Newton aproximados. Nestes casos, observou-se o mesmo comportamento analisado anteriormente para o método usando a matriz de 19 diagonais, prevalecendo o método fixo com $h_k = 10^{-2}$. Isto ocorre inclusive para $n = 1,5$, com exceção dos casos

ortogonais ($\mathbf{a} = 90^\circ$), onde todos os métodos empregam a mesma matriz Jacobiana exata. Isto pode ser explicado pela deficiência da aproximação linear existente neste métodos, que faz com que uma escolha mais apertada para \mathbf{h}_k , mesmo com um baixo grau de não-linearidade presente nas equações, introduza muito *oversolving*. Não foi testado, mas é provável que uma escolha ainda mais frouxa traga resultados melhores nestes casos.

**Tabela 6.10 – $\mathbf{a} = 90^\circ$, sem *backtracking*
(7 diag., GMRES(25), BILU 1)**

\mathbf{h}_k	Iter. Newton	Média	Iter. Solver	Média	CPU
$n = 1,5$					
Método 1	3974	3	29270	25	19,5
Método 2	3973	3	31306	26	19,7
$\mathbf{h}_k = 10^{-2}$	4118	3	21605	18	19,2
$\mathbf{h}_k = 10^{-4}$	3482	2	28573	24	17,5
$n = 2,0$					
Método 1	6584	3	40275	22	34,0
Método 2	6584	3	41482	23	34,3
$\mathbf{h}_k = 10^{-2}$	6678	3	28622	16	32,7
$\mathbf{h}_k = 10^{-4}$	6157	3	42062	23	32,8
$n = 3,0$					
Método 1	11285	4	41597	15	72,7
Método 2	11284	4	47769	17	73,8
$\mathbf{h}_k = 10^{-2}$	11284	4	35944	13	69,9
$\mathbf{h}_k = 10^{-4}$	10901	4	54903	20	73,6
$n = 4,0$					
Método 1	14174	4	45923	13	95,8
Método 2	14172	4	51935	15	97,6
$\mathbf{h}_k = 10^{-2}$	14241	4	36339	10	92,2
$\mathbf{h}_k = 10^{-4}$	14023	4	58888	17	98,0

Tabela 6.11 – $\alpha = 45^\circ$, sem *backtracking*
(7 diag., GMRES(25), BILU 1)

h_k	Iter. Newton	Média	Iter. Solver	Média	CPU
$n = 1,5$					
Método 1	82633	71	103941	90	334,3
Método 2	144622	125	162193	140	579,0
$h_k = 10^{-2}$	42452	36	199963	173	187,2
$h_k = 10^{-4}$	42496	36	481446	417	216,0
$n = 2,0$					
Método 1	110615	62	134094	76	421,2
Método 2	197284	112	213763	121	727,3
$h_k = 10^{-2}$	63956	36	244915	139	264,5
$h_k = 10^{-4}$	64561	36	610190	346	309,2

Tabela 6.12 – $\alpha = 30^\circ$, sem *backtracking*
(7 diag., GMRES(25), BILU 1)

h_k	Iter. Newton	Média	Iter. Solver	Média	CPU
$n = 1,5$					
Método 1	173067	151	248132	216	696,7
Método 2	273991	239	339162	295	1051,0
$h_k = 10^{-2}$	87206	76	627626	547	384,3
$h_k = 10^{-4}$	87055	75	1353019	1180	467,0

Tabela 6.13 – $\alpha = 60^\circ$, sem *backtracking*
(Matrix-free, GMRES(25), BILU 2)

h_k	Iter. Newton	Média	Iter. Solver	Média	CPU
$n = 1,5$					
Método 1	15679	13	40625	35	104,7

Método 2	15682	13	40686	35	105,1
$h_k = 10^{-2}$	6574	5	41035	35	69,0
$h_k = 10^{-4}$	4794	4	49030	42	70,0
$n = 2,0$					
Método 1	18179	10	50692	28	123,9
Método 2	18126	10	50958	29	125,1
$h_k = 10^{-2}$	10268	5	53728	30	98,0
$h_k = 10^{-4}$	7956	4	70819	40	104,7
$n = 3,0$					
Método 1	20689	7	63451	23	187,7
Método 2	19471	7	65239	24	183,5
$h_k = 10^{-2}$	15824	5	63255	23	167,5
$h_k = 10^{-4}$	12077	4	87055	32	177,5

Tabela 6.14 – $\alpha = 45^\circ$, sem *backtracking*
(*Matrix-free*, GMRES(25), BILU 2)

h_k	Iter. Newton	Média	Iter. Solver	Média	CPU
$n = 1,5$					
Método 1	20003	17	55810	48	136,6
Método 2	22721	19	59682	51	153,4
$h_k = 10^{-2}$	6600	5	50624	43	80,6
$h_k = 10^{-4}$	4751	4	68934	59	94,3
$n = 2,0$					
Método 1	21323	12	67937	38	154,5
Método 2	22241	12	71277	40	159,3
$h_k = 10^{-2}$	10197	5	67520	38	111,2
$h_k = 10^{-4}$	7896	4	98136	55	134,2

$n = 3,0$					
Método 1	22524	8	82672	30	213,0
Método 2	22832	8	85748	31	219,2
$h_k = 10^{-2}$	16054	5	84717	31	187,7
$h_k = 10^{-4}$	12044	4	114453	41	212,5

Tabela 6.15 – $\alpha = 30^\circ$, sem *backtracking*
(*Matrix-free*, GMRES(25), BILU 2)

h_k	Iter. Newton	Média	Iter. Solver	Média	CPU
$n = 1,5$					
Método 1	23587	20	103389	90	198,7
Método 2	28559	24	113958	99	226,7
$h_k = 10^{-2}$	6808	5	94457	82	124,7
$h_k = 10^{-4}$	4699	4	147827	128	171,7
$n = 2,0$					
Método 1	25235	14	129554	73	225,7
Método 2	27565	15	132188	74	236,7
$h_k = 10^{-2}$	10950	6	124692	70	168,3
$h_k = 10^{-4}$	8115	4	188830	106	222,8

Nas próximas seções, todos os testes utilizam o método fixo com $h_k = 10^{-2}$ sem *backtracking*.

6.2.2 Solvers e Precondicionadores

Nesta seção, é mostrado um estudo realizado com os três métodos de Krylov-Newton onde foi variado o tipo de *solver* de Krylov e o nível do preconditionador BILU. Os *solvers* empregados foram o GMRES com *restart* ou GMRES(r), o TFQMR e o BICGSTAB. Em todos os algoritmos foi utilizado condicionamento à direita e testou-se os níveis 0, 1 e 2

para a fatorização LU incompleta. Analisou-se apenas o caso do paralelepípedo deformado por um ângulo α igual a 45° .

Nas tabelas abaixo são apresentados, para cada caso analisado, da esquerda para a direita, o tempo de CPU em minutos, a porcentagem do tempo de CPU gasto, respectivamente, no *solver*, na montagem da matriz Jacobiana (DF) e na montagem da função resíduo (F), o número total de iterações newtoniana inexatas, seu valor médio e máximo durante a simulação, o número total de iterações no *solver* linear e também seu valor médio e máximo durante a simulação.

As tabelas 6.16 e 6.17 mostram os resultados obtidos com o método que utiliza a matriz Jacobiana completa de 19 diagonais para dois valores do expoente da saturação: 1,5 e 2,0. Observe que, a despeito do maior número de iterações realizadas, o método que emprega o GMRES na solução da equação linear tende a ser mais eficiente, com exceção para $n=1,5$ usando BILU 0 onde a simulação foi mais rápida com o uso do BICGSTAB. Entretanto, pode ser decretado um empate técnico entre os métodos, já que a maior diferença foi de aproximadamente 7%. Outro ponto importante a ser observado é a ineficiência do aumento do nível de aproximação da fatorização LU. Para ambos os valores de n testados, foram obtidos melhores resultados com o nível 0.

Tabela 6.16 – $\alpha = 45^\circ$ e $n = 1,5$ (19 diag.)

<i>Solver</i>	CPU**	<i>Solver</i> *	DF*	F*	Iter. Newton	Média ⁺ / Máx.	Iter. Solver	Média ⁺ / Máx.
BILU 0								
GMRES(30)	34,1	33	52	14	4312	3 / 8	46498	40 / 171
TFQMR	33,5	39	47	13	3768	3 / 37	33972	29 / 111
BICGSTAB	32,7	35	51	13	3948	3 / 37	26486	22 / 81
BILU 1								
GMRES(30)	41,5	48	40	11	4001	3 / 38	21871	18 / 81
TFQMR	43,2	52	37	10	3803	3 / 38	17092	14 / 48
BICGSTAB	41,9	50	39	10	3907	3 / 37	13530	11 / 42
BILU 2								
GMRES(30)	81,5	73	21	6	4086	3 / 37	16368	14 / 57
TFQMR	83,1	75	20	5	3880	3 / 37	12725	11 / 45
BICGSTAB	83,0	75	20	5	3832	3 / 37	10363	8 / 37

** tempo de CPU em min.

* % do tempo de CPU

+ total/nº de *time steps* realizados

Tabela 6.17 – $a = 45^\circ$ e $n = 2,0$ (19 diag.)

