



Universidade Federal da Paraíba
Centro de Tecnologia
Programa de Pós-Graduação em Engenharia Mecânica
Mestrado - Doutorado

**ESTUDO DE OTIMIZAÇÃO DA RECONSTRUÇÃO
DE IMAGENS VIA TOMOGRAFIA CAPACITIVA
ELÉTRICA ATRAVÉS DOS MÉTODOS DE
ELEMENTOS FINITOS E ENXAME DE
PARTICULAS UTILIZANDO COMPUTAÇÃO DE
ALTO DESEMPENHO**

Por

Bruno César Bezerra Nóbrega de Souza

*Defesa de Doutorado apresentada à Universidade Federal da
Paraíba para obtenção do grau de Doutor.*

João Pessoa - Paraíba

Dezembro, 2017

BRUNO CÉSAR BEZERRA NÓBREGA DE SOUZA

**ESTUDO DE OTIMIZAÇÃO DA RECONSTRUÇÃO DE IMAGENS
VIA TOMOGRAFIA CAPACITIVA ELÉTRICA ATRAVÉS DOS
MÉTODOS DE ELEMENTOS FINITOS E ENXAME DE
PARTICULAS UTILIZANDO COMPUTAÇÃO DE ALTO
DESEMPENHO**

Tese de doutorado apresentada ao curso de Pós-Graduação em Engenharia Mecânica da Universidade Federal da Paraíba, em cumprimento às exigências para obtenção do Grau de Doutor.

Orientador: Professor Dr. Francisco Antônio Belo

João Pessoa - Paraíba

Dezembro, 2017

Catálogo na publicação
Seção de Catalogação e Classificação

S719e Souza, Bruno Cesar Bezerra Nobrega de.
ESTUDO DE OTIMIZAÇÃO DA RECONSTRUÇÃO DE IMAGENS VIA
TOMOGRAFIA CAPACITIVA ELÉTRICA ATRAVÉS DOS MÉTODOS DE
ELEMENTOS FINITOS E ENXAME DE PARTICULAS UTILIZANDO
COMPUTAÇÃO DE ALTO DESEMPENHO / Bruno Cesar Bezerra
Nobrega de Souza. - João Pessoa, 2017.
176 f. : il.

Orientação: FRANCISCO ANTÔNIO BELO.
Tese (Doutorado) - UFPB/CT.

1. TOMOGRAFIA CAPACITIVA ELÉTRICA. 2. MÉTODO DE
ELEMENTOS FINITOS. 3. OTIMIZAÇÃO POR ENXAME DE
PARTÍCULAS. I. BELO, FRANCISCO ANTÔNIO. II. Título.

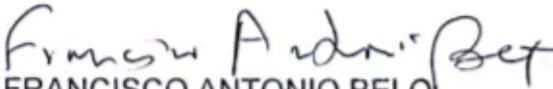
UFPB/BC

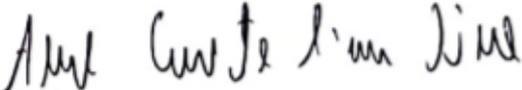
**ESTUDO DE OTIMIZAÇÃO DA RECONSTRUÇÃO DE IMAGENS VIA
TOMOGRAFIA CAPACITIVA ELÉTRICA ATRAVÉS DOS MÉTODOS DE
ELEMENTOS FINITOS E ENXAME DE PARTÍCULAS UTILIZANDO
COMPUTAÇÃO DE ALTO DESEMPENHO**

por

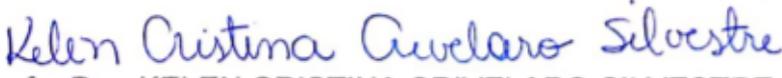
Bruno César Bezerra Nóbrega de Souza

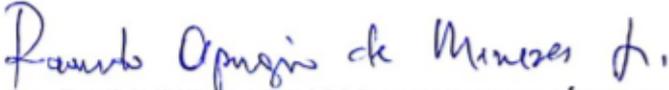
Tese aprovada em 18 de dezembro de 2017


Prof. Dr. FRANCISCO ANTONIO BELO
Orientador


Prof. Dr. ABEL CAVALCANTE LIMA FILHO
Examinador Interno


Prof. Dr. MARCELO CAVALCANTI RODRIGUES
Examinador Interno


Profa. Dra. KELEN CRISTINA CRIVELARO SILVESTRE
Examinadora Externa


Prof. Dr. RAIMUNDO APRIGIO MENEZES JÚNIOR
Examinador Externo

AGRADECIMENTOS

Agradeço ao meu orientador prof. Francisco Belo pela oportunidade e pela confiança em mim depositada.

Agradeço ao prof. Raimundo Menezes pela paciência e empenho na revisão desse trabalho.

Agradeço muito a minha noiva Kelly pelo apoio e pela compreensão nos momentos difíceis.

Agradeço a minha mãe e a minha filha, Dalvanira e Camilla, que sempre me apoiaram nessa árdua tarefa.

Agradeço também a todos que forma direta ou indireta contribuíram para que esse trabalho fosse realizado.

ESTUDO DE OTIMIZAÇÃO DA RECONSTRUÇÃO DE IMAGENS VIA TOMOGRAFIA CAPACITIVA ELÉTRICA ATRAVÉS DOS MÉTODOS DE ELEMENTOS FINITOS E ENXAME DE PARTICULAS UTILIZANDO COMPUTAÇÃO DE ALTO DESEMPENHO

RESUMO

Este trabalho tem por finalidade apresentar a heurística de enxame de partículas como um método viável de solução para o problema inverso da técnica de Tomografia Capacitiva Elétrica (TCE), com o intuito de aplicação ao imageamento de escoamentos multifásicos. Também será apresentada a filosofia de programação paralela que aliada a alterações que favoreçam a ampliação da inteligência coletiva de enxame, reduz assim o tempo computacional de processamento, parâmetro crítico no projeto do sistema de reconstrução de imagens, necessário para produção de imagens com maior resolução espacial. Uma alternativa para ampliar a eficiência do sistema de reconstrução da imagem é a retirada de cálculos redundantes no código (apropriação da esparsividade e da ordenação das matrizes associadas ao problema matemático associado) e a migração do algoritmo de reconstrução de uma programação sequenciada para uma programação paralela. Será apresentado os princípios físicos da TCE, os estudos de aproveitamento das características numéricas do método dos elementos finitos (MEF), os algoritmos heurísticos utilizados no processo de reconstrução e os conceitos dos principais paradigmas de paralelização (memória compartilhada, memória distribuída) e sua implementação de software e hardware. Os resultados encontrados são encorajadores devido a concordância entre as imagens obtidas e os tomogramas simulados e a redução significativa no tempo de processamento das mesmas, estimulado assim a continuidade de pesquisas neste tema para o futuro. Encontra-se nesta tese todos os passos para construção de um cluster Beowulf e os códigos seriais e paralelos necessários para implementação dos algoritmos.

