

UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO TECNOLÓGICO
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

ADRIANA DA COSTA BARBOSA

**IMPLEMENTAÇÃO EM PARALELO DO MÉTODO DOS
ELEMENTOS FINITOS EMPREGANDO ESTRUTURA DE DADOS
POR ARESTA PARA SIMULAÇÃO EM CLUSTER MULTI-CORE
DE ESCOAMENTO MISCÍVEL EM MEIOS POROSOS**

VITÓRIA

2009

Livros Grátis

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.

ADRIANA DA COSTA BARBOSA

**IMPLEMENTAÇÃO EM PARALELO DO MÉTODO DOS
ELEMENTOS FINITOS EMPREGANDO ESTRUTURA DE DADOS
POR ARESTA PARA SIMULAÇÃO EM CLUSTER MULTI-CORE
DE ESCOAMENTO MISCÍVEL EM MEIOS POROSOS**

Dissertação apresentada ao Programa de Pós-Graduação em Informática do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do Grau de Mestre em Informática. Orientador: Profa. Dra. Lucia Catabriga

VITÓRIA

2009

Dados Interacionais de Catalogação-na-publicação (CIP)
(Biblioteca Central da Universidade Federal do Espírito Santo, ES, Brasil)

Barbosa, Adriana da Costa, 1982 -

B238i

Implementação em paralelo do método dos elementos finitos empregando estrutura de dados por aresta para simulação em cluster multi-core de escoamento miscível em meios porosos/Adriana da Costa Barbosa.- 2009
158 f. :il.

Orientadora: Lucia Catabriga

Dissertação (mestrado) - Universidade Federal do Espírito Santo, Centro Tecnológico

1. Método dos elementos finitos. 2. Simulação (Computadores). 3. Processamento paralelo (Computadores). 4. Método de decomposição. 5. Estruturas de dados (Computação). I. Catabriga, Lucia. II. Universidade Federal do Espírito Santo. Centro Tecnológico. III. Título.

CDU:004

ADRIANA DA COSTA BARBOSA

**IMPLEMENTAÇÃO EM PARALELO DO MÉTODO DOS
ELEMENTOS FINITOS EMPREGANDO ESTRUTURA DE DADOS
POR ARESTA PARA SIMULAÇÃO EM CLUSTER MULTI-CORE
DE ESCOAMENTO MISCÍVEL EM MEIOS POROSOS**

Dissertação submetida ao programa de Pós-Graduação em Informática do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para a obtenção do Grau de Mestre em Informática .

Aprovada em 24 de Agosto de 2009.

COMISSÃO EXAMINADORA

Profa. Lucia Catabriga, D.Sc.

Prof. Alberto Ferreira de Souza, D.Sc.

Profa. Andréa Maria Pedrosa Valli, D.Sc.

Profa. Sandra Mara Cardoso Malta, D.Sc.

Para meu querido Wesley.

AGRADECIMENTOS

Agradeço a minha orientadora, Lucia Catabriga, pela presença constante nos ensinamentos. Sou grata pela oportunidade, paciência, compreensão, confiança e apoio nos vários momentos de convivência.

Agradeço a CAPES, pelo apoio financeiro, que tornou possível a realização desta dissertação.

Agradeço ao Programa de Pós-graduação em Informática da Universidade Federal do Espírito Santo, pela oportunidade de realização do mestrado.

Agradeço aos membros da banca examinadora, Andréa Maria Pedrosa Valli, Sandra Mara Cardoso Malta e Alberto Ferreira de Souza, pela leitura atenta e observações que foram muito pertinentes para a melhora do trabalho.

Agradeço a todos os professores que me ajudaram com seus valiosos ensinamentos no decorrer do curso e aos integrantes do LCAD pelo apoio técnico.

Enfim, gostaria de deixar meu MUITO OBRIGADA a TODOS que me acompanharam na realização desse trabalho.

Resumo da dissertação apresentada à PPGI/UFES como parte dos requisitos necessários para a obtenção do grau de Mestre.

Implementação em paralelo do método dos elementos finitos empregando estrutura de dados por aresta para simulação em cluster multi-core de escoamento miscível em meios porosos

Adriana da Costa Barbosa

Agosto/2009

Orientadora: Lucia Catabriga

Programa: Informática

Simulação de reservatórios tem sido extensivamente estudados por um número expressivo de autores nas últimas três décadas. Escoamentos miscíveis que modelam recuperação avançada e processos de injeção de traçadores tem sido solucionados por formulações estabilizadas do método dos elementos finitos. A concentração do fluido injetado na mistura é a principal variável. Porém, o cálculo da velocidade de Darcy, responsável pelo transporte, requer especial atenção já que possui uma forte influência na estabilidade e acurácia da aproximação da concentração. Em geral, as formulações propostas para aplicações em meios porosos podem requerer soluções de sistemas lineares envolvendo milhões de incógnitas. Esses sistemas são usualmente solucionados por métodos iterativos baseados nos espaços de Krylov e o sucesso dessas técnicas de solução requerem uma implementação eficiente do produto matriz-vetor. Esquemas elemento-por-elemento (EBE) e baseados nas arestas (EDS) tem sido usados com sucesso, pois possibilitam o tratamento de matrizes de grande porte de uma maneira simples e direta.

Neste trabalho, usamos uma formulação estabilizada de elementos finitos que considera: uma decomposição do domínio com estrutura de blocos para a matriz resultante e ambos esquemas de armazenamentos (EBE e EDS). O código foi implementado para plataformas de memória distribuída utilizando o padrão MPI e dois algoritmos de distribuição de jobs suportados pela ferramenta Sun Grid Engine (SGE): *fill-up* e *single-core*. Os jobs paralelos são escalonados para executar em dois clusters

multi-core do Laboratório de Computação de Alto Desempenho da Universidade Federal do Espírito Santo e um cluster SGI Altix 450 do Núcleo de Computação de Alto Desempenho da Universidade Federal do Rio de Janeiro. Nossos experimentos mostram que, muito embora máquinas mais modernas possuam um número maior de cores por CPU e estes sejam mais rápidos, a competição destes cores pela hierarquia de memória tem grande impacto no desempenho final das aplicações. Além disso, a estratégia baseada nas arestas reduziu o tempo de CPU quando comparada com a estratégia baseada nos elementos para qualquer número de cores.

Abstract of dissertation presented to PPGI/UFES as a partial fulfilment of the requirements for the degree of Master.

Parallel implementation of finite element method employing edge-based data in miscible displacement simulations in porous media using multi-core cluster

Adriana da Costa Barbosa

August/2009

Advisor: Lucia Catabriga

Department: Computer Science

Reservoir simulation has been studied extensively by a number of authors in the last three decades. Miscible displacements which model enhanced oil recovery and tracer injection processes have been solved by stabilized finite element formulations. The concentration of the injected fluid in the mixture is the main variable. However, the calculation of Darcy's velocity, responsible for the transport, deserves special attention, since it has a strong influence on the stability and accuracy of the concentration approximation. Finite element formulations proposed for complex porous media application can require solution of linear systems of equations involving millions of unknowns. These systems are usually solved by Krylov space iterative update techniques, and the success of these solution strategies require an efficient implementation of matrix-vector products. The element-by-element (EBE) and edge-based (EDS) data structures have been very successful because they can handle large sparse matrices in a simple and straightforward manner.

In this work, we use a parallel stabilized finite element formulation that considers: a domain decomposition strategy with block-arrowhead structure of the resulting matrix and both EBE and EDS storage formats. The code implemented is designed for parallel computing platforms with distributed-memory, particularly for clusters of workstations. The exchange of information across the sub-domains is achieved using the Message Passing Interface, MPI. We compare two job scheduling alternatives supported by the Sun Grid Engine (SGE) distributed resource management software: fill-up and single core. The parallel jobs are scheduled to run in two

clusters of multi-core computers of the Laboratório de Computação de Alto Desempenho (LCAD) at the Universidade Federal do Espírito Santo, and in the cluster SGI Altix 450 of the Núcleo de Computação o de Alto Desempenho (NACAD) at the Universidade Federal do Rio de Janeiro. Our experiments show that, although modern machines have a larger number of cores and are faster than previous ones, the competition of these cores for the memory hierarchy of their Central Processing Unit (CPU) has a strong impact on the performance of parallel applications. We also observed that the edge scheme outperforms the element scheme for all numbers of cores employed in the experiments.

Sumário

1	Introdução	7
2	Formulação Matemática	14
2.1	Equações Governantes	14
2.1.1	Conservação de Massa	15
2.1.2	Lei de Darcy	17
2.1.3	Equação da Pressão	19
2.1.4	Equação da Concentração	21
2.2	Condições de Contorno e Condições Iniciais	26
3	Formulação Numérica	29
3.1	Formulação Variacional Semi-discreta	29
3.2	Estrutura de Dados por Elementos	37
3.2.1	Matrizes de Coeficientes para a Equação da Pressão	40
3.2.2	Matriz de Massa de Galerkin para a Concentração	42
3.2.3	Matriz de Convecção de Galerkin para a Concentração	42
3.2.4	Matriz de Difusão de Galerkin para a Concentração	43
3.2.5	Correção SUPG da Matriz de Massa	44

3.2.6	Correção SUPG da Matriz de Convecção	45
3.2.7	Correção SUPG da Matriz de Difusão	46
3.2.8	Matriz de Correção do Operador de Descontinuidade	46
3.2.9	Matrizes de Coeficientes para a Equação da Velocidade	47
3.3	Estrutura de Dados por Aresta	50
3.3.1	Matrizes das Arestas	52
3.3.2	Matriz da Equação da Pressão	52
3.3.3	Matriz de Massa de Galerkin para a Concentração	53
3.3.4	Matriz de Convecção de Galerkin para a Concentração	53
3.3.5	Matriz de Difusão de Galerkin para a Concentração	53
3.3.6	Correção SUPG da Matriz de Massa	54
3.3.7	Correção SUPG da Matriz de Convecção	54
3.3.8	Correção SUPG da matriz de difusão	54
3.3.9	Matriz de Correção do Operador de Descontinuidade	54
3.3.10	Matriz de Coeficientes para a Equação da Velocidade	55
3.4	Discretização no Tempo	56
3.5	Resolução dos Sistemas Lineares Resultantes	58
3.6	Considerações Finais	60
4	Estratégias Paralelas	61
4.1	Computação Paralela	61
4.2	MPI (Message Passing Interface)	65
4.3	Geração e Particionamento de Malhas	67

4.4	Decomposição de Domínio	71
4.5	Algoritmo Paralelo	75
5	Metodologia Experimental	78
5.1	Ambiente Computacional	78
5.2	Métricas	80
6	Experimentos Numéricos	82
6.1	Injeção de Traçadores	83
6.2	Injeção Contínua	88
6.3	Condições Iniciais e de Contorno	93
6.4	Desempenho Paralelo	98
6.4.1	Tipos de Escalonamento das Enterprises	99
6.4.2	Comparação entre as Estratégias de Armazenamento	113
7	Conclusões	133

Lista de Figuras

2.1	Esquema ilustrativo do fluxo no meio poroso na direção x	15
2.2	Esquema do experimento de Darcy sobre o fluxo de água através de filtro de areia.	17
2.3	Descrição do contorno em um problema de meios porosos.	27
3.1	Elemento triangular linear.	38
3.2	Transformação do elemento genérico para o elemento padrão.	39
3.3	Elementos adjacentes a aresta s , formada pelos nós 1 e 2.	51
4.1	Esquema de interconexão para <i>Ethernet</i>	63
4.2	Esquema de comunicação <i>crossbar</i> para <i>Infiniband</i>	64
4.3	Tipos de malhas.	68
4.4	Exemplos de particionamentos de malha com e sem balanceamento de carga e minimização da comunicação.	69
4.5	Exemplo de particionamento de malhas pelo METIS.	71
4.6	Sobreposição de malhas.	72
4.7	Malha pequena com 4 partições.	73
6.1	Malha <i>five-spot</i>	84

6.2	Modelo <i>five-spot</i>	84
6.3	Curvas de isoconcentração para os níveis de dispersão, $\alpha_l = 10,0$ e $\alpha_t = 1,0$ e $M=1$ - Problema do traçador.	86
6.4	Curvas de isoconcentração para os níveis de dispersão, $\alpha_l = 1,0$, $\alpha_t = 0,0$ e $M=1$ - Problema do traçador.	86
6.5	Comparação do histórico da concentração no poço produtor observado em 3000 dias considerando dois níveis de dispersão - Problema do traçador.	87
6.6	Comparação do histórico da concentração no poço produtor observado em 3000 dias com uma solução semi-analítica - Problema do traçador.	87
6.7	Curvas de isoconcentração para os níveis de dispersão, $\alpha_l = 1$, $\alpha_t = 0$ e $M=1$ - Injeção Contínua.	90
6.8	Comparação das curvas de isoconcentração considerando a velocidade de Darcy (à esquerda) e a velocidade com esquemas de pós-processamento (à direita) - Níveis de dispersão, $\alpha_l = 10$, $\alpha_t = 1$ e $M=20$ - Injeção Contínua.	92
6.9	Tipos de orientação de malhas estruturadas.	93
6.10	Comparação das curvas de concentração do deslocamento considerando duas orientações de malha - Níveis de dispersão, $\alpha_l = 10,0$, $\alpha_t = 1,0$ e $M=20$ - Injeção Contínua	94
6.11	Condição inicial radial para o deslocamento miscível.	95
6.12	Cone de altura unitária usado na definição do raio da curva s no problema de injeção contínua.	95
6.13	Comparação das curvas de isoconcentração da considerando uma condição inicial radial para malha com orientação paralela - Níveis de dispersão, $\alpha_l = 10$, $\alpha_t = 1$ e $M=20$ - Injeção Contínua.	96
6.14	Curvas de Pressão para o problema de injeção contínua considerando a simulação para 250 dias.	97

6.15	<i>Speedup</i> para os escalonamentos <i>fill-up</i> e <i>single-core</i> - Malha 512×512 - Problema do traçador - Enterprise 2.	99
6.16	<i>Speedup</i> para os escalonamentos <i>fill-up</i> e <i>single-core</i> - Malha $1024 \times$ 1024 - Problema do traçador - Enterprise 2.	100
6.17	<i>Speedup</i> para os escalonamentos <i>fill-up</i> e <i>single-core</i> - Malha $2048 \times$ 2048 - Problema do traçador - Enterprise 2.	100
6.18	<i>Speedup</i> para o escalonamento <i>fill-up</i> em diferentes malhas - Problema do traçador - Enterprise 2.	101
6.19	<i>Speedup</i> para o escalonamento <i>single-core</i> em diferentes malhas - Pro- blema do traçador - Enterprise 2.	102
6.20	Tempo de processamento para vários processadores comparando os dois tipos de escalonamento em diversas malhas - Problema do traçador - Enterprise 2.	103
6.21	<i>Speedup</i> para os escalonamentos <i>fill-up</i> e <i>single-core</i> - 512×512 - Problema do traçador - Enterprise 3.	104
6.22	<i>Speedup</i> para os escalonamentos <i>fill-up</i> e <i>single-core</i> - 1024×1024 - Problema do traçador - Enterprise 3.	105
6.23	<i>Speedup</i> para os escalonamentos <i>fill-up</i> e <i>single-core</i> - 2048×2048 - Problema do traçador - Enterprise 3.	105
6.24	Comparação do <i>Speedup</i> para o escalonamentos <i>fill-up</i> em relação a diferentes malhas - Problema do traçador - Enterprise 3.	106
6.25	Comparação do <i>Speedup</i> para o escalonamentos <i>single-core</i> em relação a diferentes malhas - Problema do traçador - Enterprise 3.	106
6.26	Tempo de processamento dos dois tipos de escalonamento considerando diferentes malhas - Problema do traçador - Enterprise 3.	108
6.27	Comparação do <i>Speedup</i> para o escalonamento <i>fill-up</i> - Malha $512 \times$ 512 - Problema do traçador - Enterprises 2 e 3.	109

6.28	Comparação do <i>Speedup</i> para o escalonamento <i>fill-up</i> - Malha 1024×1024 - Problema do traçador - Enterprises 2 e 3.	110
6.29	Comparação do <i>Speedup</i> para o escalonamento <i>fill-up</i> Malha - 2048×2048 - Problema do traçador - Enterprises 2 e 3.	110
6.30	Tempo de processamento para vários processadores comparando os escalonamento <i>fill-up</i> em diversas malhas. - Problema do traçador- Enterprises 2 e 3.	111
6.31	Comparação do <i>Speedup</i> para o escalonamento <i>single-core</i> - Malha 512×512 - Problema do traçador - Enterprises 2 e 3.	112
6.32	Comparação do <i>Speedup</i> para o escalonamento <i>single-core</i> - Malha 1024×1024 - Problema do traçador - Enterprises 2 e 3.	112
6.33	Comparação do <i>Speedup</i> para o escalonamento <i>single-core</i> - Malha 2048×2048 - Problema do traçador - Enterprises 2 e 3.	113
6.34	Tempo de processamento para vários processadores comparando escalonamento <i>single-core</i> em diversas malhas - Problema do traçador - Enterprises 2 e 3.	114
6.35	<i>Speedup</i> obtidos pela Altix e Enterprise 2 - Malha 2048×2048 - Problema do traçador.	115
6.36	<i>Speedup</i> obtidos pela Altix e Enterprise 3 - Malha 2048×2048 - Problema do traçador.	115
6.37	<i>Speedups</i> produzidos pela Altix considerando várias malhas - Problema do traçador.	116
6.38	<i>Speedups</i> comparando Elemento e Aresta - Problema do traçador- Malha 512×512 - Enterprise 2.	117
6.39	<i>Speedups</i> comparando Elemento e Aresta - Problema do traçador - Malha 512×512 - Enterprise 3.	118

6.40	Eficiência de Elemento e Aresta - Problema do traçador- Malha 512×512 - Enterprise 2.	118
6.41	Eficiência de Elemento e Aresta - Problema do traçador- Malha 512×512 - Enterprise 3.	119
6.42	Tempo de processamento comparando as estruturas Aresta e Elemento nas duas Enterprises - Problema do traçador - Malha 512×512	120
6.43	<i>Speedups</i> comparando Elemento e Aresta - Problema do traçador- Malha 2048×2048 - Enterprise 2.	121
6.44	<i>Speedups</i> comparando Elemento e Aresta - Problema do traçador- Malha 2048×2048 - Enterprise 3.	121
6.45	Eficiência de Elemento e Aresta - Problema do traçador - Malha 2048×2048 - Enterprise 2.	122
6.46	Eficiência de Elemento e Aresta - Problema do traçador - Malha 2048×2048 - Enterprise 3.	122
6.47	Tempo de processamento comparando as estruturas Aresta e Elemento nas duas Enterprises - Problema do traçador - Malha 2048×2048	123
6.48	<i>Speedups</i> comparando Elemento e Aresta - Problema do traçador - Malha 2048×2048 - Altix.	124
6.49	<i>Speedups</i> comparando Elemento e Aresta - Problema injeção contínua- Malha 512×512 - Enterprise 2.	124
6.50	<i>Speedups</i> comparando Elemento e Aresta - Problema injeção contínua - Malha 512×512 - Enterprise 3.	125
6.51	Eficiência de Elemento e Aresta - Problema injeção contínua- Malha 512×512 - Enterprise 2.	125
6.52	Eficiência de Elemento e Aresta - Problema injeção contínua- Malha 512×512 - Enterprise 3.	126

6.59	<i>Speedups</i> comparando Elemento e Aresta - Problema injeção contínua - Malha 2048×2048 - Altix.	126
6.53	Tempo de processamento comparando as estruturas Aresta e Ele- mento - Malha 512×512 - Problema injeção contínua	127
6.54	<i>Speedups</i> comparando Elemento e Aresta - Problema injeção contínua- Malha 2048×2048 - Enterprise 2.	128
6.55	<i>Speedups</i> comparando Elemento e Aresta - Problema injeção contínua- Malha 2048×2048 - Enterprise 3.	128
6.56	Eficiência de Elemento e Aresta - Problema injeção contínua - Malha 2048×2048 - Enterprise 2.	129
6.57	Eficiência de Elemento e Aresta - Problema injeção contínua - Malha 2048×2048 - Enterprise 3.	129
6.58	Tempo de processamento comparando as estruturas Aresta e Ele- mento - Malha 2048×2048 - Problema injeção contínua	130
6.60	Distribuição de tempo observado para o processador 0 - Problema do Traçador.	131
6.61	Divisão do tempo de execução na TAU - Problema do Traçador.	132
6.62	Rotina <i>All_reduce</i> utilizada na operação de produto interno - Pro- blema do Traçador.	132

Lista de Tabelas

4.1	Comparativo da estrutura de um programa C, em serial (à esquerda) e paralelo (à direita), utilizando as rotinas básicas de gerenciamento para o padrão MPI.	65
-----	---	----

Capítulo 1

Introdução

Nos últimos anos houve um aumento nas pesquisas relacionadas simulação de escoamentos em meios porosos devido a, dentre outras coisas, sua importância para a indústria do petróleo e a crescente preocupação com o aumento de contaminação de águas subterrâneas. A modelagem de escoamento miscível em meios porosos possui, dentre outras, duas importantes aplicações, transporte de traçadores e injeção contínua. A injeção de traçadores é importante para obter informações que caracterizam o meio como barreiras ao escoamento, direção de escoamento, dispersividade, etc [14]. Especificamente, na engenharia de reservatório de petróleo, é uma técnica usada para monitorar o movimento do fluido visando informações sobre as propriedades do reservatório [3]. No campo ambiental, essa técnica é utilizada na análise do escoamento das águas subterrâneas, buscando estimar e prevenir os efeitos da movimentação de um contaminante nesse meio [15].

O deslocamento miscível está relacionado à recuperação de óleo em reservatórios. Ao ser descoberto, um reservatório apresenta uma quantidade de energia denominada de primária que, quando dissipada, ocasiona diferenças de pressão no interior do reservatório, resultando no deslocamento do fluido residente em direção aos poços de produção. Em virtude das diferenças de pressão, o volume de óleo retirado inicialmente é denominado de recuperação primária e é pouco eficiente, pois retem um volume de óleo considerável [17, 54]. Para recuperar o óleo que fica retido no reservatório, pode-se compensar a redução de energia primária introduzindo energia

denominada de secundária. Isso é feito por meio da injeção de fluidos visando deslocar o óleo residente. O volume adicional de óleo obtido por esse mecanismo denomina-se recuperação secundária convencional. Nela, o deslocamento é imiscível, ou seja, o fluido injetado e o residente não se misturam, o que pode ocasionar altas tensões nas interfaces entre os fluidos reduzindo a capacidade do fluido injetado de expulsar o óleo. Além disso, caso o fluido injetado seja menos viscoso que o óleo, haverá a formação dos dedilamentos viscosos, o que prejudicará a produção [66]. Visando contornar esses problemas para melhorar a taxa de recuperação, surgem os métodos especiais de recuperação secundária que são de três tipos, térmicos, químicos ou miscíveis. Os térmicos primam pela redução da viscosidade do óleo por meio do aumento da temperatura do reservatório através da injeção de fluidos quentes. Os químicos agem de forma semelhante por meio de compostos químicos que podem aumentar a viscosidade da água e/ou reduzir a viscosidade do óleo. Os miscíveis buscam reduzir as tensões interfaciais, pois o fluido injetado possui a capacidade de se misturar ao fluido residente, ou seja, de formar uma mistura homogênea, composta por uma única fase. Dessa forma, espera-se que o óleo seja totalmente deslocado [54, 66].

Os custos operacionais e dos fluidos injetados são muito altos nos métodos especiais de recuperação secundária, inviabilizando simulações experimentais. Assim, é preciso fazer alguns estudos antes do processo de recuperação buscando verificar a viabilidade técnica e econômica da operação. Esses estudos podem ser realizados através de simulações numéricas. O comportamento do deslocamento de fluidos em meios porosos pode ser descrito através de um modelo matemático caracterizado por um conjunto de equações parciais diferenciais transientes, não lineares e acopladas, que são vinculadas a um conjunto apropriado de condições de contorno para representar as características físicas do escoamento. Tal modelo é governado por um sistema elíptico, composto pela equação da pressão, acoplado a equação de concentração através da velocidade proveniente da Lei de Darcy. Devido à natureza desse modelo, na maior parte dos casos é inviável obter uma solução analítica. Portanto, são utilizadas técnicas numéricas estáveis, precisas e computacionalmente eficientes para resolver as equações que constituem o modelo, resultando em uma solução que

satisfatoriamente represente o escoamento no meio. As principais técnicas numéricas utilizadas na resolução de tais problemas são diferenças finitas, volumes finitos e elementos finitos.

O método de elementos finitos tem sido amplamente adotado no escoamento em meios porosos, devido a sua capacidade em trabalhar com malhas não-estruturadas, uma vez que os meios porosos de interesse são estruturas geológicas complexas que podem conter falhas, estratificações e propriedades heterogêneas. Esse método é uma técnica usada para construir soluções aproximadas através da divisão do domínio em um número finito de subdomínios, denominados elementos finitos, visando solucionar problemas de valor de contorno e de valor inicial [28]. Existem várias formulações em elementos finitos para escoamento em meios porosos, cada uma específica para determinados comportamentos.

A formulação clássica de Galerkin é aplicada com bons resultados a problema com comportamento predominantemente difusivo [4]. No entanto, se a difusão é muito pequena comparada à convecção, o que caracteriza um problema predominantemente convectivo, essa formulação não é indicada, pois haverá oscilações espúrias na solução [4]. Nesse caso, uma formulação estabilizada que acrescente difusão na direção de escoamento para evitar oscilações deve ser usada, como por exemplo, a formulação *Streamline Upwind Petrov-Galerkin* (SUPG). Quando as vizinhanças das regiões com alto gradiente são suscetíveis de soluções descontínuas, a formulação SUPG isolada não consegue evitar tais oscilações, assim, é necessário suavizar a solução nessas regiões através de um operador de choque e captura, como o operador *Consistent Approximate Upwind*(CAU) [23]. Devido a grande importância que a velocidade tem no modelo e o fato dessa grandeza ser obtida por leis empíricas, é comum sua precisão ser de menor ordem que as outras grandezas, comprometendo a precisão da solução. Portanto, é necessário utilizar algum esquema que garanta a mesma precisão de todas as grandezas, um exemplo é o pós-processamento da velocidade proposto por Malta [37].

A discretização utilizando o método dos elementos finitos recai na solução de um sistema acoplado de equações não-lineares que é dividido em dois sistemas lineares e

um sistema de equações diferenciais ordinárias. Na solução desse sistema acoplado não é indicado usar os métodos diretos, pois, a medida que se aumenta a ordem do problema, tais métodos se tornam mais dispendiosos que os métodos iterativos em relação ao tempo de processamento e ao consumo de memória. Portanto, é mais vantajoso usar os métodos iterativos. Dentre os métodos iterativos, foi adotado os não-estacionários baseados nos sub-espacos de Krylov, como o método dos Gradientes Conjugados (GC) e o método Mínimo Generalizado (GMRES) [55]. Nesses métodos, as principais operações envolvidas na solução dos sistemas lineares são o produto matriz-vetor e o produto interno entre dois vetores. A realização dessas operações influencia diretamente o custo de processamento dos métodos, pois quanto maiores são os sistemas, maior é o uso de memória e mais custosas são essas operações, conseqüentemente, maior é o tempo de processamento.

As matrizes desses sistemas são caracterizadas por uma grande esparsidade e seus coeficientes não nulos estão em posições aleatórias, ou seja, não estão dispostos segundo algum padrão. Para armazenar apenas os elementos não nulos dessas matrizes, e com isso reduzir o consumo de memória e o número de operações de ponto flutuante, é preciso definir estratégias de armazenamento especiais para as matrizes, tais como, por exemplo, elemento-por-elemento (EBE) [27], aresta-por-aresta (EDE) [9], *Modified Sparse Row* (MSR), *Compressed Storage Row* (CSR) [55], dentre outras.

Apenas o uso de estratégias de armazenamento especiais não é suficiente para garantir um melhor desempenho nos problemas de escoamento em meios porosos pois esses problemas demandam grande tempo de processamento e alto consumo de memória, portanto, necessitam de um alto poder de processamento. Uma alternativa para suprir essa demanda é utilizar a computação paralela, que se mostra muito eficaz na busca por um melhor desempenho. Dentre as arquiteturas paralelas mais utilizadas tem-se os supercomputadores e os *Clusters*. Antigamente, os supercomputadores, soluções proprietárias e de alto custo financeiro, eram quase que exclusivamente utilizados para suprir a demanda por alto desempenho. No entanto, o acesso facilitado e o crescente poder de processamento dos computadores de menor porte fez com que os *Clusters* se tornassem muito populares, devido a seu menor custo financeiro, e a seu crescente poder de processamento. O lançamento recente dos processadores

multi-core, que oferecem suporte nativo ao processamento paralelo, já influenciou os ambientes de *Cluster* e a inserção desses processadores em tais ambientes já é uma realidade [19].

Um *Cluster* pode ser entendido como um conjunto de computadores independentes e interconectados, que formam uma plataforma de alto desempenho para executar aplicações de forma paralela [5]. Nesse sistema distribuído, para que haja comunicação entre os processadores, são realizadas operações de envio e recebimento de mensagens. Esse serviço é oferecido por bibliotecas e linguagens, tais como PVM (*Parallel virtual Machine*) [24], padrão MPI (*Message Passing Interface*) [42], *TreadMarks* [32] e HPF *High Performance Fortran* [26]. O padrão MPI foi adotado nesse trabalho através da implementação lam-mpi, por apresentar maior desempenho e versatilidade em relação as demais bibliotecas [35].

Na paralelização do problema, dentre as várias estratégias que podem ser utilizadas em ambientes paralelos de memória distribuída, uma muito atrativa é o método de decomposição de domínio, que atua propondo a divisão do domínio em subdomínios através de algoritmos de particionamento. O problema original é tratado como uma série de subproblemas de tamanho reduzido. No ambiente de computação paralela, os subdomínio são enviados para processadores diferentes para que sejam solucionados de forma concorrente. A solução geral é alcançada combinando a solução dos subdomínios [55]. Essa divisão do domínio tem que ser feita de forma balanceada entre os processadores, para que o número de elementos seja aproximadamente o mesmo em cada processador e a comunicação seja minimizada [35].

Nesse sentido, esse trabalho visa resolver o modelo matemático que governa o processo de escoamento miscível em meios porosos, utilizando para tal uma versão paralela do método dos elementos finitos baseada na decomposição de domínio. Além disso, são utilizadas as estratégias especiais de armazenamento elemento-por-elemento e aresta-por-aresta que armazenam as matrizes provenientes da discretização de forma alternativa. Vários são os estudos encontrados na literatura que abordam os temas tratados no presente trabalho (escoamento miscíveis, paralelização do método dos elementos finitos via decomposição de domínio, estratégias especiais

de armazenamento e simulações em ambiente *multi-core*). Vale destacar a análise numérica do método dos elementos finitos para deslocamento miscíveis em meios porosos, realizada por Malta [38] em 1997, onde é apresentado um estudo sobre a técnica de pós-processamento para a velocidade que foi utilizada no presente trabalho. Outro destaque é dado ao estudo comparativo, utilizando formulações estabilizadas na simulação de deslocamento miscíveis em meios porosos pelo método dos elementos finitos, realizado por Ney [44], em 2002, que discretiza as equações governantes pelo método dos elementos finitos e utiliza a técnica de pós-processamento de Malta. Para as soluções em paralelo do método dos elementos finitos foi importante o estudo feito por Jimack e Touheed [29] em 2000, no qual é apresentada uma forma paralela do método dos elementos finitos utilizando decomposição de domínio. Nessa linha, encontra-se o trabalho de Lima [35], realizado em 2004, onde se faz um estudo comparativo entre estratégias especiais de armazenamento, empregando a técnica de Jimack e Touheed. Em se tratando das estratégias de armazenamento, as discussões realizadas por Martins [40] e Catabriga [10] são importantes pois apresentam a estrutura de dados aresta-por-aresta, assunto abordado no presente trabalho. Outro trabalho relevante, foi o estudo realizado por Silva Filho [19], em 2008, sobre aplicações científicas de alto desempenho em agregados de computadores *multi-core* onde ele trata a evolução e características de *Clusters* e estuda o desempenho desses sistemas distribuídos utilizando máquinas *multi-core*.

O presente trabalho está organizado em mais 5 capítulos, além dessa introdução. No Capítulo 2 é apresentado o modelo matemático que descreve o problema de escoamento miscível de fluido incompressível em meio poroso. No Capítulo 3 é descrita a formulação de elementos finitos adotada, sendo apresentadas as matrizes dos elementos e a estrutura de dados por aresta. No Capítulo 4 é apresentado um resumo sobre a computação paralela e é discutida a técnica de decomposição de domínio utilizada. No Capítulo 5 são apresentados as aplicações: injeção de traçadores e injeção contínua para validar a formulação adotada. Depois da validação, é realizado um estudo sobre a paralelização do método dos elementos finitos em diferentes *Clusters* para, em seguida, comparar as estratégias de armazenamento de matrizes esparsas, elemento-por-elemento e aresta-por-aresta. Finalmente, no Capítulo 6, são

apresentadas as conclusões e sugestões para trabalhos futuros.

Capítulo 2

Formulação Matemática

O modelo matemático que descreve o problema de escoamento miscível de um fluido incompressível em meios porosos, consiste em um sistema de equações diferenciais parciais de natureza elíptica composto pela equação da pressão, equação de concentração e Lei de Darcy, que relaciona a velocidade média de escoamento ao gradiente da pressão através do tensor de permeabilidade [50].

Para se obter esse modelo, inicialmente a equação da pressão será obtida, através da equação de conservação de massa, Lei de Darcy e uma equação de estado apropriada. A Lei de Darcy, que relaciona velocidade de escoamento do fluido ao gradiente da pressão, será usada para especificar a velocidade. Em seguida, a equação de concentração será apresentada, enfatizando a importância que as leis de viscosidade têm no comportamento do fluido. Para finalizar o modelo, as condições de contorno e iniciais são descritas.

2.1 Equações Governantes

Na formulação do modelo será utilizado um infinitésimo de meio poroso com dimensões Δx , Δy e Δz , e volume V , o qual, está ocorrendo o fluxo de um fluido composto por dois componentes, óleo e solvente, formando uma única fase, que se movimenta pelo meio com velocidade \mathbf{v} e densidade ρ . O fluxo será observado durante um

intervalo de tempo Δt . Nesse intervalo haverá entrada e saída de fluido através das faces do meio, gerando uma variação de massa, descrita pela equação de conservação de massa [54].

2.1.1 Conservação de Massa

Durante um intervalo de tempo, Δt , considere o movimento de um fluido através do meio poroso. O fluido entra no meio por uma face e sai por outra. A equação de conservação de massa descreve a variação de massa que ocorre dentro do meio devido ao fluxo [50, 54]. Significa, por exemplo, que na direção x , o fluido penetra o meio pela face x de área $A(x)$ e o deixa pela face oposta, $x + \Delta x$, cuja área é $A(x + \Delta x)$. A Fig. 2.1 ilustra esse fato.

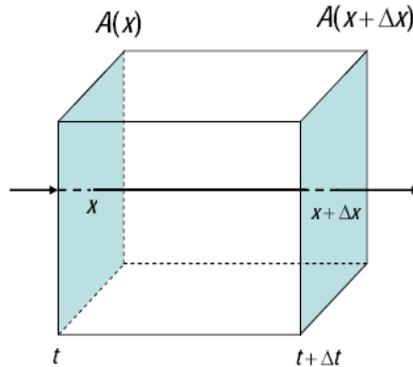


Figura 2.1: Esquema ilustrativo do fluxo no meio poroso na direção x .

A vazão de massa que entra no meio por uma face no intervalo considerado é $\rho(x)\mathbf{v}(x)A(x)\Delta t$. A vazão de massa que sai pela outra face é $\rho(x + \Delta x)\mathbf{v}(x + \Delta x)A(x + \Delta x)\Delta t$ e a taxa com que o fluido é injetado no meio é $qA(x)\Delta x\Delta t$, onde q é a taxa de injeção de massa por unidade de volume. Portanto, a massa acumulada é a vazão de entrada menos a vazão de saída, acrescida da taxa de injeção. Assim [50]:

$$\begin{aligned} \text{Massa acumulada} &= \rho(x)\mathbf{v}(x)A(x)\Delta t - \rho(x + \Delta x)\mathbf{v}(x + \Delta x)A(x + \Delta x)\Delta t \\ &\quad + qA(x)\Delta x\Delta t \end{aligned} \quad (2.1)$$

A variação de massa também pode ser obtida dentro do meio através de um balanço

de materiais. No instante t a massa do meio é representada por $(A\Delta x\phi\rho)_t$ e no instante $t + \Delta t$, por $(A\Delta x\phi\rho)_{t+\Delta t}$. A diferença entre essas duas quantidades de massa, resulta na massa acumulada durante o intervalo de tempo Δt . Assim:

$$\text{Massa acumulada} = A\Delta x[(\phi\rho)_{t+\Delta t} - (\phi\rho)_t] \quad (2.2)$$

Como há conservação de massa, as duas formas obtidas devem ser iguais. Portanto, as equações (2.1) e (2.2) podem ser igualadas o que resulta em:

$$\begin{aligned} \rho(x)\mathbf{v}(x)A(x)\Delta t - \rho(x + \Delta x)\mathbf{v}(x + \Delta x)A(x + \Delta x)\Delta t \\ + qA(x)\Delta x\Delta t = A\Delta x[(\phi\rho)_{t+\Delta t} - (\phi\rho)_t] \end{aligned} \quad (2.3)$$

Dividindo a expressão (2.3) por Δx , segue que:

$$\begin{aligned} \frac{\rho(x)\mathbf{v}(x)A(x)\Delta t}{\Delta x} - \frac{\rho(x + \Delta x)\mathbf{v}(x + \Delta x)A(x + \Delta x)\Delta t}{\Delta x} \\ + \frac{qA(x)\Delta x\Delta t}{\Delta x} = \frac{A\Delta x[(\phi\rho)_{t+\Delta t} - (\phi\rho)_t]}{\Delta x} \end{aligned} \quad (2.4)$$

Após efetuar as simplificações na igualdade (2.4), obtém-se:

$$\begin{aligned} -\frac{\rho(x + \Delta x)\mathbf{v}(x + \Delta x)A(x + \Delta x) - \rho(x)\mathbf{v}(x)A(x)}{\Delta x} + qA(x) = \\ \frac{A[(\phi\rho)_{t+\Delta t} - (\phi\rho)_t]}{\Delta t} \end{aligned} \quad (2.5)$$

Fazendo Δx e Δt tender a zero, e aplicando a definição de limite na equação (2.5), segue imediatamente que:

$$-\frac{\partial(A\rho\mathbf{v})}{\partial x} + qA = \frac{\partial(\phi\rho)A}{\partial t} \quad (2.6)$$

Portanto,

$$-\frac{\partial(\rho\mathbf{v})}{\partial x} - \frac{\partial(\phi\rho)}{\partial t} + q = 0 \quad (2.7)$$

O mesmo raciocínio pode ser feito para o movimento do fluido nas direções y e z . Fazendo isso, a equação de conservação de massa apresentará a seguinte forma:

$$-\frac{\partial(\rho\mathbf{v})}{\partial x} - \frac{\partial(\rho\mathbf{v})}{\partial y} - \frac{\partial(\rho\mathbf{v})}{\partial z} - \frac{\partial(\phi\rho)}{\partial t} + q = 0 \quad (2.8)$$

Reescrevendo a equação (2.8) de maneira mais compacta, usando a forma diferencial, a equação de conservação de massa pode ser escrita por:

$$-\nabla \cdot (\rho\mathbf{v}) - \frac{\partial(\phi\rho)}{\partial t} + q = 0 \quad (2.9)$$

O próximo passo é calcular o termo da velocidade da equação (2.9) através de uma lei que rege o movimento do fluido no meio poroso, ou seja, a Lei de Darcy.

2.1.2 Lei de Darcy

No ano de 1856, Henry Darcy realizando experiências de purificação de água através do filtro de areia, representado pela Fig. 2.2, constatou que existia uma relação direta entre a vazão que atravessava o leito de areia e a diferença de carga associada a essa vazão, ou seja, a velocidade de filtração seria proporcional ao gradiente da pressão [54].

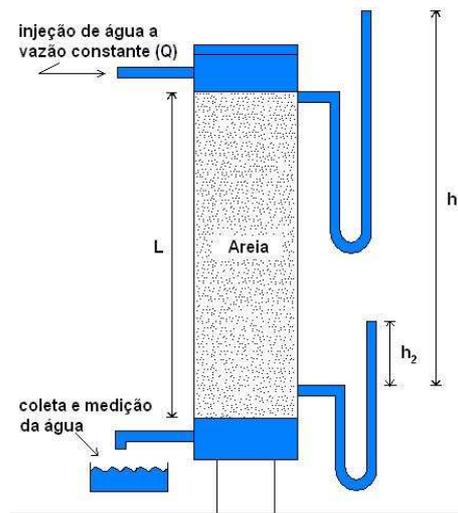


Figura 2.2: Esquema do experimento de Darcy sobre o fluxo de água através de filtro de areia.

Darcy verificou que as dimensões do leito poroso não afetavam o resultado e apresentou a relação matemática que, posteriormente, se tornaria a base para a compreensão do fenômeno de escoamento de fluidos através de meios porosos:

$$Q = \tilde{K} A \frac{h_1 - h_2}{L} \quad (2.10)$$

onde Q representa a vazão de água através do cilindro de areia, cuja seção transversal é igual a A , L é a altura do meio poroso, \tilde{K} é uma constante de proporcionalidade característica do meio poroso e do fluido, $(h_1 - h_2)$ representa a perda de carga hidráulica do fluxo através do meio poroso. As alturas h_1 e h_2 são medidas a partir de um mesmo nível de referência e representam, nessa ordem, as cargas hidráulicas na entrada e saída do filtro.

Passado algum tempo, investigadores realizando experiências com outros fluidos, descobriram que a constante \tilde{K} pode ser escrita como $\frac{K\gamma}{\mu}$, onde μ é a viscosidade do fluido, γ é o peso específico do fluido e K é a permeabilidade do meio poroso, isto é, a capacidade do meio deixar passar o fluido. Com isso a Lei de Darcy pode ser escrita por:

$$Q = \frac{KA\gamma(h_1 - h_2)}{\mu L} \quad (2.11)$$

A pressão na entrada do filtro é dada pela expressão (2.12), enquanto que na saída do filtro, seu valor é obtido pela equação (2.13):

$$p_1 = \gamma h_1 \quad (2.12)$$

$$p_2 = \gamma h_2 \quad (2.13)$$

concluindo que:

$$h_1 - h_2 = \frac{p_1 - p_2}{\gamma} \quad (2.14)$$

sendo, $p_1 - p_2$, o diferencial de pressão que está associado ao escoamento do fluido pelo meio, a Equação (2.11) pode ser escrita como [50]:

$$Q = \frac{KA\Delta p}{\mu L} \quad (2.15)$$

Considerando o fluxo em uma direção, por exemplo, na direção x , a Equação (2.15) na forma diferencial se torna:

$$\mathbf{v} = \frac{Q}{A} = -\frac{K}{\mu} \frac{\partial p}{\partial x} \quad (2.16)$$

onde \mathbf{v} é a velocidade da superfície do fluxo e $\frac{\partial p}{\partial x}$ é a variação da pressão na direção x . Generalizando, para duas dimensões, tem-se que:

$$\mathbf{v} = \frac{Q}{A} = -\frac{\mathbf{K}\nabla p}{\mu} \quad (2.17)$$

A equação (2.17) não considera a aceleração da gravidade, assim é necessário modificar essa equação a fim de acrescentar a força gravitacional. Segue que a Lei de Darcy na forma diferencial é:

$$\mathbf{v} = -\frac{\mathbf{K}}{\mu}(\nabla p - \rho g \nabla z) \quad (2.18)$$

onde $\mathbf{v} = (\mathbf{v}_x, \mathbf{v}_y)^t$ é a velocidade, \mathbf{K} é o tensor de permeabilidade que estabelece a capacidade do meio transmitir fluido, μ é a viscosidade relacionada a habilidade do fluido de suportar tensões, g é a aceleração da gravidade e ∇z é uma função vetorial da profundidade do meio.