<i>Solver</i>	CPU**	<i>Solver</i> *	DF*	F*	Iter. Newton	Média ⁺ / Máx.	Iter. Solver	Média ⁺ / Máx.
BILU 0								
GMRES(30)	56,9	34	51	13	7010	3 / 38	61482	34 / 126
TFQMR	57,9	40	47	12	6481	3 / 37	44388	25 / 83
BICGSTAB	57,1	38	49	12	6581	3 / 37	35390	20 / 63
BILU 1								
GMRES(30)	78,7	54	36	9	6631	3 / 38	29755	16 / 57
TFQMR	84,2	58	33	8	6512	3 / 38	23070	13 / 38
BICGSTAB	80,7	56	34	9	6458	3 / 37	17934	10 / 37
BILU 2								
GMRES(30)	170,0	79	16	4	6446	3 / 37	20724	11 / 42
TFQMR	180,5	80	15	4	6609	3 / 37	17530	9 / 37
BICGSTAB	180,5	80	15	4	6447	3 / 37	14100	8 / 37

** tempo de CPU em min.

* % do tempo de CPU

+ total/nº de *time steps* realizados

As tabelas 6.18 e 6.19 mostram os resultados para o método aproximado proposto por Maliska *et al.* (1997), onde a matriz Jacobiana é aproximada apenas pelas 7 diagonais originadas dos termos ortogonais. Neste caso os resultados foram muito parecidos com os anteriores; pouca diferença entre o *solvers*, com uma leve vantagem do GMRES. Todavia, o melhor condicionamento foi obtido com o BILU de nível 1.

Tabela 6.18 – $a = 45^\circ$ e $n = 1,5$ (7 diag.)

<i>Solver</i>	CPU**	<i>Solver</i> *	DF*	F*	Iter. Newton	Média ⁺ / Máx.	Iter. Solver	Média ⁺ / Máx.
BILU 0								
GMRES(30)	262,8	44	41	14	42616	36 / 163	1123566	975 / 3561
TFQMR	258,2	42	43	14	43541	37 / 163	780456	677 / 2468
BICGSTAB	249,2	40	44	14	42633	37 / 163	717284	622 / 2961
BILU 1								
GMRES(30)	191,0	23	57	19	42502	36 / 163	335145	290 / 1095
TFQMR	204,2	28	54	17	42683	37 / 163	256759	222 / 852
BICGSTAB	201,5	27	55	17	42511	36 / 163	231995	201 / 897
BILU 2								
GMRES(30)	212,7	31	51	17	42528	36 / 163	53080	263 / 957
TFQMR	229,0	35	48	16	42626	37 / 163	226022	196 / 759
BICGSTAB	226,7	34	49	16	42496	36 / 163	207343	179 / 768

** tempo de CPU em min.

* % do tempo de CPU

+ total/nº de *time steps* realizados

Tabela 6.19 – $\alpha = 45^\circ$ e $n = 2,0$ (7 diag.)

<i>Solver</i>	CPU**	<i>Solver</i> *	DF*	F*	Iter. Newton	Média ⁺ / Máx.	Iter. Solver	Média ⁺ / Máx.
BILU 0								
GMRES(30)	372,3	44	42	14	64680	36 / 163	1358956	772 / 2535
TFQMR	360,0	40	44	15	66122	37 / 163	903275	513 / 1450
BICGSTAB	363,2	42	43	14	64558	36 / 163	917952	521 / 2151
BILU 1								
GMRES(30)	288,8	26	55	18	64550	36 / 162	423153	240 / 810
TFQMR	305,3	30	52	17	64715	36 / 162	297969	169 / 558
BICGSTAB	298,0	28	53	18	64598	36 / 162	276550	157 / 579
BILU 2								
GMRES(30)	327,2	36	48	15	64539	36 / 162	373697	212 / 708
TFQMR	348,2	39	45	15	64652	36 / 162	270354	153 / 480
BICGSTAB	344,7	38	46	15	64575	36 / 162	243068	138 / 496

** tempo de CPU em min.

* % do tempo de CPU

+ total/nº de *time steps* realizados

Nas tabelas 6.20 e 6.21 estão dispostos os resultados do método *matrix-free*. Uma característica evidente é a necessidade de um bom condicionamento, visto que para BILU 0 os resultados foram muito pobres, não sendo possível a convergência para o TFQMR e o BICGSTAB. Para $n = 1,5$, chegou-se a obter resultados melhores com nível 3 de fatorização incompleta. Outro fator importante, é a notável predominância do GMRES perante os outros *solvers*, chegando a ser, para $n = 1,5$, 80% e 30% mais rápido que o TFQMR e o BICGSTAB, respectivamente. Note que os resultados utilizando o TFQMR foram muito ruins neste método.

Tabela 6.20 – $\alpha = 45^\circ$ e $n = 1,5$ (*Matrix-free*)

<i>Solver</i>	CPU**	<i>Solver</i> *	DF*	F*	Iter. Newton	Média ⁺ / Máx.	Iter. Solver	Média ⁺ / Máx.
BILU 0								
GMRES(30)	326,0	93	5	86	5866	5 / 41	314404	272 / 908
TFQMR	Falha							
BICGSTAB	Falha							
BILU 1								
GMRES(30)	77,1	76	17	72	4945	4 / 39	59993	52 / 187
TFQMR	442,0	94	4	87	6174	5 / 39	223442	193 / 2153
BICGSTAB	116,3	84	11	80	4943	4 / 39	51707	44 / 293
BILU 2								
GMRES(30)	74,7	75	17	67	4953	4 / 40	53076	46 / 161
TFQMR	355,5	94	4	84	5768	5 / 39	170477	147 / 1138
BICGSTAB	101,9	82	12	73	4853	4 / 39	40828	35 / 193
BILU 3								
GMRES(30)	73,9	75	17	54	4992	4 / 40	40601	35 / 119

** tempo de CPU em min.

* % do tempo de CPU

+ total/nº de *time steps* realizados

Tabela 6.21 – $\alpha = 45^\circ$ e $n = 2,0$ (Matrix-free)

<i>Solver</i>	CPU**	<i>Solver</i> *	DF*	F*	Iter. Newton	Média ⁺ / Máx.	Iter. Solver	Média ⁺ / Máx.
BILU 0								
GMRES(30)	324,3	90	7	83	8803	5 / 41	327776	186 / 547
TFQMR	Falha							
BICGSTAB	Falha							
BILU 1								
GMRES(30)	111,0	73	19	68	8295	4 / 40	85141	48 / 145
TFQMR	384,5	92	6	84	8885	5 / 38	199640	113 / 945
BICGSTAB	142,8	80	14	74	8147	4 / 37	62835	35 / 227
BILU 2								
GMRES(30)	111,6	73	19	60	8407	4 / 41	73840	41 / 120
TFQMR	339,3	91	79	6	8759	4 / 38	163887	93 / 831
BICGSTAB	132,3	78	15	67	8133	4 / 36	51147	29 / 110
BILU 3								
GMRES(30)	124,7	76	17	44	8378	4 / 41	56846	32 / 120

** tempo de CPU em min.

* % do tempo de CPU

+ total/n^o de *time steps* realizados

Em muitos casos, principalmente quando há a necessidade de uma malha muito refinada, um fator decisivo na viabilidade da simulação é a memória utilizada pelo código computacional. Muitas vezes o analisador numérico se vê obrigado a preferir um código mais rápido a outro mais lento, porém mais econômico no uso da memória. Neste aspecto, como é mostrado na tabela 6.22, o aumento do nível de fatorização do BILU exige uma maior disponibilidade de memória livre. Este requisito torna-se ainda mais importante quando a matriz Jacobiana é exata, já que para um mesmo nível de condicionamento ocupa de 2,3 a 3,3 vezes mais memória do que o ocupado para uma matriz aproximada.

Tabela 6.22 – Memória consumida pelos preconditionadores (em Kbytes) em função da estrutura da matriz Jacobiana

Precondicionador	Matriz Jacobiana	
	19 diagonais	7 diagonais
BILU 0	667,52	284,48
BILU 1	1.390,43	516,64
BILU 2	2.645,54	807,37
BILU 3	x	1.363,07

A tabela 6.23 mostra a divisão do tempo de processamento do problema linear entre as

iterações do *solver* de Krylov e a montagem do preconditionador. Devido ao maior número de não-nulos da matriz de 19 diagonais, uma maior quantidade do tempo de solução do problema linear é gasto na montagem do preconditionador. Daí a ineficácia do aumento da fatorização LU para este método. Deve-se observar ainda a relativamente elevada parcela do tempo gasta no *solver* para o método *matrix-free*. Isto ocorre devido à aproximação do produto Jacobiana-vetor ter de ser calculada a cada iteração do *solver* de Krylov.