Palavras-chave: Tomografia Capacitiva Elétrica, Problema Inverso, Sistemas Multifásicos, Método de elementos finitos, Otimização por Enxame de Partículas

STUDY OF OPTIMIZATION OF THE IMAGES RECONSTRUCTION IN ELECTRICAL CAPACITANCE TOMOGRAPHY THROUGH FINITE ELEMENT METHODS AND PARTICLE SWARM USING HIGH PERFORMANCE COMPUTING

ABSTRACT

This work aims to present the particle swarm heuristic as a viable method of solution for the inverse problem of the technique of Electrical Capacitance Tomography (ECT), with the purpose of applying to the imaging of multiphase flows. It will also be presented the philosophy of parallel programming that aligns with changes that favor the expansion of collective intelligence of swarm, thus reduces computational processing time, critical parameter in the design of the system of reconstruction of images, necessary for the production of images with higher spatial resolution. An alternative to increase the efficiency of the image reconstruction system is the removal of redundant calculations in the code (appropriation of the sparsity and ordering of the matrices associated with the associated mathematical problem) and the migration of the reconstruction algorithm from a sequenced programming to a parallel programming. It will be presented the physical principles of the ECT, the studies of use of the numerical characteristics of the finite element method (FEM), the heuristic algorithms used in the reconstruction process and the concepts of the main paradigms of parallelization (shared memory, distributed memory) and their software and hardware implementation. The results are encouraging because of the agreement between the images obtained and the simulated tomograms and the significant reduction in their processing time, thus stimulating the continuity of research in this topic for the future. Is found in this thesis all the steps to construct a Beowulf cluster and the serial and parallel codes necessary for implementation the algorithms.

Keywords - Electrical Capacitance Tomography, Inverse Problem, Multiphase Systems, Finite Elements methods, Particle Swarm Optimization.

SUMÁRIO

LISTA DE PSEUDOCÓDIGOS	IV
LISTA DE FIGURAS	V
LISTA DE TABELAS	VII
LISTA DE ABREVIACÕES	VIII
LISTA DE SÍMBOLOS	IX
1. ESCOAMENTO MULTIFÁSICO	
1.1. INTRODUÇÃO	01
1.1.1. Perfis de escoamentos multifásicos	03
1.1.2. Medição de vazão em escoamentos multifásicos	06
1.1.3. Avanços da área de medição multifásica	12
1.2. OBJETIVOS	14
1.2.1. Objetivo geral	14
1.2.2. Objetivo específico	15
1.3. ORGANIZAÇÃO DOS CAPÍTULOS	15
2. TOMOGRAFIA CAPACITIVA ELÉTRICA	
2.1. INTRODUÇÃO	17
2.2. AQUISIÇÃO DO SINAL ELÉTRICO	19
2.3. ELETRÔNICA DE TRANSDUÇÃO	19

2.4. RECONSTRUÇÃO DA IMAGEM	21
2.4.1. Resolução do Problema Direto da TCE	22
2.4.2. Resolução do Problema Inverso da TCE	29
3. SOLUÇÃO DO PROBLEMA DIRETO DA TCE VIA MEF	
3.1. INTRODUÇÃO	33
3.2. UM BREVE HISTÓRICO DO MEF	34
3.3. DESENVOLVIMENTO MATEMÁTICO DO MEF	35
3.4. MEF PELO MÉTODO DE GALERKIN	38
3.4.1. Discretização do domínio em um número finito de elementos	39
3.4.2. Discretização das equações	40
3.4.3. Transladação da modelagem local para a modelagem global	45
3.4.4. Resolução do sistema de equações lineares	48
4. OTIMIZAÇÃO POR ENXAME DE PARTÍCULAS	
4.1. INTRODUÇÃO	50
4.2. ALGORITMO COMPUTACIONAL	56
4.3. PARÂMETROS DO ALGORITMO	59
4.3.1. Tamanho do enxame	59
4.3.2. Número de iterações	59
4.3.3. Coeficientes comportamentais	59
4.4. MELHORIA DA CONVERGÊNCIA DO ALGORITMO	60
4.4.1. Peso de inércia	61
4.4.2. Coeficiente de constrição	62
4.4.3. Condições de fronteira	63
4.5. CRITÉRIOS DE CONVERGÊNCIA	67

5. PROGRAMAÇÃO PARALELA	
5.1. INTRODUÇÃO	70
5.2. ARQUITETURAS PARALELAS	71
5.2.2. Memória distribuída	71
5.2.1. Memória compartilhada	72
5.3. MPI	74
5.4. CLUSTER BEOWULF	75
5.5. MONTAGEM DO CLUSTER PITÁGORAS	77
5.5.1. Configuração de hardware	78
5.5.2. Configuração de software	78
5.5. IMPLEMENTAÇÃO PARALELA	86
6. RESULTADOS	
6.1. INTRODUÇÃO	89
6.2. ESTUDO DE PARÂMETROS PARA O ALGORITMO OEP	91
6.3. ALGORITMOS CONVENCIONAIS VERSUS ALGORITMO OEP	95
6.4. PARALELIZAÇÃO DO ALGORITMO OEP	103
CONCLUSÃO	108
REFERÊNCIAS	110
APÊNDICE A (ROTINA OEP SERIAL)	117
APÊNDICE B (ROTINA OEP PARALELLO)	122
APÊNDICE C (ROTINA TRL)	129
APÊNDICE D (ROTINA TRA)	132
APÊNDICE E (ROTINA TAIL)	136

APÊNDICE F (ROTINAS ASSOCIADAS A RESOLUÇÃO DO PROBLEMA DIRETO)	140
APÊNDICE G (ROTINA DE TESTE DE COMUNICAÇÃO)	149
ANEXO A (ALGORITMOS PROJETIVOS)	150
ANEXO B (FUNÇÕES BÁSICAS DA BIBLIOTECA OPENMPI)	155

LISTA DE PSEUDOCÓDIGOS

Pseudocódigo 4.1 - Algoritmo OEP Original. -----	60
Pseudocódigo 4.2 - Condição de Fronteira Refletivo. -----	65
Pseudocódigo 4.3 - Condição de Fronteira Amortecido. -----	65
Pseudocódigo 4.4 - Condição de Fronteira Absortivo. -----	65
Pseudocódigo 4.5 - Condição de Fronteira Invisível. -----	66
Pseudocódigo 4.6 - Condição de Fronteira Invisível/Refletor. -----	66
Pseudocódigo 4.7 - Condição de Fronteira Invisível/Amortecido. -----	67
Pseudocódigo 5.1 - Algoritmo OEP Paralelo. -----	87