O cálculo da velocidade de Darcy é um ponto fundamental a ser considerado na simulação de fluxo em meios porosos, pelo impacto que provoca na aproximação da solução, visto que é utilizada, tanto na equação de concentração quanto na equação da pressão.

2.1.3 Equação da Pressão

A equação da pressão é obtida introduzindo a velocidade calcula pela Lei de Darcy, representada pela expressão (2.18), na equação de conservação de massa (2.9). Procedendo dessa forma, obtém-se:

$$-\nabla \cdot \left(\rho \left(-\frac{\mathbf{K}}{\mu} (\nabla p - \rho g \nabla z) \right) \right) - \frac{\partial(\phi \rho)}{\partial t} + \tilde{q} = 0 \quad (2.19)$$

A equação (2.19) é de segunda ordem e possui duas variáveis dependentes, pressão e densidade. Para obter uma única variável dependente, pressão em função da densidade ou vice-versa, se introduzirá uma equação de estado, que está relacionada a compressibilidade do fluido, representada por \tilde{c} , ressaltando que a temperatura é considera constante [17].

A equação geral da compressibilidade dos fluidos (2.20), relaciona a variação de volume do fluido pela variação da pressão:

$$\tilde{c} = -\frac{1}{V} \frac{\partial V}{\partial p} \quad (2.20)$$

Usando o conceito de massa específica definido por:

$$\rho = \frac{m}{V} \quad (2.21)$$

onde m é a massa e V o volume do fluido. A equação (2.21) será introduzida em (2.20), resultando na expressão [17]:

$$\tilde{c} = \frac{1}{\rho} \frac{\partial \rho}{\partial p} \quad (2.22)$$

Integrando a expressão (2.22), ocorre a simplificação para:

$$\rho = \rho_0 e^{\tilde{c}(p-p_0)} \quad (2.23)$$

Considerando o mapeamento entre p e ρ descrito pelas expressões (2.22) e (2.23), pode-se escrever a equação (2.19) como:

$$\frac{\partial(\phi\rho)}{\partial t} = \nabla \cdot \left(-\frac{\mathbf{K}}{\mu\tilde{c}} (\nabla p - \rho^2 \tilde{c} g \nabla z) \right) + \tilde{q} = 0 \quad (2.24)$$

Utilizando a série de Taylor na expressão (2.23), ela se reduz a:

$$\rho = \rho_0 (1 + \tilde{c}(p - p_0)) \quad (2.25)$$

Substituindo a densidade da equação (2.25), na equação (2.24) a expressão resultante pode ser escrita por:

$$\begin{aligned} \tilde{c} \frac{\partial\phi\rho}{\partial t} &= \nabla \cdot \frac{\mathbf{K}}{\mu} (\nabla p - \rho_0 (1 + \tilde{c}(p - p_0)) g \nabla z) \\ + \tilde{c} \nabla p \cdot (\nabla p - \rho_0 (1 + \tilde{c}(p - p_0)) g \nabla z) &+ \frac{\tilde{q}}{\rho} = 0 \end{aligned} \quad (2.26)$$

Será admitido que o fluido é incompressível, $\tilde{c} = 0$, significa que a vazão \tilde{q} será a mesma ao longo de todo o meio. A equação diferencial parcial da pressão será da seguinte forma:

$$\nabla \cdot \frac{\mathbf{K}}{\mu} (\nabla p - p_0 g \nabla z) + \frac{\tilde{q}}{\rho} = 0 \quad (2.27)$$

Admitindo-se ainda que não há variação de densidade, a equação da pressão pode ser finalmente escrita como:

$$\nabla \cdot \frac{\mathbf{K}}{\mu} (\nabla p - p_0 g \nabla z) + q = 0 \quad (2.28)$$

As expressões apresentadas até agora que integram o sistema elíptico, responsável por modelar o comportamento físico do problema de deslocamento miscível de fluidos incompressíveis em meios porosos, foram a equação da pressão e a Lei de Darcy. A etapa seguinte na formulação do modelo matemático é a definição da equação da concentração.

2.1.4 Equação da Concentração

Para o desenvolvimento dessa equação é importante lembrar que o meio poroso tem volume V , e o fluido que se movimenta no meio é formado por dois componentes, óleo e solvente, que formam uma única fase. Esses componentes, óleo e solvente, serão representados, respectivamente, pelos subscritos o e s .

Inicialmente serão reunidas as expressões, taxa de massa acumulada do componente i no volume V , expressão de difusão proveniente da Lei de Fick e a lei de dispersão com uma transformação de variáveis apropriada. Em seguida, com o uso do teorema da divergência, e com algumas considerações, será obtida a equação da concentração. Detalhes podem ser consultados em [17].

Seja a taxa de massa acumulada do componente i no volume V do meio dada pela expressão (2.29) [17].

$$\frac{d}{dt} \int_V \phi \rho c_i(x, t) dx \quad i = o, s \quad (2.29)$$

onde $c_i \in [0, 1]$ é a concentração do componente i na fase. Quando se restringe a Lei de Darcy ao componente i , ela é escrita como:

$$\mathbf{v}_i = -c_i \frac{\mathbf{K}}{\mu} (\nabla p - \rho g \nabla z) = c_i \mathbf{v} \quad (2.30)$$

sendo \mathbf{v}_i a velocidade do componente i na fase. O deslocamento miscível é caracterizado tanto pelo processo de difusão, quanto pelo de dispersão. A difusão é governada pela Lei de Fick, que em linhas gerais, é o resultado do movimento aleatório das moléculas no fluido. Esse movimento é representado pela expressão:

$$\frac{\partial V}{\partial t} = -\alpha_m A \frac{\partial C}{\partial x} \quad (2.31)$$

sendo V o volume do fluido que passa pelo meio poroso na direção x , cuja face de entrada do meio tem área A , C a concentração do fluido e α_m o coeficiente de difusão molecular. Da Lei de Fick (2.31), obtem-se a taxa de fluxo difusivo representada por:

$$\mathbf{v}_{i,diff} = -\phi \alpha_m \nabla c_i \quad (2.32)$$

A taxa de fluxo dispersiva é obtida da lei de dispersão e pode ser escrita como:

$$\mathbf{v}_{i,disp} = -\alpha_l |\mathbf{v}_i| \frac{\partial c_i}{\partial \hat{e}_l} - \alpha_t |\mathbf{v}_i| \frac{\partial c_i}{\partial \hat{e}_t} \quad (2.33)$$

onde α_l é o coeficiente de dispersão longitudinal, α_t é o coeficiente de dispersão transersal, \hat{e}_l e \hat{e}_t são vetores unitários paralelos e perpendiculares a \mathbf{v} , respectivamente.

Convenientemente, será realizada uma transformação de coordenadas na equação (2.33), se valendo das relações:

$$\begin{aligned}\hat{e}_l &= \hat{e}_x \cos(\theta) + \hat{e}_y \sin(\theta) \\ \hat{e}_t &= -\hat{e}_x \sin(\theta) + \hat{e}_y \cos(\theta) \\ \cos(\theta) &= \frac{\mathbf{v}_x}{|\mathbf{v}|} \\ \sin(\theta) &= \frac{\mathbf{v}_y}{|\mathbf{v}|}\end{aligned}\tag{2.34}$$

onde θ é o ângulo entre \hat{e}_l e \hat{e}_x que é o vetor unitário cartesiano.

Utilizando as relações de transformação de coordenadas (2.34) na lei de dispersão (2.33), é obtido o seguinte resultado:

$$\begin{aligned}\mathbf{v}_{i,disp} &= -\alpha_l |\mathbf{v}| \left(\frac{\mathbf{v}_x}{|\mathbf{v}|} \frac{\partial c_i}{\partial x} + \frac{\mathbf{v}_y}{|\mathbf{v}|} \frac{\partial c_i}{\partial y} \right) \left(\frac{\mathbf{v}_x}{|\mathbf{v}|} \hat{e}_x + \frac{\mathbf{v}_y}{|\mathbf{v}|} \hat{e}_y \right) \\ &\quad - \alpha_t |\mathbf{v}| \left(-\frac{\mathbf{v}_y}{|\mathbf{v}|} \frac{\partial c_i}{\partial x} + \frac{\mathbf{v}_x}{|\mathbf{v}|} \frac{\partial c_i}{\partial y} \right) \left(-\frac{\mathbf{v}_y}{|\mathbf{v}|} \hat{e}_x + \frac{\mathbf{v}_x}{|\mathbf{v}|} \hat{e}_y \right)\end{aligned}$$

simplificando,

$$\begin{aligned}\mathbf{v}_{i,disp} &= -\frac{-\alpha_l}{|\mathbf{v}|} \left[(-\mathbf{v}_x^2 \frac{\mathbf{c}_i}{\partial x} + \mathbf{v}_x \mathbf{v}_y \frac{\partial c_i}{\partial y}) \hat{e}_x + (\mathbf{v}_x \mathbf{v}_y \frac{\mathbf{c}_i}{\partial x} + \mathbf{v}_y^2 \frac{\partial c_i}{\partial y}) \hat{e}_y \right] \\ &\quad - \frac{\alpha_t}{|\mathbf{v}|} \left[(\mathbf{v}_y^2 \frac{\mathbf{c}_i}{\partial x} - \mathbf{v}_x \mathbf{v}_y \frac{\partial c_i}{\partial y}) \hat{e}_x + (-\mathbf{v}_x \mathbf{v}_y \frac{\mathbf{c}_i}{\partial x} + \mathbf{v}_x^2 \frac{\partial c_i}{\partial y}) \hat{e}_y \right]\end{aligned}\tag{2.35}$$

Combinando as equações (2.29), (2.31), (2.35) e utilizando o teorema da divergência, tem-se que a equação de concentração para fontes e sumidouros é:

$$\int_V \frac{\partial}{\partial t} (\phi \rho c_i) dx = - \int_V \nabla \cdot (\rho c_i \mathbf{v} + \rho \mathbf{v}_{i,diff} + \rho \mathbf{v}_{i,disp}) dx + \int_V \tilde{c}_i \rho q dx\tag{2.36}$$

onde \tilde{c}_i é tanto a concentração do componente i no poço injetor quanto a concentração residente no poço produtor. Assumindo incompressibilidade do fluido e da rocha a equação de concentração para fontes e sumidouros (2.36) resulta em:

$$\int_V \left[\phi \frac{\partial c_i}{\partial t} - \nabla \cdot (\mathbf{D} \nabla c_i - \mathbf{v} c_i) \right] dx = \int_V \tilde{c}_i q dx\tag{2.37}$$

sendo \mathbf{D} o tensor de difusão-dispersão que é definido pela expressão [44]:

$$\mathbf{D} = \begin{bmatrix} \alpha_m & 0 \\ 0 & \alpha_m \end{bmatrix} + \frac{\alpha_l}{|\mathbf{v}|} \begin{bmatrix} \mathbf{v}_x \mathbf{v}_x & \mathbf{v}_x \mathbf{v}_y \\ \mathbf{v}_y \mathbf{v}_x & \mathbf{v}_y \mathbf{v}_y \end{bmatrix} + |\mathbf{v}| \begin{bmatrix} \alpha_t & 0 \\ 0 & \alpha_t \end{bmatrix} - \frac{\alpha_t}{|\mathbf{v}|} \begin{bmatrix} \mathbf{v}_x \mathbf{v}_x & \mathbf{v}_x \mathbf{v}_y \\ \mathbf{v}_y \mathbf{v}_x & \mathbf{v}_y \mathbf{v}_y \end{bmatrix} \quad (2.38)$$

onde α_m é o coeficiente de difusão molecular, α_l é o coeficiente de dispersão longitudinal e α_t o coeficiente de dispersão transversal. Será utilizada a notação:

$\mathbf{v}_x \mathbf{v}_x = \mathbf{v}_{xx}$, $\mathbf{v}_x \mathbf{v}_y = \mathbf{v}_{xy}$ e $\mathbf{v}_y \mathbf{v}_y = \mathbf{v}_{yy}$ logo,

$$\mathbf{D} = \frac{1}{|\mathbf{v}|} \begin{bmatrix} \alpha_m |\mathbf{v}| + \alpha_t |\mathbf{v}|^2 + (\alpha_l - \alpha_t) \mathbf{v}_{xx} & (\alpha_l - \alpha_t) \mathbf{v}_{xy} \\ (\alpha_l - \alpha_t) \mathbf{v}_{xy} & \alpha_m |\mathbf{v}| + \alpha_t |\mathbf{v}|^2 + (\alpha_l - \alpha_t) \mathbf{v}_{yy} \end{bmatrix} \quad (2.39)$$

Visando facilitar as operações com o tensor de difusão-dispersão \mathbf{D} descrito em (2.38), ele será representado por:

$$\mathbf{D} = \begin{bmatrix} d_{11} & d_{12} \\ d_{21} & d_{22} \end{bmatrix} \quad (2.40)$$

valendo as relações:

$$\begin{aligned} d_{11} &= \frac{1}{|\mathbf{v}|} (\alpha_m |\mathbf{v}| + \alpha_t |\mathbf{v}|^2 + (\alpha_l - \alpha_t) \mathbf{v}_{xx}) \\ d_{12} &= \frac{1}{|\mathbf{v}|} (\alpha_l - \alpha_t) \mathbf{v}_{xy} \\ d_{21} &= \frac{1}{|\mathbf{v}|} (\alpha_l - \alpha_t) \mathbf{v}_{xy} \\ d_{22} &= \frac{1}{|\mathbf{v}|} (\alpha_m |\mathbf{v}| + \alpha_t |\mathbf{v}|^2 + (\alpha_l - \alpha_t) \mathbf{v}_{yy}) \end{aligned} \quad (2.41)$$

Fazendo V um ponto arbitrário e se V é menor que a média dos volumes, então é possível retirar a integral da equação (2.37), portanto o sistema acoplado resultante é [17]:

$$\phi \frac{\partial c_i}{\partial t} - \nabla \cdot (\mathbf{D} \nabla c_i - \mathbf{v} c_i) = \tilde{c}_i q \quad i = o, s \quad (2.42)$$

As variáveis dependentes são \mathbf{v} e c_i . As equações constitutivas desse sistema são chamadas de equação do óleo e equação do solvente. Uma manipulação comum é somar essas equações e utilizar o fato que $c_o + c_i = 1$, isso para acumular os termos difusivos e convectivos no sistema:

$$-\nabla \cdot \left(\frac{\mathbf{K}}{\mu(c)} (\nabla p - \rho g \nabla z) \right) = \nabla \cdot \mathbf{v} - q \quad (2.43)$$

$$\phi \frac{\partial c}{\partial t} - \nabla \cdot (\mathbf{D}(\mathbf{v}) \nabla c - \mathbf{v}c) = \tilde{c}q \quad (2.44)$$

onde $c = c_s$ e $\tilde{c} = \tilde{c}_s$. A equação (2.43) equivale a equação da pressão, já obtida em (2.28). A expressão (2.44) representa a equação da concentração do solvente.

$$\phi \frac{\partial c}{\partial t} + \nabla \cdot (c\mathbf{v}) - \nabla \cdot (\mathbf{D}(\mathbf{v}) \nabla c) = \tilde{c}q \quad (2.45)$$

Usando a regra de Leibniz, $\nabla \cdot (c\mathbf{v}) = \mathbf{v} \cdot \nabla c + c \nabla \cdot \mathbf{v} = \mathbf{v} \cdot \nabla c + cq$, é possível escrever a equação da concentração (2.45) por:

$$\phi \frac{\partial c}{\partial t} + \mathbf{v} \cdot \nabla c - \nabla \cdot (\mathbf{D}(\mathbf{v}) \nabla c) + cq = \hat{c}q \quad (2.46)$$

onde q é uma fonte externa, escrita por $q = q_i - q_0$, sendo $q = \max(0, q_i) \geq 0$ e $q = \max(0, -q_0) \geq 0$. A fonte externa q é diferente de zero apenas no poço injetor e no poço produtor. Além disso \hat{c} é dado por [37]:

$$\hat{c} = \begin{cases} c_i & \text{no poço injetor onde } q = q_i \geq 0 \\ c & \text{no poço produtor onde } q = -q_0 \geq 0 \end{cases}$$

sendo c_i a concentração prescrita de fluido injetado. Portanto, como a concentração injetada no poço injetor e a concentração residente no poço produtor são iguais, segue que a equação de concentração pode ser reescrita por:

$$\phi \frac{\partial c}{\partial t} + \mathbf{v} \cdot \nabla c - \nabla \cdot (\mathbf{D}(\mathbf{v}) \nabla c) = 0 \quad (2.47)$$

Na equação (2.43) o comportamento da viscosidade do fluido μ é não-linear e pode ser obtido por leis empíricas, como por exemplo [62]:

$$\mu(c) = e^{Rc} \quad (2.48)$$

onde R é uma constante tal que $R = \ln(M)$, sendo M a mobilidade do fluido, isto é, a razão entre a viscosidade do fluido residente e o fluido deslocante:

$$M = \frac{\mu_r}{\mu_d} = \frac{\mu_o}{\mu_s} = \frac{\mu(0)}{\mu(1)} \quad (2.49)$$

Outra expressão muito popular, encontrada na literatura para a viscosidade, é representada por [68]:

$$\mu(c) = (1 - c + M^{0.25}c)^{-4} \mu_r \quad (2.50)$$

Em ambas as leis (2.48) e (2.50), quando $M > 1$, tem-se que a viscosidade do fluido deslocante, no caso o solvente, é menor que o fluido residente, ou seja o óleo. Portanto, pode ocorrer uma instabilidade nas frentes de concentração, ocasionando a formação dos dedilamentos viscosos ("viscous fingers") [17].

Os dedilamentos viscosos, como o próprio nome sugere, são deformações no óleo, semelhantes a longos dedos provocadas pelo fluido deslocante. Essas deformações ocorrem devido a instabilidade do encontro entre os fluidos. Usualmente, o fluido injetado no reservatório para expelir o óleo residente é menos viscoso. Assim na interface entre os dois fluidos, o menos viscoso invade o óleo, gerando formas semelhante a longos dedos. Quando um desses dedos atinge o poço produtor, o fluido deslocante prioriza esse caminho e a extração do óleo fica comprometida, pois o fluido injetado através desses caminhos preferenciais se dirige rapidamente para o poço produtor, não se propagando de forma adequada no reservatório, deixando grandes volumes de hidrocarboretos nos locais onde o deslocamento não ocorreu [50, 66].

No caso de deslocamento totalmente miscíveis, mesmo não havendo interface entre dois fluidos, ainda pode ocorrer dedilamentos viscosos, devido aos caminhos preferenciais que o fluido segue no meio poroso. Além disso, nesse mesmo deslocamento, quando a mobilidade é adversa, isto é, $M > 1$, pode ocorrer instabilidades numéricas, que decorrem da não linearidade introduzida pela lei da viscosidade. Como a lei da viscosidade está expressa em termos de M , e essa lei acopla o sistema de equações formados pela equação da pressão (2.18), a Lei da Darcy (2.28) e a equação da concentração (2.47), então pode-se pensar que quanto maior o valor de M , maior é o acoplamento do sistema, e maior a ocorrência das oscilações numéricas.

A equação da pressão (2.28), a Lei da Darcy (2.18) e a equação da concentração (2.47), compõem o modelo matemático que descreve o comportamento do escoamento miscível do fluido incompressível injetado no meio poroso. Descrito apropri-

adadamente por,

$$\nabla \cdot (\mathbf{A}(c) \nabla p) + q = 0 \quad (2.51)$$

$$\mathbf{v} = -\mathbf{A}(c) \nabla p \quad (2.52)$$

$$\phi \frac{\partial c}{\partial t} + \mathbf{v} \cdot \nabla c - \nabla \cdot (\mathbf{D}(\mathbf{v}) \nabla c) = 0 \quad (2.53)$$

onde \mathbf{v} é a velocidade, q representa os termos fontes e sumidouros, c é a concentração, p é a pressão e ϕ é a porosidade do meio. A equação da pressão também pode ser representada por:

$$\nabla \cdot (-\mathbf{A}(c) \nabla p) - q = 0 \quad (2.54)$$

onde o tensor $\mathbf{A}(c)$ é,

$$\mathbf{A}(c) = \frac{\mathbf{K}}{\mu(c)} \quad (2.55)$$

sendo \mathbf{K} o tensor de permeabilidade e $\mu(c)$ a viscosidade da mistura. O tensor de permeabilidade é dado por:

$$\mathbf{K} = \begin{bmatrix} \lambda_{11} & \lambda_{12} \\ \lambda_{21} & \lambda_{22} \end{bmatrix}$$

Portanto, $\mathbf{A}(c)$ será representado por:

$$\mathbf{A} = \begin{bmatrix} \Lambda_{11} & \Lambda_{12} \\ \Lambda_{21} & \Lambda_{22} \end{bmatrix} \quad (2.56)$$

onde $\Lambda_{ij} = \frac{\lambda_{ij}}{\mu(c)}$ para $i, j = 1, 2$.

Como pode ser observado, esse modelo é representado por um sistema acoplado de equações diferenciais transiente e não-linear. Para que o problema fique completamente descrito é necessário especificar um conjunto apropriado de condições de contorno e condições iniciais.

2.2 Condições de Contorno e Condições Iniciais

Considere Ω uma região limitada de R^2 com fronteira Γ . A fronteira Γ do domínio Ω pode apresentar regiões de fronteira relacionada a poços injetores, a poços produtores, a regiões com pressão prescrita e a regiões com fluxo nulo. Assim, a fronteira

do domínio pode ser subdividida de forma que $\Gamma = \Gamma_i \cup \Gamma_p \cup \Gamma_d \cup \Gamma_n$, onde os subscritos i, p, d, n denotam, respectivamente, o poço injetor, o poço produtor, pressão prescrita e fluxo nulo [44, 41]. A Fig. 2.3 ilustra as subdivisões das condições de contorno.

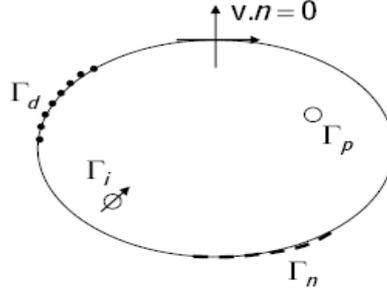


Figura 2.3: Descrição do contorno em um problema de meios porosos.

As condições de contorno associadas a equação da pressão podem ser tanto de Dirichlet quanto de Neumann. A pressão é prescrita nas regiões onde não existe vazão. As condições de contorno de Neumann são especificadas para o fluxo de massa através do contorno, assim, o vetor velocidade pelo vetor normal, equivale a uma função que relaciona comprimento e tempo [17]. No presente modelo, essa função é nula. Para a equação da concentração, apenas a concentração no poço injetor é especificada como condição de contorno. Portanto, as condições de contorno adotadas são,

$$\mathbf{v} \cdot \mathbf{n} = 0 \quad \text{em } \Gamma_n, \forall t \in [0, T] \quad (2.57)$$

$$\mathbf{D}(\mathbf{v}) \nabla c \mathbf{n} = 0 \quad \text{em } \Gamma_n \quad (2.58)$$

$$c(x, t) = \tilde{c} \quad \text{em } \Gamma_i \quad (2.59)$$

As condições iniciais empregadas para as equações da pressão, concentração e velocidade são:

$$p(x, 0) = p_0 \quad \text{em } \Omega \quad (2.60)$$

$$c(x, 0) = c_0(x) \quad \text{em } \Omega \quad (2.61)$$

$$\mathbf{v}(x, 0) = 0 \quad \text{em } \Omega \quad (2.62)$$

As condições de contorno e iniciais, descritas até agora, estão postas de forma

genérica. Detalhes serão abordados posteriormente, quando os problemas propostos forem apresentados .

Reunindo as equações da pressão, velocidade, concentração, condições iniciais e de contorno, o problema pode ser completamente descrito através do modelo matemático. O fato desse modelo ser composto por um sistema de equações diferenciais parciais, acopladas, dependente do tempo e não-linear inviabiliza uma solução analítica satisfatória. Assim, serão utilizadas técnicas numéricas para obter uma solução aproximada.

Capítulo 3

Formulação Numérica

A solução analítica do modelo representado pelas equações (2.51), (2.52) e (2.53) dificilmente pode ser obtida. No entanto existem métodos numéricos estáveis o suficiente para se produzir uma solução aproximada, que represente satisfatoriamente o fenômeno físico. As principais técnicas numéricas utilizadas para resolver tais problemas são: diferenças finitas, volumes finitos e elementos finitos.

Nesse trabalho, o método dos elementos finitos será adotado para se obter uma solução aproximada do modelo matemático descrito anteriormente, devido a sua flexibilidade em solucionar problemas envolvendo geometrias complexas. Será utilizada uma formulação variacional estabilizada semi-discreta que consiste em uma discretização de elementos finitos no espaço seguido de uma discretização de diferenças finitas no tempo.

3.1 Formulação Variacional Semi-discreta

A discretização de elementos finitos no espaço, considera a divisão do domínio do problema Ω em nel elementos finitos, de forma que $\Omega = \bigcup_{e=1}^{nel} \Omega^e$ e $\bigcap_{e=1}^{nel} \Omega^e = \emptyset$, sendo nel o número de elementos totais da malha.

São estabelecidos dois grupos de funções. O primeiro grupo é composto por funções candidatas a solução, que atendam as condições de contorno e sejam quadrado in-

tegráveis. Deseja-se obter a solução de três grandezas: pressão, velocidade e concentração. Portanto, são definidos grupos de funções candidatas a solução para a pressão p , para a velocidade \mathbf{v} e para a concentração c . O segundo grupo é formado por funções similares às funções testes, representadas por w , que diferem das funções anteriores, por assumirem o valor nulo no contorno. As funções $p \in P$, $\mathbf{v} \in U$ e $c \in C$ são aproximadas por funções de dimensões finitas $p^h \in P^h \subset P$, $\mathbf{v}^h \in V^h \subset V$, $c^h \in C^h \subset C$ e $w^h \in W^h \subset W$ representadas por:

$$\begin{aligned}
P^h &= \{p^h | p^h \in H^1, p^h = p_p \text{ em } \Gamma_p\} \\
U^h &= \{\mathbf{v}^h | \mathbf{v}^h \in H^1, \mathbf{v}^h = v_n \text{ em } \Gamma_n\} \\
C^h &= \{c^h | c^h \in H^1, c^h = g_i \text{ em } \Gamma_i\} \\
W^h &= \{w^h | w^h \in H^1, w^h = 0 \text{ em } \Gamma\}
\end{aligned} \tag{3.1}$$

onde H^1 representa o conjunto das funções quadradas integráveis admissíveis para o sistema, aproximadas em um espaço de dimensão finita.

Na discretização espacial de elementos finitos é importante considerar as particularidades e características de cada equação para se escolher uma formulação adequada, aumentando dessa forma, a qualidade da aproximação. Geralmente, a formulação clássica de Galerkin é suficiente para obter uma boa solução quando o problema é predominantemente difusivo. No entanto, quando o problema é predominantemente convectivo, ou seja, o termo difusivo é muito pequeno comparado ao termo convectivo, a formulação de Galerkin não gera bons resultados, devido as oscilações espúrias proveniente de instabilidades próprias dessa formulação [4]. Para contornar a limitação da formulação de Galerkin usa-se uma formulação estabilizada que acrescenta difusão na direção de escoamento para evitar essas oscilações. A formulação *Streamline Upwind Petrov-Galerkin* (SUPG) apresenta uma boa melhora sobre a formulação pura de Galerkin para equações com esse comportamento [4]. Contudo, ainda pode ocorrer soluções descontínuas nas vizinhanças de regiões com alto gradiente, nesse caso, é preciso um termo adicional que atue somente nessas regiões tornando a solução suave. O operador de captura de choque, *Consistent Approximate Upwind* (CAU), atua nesse sentido [23]. Um outro ponto relevante é quanto a ordem de aproximação das equações que devem possuir a mesma precisão para

uma solução satisfatória. Assim, se for usada uma aproximação de ordem k para a pressão e a velocidade for calculada diretamente pela Lei de Darcy, essa velocidade terá ordem de aproximação $k-1$, o que compromete diretamente a precisão da concentração. Para contornar esse problema existem diversos esquemas que podem ser utilizados. Uma boa estratégia é o pós-processamento da velocidade proposto por Malta [37].

Como o comportamento da equação da pressão é predominantemente difusivo, será utilizada a formulação clássica de Galerkin que gera boas aproximações. Assim, a equação da pressão (2.54) é dada por:

$$\int_{\Omega} \nabla w^h \mathbf{A}(c) \nabla p^h d\Omega = \int_{\Omega} w^h q d\Omega \quad (3.2)$$

No entanto, para a equação de concentração, que apresenta um comportamento convectivo, a formulação de Galerkin é inadequada por apresentar oscilações numéricas. Portanto, nessa equação será utilizada a formulação estabilizada SUPG que acrescenta uma quantidade necessária de difusão na direção do escoamento para eliminar as instabilidades.

Na formulação de Galerkin as funções peso são consideradas contínuas ao longo das fronteiras entre os elementos. Entretanto, na formulação SUPG as funções testes são compostas por duas partes: uma contínua e a outra descontínua, essa última, visando reduzir as oscilações que podem vir a ocorrer na direção do movimento do fluido. Seja então:

$$\widetilde{w}^h = w^h + \widetilde{p}^h \quad (3.3)$$

onde $w^h \in V$ é a função teste contínua e \widetilde{p}^h é a contribuição descontínua no contorno, ambas suaves no interior do elemento. A contribuição descontínua pode ser definida como [4]:

$$\widetilde{p}^h = \tau^e \mathbf{v} \cdot \nabla w^h \quad (3.4)$$

onde o parâmetro de estabilização SUPG é calculado segundo a expressão:

$$\tau^e = \frac{1}{2\|\mathbf{v}\|} h^e \min\left(\frac{Pe}{3}, 1.0\right) \quad (3.5)$$

sendo \mathbf{v} a velocidade total no elemento, h^e uma estimativa do tamanho do elemento, representado pela raiz quadrada do dobro da área do elemento, isto é, $h^e = \sqrt{2A^e}$

e Pe o número de Peclet dado pela seguinte expressão:

$$Pe = \frac{1}{2} h^e \frac{\|\mathbf{v}\|^3}{(\mathbf{v})^T \mathbf{D} \mathbf{v}} \quad (3.6)$$

Empregando a estabilização SUPG na equação da concentração (2.53) temos:

$$\sum_{e=1}^{nel} \int_{\Omega^e} \tilde{w}^h \left(\phi \frac{\partial c^h}{\partial t} + \mathbf{v} \cdot \nabla c^h + \nabla \cdot \mathbf{D} \nabla c^h \right) d\Omega = 0 \quad (3.7)$$

Apesar da formulação SUPG conduzir a uma solução numérica estável, ela não evita oscilações próximas às regiões com elevados gradientes. Portanto, é necessário usar métodos de captura de descontinuidade para aumentar a dissipação numérica onde a solução não é suave. O operador de captura CAU evita oscilações próximas as descontinuidades, que ocorrem nas regiões com alto gradiente e desaparece nas regiões onde a solução é suave [23].

Os efeitos espúrios próximos as descontinuidades serão suavizados acrescentando uma pequena parcela não-linear ao problema. Segue que a equação da concentração (2.51) será acrescida do operador CAU definidor por:

$$\int_{\Omega^e} \delta_{cau}^e \nabla w^h \nabla c^h d\Omega \quad (3.8)$$

O parâmetro de difusão δ^e é não-linear e pode ser definido como [1, 12, 13, 44]:

$$\delta_{cau}^e = \frac{1}{2} h \min\left(\frac{Pe_p}{3}, 0, 7\right) \frac{|L(c^h)|}{\|\nabla c^h\|} \quad (3.9)$$

$$Pe_p = \frac{1}{2} h^e \frac{\|\mathbf{v}_p\|^3}{(\mathbf{v}_p^T) \mathbf{D} \mathbf{v}_p} \quad (3.10)$$

$$\mathbf{v}_p = \frac{\mathbf{v} \cdot \nabla c^h}{\|\nabla c^h\|^2} \nabla c^h \quad (3.11)$$

onde Pe_p é o número de Peclet correspondente a \mathbf{v}_p^h , que é a velocidade projetada na direção paralela ao gradiente da solução ∇c^h e o operador $|L(c^h)|$ é o valor absoluto do resíduo no interior do elemento definido por todos os termos da equação (2.53) logo:

$$|L(c^h)| = \left| \phi \frac{\partial c^h}{\partial t} + \mathbf{v} \cdot \nabla c^h - \nabla \cdot (\mathbf{D}(\mathbf{v}) \nabla c^h) \right| \quad (3.12)$$

Ao substituir a função teste (3.3) na equação da concentração (3.7) e acrescentar o operador de captura (3.8) o resultado é a formulação estabilizada da equação

de concentração (3.13) que possui três contribuições distintas: Galerkin, Petrov-Galerkin e CAU. Assim,

$$\begin{aligned}
& \underbrace{\int_{\Omega} \tilde{w}^h \left(\phi \frac{\partial c^h}{\partial t} + \mathbf{v}^h \cdot \nabla c^h + \nabla \cdot \mathbf{D} \nabla c^h \right) d\Omega}_{GALERKIN} + \\
& \underbrace{\sum_{e=1}^{nel} \int_{\Omega^e} \tilde{p}^h \left(\phi \frac{\partial c^h}{\partial t} + \mathbf{v}^h \cdot \nabla c^h + \nabla \cdot \mathbf{D} \nabla c^h \right) d\Omega}_{PETROV-GALERKIN} + \\
& \underbrace{\sum_{e=1}^{nel} \int_{\Omega^e} \delta_{cau}^e \nabla w^h \nabla c^h d\Omega}_{CAU} = 0 \quad (3.13)
\end{aligned}$$

Aplicando a integração por partes e o teorema da divergência na expressão (3.13) e usando a definição da função teste descontínua em (3.4) para cada contribuição da equação (3.13), obtém-se:

$$\begin{aligned}
& \underbrace{\int_{\Omega} w^h \phi \frac{\partial c^h}{\partial t} d\Omega + \int_{\Omega} w^h \mathbf{v}^h \cdot \nabla c^h d\Omega + \int_{\Omega} \nabla w^h \cdot \mathbf{D} \nabla c^h d\Omega}_{GALERKIN} + \\
& \underbrace{\sum_{e=1}^{nel} \tau^e \left(\int_{\Omega^e} \mathbf{v}^h \cdot \nabla w^h \phi \frac{\partial c^h}{\partial t} d\Omega + \int_{\Omega^e} \mathbf{v}^h \cdot \nabla w^h \mathbf{v}^h \cdot \nabla c^h d\Omega - \int_{\Omega^e} \nabla (\mathbf{v}^h \cdot \nabla w^h) \cdot \mathbf{D} \nabla c^h d\Omega \right)}_{PETROV-GALERKIN} \\
& + \underbrace{\sum_{e=1}^{nel} \int_{\Omega^e} \delta_{cau}^e \nabla w^h \nabla c^h d\Omega}_{CAU} = 0 \quad (3.14)
\end{aligned}$$

Cada parcela da equação (3.14) recebe uma denominação específica conforme sua

representação física, isto é:

$$\text{massa:} \quad \int_{\Omega} w^h \phi \frac{\partial c^h}{\partial t} d\Omega \quad e \quad \sum_{e=1}^{nel} \tau^e \int_{\Omega^e} \mathbf{v}^h \cdot \nabla w^h \phi \frac{\partial c^h}{\partial t} d\Omega \quad (3.15)$$

$$\text{advecção:} \quad \int_{\Omega} w^h \mathbf{v}^h \cdot \nabla c^h d\Omega \quad e \quad \sum_{e=1}^{nel} \tau^e \int_{\Omega^e} \mathbf{v}^h \cdot \nabla w^h \mathbf{v}^h \cdot \nabla c^h d\Omega \quad (3.16)$$

$$\text{difusão:} \quad \int_{\Omega} \nabla w^h \cdot \mathbf{D} \nabla c^h d\Omega \quad e \quad \sum_{e=1}^{nel} \tau^e \int_{\Omega^e} \nabla (\mathbf{v}^h \cdot \nabla w^h) \cdot \mathbf{D} \nabla c^h d\Omega \quad (3.17)$$

$$\text{CAU:} \quad \int_{\Omega_e} \delta_{cau}^e \nabla w^h \nabla c^h d\Omega \quad (3.18)$$

Garantida a estabilidade é preciso assegurar a precisão da solução. É fundamental que todas as grandezas envolvidas no processo de solução do problema tenham a mesma ordem de precisão. Para obter uma solução com maior ordem de acurácia e mesma ordem de aproximação são usados esquemas de pós-processamento para a velocidade visto que, quando calculada diretamente pela Lei de Darcy sua precisão é menor do que as outras variáveis [37]. A estratégia de pós-processamento adotada utiliza a formulação variacional da Lei de Darcy combinada com a equação da pressão. A Lei de Darcy, dada em (2.52), pode ser representada pela equação:

$$\mathbf{A}(c)^{-1} \mathbf{v} + \nabla p = 0 \quad (3.19)$$

onde o tensor $\mathbf{A}(c)^{-1}$ é definido através de (2.55) e (2.56) que resulta em:

$$\mathbf{A}(c)^{-1} = \frac{1}{\det(\mathbf{A})} \begin{bmatrix} \Lambda_{22} & -\Lambda_{12} \\ -\Lambda_{21} & \Lambda_{11} \end{bmatrix} \quad (3.20)$$

No desenvolvimento da formulação da velocidade com pós-processamento são combinadas a formulação variacional da lei de Darcy (3.19) e a equação da pressão (2.51), também denominada de resíduo de balanço de massa, detalhes podem ser vistos em [37]. O pós-processamento do campo de velocidades consiste em encontrar \mathbf{v}^h tal que:

$$\int_{\Omega} \mathbf{w}^h \cdot (\mathbf{A}(c^h)^{-1} \mathbf{v}^h + \nabla p^h) d\Omega + \sum_{e=1}^{nel} \delta_{pp}^e \int_{\Omega} \nabla \cdot \mathbf{w}^h (\nabla \cdot \mathbf{v}^h - q) d\Omega = 0 \quad (3.21)$$

onde o parâmetro $\delta_{pp}^e = M^{1/2} \frac{h_e}{2}$.

Cada parcela da expressão (3.21) realiza alguma ponderação específica, em relação, à forma variacional da Lei de Darcy, ao divergente da velocidade, ao gradiente da pressão e ao divergente de fluxo.

$$\text{Lei de Darcy:} \quad \int_{\Omega} \mathbf{w}^h \cdot \mathbf{A}(c^h)^{-1} \mathbf{v}^h d\Omega \quad (3.22)$$

$$\text{Divergente da velocidade:} \quad \sum_{e=1}^{nel} \delta_{pp}^e \int_{\Omega} \nabla \cdot \mathbf{w}^h \nabla \cdot \mathbf{v}^h d\Omega \quad (3.23)$$

$$\text{Gradiente da pressão:} \quad \int_{\Omega} \mathbf{w}^h \cdot \nabla p^h d\Omega \quad (3.24)$$

$$\text{Divergente de fluxo:} \quad - \sum_{e=1}^{nel} \delta_{pp}^e \int_{\Omega} \nabla \cdot \mathbf{w}^h q d\Omega \quad (3.25)$$

Realizada a discretização, a solução por elementos finitos é obtida aproximando as funções incógnitas e peso, ambas contínuas, por funções discretas nodais dadas por:

$$w^h(x) = \sum_{i \in \eta - \eta_c} d_i N_i(\mathbf{x}) \quad (3.26)$$

$$p^h(x, t) = \sum_{i \in \eta - \eta_c} p_i(t) N_i(\mathbf{x}) \quad (3.27)$$

$$c^h(x, t) = \sum_{i \in \eta - \eta_c} c_i(t) N_i(\mathbf{x}) \quad (3.28)$$

$$\frac{\partial c^h}{\partial t}(x, t) = \sum_{i \in \eta - \eta_c} a_i(t) N_i(\mathbf{x}) \quad (3.29)$$

$$\mathbf{v}^h(x, t) = \sum_{i \in \eta - \eta_c} \mathbf{v}_i(t) \tilde{N}_i(\mathbf{x}) \quad (3.30)$$

onde η é o número de nós da malha, η_c são os nós do contorno da malha, d_i é uma constante, a_i é a derivada de c_i em relação ao tempo, N_i é uma função de interpolação usual do nó i para concentração e pressão, \tilde{N}_i é uma função de interpolação usual do nó i para velocidade, p_i , c_i e \mathbf{v}_i são respectivamente pressão, concentração e velocidade relativa ao nó i e $\mathbf{x} = (x, y)$.

Decorrente da aproximação por elementos finitos, a partir das expressões para pressão (3.2), para velocidade (3.21) e para concentração (3.14), é originado um sistema acoplado de equações que podem ser dividido em 3 sistemas. O primeiro é um sistema linear de equações e diz respeito a equação da pressão, dado por:

$$\mathbf{Kp} = \mathbf{Q} \quad (3.31)$$

sendo \mathbf{p} o vetor de incógnita das pressões, \mathbf{K} a matriz obtida da substituição de (3.26) e (3.27) no primeiro termo de (3.2) e \mathbf{Q} o vetor decorrente da substituição de (3.26) no segundo termo de (3.2).

O segundo sistema linear gerado está relacionado a equação da velocidade, sendo representado por:

$$\overline{\mathbf{M}}\mathbf{v} = \mathbf{F} \quad (3.32)$$

onde $\overline{\mathbf{M}}$ é obtida pela substituição das funções nodais (3.26) e (3.30) nas expressões (3.22) e (3.23), \mathbf{F} é originado com a substituição de (3.26) e (3.30) em (3.24) e (3.25), e \mathbf{v} é o vetor de incógnita das velocidades.

O terceiro sistema, é um sistema de equações diferenciais ordinárias e refere-se a equação da concentração, sendo representado por:

$$\widetilde{\mathbf{M}}\mathbf{a} + \widetilde{\mathbf{C}}\mathbf{c} = 0 \quad (3.33)$$

onde \mathbf{c} é o vetor de incógnita das concentrações, $\mathbf{a} = \frac{\partial \mathbf{c}}{\partial t}$ é o vetor que contém os valores da derivada de \mathbf{c} , $\widetilde{\mathbf{M}}$ conhecida por matriz de massa é obtida pela substituição das funções nodais (3.26), (3.29) e (3.30) na expressão (3.15). A matriz conhecida como matriz de rigidez, $\widetilde{\mathbf{C}}$, foi originada com a substituição das funções (3.26), (3.28) e (3.30) nas expressões (3.16), (3.17) e (3.18).

As matrizes \mathbf{K} , $\overline{\mathbf{M}}$, $\widetilde{\mathbf{M}}$ e $\widetilde{\mathbf{C}}$ são esparsas, e devido as particularidades do método de elementos finitos, tal esparsidade é aleatória, significa que os elementos não nulos ocorrem em posições arbitrárias. Visando armazenar somente os elementos diferentes de zero dessas matrizes e com isso reduzir o consumo de memória e o número de operações de ponto flutuante, várias estratégias podem ser usadas, como por exemplo, elemento-por-elemento (EBE) [27], aresta-por-aresta (EDE) [9], *Modified Sparse Row* (MSR), *Compressed Storage Row* (CSR) [55], dentre outras. Nesse trabalho serão discutidas duas dessas estratégias: elemento-por-elemento (EBE) e aresta-por-aresta (EDE). A estratégia de armazenamento elemento-por-elemento é a mais utilizada, nela não se monta a matriz de discretização por elementos finitos, no entanto, suas contribuições são armazenadas em estruturas de dados que representam cada elemento. A estratégia aresta-por-aresta é semelhante a estratégia elemento-por-elemento e nela também não se monta a matriz de discretização

por elementos finitos, entretanto suas contribuições são guardadas em estruturas de arestas dos elementos.