Tabela 6.23 – Distribuição do tempo de processamento do problema linear em % do tempo total de CPU (GMRES, $\alpha = 45^\circ$ e $n = 1,5$)

Precondicionador	Krylov-Newton					
	19 diagonais		7 diagonais		<i>Matrix-free</i>	
	<i>Solver</i>	Prec.	<i>Solver</i>	Prec.	<i>Solver</i>	Prec.
BILU 0	23	10	42	2	93	0
BILU 1	17	31	17	6	74	2
BILU 2	13	60	18	13	70	5

6.2.3 Avaliação dos métodos de Krylov-Newton

Esta seção é dedicada à comparação direta entre os três métodos de Krylov-Newton apresentados anteriormente. Os parâmetros variados são o ângulo de deformação α e o expoente das saturações n . O primeiro está ligado diretamente com a magnitude dos termos não-ortogonais das equações discretizadas; quanto menor o valor de α , maior é o peso das derivadas cruzadas e conseqüentemente maiores são os elementos das diagonais extras da matriz Jacobiana completa. Já o expoente n é a principal fonte de não-linearidade das equações; quanto maior for o valor de n , maior é o grau de não-linearidade das equações discretizadas.

Nos resultados detalhados na tabela **Error! Reference source not found.**, para cada método de Krylov-Newton, foram usados os melhores parâmetros segundo os estudos apresentados nas seções anteriores. Estes parâmetros estão discriminados na tabela 6.24.

Tabela 6.24 – Parâmetros numéricos usados em cada método de Krylov-Newton

	19 diag.	7 diag.	Matrix-free
Solver	GMRES(25)	GMRES(25)	GMRES(25)
Precondicionador	BILU 0, à direita	BILU 1, à direita	BILU 2, à direita
Backtracking	não	não	não
h_k	fixo, 10^{-2}	fixo, 10^{-2}	fixo, 10^{-2}
Memória em Kbytes*	1488,19	842,44	836,51

* estimativa que leva em conta apenas as estruturas mais representativas, tais como: matriz Jacobiana e de condicionamento, vetores F e s, etc.

Como era de se esperar, o efeito da não-ortogonalidade sobre os métodos de Krylov-Newton que utilizam a matriz Jacobiana aproximada (apenas 7 diagonais) é grande. Note que mesmo para deformações muito pequenas, $\alpha = 80^\circ$, obteve-se melhores resultados com o método que utiliza a matriz Jacobiana de 19 diagonais. Isto ocorre devido a este método possuir um processo iterativo muito mais eficiente, evidenciado através do sensível menor número de iterações newtonianas e do *solver* linear.

É importante ter em mente que o custo por iteração, tanto da iteração do Newton inexato, quanto da iteração do sistema linear interno, é maior para o método que utiliza a matriz Jacobiana completa comparado ao método sugerido por Maliska *et al.* (1997), já que esta é muito menos esparsa do que a matriz aproximada. No entanto, esta relação se inverte para o método *matrix-free*. Na verdade, o método *matrix-free* possui o maior custo por iteração dentre todos os métodos aqui apresentados. Isto ocorre devido à aproximação do produto Jacobiana-vetor ser calculada a cada iteração do *solver* linear. Todavia, para ângulos de 60° a 30° , este método chega a ser de 50% a quase 4 vezes mais rápido do que o método de Maliska *et al.*, sendo um método bastante atrativo quando a memória disponível é restrita.

Tabela 6.25 – Comparação entre os métodos de Krylov-Newton para vários valores de α e n

α	Krylov-Newton	CPU**	<i>Solver</i> *	DF*	F*	Iter. Newton	Méd.+ / Máx.	Iter. Solver	Méd.+ / Máx.
90°	$n = 1,5$								
	19 diag.	25,7	21	61	17	4032	3 / 35	23229	19 / 75
	7 diag.	19,2	19	56	24	4118	3 / 35	21605	18 / 78
	<i>Matrix-free</i>	47,3	62	26	63	4846	4 / 35	28939	24 / 79
	$n = 2,0$								
	19 diag.	45,1	23	59	16	6731	3 / 35	31910	18 / 57
7 diag.	32,7	23	53	22	6678	3 / 34	28622	16 / 48	

	<i>Matrix-free</i>	63,9	57	30	57	8133	4 / 35	40066	22 / 69	
	<i>n</i> = 3,0									
	19 diag.	97,4	33	51	15	11284	4 / 35	35944	13 / 40	
	7 diag.	69,9	31	48	21	11284	4 / 35	35944	13 / 40	
	<i>Matrix-free</i>	155,0	63	25	35	15132	5 / 34	46299	17 / 56	
80°	<i>n</i> = 1,5									
	19 diag.	27,4	25	58	15	4039	3 / 36	22993	19 / 78	
	7 diag.	40,1	18	59	22	9599	8 / 66	34016	29 / 93	
	<i>Matrix-free</i>	60,4	63	26	57	6469	5 / 35	35951	30 / 90	
	<i>n</i> = 2,0									
	19 diag.	51,0	28	57	14	6655	3 / 36	29696	16 / 54	
	7 diag.	65,2	21	57	20	15392	8 / 66	45675	25 / 69	
	<i>Matrix-free</i>	83,4	61	28	50	9921	5 / 35	44329	25 / 78	
	<i>n</i> = 3,0									
	19 diag.	99,1	33	52	14	11090	4 / 36	37091	13 / 46	
	7 diag.	124,75	26	53	20	25261	9 / 66	56866	20 / 68	
	<i>Matrix-free</i>	131,4	62	26	40	13075	4 / 35	45887	16 / 70	
	60°	<i>n</i> = 1,5								
		19 diag.	32,5	27	56	15	4233	3 / 39	32354	27 / 102
		7 diag.	102,1	23	57	20	23087	19 / 116	158086	136 / 475
<i>Matrix-free</i>		69,0	66	24	60	6574	5 / 39	41035	35 / 113	
<i>n</i> = 2,0										
19 diag.		57,9	32	54	13	7003	3 / 39	43436	24 / 81	
7 diag.		152,1	24	56	19	35341	20 / 116	192193	109 / 324	
<i>Matrix-free</i>		98,0	63	26	54	10268	5 / 39	53728	30 / 96	
<i>n</i> = 3,0										
19 diag.		120,7	37	49	13	12505	4 / 40	52951	19 / 67	
7 diag.		263,2	24	55	19	56060	20 / 116	125683	46 / 118	
<i>Matrix-free</i>		167,5	63	26	44	15824	5 / 40	63255	23 / 98	
45°		<i>n</i> = 1,5								
		19 diag.	33,4	32	53	14	4377	3 / 39	45376	39 / 144
		7 diag.	187,2	17	62	20	42452	36 / 162	199963	173 / 567
	<i>Matrix-free</i>	80,6	70	21	62	6600	5 / 39	50624	43 / 143	
	<i>n</i> = 2,0									
	19 diag.	52,6	32	53	14	7034	3 / 39	56179	31 / 111	
	7 diag.	264,5	20	60	20	63956	36 / 162	244915	139 / 408	
	<i>Matrix-free</i>	111,2	68	23	56	10197	5 / 39	67520	38 / 117	
	<i>n</i> = 3,0									
	19 diag.	83,8	36	50	14	10212	4 / 39	67009	29 / 81	
	7 diag.	449,2	25	55	19	96518	35 / 162	259803	95 / 261	
	<i>Matrix-free</i>	187,7	67	23	48	16054	5 / 40	84717	31 / 119	
	30°	<i>n</i> = 1,5								
		19 diag.	41,6	39	48	13	4542	3 / 42	76083	66 / 252
		7 diag.	384,3	22	59	19	87206	76 / 267	627626	547 / 1770
<i>Matrix-free</i>		124,7	80	14	70	6808	5 / 43	94457	82 / 257	
<i>n</i> = 2,0										
19 diag.		66,4	39	47	13	8000	4 / 42	106571	60 / 192	
7 diag.		632,6	31	51	17	134354	75 / 246	1330197	752 / 2811	
<i>Matrix-free</i>		168,3	77	16	64	10950	6 / 42	124692	70 / 199	
<i>n</i> = 3,0										
19 diag.		124,5	39	47	13	13562	4 / 42	118077	42 / 132	

7 diag.	881,3	28	53	18	183697	66 / 246	782794	283 / 801
Matrix-free	245,8	74	18	54	17024	6 / 43	140166	50 / 150

** tempo de CPU em min.

* % do tempo de CPU

+ total/nº de *time steps* realizados

Outro comportamento interessante de ser observado, é a mínima variação dos valores médios e máximos das iterações newtonianas com relação à deformação da malha para o método que emprega a matriz de 19 diagonais e o método *matrix-free* (ver Fig. 6.9). Isso demonstra a robustez inerente a uma boa aproximação da matriz Jacobiana. É bom lembrar que, apesar da matriz Jacobiana ser “discreta” no método *matrix-free*, sua aproximação é muito melhor do que a realizada no método de Maliska *et al.*. A aproximação do método *matrix-free*, de certa forma, leva em consideração todas as 19 diagonais, usando a matriz com as 7 diagonais ortogonais apenas para condicionamento.