LISTA DE FIGURAS

Figura 1.1 – Tipos de plataforma para exploração em águas profundas. -----	3
Figura 1.2 – Padrões de escoamento em tubulação horizontal. -----	5
Figura 1.3 – Padrões de escoamentos em tubulação vertical. -----	6
Figura 1.4 – Separador de teste. -----	7
Figura 1.5 – Placa de orifício concêntrico com lingueta metálica. -----	8
Figura 1.6 – Medidor baseado em pressão diferencial utilizando um tubo de Venturi. -----	8
Figura 1.7 – Medidor do tipo turbina. -----	9
Figura 1.8 – Medidor Coriolis. -----	9
Figura 1.9 – Medidor de deslocamento positivo. -----	10
Figura 1.10 – Três abordagens típicas de aplicação da técnica ultrassônica. -----	11
Figura 2.1 – Sistema Tomográfico via Capacitância Elétrica. -----	18
Figura 2.2 – Visão do sensor ao redor da tubulação a ser imageada. -----	18
Figura 2.3 – Medida da capacitância C_x baseada no método de carga/descarga. -----	20
Figura 2.4 – Medida da capacitância C_x baseada em corrente alternada. -----	20
Figura 2.5 – Sensor de capacitância com 8 eletrodos. -----	28
Figura 2.6 – Representação do processo de obtenção do sinal nos sensores alocados na periferia do meio a ser imageado. -----	31
Figura 2.7 – Representação do processo de sobreposição algébrica dos resultados individuais de cada sensor para formação da imagem. -----	31
Figura 2.8 – Diagrama esquemático do procedimento utilizado pelos algoritmos iterativos durante o processo de reconstrução. -----	32
Figura 3.1 – Representação esquemática do procedimento de solução via MEF. -----	36
Figura 3.2 – Corte da Secção Transversal do Sensor. -----	40
Figura 3.3 – Discretização do Domínio do Sensor. -----	40
Figura 3.4 – Elemento Triangular Linear. -----	42
Figura 3.5 – Representação gráfica das funções interpoladoras sobre cada nó. -----	43

Figura 3.6 – Discretização do domínio com 3 elementos triangulares linear. -----	45
Figura 4.1 – Número de publicações da técnica OEP por país. -----	55
Figura 4.2 – Atualização de uma partícula. -----	57
Figura 5.1 – Configuração esquemática de um sistema de memória compartilhada. -----	73
Figura 5.2 – Configuração esquemática de um sistema de memória distribuída. -----	73
Figura 5.3 – Configuração típica de um cluster Beowulf. -----	75
Figura 5.4 – Cluster Pitágoras. -----	77
Figura 5.5 – Máquina utilizada. -----	78
Figura 5.6 – Roteador utilizado. -----	78
Figura 5.7 – Estrutura da rede utilizada. -----	80
Figura 5.8 – Tela de confirmação de instalação do iptables-persistent. -----	82
Figura 6.1 – Discretização do domínio (a) com 128 elementos e (b) com 32 elementos na parte interior da tubulação. -----	91
Figura 6.2 – Distribuições de dielétricos simuladas. -----	96
Figura 6.3 – Imagens reconstruídas a partir dos valores de capacitância obtidas das distribuições de dielétricos simuladas sem adição de ruído numérico. -----	98
Figura 6.4 – Imagens reconstruídas a partir dos valores de capacitância obtidas das distribuições de dielétricos simuladas com adição de 1% de ruído numérico. -----	100
Figura 6.5 – Imagens reconstruídas a partir dos valores de capacitância obtidas das distribuições de dielétricos simuladas com adição de 3% de ruído numérico. -----	102
Figura 6.6 – Speedup obtido para uma configuração de bolhas. -----	105
Figura 6.7 – Eficiência paralela obtida para uma configuração de bolhas. -----	106
Figura 6.8 – Overhead obtido para uma configuração de bolhas. -----	106

LISTA DE TABELAS

Tabela 2.1 – Características, vantagens e desvantagens dos circuitos eletrônicos. -----	21
Tabela 3.1 – Associação do elemento com sua numeração local e global. -----	46
Tabela 3.2 – Mapeamento dos componentes locais nos componentes globais da matriz de capacitância relacionadas ao elemento 3. -----	46
Tabela 4.1 – Evolução temporal do número de publicação anual a respeito da técnica de OEP pelas principais bases de dados listadas no portal CAPES. -----	53
Tabela 4.2 – Número de publicações por periódico. -----	54
Tabela 4.3 – Distribuição de publicações da técnica OEP por área do conhecimento. -----	55
Tabela 6.1 – Valores da permissividade relativa dos materiais empregados no sensor. ----	90
Tabela 6.2 – Dimensões físicas do tubo simulado. -----	90
Tabela 6.3 – Solução do algoritmo OEP utilizando um mecanismo auto adaptativo para o número de partículas no enxame. -----	92
Tabela 6.4 – Solução do algoritmo OEP utilizando diferentes tipos de condições de fronteiras. -----	93
Tabela 6.5 – Solução do algoritmo OEP em relação as condições de parada. -----	94
Tabela 6.6 – Valores de erro de permissividade relativa referente ao processo de reconstrução de imagens sem ruído numérico. -----	97
Tabela 6.7 – Valores do tempo médio de processamento referente ao processo de reconstrução de imagens sem ruído numérico. -----	97
Tabela 6.8 – Valores de erro de permissividade relativa referente ao processo de reconstrução de imagens para um ruído numérico de 1%. -----	99
Tabela 6.9 – Valores do tempo médio de processamento referente ao processo de reconstrução de imagens para um ruído numérico de 1%. -----	99
Tabela 6.10 – Valores de erro de permissividade relativa referente ao processo de reconstrução de imagens para um ruído numérico de 3% .-----	101

Tabela 6.11 – Valores do tempo médio de processamento referente ao processo de reconstrução de imagens para um ruído numérico de 3%. -----	101
Tabela 6.12 – Valores das métricas de caracterização do sistema paralelo durante o processo de obtenção da imagem de um perfil de bolhas com um ruído numérico de 3%. -----	105
Tabela 6.13 – Valores obtidos para o speedup teórico de um sistema paralelo a partir da lei de Amdahl. -----	107

LISTA DE ABREVIACOES

TCE – Tomografia via Capacitância Elétrica
MDF – Método de Diferenças Finitas
MEF – Métodos de Elementos Finitos
MEC – Método de Elemento de Contorno
EDP – Equações Diferenciais Parciais
NASA – National Aeronautics and Space Administration
MGC – Método do Gradiente Conjugado
MPI – Message Passing Interface
API – Applications Programming Interface
RAM – Random Access Memory
IP – Internet Protocol
WAN – Wide Area Network
CP – Cluster Pitágoras
OEP – Otimização por Enxame de Partículas
TRL – Tomografia por Retroprojeção Linear
TRA – Tomografia por Reconstrução Algébrica
TAIL – Tomografia por Algoritmo Iterativo de Landweber
CFR – Condição de Fronteira Refletivo
CFAM – Condição de Fronteira Amortecida
CFA – Condição de Fronteira Absortiva
CFI – Condição de Fronteira Invisível
CFIR – Condição de Fronteira Invisível/Refletor
CFIA – Condição de Fronteira Invisível/Amortecido