3.2 Estrutura de Dados por Elementos

É conveniente representar os termos das parcelas das equações (3.31), (3.32) e (3.33) em função das contribuições de cada elemento. Assim, as matrizes \mathbf{K} , $\overline{\mathbf{M}}$, $\widetilde{\mathbf{M}}$ e $\widetilde{\mathbf{C}}$ e os vetores \mathbf{Q} e \mathbf{F} podem ser calculados por uma soma inteligente das contribuições de cada elemento, em um processo chamado de *assembling*. Seque que,

$$\mathbf{K} = \mathbf{A}_{e=1}^{nel}(\widetilde{k}^e) \quad (3.34)$$

$$\widetilde{\mathbf{M}} = \mathbf{A}_{e=1}^{nel}(m_g^e + m_{pg}^e) \quad (3.35)$$

$$\widetilde{\mathbf{C}} = \mathbf{A}_{e=1}^{nel}(\widetilde{k}_g^e + \widetilde{k}_{pg}^e + c_{cau}^e) \quad (3.36)$$

$$\overline{\mathbf{M}} = \mathbf{A}_{e=1}^{nel}(m_w^e + m_{div}^e) \quad (3.37)$$

$$\mathbf{F} = \mathbf{A}_{e=1}^{nel}(f_g^e + f_{pos}^e) \quad (3.38)$$

$$\mathbf{Q} = \mathbf{A}_{e=1}^{nel}f_q^e \quad (3.39)$$

onde os subscritos g e w, pg, cau, pos e div relacionam-se as contribuições obtidas, respectivamente, a partir das formulações de Galerkin, SUPG, CAU e pós-processamento da velocidade e \mathbf{A} indica o *assembling* das contribuições de cada elemento.

A matriz $\widetilde{\mathbf{C}}$ é obtida a partir da soma dos termos convectivos (3.16), difusivos (3.17) e choque (3.18). Para uma melhor clareza e organização, as contribuições da matriz $\widetilde{\mathbf{C}}$ serão desmembradas em contribuições convectivas, \mathbf{C} , difusivas, \mathbf{D} e de choque, \mathbf{CAU} . Sendo que tanto as contribuições convectivas quanto as difusivas são compostas por parcelas referentes a formulação de Galerkin e a formulação SUPG. Reescrevendo $\widetilde{\mathbf{C}}$ considerando essa nova organização tem-se:

$$\widetilde{\mathbf{C}} = \mathbf{D} + \mathbf{C} + \mathbf{CAU} = \mathbf{A}_{e=1}^{nel}(c_g^e + c_{pg}^e + d_g^e + d_{pg}^e + c_{cau}^e) \quad (3.40)$$

Na discretização do domínio Ω , os elementos adotados são triangulares lineares conforme mostra a Fig. 3.1.

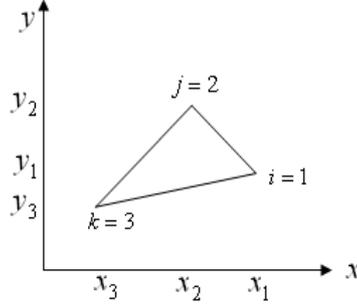


Figura 3.1: Elemento triangular linear.

No elemento adotado, as aproximações da pressão (3.27), concentração (3.28) e velocidade (3.30) podem ser descritas como:

$$p^e(x, t) \simeq \sum_{i=1}^3 p_i(t) N_i(x) \quad (3.41)$$

$$c^e(x, t) \simeq \sum_{i=1}^3 c_i(t) N_i(x) \quad (3.42)$$

$$\mathbf{v}^e(x, t) \simeq \sum_{i=1}^3 \mathbf{v}_i(t) \tilde{\mathbf{N}}_i(x) \quad (3.43)$$

onde N_i e $\tilde{\mathbf{N}}_i$ com $i=1,2,3$ são, respectivamente, as funções de interpolação no interior do elemento para pressão e concentração e para velocidade.

Visando simplificar as funções de interpolação N_i para solucionar as integrais nos elementos sem o uso de aproximações numéricas, será realizada uma transformação de variáveis globais (x, y) para variáveis locais (ξ, η) . Esse mapeamento é realizado conforme ilustra a Fig. 3.2.

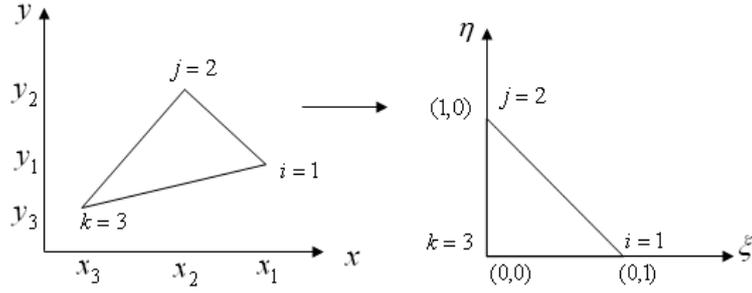


Figura 3.2: Transformação do elemento genérico para o elemento padrão.

Após a transformação, as funções de interpolação podem ser definidas como:

$$\begin{aligned}
 N_1 &= \xi \\
 N_2 &= \eta \\
 N_3 &= 1 - \xi - \eta
 \end{aligned} \tag{3.44}$$

A matriz que define a transformação das coordenadas globais para as coordenadas locais, é denominada matriz Jacobiana, e é representada por:

$$\mathbf{J} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} \end{bmatrix} = \begin{bmatrix} x_{13} & -x_{32} \\ -y_{31} & y_{23} \end{bmatrix} \tag{3.45}$$

sendo x_i as coordenadas globais do elemento. A diferença de coordenadas globais é representada por $x_{ij} = x_i - x_j$, com $i, j = 1, 2, 3$. Ao se calcular a transformação inversa da matriz Jacobiana para elementos triangulares, obtém-se:

$$\mathbf{J}^{-1} = \begin{bmatrix} \frac{\partial \xi}{\partial x} & \frac{\partial \xi}{\partial y} \\ \frac{\partial \eta}{\partial x} & \frac{\partial \eta}{\partial y} \end{bmatrix} = \frac{1}{2A^e} \begin{bmatrix} y_{23} & x_{32} \\ y_{31} & x_{13} \end{bmatrix} \tag{3.46}$$

sendo A^e a área do elemento. O vetor de interpolação contém as funções de interpolação no interior do elemento, assim

$$\mathbf{N} = \begin{bmatrix} N_1 \mathbf{I} & N_2 \mathbf{I} & N_3 \mathbf{I} \end{bmatrix} \tag{3.47}$$

onde \mathbf{I} é a matriz de identidade de ordem 1 para a pressão e concentração e de ordem 2 para a velocidade, visto que a velocidade é representada por um vetor com dois

graus de liberdade, correspondentes as componentes v_x e v_y . Assim, para a pressão e concentração o vetor de interpolação é dado por:

$$\mathbf{N} = \begin{bmatrix} N_1 & N_2 & N_3 \end{bmatrix} \quad (3.48)$$

E para a velocidade, o vetor de interpolação é:

$$\tilde{\mathbf{N}} = \begin{bmatrix} N_1 & 0 & N_2 & 0 & N_3 & 0 \\ 0 & N_1 & 0 & N_2 & 0 & N_3 \end{bmatrix} \quad (3.49)$$

O operador gradiente discreto das funções de interpolação, tem a forma:

$$\mathbf{B} = \begin{bmatrix} \mathbf{B}_x \\ \mathbf{B}_y \end{bmatrix} = \frac{1}{2A^e} \begin{bmatrix} \frac{\partial N_1}{\partial x} & \frac{\partial N_2}{\partial x} & \frac{\partial N_3}{\partial x} \\ \frac{\partial N_1}{\partial y} & \frac{\partial N_2}{\partial y} & \frac{\partial N_3}{\partial y} \end{bmatrix} \quad (3.50)$$

Para a pressão e concentração o operador gradiente discreto no elemento triangular pode ser representado por:

$$\mathbf{B} = \frac{1}{2A^e} \begin{bmatrix} y_{23} & y_{31} & y_{12} \\ x_{32} & x_{13} & y_{21} \end{bmatrix} \quad (3.51)$$

O operador gradiente discreto para a velocidade é representado por:

$$\tilde{\mathbf{B}} = \frac{1}{2A^e} \begin{bmatrix} y_{23} & 0 & y_{31} & 0 & y_{12} & 0 \\ 0 & y_{23} & 0 & y_{31} & 0 & y_{12} \\ x_{32} & 0 & x_{13} & 0 & y_{21} & 0 \\ 0 & x_{32} & 0 & x_{13} & 0 & y_{21} \end{bmatrix} \quad (3.52)$$

O operador divergente discreto para a velocidade é:

$$\mathbf{B}_{Div} = \frac{1}{2A^e} \begin{bmatrix} y_{23} & x_{32} & y_{31} & x_{13} & y_{12} & x_{21} \end{bmatrix} \quad (3.53)$$

A definição do operador gradiente é amplamente usada no desenvolvimento das matrizes dos elementos para a concentração, pressão e velocidade.

3.2.1 Matrizes de Coeficientes para a Equação da Pressão

Utilizando as aproximações de elementos finitos descritas na Seção 3.2 para cada elemento, segue da forma variacional da equação da pressão, Equação (3.2), que:

$$\int_{\Omega^e} \nabla w^h \mathbf{A} \nabla p^h d\Omega^e \Rightarrow \mathbf{k}^e = \int_{\Omega^e} \mathbf{B}^T \mathbf{A} \mathbf{B} d\Omega^e \quad (3.54)$$

No caso particular de permeabilidade homogênea, as componentes $\lambda_{12} = \lambda_{21} = 0$ no tensor de permeabilidade, logo $\mathbf{A}(\mathbf{v})$ será reescrito como:

$$\mathbf{A}(\mathbf{v}) = \begin{bmatrix} \Lambda_{11} & 0 \\ 0 & \Lambda_{22} \end{bmatrix}$$

onde $\Lambda_{ii} = \frac{\lambda_{ii}}{\mu}$ para $i = 1, 2$.

Como o integrando $\mathbf{B}^T \mathbf{A} \mathbf{B}$ é constante então,

$$\mathbf{k}^e = \frac{1}{4A^e} \begin{bmatrix} y_{23} & x_{32} \\ y_{31} & x_{13} \\ y_{12} & x_{21} \end{bmatrix} \begin{bmatrix} \Lambda_{11} & 0 \\ 0 & \Lambda_{22} \end{bmatrix} \begin{bmatrix} y_{23} & y_{31} & y_{12} \\ x_{32} & x_{13} & x_{21} \end{bmatrix} \quad (3.55)$$

É fácil verificar que a expressão resulta em uma matriz simétrica

$$\mathbf{k}^e = \frac{1}{4\mu A^e} \begin{bmatrix} P_{11} & P_{12} & P_{13} \\ & P_{22} & P_{23} \\ sim & & P_{33} \end{bmatrix} \quad (3.56)$$

sendo os coeficientes P_{ij} dados por:

$$P_{11} = \lambda_{11} y_{23} y_{23} + \lambda_{22} x_{32} x_{32}$$

$$P_{12} = \lambda_{11} y_{31} y_{23} + \lambda_{22} x_{13} x_{32}$$

$$P_{13} = \lambda_{11} y_{23} y_{12} + \lambda_{22} x_{32} x_{21}$$

$$P_{22} = \lambda_{11} y_{31} y_{31} + \lambda_{22} x_{13} x_{13}$$

$$P_{23} = \lambda_{11} y_{31} y_{12} + \lambda_{22} x_{13} x_{21}$$

$$P_{33} = \lambda_{11} y_{12} y_{12} + \lambda_{22} x_{21} x_{21}$$

A matriz \mathbf{K} tem deficiência de posto, ou seja, os elementos da diagonal podem ser obtidos como a soma dos elementos de fora da diagonal principal em cada linha.

Assim, considerando que a matriz é simétrica observa-se que:

$$\mathbf{k}^e = \frac{1}{4\mu A^e} \begin{bmatrix} -(P_{12} + P_{13}) & P_{12} & P_{13} \\ & -(P_{12} + P_{23}) & P_{23} \\ sim & & -(P_{13} + P_{23}) \end{bmatrix} \quad (3.57)$$

A deficiência de posto e simetria da matriz \mathbf{K} são características muito vantajosas para a implementação, pois através de apenas 3 coeficientes, P_{12} , P_{13} e P_{23} , todos os outros são calculados.

3.2.2 Matriz de Massa de Galerkin para a Concentração

A matriz de massa para cada elemento pode ser obtida por:

$$\int_{\Omega^e} w^h \phi \frac{\partial c^h}{\partial t} d\Omega^e \Rightarrow \mathbf{m}_g^e = \int_{\Omega^e} \phi \mathbf{N}^T \mathbf{N} d\Omega^e \quad (3.58)$$

A integral das funções de interpolação definidas em (3.44) pode ser facilmente calculada. Tem-se que $\int_{\Omega^e} N_i N_j d\Omega^e = \frac{A^e}{12}$ se $i \neq j$ e $\int_{\Omega^e} N_i N_j d\Omega^e = \frac{2A^e}{12}$ se $i = j$. Assim, a matriz \mathbf{m}_g^e é:

$$\mathbf{m}_g^e = \frac{\phi A^e}{12} \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix} \quad (3.59)$$

3.2.3 Matriz de Convecção de Galerkin para a Concentração

A matriz que contém os termos convectivos devido a formulação de Galerkin é dada por:

$$\int_{\Omega^e} w^h \nabla \cdot (c^h \mathbf{v}^h) d\Omega^e \Rightarrow \mathbf{c}_g^e = \int_{\Omega^e} \mathbf{N}^T \mathbf{v}^T \mathbf{B} d\Omega^e \quad (3.60)$$

Como o integrando $\mathbf{B}^T \mathbf{v}$ é constante, se verifica que:

$$\mathbf{c}_g^e = \left(\int_{\Omega^e} \mathbf{N}^T d\Omega^e \right) \mathbf{v}^T \mathbf{B} \quad (3.61)$$

A integral das funções de interpolação N_i , definidas em (3.44) pode ser obtida considerando $\int_{\Omega^e} N_i d\Omega^e = \frac{2A^e}{6}$, logo:

$$\int_{\Omega^e} \mathbf{N}_i d\Omega^e = \frac{2A^e}{6} \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \quad (3.62)$$

Considerando os resultados contidos em (3.61), a matriz \mathbf{c}_g^e pode ser representada por:

$$\mathbf{c}_g^e = \frac{2A^e}{6} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} v_x & v_y \end{bmatrix} \frac{1}{2A^e} \begin{bmatrix} y_{23} & y_{31} & y_{12} \\ x_{32} & x_{13} & x_{21} \end{bmatrix} \quad (3.63)$$

Após alguns cálculos, \mathbf{c}_g^e resulta em:

$$\mathbf{c}_g^e = \frac{1}{6} \begin{bmatrix} y_{23}v_x + x_{32}v_y & y_{31}v_x + x_{13}v_y & y_{12}v_x + x_{21}v_y \\ y_{23}v_x + x_{32}v_y & y_{31}v_x + x_{13}v_y & y_{12}v_x + x_{21}v_y \\ y_{23}v_x + x_{32}v_y & y_{31}v_x + x_{13}v_y & y_{12}v_x + x_{21}v_y \end{bmatrix} \quad (3.64)$$

A matriz \mathbf{c}_g tem deficiência de posto, portanto, pode ser escrita como:

$$\mathbf{c}_g^e = \frac{\tau^e}{6} \begin{bmatrix} -(c_{12} + c_{13}) & c_{12} & c_{13} \\ c_{11} & -(c_{11} + c_{13}) & c_{13} \\ c_{11} & c_{12} & -(c_{11} + c_{13}) \end{bmatrix} \quad (3.65)$$

onde os coeficientes c_{ij} com $i, j = 1, 2, 3$ representam as parcelas:

$$c_{11} = y_{23}v_x + x_{32}v_y$$

$$c_{12} = y_{31}v_x + x_{13}v_y$$

$$c_{13} = y_{12}v_x + x_{21}v_y$$

A matriz \mathbf{c}_g não é simétrica mas pode ser representada com apenas três coeficientes distintos.

$$\mathbf{c}_g^e = \frac{\tau^e}{6} \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{11} & c_{12} & c_{13} \\ c_{11} & c_{12} & c_{13} \end{bmatrix} \quad (3.66)$$

3.2.4 Matriz de Difusão de Galerkin para a Concentração

Reescrevendo o termo referente à difusão contido na expressão (3.17), segue:

$$\int_{\Omega^e} \nabla w^h \cdot \mathbf{D} \nabla c^h d\Omega^e \Rightarrow \mathbf{d}_g^e = \int_{\Omega^e} \mathbf{B}^T \mathbf{D} \mathbf{B} d\Omega^e \quad (3.67)$$

Como o integrando $\mathbf{B}^T \mathbf{D} \mathbf{B}$ é constante, segue dessa maneira, que:

$$\mathbf{d}_g^e = \mathbf{B}^T \mathbf{D} \mathbf{B} \int_{\Omega^e} d\Omega^e \quad (3.68)$$

Considerando as definições do tensor \mathbf{D} descritas em (2.40), segue:

$$\mathbf{d}_g^e = \frac{1}{2A^e} \begin{bmatrix} y_{23} & x_{32} \\ y_{31} & x_{13} \\ y_{12} & x_{21} \end{bmatrix} \begin{bmatrix} d_{11} & d_{12} \\ d_{21} & d_{22} \end{bmatrix} \frac{1}{2A^e} \begin{bmatrix} y_{23} & y_{31} & y_{12} \\ x_{32} & x_{13} & x_{21} \end{bmatrix} A^e \quad (3.69)$$

Dado que no tensor \mathbf{D} , a igualdade se verifica, $d_{12} = d_{21}$, tem-se que a matriz \mathbf{d}_g^e é simétrica. Além disso, apresenta a propriedade de deficiência de posto, portanto

pode ser representada por:

$$\mathbf{d}_g^e = \frac{1}{4A^e} \begin{bmatrix} -(D_{12} + D_{13}) & D_{12} & D_{13} \\ & -(D_{21} + D_{23}) & D_{23} \\ \text{sim} & & -(D_{31} + D_{32}) \end{bmatrix} \quad (3.70)$$

onde os coeficientes D_{ij} são:

$$\begin{aligned} D_{12} &= y_{31}(d_{11}y_{32} + d_{21}x_{32}) + x_{13}(d_{12}y_{32} + d_{22}x_{32}) \\ D_{13} &= y_{12}(d_{11}y_{32} + d_{21}x_{32}) + x_{21}(d_{12}y_{32} + d_{22}x_{32}) \\ D_{23} &= y_{12}(d_{11}y_{31} + d_{21}x_{13}) + x_{21}(d_{12}y_{31} + d_{22}x_{13}) \end{aligned}$$

3.2.5 Correção SUPG da Matriz de Massa

Reescrevendo o termo referente à correção SUPG da matriz de massa contido em (3.15), tem-se:

$$\tau^e \int_{\Omega^e} \mathbf{v}^h \nabla w^h \phi \frac{\partial c^h}{\partial t} d\Omega^e \Rightarrow \mathbf{m}_{pg}^e = \tau^e \phi \int_{\Omega^e} \mathbf{B}^T \mathbf{v} \mathbf{N} d\Omega^e \quad (3.71)$$

Como o integrando $\mathbf{B}^T \mathbf{v}$ é constante se verifica,

$$\mathbf{m}_{pg}^e = \tau^e \phi \mathbf{B}^T \mathbf{v} \int_{\Omega^e} \mathbf{N} d\Omega^e \quad (3.72)$$

A integral das funções de interpolação N_i definidas em (3.44) é $\int_{\Omega^e} N_i d\Omega^e = \frac{2A^e}{6}$.

Assim,

$$\int_{\Omega^e} \mathbf{N} d\Omega^e = \frac{2A^e}{6} \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \quad (3.73)$$

Considerando os resultados contidos em (3.73) a matriz \mathbf{m}_{pg}^e pode ser representada por,

$$\mathbf{m}_{pg}^e = \tau^e \phi \frac{1}{2A^e} \begin{bmatrix} y_{23} & x_{32} \\ y_{31} & x_{13} \\ y_{12} & x_{21} \end{bmatrix} \begin{bmatrix} \mathbf{v}_x \\ \mathbf{v}_y \end{bmatrix} \frac{2A^e}{6} \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \quad (3.74)$$

Portanto,

$$\mathbf{m}_{pg}^e = \frac{\tau^e \phi}{6} \begin{bmatrix} y_{23}v_x + x_{32}v_y & y_{23}v_x + x_{32}v_y & y_{23}v_x + x_{32}v_y \\ y_{31}v_x + x_{13}v_y & y_{31}v_x + x_{13}v_y & y_{31}v_x + x_{13}v_y \\ y_{12}v_x + x_{21}v_y & y_{12}v_x + x_{21}v_y & y_{12}v_x + x_{21}v_y \end{bmatrix} \quad (3.75)$$

Observa-se que \mathbf{m}_{pg} tem deficiência de posto e pode ser representada por:

$$\mathbf{m}_{pg}^e = \frac{\tau^e \phi}{6} \begin{bmatrix} -(m_{21} + m_{31}) & m_{11} & m_{11} \\ m_{21} & -(m_{11} + m_{31}) & m_{21} \\ m_{31} & m_{31} & -(m_{11} + m_{21}) \end{bmatrix} \quad (3.76)$$

onde os coeficientes m_{i1} com $i = 1, 2, 3$ são dados por:

$$m_{11} = y_{23}v_x + x_{32}v_y$$

$$m_{21} = y_{31}v_x + x_{13}v_y$$

$$m_{31} = y_{12}v_x + x_{21}v_y$$

Nota-se que \mathbf{m}_{pg} é igual a \mathbf{c}_g^T , a menos de uma constante. Portanto, \mathbf{m}_{pg} também pode ser representada por apenas três coeficientes distintos:

$$\mathbf{m}_{pg}^e = \frac{\tau^e \phi}{6} \begin{bmatrix} m_{11} & m_{11} & m_{11} \\ m_{21} & m_{21} & m_{21} \\ m_{31} & m_{31} & m_{31} \end{bmatrix} = \frac{\tau^e \phi}{6} \begin{bmatrix} c_{11} & c_{11} & c_{11} \\ c_{12} & c_{12} & c_{12} \\ c_{13} & c_{13} & c_{13} \end{bmatrix} \quad (3.77)$$

onde c_{11}, c_{12} e c_{13} são os termos da matriz (3.66).

3.2.6 Correção SUPG da Matriz de Convecção

Reescrevendo o termo referente à correção SUPG da matriz de convecção contido em (3.16), segue,

$$\tau^e \int_{\Omega^e} \mathbf{v}^h \cdot \nabla w^h \mathbf{v}^h \cdot \nabla c^h d\Omega^e \Rightarrow \mathbf{c}_{pg}^e = \tau^e \int_{\Omega^e} \mathbf{B}^T \mathbf{v} \mathbf{v}^T \mathbf{B} d\Omega^e \quad (3.78)$$

Considerando a velocidade constante no elemento. Pode-se dizer que:

$$\mathbf{c}_{pg}^e = \tau^e \frac{1}{2A^e} \begin{bmatrix} y_{23} & x_{32} \\ y_{31} & x_{13} \\ y_{12} & x_{21} \end{bmatrix} \begin{bmatrix} v_{xx} & v_{xy} \\ v_{xy} & v_{yy} \end{bmatrix} \frac{1}{2A^e} \begin{bmatrix} y_{23} & y_{31} & y_{12} \\ x_{32} & x_{13} & x_{21} \end{bmatrix} A^e \quad (3.79)$$

Visto que \mathbf{c}_{pg}^e é simétrica e também possui a propriedade de deficiência de posto, ela pode ser apresentada da seguinte maneira:

$$\mathbf{c}_{pg}^e = \tau^e \frac{1}{4A^e} \begin{bmatrix} -(C_{12} + C_{13}) & C_{12} & C_{13} \\ & -(C_{12} + C_{23}) & C_{23} \\ sim & & -(C_{13} + C_{23}) \end{bmatrix} \quad (3.80)$$

onde aparecem três termos distintos a saber:

$$C_{12} = y_{31}(v_{xx}y_{23} + v_{xy}x_{32}) + x_{13}(v_{xy}y_{32} + v_{yy}x_{32})$$

$$C_{13} = y_{12}(v_{xx}y_{23} + v_{xy}x_{32}) + x_{21}(v_{xy}y_{23} + v_{yy}x_{32})$$

$$C_{23} = y_{12}(v_{xx}y_{31} + v_{xy}x_{13}) + x_{21}(v_{xy}y_{31} + v_{yy}x_{13})$$

3.2.7 Correção SUPG da Matriz de Difusão

Da parcela de difusão da expressão (3.17) é possível obter para cada elemento:

$$\tau^e \int_{\Omega^e} \nabla(\mathbf{v}^h \cdot \nabla w^h) \cdot \mathbf{D} \nabla c^h d\Omega^e \quad (3.81)$$

O integrando da expressão (3.81) depende das derivadas de segunda ordem das funções de interpolação. Como estão sendo consideradas funções lineares, então a contribuição SUPG do termo de difusão é nula, isto é,

$$\mathbf{d}_{pg}^e = 0 \quad (3.82)$$

3.2.8 Matriz de Correção do Operador de Descontinuidade

A matriz que contém os termos do operador de captura de descontinuidades, presente em (3.18), pode ser reescrita por:

$$\int_{\Omega_e} \delta_{cau}^e \nabla w^h \nabla c^h d\Omega \Rightarrow \mathbf{c}_{cau}^e = \int_{\Omega^e} \delta_{cau}^e \mathbf{B}^T \mathbf{B} d\Omega^e \quad (3.83)$$

sendo, δ_{cau}^e constante no interior do elemento, a matriz \mathbf{c}_{cau}^e pode ser representada por:

$$\mathbf{c}_{cau}^e = \frac{\delta_{cau}^e}{4A^e} \begin{bmatrix} y_{23} & x_{32} \\ y_{31} & x_{13} \\ y_{12} & x_{21} \end{bmatrix} \begin{bmatrix} y_{23} & y_{31} & y_{12} \\ x_{32} & x_{13} & x_{21} \end{bmatrix} \quad (3.84)$$

Como a matriz \mathbf{c}_{cau}^e é simétrica e apresenta deficiência de posto, segue:

$$\mathbf{c}_{cau}^e = \frac{\delta_{cau}^e}{4A^e} \begin{bmatrix} -(O_{12} + O_{13}) & O_{12} & O_{13} \\ & -(O_{12} + O_{23}) & O_{23} \\ & & -(O_{13} + O_{23}) \end{bmatrix} \quad (3.85)$$

onde aparecem três termos distintos:

$$O_{12} = y_{31}y_{23} + x_{32}x_{13}$$

$$O_{13} = y_{12}y_{23} + x_{32}x_{21}$$

$$O_{23} = y_{12}y_{31} + x_{13}x_{21}$$

3.2.9 Matrizes de Coeficientes para a Equação da Velocidade

O termo referente a forma variacional da Lei de Darcy é:

$$\int_{\Omega^e} w^h \mathbf{A}(c^h)^{-1} \mathbf{v}^h d\Omega^e \Rightarrow \tilde{\mathbf{m}}_w^e = \int_{\Omega^e} \tilde{\mathbf{N}} \mathbf{A}^{-1} \tilde{\mathbf{N}} d\Omega^e \quad (3.86)$$

A matriz $\tilde{\mathbf{m}}_w^e$, considerando o inverso do tensor $\mathbf{A}(c)$, definido em (3.20), resulta em:

$$\tilde{\mathbf{m}}_w^e = \int_{\Omega^e} \begin{bmatrix} N_1 & 0 \\ 0 & N_1 \\ N_2 & 0 \\ 0 & N_2 \\ N_3 & 0 \\ 0 & N_3 \end{bmatrix} \frac{1}{\det(\mathbf{A})} \begin{bmatrix} \Lambda_{22} & -\Lambda_{12} \\ -\Lambda_{21} & \Lambda_{11} \end{bmatrix} \begin{bmatrix} N_1 & 0 & N_2 & 0 & N_3 & 0 \\ 0 & N_1 & 0 & N_2 & 0 & N_3 \end{bmatrix} d\Omega \quad (3.87)$$

Uma vez que $\int_{\Omega^e} N_i N_j d\Omega^e = \frac{2A^e}{12}$ se $i = j$ e $\int_{\Omega^e} N_i N_j d\Omega^e = \frac{A^e}{12}$ se $i \neq j$ segue:

$$\tilde{\mathbf{m}}_w^e = \frac{A^e}{12 \det(\mathbf{A})} \begin{bmatrix} 2\Lambda_{22} & -2\Lambda_{12} & \Lambda_{22} & -\Lambda_{12} & \Lambda_{22} & -\Lambda_{12} \\ -2\Lambda_{21} & 2\Lambda_{11} & -\Lambda_{21} & \Lambda_{11} & -\Lambda_{21} & \Lambda_{11} \\ \Lambda_{22} & -\Lambda_{12} & 2\Lambda_{22} & -2\Lambda_{12} & \Lambda_{22} & -\Lambda_{12} \\ -\Lambda_{21} & \Lambda_{11} & -2\Lambda_{21} & 2\Lambda_{11} & -\Lambda_{21} & \Lambda_{11} \\ \Lambda_{22} & -\Lambda_{12} & \Lambda_{22} & -\Lambda_{12} & 2\Lambda_{22} & -2\Lambda_{12} \\ -\Lambda_{21} & \Lambda_{11} & -\Lambda_{21} & \Lambda_{11} & -2\Lambda_{21} & 2\Lambda_{11} \end{bmatrix} \quad (3.88)$$

Como o tensor $\mathbf{A}(c)$ depende do tensor \mathbf{K} , conforme expresso em (2.55), a matriz $\tilde{\mathbf{m}}_w^e$ pode ser reescrita da seguinte maneira:

$$\tilde{\mathbf{m}}_w^e = \frac{A^e}{12\mu \det(\mathbf{A})} \begin{bmatrix} 2\lambda_{22} & -2\lambda_{12} & \lambda_{22} & -\lambda_{12} & \lambda_{22} & -\lambda_{12} \\ -2\lambda_{21} & 2\lambda_{11} & -\lambda_{21} & \lambda_{11} & -\lambda_{21} & \lambda_{11} \\ \lambda_{22} & -\lambda_{12} & 2\lambda_{22} & -2\lambda_{12} & \lambda_{22} & -\lambda_{12} \\ -\lambda_{21} & \lambda_{11} & -2\lambda_{21} & 2\lambda_{11} & -\lambda_{21} & \lambda_{11} \\ \lambda_{22} & -\lambda_{12} & \lambda_{22} & -\lambda_{12} & 2\lambda_{22} & -2\lambda_{12} \\ -\lambda_{21} & \lambda_{11} & -\lambda_{21} & \lambda_{11} & -2\lambda_{21} & 2\lambda_{11} \end{bmatrix} \quad (3.89)$$

No caso particular de permeabilidade homogênea, ou seja, $\lambda_{12} = \lambda_{21} = 0$, o tensor $\mathbf{A}(c)$ será redefinido por:

$$\mathbf{A}(c) = \begin{bmatrix} \frac{\lambda_{11}}{\mu} & 0 \\ 0 & \frac{\lambda_{22}}{\mu} \end{bmatrix} \quad (3.90)$$

Modificando a matriz $\tilde{\mathbf{m}}_w^e$ com o tensor homogêneo, Eq. (3.90), tem-se:

$$\tilde{\mathbf{m}}_w^e = \frac{\mu A^e}{12} \begin{bmatrix} 2\lambda_{22} & 0 & \lambda_{22} & 0 & \lambda_{22} & 0 \\ 0 & 2\lambda_{11} & 0 & \lambda_{11} & 0 & \lambda_{11} \\ \lambda_{22} & 0 & 2\lambda_{22} & 0 & \lambda_{22} & 0 \\ 0 & \lambda_{11} & 0 & 2\lambda_{11} & 0 & \lambda_{11} \\ \lambda_{22} & 0 & \lambda_{22} & 0 & 2\lambda_{22} & 0 \\ 0 & \lambda_{11} & 0 & \lambda_{11} & 0 & 2\lambda_{11} \end{bmatrix} \quad (3.91)$$

A parcela responsável pela ponderação do divergente da velocidade é expressa por:

$$\sum_{e=1}^{nel} \delta_{pp}^e \int_{\Omega^e} \nabla \cdot \mathbf{w}^h \nabla \cdot \mathbf{v}^h d\Omega^e \Rightarrow \tilde{\mathbf{m}}_{div}^e = \int_{\Omega^e} \mathbf{B}_{Div}^T \mathbf{B}_{Div} d\Omega^e \quad (3.92)$$

Utilizando o operador \mathbf{B}_{Div} , descrito em (3.53), a matriz $\tilde{\mathbf{m}}_{div}^e$, pode ser representada por:

$$\tilde{\mathbf{m}}_{div}^e = \frac{\delta_{pp}^e}{4A^e} \begin{bmatrix} y_{23} \\ x_{32} \\ y_{31} \\ x_{13} \\ y_{12} \\ x_{21} \end{bmatrix} \begin{bmatrix} y_{23} & x_{32} & y_{31} & x_{13} & y_{12} & x_{21} \end{bmatrix} \quad (3.93)$$

A matriz $\tilde{\mathbf{m}}_{div}^e$ é:

$$\tilde{\mathbf{m}}_{div}^e = \frac{\delta_{pp}^e}{4A^e} \begin{bmatrix} y_{23}y_{23} & y_{23}x_{32} & y_{23}y_{31} & y_{23}x_{13} & y_{23}y_{12} & y_{23}x_{21} \\ & x_{32}x_{32} & x_{32}y_{31} & x_{32}x_{13} & x_{32}y_{12} & x_{32}x_{21} \\ & & y_{31}y_{31} & y_{31}x_{13} & y_{31}y_{12} & y_{31}x_{21} \\ & & & x_{13}x_{13} & x_{13}y_{12} & x_{13}x_{21} \\ & & & & y_{12}y_{12} & y_{12}x_{21} \\ sim & & & & & x_{21}x_{21} \end{bmatrix} \quad (3.94)$$

A parcela relacionada ao gradiente da pressão é representada por:

$$\int_{\Omega^e} \mathbf{w}^h \cdot \nabla p^h d\Omega^e = \left(\int_{\Omega^e} \mathbf{w}^h \right) \cdot \nabla p^h d\Omega^e \Rightarrow \tilde{\mathbf{f}}_g^e = \left(\int_{\Omega^e} \tilde{\mathbf{N}}^T d\Omega^e \right) \mathbf{Bp} \quad (3.95)$$

$$\tilde{\mathbf{f}}_g^e = \int_{\Omega^e} \begin{bmatrix} N_1 & 0 \\ 0 & N_1 \\ N_2 & 0 \\ 0 & N_2 \\ N_3 & 0 \\ 0 & N_3 \end{bmatrix} d\Omega^e \frac{1}{2A^e} \begin{bmatrix} y_{23} & y_{31} & y_{12} \\ x_{32} & x_{13} & x_{21} \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix} \quad (3.96)$$

Lembrando que $\int_{\Omega^e} N_i d\Omega^e = \frac{2A^e}{6}$ segue imediatamente que:

$$\tilde{\mathbf{f}}_g^e = \frac{2A^e}{6} \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \frac{1}{2A^e} \begin{bmatrix} y_{23} & y_{31} & y_{12} \\ x_{32} & x_{13} & x_{21} \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix} \quad (3.97)$$

Portanto, tem-se:

$$\tilde{\mathbf{f}}_g^e = \frac{1}{6} \begin{bmatrix} y_{23}p_1 + y_{31}p_2 + y_{12}p_3 \\ x_{32}p_1 + x_{13}p_2 + x_{21}p_3 \\ y_{23}p_1 + y_{31}p_2 + y_{12}p_3 \\ x_{32}p_1 + x_{13}p_2 + x_{21}p_3 \\ y_{23}p_1 + y_{31}p_2 + y_{12}p_3 \\ x_{32}p_1 + x_{13}p_2 + x_{21}p_3 \end{bmatrix} \quad (3.98)$$

A consideração relacionada ao divergente de fluxo é:

$$\delta_{pp}^e \sum_{e=1}^{nel} \int_{\Omega^e} \nabla \cdot w^h q d\Omega^e \Rightarrow \tilde{\mathbf{f}}_{pos}^e = \delta_{pp}^e \mathbf{B}_{Div}^T \quad (3.99)$$

$$\tilde{\mathbf{f}}_{pos}^e = \frac{\delta_{pp}^e}{2A^e} \delta(x - x_i) \begin{bmatrix} y_{23} \\ x_{32} \\ y_{31} \\ x_{13} \\ y_{12} \\ x_{21} \end{bmatrix} \quad (3.100)$$

sendo $q = \delta(x - x_i)$ e δ , a função Delta de Dirac. Tem-se que q só possui valor não nulo nos poços injetor e produtor, portanto, o Delta de Dirac é diferente de zero apenas nos nós que representam o poço injetor e produtor.

3.3 Estrutura de Dados por Aresta

Os coeficiente das matrizes dos elementos obtidas serão desmembrados para gerar as contribuições em cada aresta [8, 9, 40]. Considere a matriz de um elemento triangular qualquer denominado de e . Sejam os coeficientes dessa matriz representados por T_{ij} com $i, j = 1, 2, 3$, com a propriedade de deficiência de posto. Assim, por exemplo, o coeficiente T_{11} pode ser escrito como a soma de T_{12} e T_{13} onde o coeficiente T_{12} é a contribuição do nó 1 no nó 2, e o coeficiente T_{13} é a contribuição do nó 1 no nó 3. Escrevendo a matriz do elemento utilizando a deficiência de posto é possível observar que ela será formada pela soma de 3 submatrizes, que estão relacionadas as arestas do elemento, conforme ilustra a Eq. (3.101).

$$\underbrace{\begin{bmatrix} -(T_{12} + T_{13}) & T_{12} & T_{13} \\ T_{21} & -(T_{21} + T_{23}) & T_{23} \\ T_{31} & T_{32} & -(T_{31} + T_{32}) \end{bmatrix}}_{\text{elemento } e \text{ } 123} = \underbrace{\begin{bmatrix} -T_{12} & T_{12} & 0 \\ T_{21} & -T_{21} & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{\text{aresta } 12} + \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 0 & -T_{23} & T_{23} \\ 0 & T_{32} & -T_{32} \end{bmatrix}}_{\text{aresta } 23} + \underbrace{\begin{bmatrix} -T_{13} & 0 & T_{13} \\ 0 & 0 & 0 \\ T_{31} & 0 & -T_{31} \end{bmatrix}}_{\text{aresta } 31} \quad (3.101)$$

Qualquer que seja a aresta do elemento e , as informações que incidem sobre a matriz dessa aresta não são originadas apenas por um elemento, mas por todos os elementos

adjacentes a essa aresta. Observando a Fig. 3.3, nota-se que a aresta formada pelos nós 1 e 2, nomeada por s , pertence aos dois elementos e e f , logo sua matriz recebe contribuição desses dois elementos. Denominando de a e b os coeficientes da matriz da resta s , temos que $a = T_{12}^e + T_{12}^f$ e $b = T_{21}^e + T_{21}^f$, onde T_{ij}^e com $i, j = 1, 2, 3$ são os coeficientes da matriz do elemento e , e T_{ij}^f com $i, j = 1, 2, 3$ são os coeficientes da matriz do elemento f . A Eq. (3.102) ilustra esse fato.

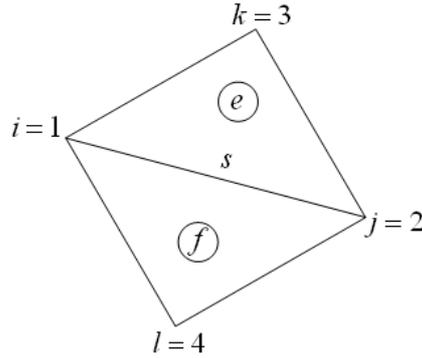


Figura 3.3: Elementos adjacentes a aresta s , formada pelos nós 1 e 2.

$$\underbrace{\begin{bmatrix} -a & a \\ b & -b \end{bmatrix}}_{\text{aresta } s} = \underbrace{\begin{bmatrix} -T_{12}^e & T_{12}^e \\ T_{21}^e & -T_{21}^e \end{bmatrix}}_{\text{elemento } e} + \underbrace{\begin{bmatrix} -T_{12}^f & T_{12}^f \\ T_{21}^f & -T_{21}^f \end{bmatrix}}_{\text{elemento } f} \quad (3.102)$$

É importante lembrar que todas as parcelas descritas na Seção 3.2 podem se escritas na forma da Eq. (3.101). Assim, a geração das matrizes de todas as arestas da malha é realizada, percorrendo a malha, por seus elementos, e para cada elemento, armazenando as contribuições de suas arestas. Como as contribuições são obtidas ao nível do elemento, e uma aresta é compartilhada por dois elementos, haverá a necessidade de se acumular os valores calculados para cada aresta, ocasionando diversas buscas. Assim, será utilizada uma estrutura conveniente, conhecida por tabela Hash, descrita em detalhes para essa aplicação em [35].

Os termos das parcelas do sistemas da pressão (3.31), da velocidade (3.32) e da concentração (3.33), assim como feito na estratégia por elementos em (3.39), serão

representados agora em função das contribuições de suas arestas. Logo:

$$\mathbf{K} = \sum_{s=1}^{nedges} \mathbf{A} (k^s) \quad (3.103)$$

$$\widetilde{\mathbf{M}} = \sum_{s=1}^{nedges} \mathbf{A} (m_g^s + m_{pg}^s) \quad (3.104)$$

$$\widetilde{\mathbf{C}} = \sum_{s=1}^{nedges} \mathbf{A} (k_g^s + k_{pg}^s + c_{cau}^s) \quad (3.105)$$

$$\overline{\mathbf{M}} = \sum_{s=1}^{nedges} \mathbf{A} (m_w^s + m_{div}^s) \quad (3.106)$$

$$\mathbf{F} = \sum_{s=1}^{nedges} \mathbf{A} (f_g^s + f_{pos}^s) \quad (3.107)$$

$$\mathbf{Q} = \sum_{s=1}^{nedges} \mathbf{A} f_q^s \quad (3.108)$$

onde *nedges* é o número de arestas da malha.

3.3.1 Matrizes das Arestas

As matrizes das arestas podem ser obtidas a partir das contribuições de Galerkin, correção SUPG, operador de captura e técnica de pós-processamento das matrizes dos elementos descritas na Seção 3.2. E os elementos, *e* e *f*, são considerados como tendo as respectivas conectividades 1-2-3 e 2-1-4, segundo a Fig. 3.3.