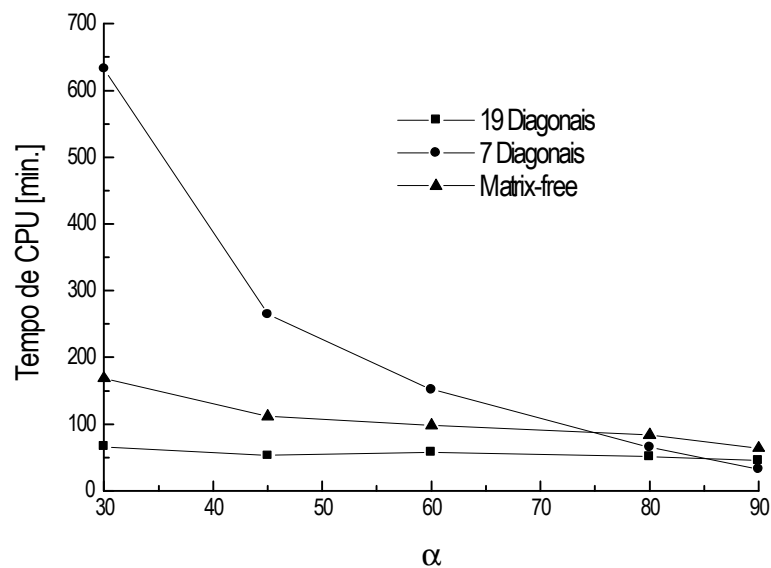


Figura 6.9 – Influência da não-ortogonalidade nos métodos de Krylov-Newton para $n = 2,0$

Com relação a não-linearidade introduzida pelo expoente das saturações, n , é interessante observar que todos os métodos apresentam uma queda de performance com o aumento do valor de n e, quanto maior a aproximação da matriz Jacobiana promovida pelo método, maior pronunciado torna-se este efeito (ver Fig. 6.10).

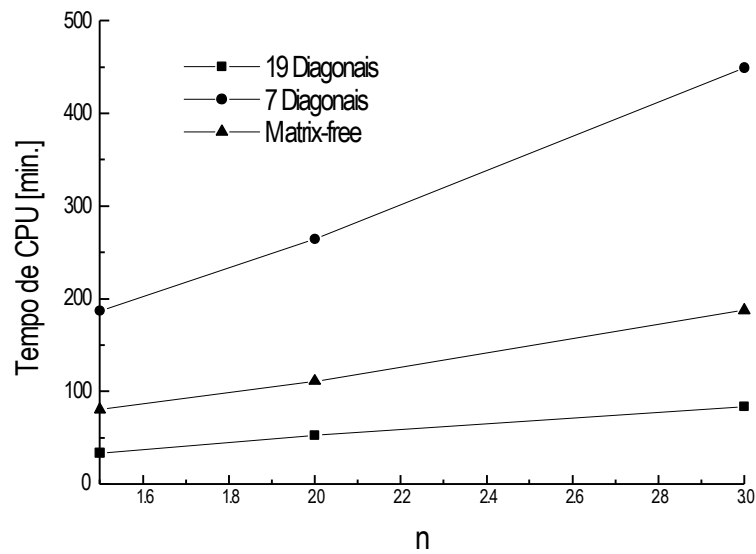


Figura 6.10 – Influência da não-linearidade nos métodos de Krylov-Newton para $\alpha = 45^\circ$

As tabelas 6.26, 6.27 e 6.28 quantificam a eficiência relativa de cada método com relação ao tempo CPU ilustrando a análise feita acima.

Nestas tabelas é ainda possível observar um comportamento interessante com relação a não-linearidade. Com o aumento do valor do expoente n , há em geral uma pequena queda na diferença entre os métodos aproximados e o método que emprega a matriz completa. Isso pode ocorrer devido a um aumento mais significativo nos valores das 7 diagonais “ortogonais” e menor nas outras 12.

Tabela 6.26 – Razão $\text{CPU}^7 / \text{CPU}^{19}$

	90°	80°	60°	45°	30°
$n = 1,5$	0,75	1,5	3,1	5,6	9,2
$n = 2,0$	0,73	1,3	2,6	5,0	9,5
$n = 3,0$	0,72	1,3	2,2	5,4	7,1

Tabela 6.27 – Razão $\text{CPU}^7 / \text{CPU}^{\text{mf}}$

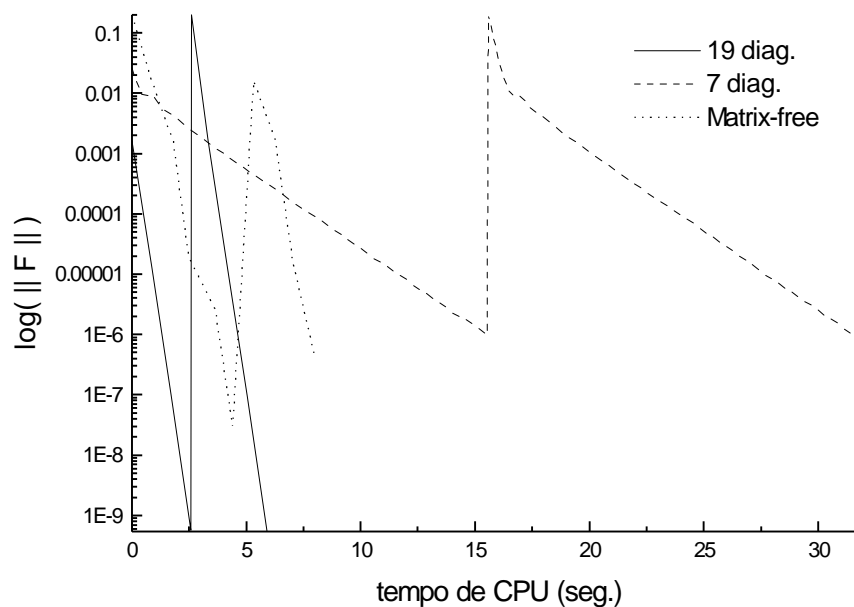
	90°	80°	60°	45°	30°
$n = 1,5$	0,41	0,66	1,5	2,3	3,1
$n = 2,0$	0,51	0,78	1,6	2,4	3,8
$n = 3,0$	0,45	0,95	1,6	2,4	3,6

Tabela 6.28 – Razão $\text{CPU}^{\text{mf}} / \text{CPU}^{19}$

	90°	80°	60°	45°	30°
$n = 1,5$	1,8	2,2	2,1	2,4	3,0
$n = 2,0$	1,4	1,6	1,7	2,1	2,5
$n = 3,0$	1,6	1,3	1,4	2,2	2,0

Os gráficos das figuras abaixo apresentam, para diferentes valores do VPI, a norma da função resíduo em função do tempo de CPU para o caso com $\mathbf{a} = 45^\circ$ e $n = 2,0$. Nestas curvas pode ser vista a forte convergência do método com a matriz de 19 diagonais, seguido pelo método *matrix-free*; ambos muito mais rápidos do que método que desconsidera os termos não-ortogonais. As oscilações nas curvas dos métodos aproximados para VPI igual a 1,0 e 1,5 evidenciam a persistência do problema de *oversolving* presente nos métodos quase-newtonianos.

É importante salientar, que os resultados aqui apresentados, foram obtidos com tolerâncias moderadamente restritivas, aumentando ainda mais a vantagem do método que utiliza a matriz Jacobiana exata.