LISTA DE SÍMBOLOS

- \vec{E} – Vetor Campo Elétrico
 \vec{D} – Vetor Deslocamento Elétrico
 \vec{J} – Vetor Densidade de Corrente Elétrica
 \vec{B} – Vetor Indução Magnética
 \vec{H} – Vetor Campo Magnético
 φ – Potencial Escalar
 \vec{A} – Potencial Vetorial
 $\vec{\nabla}$ – Operador Gradiente
 $\vec{\nabla} \cdot$ – Operador Divergente
 $\vec{\nabla} \times$ – Operador Rotacional
 ρ – Densidade de Carga
 μ – Permeabilidade Magnética
 σ – Condutividade Elétrica
 ε – Permissividade Elétrica do Meio
 ε_0 – Permissividade do Vácuo
 ε_r – Permissividade Elétrica Relativa do Meio
 Ω – Dominio
 Γ – Contorno
 φ^n – Potencial Aproximador
 φ_j – Potencial Admissível
 a_j – Coeficientes a Ajustar
 $D[]$ – Operador Diferencial
 $\{\varphi_j\}$ – Subespaço Vetorial
 ψ – Energia Potencial

c_{ij}^e –

φ_{ei} – Potencial nos Nós na Cada Elemento

$[C_{ij}^e]_{LOCAL}$ – Matriz de Capacitância na Escala do Elemento

$[C_{ij}^e]_{GLOBAL}$ – Matriz de Capacitância na Escala da Malha

ω – Peso de Inércia

ω_{max} – Peso de Inércia Máxima

ω_{min} – Peso de Inércia Mínima

c_1 – Coeficiente Comportamental Cognitivo

c_2 – Coeficiente Comportamental Social

r_1, r_2 e κ – Variáveis Aleatórias uniformemente distribuídas entre 0 e 1

χ – Coeficiente de Constrição

x_{max} – Posição Máxima

x_{min} – Posição Mínima

x_i^k – Posição da partícula i na iteração k

v_i^k – Velocidade da partícula i na iteração k

p_i^k – Melhor posição local da partícula i até a iteração k

g_{best}^k – Melhor posição global do enxame até a iteração k

Δ^k – Indicador de Convergência de Aptidão

S^k – Indicador de Convergência de Agregação

$\Delta\varepsilon$ – Erro Quadrático Normalizado

$\delta\varepsilon$ – Erro Quadrático Normalizado

R_{xy} – Coeficiente de Correlação

$S(p)$ – Speedup

$E(p)$ – Eficiência Paralela

$V(p)$ – Overhead

μ – Valor médio

σ_s – Desvio Padrão da amostra

max – Valor máximo

min – Valor mínimo

$par\%$ – Fração percentual do código que é paralelizável

CAPÍTULO I

ESCOAMENTO MULTIFÁSICO

1.1 – INTRODUÇÃO

Segundo Ferreira (1999) o conceito de escoamento é definido como sendo o ato de fazer ou deixar correr ou escorrer um fluido em um duto ou canal, e para Paladino (2005) um sistema pode ser definido como sendo multifásico quando coexistem dois ou mais meios imiscíveis separados por uma interface, podendo ser esta conexa ou desconexa ou uma combinação de ambos os casos onde a mesma fase pode aparecer em forma contínua ou dispersa. Desta forma, um escoamento multifásico pode ser definido como dois ou mais fluidos com propriedades diferentes e imiscíveis, escoando simultaneamente em uma dada tubulação.

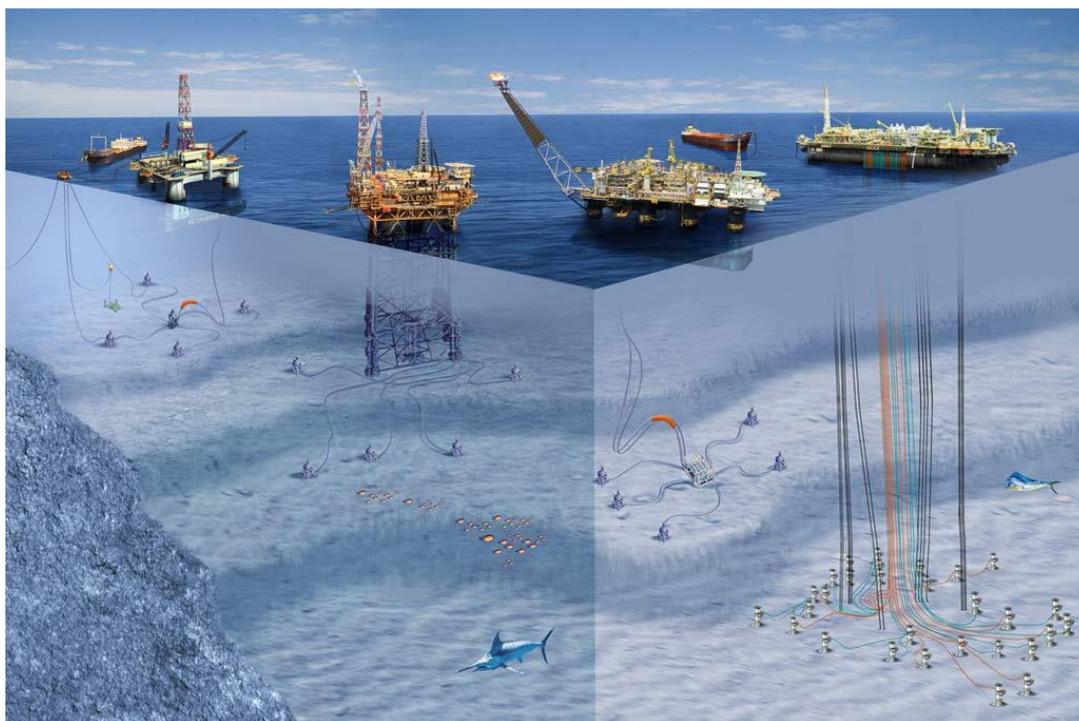
Escoamentos multifásicos podem ser encontrados de maneira bem comum na natureza, onde por exemplo, tem-se rios de água doce que transportam sedimentos de um extremo a outro de sua extensão devido a força da sua correnteza, nuvens que transportam vapor de água e gases de um ponto a outro do globo, vulcões que transportam magna do interior da terra para a superfície, sistemas cardiocirculatórios de animais e etc. Na indústria também é bastante comum a presença de escoamentos multifásicos (ANDRADE, 2016), seja devido a processos que envolvam misturas químicas como no caso da indústria alimentícia, sistemas de transporte de calor como na indústria de energia termoelétrica ou no transporte de substâncias em tubulações como no caso da indústria de petróleo e gás.