3.3.2 Matriz da Equação da Pressão

Na formulação baseada nos elementos, a matriz \mathbf{K} , representada em (3.57), pode ser descrita em função da aresta *s* da seguinte forma:

$$\mathbf{k}^s = \frac{(A^e + A^f)}{4A^e A^f} \begin{bmatrix} -k_1 & k_1 \\ k_1 & -k_1 \end{bmatrix} \quad (3.109)$$

onde:

$$k_1 = \lambda_{11}(y_{31}y_{23} + y_{14}y_{42}) + \lambda_{22}(x_{13}x_{32} + x_{41}x_{24})$$

3.3.3 Matriz de Massa de Galerkin para a Concentração

Na formulação baseada nos elementos a matriz \mathbf{m}_g^e , representada em (3.59), pode ser descrita em função da aresta, s , nos elementos, e e f , como:

$$\mathbf{m}_g^s = \frac{(A^e + A^f)}{12} \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \quad (3.110)$$

3.3.4 Matriz de Convecção de Galerkin para a Concentração

A partir da matriz \mathbf{c}_g^e , que contém os termos convectivos do elemento e , representada em (3.65), é possível obter a matriz da aresta s , dada por:

$$\mathbf{c}_g^s = \frac{1}{3} \begin{bmatrix} -c_2 & c_2 \\ c_1 & -c_1 \end{bmatrix} \quad (3.111)$$

onde:

$$c_1 = v_x(y_{23}y_{42}) + v_y(x_{32}x_{24})$$

$$c_2 = v_x(y_{31}y_{14}) + v_y(x_{13}x_{41})$$

3.3.5 Matriz de Difusão de Galerkin para a Concentração

A matriz de difusão, \mathbf{d}_g^e , dada pela expressão (3.70), possui as seguintes contribuições na aresta s :

$$\mathbf{d}_g^s = \frac{\phi}{3} \begin{bmatrix} -d_1 & d_1 \\ d_2 & -d_2 \end{bmatrix} \quad (3.112)$$

onde:

$$d_1 = \tau^e[\lambda_{11}y_{31}y_{23} + \lambda_{22}x_{32}x_{13}]$$

$$d_2 = \tau^f[\lambda_{11}y_{42}y_{41} + \lambda_{22}x_{14}x_{24}]$$

3.3.6 Correção SUPG da Matriz de Massa

A correção SUPG da matriz de massa para o elemento e , representada em (3.76), possui as seguintes contribuições na aresta s :

$$\mathbf{m}_{pg}^s = \frac{1}{6} \begin{bmatrix} -c_2 & c_1 \\ c_2 & -c_1 \end{bmatrix} \quad (3.113)$$

onde:

$$\begin{aligned} c_1 &= \tau^e(y_{31}v_x + x_{13}v_y) + \tau^f(y_{14}v_x + x_{41}v_y) \\ c_2 &= \tau^e(y_{23}v_x + x_{32}v_y) + \tau^f(y_{42}v_x + x_{24}v_y) \end{aligned}$$

3.3.7 Correção SUPG da Matriz de Convecção

A correção SUPG da matriz de convecção, dada pela expressão (3.80), para o elemento e , possui as seguintes contribuições na aresta s :

$$\mathbf{c}_{pg}^s = \begin{bmatrix} -g_1 & g_1 \\ g_2 & -g_2 \end{bmatrix} \quad (3.114)$$

onde:

$$\begin{aligned} g_1 &= \frac{1}{4A^e} [y_{31}(v_{xx}y_{23} + v_{xy}x_{32}) + x_{13}(v_{xy}y_{32} + v_{yy}x_{32})] \\ g_2 &= \frac{1}{4A^f} [y_{42}(v_{xx}y_{14} + v_{xy}x_{41}) + x_{24}(v_{xy}y_{41} + v_{yy}x_{41})] \end{aligned}$$

3.3.8 Correção SUPG da matriz de difusão

A correção SUPG da matriz de difusão para o elemento e é nula. Logo, suas contribuições na aresta s também serão nulas.

3.3.9 Matriz de Correção do Operador de Descontinuidade

A matriz com os termos do operador de captura para o elemento e , representada em (3.85), possui as contribuições para a aresta s dadas por:

$$\mathbf{c}_{cau}^s = \begin{bmatrix} -o_1 & o_1 \\ o_2 & -o_2 \end{bmatrix} \quad (3.115)$$

onde:

$$o_1 = \frac{\delta_{cau}^e}{4A^e}(y_{31}y_{23} + x_{13}x_{32})$$

$$o_2 = \frac{\delta_{cau}^f}{4A^f}(y_{42}y_{14} + x_{24}x_{41})$$

3.3.10 Matriz de Coeficientes para a Equação da Velocidade

A matriz $\tilde{\mathbf{m}}_w^e$ que contém os termos referentes a forma variacional da Lei de Darcy, representada em (3.91), possui as seguintes contribuições para a aresta s :

$$\tilde{\mathbf{m}}_w^s = \frac{\mu(A^e + A^f)}{12} \begin{bmatrix} \lambda_{22} & 0 & \lambda_{22} & 0 \\ 0 & \lambda_{11} & 0 & \lambda_{11} \\ \lambda_{22} & 0 & \lambda_{22} & 0 \\ 0 & \lambda_{11} & 0 & \lambda_{11} \end{bmatrix} \quad (3.116)$$

A matriz $\tilde{\mathbf{m}}_{div}^e$ que pondera o divergente da velocidade, representada em (3.94) para o elemento e , possui as seguintes contribuições para a aresta s :

$$\tilde{\mathbf{m}}_{div}^s = \begin{bmatrix} -m_1 & -m_2 & m_1 & m_2 \\ -m_3 & -m_4 & m_3 & m_4 \\ m_1 & m_2 & -m_1 & -m_2 \\ m_3 & m_4 & -m_3 & -m_4 \end{bmatrix} \quad (3.117)$$

onde:

$$m_1 = \frac{\delta_{pp}^e}{4A^e}y_{23}y_{31} + \frac{\delta_{pp}^f}{4A^f}y_{14}y_{42}$$

$$m_2 = \frac{\delta_{pp}^e}{4A^e}y_{23}x_{13} + \frac{\delta_{pp}^f}{4A^f}y_{14}x_{24}$$

$$m_3 = \frac{\delta_{pp}^e}{4A^e}y_{31}x_{32} + \frac{\delta_{pp}^f}{4A^f}y_{42}x_{42}$$

$$m_4 = \frac{\delta_{pp}^e}{4A^e}x_{32}x_{13} + \frac{\delta_{pp}^f}{4A^f}x_{41}x_{24}$$

Ao final da discretização espacial, seja por elemento ou por aresta, é necessário solucionar os sistemas da pressão (3.31), concentração (3.32) e velocidade (3.33). Como a formulação variacional estabilizada semi-discreta está sendo utilizada o próximo

passo é discretizar a derivada temporal da concentração por diferenças finitas e resolver os sistemas. Para tal, será utilizado um algoritmo membro da família de métodos da regra trapezoidal denominado preditor/multicorretor [28].

3.4 Discretização no Tempo

O método preditor/multicorretor é amplamente adotado na resolução de problemas de escoamento [1, 12, 15, 41, 44]. Seja n o contador de passos de tempo, $p(t_n)$ a aproximação da pressão, $a(t_n)$ a derivada temporal para a concentração, $v(t_n)$ a aproximação da velocidade, $c(t_n)$ a aproximação da concentração, Δt o passo de tempo, Δa o incremento da derivada temporal da concentração e i o contador das multicorreções (iterações não-lineares). Representando as aproximações $p(t_n)$, $a(t_n)$, $v(t_n)$ e $c(t_n)$ simplesmente por p_n , a_n , v_n e c_n , o algoritmo preditor/multicorretor pode ser resumido nas seguintes etapas:

Loop no tempo

Predição:

$$\mathbf{c}_{n+1}^0 = \mathbf{c}_n + 0,5 \Delta t \mathbf{a}_n$$

$$\mathbf{a}_{n+1}^0 = 0$$

Correção:

Loop não linear

Bloco 1 : Resolver a equação da pressão

$$\mathbf{K}(\mathbf{c}_{n+1}^i) \mathbf{p}_{n+1}^i = \mathbf{Q} \quad (3.118)$$

Bloco 2 : Resolver a equação da velocidade

$$\overline{\mathbf{M}}(\mathbf{c}_{n+1}^i) \mathbf{v}_{n+1}^i = \mathbf{f}_g(\mathbf{p}_{n+1}^{i+1}) + \mathbf{f}_{pos} \quad (3.119)$$

Bloco 3 : Resolver a equação da concentração

$$\mathbf{M}^* \Delta \mathbf{a} = \mathbf{R} \quad (3.120)$$

onde

$$\mathbf{M}^* = \widetilde{\mathbf{M}} + 0,5\Delta t\widetilde{\mathbf{C}} \quad (3.121)$$

$$\mathbf{R} = -\widetilde{\mathbf{M}}(\mathbf{a}_{n+1}^i) \mathbf{a}_{n+1}^i - \widetilde{\mathbf{C}}(\mathbf{u}_{n+1}^{i+1}, \mathbf{p}_{n+1}^{i+1}, \mathbf{c}_{n+1}^i) \mathbf{c}_{n+1}^i \quad (3.122)$$

Atualização das variáveis

$$\mathbf{c}_{n+1}^{i+1} = \mathbf{c}_{n+1}^i + 0,5\Delta a\Delta t \quad (3.123)$$

$$\mathbf{a}_{n+1}^{i+1} = \mathbf{a}_{n+1}^i + \Delta \mathbf{a} \quad (3.124)$$

Até obter a convergência

fim do Loop no tempo

No algoritmo acima o processo iterativo continua até que se atinja um critério de convergência pré-determinado, que é avaliado empregando-se as seguintes medidas de erros:

$$e_p = \frac{\|\mathbf{p}_{n+1}^{i+1} - \mathbf{p}_{n+1}^i\|}{\|\mathbf{p}_{n+1}^{i+1}\|} \quad (3.125)$$

$$e_u = \frac{\|\mathbf{u}_{n+1}^{i+1} - \mathbf{u}_{n+1}^i\|}{\|\mathbf{u}_{n+1}^{i+1}\|} \quad (3.126)$$

$$e_a = \frac{\|\Delta \mathbf{a}_{n+1}^{i+1}\|}{\|\mathbf{a}_{n+1}^{i+1}\|} \quad (3.127)$$

$$e_R = \frac{\|\mathbf{R}_{n+1}^{i+1}\|}{\|\mathbf{R}_{n+1}^i\|} \quad (3.128)$$

$$e_E = \sqrt{\frac{(\Delta \mathbf{a}_{n+1}^{i+1})^T \mathbf{R}_{n+1}^{i+1}}{(\Delta \mathbf{a}_{n+1}^{i=0})^T \mathbf{R}_{n+1}^{i=0}}} \quad (3.129)$$

onde $\|\cdot\|$ indica a norma Euclidiana, e_p , e_u , e_a são, respectivamente, os erros relativos à pressão, à velocidade e à derivada temporal da concentração. A quantidade e_R mensura o erro no balanço de massa da equação da concentração e e_E representa o erro relativo à norma da energia. Para se alcançar a convergência é preciso satisfazer as seguintes condições:

$$e_p \leq tol \quad (3.130)$$

$$e_u \leq tol \quad (3.131)$$

$$e_c = \max\{e_a, e_R, e_E\} \leq tol \quad (3.132)$$

onde tol uma tolerância pré-fixada. Se as condições (3.130) e (3.131) forem satisfeitas os valores da pressão e da velocidade permaneceram os mesmos durante as iterações não-lineares e o algoritmo continuará atualizando somente os valores da concentração. O processo iterativo se encerra quando a condição (3.132) é atingida.

No algoritmo bloco iterativo preditor-multicorretor, a cada iteração não-linear, são resolvidos três sistemas de equações lineares. É importante que na resolução dos sistemas lineares sejam utilizados métodos eficientes para melhoria do tempo de processamento.

3.5 Resolução dos Sistemas Lineares Resultantes

A formulação variacional semi-discreta conduz a um sistema acoplado de equações não-lineares que é dividido em dois sistemas lineares, pressão e velocidade, e um sistema de equações diferenciais ordinárias, concentração. Para solucionar esse sistema acoplado basta resolver esses três sistemas. A solução de tais sistemas pode ser obtida por dois tipos de métodos: diretos e iterativos.

Os métodos diretos são aqueles que forneceriam a solução exata, se não fossem os erros de arredondamento. No entanto, não são apropriados na resolução de problemas de grande porte pois na resolução desses sistemas ocasionam erros de arredondamento e tempos de processamento muito altos. Os métodos iterativos, por sua vez, são aqueles que a partir de uma condição inicial para a solução, geram uma sequência de aproximações, sendo que a cada etapa do processo as aproximações se tornam mais próximas da solução exata. Em princípio os métodos diretos poderiam ser utilizados para solucionar os sistemas lineares, no entanto, a medida que se aumenta o número de equações, esses métodos se tornam mais dispendiosos que os métodos iterativos em relação ao tempo de processamento e ao consumo de memória. Portanto, é interessante usar os métodos iterativos [6, 35].

Os métodos iterativos podem ser de dois tipos: estacionários e não-estacionários. Os estacionários não utilizam informações de iterações anteriores sendo simples de entender e implementar, mas, em geral, são menos eficientes que os métodos

não-estacionários. Em contra-partida, os métodos não-estacionários utilizam informações das iterações anteriores o que resulta em taxas mais elevadas de convergência [30]. Considerando a eficiência dos métodos iterativos não-estacionários, os sistemas de equações lineares referentes à pressão, à velocidade e à concentração serão resolvidos através desses métodos. Como cada sistema possui característica próprias não foi conveniente adotar o mesmo método na resolução dos três sistemas. Para o sistema de equação referente a pressão, a matriz \mathbf{K} é simétrica, logo para resolver o sistema foi empregado o método dos gradientes conjugados (GC) [55]. Já para a velocidade, apesar da matriz $\overline{\mathbf{M}}$ ser simétrica, ela perde a simetria com a aplicação das condições de contorno, portanto foi utilizado o método do resíduo mínimo generalizado (GMRES) [56]. Por fim, o GMRES também é utilizado para resolver o sistema da concentração devido a sua falta de simetria.

Os métodos GC e GMRES são métodos iterativos não-estacionários baseados nos sub-espacos de Krylov. O GMRES foi proposto por Saad e Schultz para solucionar sistemas lineares não simétricos calculando uma solução que minimiza a norma residual sobre uma base dos sub-espacos de Krylov [55]. A implementação originalmente proposta usa a ortogonalização de Gram-Schmidt na construção dos vetores de Krylov. No entanto, problemas surgem quando o número de vetores armazenados na base cresce consideravelmente, assim, uma solução é reiniciar o processo a cada ciclo de m iterações, sendo então o método denominado GMRES(m). Esses *restart* resultam no condicionamento da convergência ao valor de m . A grande dificuldade do processo de *restart* fica na escolha do m , para m muito pequeno a convergência é lenta ou pode nem vir a ocorrer. Entretanto, para m muito grande o método fará um esforço excessivo, consumindo mais tempo e memória [6, 55].

Nos métodos iterativos GMRES e Gradiente Conjugado, as principais operações envolvidas na solução dos sistemas lineares são o produto matriz-vetor e o produto interno entre dois vetores. O desempenho na realização dessas operações impacta diretamente no custo de processamento desses métodos. Quanto maiores são os sistemas maior é o uso de memória e mais custosos são esses cálculos, conseqüentemente maior é o tempo de convergência dos métodos. Mesmo utilizando técnicas especiais de armazenamento que visam reduzir o armazenamento de coeficientes nu-

los da matriz global de discretização, ainda assim, essas operações utilizam uma grande quantidade de memória e grande parte dos cálculos decorrem delas. Uma alternativa para melhorar o desempenho é a paralelização do processo de solução de tais problemas.

3.6 Considerações Finais

Fica claro que, mesmo depois de ter modelado o problema matematicamente e discretizado-o numericamente, a solução computacional pode se tornar inviável devido às limitações de tempo de processamento e de consumo de memória. Essas limitações, estão intrinsecamente relacionadas ao tamanho dos sistemas que se deseja solucionar. Mesmo utilizando técnicas de armazenamento especiais para reduzir o armazenamento de elementos nulos, como elemento-por-elemento, aresta-por-aresta, dentre outras, tem-se que, o tamanho dos sistemas pode ser tal, que as principais operações dos métodos iterativos necessitam de muita memória e tempo para serem realizadas inviabilizando a solução em uma única máquina. Portanto, é necessário o uso do processamento paralelo para otimizar a resolução desses problemas, ou mesmo viabilizá-la em alguns casos.

A paralelização está relacionada a divisão do problema em partes menores, de forma que essas partes sejam tratadas separadamente através da execução em vários processadores [64]. Assim, o domínio será dividido em subdomínios, os quais serão enviados aos diversos processadores que executarão operações idênticas sobre esses subdomínio resolvendo, assim, o problema em conjunto [16]. Nesse sentido, as operações produto matriz-vetor e produto interno dos métodos iterativos serão realizadas de forma distribuída entre os vários processadores, ocasionando um ganho computacional na realização dessas tarefas. No entanto, é importante que a divisão do domínio seja realizada de forma balanceada, evitando que alguns processadores trabalhem muito e outros fiquem ociosos [16, 35]. Outro ponto importante é tentar minimizar as fronteiras entre as partições para diminuir a comunicação entre os processadores.

Capítulo 4

Estratégias Paralelas

O modelo numérico constitui uma ferramenta útil para os engenheiros de petróleo na realização de simulações de recuperação de hidrocarbonetos e mesmo para os hidrologistas na análise do impacto ambiental do transporte de poluentes em reservatórios de águas subterrâneas. No entanto, muitas vezes essas simulações resultam em soluções imprecisas devido ao uso de malhas grossas. Assim, para melhorar a precisão dos resultados é preciso utilizar malhas mais refinadas, sendo que simulações em malhas finas demandam enorme tempo de processamento e grande consumo de memória. A solução é usar computação paralela para atender à demanda computacional requerida na resolução desses problemas.

4.1 Computação Paralela

Nas últimas décadas, motivada pela crescente demanda por poder computacional para que aplicações complexas pudessem ser resolvidas com o auxílio da computação, a indústria buscou a contínua evolução dos microprocessadores pois se acreditava que o advento de processadores cada vez mais rápidos e eficientes supriria essa demanda. No entanto, essa busca atingiu seus limites físicos, principalmente no que diz respeito ao aquecimento. Tal limitação levou ao surgimento de tecnologias alternativas em busca de um elevado desempenho computacional, resultando em um

campo da computação denominado computação paralela [49].

A computação paralela pode ser entendida como uma evolução da computação sequencial em que cada instrução de um programa é executada, uma após a outra, pelo processador. Na computação paralela busca-se um paralelismo de ações que está relacionado à execução de uma mesma tarefa em mais de um elemento de processamento, simultaneamente [64]. Nesse contexto, um computador paralelo pode ser entendido como aquele formado por vários elementos de processamento trabalhando em conjunto na resolução de determinados problemas [21]. Tal conceito abrangeria, dentre outros, os supercomputadores, os computadores multiprocessados e os clusters.

Os computadores paralelos podem ser classificados sob diferentes aspectos arquiteturais. Será destacada a tradicional classificação de Flynn [20] e a classificação segundo o compartilhamento de memória. Flynn baseou-se no número de fluxos de instruções e no número de fluxos de dados para classificar os computadores. Combinando esses dois fatores tem-se quatro classes: (SISD) único fluxo de instrução e único fluxo de dado, (SIMD) único fluxo de instrução e múltiplos fluxos de dados, (MISD) múltiplos fluxos de instruções e único fluxo de dado e (MIMD) múltiplos fluxos de instruções e múltiplos fluxos de dados. Na classe SISD estão os tradicionais computadores sequenciais. Os processadores vetoriais encaixam-se na classe SIMD, e a maioria dos computadores paralelos estão na classe MIMD [19, 21, 64].

De acordo com o uso de memória tem-se máquinas com memória compartilhada (multiprocessadores) e máquinas com memória distribuída (multicomputadores). As máquinas com memória compartilhada possuem um único espaço de endereçamento comum a todos os processadores. Elas se subdividem em Acesso uniforme à memória (UMA) e Acesso não uniforme à memória (NUMA). Em UMA o acesso à memória se dá de forma compartilhada e centralizada e o tempo gasto no acesso à uma mesma posição de memória é igual para todos os processadores. Em NUMA o acesso à memória é de forma compartilhada e distribuída e os processadores acessam a memória em tempos diferenciados segundo a localização física da parte acessada [63]. Os processos interagem entre si através de um espaço de endereçamento comum

a todos, no qual podem ler e escrever dados [21]. Nas máquinas com memória distribuída, o espaço de endereçamento consiste de múltiplos espaços privados de memória destinados a cada processador. A comunicação entre os processos durante a execução se dá por trocas de mensagem pela rede de interconexão [19].

A evolução nas tecnologias de redes de interconexão permitiu o crescimento dos sistemas distribuídos, que são coleções de computadores independentes, interconectados por uma rede de comunicação rápida e que aparentam, aos olhos do usuário, um sistema singular e coerente [64]. Os computadores que formam o sistema são autônomos e podem ser tanto homogêneos quanto heterogêneos em sua composição. Dentre os sistemas distribuídos estão os *Clusters* conhecidos como agregados de computadores e as grids ou grades computacionais [19].

Um *Cluster* é definido como um conjunto de computadores independentes e interconectados que formam uma plataforma de alto desempenho para executar aplicações de forma paralela [5]. Cada computador do *Cluster* é denominado nó e pode ser multiprocessado ou monoprocessado. Antigamente, nós monoprocessados eram quase que predominantemente utilizadas em *Cluster*. Hoje, nós com processadores *multi-core* que oferecem suporte nativo ao processamento paralelo, são predominantes.

Em um *Cluster* os nós devem ser interconectados por uma rede de alta velocidade. Usualmente, a tecnologia de rede utilizada na composição de *Clusters* é a *Ethernet* interconectada com *switch*, que permite operações de comunicação nos dois sentidos (duplex) a uma taxa de 1Gbits/s, quando se trata da categoria *Gigabit Ethernet*. A Fig. 4.1 ilustra a interconexão dos processadores por meio do *switch* na tecnologia de rede *Ethernet*.

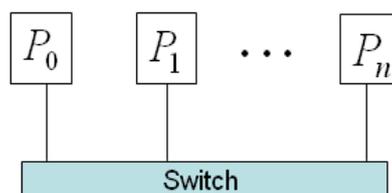


Figura 4.1: Esquema de interconexão para *Ethernet*.

Outra opção mais eficiente, no entanto, economicamente menos acessível, é o emprego da tecnologia Infiniband [19]. Nessa tecnologia todos os processadores estão completamente conectados uns com os outros em uma topologia *crossbar* que pode ser visualizada na Fig. 4.2. Além disso, a placa de rede dispõe de um processador dedicado ao processo de comunicação e os processadores fazem acesso direto a memória principal sem interromper o sistema operacional para realizar a comunicação.

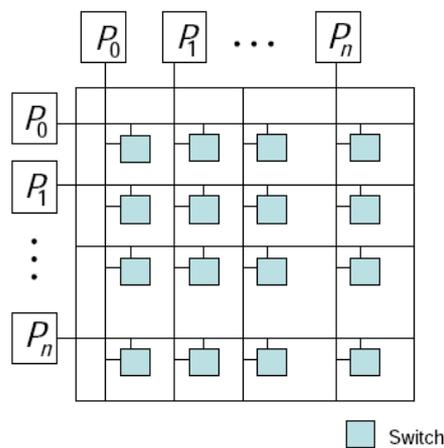


Figura 4.2: Esquema de comunicação *crossbar* para *Infiniband*.

A conexão de rede desempenha um papel muito importante nos *Clusters*, uma vez que interferem no desempenho das aplicações, por causa do tempo de comunicação entre os nós. Como a memória do *Cluster* é distribuída, a comunicação entre os nós será realizada pelo envio e recebimento de mensagens. Esse serviço é oferecido por algumas bibliotecas e linguagens, tais como PVM (*Parallel virtual Machine*)[24] e padrão MPI (*Message Passing Interface*) [42] com o paradigma simplesmente de troca de mensagens, o *TreadMarks* [32], com paradigma de memória compartilhada distribuída e a linguagem *High Performance Fortran* HPF [26] com o paradigma de dados paralelos. O padrão MPI foi o escolhido para ser utilizado nesse trabalho por demonstrar maior desempenho em relação ao *TreadMarks* e HPF e maior versatilidade comparado ao PVM [35].

<code>#include <stdio.h></code>	<code>#include <stdio.h></code> <code>#include "mpi.h"</code>
<code>main() {</code>	<code>main (int argc, char** argv) {</code>
	<code>MPI_Init(& argc, & argv);</code> <code>MPI_Comm_size(MPI_COMM_WORLD ,&ProcNum)</code> <code>MPI_Comm_rank(MPI_COMM_WORLD ,&ProcID)</code>
código	código
	<code>MPI_Finalize();</code>
<code>}</code>	<code>}</code>

Tabela 4.1: Comparativo da estrutura de um programa C, em serial (à esquerda) e paralelo (à direita), utilizando as rotinas básicas de gerenciamento para o padrão MPI.

4.2 MPI (Message Passing Interface)

A MPI (*Message Passing Interface*) pode ser utilizada em equipamentos de diferentes fabricantes [48]. Dentre as bibliotecas mais conhecidas que implementam o padrão MPI estão as bibliotecas LAM e MPICH. Nesse trabalho foi adotada a biblioteca LAM [35, 39].

O padrão MPI especifica a sintaxe e a semântica para um conjunto de funções que podem ser divididas, basicamente, em três grupos: rotinas de gerenciamento; rotinas de comunicação ponto a ponto, que envolvem a comunicação entre dois processos; e rotinas de comunicação coletiva, que envolvem todos os processos de um grupo. Essas funções são usadas para viabilizar a implementação de programas paralelos [48]. As rotinas de gerenciamento têm o propósito de inicializar e finalizar o ambiente MPI. A Tab. 4.1 ilustra as diferenças entre um programa escrito em C, na sua versão serial e paralela, inserindo as principais rotinas desse grupo.

A primeira diferença está na inclusão de `mpi.h` que contém as definições necessárias para a compilação de funções MPI. As rotinas de gerenciamento chamadas são: `MPI_Init`, que inicializa o ambiente e precisa ser invocada antes de qualquer outra rotina do padrão; a rotina `MPI_Comm_size` determina o número de processos que está

sendo utilizado na aplicação e a rotina `MPI_Comm_rank` devolve o identificador do processo. Ambas utilizam o parâmetro `MPI_COMM_WORLD` que especifica o conjunto de processos que podem ser invocados. Por fim, a rotina `MPI_Finalize` é chamada para finalizar o ambiente de execução [48].

As rotinas de comunicação são destinadas as trocas de mensagens entre os processos. Elas podem ser ponto a ponto e coletivas. Em relação ao modo de transmissão das mensagens há o bloqueante e não-bloqueante. O primeiro modo não permite quaisquer execuções até que a rotina chamada termine, ou seja, enquanto a comunicação não for finalizada, os processos ficam parados. No modo não-bloqueante, o processo é liberado para continuar sua execução após a chamada da rotina. Esse modo de transmissão não garante que o conteúdo da mensagem esteja seguro, cabendo ao programador tomar as medidas necessárias para que não ocorra modificações na mensagem [49]. Nesse trabalho foram adotados as rotinas que utilizam o modo de transmissão bloqueante.

As funções `MPI_Send` e `MPI_Recv` são as funções básicas do grupo de comunicação ponto a ponto e se destinam a comunicação entre dois processos. A função `MPI_Send` envia uma mensagem para um dado processo e a função `MPI_Recv` recebe a mensagem enviada por esse dado processo. Essas funções exigem algumas informações. Na rotina *send* é preciso especificar a mensagem a ser enviada, o seu tamanho, o tipo de dados que a compõe, o processo a que se destina, uma identificação do envio e o grupo do processo. Na rotina *receive* é definido um local para armazenar a mensagem recebida, o seu tamanho, o seu tipo, o processo que a enviou, uma identificação do envio da mensagem, o grupo de processos e uma variável que armazena o *status* do envio.

Nos grupos de comunicação coletiva, várias comunicações ponto a ponto são executadas, a partir de um único comando. Nesse grupo, as principais funções são *gather*, *scatter* e *reduce*. As demais funções são combinações ou variações dessas [48]. A função *scatter* envia dados distintos do processo raiz para os demais processos do grupo. A função *gather* recebe mensagens distintas de cada processo do grupo e a função *reduce* aplica alguma operação de redução sobre os dados de todos os pro-

cessos e armazena o resultado no processo raiz. Algumas variações são: a função *allgather* que reúne os dados de todos os processos do grupo, a função *allreduce* que aplica uma operação de redução e armazena o resultado em todos os processos do grupo, *reduce_scatter* e que combina as duas primitivas, dentre outras.

Uma das principais limitações na comunicação por troca de mensagens é o tempo de comunicação. Vários fatores podem influenciar esse tempo, tais como, a rede de interconexão, o tipo de comunicação (bloqueante ou não bloqueante), o tamanho das mensagens enviadas e a proporção das comunicações em relação à computação local [49]. No que se refere a quantidade de comunicações em relação ao número de computação, uma medida simples que pode ser tomada é o particionamento do domínio representado pela malha, de forma a minimizar o número de fronteiras entre as partições [39].

4.3 Geração e Particionamento de Malhas

No processo de discretização o domínio contínuo é mapeado em uma estrutura composta por um número finito de pontos denominada malha [22]. No método dos elementos finitos a malha divide o domínio em elementos geométricos, que podem ter tanto forma de triângulos ou quadriláteros, no plano, quanto tetraedros ou hexaedros, no espaço. No que se refere à sua conectividade, as malhas podem ser de dois tipos: estruturadas e não estruturadas [35].

Nas malhas estruturadas todos os pontos pertencente ao interior da malha são adjacentes ao mesmo número de elementos e é possível identificar os vizinhos de um ponto somando índices, por exemplo, dado o ponto (i,j) seus vizinhos serão $(i,j+1)$, $(i+1,j)$, $(i,j-1)$ e $(i-1,j)$ [22]. Nas malhas não estruturadas o número de elementos adjacentes a um ponto não é constante. Essas malhas fornecem uma melhor modelagem de geometrias complexas e irregulares [60]. A Fig. 4.3 ilustra duas malhas triangulares, uma estruturada, Fig. 4.3(a), e a outra não-estruturada, Fig. 4.3(b). Dentre as vantagens do método dos elementos finitos em relação a outros métodos, está sua capacidade de tratar geometrias complexas por meio de malhas não-estruturadas

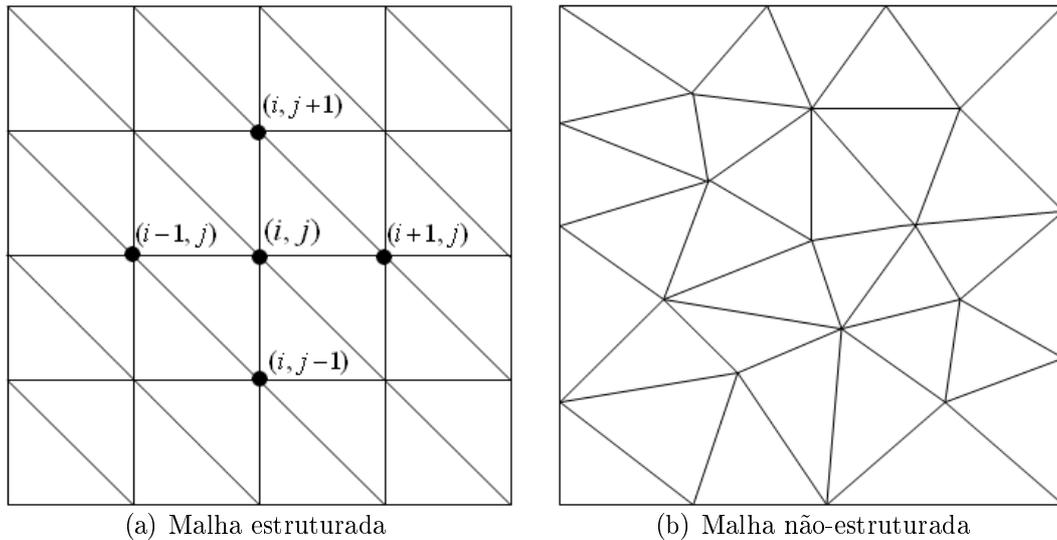


Figura 4.3: Tipos de malhas.

[6, 35]. A geração de malhas tanto estruturadas quanto não-estruturadas pode ser feita através de vários pacotes de domínio público, tais como *triangle* [59], *easyMesh* [45], *Gid* [53], *GMSH* [25], etc. Uma lista ds principais pacotes da área de geração, particionamento e visualização de malhas pode ser vista em [57].

O *easyMesh* gera malhas não-estruturadas bidimensionais em domínios genéricos, que podem ter buracos e serem formados por materiais distintos. As malhas podem ser visualizadas pelo software *showMesh* que permite exibir a numeração dos pontos, dos elementos, das fronteiras além de disponibilizar recursos de zoom e rotação de malha [22]. Para a geração das malhas é necessário um arquivo de entrada contendo o número de vértices, as coordenadas dos vértices, a precisão do particionamento, a quantidade de segmentos que compõem o domínio e quais vértices formam o segmento. A partir desse arquivo são gerados três outros: o primeiro com as coordenadas dos vértices da malha, o segundo com a conectividade dos triângulos e o terceiro com a vizinhança dos elementos.

A malha desempenha um papel fundamental na precisão da solução. O espaço entre os pontos da malha determina a acurácia com que a solução numérica será obtida, uma vez que esse espaçamento está relacionado ao tamanho do sistema linear [6]. Basicamente, quanto mais pontos a malha tiver, maior será o sistema resultante da discretização e, conseqüentemente, mais elevado será o tempo de processamento.

Portanto, uma primeira ação a ser tomada visando o processamento paralelo é a divisão do domínio em subdomínios, os quais serão distribuídos entre os processadores do *Cluster*.

O particionamento da malha deve levar em conta uma distribuição balanceada entre os processadores, de forma que o número de elementos seja aproximadamente o mesmo em cada processador. Além disso, o número de pontos entre as partições deve ser o menor possível para minimizar a quantidade de comunicação [35]. A Fig. 4.4(a) mostra que a divisão da malha ocorreu de forma balanceada, no entanto há muita comunicação entre as partições. Na Fig. 4.4(b), por sua vez, a comunicação está minimizada, no entanto a distribuição dos elementos não está balanceada. A Fig. 4.4(c) mostra uma divisão balanceada com o mínimo de comunicação.

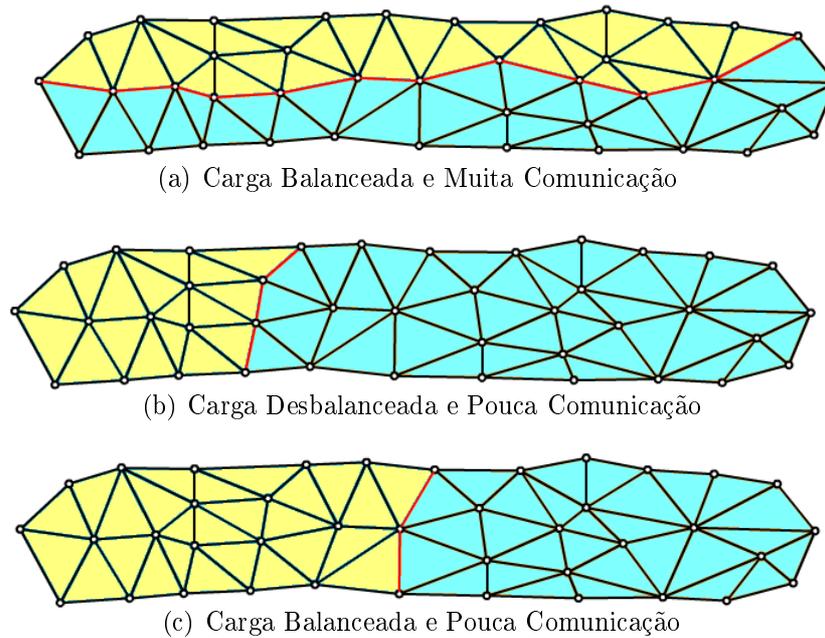


Figura 4.4: Exemplos de particionamentos de malha com e sem balanceamento de carga e minimização da comunicação.

Atender as exigência de balanceamento, como ilustrado na Fig. 4.4(a) não exige muito esforço, no entanto, minimizar o número de comunicações mantendo o balanceado não é uma tarefa simples. Esse problema pode ser modelado como um problema de particionamento de grafos [7, 39]. Existem várias bibliotecas de domínio público que implementam algoritmos de particionamento de grafos para realizar o

particionamento de malhas, dentre as quais, as mais conhecidas são METIS [31] e Jostle [67]. Nesse trabalho, optou-se pelo METIS.

O software METIS é um conjunto de funções implementadas na linguagem C para realizar o particionamento de malhas, grafos e matrizes. Ele se destaca devido a alta qualidade do particionamento e por tratar tanto malhas estruturadas quanto não-estruturadas. Inicialmente, o programa converte a malha em um grafo, em seguida realiza o particionamento. Existem dois tipos de particionamento, o dual e o nodal. No primeiro, cada elemento da malha torna-se um vértice do gráfico que será particionado. No segundo, é o nó da malha que converte-se em vértice do grafo. No particionamento dual, o METIS disponibiliza o software PartMetis que lê o arquivo com as características da malha e gera outro arquivo composto por *nel* linhas, onde cada linha contém o número do processador que será encarregado pelo elemento, sendo *nel* o número de elementos da malha. O METIS suporta quatro tipos de elementos, triangular, quadrilátero, tetraedro e hexaedro.

A visualização das malhas particionadas pelo METIS, pode ser feita através do programa PMVIS (*Partitioned Mesh Visualizer*) [47]. Esse programa possibilita a visualização de malhas bidimensionais e tridimensionais, com recursos de zoom, rotação, etc. A Fig. 4.5(a) exibe um domínio quadrado, discretizado por uma malha de 40×40 células, sendo cada célula dividida em dois elementos, totalizando 3200 elementos triangulares e 1681 nós. Na Fig. 4.5(b) é mostrado o mesmo domínio particionado em 16 subdomínios, onde pode ser observado que o METIS numera as partições próximas com números consecutivos. Isso permite que, na distribuição dos processos em ambientes *multi-core*, os processos vizinhos fiquem na mesma máquina, reduzindo o tempo de comunicação via interface de rede, uma vez que, nesse tipo de ambiente, o MPI usa a memória para realizar a troca de mensagens entre os processos.

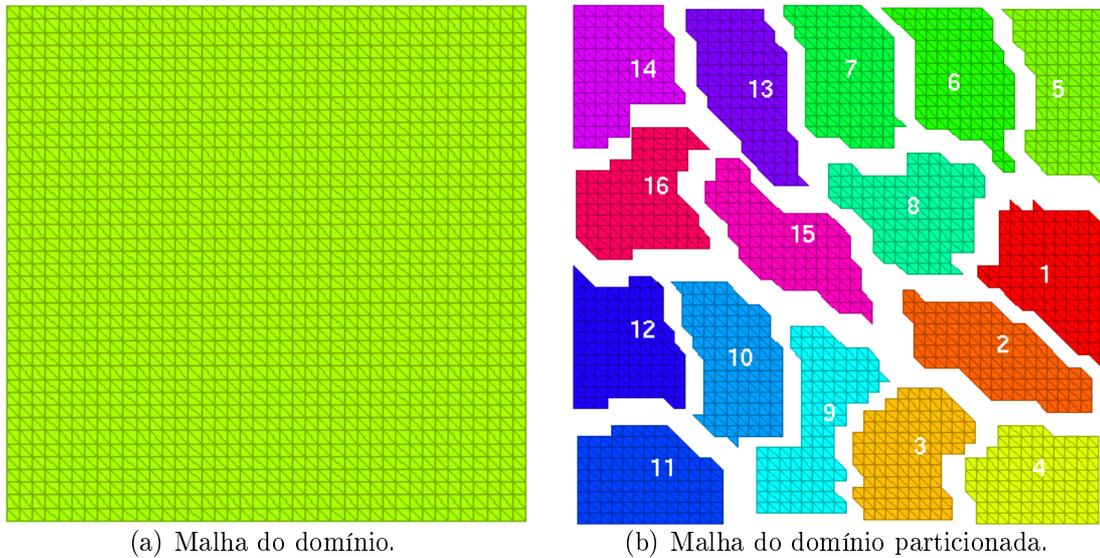


Figura 4.5: Exemplo de particionamento de malhas pelo METIS.

4.4 Decomposição de Domínio

Uma maneira natural de se criar algoritmos paralelos é através dos métodos de decomposição de domínio. Tais métodos atuam propondo a divisão do domínio em subdomínios através de algoritmos de particionamento. Assim, o problema original é tratado como uma série de subproblemas de tamanho reduzido. No ambiente de computação paralela, cada subdomínio é enviado para um processador diferente para que seja resolvido concorrentemente. A solução global do sistema é alcançada combinando a solução dos subsistemas que foram resolvidos localmente [55].

Os métodos de decomposição de domínio podem ser divididos em duas classes, os que têm sobreposição de domínio, também conhecidos como *overlapping*, os quais incluem os métodos de Schwarz, e os que não possuem sobreposição de domínio, *non-overlapping*, como o método de Shur [55]. No domínio com sobreposição existe uma região de sobreposição próxima a fronteira. Nessa região, as variáveis associadas aos nós pertencem a mais de um subproblema. No domínio sem sobreposição somente as variáveis associadas aos nós da fronteira participam de mais de um subproblema. A Fig. 4.6 mostra exemplos de domínios sem sobreposição, Fig. 4.6(a) e com sobreposição, Fig. 4.6(a).

Nesse trabalho será abordado o método de Shur [55], mais especificamente o método

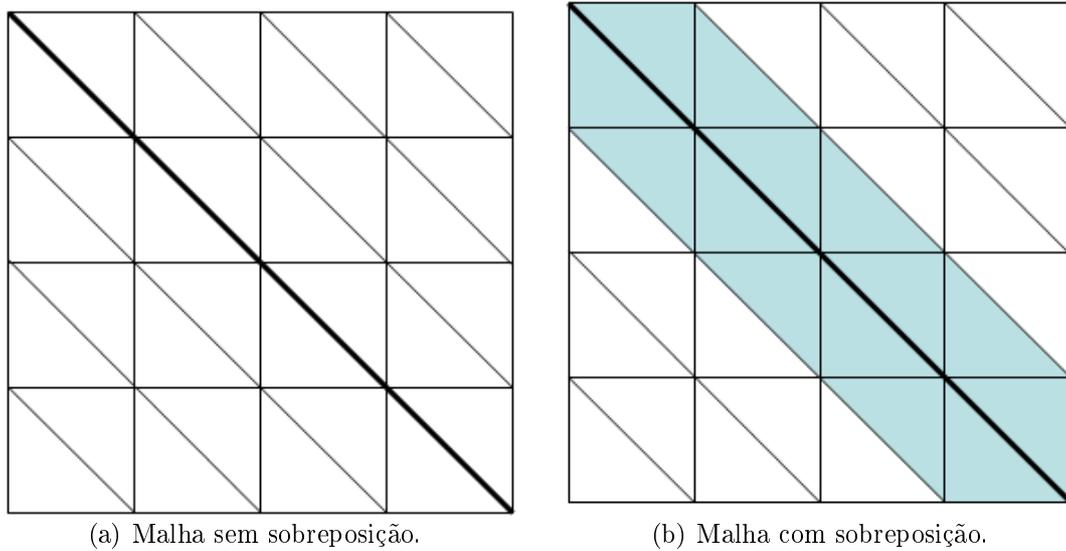


Figura 4.6: Sobreposição de malhas.

complemento de Shur. Existem algumas variantes desse método. A abordagem adotada pode ser vista em detalhes nos trabalhos de [29, 35]. Ela se assemelha a variante descrita por Saad [55], em que os nós do subdomínio são ordenados de forma que, em cada subdomínio, os nós internos ao subdomínio são listados primeiro, para depois listar os nós de fronteira. Isso ocasiona uma comunicação mais eficiente entre os processos. Adotando essa definição ou classificação, o vetor de incógnita local pode ser dividido em duas partes.

$$\mathbf{x} = \begin{bmatrix} \underline{u}_i \\ \underline{y}_i \end{bmatrix} \quad (4.1)$$

onde \underline{u}_i representam os nós incógnitas no interior do subdomínio e \underline{y}_i os nós incógnitas da fronteira. Da mesma forma o vetor dos termos independentes será:

$$\mathbf{b} = \begin{bmatrix} \underline{f}_i \\ \underline{g}_i \end{bmatrix} \quad (4.2)$$

onde \underline{f}_i representam os nós no interior do subdomínio e \underline{g}_i os nós da fronteira.

A matriz local pertencente a cada subdomínio, \mathbf{A}^i , apresenta a seguinte estrutura:

$$\mathbf{A}^i = \begin{bmatrix} \mathbf{A}_i & \mathbf{B}_i \\ \mathbf{C}_i & \mathbf{A}_s \end{bmatrix} \quad (4.3)$$

onde \mathbf{A}_i é a matriz associada aos nós internos ao subdomínio, \mathbf{B}_i e \mathbf{C}_i representam as iterações entre os nós internos e os nós da fronteira, \mathbf{A}_s está relacionada as con-

tribuições dos nós da fronteira do subdomínio e i varia segundo o número de divisões do domínio.

A técnica descrita em [29] obtém um efeito semelhante classificando os nós do domínio em três tipos: *IntNodes*, *IBNodes* e de valor prescrito. Os nós *IntNodes* são os nós incógnitas que pertence ao interior do subdomínio. Os nós *IBNodes* são os nós incógnitas que pertencem à fronteira entre os subdomínios e os nós de valor prescrito são os nós, nos quais a solução é conhecida. Para tornar mais claro esse entendimento, considere a Fig. 4.7 que representa uma malha dividida em quatro subdomínios. Os nós 8 e 14 são exemplos de nós *IntNodes* para as partições 3 e 4, respectivamente. O nó 7 é *IBNodes* pois pertence a fronteira das partições 2 e 3, e os nós 1 e 25 são exemplos de nós com valor prescrito.