Figura 6.11 – VPI = 0,2 para $\mathbf{a} = 45^\circ$ e $n = 2,0$

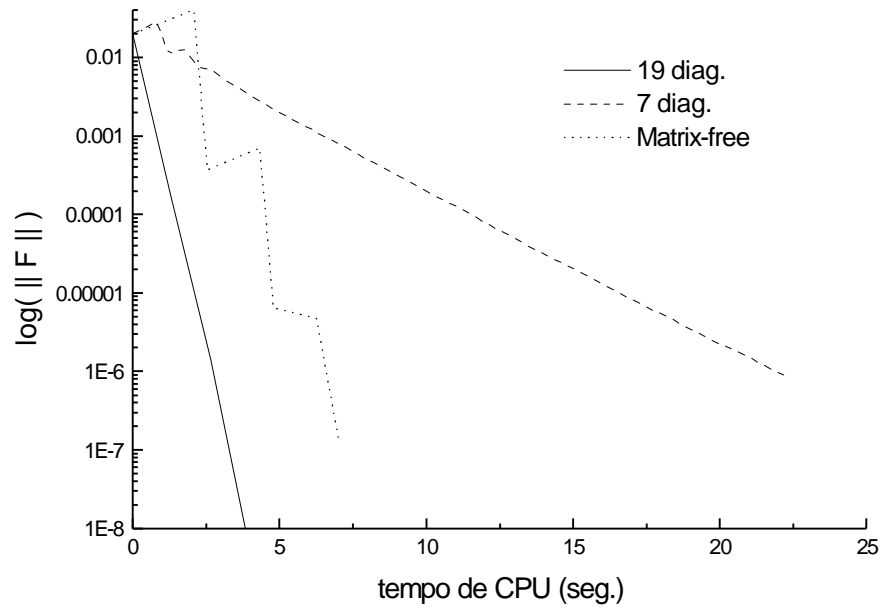


Figura 6.12 – $VPI = 1,0$ para $a = 45^\circ$ e $n = 2,0$

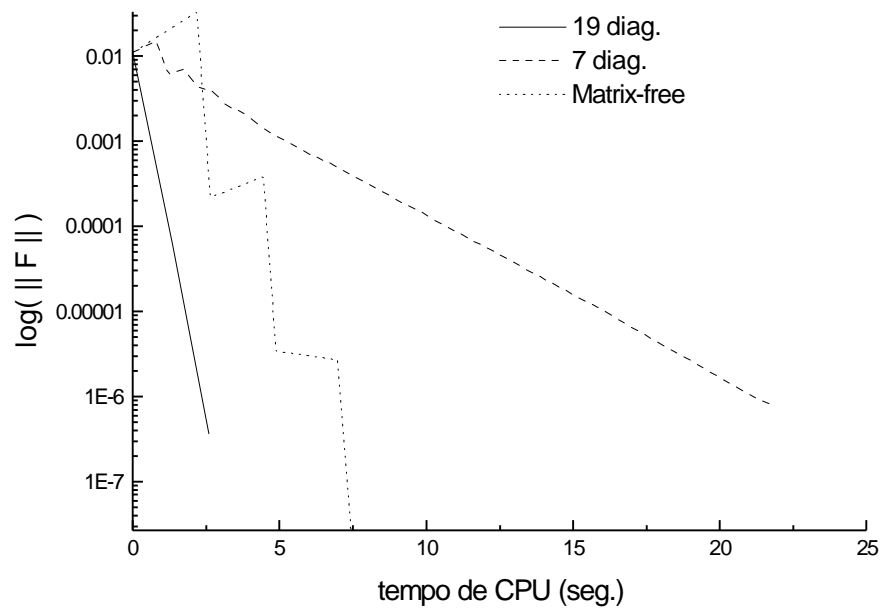


Figura 6.13 – $VPI = 1,5$ para $a = 45^\circ$ e $n = 2,0$

6.3 Reservatório Real

Nos problemas da seção anterior, a não-ortogonalidade da malha tem uma distribuição homogênea, o que, na prática, não ocorre em reservatórios mais realistas. Nesta seção, serão apresentados os resultados da aplicação dos três métodos de Krylov-Newton na modelagem de um escoamento envolvendo os mesmos tipos de fluidos dos problemas anteriores, porém em um reservatório com geometria mais próxima a um caso real.

A Fig. 6.14 apresenta a malha generalizada utilizada na simulação. Observe que há uma distribuição razoavelmente heterogênea da não-ortogonalidade, sendo um pouco mais acentuada ao longos dos eixos x e h . Como é mostrado na figura, há dois poços injetores de água e dois poços produtores dispostos em cantos opostos do reservatório. As vazões de injeção e produção são constantes e iguais a 1920 barris/dia ($3,53 \times 10^{-3} \text{ m}^3/\text{s}$) em cada poço. O volume total do reservatório é de $7,55 \times 10^8 \text{ m}^3$ e a simulação é realizada até 2 VPI. Os demais dados do problema físico e numérico são idênticos aos fornecidos nas tabelas 6.3 e 6.4 para o problema do paralelepípedo deformado.

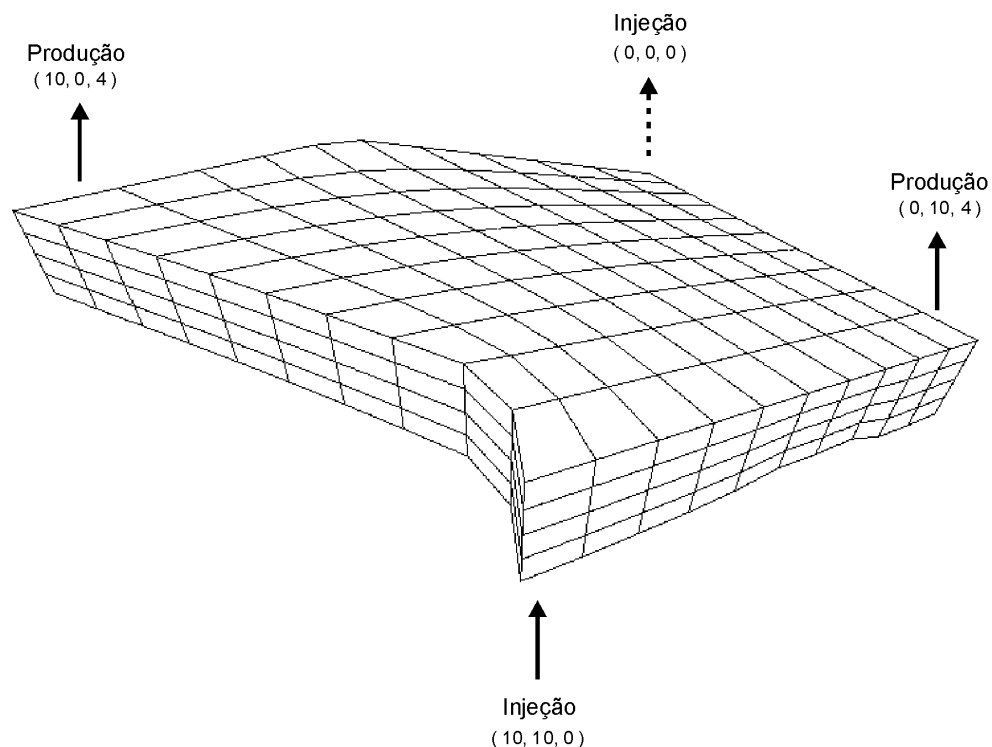


Figura 6.14 – Malha $10 \times 10 \times 4$ (x , h , z) e disposição dos poços

Na tabela 6.29 são apresentados os parâmetros numéricos utilizados em cada um dos

métodos de Newton inexatos.

Tabela 6.29 – Parâmetros numéricos usados em cada método de Krylov-Newton

	19 diag.	7 diag.	Matrix-free
Solver	GMRES(25)	GMRES(25)	GMRES(25)
Precondicionador	BILU 0, à direita	BILU 1, à direita	BILU 2, à direita
Backtracking	não	não	não
h_k	fixo, 10^{-2}	fixo, 10^{-2}	fixo, 10^{-2}
Memória em Kbytes *	527,98	306,02	289,46

* estimativa que leva em conta apenas as estruturas mais representativas, tais como: matriz Jacobiana e de condicionamento, vetores F e s, etc.

A tabela 6.30 mostra os resultados obtidos com três diferentes valores do expoente das saturações. Nas três condições, o método que emprega a Jacobiana completa, com 19 diagonais, teve a melhor performance, seguido pelo método proposto por Maliska *et al.* (1997).

Tabela 6.30 – Comparação entre os métodos de Krylov-Newton na simulação de um reservatório real

Krylov-Newton	CPU**	Solver*	DF*	F*	Iter. Newton	Méd. + / Máx.	Iter. Solver	Méd. + / Máx.
<i>n</i> = 1,5								
19 diag.	146,5	24	60	15	72052	3 / 4	568307	23 / 26
7 diag.	178,0	23	57	19	130621	5 / 8	939448	39 / 44
Matrix-free	222,0	64	25	61	71743	3 / 4	513096	21 / 24
<i>n</i> = 2,0								
19 diag.	124,6	24	60	14	64595	2 / 6	470680	19 / 27
7 diag.	167,7	23	57	19	134989	5 / 10	893009	37 / 44
Matrix-free	200,0	64	25	59	72048	3 / 6	495430	20 / 23
<i>n</i> = 3,0								
19 diag.	139,3	23	61	15	68605	2 / 9	446897	18 / 42
7 diag.	163,6	19	59	20	122917	5 / 18	681750	28 / 60
Matrix-free	188,7	60	28	58	67284	2 / 9	398052	16 / 39

** tempo de CPU em min.

* % do tempo de CPU

+ total/*n*° de *time steps* realizados

A tabela 6.31 resume a análise aos tempos de CPU, mostrando que o método com 19 diagonais é em média 20% mais rápido do que o método com a matriz de 7 diagonais. Outro

ponto interessante é a baixa performance do método *matrix-free*, apesar do relativamente baixo número de iterações lineares e não-lineares realizadas. Atribui-se este fato ao custo por iteração mais elevado deste método.

Tabela 6.31 – Razão entre tempos de CPU

	$\text{CPU}^7 / \text{CPU}^{19}$	$\text{CPU}^7 / \text{CPU}^{\text{mf}}$	$\text{CPU}^{\text{mf}} / \text{CPU}^{19}$
$n = 1,5$	1,2	0,80	1,5
$n = 2,0$	1,3	0,84	1,2
$n = 3,0$	1,2	0,87	1,4

7 Conclusões e Recomendações

7.1 Conclusões

Ao longo deste trabalho foram apresentados e analisados três classes distintas de métodos de Krylov-Newton para a solução do sistema não-linear, oriundo da discretização em volumes finitos, usando coordenadas generalizadas, das equações do modelo *Black-Oil* padrão. A primeira dessas classes, que seria a implementação padrão, utiliza a matriz Jacobiana completa com as 19 diagonais referentes à discretização não-ortogonal em três dimensões. Já as outras duas classes de métodos de Krylov-Newton baseiam-se em uma aproximação da matriz Jacobiana, em que apenas os termos ortogonais das equações são utilizados na construção da matriz. Em uma delas apenas desconsidera-se os termos não-ortogonais, dando origem a um método quase-newtoniano, onde a linearização passa a ser aproximada através de uma matriz com 7 diagonais. Este método foi reportado no trabalho de Maliska *et al.* (1997). A terceira classe, introduzida neste trabalho, é uma versão *matrix-free*, onde a matriz Jacobiana não é calculada diretamente, mas através de uma aproximação em diferenças finitas do produto matriz-vetor usado nos *solvers* de Krylov. Neste caso, a linearização, apesar de ainda ser aproximada, leva em conta todas as 19 diagonais. Todavia, neste trabalho, foi proposto manter a montagem da matriz Jacobiana aproximada de 7 diagonais para servir de matriz de condicionamento.