CAPÍTULO I

Em particular, na indústria de petróleo, a ocorrência de escoamentos multifásicos pode ser evidenciada em dutos verticais, horizontais e inclinados que ligam as unidades de extração do poço reservatório até às unidades de separação, conforme a Figura 1.1, passando pelas unidades de refino de óleo e gás chegando até as unidades de transferência de custódia.

Com o aumento da complexidade das plantas industriais do setor de petróleo e gás, é observado a necessidade de adoção de ferramentas que possam otimizar as tomadas de decisão a respeito dos processos envolvendo a extração, transporte e refino do petróleo. O correto entendimento de como o petróleo se comporta durante o processo de escoamento na sua cadeia produtiva é de fundamental importância para as questões relacionadas com o retorno econômico desta área, tais como, a determinação da variação de pressão nos dutos, a determinação da fração de óleo em outros meios, a detecção de vazamentos, a medição e fiscalização das vazões volumétricas de óleo entregue e etc.

Figura 1.1 – Tipos de plataforma para exploração em águas profundas.



Legenda: Da esquerda para a direita plataformas fixas, plataformas semissubmersíveis, plataformas flutuantes que retiram e armazenam o produto e plataformas auto elevatórias.

Fonte: Agência Petrobras

Um dos processos mais comuns em uma linha de produção petrolífera, seja em terra (*onshore*) ou no mar (*offshore*) é o de medição de fração de material em trânsito no

CAPÍTULO I

escoamento, que pode variar de acordo com o perfil das fases presentes no meio. O perfil de escoamento pode adquirir diversas configurações, denominados padrões de escoamento.

Segundo Barbosa Jr. (2011), os padrões de escoamento advêm de uma condição de equilíbrio entre forças (de campo e de superfície) locais no interior de cada fase e nas interfaces entre elas que dependem em maior ou menor intensidade, de parâmetros associados a geometria do duto (comprimento, diâmetro, rugosidade e inclinação), das condições de operação do escoamento (pressão, temperatura e ação gravitacional) e das propriedades físico-químicas dos fluidos (densidade, viscosidade, tensão superficial, pressão de vaporização e solubilidade) sendo que estas características podem variar significativamente no tempo e espaço, assim como os fenômenos interfaciais entre as fases.

A transição de um padrão de escoamento ocorre quando o regime existente se torna instável devido a uma mudança brusca ou gradual de um dos parâmetros mencionados anteriormente. Uma nova topologia entre as fases emerge a partir do reequilíbrio das forças locais.

O conhecimento dos tipos de padrões e como estes padrões podem ser influenciados é de fundamental importância para o planejamento e dimensionamento adequado de dutos, equipamentos e técnicas utilizadas na cadeia produtiva de petróleo e gás, não apenas pela capacidade de predição a priori das configurações de escoamento, mas também pela possibilidade de uma eventual necessidade de transição de regime a partir da manipulação adequada dessas propriedades, afim de se conformar a um padrão desejado, permitindo-se assim encontrar soluções mais adequadas para a redução dos custos, o aumento da produtividade, a melhoria na qualidade de produção e a minimização de desperdícios nesta área.

1.1.1 – Perfis de escoamentos multifásicos

A presença de escoamentos multifásicos é recorrente no setor de petróleo e gás nas diversas instancias de beneficiamento dos insumos produzidos. Os padrões de escoamento determinam a magnitude dos fenômenos de transporte de interesse, além da magnitude de variáveis específicas, por exemplo: a quantidade de movimento (perda de carga), a transferência de calor e de massa do fluido.

Devido à importância e a complexidade do tema, foram desenvolvidos diversos estudos com o intuito de identificar e classificar os padrões de escoamento a partir de

CAPÍTULO I

especificações das condições operacionais, das características do sistema e das propriedades dos fluidos em mistura.

A partir de 1960, diversas abordagens fenomenológicas foram propostas para determinação do regime de escoamento multifásico, tais como as desenvolvidas por Zuber e Findlay (1965), Taitel e Dukler (1976), Oliemans e Pots (1986).

Os padrões de escoamentos são determinados utilizando métodos analíticos, ou de modelos fenomenológicos de classificação que preveem as configurações de escoamento a partir de relações entre as seguintes variáveis: vazão mássica, densidade e viscosidade da fração gasosa e líquida do fluido em escoamento como também o diâmetro e rugosidade da tubulação e etc.

1.1.1.1 – Escoamentos horizontais

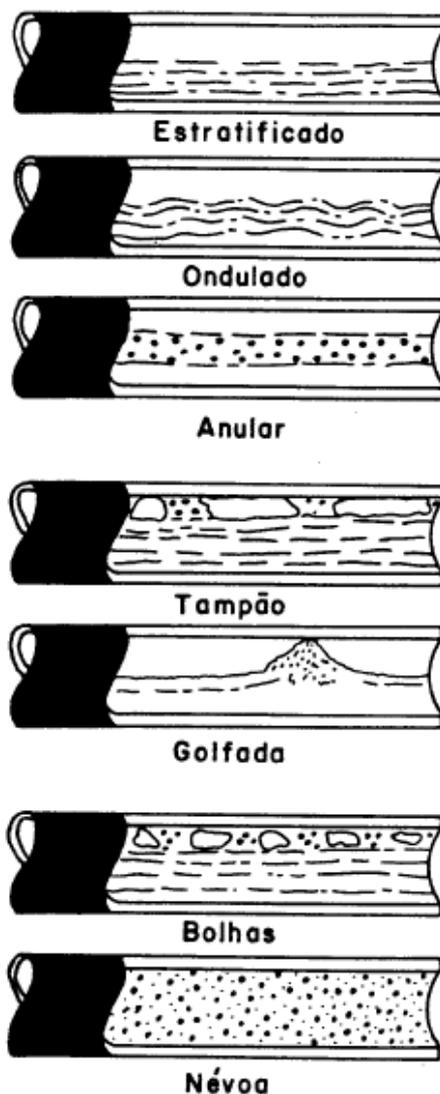
Na Figura 1.2, é possível observar alguns exemplos de padrões de escoamento, no sentido horizontal, representando neste caso, exemplos de escoamentos bifásicos do tipo líquido-gás.