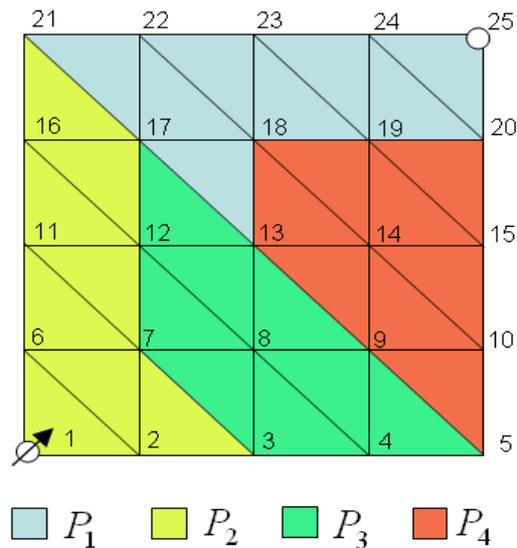


Figura 4.7: Malha pequena com 4 partições.

Adotando essa classificação, o vetor local de incógnitas da partição i terá o seguinte formato:

$$\mathbf{x}_i = \begin{bmatrix} \underline{u}_i \\ \underline{u}_s \end{bmatrix} \quad (4.4)$$

onde \underline{u}_i representam os nós incógnitas relativas aos nós *IntNodes* e \underline{u}_s é um vetor relacionado aos nós *IBNodes*, sendo formado pela união das contribuições das partições fronteiras. De forma semelhante o vetor dos termos independentes é dado

por:

$$\mathbf{b}_i = \begin{bmatrix} \underline{v}_i \\ \underline{v}_s \end{bmatrix} \quad (4.5)$$

onde \underline{v}_i são os nós referentes aos nós *IntNodes* e \underline{v}_s está relacionado aos nós *IBNodes*.

A matriz local, A_i , pertencente a partição i , apresenta a seguinte estrutura:

$$\mathbf{A}_i = \begin{bmatrix} \mathbf{A}_{II} & \mathbf{B}_{IB} \\ \mathbf{C}_{BI} & \mathbf{A}_{BB} \end{bmatrix} \quad (4.6)$$

onde A_{II} representam as contribuições dos nós do tipo *IntNodes* da partição i nos nós do tipo *IntNodes* da mesma partição i . A matriz B_{IB} armazena as contribuições que os nós do tipo *IntNodes* da partição i têm sobre os nós do tipo *IBNodes* também da partição i . De forma similar, os blocos C_{BI} representam as contribuições dos nós do tipo *IBNodes* da partição i nos nós *IntNodes* da partição i . E a matriz A_{BB} está relacionada as contribuições dos nós do tipo *IBNodes* da partição i nos nós do tipo *IBNodes* da mesma partição i .

Com isso, as operações locais relativas ao produto matriz-vetor e produto interno entre dois vetores globais \underline{u} e \underline{v} , podem ser escritas como:

$$\underline{v}_i = A_{II}\underline{u}_i + B_{IB}\underline{u}_s \quad (4.7)$$

$$\underline{v}_s = C_{BI}\underline{u}_i + A_{BB}\underline{u}_s \quad (4.8)$$

para $i = 1, 2, \dots, p$ e,

$$\underline{u} \cdot \underline{v} = \sum_{i=1}^p (\underline{u}_i \cdot \underline{v}_i + \underline{u}_s \cdot \underline{v}_s) \quad (4.9)$$

Considerando um particionamento geral em p subdomínios, distribuídos entre p processadores, o sistema linear associado ao problema terá a seguinte estrutura:

$$Ax = \begin{bmatrix} A_{II}^1 & & & B_{IB}^1 \\ & A_{II}^2 & & B_{IB}^2 \\ & & \ddots & \vdots \\ & & & A_{II}^p & B_{IB}^p \\ C_{BI}^1 & C_{BI}^2 & \dots & C_{BI}^p & A_{BB}^* \end{bmatrix} \begin{bmatrix} \underline{x}_1 \\ \underline{x}_2 \\ \vdots \\ \underline{x}_p \\ \underline{x}_s^* \end{bmatrix} = \begin{bmatrix} \underline{b}_1 \\ \underline{b}_2 \\ \vdots \\ \underline{b}_p \\ \underline{b}_s^* \end{bmatrix} = \underline{b} \quad (4.10)$$

A matriz A_{BB}^* é a união de várias matrizes ao longo dos p processadores, de forma que, $A_{BB}^* = \bigcup_{i=1}^p A_{BB}^i$, onde cada matriz A_{BB}^i armazena as contribuições dos nós do tipo *IBNodes* nos nós do tipo *IBNodes* do processador i . Os vetores \underline{x}_i e \underline{b}_i , com $i = 1, \dots, p$, referem-se as incógnitas relativas aos nós do tipo *IntNodes* e os termos independentes para o processador i . Os vetores \underline{x}_s e \underline{b}_s , de forma semelhante a A_{BB}^* , são gerados pela montagem de vários vetores ao longo dos p processadores, portanto, $\underline{x}_s^* = \bigcup_{i=1}^p \underline{x}_s^i$ e $\underline{b}_s^* = \bigcup_{i=1}^p \underline{b}_s^i$, onde \underline{x}_s^i e \underline{b}_s^i representam, respectivamente, as incógnitas e os termos independentes dos nós do tipo *IBNodes* do processador i . Detalhes podem ser vistos em [35].

Na resolução dos sistemas associados ao problema, as operações básicas produto matriz-vetor e produto interno entre dois vetores podem ser definidas da seguinte forma [35]:

$$\underline{v}_i = A_{II}^i \mathbf{x}_i + B_{IB}^i \mathbf{x}_s \underline{v}_s^* = C_{BI}^i \underline{u}_i + A_{BB}^* \underline{u}_s^* \quad (4.11)$$

e ,

$$\underline{u} \cdot \underline{v} = \sum_{i=1}^p (\underline{u}_i \cdot \underline{v}_i + \underline{u}_s^* \cdot \underline{v}_s^*) \quad (4.12)$$

Cada vez que o produto matriz-vetor é realizado é preciso comunicar as alterações relativas aos nós da fronteira através de uma comunicação entre os processadores que participam dessa fronteira. O produto interno também necessita de uma comunicação, no entanto, essa comunicação é global, pois há o uso de todos os elementos do vetor e o resultado da operação deve ser comunicado a todos os processadores. Essas comunicações são realizadas por funções específicas do padrão MPI chamadas em algumas rotinas na aplicação desenvolvida nesse trabalho.

4.5 Algoritmo Paralelo

O processo para criação do programa paralelo que implementa as idéias até agora apresentadas consiste de três fases distintas, denominadas: pré-processamento, processamento e pós-processamento. O pré-processamento é a fase de criação e preparação dos dados. O processamento em si é a execução do algoritmo e o pós-processamento

é a apresentação dos resultados alcançados pelo processamento.

A fase de pré-processamento é necessária antes da execução do algoritmo para a criação e preparação dos dados a serem processados. Essa fase consiste em discretizar o domínio através da criação de um arquivo de malha, em seguida, particionar tal malha ocasionando a criação dos arquivos para cada submalha. Cada um desses arquivos conterá as características geométricas das submalhas tais como número de elementos, número de nós, coordenadas dos nós, conectividade dos elementos, etc. Além disso, esses arquivos ainda conterão informações sobre a classificação dos nós da malha, ou seja, se o nó é *IntNode*, *IBNode* ou valor prescrito, assim como, uma lista mapeando os nós *IBNodes* às partições que esses nós pertencem.

A parte de criação de dados, principalmente no que diz respeito a criação de malhas não-estruturadas e o particionamento das malhas, é realizada por softwares de domínio público. Nas malhas estruturadas foi empregado um gerador de malha simples. Gerada a malha, ela deve ser particionada. Assim, o programa *preProcess* lê os dados da malha, utiliza as funções do METIS para particioná-la e cria os arquivos das submalhas em pastas específicas. A ideia do *preProcess* está bem detalhada em [29, 35].

Uma vez que os dados foram criados e preparados, a execução do algoritmo pode ser iniciada. É importante lembrar que cada processo executa o mesmo algoritmo, o que muda são os dados de entrada que se referem aos subdomínios que são distintos. A estrutura do programa é bem simples. Basicamente a função principal, depois de chamar as funções de inicialização do padrão MPI, lê os arquivos com as características da malha, armazena os dados em algumas estruturas básicas e, depois de alguma manipulação específica sobre esses dados, é iniciada a execução da função preditor-multicorretor descrita em detalhes na Seção 3.4. Por fim, os resultados são armazenados em arquivos. O resultado do processamento é obtido reunindo esses arquivos, pois cada processo gerará um resultado, e depois exibindo o arquivo resultante na forma gráfica.

É importante frisar que a primeira vista, o código não aparenta ser paralelo. Isso porque apenas três funções utilizam as primitivas do MPI. A função *main* onde

estão explícitas as diretivas de gerenciamento, responsáveis por iniciar e finalizar o ambiente MPI, a função *Update* na qual são usadas as rotinas de comunicação ponto a ponto, *send* e *receive*, para comunicar o resultado do produto matriz-vetor, e a função *InnerProduct*, que invoca a rotina *allreduce* que pertence ao grupo de comunicação coletiva. Essa função distribui entre todos os processadores o resultado do produto interno entre dois vetores. As funções *Update* e *InnerProduct* podem ser encontradas em detalhes nos trabalhos de [29, 35].

Capítulo 5

Metodologia Experimental

Neste trabalho será utilizado todo o poder computacional dos *Clusters* multi-core para simular o escoamento miscível em meios porosos usando malhas refinadas. As simulações foram realizadas no Laboratório de Computação de Alto Desempenho da Universidade Federal do Espírito (LCAD) e no Núcleo de Computação de Alto Desempenho da Universidade Federal do Rio de Janeiro (NACAD). Na avaliação experimental dos algoritmos paralelos foram utilizadas métricas como o tempo de processamento, o *speedup* e a eficiência.

5.1 Ambiente Computacional

A execução do algoritmo paralelo foi realizada nos dois *Clusters* de computadores *multi-core* do LCAD, um composto por 31 máquinas *dual-core*, totalizando 62 núcleos (Enterprise 2) e outro formado por 29 máquinas *quad-core*, totalizando 116 núcleos (Enterprise 3). Além desses, os testes também foram executados no *Cluster* SGI Altix 450 do NACAD. O *Cluster* Altix é composto por 36 CPUs Dual Core Intel Itanium2 totalizando 72 núcleos. Cada teste foi executado 3 vezes e o tempo de execução adotado foi a média entre os valores obtidos.

Nas Enterprises, todos os nós operam com o sistema Linux Fedora 8.0, baseado no kernel versão 2.6.26.6-49.fc8. Os *Cluster* estão interconectados por uma rede de

interconexão *Gigabit Ethernet* composta por um *switchs* 3COM modelo 4200G que possui 48 portas. Os nós de processamento do *Cluster Enterprise 2* são máquinas equipadas com processador Intel Core 2 4300 1,8GHZ, cache L2 de 2 MB, memória RAM de 2GB e disco rígido de 250 GB. Já as máquinas do *Cluster Enterprise 3* operam com processador Intel 2 Q6600 2,4GHZ, cache L2 de 4 MB, memória RAM de 2GB e 160 GB de disco rígido. As máquinas servidoras possuem configurações semelhantes aos nós de processamento, exceto por possuírem duas placas de rede.

As máquinas servidoras também são responsáveis pela distribuição dos processos (*jobs*) entre os núcleos da Enterprise 2 e Enterprise 3. Essa distribuição é realizada por meio do Sun Grid Engine (SGE), da Sun [61]. Esse programa distribui de forma eficiente os processos entre as máquinas, sendo que a maneira como os processos são distribuídos pode variar segundo o algoritmo de escalonamento de *jobs*. Nesse trabalho, foram utilizadas duas formas de distribuição: *fill-up* e *single-core*. Na *fill-up*, o SGE distribui os *jobs* de forma a ocupar todos os *cores* de uma máquina para então passar para a máquina seguinte. Na distribuição *single-core*, o SGE considera que há apenas um *core* por máquina.

Na Altix, os nós utilizam o sistema Suse Linux Enterprise Server. Este *Cluster* se vale da tecnologia *Infiniband* para interconectar as CPUs. A máquina possui 36 CPUs Dual core Intel Itanium Series 9000 1,5 GHZ , cache L3 de 12 MB e 144 GB de RAM compartilhada (NUMAflex) e sua capacidade de armazenamento em disco rígido é de 3.5 TB. A distribuição dos processos entre as CPUs é realizada pelo Torque [52], na sua versão 2.3.3, que distribui os processos em nós consecutivos das máquinas livres.

A biblioteca do padrão MPI adotada nas Enterprises para a troca de mensagens entre os nós de processamento foi a LAM, na sua versão 7.1.3. Na implementação dessa biblioteca para *Ethernet*, a comunicação MPI é realizada por meio do protocolo TCP, que é orientado à conexão e reside no sistema operacional. Significa que quando um *job* solicita uma comunicação para a biblioteca, o pedido é repassado ao sistema operacional. Portanto, sempre que for necessário enviar e receber pacotes de rede, o sistema operacional será chamado, concorrendo com os processos da aplicação

[19]. Na Altix a biblioteca do padrão MPI adotada é o openMPI versão 1.2.8. A arquitetura *Infiniband* favorece a comunicação ponto-a-ponto, fornecendo um serviço não-orientado a conexão e externo ao sistema operacional. Logo, as comunicações são desviadas do sistema operacional e tratadas pelo acesso direto à memória. Os *switches* e placas de redes dispõem de processador e memória especializados, que se encarregam do processamento de pacotes de rede. A tecnologia *Infiniband* é mais eficiente que a *Ethernet*, no entanto, seu custo é muito mais elevado.

Para análise dos aspectos de eficiência no que se refere aos algoritmos paralelos, pode-se usar diferentes métricas. A análise do desempenho pode ser auxiliada por ferramentas computacionais que facilitam a coleta e visualização dos dados, possibilitando ao usuário identificar pontos de estrangulamento e fontes de ineficiência do algoritmo [18].

5.2 Métricas

A avaliação dos algoritmos paralelos foi realizada por meio de alguns parâmetros que mensuram o desempenho desses algoritmos. Foram utilizadas as seguintes métricas: tempo de processamento, *speedup* e eficiência.

O tempo de processamento considerado refere-se ao tempo decorrido desde o início da execução até o fim. O cálculo desse tempo foi realizado com o auxílio da biblioteca `time.h`. Para medir a execução do programa, a função `time` inicia uma variável no formato `time_t` no começo do algoritmo e uma segunda variável do mesmo formato no final. Depois é efetuada a diferença desses tempos através da função `difftime`.

Teoricamente, um problema pode ser dividido em n subproblemas, que poderiam ser resolvidos por n processadores de forma concorrente. Assim, no caso ideal, o ganho de desempenho seria linear pois o tempo paralelo seria o tempo serial dividido pelo número de processadores utilizados em paralelo. O *speedup* refere-se ao ganho de desempenho obtido com essa paralelização [19]. Ele seria entendido como a quantidade de vezes que o processo paralelo fica mais rápido em relação à velocidade de execução serial. O cálculo do *speedup* (S_i) é feito com base na razão

entre o tempo de execução do algoritmo sequencial (T_1) pelo tempo da sua forma paralela (T_n).

$$S_i = \frac{T_1}{T_n} \quad (5.1)$$

A partir do *speedup* é possível derivar a eficiência (E_i), que está relacionada à qualidade da implementação paralela ao tamanho de problema. E_i é definida como a razão entre o *speedup* (S_i) e o número de processadores utilizados (n).

$$E_i = \frac{S_i}{n} \quad (5.2)$$

Algumas ferramentas podem ser utilizadas para auxiliar a análise de desempenho de aplicações paralelas que utilizam o padrão MPI. Com o auxílio dessas ferramentas é possível identificar o andamento da aplicação através do levantamento de perfis (*profiling*) e o registro de execução (*trace*). O *profiling* de forma geral mostra a contribuição individual de cada processador para a formação do tempo de execução, como o número de vezes que ocorre determinada ação e quanto tempo foi gasto na ação. O *trace* identifica os eventos que caracterizam a execução possibilitando, por exemplo, identificar pontos de estrangulamento provenientes da comunicação entre os processadores [18]. Dentre as ferramentas para levantamento de perfil e registro de execução é possível citar Vampir [43], TotalView [65], XMPI [34] e TAU [33]. Nesse trabalho foi escolhida a ferramenta TAU para análise de desempenho da aplicação desenvolvida por ser uma ferramenta de domínio público de boa qualidade [58]. O TAU é formado por um conjunto de ferramentas que monitora as funções da aplicação, inclusive as funções específicas do MPI. A partir desse monitoramento ele gera dados de desempenho do tipo *profiling* e *trace*. Na visualização dos dados foi utilizada a ferramenta *paraProf*, disponível com o TAU.

Capítulo 6

Experimentos Numéricos

A simulação paralela da formulação semi-discreta com esquemas de pós-processamento adotada nesse trabalho foi realizada nos três ambientes descritos no Capítulo 5 visando obter dois grupos diferentes de resultados. O primeiro grupo é formado pelos exemplos de validação que buscam demonstrar a qualidade dos resultados obtidos comparando-os com os resultados disponíveis na literatura. Uma vez demonstrada a corretude da formulação, o segundo grupo vai abordar a eficiência do paralelismo na formulação. Para tal, os testes são realizados em duas etapas. Na primeira, é feito um estudo nos *Clusters Enterprise*, do comportamento do código com as duas formas de escalonamento, *single-core* e *fill-up*, onde é realizada uma análise da interferência da rede de interconexão no desempenho da aplicação e o impacto da competição por recursos internos no ambiente *multi-core*. Na segunda etapa, deseja-se comparar o desempenho das duas estratégias de armazenamento, elemento-por-elemento e aresta-por-aresta.

Os testes foram realizados em dois grupos de problemas. O primeiro grupo é a simulação de injeção de traçadores, útil para obter informações que caracterizam o meio, como barreiras ao escoamento, direção do escoamento, dispersividade, etc. O outro grupo está relacionado à recuperação de óleo, sendo caracterizado pela injeção de um fluido buscando deslocar o fluido. Esse deslocamento pode ocorrer com razão de mobilidade unitária ($M=1$), logo os dois fluidos têm viscosidades iguais, o que caracteriza um escoamento estável em relação às oscilações espúrias e aos efeitos de

orientação da malha. No entanto, na prática, a maioria dos casos de recuperação de óleo envolvem deslocamento com razão de mobilidade adversa ($M > 1$), e são abordados na seção 6.2. Em todos os exemplos tratados nesse capítulo, foram utilizados cinco vetores na base ortogonal do espaço de Krylov, GMRES(5), tolerância igual a 10^{-6} para os métodos iterativos e 10^{-3} para o algoritmo multicorretor.

6.1 Injeção de Traçadores

Traçador é um componente usado para destacar um fluido visando marcar seu transporte no tempo e no espaço. A injeção de traçadores ocorre por um período de tempo determinado e, posteriormente, há a sua interrupção. Seu objetivo é avaliar o traçado de escoamento, ou seja, a distribuição de concentração do traçador a partir do ponto de injeção por um período de tempo [14].

No contexto da engenharia de reservatórios de petróleo, a injeção de traçadores é uma técnica robusta e barata utilizada para monitorar o movimento do fluido a fim de adquirir informações sobre as propriedades do reservatório, uma vez que o tempo de chegada do traçador aos poços produtores e sua concentração são informações úteis para se estimar barreiras ao escoamento, comunicação entre reservatórios, condutividade hidráulica, dispersividade e as tendências na direção do escoamento [3].

O transporte conservativo de poluentes em reservatórios de água subterrânea também pode ser modelado como um problema de traçador. Nesse caso, pode ser utilizado o mesmo modelo empregado na simulação de reservatórios de petróleo, onde a distribuição do fluxo no meio é governado pela equação da pressão associada à Lei de Darcy e a distribuição do contaminante no decorrer do tempo é dada pela equação da concentração que descreve o transporte. Os hidrologistas podem usar o modelo na análise do escoamento das águas, na qual envolve simulações de diferentes situações, visando estimar e prevenir os efeitos da movimentação de um contaminante miscível em meios porosos [15].

Na simulação do problema do traçador foi utilizado o esquema de injeção conhecido como *five-spot* (cinco-poços). Nesse esquema, os poços injetores podem ser distribuí-

dos uniformemente no reservatório e os produtores estão dispostos equidistantes destes, conforme pode ser observado na Fig. 6.1.

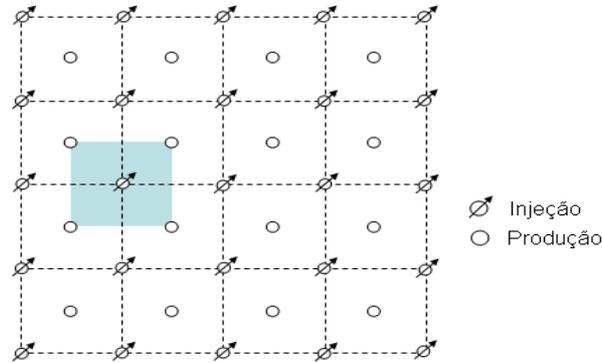


Figura 6.1: Malha *five-spot*.

Considere um reservatório de petróleo hipotético constituído por um poço produtor localizado no centro do reservatório, e quatro poços injetores, dispostos nos vértices do reservatório, como apresentado na Fig. 6.2.

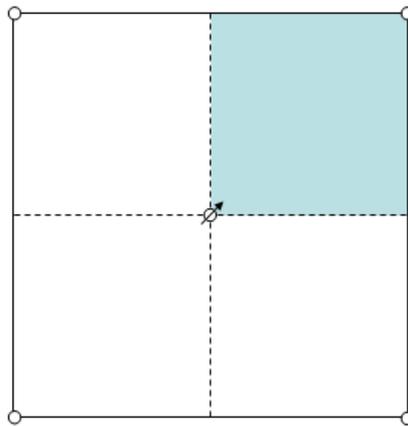


Figura 6.2: Modelo *five-spot*.

Na Fig. 6.2 observa-se que o reservatório pode ser dividido em quatro setores iguais em relação a forma e ao comportamento no que se refere a distribuição de pressões, de escoamento de fluxos, etc [54]. Assim sendo, a simulação pode ser realizada em apenas um setor, e o que ocorrer nesse setor será observado no restante do reservatório. Portanto, o estudo será considerado em um quarto do domínio.

O setor do reservatório a ser utilizado na simulação é representado por um malha estruturada com 80×80 células, onde cada célula está dividida em dois triângulos. Os lados do setor medem $L = 1000,0ft$. O poço injetor está localizado no canto inferior esquerdo ($x = y = 0$) e o poço produtor no canto superior direito ($x = y = 1000$). O meio poroso é homogêneo com permeabilidade $\Lambda_{11} = \Lambda_{22} = 100mD$, porosidade $\phi = 0,1$, difusão molecular $\alpha_m = 0,0$ e a viscosidade do fluido residente é $\mu(1) = 1,0cP$.

No problema de injeção de traçador, um volume de traçador é injetado no poço injetor que é movimentado pelo escoamento de água sendo retirado no poço produtor. No teste considerado, o volume do traçador injetado é de $0,25\%$ do volume poroso ($250ft^2$) o que equivale a uma injeção realizada por um período de 5 dias a uma taxa de $200ft^2/dia$. A viscosidade do traçador é a mesma do fluido residente. Nesse caso, a razão de mobilidade é unitária (Eq. 2.49), portanto o problema é linear e o escoamento é livre de qualquer instabilidade, física ou numérica. Além disso, a orientação da malha não influencia o resultado.

O comportamento do traçador foi observado em 800 e 1200 dias com dois níveis de dispersão $\alpha_l = 10,0$, $\alpha_t = 1,0$ (Fig. 6.3) e $\alpha_l = 1,0$, $\alpha_t = 0,0$ (Fig. 6.4), em ambas as figuras, o eixo x apresenta o tempo de observação em dias e o eixo y o volume de concentração. A execução desse teste considerou apenas 1 processador. Os resultados alcançados condizem com o esperado sendo similares àqueles obtidos por outros autores [11, 15, 37, 38, 44].

Traçando o histórico da concentração no poço produtor, para os dois níveis de dispersão considerados, é possível observar a dependência da solução em relação aos coeficientes de dispersão. Quanto maior é o nível de dispersão mais difusiva é a solução, conseqüentemente menor é o pico máximo da concentração. Porém, o comportamento das curvas é parecido após o fluido atingir o poço produtor. Portanto, o valor máximo da concentração está relacionado ao nível de dispersão e a produção do poço depende das variáveis do reservatório. Esses resultados podem ser observados na Fig. 6.5, o eixo x apresenta o tempo de observação em dias e o eixo y o volume de concentração retirado no poço produtor. e estão condizentes com os

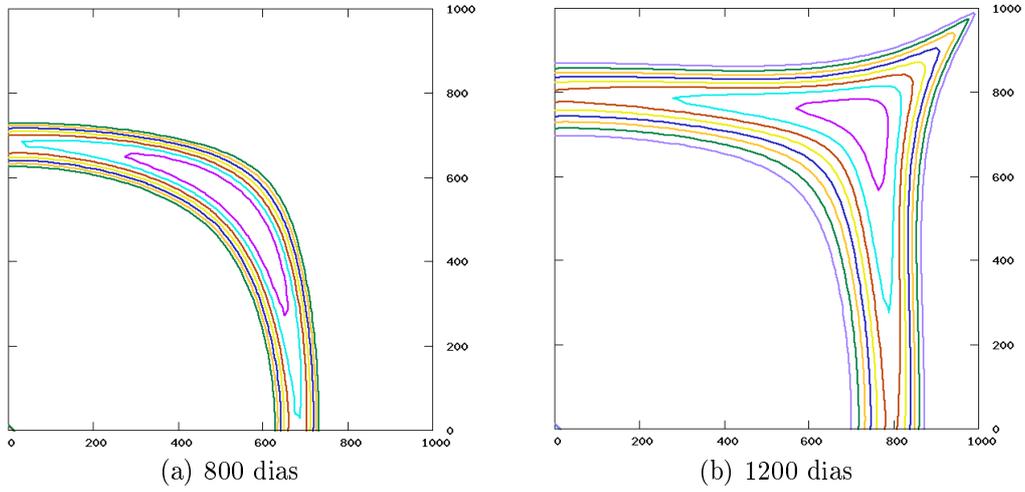


Figura 6.3: Curvas de isoconcentração para os níveis de dispersão, $\alpha_l = 10,0$ e $\alpha_t = 1,0$ e $M=1$ - Problema do traçador.

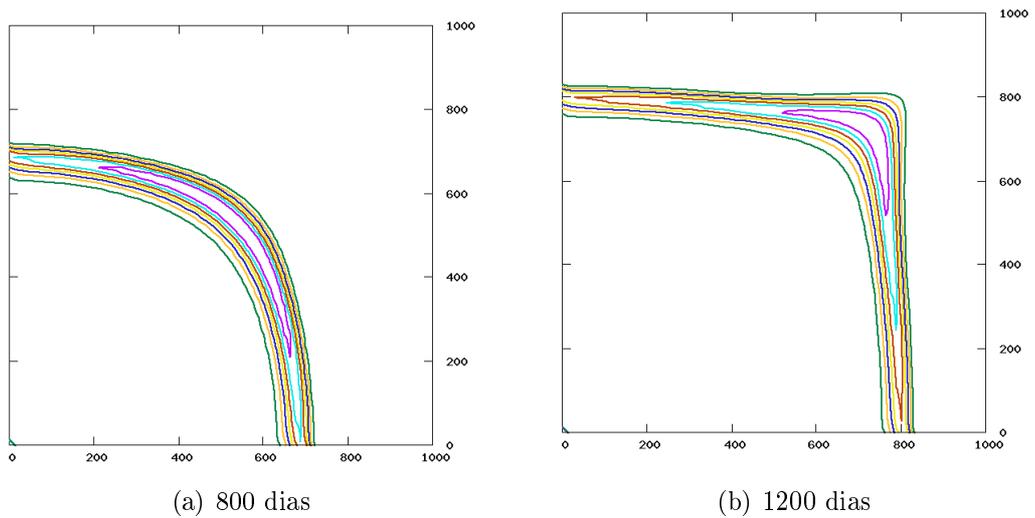


Figura 6.4: Curvas de isoconcentração para os níveis de dispersão, $\alpha_l = 1,0$, $\alpha_t = 0,0$ e $M=1$ - Problema do traçador.

apresentados em [15, 38, 44].

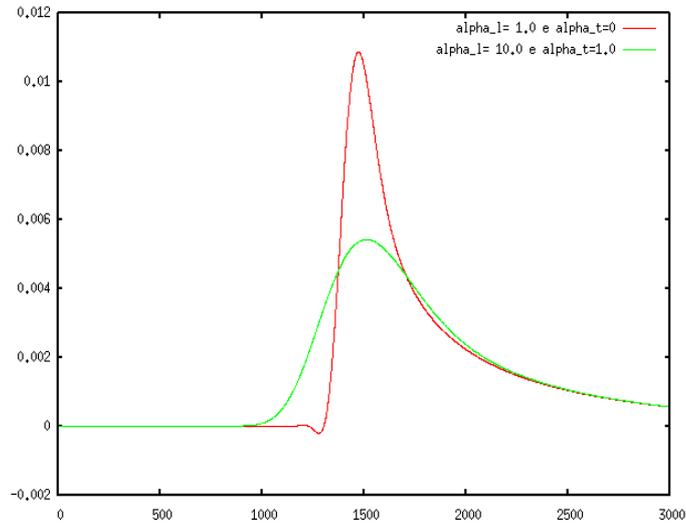


Figura 6.5: Comparação do histórico da concentração no poço produtor observado em 3000 dias considerando dois níveis de dispersão - Problema do traçador.

O histórico da concentração no poço produtor para o nível de dispersão, $\alpha_l = 1,0$, $\alpha_t = 0,0$, foi comparado com uma solução semi-analítica obtida por Abbaszadeh-Dehghani [36], e pode ser observado na Fig. 6.6. Esses resultados também se assemelham aos obtidos por [15, 44].

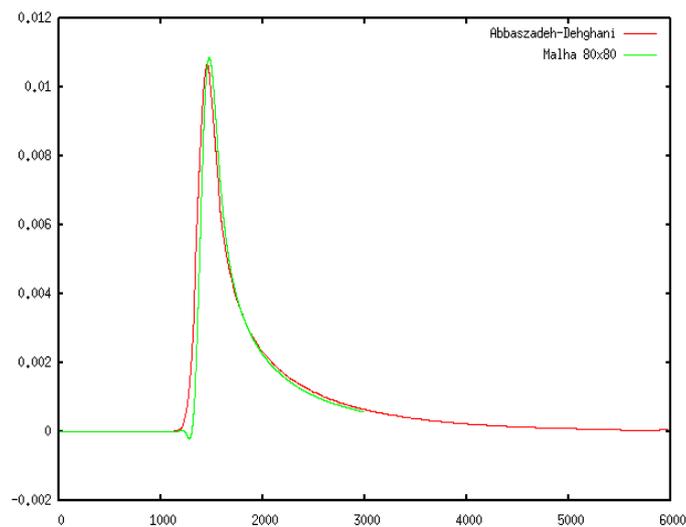


Figura 6.6: Comparação do histórico da concentração no poço produtor observado em 3000 dias com uma solução semi-analítica - Problema do traçador.

6.2 Injeção Contínua

Quando um reservatório é descoberto, os fluidos contidos na rocha possuem uma certa quantidade de energia, chamada de energia primária. Ao se iniciar a exploração, ocorre diferenças de pressão no reservatório, ocasionando o deslocamento do fluido residente rumo aos poços de produção em razão dessa energia. A medida que o óleo é extraído, a energia primária é naturalmente dissipada o que ocasiona uma queda de pressão no interior do reservatório, e conseqüentemente, a redução de produção do óleo. O volume de óleo retirado apenas em razão da diferença de pressão decorrente da energia primária do reservatório é denominado de recuperação primária, que é pouco eficiente, pois retém grandes volumes de óleo no reservatório [17, 54].

Para extrair o óleo retido no reservatório, pode-se compensar a redução de energia primária, introduzindo-se artificialmente energia secundária. Isso é feito através da injeção de certos fluidos no reservatório, cuja finalidade é deslocar o fluido residente ocupando o espaço deixado por ele. Ao volume adicional de óleo retirado pelo acréscimo de energia secundária denomina-se recuperação secundária convencional. Na recuperação secundária, o deslocamento é imiscível, ou seja, o fluido injetado e o residente não se misturam. Devido a presença de dois fluidos no reservatório, na interface entre eles pode surgir altas tensões, reduzindo a capacidade do fluido injetado de expelir o óleo. Além disso, se o fluido injetado possui viscosidade menor que o óleo, o fluido injetado vai se movimentar facilmente pelos canais porosos adentrando no óleo rumo ao poço produtor, formando os dedilamentos viscosos, assim não vai se propagar adequadamente no reservatório e comprometerá a produção [66].

Visando melhorar a taxa de recuperação dos reservatórios, contornando esses problemas, surgem os métodos especiais de recuperação secundária que podem ser térmicos, químicos ou miscíveis. Os térmicos, de forma geral, visam reduzir a viscosidade do óleo pelo aumento de temperatura na injeção de fluidos quentes. Os químicos atuam no mesmo sentido através de compostos químicos que podem aumentar a viscosidade da água e/ou reduzir a viscosidade do óleo. Os miscíveis eliminam as tensões interfaciais, pois o fluido injetado possui a capacidade de se misturar ao

fluido residente, ou seja, de formar uma mistura homogênea composta por uma única fase. Dessa forma se espera que o óleo seja totalmente deslocado. Normalmente, gases como etano, butano, propano, nitrogênio e dióxido de carbono podem ser utilizados no deslocamento miscível [54, 66].

Os custos operacionais e dos fluidos a serem injetados são altos nos métodos especiais de recuperação secundária. Por isso, várias análises devem ser realizadas antes da recuperação ser iniciada com o intuito de se constatar a viabilidade técnica e econômica da operação. Uma ferramenta muito útil que o engenheiro de petróleo dispõe para essas análises são as simulações numéricas de reservatório, usadas para realizar ensaios buscando estimar as características do reservatório e também prever o comportamento do fluido no reservatório [54].

Na simulação do problema de deslocamento miscível com mobilidade unitária ($M=1$) foi utilizado o mesmo reservatório hipotético definido para o problema do traçador na Fig. 6.2. Significa que o domínio é um quarto do reservatório, discretizado por um malha estruturada de 80×80 células, divididas em dois triângulos. O lado do domínio mede $L = 1000,0ft$, os poços injetor e produtor estão localizados, respectivamente, no canto inferior esquerdo e no canto superior direito e em todo o meio, a permeabilidade é $\Lambda_{11} = \Lambda_{22} = 100mD$, a porosidade é $\phi = 10^{-1}$, a difusão molecular é $\alpha_m = 0$, a taxa de injeção é $200ft^2/dia$ e a razão de mobilidade da mistura é unitária, isto é, $M = 1$. As curvas de isoconcentração podem ser observadas em 250, 500, 750 e 1000 dias com o nível de dispersão $\alpha_l = 10$, $\alpha_t = 1$, e a execução em 1 processador. Nas imagens da Fig. 6.7, o eixo x apresenta o tempo em dias e o eixo y o volume de concentração injetado. Os resultados alcançados estão de acordo com os obtidos em [37] e podem ser observados na Fig. 6.7.

O deslocamento miscível com razão de mobilidade adversa ($M>1$) é uma importante categoria de problemas envolvendo os métodos miscíveis pertencentes a recuperação especial secundária. A maioria dos deslocamentos miscíveis ocorrem com mobilidade adversa, ou seja, com razão de mobilidade maior que 1, influenciando a fração da área do reservatório que será invadida pelo fluido injetado.

Para ilustrar, suponha um deslocamento imiscível, onde injeta-se água ou qualquer

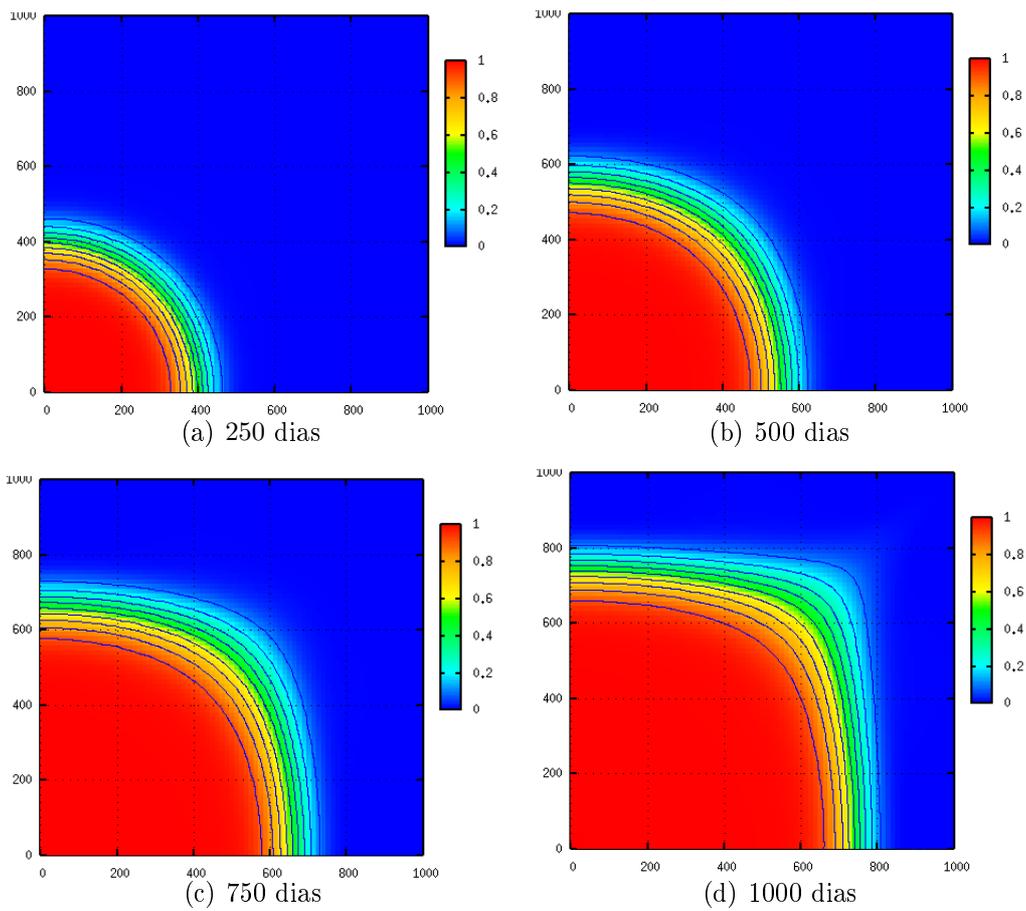


Figura 6.7: Curvas de isoconcentração para os níveis de dispersão, $\alpha_l = 1$, $\alpha_t = 0$ e $M=1$ - Injeção Contínua.

outro fluido para movimentar o óleo residente no poço. Se a razão de mobilidade é unitária, significa que o fluido injetado e o óleo do reservatório podem fluir através do meio poroso com a mesma facilidade ou mobilidade. Se, por exemplo, a razão de mobilidade é 20 então a água tem a capacidade de fluir 20 vezes mais rapidamente que o óleo. Portanto, a água injetada tende a se movimentar através do óleo em direção aos poços produtores, deixando uma área considerável do reservatório sem ser invadida [54]. No deslocamento miscível, a viscosidade do solvente influencia na razão de mobilidade da mistura, formada entre o fluido injetado e o óleo residente. Se o solvente possui viscosidade similar ao óleo, então a mobilidade da mistura será unitária ou favorável e seu deslocamento varrerá uma boa área do poço, garantindo uma recuperação satisfatória. Caso, o solvente possua viscosidade menor que o óleo então a razão de mobilidade da mistura será maior que 1, logo desfavorável, conseqüentemente, a área invadida será menor que a prevista, comprometendo a retirada de óleo do reservatório.

Na simulação desses deslocamentos, fisicamente instáveis, é frequente a ocorrência de oscilações espúrias que se deve a alguns fatores tais como, a forte não-linearidade das equações governantes, o comportamento predominantemente convectivo do problema e as diferenças de precisão entre a velocidade e as demais variáveis. Para amenizar esses problemas, foram utilizadas técnicas de estabilização como SUPG, CAU e Pós-processamento. Considerando o papel que a velocidade desempenha no problema e a importância do pós-processamento apresentado por Malta [38] foram realizados testes para ilustrar os efeitos dessa estabilização visando a validação da implementação. Quando a velocidade é calculada diretamente pela Lei de Darcy terá ordem de precisão inferior as demais grandezas, pressão e concentração, o que vai ocasionar instabilidades numéricas que não possuem representatividade física na solução da concentração. As Fig. 6.8(a) e 6.13(c) ilustram as curvas de isoconcentração obtidas com a velocidade calculada diretamente pela Lei de Darcy e as Fig. 6.13(b) e 6.13(d) apresentam as curvas de isoconcentração quando a velocidade é calculada usando a técnica de pós-processamento apresentada por Malta [38]. Nas imagens da Fig. 6.8 o eixo x apresenta o tempo em dias e o eixo y o volume de concentração injetado. os resultados obtidos consideraram $M=20$ e estão de acordo

com os observados na literatura [38].

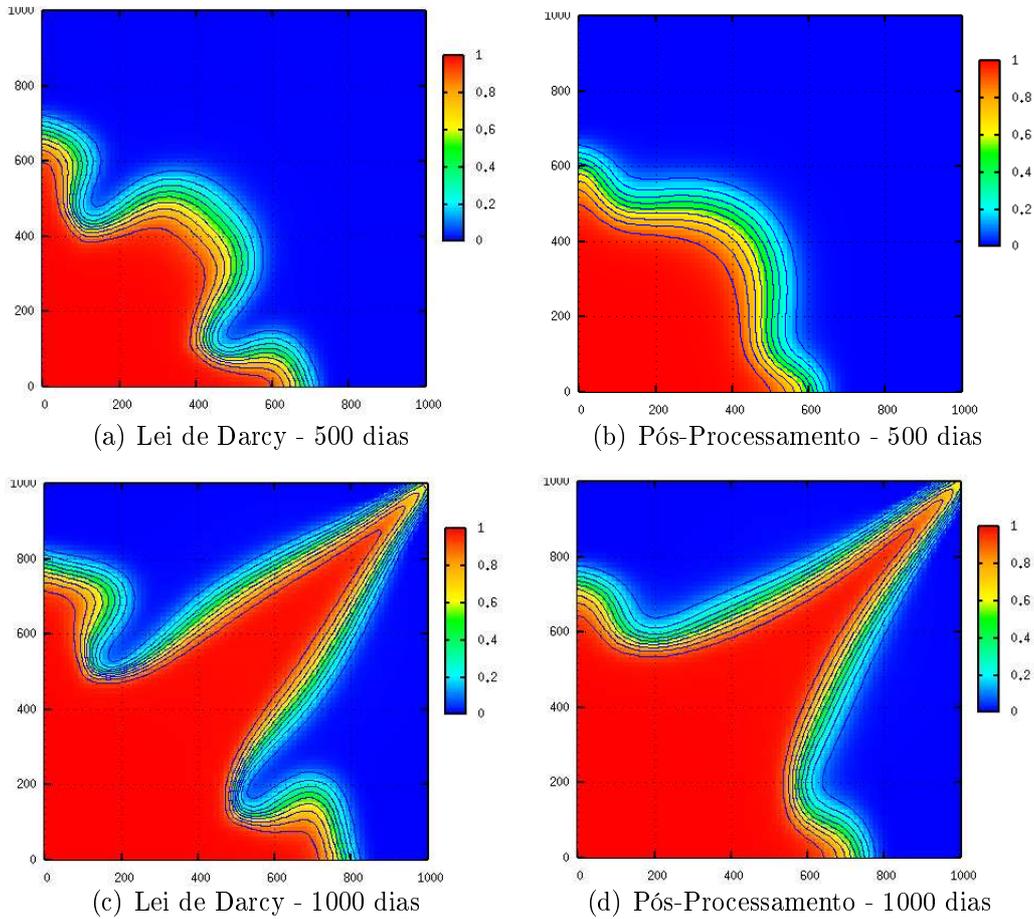


Figura 6.8: Comparação das curvas de isoconcentração considerando a velocidade de Darcy (à esquerda) e a velocidade com esquemas de pós-processamento (à direita) - Níveis de dispersão, $\alpha_l = 10$, $\alpha_t = 1$ e $M=20$ - Injeção Contínua.