Neste trabalho foram ainda apresentados alguns conceitos e procedimentos importantes, introduzidos por Eisenstat e Walker (1994, 1996) para os métodos de Newton inexatos, tais como: *oversolving*, *backtracking*, *forcing terms* e métodos adaptativos para a escolha dos *forcing terms*. É importante ressaltar que atualmente os métodos de Newton inexatos, sustentados pelo crescente avanço de conhecimento, tanto teórico, quanto no desenvolvimento de algoritmos, vêm ganhando cada vez mais adeptos motivados pela sua excelente performance na solução de complexos sistemas de equações não-lineares. Os trabalhos de Zingg e seus alunos (e Pueyo, 1997, 1998; e Blanco, 1997) já mostram métodos de Krylov-Newton, aplicados à solução simultânea das equações de Navier-Stokes, obtendo resultados melhores do que alguns algoritmos consagrados que utilizam métodos *multigrid*. Como se vê,

esta é uma área em expansão no campo da análise numérica e muita contribuição ainda pode ser feita.

As principais conclusões são listadas abaixo:

- Em problemas transientes, tais como o da extração de petróleo estudado aqui, o procedimento de *backtracking* é pouco efetivo, já que a convergência está sempre próxima do regime assintótico, fazendo com que a condição necessária ao início do *backtracking* seja raramente atendida.
- Apesar de muito requisitado, o algoritmo de *backtracking* sugerido por Eisenstat e Walker (1994) mostrou-se ineficaz para os métodos que utilizam a matriz Jacobiana de 7 diagonais.
- Os métodos adaptativos para a escolha dos *forcing terms* sugeridos por Eisenstat e Walker (1996) apresentaram resultados inexpressivos se comparados com o método fixo com $h_k = 10^{-2}$; que, para o método que emprega a Jacobiana completa, praticamente não causa *oversolving*.
- Os métodos de Krylov-Newton que utilizam a matriz Jacobiana aproximada sofrem graves problemas de *oversolving*.
- Dos métodos iterativos de Krylov testados (GMRES(r), TFQMR e BICGSTAB), o que apresentou, em geral, melhores resultados foi o GMRES com *restart*, principalmente quando empregado no método *matrix-free*.
- Com relação ao condicionamento utilizando o algoritmo baseado na fatorização LU incompleta de nível n especializado para uma matriz de bloco, BILU(n), diferentes níveis mostraram-se mais eficientes para cada método de Newton inexato. Para o método baseado na matriz Jacobiana exata, a fatorização de nível 0 obteve melhores resultados. Já para o método sugerido por Maliska *et al.*, sobressaiu-se o BILU de nível 1. E, para o método *matrix-free*, foi necessário utilizar um maior condicionamento, de nível 2 a 3.
- Para níveis do BILU de 0 a 2, o método *matrix-free* ocupa de 2 a 1,4 vezes menos memória do que o método com a matriz de 7 diagonais e de 5 a 4,2 vezes menos do que o método com a matriz de 19 diagonais.
- Para os métodos que utilizam a matriz Jacobiana aproximada, foi observada uma

forte dependência da não-ortogonalidade da malha. Já o método que emprega a matriz de 19 diagonais mostrou-se muito mais robusto e pouco suscetível às distorções da malha.

- Mesmo para malhas muito pouco deformadas, como o paralelepípedo deformado tridimensionalmente por um ângulo de 80° , o método de Krylov-Newton que utiliza a matriz de 19 diagonais obteve a melhor performance.
- Para ângulos de deformação menores ou iguais a 60° , nos casos baseados no paralelepípedo deformado, o método *matrix-free* possui uma performance maior do que o método que emprega a matriz Jacobiana de 7 diagonais, tornando-se uma boa alternativa quando os recursos de memória são escassos – apesar de ainda muito atrás do método com a Jacobiana de exata.
- Observou-se que, em geral, há uma tênue queda na diferença de performance entre os métodos com o aumento da não-linearidade das equações.
- Por fim, os três métodos de Krylov-Newton foram comparados em um caso envolvendo uma malha generalizada com uma distribuição irregular de não-ortogonalidade. Novamente o método que emprega a Jacobiana completa obteve a melhor performance, seguido pelo método de Maliska *et al.*.

7.2 Recomendações

Como recomendações sugere-se:

- Utilizar o método de Krylov-Newton que emprega a matriz Jacobiana com todas as 19 diagonais derivadas da discretização em coordenadas generalizadas. Apesar da área livre de memória necessária ser razoavelmente maior, este método é muito robusto e rápido.
- Não empregar *backtracking* em problemas transientes e sempre testar algumas alternativas de escolha dos *forcing terms* (fixas ou adaptativas), quando o problema apresentar muito *oversolving*.

Para os trabalhos futuros que porventura sejam realizados seguindo a linha de pesquisa do presente trabalho, sugere-se:

- Testar outras alternativas de escolha dos *forcing terms* para evitar o *oversolving* nos métodos que utilizam a matriz Jacobiana completa. Uma escolha interessante e simples seria fixar \mathbf{h}_k em um valor relativamente alto por um número pequeno de iterações iniciais e depois apertar a tolerância nas demais iterações até a convergência.
- Aplicar o desenvolvimento teórico de Eisenstat e Walker (1994, 1996) para métodos quase-newtonianos que aproximam a matriz Jacobiana, de forma a obter algoritmos de *backtracking* e escolhas dos *forcing terms* que se ajustam melhor a estes tipos de métodos.
- Desenvolver, para as equações da simulação de reservatórios, aproximações da matriz Jacobiana com base nas funções interpolações, tal como é comumente empregado na solução das equações de Navier-Stokes. Neste caso, a matriz Jacobiana é aproximada apenas pelos termos de baixa ordem, aplicando-se métodos de alta ordem apenas na montagem das funções resíduo.

Apêndice A - Avaliação das Derivadas dos Potenciais

As equações que governam o escoamento dos fluidos no reservatório de petróleo, quando discretizadas, necessitam da avaliação das derivadas dos potenciais em todas as faces, como explanado no capítulo 3. Estas derivadas serão aqui apresentadas.

As derivadas diretas dos potenciais são dadas por

$$\left(\frac{\nabla F_p}{\nabla ?} \right)_e = F_e \left[\frac{F_p|_E - F_p|_P}{??} \right] \quad (\text{A.1})$$

$$\left(\frac{\nabla F_p}{\nabla ?} \right)_w = F_w \left[\frac{F_p|_P - F_p|_W}{??} \right] \quad (\text{A.2})$$

$$\left(\frac{\nabla F_p}{\nabla ?} \right)_n = F_n \left[\frac{F_p|_N - F_p|_P}{??} \right] \quad (\text{A.3})$$

$$\left(\frac{\nabla F_p}{\nabla ?} \right)_s = F_s \left[\frac{F_p|_P - F_p|_S}{??} \right] \quad (\text{A.4})$$

$$\left(\frac{\nabla F_p}{\nabla ?} \right)_t = F_t \left[\frac{F_p|_T - F_p|_P}{??} \right], \quad (\text{A.5})$$

$$\left(\frac{\nabla F_p}{\nabla ?} \right)_b = F_b \left[\frac{F_p|_P - F_p|_B}{??} \right]. \quad (\text{A.6})$$

Por sua vez, as derivadas cruzadas são dadas por

$$\left(\frac{\nabla F_p}{\nabla ?} \right)_e = F_e \frac{\left(F_{\gamma_{SE}}^E F_p|_{SE} + F_{\gamma_E}^E F_p|_E + F_{\gamma_{NE}}^E F_p|_{NE} + F_{\gamma_S}^P F_p|_S + F_{\gamma_P}^P F_p|_P + F_{\gamma_N}^P F_p|_N \right)}{??}, \quad (\text{A.7})$$

$$\left(\frac{\nabla F_p}{\nabla ?} \right)_e = F_e \frac{\left(F_{\gamma_{BE}}^E F_p|_{BE} + F_{\gamma_E}^E F_p|_E + F_{\gamma_{TE}}^E F_p|_{TE} + F_{\gamma_B}^P F_p|_B + F_{\gamma_P}^P F_p|_P + F_{\gamma_T}^P F_p|_T \right)}{??}, \quad (\text{A.8})$$

$$\left(\frac{\mathcal{I}F_p}{\mathcal{I}^?} \right)_w = F_w \frac{\left(F_{?SW}^W F_p|_{SW} + F_{?W}^W F_p|_W + F_{?NW}^W F_p|_{NW} + F_{?S}^P F_p|_S + F_{?P}^P F_p|_P + F_{?N}^P F_p|_N \right)}{??}, \quad (A.9)$$

$$\left(\frac{\mathcal{I}F_p}{\mathcal{I}^?} \right)_w = F_w \frac{\left(F_{?BW}^W F_p|_{BW} + F_{?W}^W F_p|_W + F_{?TW}^W F_p|_{TW} + F_{?B}^P F_p|_B + F_{?P}^P F_p|_P + F_{?T}^P F_p|_T \right)}{??}, \quad (A.10)$$