Abaixo temos a descrição dos tipos de escoamentos bifásicos horizontais conforme Silva *et al.* (2000, pp. 46):

- *Estratificado: o escoamento estratificado se caracteriza pela separação gravitacional total das fases líquida e gasosa com escoamento contínuo de cada fase;*
- *Ondulado: com o aumento da velocidade da fase gasosa, no escoamento estratificado, ondas são formadas na interface de separação entre a fase líquida e a fase gás, gerando o regime ondulado;*
- *Anular: o escoamento anular ocorre com altas vazões de gás. O líquido escoar na superfície interna da tubulação, formando um filme com bolhas dispersas, e o gás escoar no centro da tubulação, carregando líquido atomizado. Devido ao efeito gravitacional, a espessura do filme formado pela fase líquida diminui quando se caminha para o topo da tubulação.*
- *Tampão: com o aumento da velocidade da fase gasosa no escoamento em bolhas, há o coalescimento, formando bolhas alongadas em forma de balas, que tendem a escoar no topo da tubulação;*
- *Golfada: aumentando ainda mais a velocidade do gás no escoamento ondulado, as ondas passam a ocupar, em alguns casos, toda a seção transversal da tubulação formando golfadas de gás e/ou líquido, que são rapidamente deslocadas dentro da tubulação. Golfadas de gás e líquido se sucedem na tubulação com a de líquido carregando pequenas bolhas dispersas;*
- *Bolhas ou borbulhante: o escoamento em bolhas, devido ao efeito gravitacional (a fase gasosa em forma de bolhas discretas), tende a se dispersar no topo da tubulação com a fase líquida contínua;*
- *Névoa: este padrão de escoamento é caracterizado por velocidades superficiais de gás e líquido muito altas. Todo líquido encontra-se disperso no núcleo gasoso e as gotículas formadas pelo líquido viajam à mesma velocidade superficial do gás.*

CAPÍTULO I

Figura 1.2 - Padrões de escoamento em tubulação horizontal



Fonte: Silva *et al.* (2000)

1.1.1.2 – Escoamentos verticais

Da mesma forma que no escoamento horizontal, pode-se estabelecer padrões de escoamentos em tubulações dispostas na direção vertical para o escoamento bifásico líquido e gás, representados na Figura 1.3.

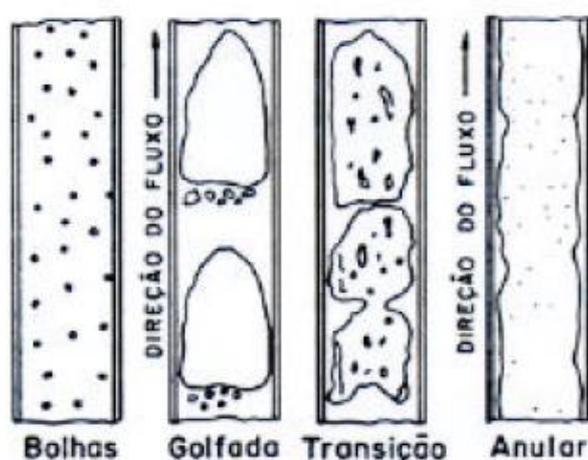
Abaixo temos a descrição dos tipos de escoamentos bifásicos verticais conforme Silva *et al.* (2000, pp. 47):

- *Bolhas ou bolhas discretas: no escoamento em bolhas, a fase gasosa se encontra dispersa na fase líquida em bolhas discretas, sendo a fase líquida contínua. Esse escoamento ocorre tipicamente para baixas velocidades superficiais de gás;*

CAPÍTULO I

- *Golfadas: aumentando a velocidade da fase gasosa, as bolhas coalescem e o diâmetro destas novas bolhas atingem dimensões similares com a da tubulação. Quando isto ocorre, formam-se bolhas grandes em formato de bala, também denominadas bolhas de Taylor, com isto, golfadas de gás e líquido se sucedem na tubulação com a golfada de líquido apresentando pequenas bolhas dispersas;*
- *Transição: com as velocidades de gás e líquido maiores do que no caso anterior, a quebra das bolhas do escoamento em golfadas conduz a um padrão instável e desordenado, onde existe um movimento oscilatório de líquido para cima e para baixo na tubulação;*
- *Anular: o líquido escoar na periferia do duto formando um filme com bolhas dispersas, e o gás escoar no centro da tubulação, carregando gotículas de líquido.*

Figura 1.3 - Padrões de escoamentos em tubulação vertical.



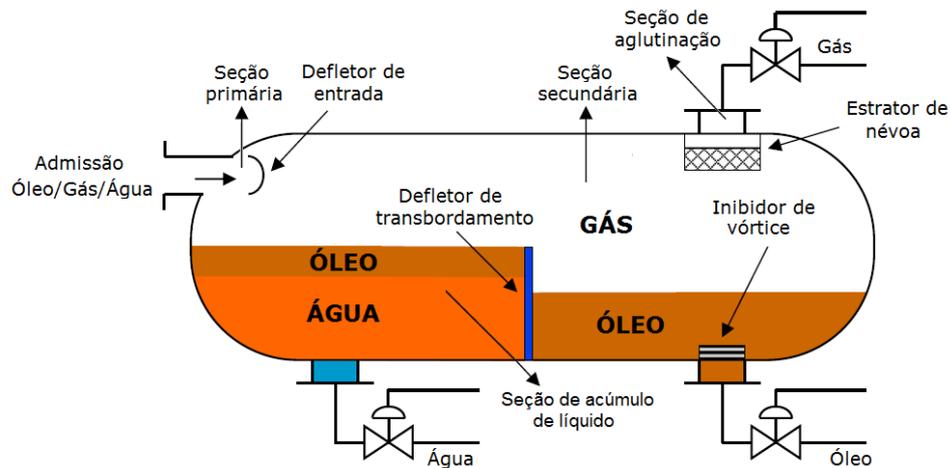
Fonte: Silva *et al.* (2000)

1.1.2 – Medição de vazão em escoamentos multifásicos

Um dos principais procedimentos empregados no processo de medição de vazão é a utilização de um equipamento na linha de produção chamado de separador de teste. Neste equipamento, a mistura multifásica é admitida a montante, onde as fases água, óleo, gás e outros componentes são separadas por meio de diversos dispositivos (placas defletoras de entrada, quebradores de espuma, de ondas e de vórtice, extrator de névoa e etc) que usam como principal mecanismo de separação a segregação das fases devido a ação da força gravitacional e a diferença de densidade entre os elementos componentes da mistura.

Silva *et al.* (2007) apresenta uma representação didática de um separador de teste típico, divido-o em quatro seções: (a) seção de separação primária, (b) seção de acumulação (ou de coleta de líquido), (c) seção de separação secundária (ou de decantação) e (d) seção de aglutinação. Este equipamento pode ser visualizado em detalhes na Figura 1.4.

Figura 1.4 - Separador de teste.



Fonte: Adaptado de Crabtree. (2009)

A seção primária, localiza-se na entrada do vaso, nela o fluido (água, óleo e gás) é admitido com uma velocidade considerável, chocando-se com os defletores posicionados na parte superior do vaso, favorecendo a dissociação da parcela gasosa do fluido da parte líquida, isto é proporcionado pela mudança brusca de velocidade. A maior parte da fração líquida do fluido fica retida na seção de acúmulo de líquido, isto devido ao fato das gotículas líquidas serem mais pesadas se desprendendo da corrente de gás e se acumulando no fundo do separador. Nesta parte do separador, a parcela líquida repousa por um tempo de 3 a 4 minutos, tempo suficiente para permitir a separação de grande parte da água do óleo devido ao processo de decantação. Na seção secundária, as gotículas maiores de óleo, oriundas da fase gasosa, são separadas por condensação do vapor saturado. As gotículas de líquido remanescentes na fase gasosa são arrastadas pela corrente e são retidas em sua maioria na seção de aglutinação (localizada geralmente próxima a saída da parte gasosa), sendo removidas do fluxo gasoso através de filtros porosos que favorecem a coalescência e decantação das gotas.