Nas simulações desse deslocamento ainda é frequente a ocorrência de sensibilidades a orientação da malha e um maior esforço computacional decorrente de dificuldades de convergência dos sistemas não lineares [44]. Tais dificuldades, assim como a definição incorreta das condições de contorno, podem comprometer a solução do problema.

6.3 Condições Iniciais e de Contorno

As condições de contorno e iniciais que devem ser aplicadas ao sistema acoplado de equações diferenciais transiente e não-linear foram descritas de maneira geral na Seção 2.1.5. Agora, essas condições serão apresentadas de forma mais detalhadas, considerando um quarto do reservatório hipotético apresentado na Fig. 6.1 com as mesmas características geométricas dos exemplos anteriores.

No deslocamento miscível com mobilidade adversa, existe uma forte dependência da solução em relação a malha utilizada. Considerando malhas estruturadas, existem duas maneiras de se orientar a malha, de forma diagonal ou paralela ao percurso seguido pelo fluido injetado. Na orientação diagonal, o caminho que o fluxo segue entre o poço injetor e o poço produtor sendo diagonal as linhas da malha. Na orientação paralela, a trajetória do fluxo é paralela a essas linhas. No deslocamento com razão de mobilidade adversa, dependendo da orientação da malha, diferentes resultados podem ser obtidos, devido ao fluxo preferencial que o fluido segue nas linhas da malha [51]. A Fig. 6.9 ilustra as duas orientações da malha considerando uma discretização por elementos triangulares. Para observar os efeitos da orientação

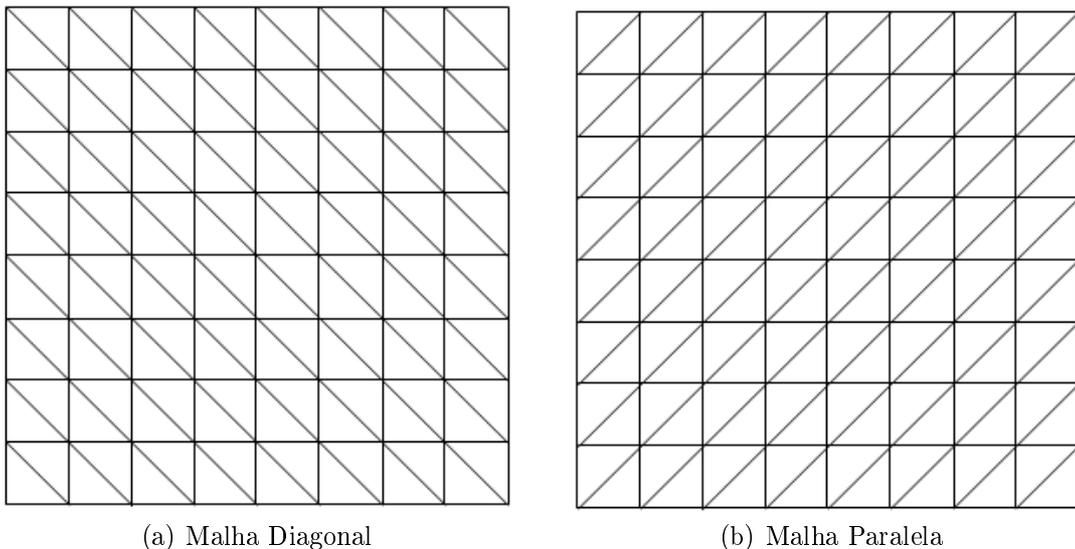


Figura 6.9: Tipos de orientação de malhas estruturadas.

da malha nos resultados das simulações foi realizado um teste com as duas configurações de malha descritas pela Fig. 6.9. Para tal, utilizou-se o mesmo reservatório

hipotético apresentado na Fig. 6.2 e os mesmos parâmetros definidos para o deslocamento miscível com mobilidade adversa ($M=20$). Para a observação do deslocamento foram escolhidos 500 e 1000 dias. As soluções obtidas com as duas orientações de malhas citadas podem ser visualizadas na Fig. 6.10 que segue o mesmo padrão da Fig. 6.8, e são semelhantes ao resultado obtido no trabalho apresentado em [51]. É possível observar que a solução obtida empregando a malha paralela, representa melhor o fenômeno físico.

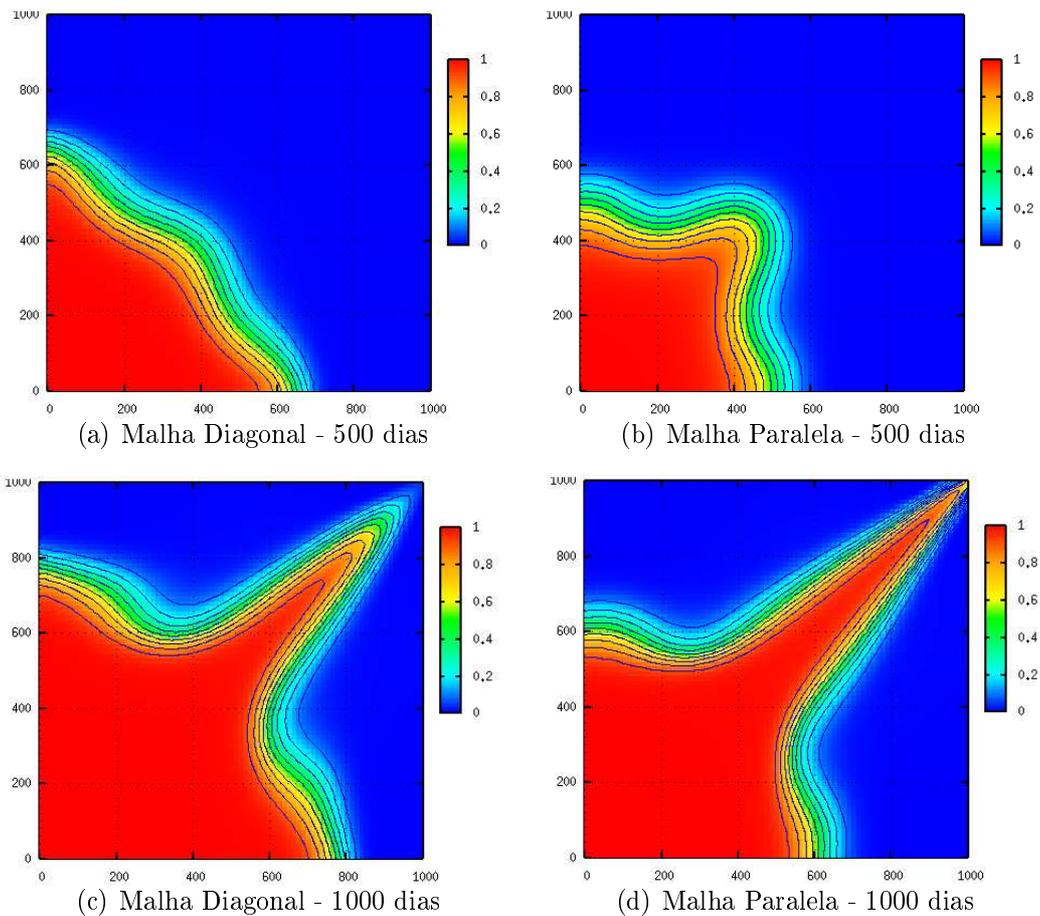


Figura 6.10: Comparação das curvas de concentração do deslocamento considerando duas orientações de malha - Níveis de dispersão, $\alpha_l = 10,0$, $\alpha_t = 1,0$ e $M=20$ - Injeção Contínua

Uma solução alternativa, adotada nesse trabalho com o intuito de amenizar os efeitos da orientação da malha, observados nas Figs. 6.10(a) e 6.10(c), foi o uso de uma condição inicial radial para a concentração [2]. Assim, delimitou-se uma região s em torno do poço injetor, de forma que os pontos contidos nessa região que estivessem

radialmente distribuídos, ou seja, sob os s^* círculos abaixo de s receberam certa fração de concentração, tornando a injeção mais suave. A Fig. 6.11 ilustra esse fato e a Equação (6.1) delimita a fração de concentração utilizada nos pontos sob as curvas s^* .

$$c = 1 - \frac{s}{r} \quad (6.1)$$

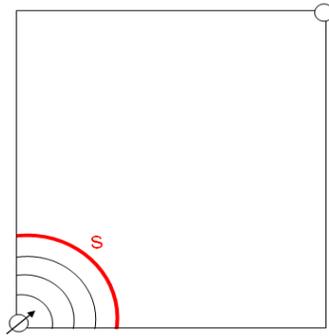


Figura 6.11: Condição inicial radial para o deslocamento miscível.

onde r é o raio da curva s . Para determinar esse raio foi utilizado um cone de altura unitária sobre o poço injetor. A Fig. 6.12 ilustra esse cone. Inicialmente foi definido uma fração pequena de volume, depois esse volume foi igualado a um quarto do volume do cone o que determinou o raio. Lembrando que foi utilizado um quarto devido a simetria do domínio. A definição dessa fração de volume foi feita tomando como referência o problema do traçador, assim foi usado 0,25% de volume o que resultou em $r \approx 31$.

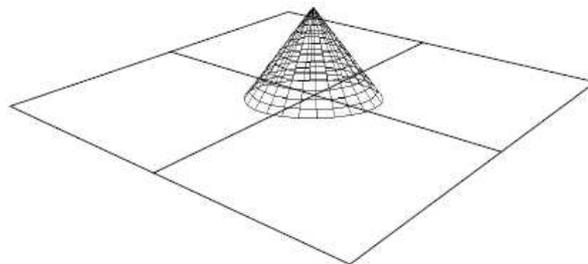


Figura 6.12: Cone de altura unitária usado na definição do raio da curva s no problema de injeção contínua.

As soluções utilizando essa condição inicial para um deslocamento de 500 e 1000

dias considerando a orientação de malha diagonal, que é o pior caso, podem ser observadas na Fig. 6.13 que segue a mesma forma da Fig. 6.8. Nota-se que, apesar de ser utilizada uma malha com orientação desfavorável, malha diagonal, as Figs 6.13(b) e 6.13(d) representam soluções satisfatórias quando comparadas com as soluções obtidas com a malha diagonal (Fig. 6.10(b) e 6.10(d)). Então, para malhas não-estruturadas a condição inicial pode vir a melhorar a solução.

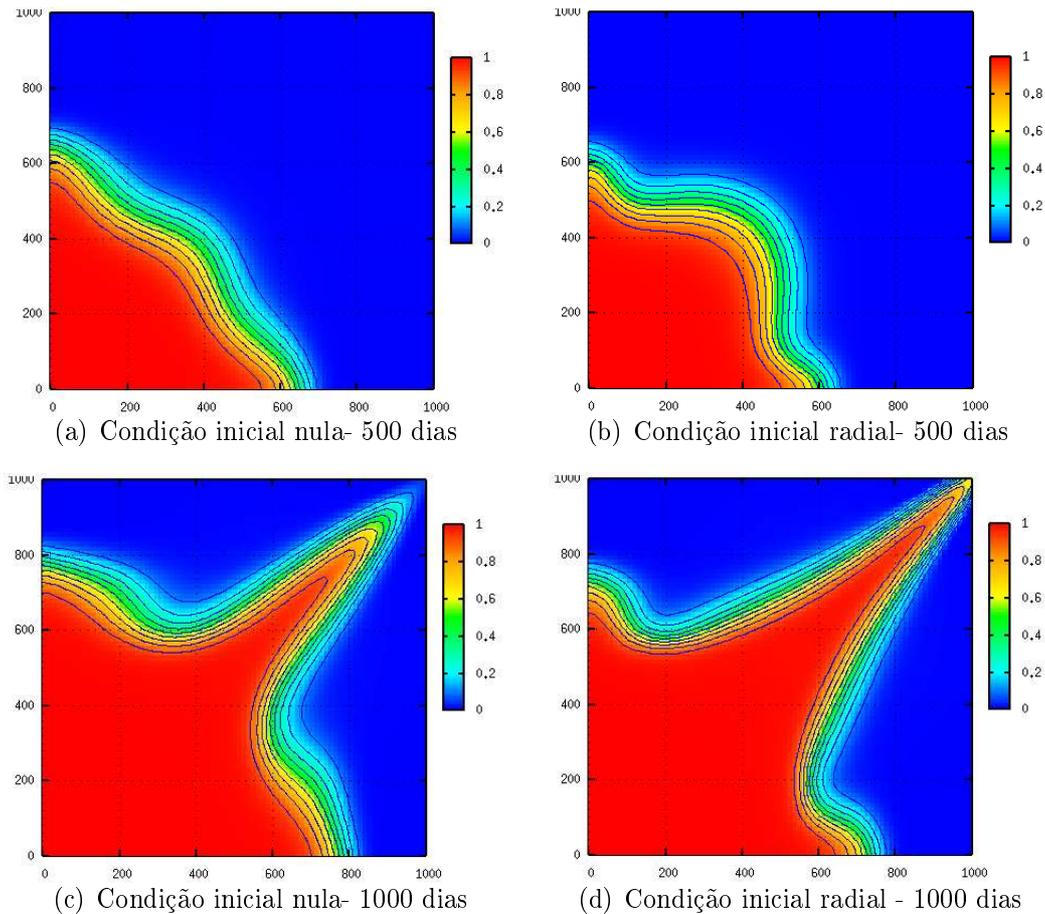


Figura 6.13: Comparação das curvas de isoconcentração da considerando uma condição inicial radial para malha com orientação paralela - Níveis de dispersão, $\alpha_t = 10$, $\alpha_t = 1$ e $M=20$ - Injeção Contínua.

Nas Fig. 6.10(b) e Fig. 6.10(d), as condições iniciais para a pressão, velocidade e concentração são nulas, pois foi usada uma orientação de malha paralela que é a mais indicada. Foi utilizada uma vazão de 200ft^3 ao reservatório, mas devido a simetria aplica-se um quarto dessa vazão, logo, $q = 50$ é aplicada no poço injetor localizado no ponto $(0, 0)$ e $q = -50$ no poço produtor posicionado em $(1000, 1000)$. A pressão

é prescrita em um dos vértices onde não existem vazão, assim, no ponto $(0,1000)$ ou no ponto $(1000,0)$ o valor da pressão é nulo. É importante ressaltar que não se deve forçar uma simetria prescrevendo o valor da pressão em $(0,1000)$ e $(1000,0)$ simultaneamente, pois, isso vai comprometer a simetria da solução para $M > 1$. É razoável prescrever em apenas um dos pontos. A Fig. 6.14 apresenta as curvas de pressão para o problema de injeção contínua para $M=1$ e $M=20$ em uma malha 80×80 células, considerando o ponto $(0,1000)$ como prescrito e uma injeção realizada por um período de 250 dias. O eixo x apresenta o tempo em dias e o eixo y o valor da pressão. Pode ser observado que a pressão é simétrica para $M=1$ (6.14(a)), mas não é simétrica para $M=20$ (6.14(b)). As condições de contorno para a velocidade dizem respeito as suas componentes v_x e v_y . Em $x = 0$ a componente v_x é nula, e em $y = 0$ a componente v_y é nula, portanto representam condições de contorno de valor prescrito para a técnica de pós-processamento. Para a concentração apenas o ponto $(0,0)$ é prescrito e equivale a 1.0.

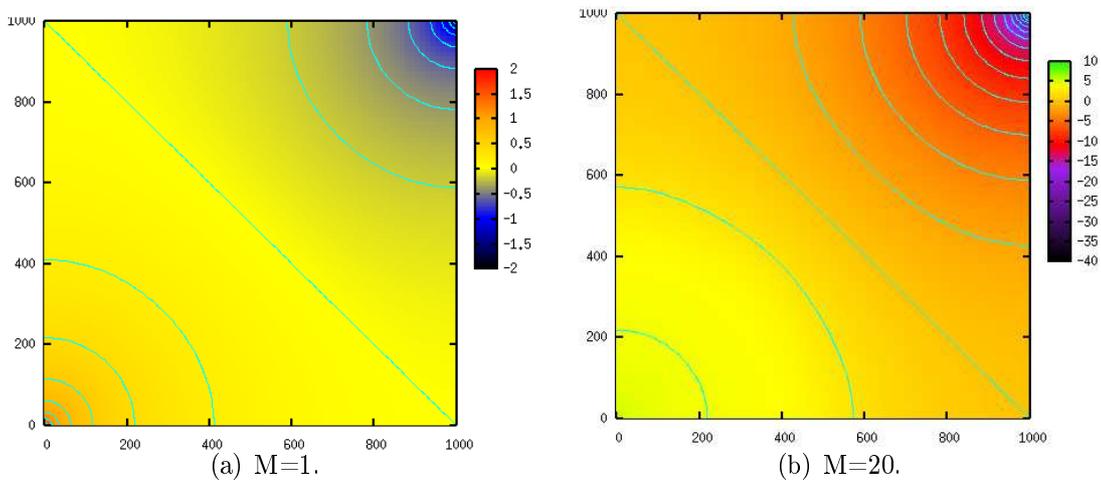


Figura 6.14: Curvas de Pressão para o problema de injeção contínua considerando a simulação para 250 dias.

As simulações de processo de recuperação de óleo são tarefas que exigem intensa atividade computacional, principalmente escoamentos com mobilidade adversa. Assim, simulações usando malhas grossas demandam tempo e a solução pode não ter a acurácia desejada. Visando obter soluções mais precisas em tempos razoáveis será utilizado o poder de processamento de vários processadores na realização dessas tarefas.

6.4 Desempenho Paralelo

Na realização dos testes em paralelo foi utilizado o problema do traçador e o deslocamento com mobilidade adversa em malhas mais refinadas. A análise do desempenho paralelo da injeção de traçador foi feita para o nível de dispersão, $\alpha_l = 1$ e $\alpha_t = 0$, observado em 800 dias. Já o problema de injeção contínua com razão de mobilidade adversa foi considerado em 500 dias. Os testes foram considerados nos três ambientes descritos no Capítulo 5 e podem ser divididos em dois grupos. O primeiro grupo se preocupou em estudar nas máquinas Enterprises, o comportamento do problema do traçador com as duas formas de escalonamento, *single-core* e *fill-up*. Foi analisado a interferência da rede de interconexão no desempenho, o impacto da competição por recursos internos no ambiente *multi-core* e o tempo de execução nas duas máquinas. Em seguida, foi verificado o desempenho da aplicação em dois ambientes que utilizam tecnologias de interconexão distintos, isto é, entre as Enterprises que empregam a tecnologia de interconexão *Ethernet* e a Altix que utiliza a tecnologia *Infiniband*. Não foi a intenção comparar as duas formas de interconexão pois os ambientes não possuem as mesmas características. O objetivo foi analisar o comportamento da implementação em uma máquina muito mais eficiente e de certa forma verificar se os ganhos alcançados com uma arquitetura melhor compensam os custos que ela demanda. Em um segundo momento, foram realizados os testes comparativos entre as duas estratégias de armazenamento aresta-por-aresta e elemento-por-elemento, considerando o problema do traçador e o problema de injeção contínua.

Foram utilizadas três malhas estruturadas, uma pequena com 512×512 células, outra média com 1024×1024 células e a última um pouco maior com 2048×2048 células. Cada célula foi dividida em dois elementos triangulares, o que resultou em 263169 nós e 524288 elementos para a malha pequena, 1050625 nós e 2097152 elementos para a malha média e 8388608 nós e 4198401 elementos para a malha maior.

6.4.1 Tipos de Escalonamento das Enterprises

Inicialmente os testes foram realizados para o problema do traçador na Enterprise 2, utilizando as três malhas e os dois algoritmos de escalonamento *fill-up* e *single-core*. As Figs. 6.15, 6.16 e 6.17 apresentam os *speedups* para as três malhas, comparando os dois tipos de escalonamento.

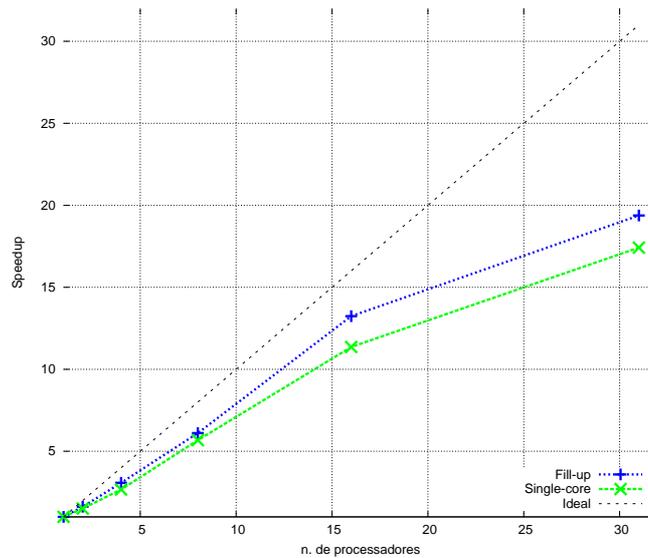


Figura 6.15: *Speedup* para os escalonamentos *fill-up* e *single-core* - Malha 512×512 - Problema do traçador - Enterprise 2.

Na Fig. 6.15 pode ser observado que o escalonamento *fill-up* produz um *speedup* maior que o escalonamento *single-core* para todos os processadores na Enterprise 2. Como a malha é pequena o tempo de comunicação externa é muito maior que o tempo de processamento, assim a comunicação acaba prejudicando o desempenho do escalonamento *single-core*.

Na Fig. 6.16 o escalonamento *single-core* tem um *speedup* melhor até 4 processadores, possivelmente a competição por recursos dentro de uma mesma máquina prejudica o desempenho do *fill-up*. Quando se utiliza mais que 8 processadores, o *fill-up* demonstra um desempenho progressivamente melhor que o *single-core*. Na Fig. 6.17 o escalonamento *single-core* apresenta um melhor desempenho até 16 processadores, como a malha é maior, o tempo de processamento prejudica o escalonamento *fill-up*, pois existe competição por recursos dentro da mesma máquina.

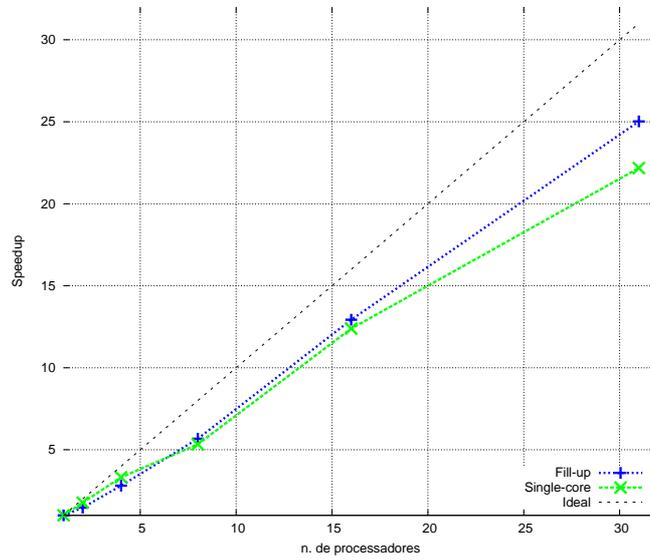


Figura 6.16: *Speedup* para os escalonamentos *fill-up* e *single-core* - Malha 1024×1024 - Problema do traçador - Enterprise 2.

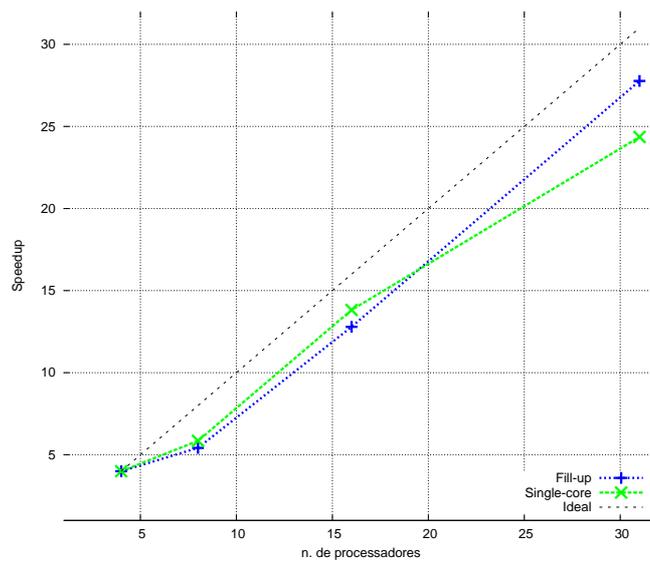


Figura 6.17: *Speedup* para os escalonamentos *fill-up* e *single-core* - Malha 2048×2048 - Problema do traçador - Enterprise 2.

Acima de 16 processadores a situação se inverte, o *fill-up* apresenta um melhor desempenho pois a distribuição do volume de computação entre mais processadores acaba reduzindo a competição por recursos e aumentando a comunicação externa.

As Figs. 6.18 e 6.19 apresentam a comparação do desempenho entre os dois tipos de escalonamentos nas malhas adotadas na Enterprise 2. Na Fig. 6.18 percebe-se que o desempenho do escalonamento *fill-up* é melhor quanto maior é o problema e há mais processadores envolvidos no processamento. A Fig. 6.19 mostra o escalonamento *single-core* e percebe-se uma melhora de desempenho quando o problema aumenta. No entanto, à medida que o número de processadores aumenta seu desempenho tende a cair mais, devido aos custos com a comunicação, e essa perda é bem maior para as malhas menores. Isso pode ser comprovado comparando os tempos de processamento relativos aos dois escalonamentos exibidos na Fig. 6.20, em que o eixo *y* apresenta o tempo em segundos e o eixo *x* o número de processadores. É possível constatar que para uma malha pequena o *fill-up* possui um tempo menor que o *single-core*. Em malhas um pouco maiores, o tempo do *fill-up* é maior que o *single-core* quando se utiliza poucos processadores. A medida que se aumenta o número de processadores, o *fill-up* tende a se igualar com o *single-core* e passa, até mesmo, a ser mais rápido.

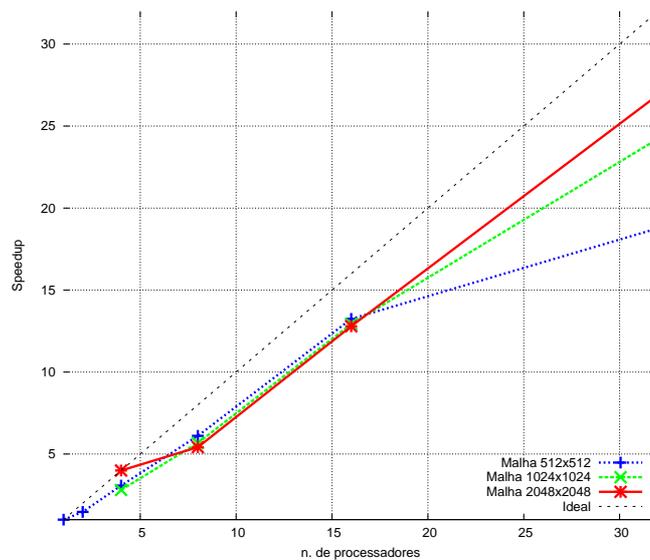


Figura 6.18: *Speedup* para o escalonamento *fill-up* em diferentes malhas - Problema do traçador - Enterprise 2.

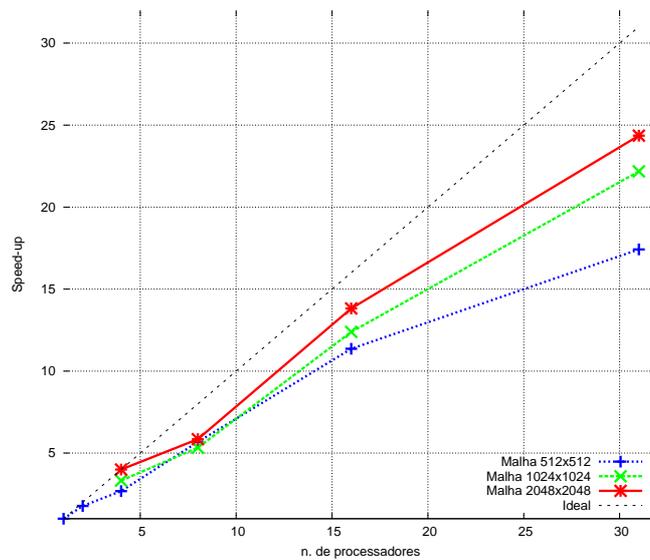
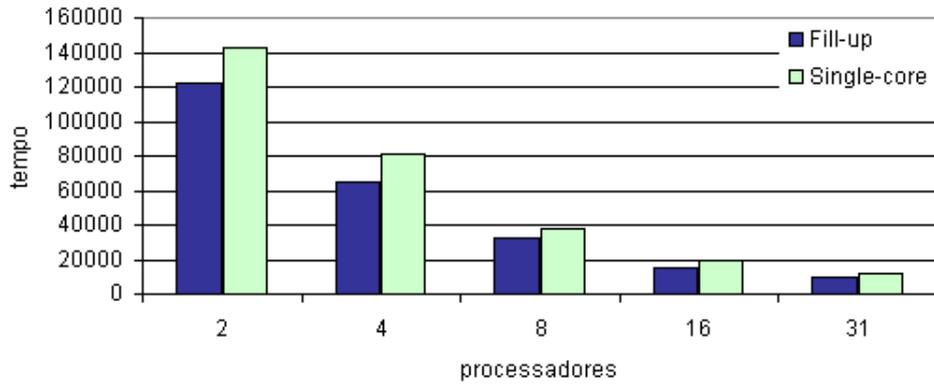
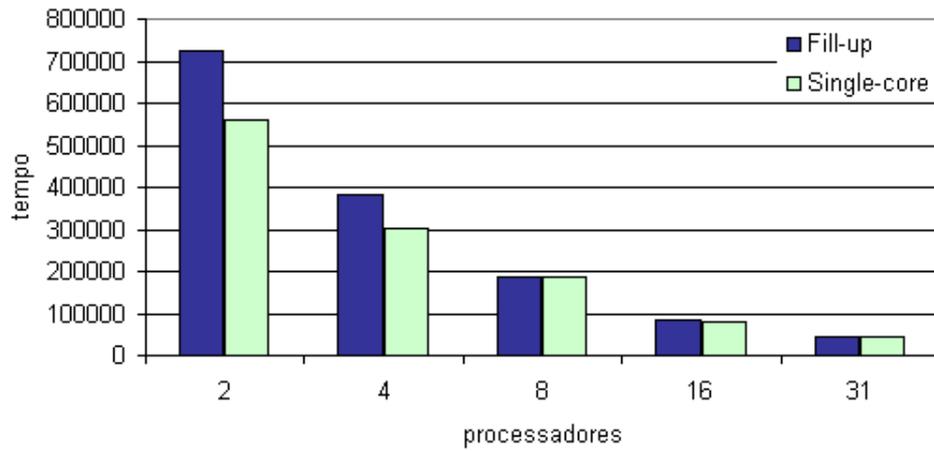


Figura 6.19: *Speedup* para o escalonamento *single-core* em diferentes malhas - Problema do traçador - Enterprise 2.

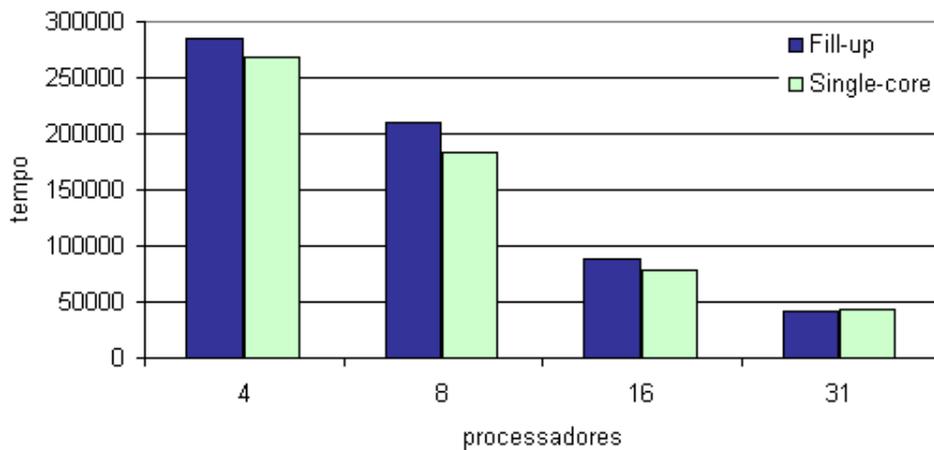
Em seguida, os mesmos testes foram realizados na Enterprise 3, sendo semelhantes àqueles obtidos na Enterprise 2. Na Fig. 6.21, que traz os resultados obtidos na malha menor, é observado que o escalonamento *fill-up* produz um *speedup* melhor que o escalonamento *single-core* até 16 processadores, mas para 29 processadores o *speedup* do escalonamento *single-core* foi maior, surpreendendo. Significa que distribuir o trabalho entre mais que 29 processadores que possuem exclusividade na hierarquia de memória foi melhor que distribuir o trabalho e produzir competição por recursos na mesma máquina, como a malha não é muito grande houve surpresa. Observando o resultado apresentado na Fig. 6.22 para uma malha maior, percebe-se que até 8 processadores o *single-core* produz um *speedup* maior. A competição por recursos dentro da mesma máquina acaba prejudicando o desempenho do escalonamento *fill-up*. Acima de 8 processadores, o *fill-up* passa a produzir um *speedup* maior, pois a comunicação acaba prejudicando o desempenho do escalonamento *single-core*. Na malha grande, Fig. 6.23, o mesmo comportamento pode ser percebido, dessa vez, o *single-core* apresenta um melhor desempenho até 16 processadores, no entanto, perde desempenho quando o trabalho é dividido entre mais de 29 processadores. Nessas malhas, o volume de processamento inicialmente é maior que o volume de comunicação externa, fazendo com que os recursos dentro



(a) 512×512



(b) 1024×1024



(c) 2048×2048

Figura 6.20: Tempo de processamento para vários processadores comparando os dois tipos de escalonamento em diversas malhas - Problema do traçador - Enterprise 2.

de um mesma máquina sejam mais utilizados que a rede de interconexão, assim, o *fill-up* fica prejudicado pois compartilha esses recursos. A medida que o trabalho é distribuído, há uma redução do volume de processamento dentro de uma mesma máquina, fazendo com que a competição diminua, e a necessidade por comunicação aumente resultando na melhora do *fill-up* em relação ao *single-core*.

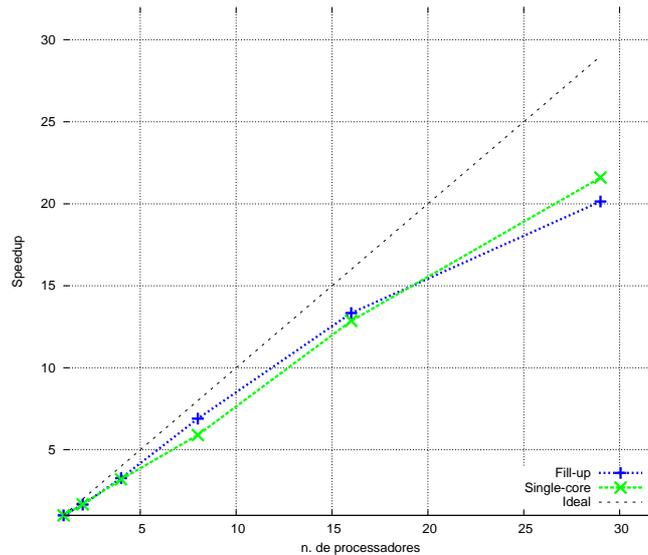


Figura 6.21: *Speedup* para os escalonamentos *fill-up* e *single-core* - 512×512 - Problema do traçador - Enterprise 3.

Nas Figs. 6.24 e 6.25 pode ser visto o desempenho dos dois tipos de escalonamentos em relação as malhas na Enterprise 3. Na Fig. 6.24 é apresentado o desempenho do escalonamento *fill-up* que se mostra bastante dependente do tamanho do problema. Quanto maior é o problema maiores são os ganhos proporcionados por esse escalonamento, no entanto, problemas maiores exigem mais memória, e é perceptível que o desempenho desse escalonamento fica prejudicado quando poucos processadores estão envolvidos, pois há muita competição pela hierarquia de memória dentro da mesma máquina. No escalonamento *single-core* apresentado na Fig. 6.25 também ocorre melhoras no desempenho quando o problema aumenta mas, esse ganho está condicionado ao volume de comunicação para a realização do trabalho.

O tempo dos escalonamentos na Enterprise 3 pode ser observado na Fig.6.26, o eixo *y* apresenta o tempo em segundos e o eixo *x* o número de processadores. Na malha

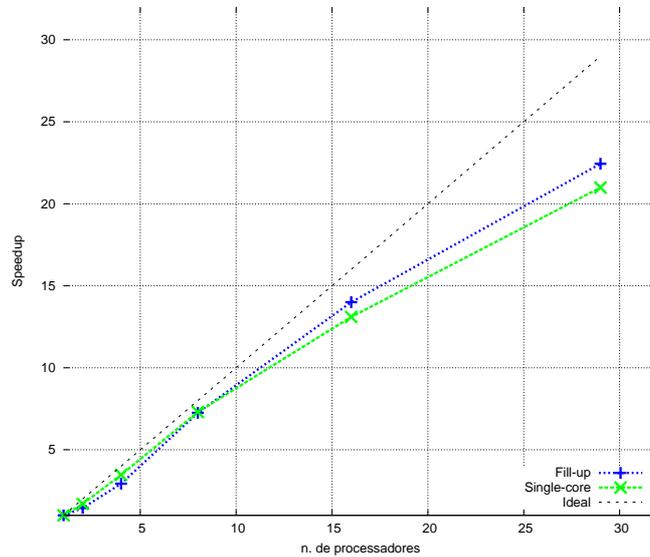


Figura 6.22: *Speedup* para os escalonamentos *fill-up* e *single-core* - 1024×1024 - Problema do traçador - Enterprise 3.

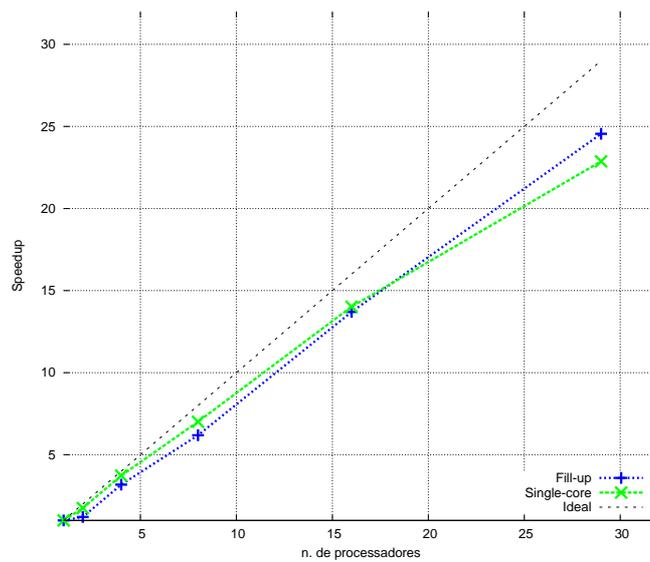


Figura 6.23: *Speedup* para os escalonamentos *fill-up* e *single-core* - 2048×2048 - Problema do traçador - Enterprise 3.

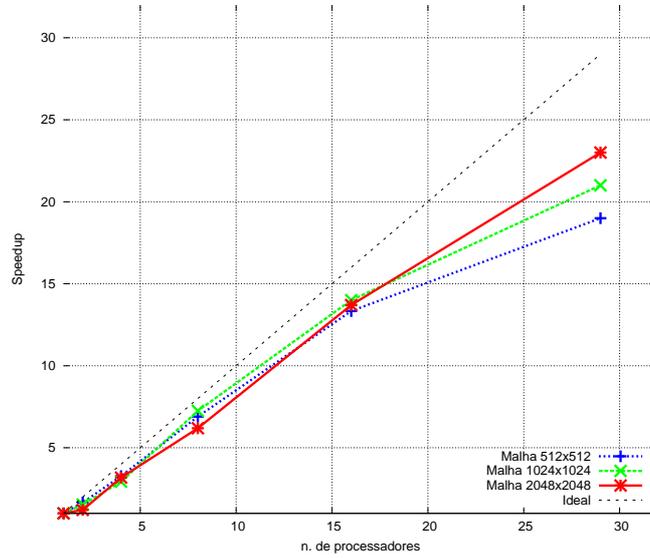


Figura 6.24: Comparação do *Speedup* para o escalonamentos *fill-up* em relação a diferentes malhas - Problema do traçador - Enterprise 3.

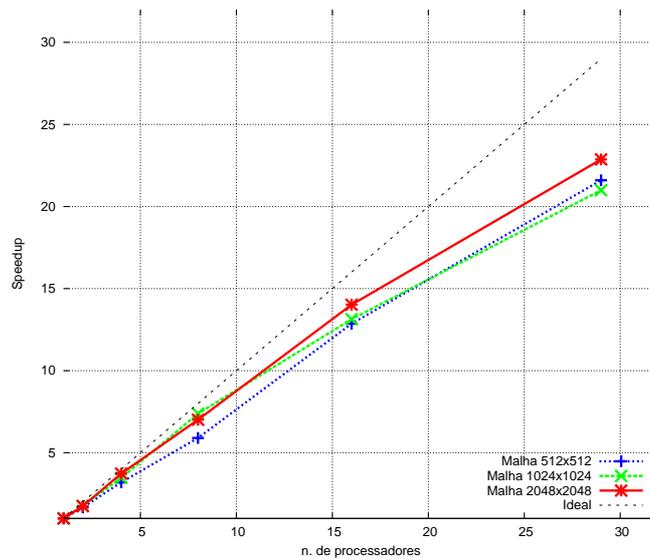
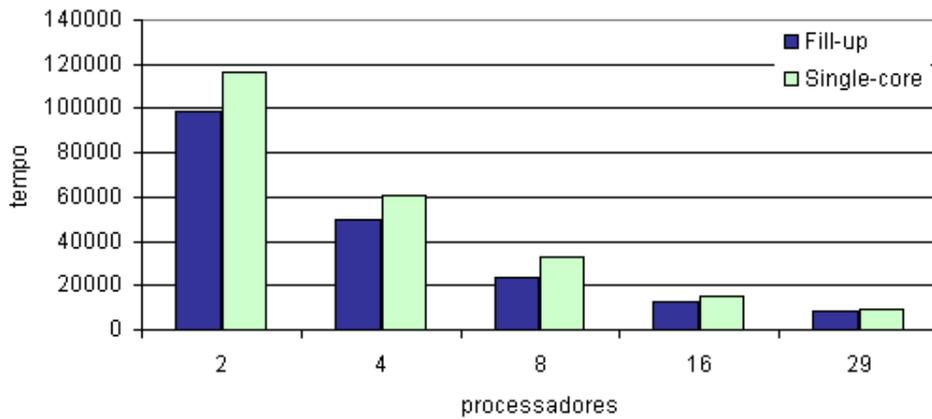


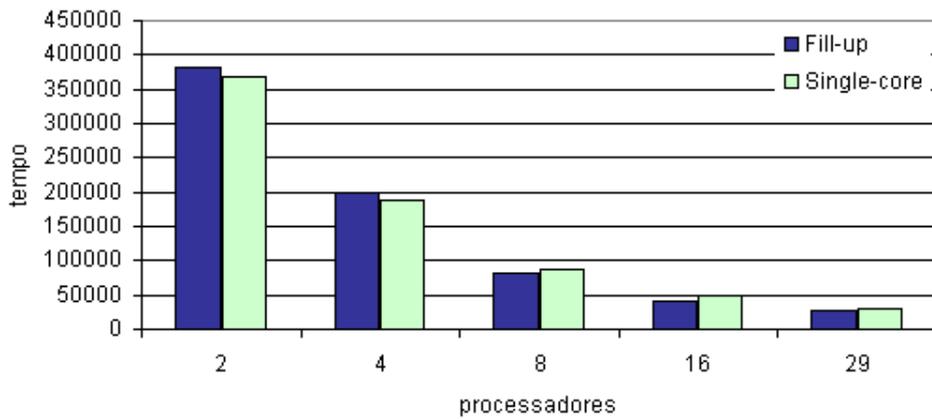
Figura 6.25: Comparação do *Speedup* para o escalonamentos *single-core* em relação a diferentes malhas - Problema do traçador - Enterprise 3.

pequena, Fig.6.26(a), o escalonamento *fill-up* apresenta tempo de processamento menor que o escalonamento *single-core* para todos os processadores. Na malha média, Fig. 6.26(b), inicialmente o escalonamento *single-core* apresenta um tempo maior que o escalonamento *fill-up*, em seguida o *fill-up* passa a gastar mais tempo e a partir de 8 processadores o escalonamento *single-core* volta a demandar mais tempo de processamento. Na malha maior, Fig. 6.26(b), até 8 processadores o escalonamento *fill-up* demanda mais tempo de processamento que o *single-core*, a partir de 16 processadores há uma inversão, o escalonamento *single-core* apresenta um maior tempo de processamento. Os resultados estão condizentes com os obtidos na Enterprise 2, ou seja, o *fill-up* é tanto melhor quanto maiores forem as malhas e maiores forem o número de processadores envolvidos.