$$\left(\frac{\mathcal{I}F_p}{\mathcal{I}^?} \right)_n = F_n \frac{\left(F_{?NW}^N F_p|_{NW} + F_{?N}^N F_p|_N + F_{?NE}^N F_p|_{NE} + F_{?W}^P F_p|_W + F_{?P}^P F_p|_P + F_{?E}^P F_{?P}^P F_p|_E \right)}{??}, \quad (A.11)$$

$$\left(\frac{\mathcal{I}F_p}{\mathcal{I}^?} \right)_n = F_n \frac{\left(F_{?BN}^N F_p|_{BN} + F_{?N}^N F_p|_N + F_{?TN}^N F_p|_{TN} + F_{?B}^P F_p|_B + F_{?P}^P F_p|_P + F_{?T}^P F_p|_T \right)}{??}, \quad (A.12)$$

$$\left(\frac{\mathcal{I}F_p}{\mathcal{I}^?} \right)_s = F_s \frac{\left(F_{?SW}^S F_p|_{SW} + F_{?S}^S F_p|_S + F_{?SE}^S F_p|_{SE} + F_{?W}^P F_p|_W + F_{?P}^P F_p|_P + F_{?E}^P F_p|_E \right)}{??}, \quad (A.13)$$

$$\left(\frac{\mathcal{I}F_p}{\mathcal{I}^?} \right)_s = F_s \frac{\left(F_{?BS}^S F_p|_{BS} + F_{?S}^S F_p|_S + F_{?TS}^S F_p|_{TS} + F_{?B}^P F_p|_B + F_{?P}^P F_p|_P + F_{?T}^P F_p|_T \right)}{??}, \quad (A.14)$$

$$\left(\frac{\mathcal{I}F_p}{\mathcal{I}^?} \right)_t = F_t \frac{\left(F_{?TW}^T F_p|_{TW} + F_{?T}^T F_p|_T + F_{?TE}^T F_p|_{TE} + F_{?W}^P F_p|_W + F_{?P}^P F_p|_P + F_{?E}^P F_p|_E \right)}{??}, \quad (A.15)$$

$$\left(\frac{\mathcal{I}F_p}{\mathcal{I}^?} \right)_t = F_t \frac{\left(F_{?TS}^T F_p|_{TS} + F_{?T}^T F_p|_T + F_{?TN}^T F_p|_{TN} + F_{?S}^P F_p|_S + F_{?P}^P F_p|_P + F_{?N}^P F_p|_N \right)}{??}, \quad (A.16)$$

$$\left(\frac{\mathcal{I}F_p}{\mathcal{I}^?} \right)_b = F_b \frac{\left(F_{?BW}^B F_p|_{BW} + F_{?B}^B F_p|_B + F_{?BE}^B F_p|_{BE} + F_{?W}^P F_p|_W + F_{?P}^P F_p|_P + F_{?E}^P F_p|_E \right)}{??}, \quad (A.17)$$

$$\left(\frac{\mathcal{I}F_p}{\mathcal{I}^?} \right)_b = F_b \frac{\left(F_{?BS}^B F_p|_{BS} + F_{?B}^B F_p|_B + F_{?BN}^B F_p|_{BN} + F_{?S}^P F_p|_S + F_{?P}^P F_p|_P + F_{?N}^P F_p|_N \right)}{??}, \quad (A.18)$$

Os coeficientes F presentes nas equações acima são chamados de coeficientes de falha e estão reportados em detalhe em Cunha (1996).

Referências Bibliográficas

- AAVATSMARK, I.; BARKVE, T.; BØE, Ø. and MANNSETH, T. “*Discretization on non-orthogonal curvilinear grids for multi-phase flow.*” Proc. 4th European Conference on the Mathematics of Oil Recovery, vol. D, pp. 17, 1994.
- ABOU-KASSEM, J.H. and AZIZ, K. “*Handling of phase change in thermal simulators.*” SPEJ, Sep. 1985.
- AZIZ, K. and SETTARRI, A. “*Petroleum reservoir simulation.*” New York: ed. Applied Science Publishers, 1979.
- BARUA, J. and HORNE, R.N. “*Improving the performance of parallel (and serial) reservoir simulators.*” SPE 18408, 10th SPE Symposium on Reservoir Simulation, Houston, Feb. 6-8, 1989.
- BALAY, S.; GROPP, W.D.; McINNES, L.C. and SMITH, B.F. “*Efficient management of parallelism in object oriented numerical software libraries.*” In: ARGE, E.; BRUASET, A. M. and LANGTANGEN, H. P. “*Modern Software Tools in Scientific Computing.*” Birkhauser Press, pp. 163-202, 1997.1
- _____. “*{PETSc} 2.0 Users Manual.*” Argonne National Laboratory, ANL-95/11 - Revision 2.0.28, 2000.
- BARRETT, R.; BERRY, M.; CHAN, T.F.; DEMMEL, J.; DONATO, June; DONGARRA, J.; EIJKHOUT, V.; POZO, R.; ROMINE, C. and van der VORST, H. “*Templates for the solution of linear systems: building blocks for iterative methods*” SIAM, Philadelphia, 1994.
- BERTIGER, W.I. and KELSEY, F.J. “*Inexact adaptive Newton methods.*” SPE 13501, SPE 1985 Reservoir Simulation Symposium, Dallas, Feb. 10-13, 1985.

- BLANCO, M. and ZINGG, D.W. "A fast solver for the Euler equations on unstructured grids using a Newton-GMRES method." AIAA Paper 97-0331 (A97-15398), AIAA, Aerospace Sciences Meeting, 35th, Reno, NV, Jan. 6-9, 1997.
- BRITTO, P.R.F.; PEDROSA JR., O.A. and HORNE, R. "Application of boundary-fitted curvilinear grid to reservoir simulation." XI COBEM, pp. 129-132, São Paulo, Dez. 1991.
- BROWN, P.N. and SAAD, Y. "Hybrid Krylov methods for nonlinear systems of equations." SIAM J. Sci. Stat. Comput., vol. 11, pp. 450-481, 1990.
- BROYDEN, C.G. "A class of methods for solving nonlinear simultaneous equations." Math. Comp., 19, pp. 577-593, 1965.
- CHIN, P.; D'AZEVEDO, E.F.; FORSYTH, P.A. and TANG, W.P. "Preconditioned conjugate gradient methods for the incompressible Navier-Stokes equations." Int. J. Numer. Meth. Fluids, vol. 15, pp.273-295, 1992.
- CHOQUET, R. and ERHEL, J. "Newton-GMRES algorithm applied to compressible flows." Int. J. for Numer. Meth. Fluids, vol. 23, pp.177-190, 1996.
- COATS, K.H. "Simulation of steamflooding with distillation and solution gas." SPEJ, 16, No. 5, pp. 235-47, 1976.
- _____. "Elements of Reservoir Simulation" Lecture Notes, University of Texas, reprinted by Intercomp Resources Development and Engineering Inc., Houston.
- COATS, K.H.; NIELSEN, R.L.; TERHUNE, M.H. and WEBER, A.G. "Simulation of three-dimensional, two-phase flow in oil and gas reservoirs." Trans. SPE of AIME, 240, pp. 377-88, 1967.

- COATS, K.H.; GEORGE, W.D. and MARCUM, B.E. “*Three-dimensional simulation of steamflooding.*” Trans. SPE of AIME, 257, pp. 573-92, 1974.
- COLLINS, R.E. “*Flow of fluids through porous materials.*” Van Nostrand Reinhold, New York, 1961.
- CUNHA, A.R. “*Uma metodologia para simulação numérica tridimensional de reservatórios de petróleo utilizando modelo Black-Oil e formulação em frações mássicas.*” Florianópolis, 1996. Dissertação de Mestrado - Departamento de Engenharia Mecânica, UFSC.
- CZESNAT, A.O. “*Implementação de esquemas de interpolação para minimização da difusão numérica em simulação de reservatório de petróleo.*” Florianópolis, 1999. Dissertação de Mestrado - Departamento de Engenharia Mecânica, UFSC.
- DARCY, H. “*Les fontaines publiques de la Ville de Dijon*”, Dalmount, Paris, 1856. (republicado na integra em Hubbert, 1969)
- DEMBO, R.; EISENSTAT, S.C. and STEIHAUG, T. “*Inexact Newton methods.*” SIAM J. Numer. Anal., 19, pp. 400-408, 1982.
- DENNIS Jr., J.E. and SCHNABEL, R.B. “*Numerical methods for unconstrained optimization and nonlinear equations.*” Series in Automatic Computation, Prentice-Hall, Englewood Cliffs, NJ, 1983.
- DOUGLAS Jr., J.; PEACEMAN, D.W. and RACHFORD Jr., H.H. “*A method for calculating multi-dimensional immiscible displacement.*” Trans. SPE of AIME, 216, pp. 297-306, 1959.
- EISENSTAT, S.C. and WALKER, H.F. “*Globally convergent inexact Newton methods.*” SIAM J. Optimization, 4, pp.393-422, 1994.