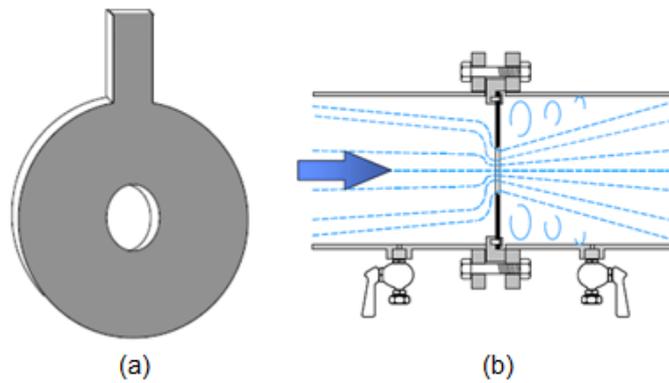
A pressão no separador é mantida por um controlador que atua regulando o fluxo de saída do gás na parte superior. O líquido separado deixa o vaso através de uma válvula de descarga, cuja abertura ou fechamento é efetuado através de um controlador de nível.

A jusante do separador, são empregados medidores monofásicos para se determinar a quantidade de cada parcela de material no escoamento (THORN *et al.*, 2013). A seguir, são descritos os principais dispositivos utilizados para este fim:

CAPÍTULO I

- Medidores baseados em pressão diferencial - através de uma redução da seção de passagem, aumenta-se a velocidade do fluido, provocando uma diminuição da pressão. A velocidade média do escoamento é uma função da queda de pressão (PALADINO, 2005). Os principais tipos de medidores são visualizados nas figuras 1.5 no caso de medidas de fluxos gasosos e figura 1.6 no caso de fluxos de líquidos.

Figura 1.5 - Placa de orifício concêntrico com lingueta metálica



Legenda: (a) visão espacial, (b) placa instalada entre flanges adjacentes

Fonte: Fonte: Adaptado de Crabtree (2009)

Figura 1.6 - Medidor baseado em pressão diferencial utilizando um tubo de Venturi.

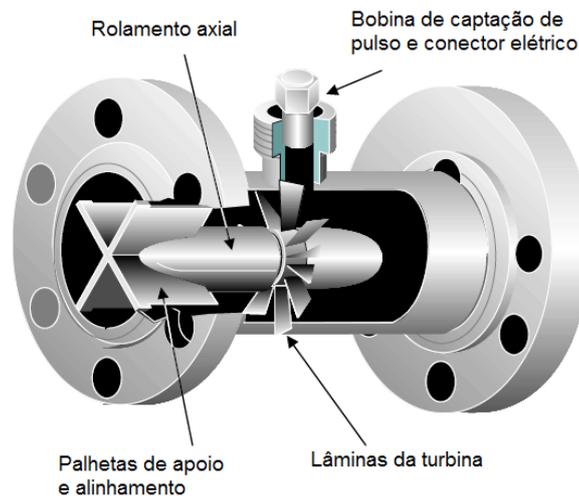


Fonte: Paladino (2005)

CAPÍTULO I

- Medidor do tipo turbina – uma pequena turbina é impulsionada pela corrente fluida e a velocidade de rotação da mesma será proporcional à vazão mássica circulante (PALADINO, 2005). O princípio de funcionamento deste tipo de dispositivo pode ser visualizado na figura 1.7.

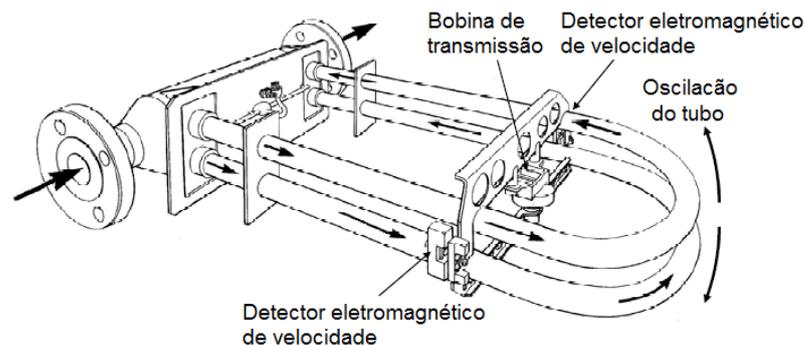
Figura 1.7 - Medidor do tipo turbina.



Fonte: Adaptado de Crabtree (2009)

- Medidor Coriolis - Parte do fluido é dividida em dois tubos iguais em formato de U com diâmetros inferiores ao da tubulação escoamento. Estes tubos vibram devido a uma fonte externa (geralmente obtida por uma ação magnética). Devido à vibração, o fluido ao passar pelos condutos sofre a atuação das forças de Coriolis, o que gera uma deformação elástica nos condutos. A partir destas deformações é possível inferir a vazão mássica (CRABTREE, 2009). A descrição deste tipo de dispositivo pode ser visualizada na figura 1.8.

Figura 1.8 - Medidor Coriolis



Fonte: Adaptado de Crabtree (2009)

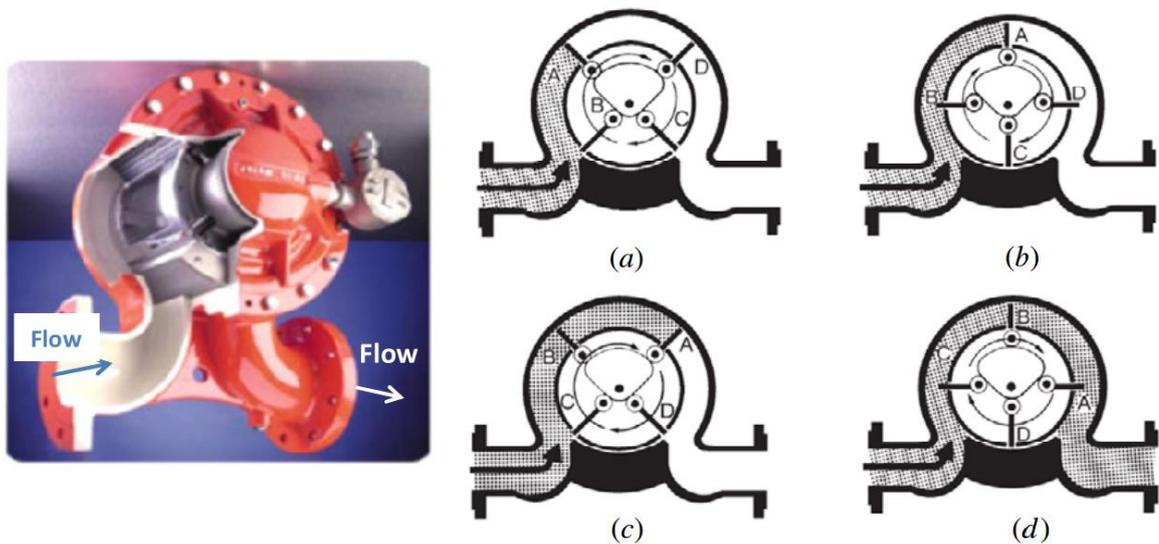
CAPÍTULO I

- Medidor por deslocamento positivo - Existem diversos modelos de medidores de deslocamento positivo, mas todos operam segundo o mesmo princípio. Utilizando-se uma câmara rotativa de volume fixo, o escoamento é discretizado e transportado através de cavidades com volume conhecido, conforme pode ser visto na figura 1.9.