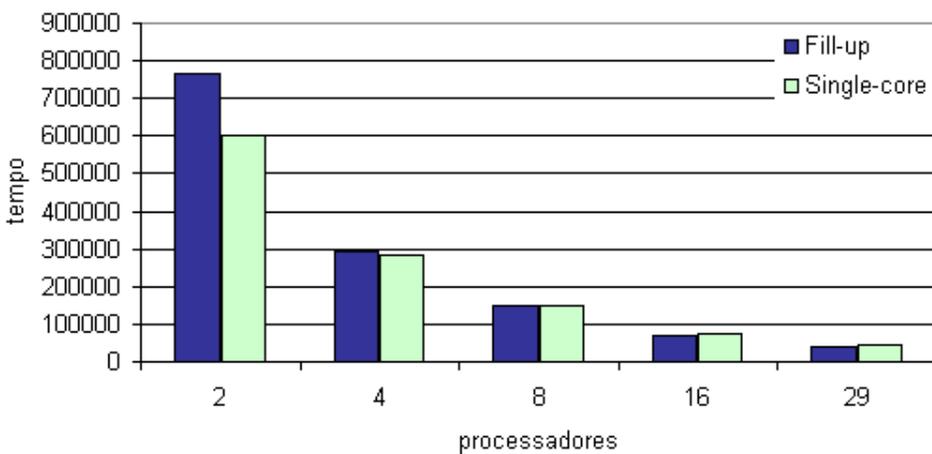
A próxima análise teve por objetivo comparar o desempenho dos escalonamentos *fill-up* e *single-core* nas duas Enterprises. As Figs. 6.27, 6.28 e 6.29 mostram o *speedup* obtido com o escalonamento *fill-up* nas duas máquinas, respectivamente, para as malhas 512×512 , 1024×1024 e 2048×2048 considerando até 62 processadores. Na malha pequena, Fig. 6.27 a Enterprise 3 produz um *speedup* melhor que a Enterprise 2, indicando que o trabalho é melhor escalonado nessa máquina. Na malha média, Fig. 6.28, o mesmo comportamento é observado, exceto para 32 processadores onde a Enterprise 2 obteve um melhor resultado que a Enterprise 3. Portanto, de forma geral, a competição por recursos na mesma máquina é mais acirrada na Enterprise 2. Na Fig. 6.29 a Enterprise 3 obtém um melhor resultado até 16 processadores, depois nota-se uma inversão, a Enterprise 2 obtem um melhor resultado para 32 processadores e ligeiramente maior para 62 processadores. Como a malha é maior, a competição por recursos dentro de cada máquina na Enterprise 3 prejudica mais o desempenho do que o volume de comunicação externa exigido na Enterprise 2 para 32 processadores. No entanto, a Enterprise 3 melhora seu *speedup* para 62 processadores, enquanto que a Enterprise 2 tem uma piora. Logo, aparentemente, aumentando-se a quantidade de processadores, a Enterprise 3 teria um melhor desempenho. É importante destacar que o tempo de processamento na Enterprise 3 foi sempre menor que o da Enterprise 2, nas malhas pequena e média, observadas respectivamente em Fig. 6.30(a) e Fig. 6.30(b), que segue o



(a) 512×512



(b) 1024×1024



(c) 2048×2048

Figura 6.26: Tempo de processamento dos dois tipos de escalonamento considerando diferentes malhas - Problema do traçador - Enterprise 3.

mesmo padrão das anteriores. Na malha maior, Fig. 6.30(c), para 4 processadores a Enterprise 3 gastou um pouco de tempo a mais que a Enterprise 2, nos demais casos, a Enterprise 3 foi mais rápida.

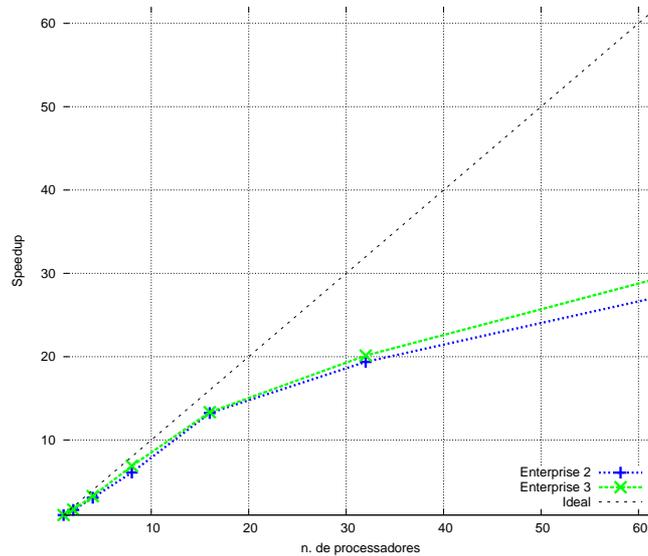


Figura 6.27: Comparação do *Speedup* para o escalonamento *fill-up* - Malha 512×512 - Problema do traçador - Enterprises 2 e 3.

A Fig. 6.31 apresenta o *speedup* nas duas máquinas para o escalonamento *single-core* considerando uma malha 512×512 . A Enterprise 3 para as três malhas apresentou um melhor *speedup* para todos os processadores conforme pode ser visto, respectivamente, nas Figs. 6.31, 6.32 e 6.33. Portanto, a Enterprise 3 escalonou melhor o trabalho que a Enterprise 2. A Fig. 6.34 apresenta os tempos de processamento nas duas Enterprises, seguindo a mesma forma das figuras anteriores. Nota-se um comportamento semelhante ao da malha anterior. Em ambas as máquinas, o ganho de tempo tende a diminuir conforme o número de processadores aumenta, o que é normal, pois quanto maior o número de processadores maior é o tempo gasto com a comunicação.

Esse problema ainda foi testado na Altix que utiliza uma tecnologia de interconexão mais eficiente que a usada pelas Enterprises. A Fig. 6.35 mostram os *speedups* obtidos pela Enterprise 2 e os da Altix considerando a malha 2048×2048 . Nota-se que o escalonamento *fill-up* apresenta um resultado relativamente próximo ao da Altix

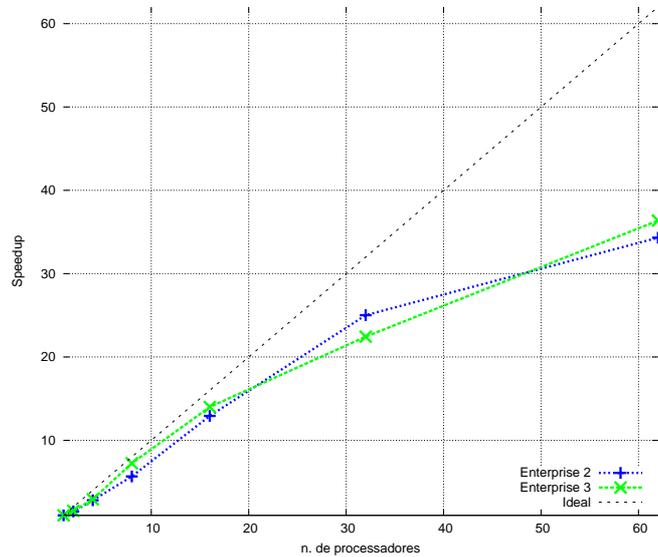


Figura 6.28: Comparação do *Speedup* para o escalonamento *fill-up* - Malha 1024×1024 - Problema do traçador - Enterprises 2 e 3.

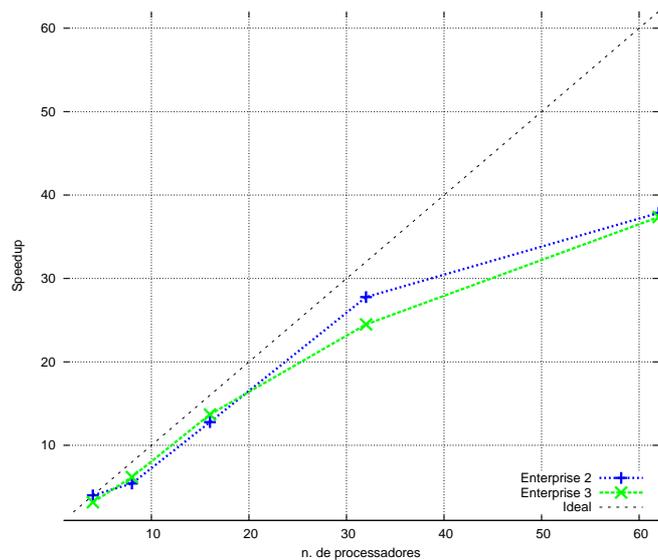
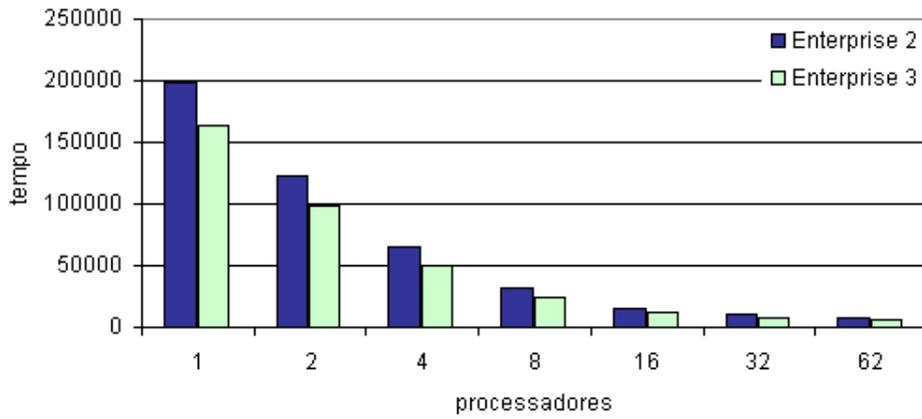
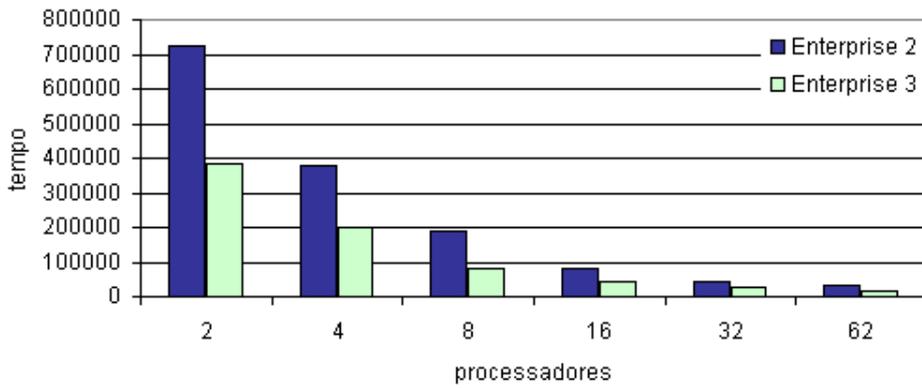


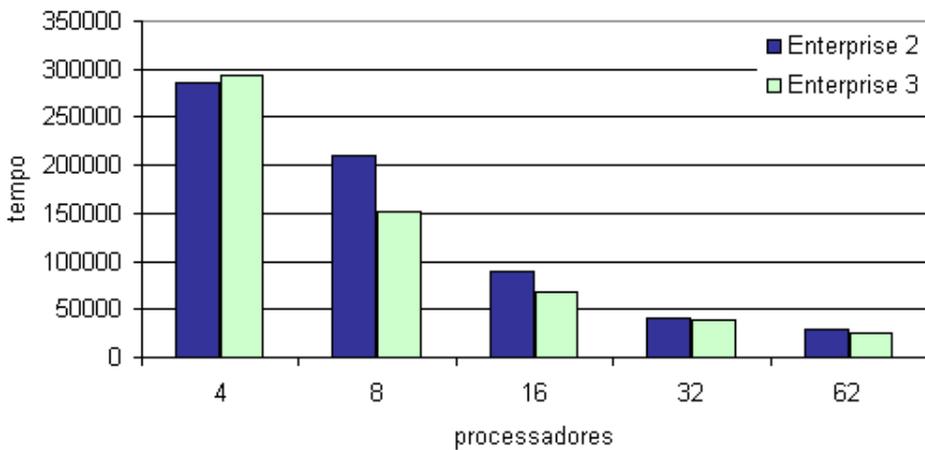
Figura 6.29: Comparação do *Speedup* para o escalonamento *fill-up* Malha - 2048×2048 - Problema do traçador - Enterprises 2 e 3.



(a) 512×512



(b) 1024×1024



(c) 2048×2048

Figura 6.30: Tempo de processamento para vários processadores comparando os escalonamento *fill-up* em diversas malhas. - Problema do traçador- Enterprises 2 e 3.

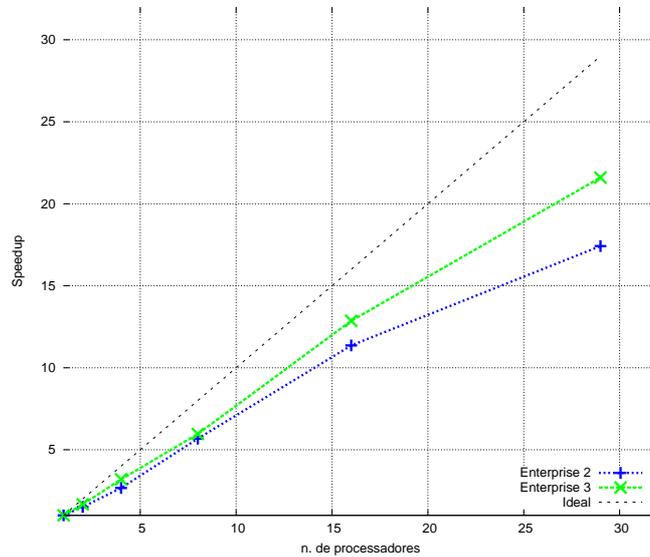


Figura 6.31: Comparação do *Speedup* para o escalonamento *single-core* - Malha 512×512 - Problema do traçador - Enterprises 2 e 3.

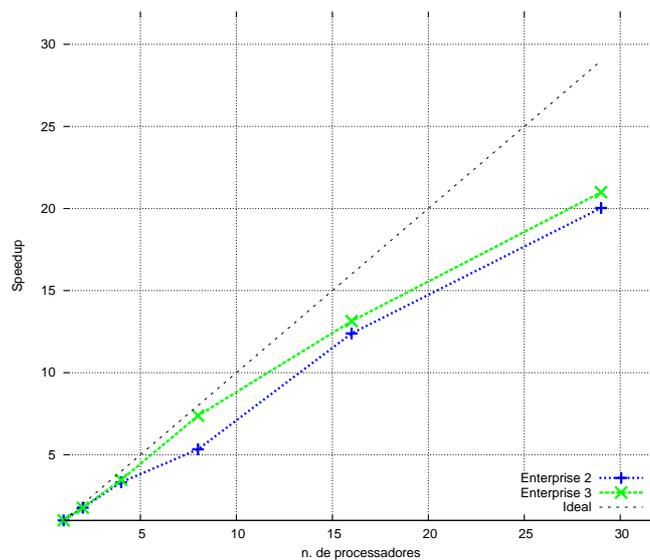


Figura 6.32: Comparação do *Speedup* para o escalonamento *single-core* - Malha 1024×1024 - Problema do traçador - Enterprises 2 e 3.

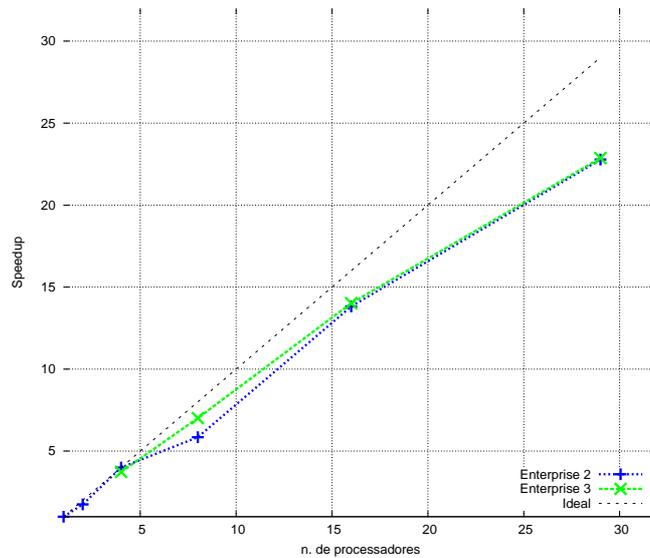
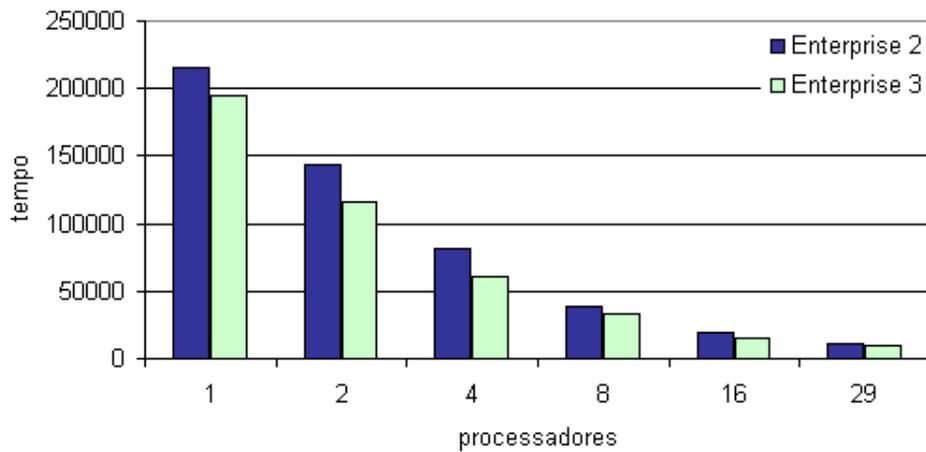


Figura 6.33: Comparação do *Speedup* para o escalonamento *single-core* - Malha 2048×2048 - Problema do traçador - Enterprises 2 e 3.

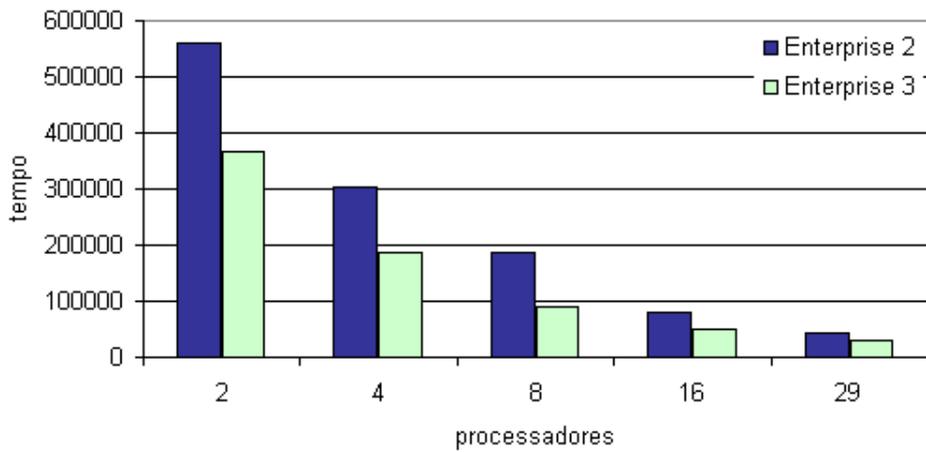
para 31 processadores. A Fig. 6.36 apresenta os resultados da Enterprise 3 e Altix considerando a mesma malha. Se observa que os resultados obtidos pela Enterprise 3 até 16 computadores, equiparados aos alcançados pela Altix. Interessante observar que a medida que se aumenta o número de processadores, o tempo gasto com comunicação acaba prejudicando mais o grau de melhora no desempenho nas Enterprises que na Altix. Isso ocorre pois, as Enterprises utilizam a tecnologia *Ethernet* menos eficiente que a tecnologia *Infiniband* da Altix. Como era de se esperar o desempenho da Altix foi superior ao das Enterprises, no entanto, vale lembrar que os custos também são superiores e na Altix também há perdas decorrentes da comunicação. A Fig. 6.37 traz os *speedups* na Altix para as três malhas, e é visível que o desempenho tende a cair à medida que se distribui o trabalho entre mais processadores. Os resultados foram animadores, de certa forma, mostrando a qualidade da técnica de paralelização utilizada no presente trabalho.

6.4.2 Comparação entre as Estratégias de Armazenamento

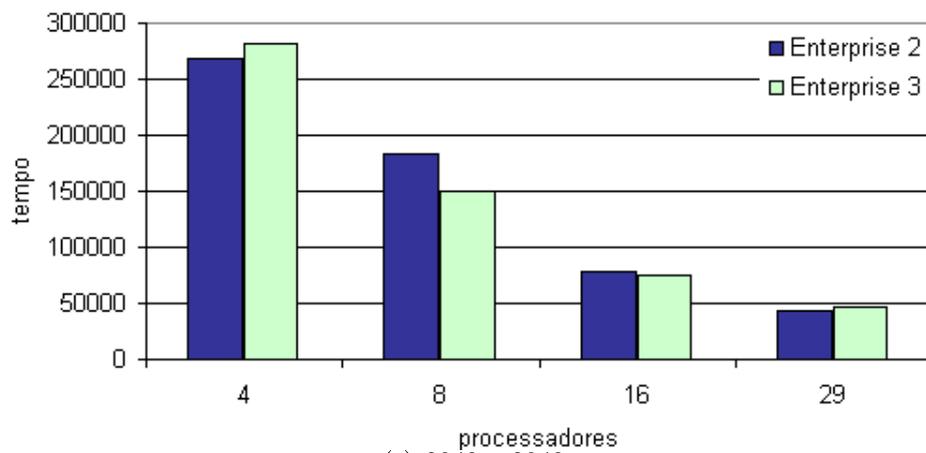
As próximas análises referem-se ao estudo comparativo entre as duas formas de armazenamento, aresta-por-aresta e elemento-por-elemento. Visando tornar o texto



(a) 512×512



(b) 1024×1024



(c) 2048×2048

Figura 6.34: Tempo de processamento para vários processadores comparando escalonamento *single-core* em diversas malhas - Problema do traçador - Enterprises 2 e 3.

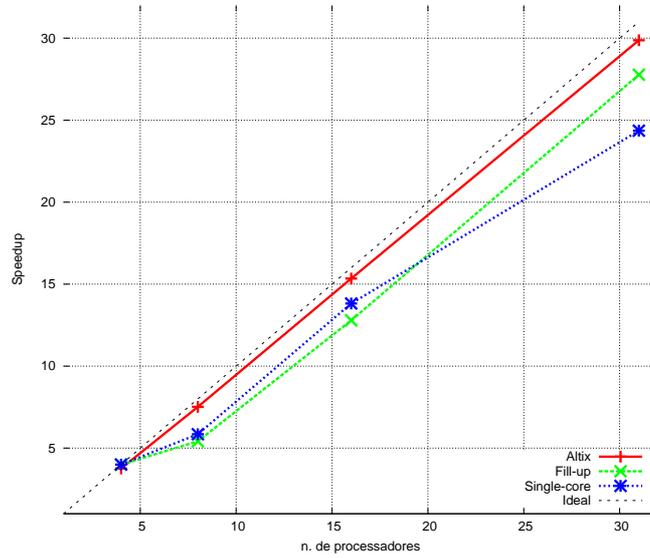


Figura 6.35: *Speedup* obtidos pela Altix e Enterprise 2 - Malha 2048×2048 - Problema do traçador.

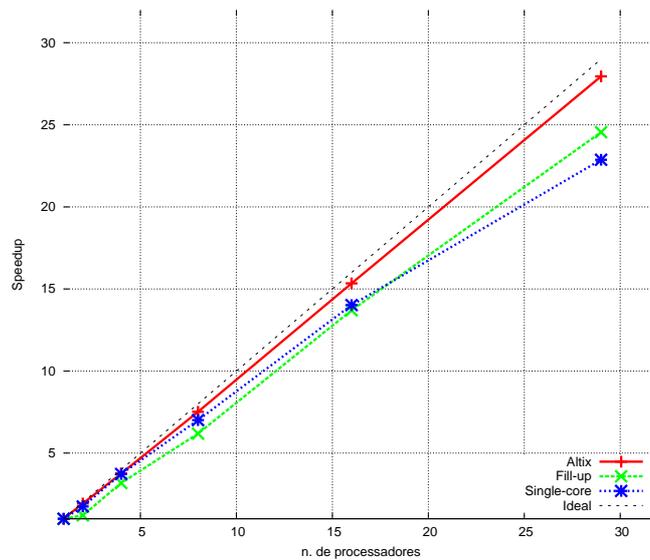


Figura 6.36: *Speedup* obtidos pela Altix e Enterprise 3 - Malha 2048×2048 - Problema do traçador.

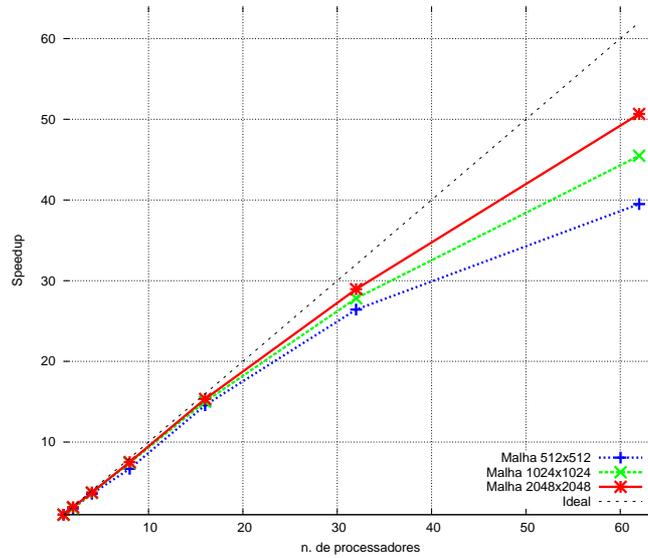


Figura 6.37: *Speedups* produzidos pela Altix considerando várias malhas - Problema do traçador.

menos carregado, a estrutura de dados aresta-por-aresta será referenciada apenas por Aresta e a estratégia elemento-por-elemento será denominada apenas por Elemento. Na simulação foram utilizados os dois problemas, injeção de traçador e injeção contínua para as malhas 512×512 e 2048×2048 . Considerando os resultados obtidos na Seção 6.4.1, no qual apontam, em geral, para um melhor desempenho do escalonamento *fill-up*, na comparação entre as duas estratégias será adotado esse escalonamento com os mesmos parâmetros utilizados nos teste anteriores. Na Fig. 6.38 e 6.39 que se referem aos *speedups* obtidos com o problema do traçador para, respectivamente, Enterprise 2 e Enterprise 3, os resultados indicam um comportamento homogêneo, e nota-se um melhor desempenho da estratégia Aresta em todos os processadores, principalmente quando se utiliza mais máquinas na realização dos trabalhos. Na Fig 6.38 observa-se que até 16 processadores as duas estratégias produzem *speedups* próximos. Para mais de 32 processadores a Aresta tem uma melhora mais significativa. Na Fig. 6.39 o mesmo comportamento é verificado. A eficiência da Aresta nos dois casos também é superior ao do Elemento como pode ser visto nas Figs. 6.40 e 6.41. No que se refere ao tempo de processamento, o desempenho da Aresta foi sempre mais rápido. Interessante observar que os ganhos alcançados na Enterprise 2 foram proporcionalmente mais significativos que os apresentados na

Enterprise 3. Os tempos de processamento podem ser observados na Fig. 6.42(a) para a malha pequena, 512×512 , e na Fig. 6.42(b) para a malha maior, 2048×2048 .

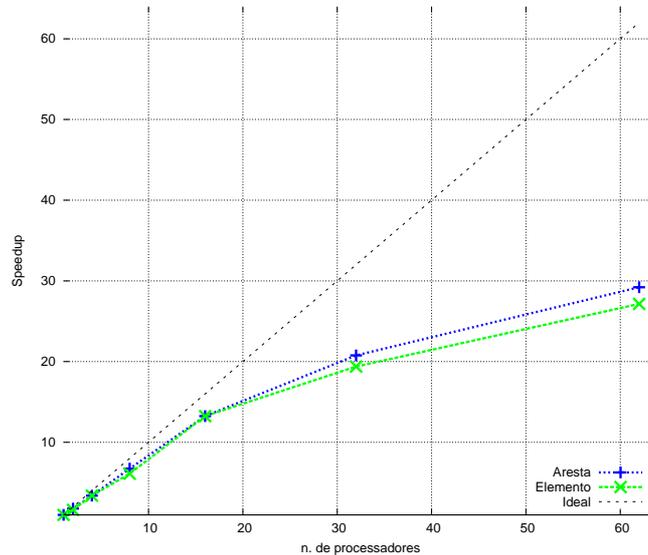


Figura 6.38: *Speedups* comparando Elemento e Aresta - Problema do traçador-Malha 512×512 - Enterprise 2.

Na malha maior, novamente, foi observado que houve um melhor desempenho da estrutura Aresta em praticamente todos os casos, em ambas as máquinas, conforme ilustram a Fig. 6.43 e Fig. 6.44. A exceção ocorreu para 32 processadores na Enterprise 2, Fig. 6.43, onde a estrutura Elemento obteve desempenho ligeiramente superior ao da Aresta. A eficiência da Aresta também foi superior conforme indica a Fig. 6.45 e Fig. 6.46. Outro ganho que a estrutura Aresta teve, foi com relação ao tempo de processamento que é apresentado Fig. 6.47(a) para a malha 2048×2048 , onde se observa uma melhor performance da Aresta, no entanto, para essa malha os ganhos na Enterprise 3, proporcionalmente, foram mais significativos que na Enterprise 2.

Na Altix, a estrutura Aresta também demonstrou um melhor desempenho. A Fig. 6.48 mostra os *speedups* de ambas as estruturas obtidos nessa máquina, para uma malha 2048×2048 .

A próxima análise será realizada utilizando o problema de injeção contínua observado

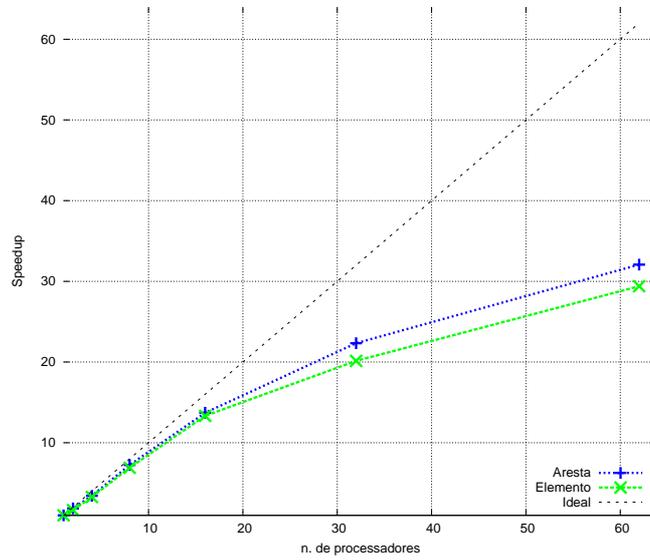


Figura 6.39: *Speedups* comparando Elemento e Aresta - Problema do traçador - Malha 512×512 - Enterprise 3.

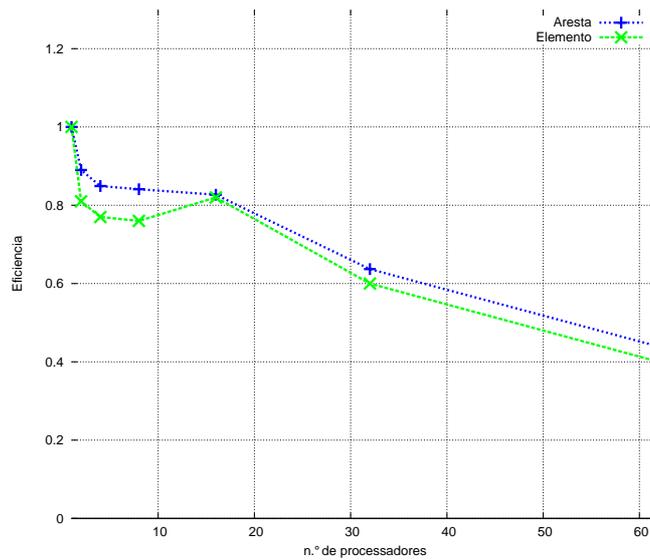


Figura 6.40: Eficiência de Elemento e Aresta - Problema do traçador- Malha 512×512 - Enterprise 2.

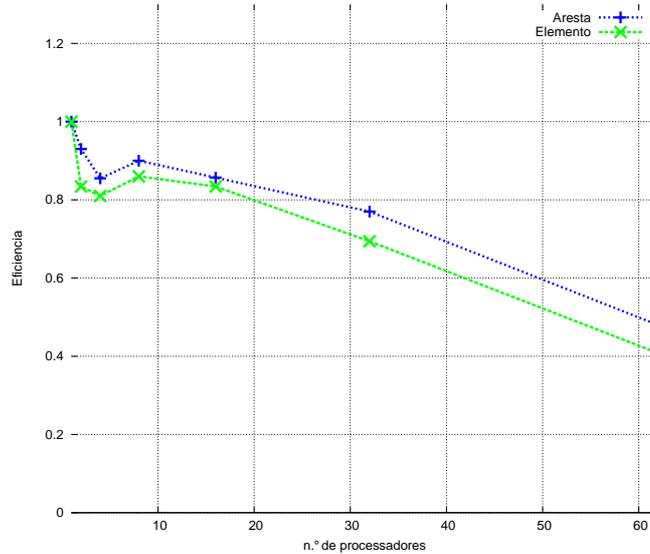
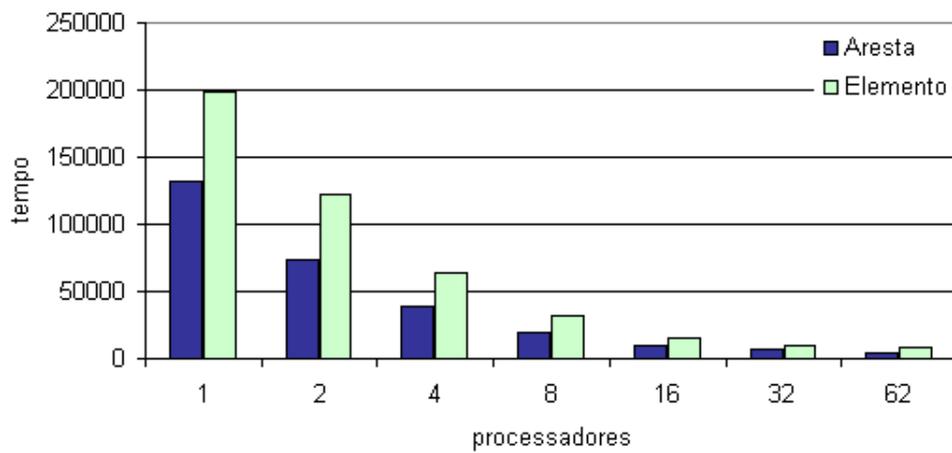


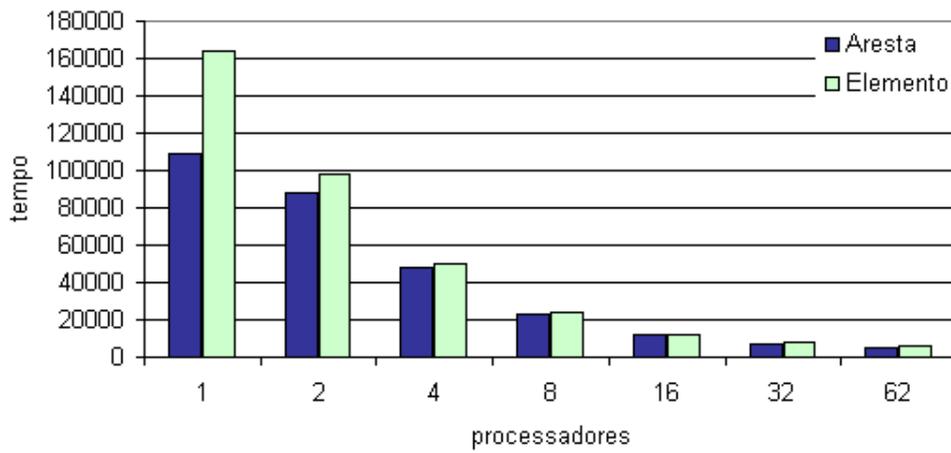
Figura 6.41: Eficiência de Elemento e Aresta - Problema do traçador- Malha 512×512 - Enterprise 3.

em 500 dias. As malhas consideradas foram as mesmas do problema de injeção de traçador. A Fig. 6.49 e 6.50 apresentam os *speedups* obtidos com o problema para as duas Enterprises na malha 512×512 . Pode ser observado que a estratégia Aresta obteve um melhor *speedup* em todos os pontos do gráfico. A eficiência também reafirma o melhor desempenho da estrutura Aresta e é apresentada na Fig. 6.51 e 6.52.

Na malha maior, o comportamento da estrutura Aresta não se altera e ela continua a produzir um *speedup* melhor que a estrutura Elemento. Esse fato é apresentado na Fig. 6.54 e Fig. 6.55. A eficiência também consolida esse melhor desempenho da estrutura Aresta e pode ser conferida na Fig. 6.56 e Fig. 6.57. Com relação ao tempo de processamento, os resultados mantêm o mesmo comportamento da superioridade da Aresta em relação ao Elemento. Dessa vez os ganhos, proporcionais em ambas as Enterprises, se mantiveram equiparados. Os tempos podem ser visualizados na Fig. 6.58. Na Altix, a estrutura Aresta também demonstrou um melhor desempenho para esse problema. A Fig. 6.59 mostra os *speedups* de ambas as estruturas obtidos nessa máquina para uma malha 2048×2048 .



(a) Enterprise 2



(b) Enterprise 3

Figura 6.42: Tempo de processamento comparando as estruturas Aresta e Elemento nas duas Enterprises - Problema do traçador - Malha 512×512 .

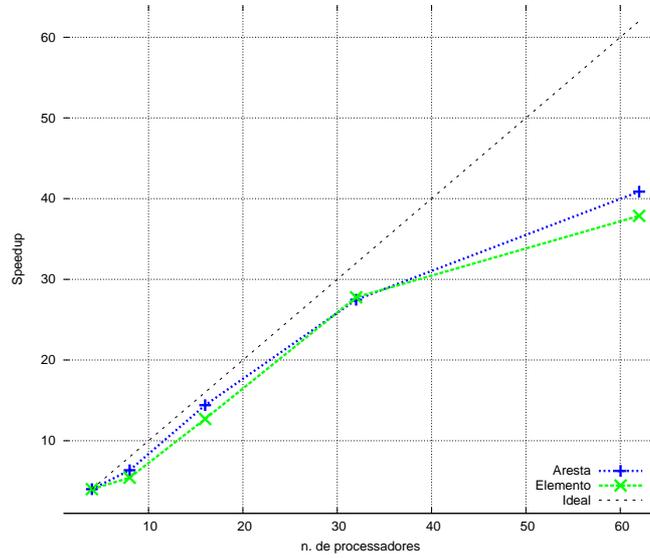


Figura 6.43: *Speedups* comparando Elemento e Aresta - Problema do traçador-Malha 2048×2048 - Enterprise 2.

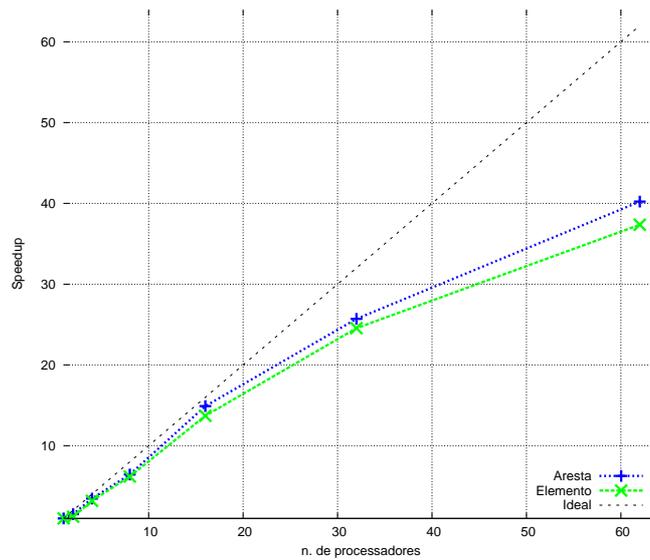


Figura 6.44: *Speedups* comparando Elemento e Aresta - Problema do traçador-Malha 2048×2048 - Enterprise 3.

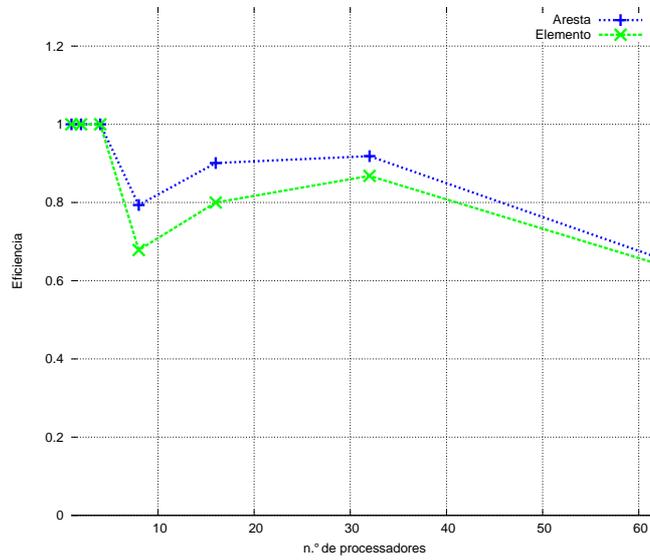


Figura 6.45: Eficiência de Elemento e Aresta - Problema do traçador - Malha 2048×2048 - Enterprise 2.