- _____. “*Choosing the forcing terms in an inexact Newton method.*” SIAM J. Sci. Comput., 17, pp. 16-32, 1996.
- FLEMING, G.C. “*Modeling the performance of fractured wells in pattern floods using orthogonal curvilinear grids*” SPEJ, Oct. 1984.
- FORSYTH, P.A. and SAMMON, P.H. “*Gas phase appearance and disappearance in fully implicit Black Oil simulation.*” SPEJ, Oct. 1984.
- _____. “*Practical consideration for adaptive implicit methods in reservoir simulation.*” Journal of Computational Physics, Sep. 1986.
- FREUND, R.W. “*A Transpose-Free Quasi-Minimal Residual algorithm for non-Hermitian linear systems.*” SIAM Journal on Scientific Computing, 14(2), pp. 470-482, 1993.
- GROPP, W.D. and KEYES, D.E. “*Domain decomposition methods in computational fluid dynamics.*” Int. J. Numer. Meth. Fluids, vol. 14, pp. 147-165, 1992.
- HEGRE, T.M.; DALEN, V. and HENRIQUEZ, A. “*Generalized transmissibilities for distorted grids in reservoir simulation.*” SPE 15622, Proc. SPE 61th Annual Technical Conference and Exhibition, New Orleans, 1986.
- HIRASAKI, G.J. and O’DELL, P.M. “*Representation of reservoir geometry for numerical simulation.*” SPEJ, Dec. 1970.
- HUBBERT, M.K. “*Darcy’s Law and the field equations of the flow of underground fluids.*” Trans. SPE of AIME, 207, pp. 222-39, 1956. (republicado com correções em Hubbert, 1969)
- _____. “*The theory of ground-water motion and related papers.*”, Hafner Publishing, New York, 1969

JOHNSON, R.W.; McHUGH, P.R. and KNOLL, D.A. “*High-order scheme implementation using Newton-Krylov solution methods.*” Numerical Heat Transfer, Part B, 31, pp. 295-312, 1997.

_____. “*Defect correction with a fully coupled inexact Newton method.*” Numer. Heat Transfer, Part B, vol. 14, pp. 147-165, 1992.

KIM, J.G. and DEO, M.D. “*Inexact Newton-Krylov methods for the solution of implicit reservoir simulation problems.*” SPE 51908, 1999 SPE Reservoir Simulation Symposium, Houston, Feb. 14-17, 1999.

MacDONALD, R.C. and COATS, K. H. “*Methods for numerical simulation of water and gas coning.*” Trans. SPE of AIME, 249, pp. 425-36, 1970.

MALISKA, C.R. “*A solution method for three-dimensional parabolic fluid flow problems in nonorthogonal coordinates.*” PhD. Thesis, University of Waterloo, Waterloo, Canada, 1981.

_____. C.R. “*Transferência de Calor e Mecânica dos Fluidos Computacional.*” Rio de Janeiro, RJ, Brasil: ed. LTC - Livros Técnicos e Científicos Editora S. A., 1995.

MALISKA, C.R.; CUNHA, A.R.; LIVRAMENTO, M.A. and SILVA, A.F.C. “*Tridimensional petroleum reservoir simulation using generalized curvilinear coordinates.*” V Encontro Nacional de Ciências Térmicas, pp. 363-366, São Paulo, Dez. 1994.

MALISKA, C.R.; SILVA, A.F.C.; CZESNAT, A.; LUCIANETTI, R.M. and MALISKA Jr., C.R., “*Three-dimensional multiphase flow simulation in petroleum reservoirs using the mass fractions as dependent variables.*” SPE 39067, Aug. 1997.

- MARCONDES, F. “*Solução numérica usando métodos adaptativos-implícitos e malha de Voronoi de problemas de reservatórios de petróleo.*” Florianópolis, 1996. Tese de Doutorado - Departamento de Engenharia Mecânica, UFSC.
- MATTHEIS, H. and STRANG, G. “*The solution of nonlinear finite element equations.*” Int. J. Num. Meth. Eng., 14, pp.1613-1626, 1979.
- MEZZOMO, R.F. “*O escoamento em meios porosos na indústria do petróleo.*” B. Técn. PETROBRÁS, Rio de Janeiro, 35, pp. 21-22, 1992.
- NGHEIM, L.X. “*A new approach to Quasi-Newton methods with application to compositional modeling*” SPE 12242, 1983 SPE Reservoir Simulation Symposium, San Francisco, Nov. 15-18, 1983.
- PEACEMAN, D.W. “*Fundamentals of numerical reservoir simulation.*” Elsevier Science Publishing Co., New York, 1977.
- PERNICE, M. and WALKER, H.F. “*NITSOL: a Newton iterative solver for nonlinear systems.*” Tech. Rep. 5/96/85, Mathematics and Statistics Department, Utah State University, May 1996.
- PRAIS, F. e CAMPAGNOLO, E.A. “*Modelagem do escoamento multifásico em simulação de reservatório.*” XI COBEM, pp. 545-548, São Paulo, Dez. 1991.
- PUEYO, A. and ZINGG, D. W. “*An efficient Newton-GMRES solver for aerodynamic computations.*” AIAA Paper 97-1955 (A97-32464), AIAA, Computational Fluid Dynamics Conference, 13th, Snowmass Village, CO, June 29-July 2, 1997.
- _____. “*Progress in Newton-Krylov methods for aerodynamic calculations.*” AIAA Paper 97-0877 (A97-15875), AIAA, Aerospace Sciences Meeting, 35th, Reno, NV, Jan. 6-9, 1997.

- _____. “Improvements to a Newton-Krylov solver for aerodynamic flows.” AIAA Paper 98-0619 (A98-16464), AIAA, Aerospace Sciences Meeting & Exhibit, 36th, and 1998 ASME Wind Energy Symposium, Reno, NV, Jan. 12-15, 1998.
- RHEINBOLDT, W.C. “*Methods for solving systems of nonlinear equations.*” 2nd Edition, SIAM, Philadelphia, 1998.
- ROBERTSON, G.E. and WOO, P.T. “Grid orientation effects and the use of orthogonal curvilinear coordination in reservoir simulation.” SPEJ, Feb. 1978.
- RUSSEL, T.F. “*Stability analysis and switching criteria for adaptive implicit methods based on the CFL condition.*” SPEJ, Feb. 1989.
- SAAD, Y. “*Iterative Methods for Sparse Linear Systems.*” PWS Publishing Co., Boston, 1996.
- SAAD, Y and SCHULTZ, M.H. “GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems.” SIAM Journal on Scientific and Statistical Computing, 7, pp. 856-869, 1986.
- SCHEIDEGGER, A.E. “*Physics of flow through porous media.*” 3rd end., University of Toronto Press, Toronto, 1974.
- SHADID, J.N.; TUMINARO, R.S. and WALKER, H.F. “An inexact Newton method for fully-coupled solution of the Navier-Stokes equations with heat and mass transport.” SANDIA Report, SAND97-0132, Feb. 1997.
- SHARPE, H.N. and ANDERSON, D.A. “A new adaptive orthogonal grid generation procedure for reservoir simulation.” SPE 20744, 65th ATC&E, New Orleans, Sep. 23-26, 1990.

- _____. “*Orthogonal curvilinear grid generation with preset internal boundaries for reservoir simulation.*” SPE 21235, 11th Reservoir Simulation Symposium, Anaheim, Feb 17-20, 1991.
- SHEFFIELD, M. “*Three phase flow including gravitational, viscous, and capillary forces.*” Trans. SPE of AIME, 246, pp. 255-69, 1969.
- SHELDON, J.W.; ZONDEK, B. and CARDWELL, W. T. “*One-dimensional, incompressible, non-capillary, two-phase fluid flow in a porous medium.*” Trans. SPE of AIME, 216, pp. 290-6, 1959.
- SONIER, F. and CHAUMET, P. “*A fully implicit three dimensional model in curvilinear coordinates.*” Trans. SPE 253, pp. 361-70, 1974.
- SONNEVELD, P. “*CGS, a fast Lanczos-type solver for nonsymmetric linear systems.*” SIAM Journal on Scientific and Statistical Computing, 10(1), pp. 36-52, 1989.
- SPILLETTE, A.G.; HILLESTAD, J.G. and STONE, H.L. “*A high-stability sequential solution approach to reservoir simulation*” SPE 4542, 48th Annual Fall Meeting, Las Vegas, 1973.
- STONE, H.L. and GARDER Jr., A.O. “*Analysis of gas-gap or dissolved-gas reservoirs.*” Trans. SPE of AIME, 222, pp. 92-104, 1961.
- THOMAS, G.W. and THURNAU, D.H. “*Reservoir simulation using and adaptive implicit method.*” SPEJ, Feb. 1983.
- THOMPSON, J.F.; THAMES, F.C. and MASTIN, W.C. “*Boundary-fitted curvilinear coordinates system for solution of partial differential equation on fields containing any number of arbitrary two-dimensional bodies.*” NASA Langley Research Centre, CR-2729, EUA, 1976.

_____. “*TOMCAT – A code for numerical generation of boundary-fitted curvilinear coordinate systems on fields containing any number or arbitrary two-dimensional bodies.*” *Journal of Computational Physics*, vol. 24, pp. 274-302, 1977.

van der VORST, “*Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of non-symmetric linear systems.*” *SIAM Journal on Scientific and Statistical Computing*, 12, pp. 631-644, 1992.

WHITAKER, S. “*The equations of motion in porous media.*” *Chem. Eng. Sci.*, 21, pp. 291-300, 1966.

_____. “*Advances in theory of fluid motion in porous media.*” *Indust. and Engineering Chemistry*, 61, No. 12, pp. 14-28, 1969.