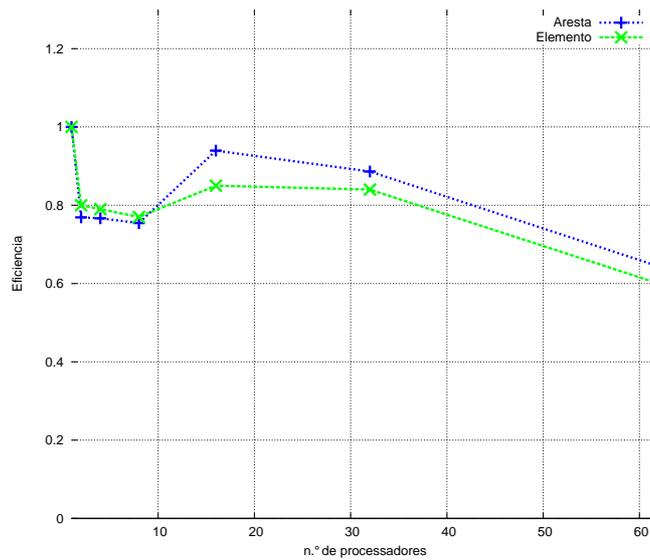
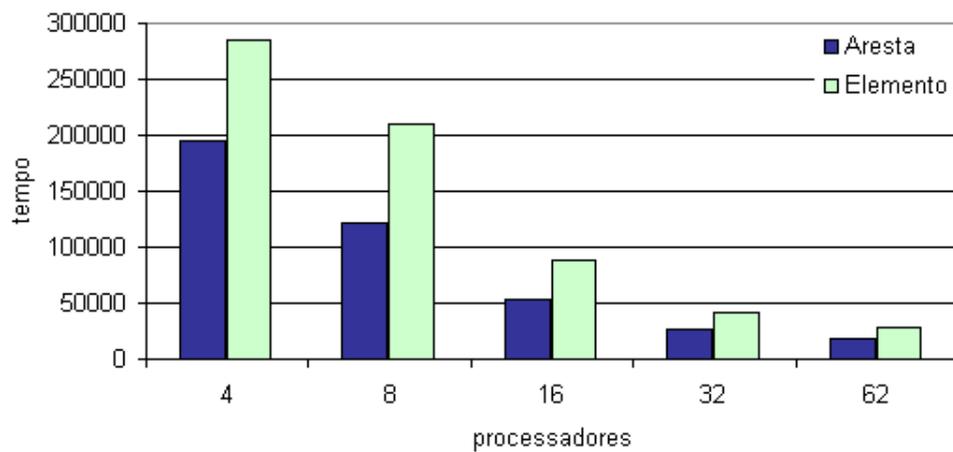
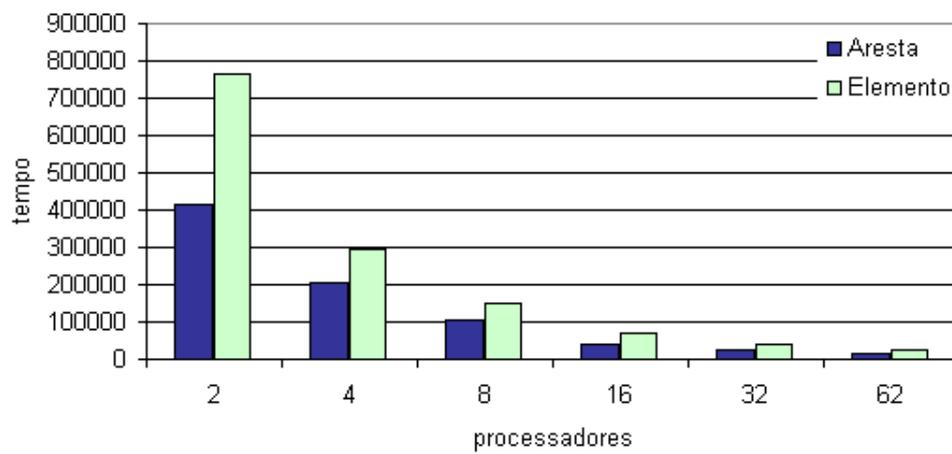


Figura 6.46: Eficiência de Elemento e Aresta - Problema do traçador - Malha 2048×2048 - Enterprise 3.



(a) Enterprise 2



(b) Enterprise 3

Figura 6.47: Tempo de processamento comparando as estruturas Aresta e Elemento nas duas Enterprises - Problema do traçador - Malha 2048×2048 .

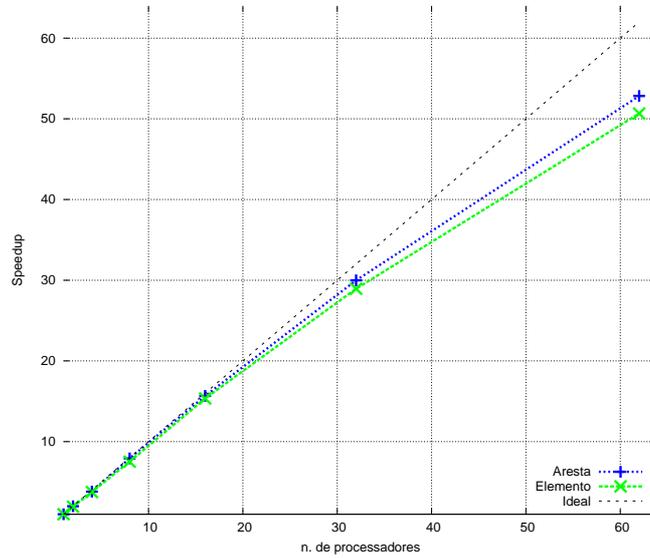


Figura 6.48: *Speedups* comparando Elemento e Aresta - Problema do traçador - Malha 2048×2048 - Altix.

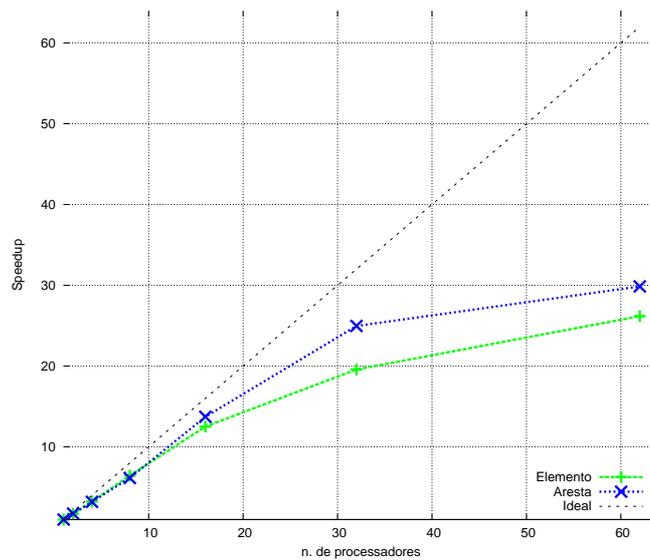


Figura 6.49: *Speedups* comparando Elemento e Aresta - Problema injeção contínua - Malha 512×512 - Enterprise 2.

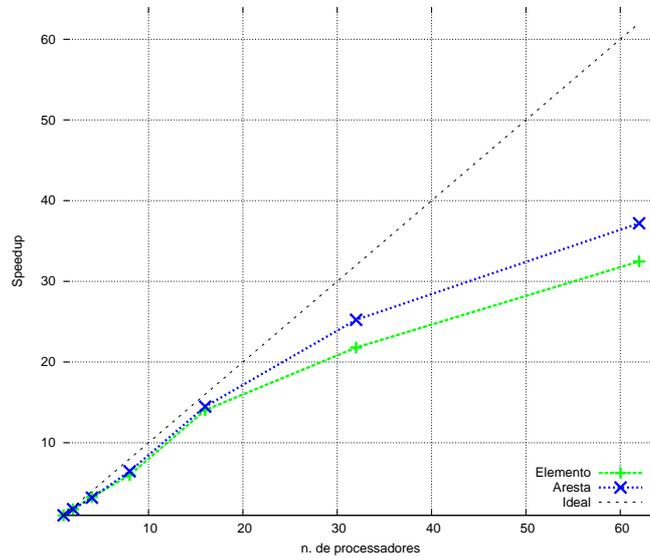


Figura 6.50: *Speedups* comparando Elemento e Aresta - Problema injeção contínua - Malha 512×512 - Enterprise 3.

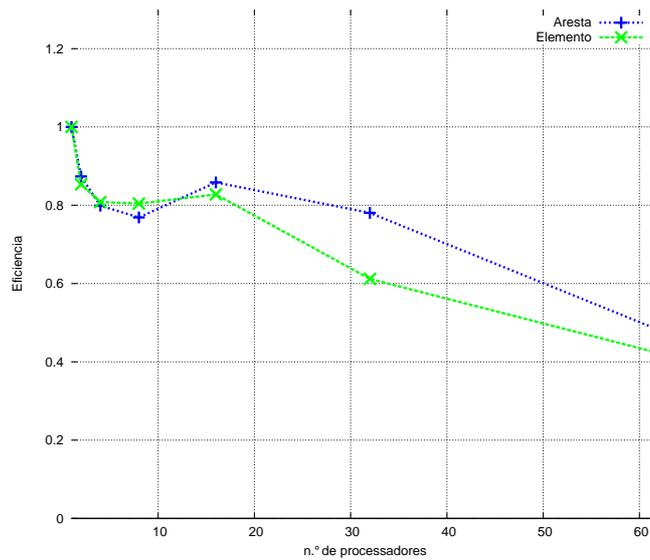


Figura 6.51: Eficiência de Elemento e Aresta - Problema injeção contínua- Malha 512×512 - Enterprise 2.

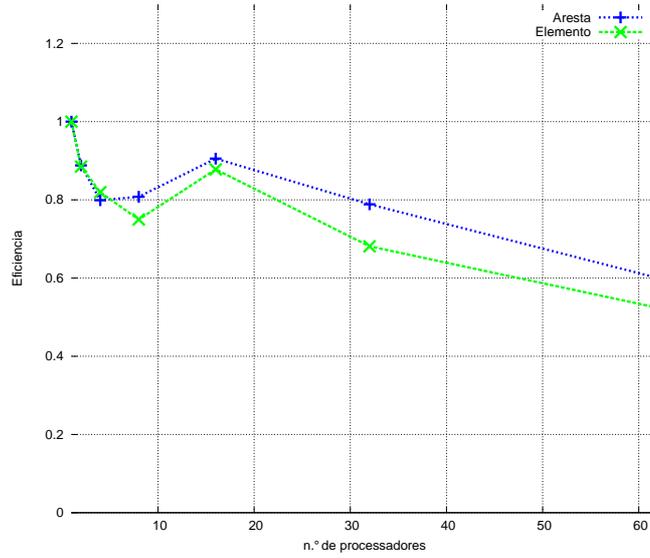


Figura 6.52: Eficiência de Elemento e Aresta - Problema injeção contínua- Malha 512×512 - Enterprise 3.

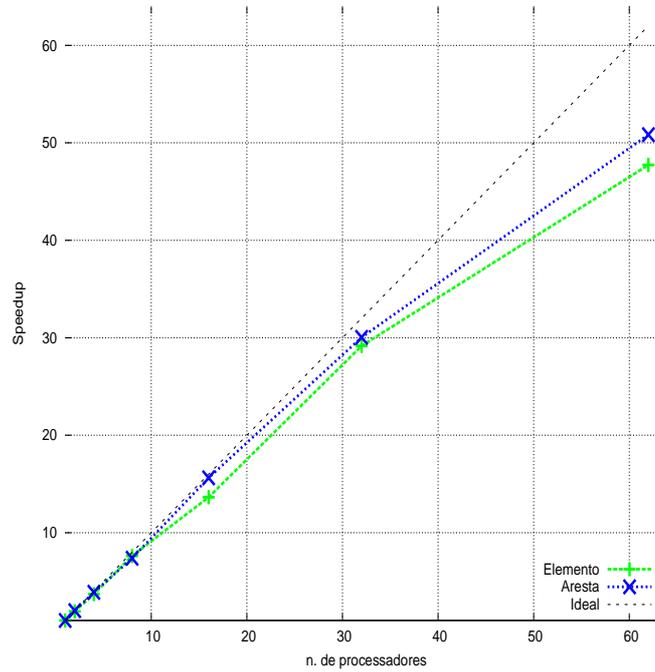
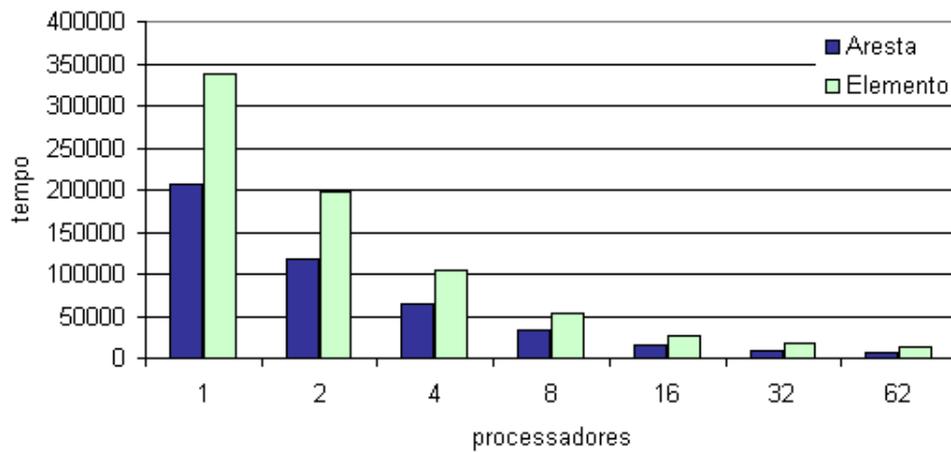
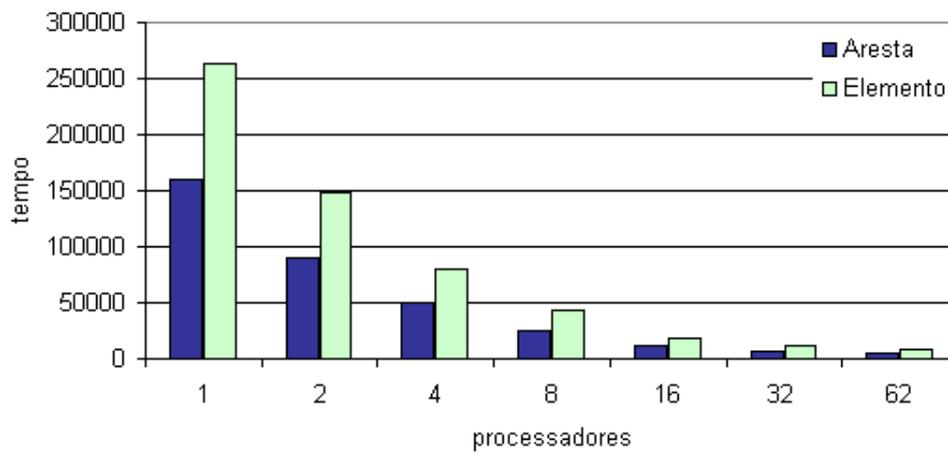


Figura 6.59: *Speedups* comparando Elemento e Aresta - Problema injeção contínua - Malha 2048×2048 - Altix.

Na realização dos testes foi constatado que grande parte do tempo de processamento é gasto nas rotinas que realizam as operações de comunicação. Essas conclusões foram obtidas através da ferramenta TAU. Para ilustrar como os resultados foram



(a) Enterprise 2



(b) Enterprise 3

Figura 6.53: Tempo de processamento comparando as estruturas Aresta e Elemento
 - Malha 512×512 - Problema injeção contínua

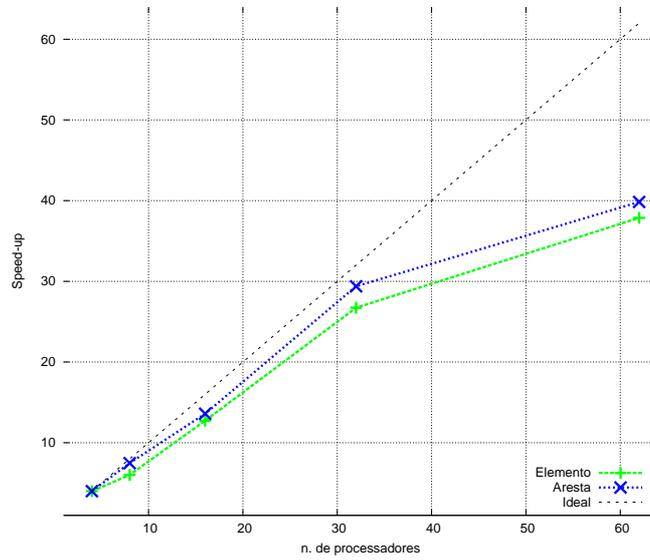


Figura 6.54: *Speedups* comparando Elemento e Aresta - Problema injeção contínua-
Malha 2048×2048 - Enterprise 2.

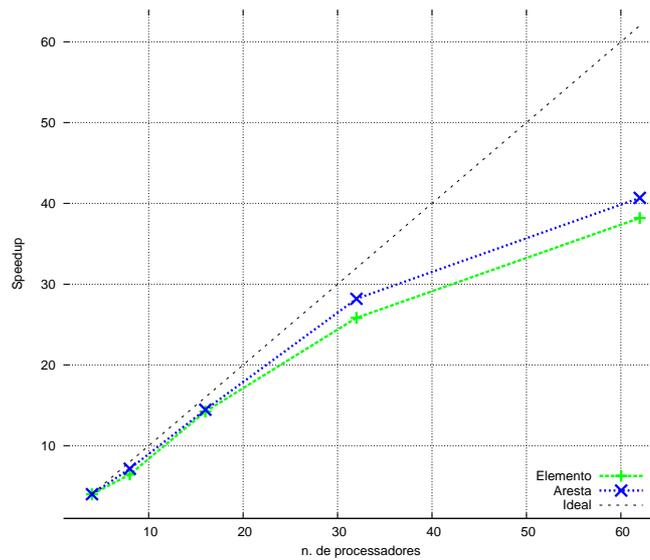


Figura 6.55: *Speedups* comparando Elemento e Aresta - Problema injeção contínua-
Malha 2048×2048 - Enterprise 3.

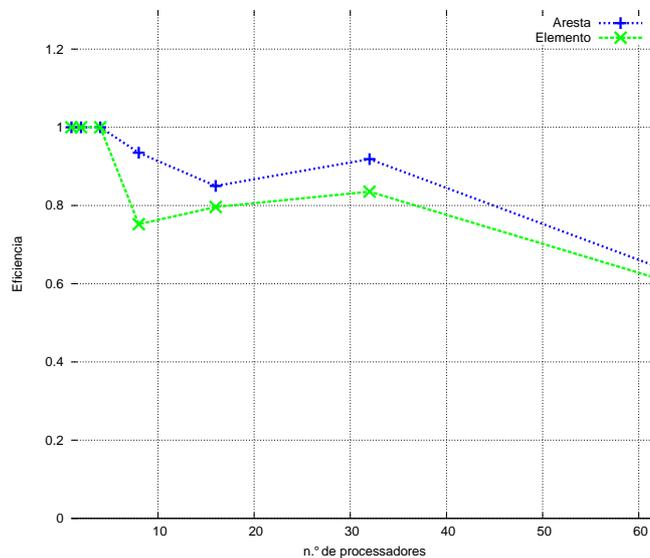


Figura 6.56: Eficiência de Elemento e Aresta - Problema injeção contínua - Malha 2048×2048 - Enterprise 2.

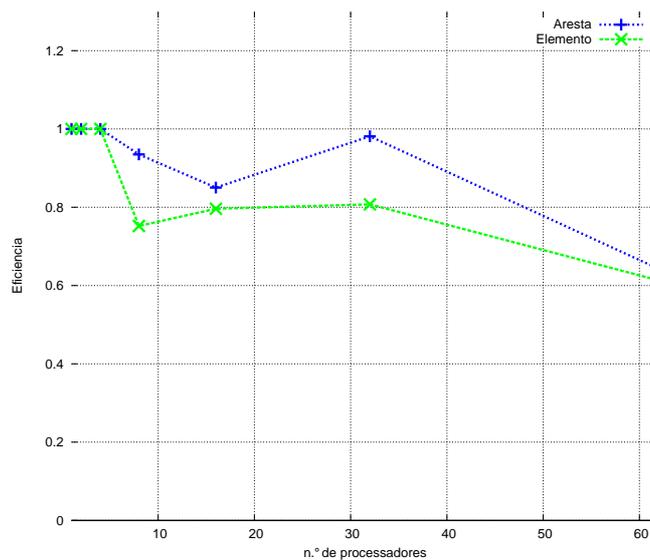
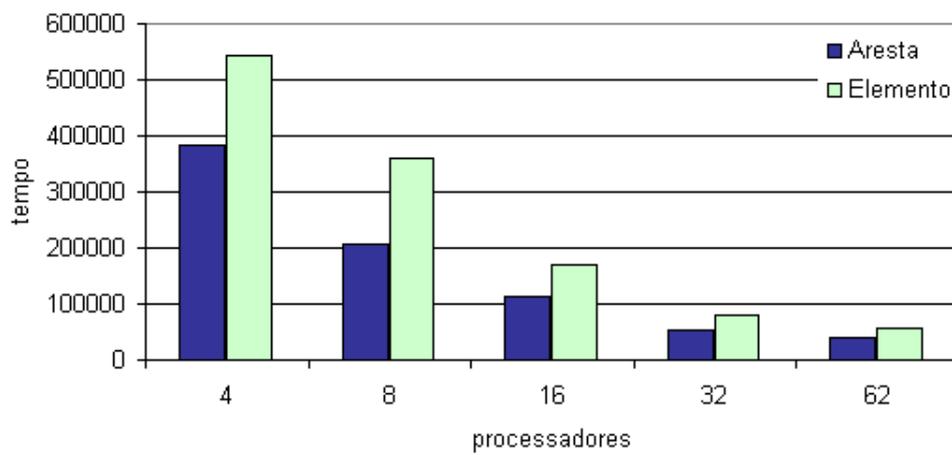
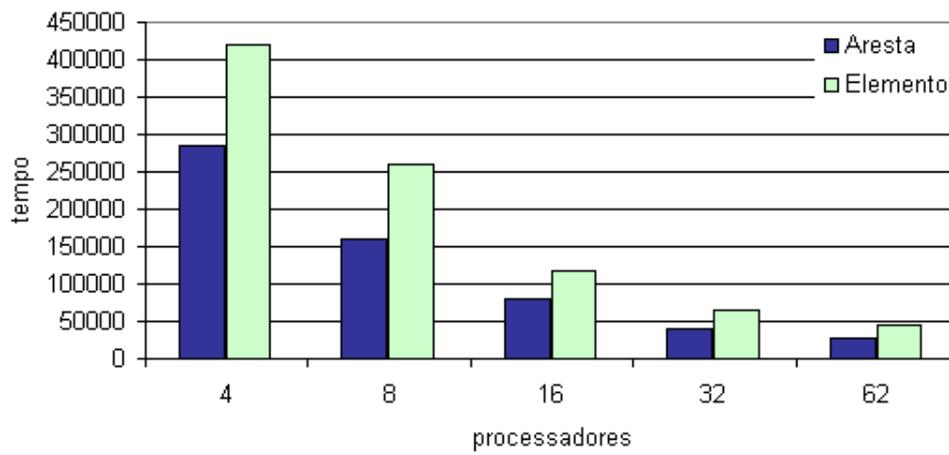


Figura 6.57: Eficiência de Elemento e Aresta - Problema injeção contínua - Malha 2048×2048 - Enterprise 3.



(a) Enterprise 2



(b) Enterprise 3

Figura 6.58: Tempo de processamento comparando as estruturas Aresta e Elemento
- Malha 2048×2048 - Problema injeção contínua

obtidos através dessa ferramenta, o problema do traçador foi executado em dois processadores considerando uma malha bem pequena, 80×80 células, em uma máquina *dual-core*. A Fig. 6.60 mostra a distribuição do tempo observada para o processador 0 considerando as funções mais significativas com relação ao tempo. A Fig. 6.61 apresenta a divisão do tempo para os dois processadores e também mostra o desvio padrão e a média dos tempos consumidos com as principais funções. A Fig. 6.62 exhibe a função que demanda mais tempo na aplicação. Essa função está relacionada a função do MPI, *All_reduce*, utilizada na rotina que realiza o produto interno. Ela confirma como a comunicação tem importância na aplicação, em especial, a comunicação coletiva.

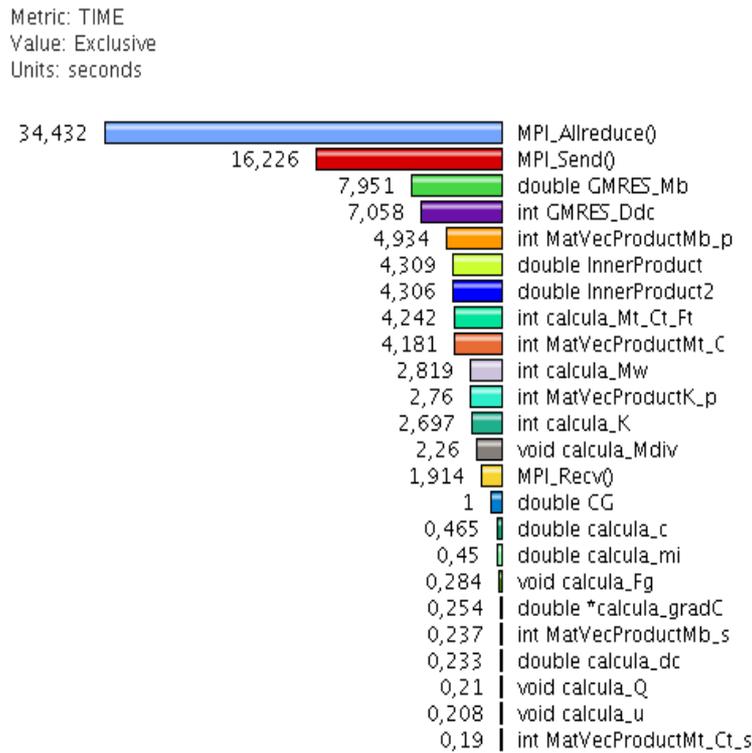


Figura 6.60: Distribuição de tempo observado para o processador 0 - Problema do Traçador.



Figura 6.61: Divisão do tempo de execução na TAU - Problema do Traçador.



Figura 6.62: Rotina *All_reduce* utilizada na operação de produto interno - Problema do Traçador.

Capítulo 7

Conclusões

Nesse trabalho foi utilizada uma versão paralela do método dos elementos finitos baseada na decomposição de domínio para resolver o modelo matemático que governa o fenômeno de escoamento miscível em meios porosos. A implementação foi empregada para realizar um estudo comparativo entre as estratégias de armazenamento elemento-por-elemento [28] e aresta-por-aresta [10, 40] em ambientes de processamento paralelo com memória distribuída, em particular, *Clusters* de estações de trabalho *multi-core*. No entanto, antes da comparação em si, fez-se necessário desenvolver uma análise sobre dois tipos de escalonamento *fill-up* e *single-core* nos *Clusters* utilizados.

A formulação utilizada para resolver o conjunto de equações diferenciais parciais que descreve o problema tratado é uma ferramenta importante na simulação da invasão de fluidos em meios porosos. Duas aplicações importantes foram simuladas, injeção de traçadores e injeção contínua. Foi, observado através dos experimentos de validação, que a implementação proposta levou a resultados compatíveis com os resultados encontrados na literatura. No decorrer dos estudos, foi observado que o problema é muito instável, principalmente o deslocamento com mobilidade adversa. Verificou-se uma forte dependência com a orientação de malha, com as técnicas de estabilização e com a precisa definição das condições iniciais e de contorno. Além disso, a resolução das aplicações demandou um tempo de processamento considerável

para malhas pequenas, o que torna a solução inviável em uma máquina, para malhas mais refinadas. Sendo assim, ficou evidente a necessidade de paralelização.

Na investigação do melhor tipo de escalonamento para a aplicação, considerando os dois *Clusters multi-core* utilizados, observou-se através dos gráficos de *speedup*, eficiência e tempo de processamento que o escalonamento *fill-up*, no geral, apresentou um desempenho melhor que o escalonamento *single-core*. No entanto, seu desempenho ficou comprometido quando houve mais competição por recursos dentro da mesma máquina, caso observado para problemas grandes, distribuídos entre poucos processadores. Portanto, para retirar o máximo de proveito das arquiteturas *multi-core*, é preciso tomar cuidado para que o balanceamento de carga não ocasione muita competição interna por recursos dentro da mesma máquina.

Na comparação entre as duas estratégias de armazenamento propostas, os resultados apontaram que a estratégia aresta-por-aresta apresentou maiores valores de *speedup* e eficiência que a estratégia elemento-por-elemento. Logo, a estratégia aresta-por-aresta mostrou-se mais adequada ao processo de paralelização. Além disso, foi a estratégia mais rápida em todos os testes realizados. Ainda foi verificado que os resultados obtidos são essencialmente os mesmos para as duas técnicas de armazenamento. Significa que o número de iterações não-lineares e lineares, o número de passos no tempo e o resíduo em cada passo no tempo, considerando a mesma tolerância, são idênticos para as duas.

Um outro ponto importante é quanto a precisão da técnica de paralelização adotada. Nos testes realizados, dentro da tolerância estabelecida, foi observado que o número de iterações e o resíduo da solução foram os mesmos, não sendo influenciada pelo número de processadores envolvidos, pelas estratégias de armazenamento utilizadas ou pelo método de resolução dos sistemas. Isso é um bom indício, para atestar a validade e qualidade do algoritmo paralelo, pois, obter os mesmos resultados do algoritmo sequencial equivalente é uma forma natural de validação [35].

Em pesquisas futuras pode-se aprimorar este trabalho implementando pré-condicionadores para os algoritmos utilizados na solução dos sistemas, implementar o armazenamento CSR *Compressed Storage Row* [55], utilizar malhas não-estruturadas,

extendê-lo para escoamento imiscíveis, implementar modelos tridimensionais com aplicações mais práticas, realizar estudos com dedilhamentos viscosos ("*viscous fingering*") em meios anisotrópicos [15], estudar o problema de concorrência da hierarquia de memória nos *Clusters multi-core* e utilizar todo o poder de processamento das GPUs (*Graphic Processor Unit*) [46], atualmente massivamente empregado em aplicações gráficas, para resolver problemas de escoamento em meios porosos. E mesmo utilizar *Cluster* composto por GPUs.

Referências Bibliográficas

- [1] A.L.G.A.Coutinho and J.L.D. Alves. Finite element simulation of nonlinear viscous fingering in miscible displacement with anisotropic dispersion and non-monotonic viscosity profiles. *Computational Mechanics*, 423:108–116, 1999.
- [2] A. C. Barbosa, L. Catabriga, A. M. P. Valli, S. M. C. Malta, and L. M. Lima. Experiments using a finite element formulation of incompressible miscible displacements in porous media. In *XXXII Congresso Nacional de Matemática Aplicada e Computacional (a aparecer)*, 2009.
- [3] M. R. Borges. *Injeção de traçadores em reservatórios de petróleo: modelagem multi-escala e simulação numérica*. PhD thesis, Departamento de Modelagem Computacional - IPRJ/UERJ, Rio de Janeiro, RJ, Fevereiro 2006.
- [4] A. N. Brooks and T. J. R. Hughes. Streamline upwind/ Petrov-galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Comput. Methods Appl. Mech. and Engrg.*, 32:199–259, 1982.
- [5] R. Buyya. *High Performance Cluster Computing: Architecture and System*. Prentice Hall, v.1 edition, 1999.
- [6] J. J. Camata. Algoritmos adaptativos para o método dos elementos finitos utilizando a biblioteca libmesh. Master's thesis, Programa de Pós-Graduação em Informática - UFES, Espírito Santo, ES, Outubro 2006.

- [7] E. C. A. Carvalho. Particionamento de grafos de aplicações e mapeamento em grafos de arquiteturas heterogêneas. Master's thesis, Programa de Pós-Graduação em Computação - UFRGS, Porto Alegre, Junho 2002.
- [8] L. Catabriga. *Soluções implícitas das equações de Euler empregando estruturas de dados por aresta*. PhD thesis, Programa de Engenharia Civil - COPPE/UFRJ, Rio de Janeiro, RJ, Maio 2000.
- [9] L. Catabriga, M. D. A. Martins, A. L. G. A. Coutinho, and J. L. D. Alves. Clustered edge-by-edge preconditioners for non-symmetric finite element equations. In *4th World Congress on Computational Mechanics*, 1998.
- [10] L. Catabriga, M. D. A. Martins, A. L. G. A. Coutinho, and J. L. D. Alves. Clustered edge-by-edge preconditioners for non-symmetric finite element equations. In *4th World Congress on Computational Mechanics*, 1998.
- [11] M.R. Correa, A.E.D. Loula, and E.L.M. Garcia. Métodos de diferenças finitas de direções alternadas implícitas para modelagem de águas subterrâneas. *TEMA Tend. Mat. Apl. Comput.*, 5(1):65–76, 2004.
- [12] A.L.G.A Coutinho and J.L.D. Alves. Parallel finite element simulation of miscible displacement in porous media. *SPE Journal*, 4(1):487–500, 1996.
- [13] A.L.G.A. Coutinho, C.M. Dias, and J.L.D. Alves. Stabilized methods and post-processing techniques for miscible displacement. *Computational Mechanics*, 193:1421–1436, 2004.
- [14] M. C. Damiani. Determinação de padrões de fluxo em simulações de reservatórios de petróleo utilizando traçadores. Master's thesis, Programa de Engenharia Civil - COPPE/UFRJ, Rio de Janeiro, RJ, Maio 2007.
- [15] M. C. Dias. *Técnicas de integração reduzida para simulação de problemas não-lineares de transporte pelo método dos elementos finitos*. PhD thesis, Programa de Engenharia Civil - COPPE/UFRJ, Rio de Janeiro, RJ, Novembro 2001.

- [16] R. V. Dorneles. *Particionamento de Domínio e Balanceamento de Carga no modelo HIDRA*. PhD thesis, Programa de Pós-Graduação em Computação - UFRGS, Porto Alegre, Junho 2003.
- [17] R. E. Ewing. *The Mathematics of Reservoir Simulation*. SIAM, Philadelphia, 1983.
- [18] C.A.C. Fernandes. Estudo de algumas ferramentas de coleta e visualização de dados de desempenho de aplicações paralelas no ambiente mpi. Master's thesis, Programa de Pós-Graduação em Engenharia Elétrica e Computação - UFRN, Setembro 2003.
- [19] L. C. P. Silva Filho. Estudo de casos com aplicações científicas de alto desempenho em agregados de computadores multi-core. Master's thesis, Programa de Pós-Graduação em Computação - UFSC, Florianópolis, SC, 2008.
- [20] M. J. Flynn. Very high-speed computing systems. *IEEE*, 54:1901–1909, 1966.
- [21] I. T. Foster. *Designing and Building Parallel Programs*. Addison Wesley, Usa, 1995.
- [22] G. Galante. Métodos multigrid paralelos em malhas não-estruturadas aplicados a simulação de problemas de dinâmica de fluidos computacional e transferência de calor. Master's thesis, Programa de Pós-Graduação em Computação - UFRGS, Porto Alegre, Março 2006.
- [23] A.C Galeao and E.G.D Carmo. A consistent approximate upwind petrov-galerkin method for convection-dominated problems. *Computer Methods in Applied Mechanics and Engineering*, 68:83–95, 1988.
- [24] A. Geist, A. Beguelin, J. Dongarra, W. Jiang, R. Manchek, and V. Sunderam. *PVM Parallel Virtual Machine, A User's Guide and Tutorial for Networked Parallel Computing*. MIT Press, Cambridge, Mass., 1994.
- [25] C. Geuzaine and J. F. Remacle. *The documentation for Gmsh 2.3. A finite element mesh generator with built-in-pre- and post-processing facilities*, March 2009.

- [26] High Performance Fortran Forum. High Performance Fortran language specification, version 1.0. Technical Report CRPC-TR92225, Houston, Tex., 1993.
- [27] T. J. R. Hughes. *The Finite Element Method*. Prentice-Hall International, 1987.
- [28] T. J. R. Hughes. *The Finite Element Method - Linear Static and Dynamic Finite Element Analysis*. Prentice-Hall International, Inc., 1987.
- [29] P. K. Jimack and N. Touheed. Developing parallel finite element software using mpi. In B.H.V. Topping and L. Lammer, editors, *High Performance Computing for Computational Mechanics*, pages 15–38. Saxe-Coburg Publications, 2000.
- [30] D. Pecinin Jr. Paralelizações de métodos numéricos em cluster empregando as bibliotecas mpich,deck e pthread. Master’s thesis, Programa de Pós-Graduação em Computação - UFRGS, Porto Alegre, Fevereiro 2003.
- [31] G. Karypis and V. Kumar. A software package for partitioning unstructure graphs, partitioning meshes, and computing fill-orderings of sparse matrices. Technical report, University of Minnesota, Department of Computer Science, 1998.
- [32] P. Keleher, S. Dwarkadas, A. L. Cox, and W. Zwaenepoel. Treadmarks: Distributed shared memory on standard workstations and operating systems. In *Proc. of the Winter 1994 USENIX Conference*, pages 115–131, 1994.
- [33] Advanced Computing Laboratory. Tau: Tuning and analysis utilities. Technical report, Advanced Computing Laboratory, Los Alamos National Laboratory Publication, November 1999.
- [34] Open Systems Laboratory. *X/Modif based graphical user interface for running, debugging and visualizing MPI programs*. Bloomington,IN, USA, 2009.
- [35] L. M. Lima. Estratégias de armazenamento paralelas aplicadas ao método dos elementos finitos. Master’s thesis, Programa de Pós-Graduação em Informática - UFES, Espírito Santo, ES, Dezembro 2004.

- [36] W. E. Brigham M. Abbaszadeh-Dehghani. Analysis of unit mobility ratio well-to-well tracer flow to determine reservoir heterogeneity. Technical report, University of Stanford, Petroleum Research Institute, August 1982.
- [37] S.M.C. Malta, A.F.D. Loula, and E.L.M Garcia. A post-processing technique to approximate the velocity field in miscible displacement simulations. *Matemática contemporânea*, 8:239–268, 1995.
- [38] S.M.C. Malta, A.F.D. Loula, and E.L.M Garcia. Numerical analysis of a stabilized finite element method for tracer injection simulations. *Computer Methods in Applied Mechanics and Engineering*, 187:119–136, 1998.
- [39] A. L. Martinotto. Resolução de sistemas de equações lineares através de métodos de decomposição de domínio. Master’s thesis, Programa de Pós-Graduação em Computação - UFRGS, Porto Alegre, Fevereiro 2004.
- [40] M. A. D. Martins. *Solução iterativa em paralelo de sistemas de equações do método dos elementos finitos empregando estruturas de dados por arestas*. PhD thesis, Programa de Engenharia Civil - COPPE/UFRJ, Rio de Janeiro, RJ, Agosto 1996.
- [41] A. L. Mendonça. *Simulação numérica de escoamentos incompressíveis bifásicos de fluidos não-newtonianos e imiscíveis em meios porosos via método dos elementos finitos*. PhD thesis, Programa de Engenharia Civil - COPPE/UFRJ, Rio de Janeiro, RJ, Outubro 2003.
- [42] Message Passing Interface Forum. MPI: A Message Passing Interface. In *Proceedings of Supercomputing '93*, pages 878–883. IEEE Computer Society Press, 1993.
- [43] W. E. Nagel, A. Arnold, and M. Weber. Vampir visualization and analysis of mpi resources. Technical report, Central Institute for Applied Mathematics Research Centre Julich, Julich, Germany, 1996.
- [44] W. G. Ney. Um estudo comparativo sobre formulações estabilizadas e adaptatividade na simulação de deslocamento miscíveis em meios porosos pelo

método dos elementos finitos. Master's thesis, Programa de Engenharia Civil - COPPE/UFRJ, Rio de Janeiro, RJ, Dezembro 2002.

- [45] B. Nicendo. *EasyMesh: A Two-Dimensional Quality Mesh Generator*, 2009.
- [46] Nvidia. *Nvidia Cuda Compute Unified device Architecture: Programming guide 2.0*. Nvidia, 2008.
- [47] B. U. Oztekin, G. Karypis, and V. Kumar. *Partitioned Mesh Visualizer: A program to visualize partitioned meshes*, 2009.
- [48] P.S. Pacheco. *A user's guide to mpi*. Morgan Kaufman Publishers, University of San Francisco, San Francisco, 1998.
- [49] M. F. Palin. *Técnicas de Decomposição de Domínio em Computação Paralela para simulação de campos eletromagnéticos pelo Método dos Elementos Finitos*. PhD thesis, Escola Politécnica - USP, São Paulo, SP, 2007.
- [50] D.W. Peaceman. *Fundamentals of Numerical Reservoir Simulation*. Elsevier Scientific Publishing Company, Amsterdam, The Netherlands, third edition, 1982.
- [51] D.W. Peaceman. Survey of problems in numerical reservoir simulation. *SIAM. J. Sci. Stat. Comput.*, 1986.
- [52] Cluster Resources. *Torque Resource Manager*. Cluster Resources, 2009.
- [53] R. Ribo, M. P. Riera, E. E.Terceno, J.S.P. Ronda, A. C. Sans, A. M. Ribera, and S. G. Horcas. *GID the personal pre and postprocessor*. Barcelona, Spain, 2008.
- [54] A. J. Rosa, R. S. Carvalho, and J. A. D. Xavier. *Engenharia de Reservatório de Petróleo*. Interciência, Rio de Janeiro, RJ, 2006.
- [55] Y. Saad. *Iterative Methods for Sparse Linear Systems*. SIAM, second edition, 2003.

- [56] Y. Saad and H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM. J. Sci. Stat. Comput.*, 7(3):856–869, 1986.
- [57] R. Schneiders. Software. In *Disponível em <http://www-users.informatik.rwth-aachen.de/~roberts/software.html>*, Acesso em 10 jul. 2009.
- [58] S. Shende and A. D. Malony. The tau parallel performance system. *International journal of High Performance Computing Applications*, 20:287–311, 2006.
- [59] J. R. Shewchuck. *Triangle: A Two-Dimensional Quality Mesh Generator and Delaunay Triangulator*. University of California, Berkeley, California, July 2005.
- [60] I. Slobodcicov. Implementação em paralelo do método dos elementos finitos para equações de águas rasas. Master’s thesis, Programa de Engenharia Civil - COPPE/UFRJ, Rio de Janeiro, RJ, Março 2003.
- [61] Sun. *Sun Grid Engine: Enterprise Edition 5.3 Administration User’s Guide*. Sun Microsystems, 2002.
- [62] C.T. Tan and G.M Homsy. Viscous fingerings with permeability heterogeneity. *Phys. fluids A*, 4(6):1099–1101, 1992.
- [63] A. Tanenbaum. *Organização estruturada de computadores*. Prentice Hall, Amsterdam, The Netherlands, 2001.
- [64] A. Tanenbaum and M. Steen. *Distributed system: principles and paradigms*. Prentice Hall, Amsterdam, The Netherlands, fourteen edition, 2002.
- [65] TotalView Technologies. *TotalView documentation*. TotalView Technologies, 2009.
- [66] J. E. Thomas. *Fundamentos de Reservatório de Petróleo*. Interciência, Rio de Janeiro, RJ, 2001.
- [67] C. Walshaw and M. Cross. Mesh partitioning: a multilevel balancing and refinement algorithm. Technical report, University of Greenwich, 1998.

- [68] L.C. Young. A finite element method for reservoir simulation. *Society of Petroleum Engineers, SPE*, 4, 1981.

Livros Grátis

(<http://www.livrosgratis.com.br>)

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)
[Baixar livros de Literatura de Cordel](#)
[Baixar livros de Literatura Infantil](#)
[Baixar livros de Matemática](#)
[Baixar livros de Medicina](#)
[Baixar livros de Medicina Veterinária](#)
[Baixar livros de Meio Ambiente](#)
[Baixar livros de Meteorologia](#)
[Baixar Monografias e TCC](#)
[Baixar livros Multidisciplinar](#)
[Baixar livros de Música](#)
[Baixar livros de Psicologia](#)
[Baixar livros de Química](#)
[Baixar livros de Saúde Coletiva](#)
[Baixar livros de Serviço Social](#)
[Baixar livros de Sociologia](#)
[Baixar livros de Teologia](#)
[Baixar livros de Trabalho](#)
[Baixar livros de Turismo](#)