O processo de medição ocorre baseado na contagem da quantidade de volumes transportados no interior do sistema medidor utilizando o seguinte procedimento: a palheta A fecha a câmara e a palheta B permite que o fluido flua, Figura 1.9 (a). Então, palheta A move-se para a posição vertical, Figura 1.9 (b), mas o resto das lâminas não fecham o fluido até B ocupar a posição inicial de A, Figura 1.9 (c). Neste momento, há um volume definido que é transferido para o fluxo quando B é vertical, Figura 1.9 (d).

Totalizando o número de cavidades é possível obter o volume total de fluido que atravessou o medidor em uma determinada fração de tempo (CRABTREE, 2009).

Figura 1.9 - Medidor de deslocamento positivo.



Legenda: Câmara rotativa de volume fixo, representado as etapas de discretização e contagem da quantidade de volumes fixos transportados.

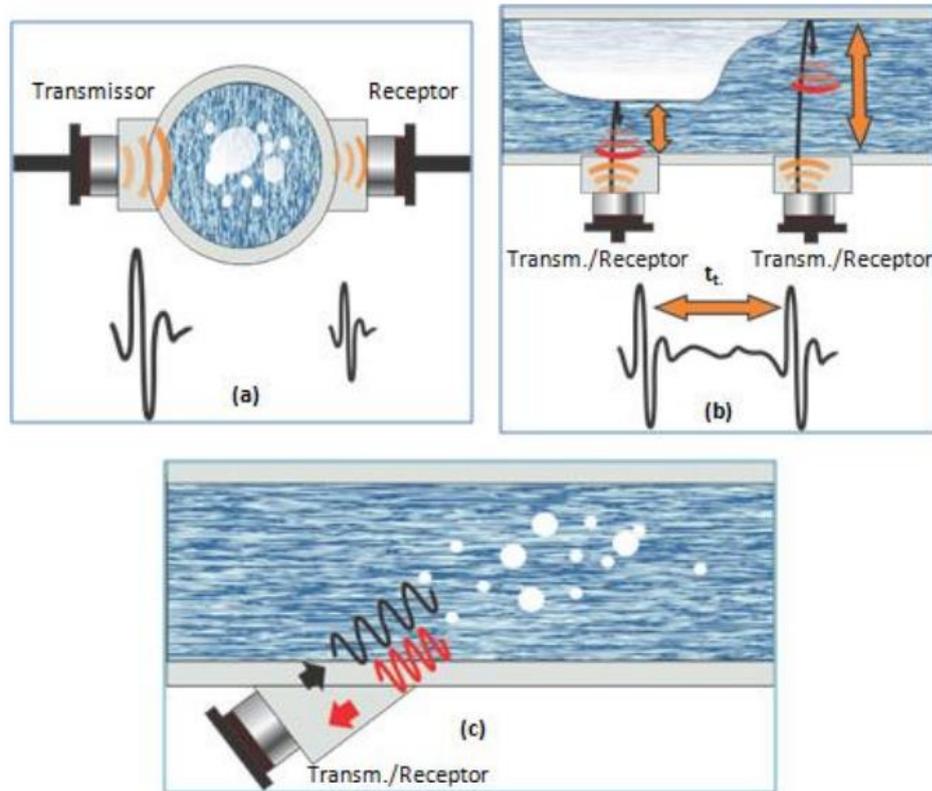
Fonte: Adaptado de Garcia-Berrocal *et al.* (2013)

- Medidores ultrassônicos - A técnica de ultrassom baseia-se no transporte de uma onda sonora através de um fluido, conforme Figura 1.10. No caso de um escoamento bifásico gás-líquido, pode-se operar o ultrassom no princípio de tempo de percurso,

CAPÍTULO I

obtendo-se a espessura de um filme de líquido, ou no princípio o princípio de atenuação da onda transmitida, obtendo-se a fração de vazio média ao longo do diâmetro da tubulação (OLIVEIRA, 2007).

Figura 1.10 - Três abordagens típicas de aplicação da técnica ultrassônica.



Legenda: Em (a) a técnica de atenuação, (b) tempo de trânsito e (c) doppler pulsado.

Fonte: Retirado de Coutinho (2014)

Devido ao fato de que o processo de separação por ação gravitacional não consegue separar perfeitamente os componentes da mistura multifásica, é comum se encontrar algum óleo permanecendo na água, alguma água permanecendo no óleo, algum gás permanecendo no líquido e algum líquido permanecendo no gás. Estas condições causam erros nos instrumentos convencionais empregados como medidores de vazão na saída do separador, pois estes medidores são desenvolvidos para medir unicamente escoamentos monofásicos (ATKINSON, *et al.*, 2004).

Além da limitação mencionada anteriormente, a abordagem convencional possui algumas limitações óbvias, tais como a necessidade de equipamentos que possam suportar

CAPÍTULO I

pressões que podem variar de quase 0 a 2000 bar e de temperatura que podem variar de -40 a 200 °C, implicando no uso de dispositivos volumosos com alto custos de instalação e manutenção especializada. Vale ressaltar que além de todos estes inconvenientes o processo de medição convencional age de maneira intrusiva e invasiva ao escoamento.

Devido a estes problemas, a concepção de um método simples e direto que seja capaz de medir cada fração de componente na mistura diretamente sem passar por uma fase de separação é extremamente desejada.

1.1.3 – Avanços da área de medição multifásica

Segundo Paladino (2005), medição multifásica é a medição da vazão de duas ou mais fases sem a separação prévia das mesmas antes de entrar no medidor. Por outro lado, a medição sem perturbação do escoamento pode-se tornar complexa, dependendo do padrão de escoamento encontrado.

Diversas pesquisas vêm sendo realizadas desde meados de 1940 com o objetivo de desenvolver métodos para previsão das características descritoras dos padrões de escoamento no interior de tubulações com o propósito de influenciar estes perfis para maximizar a produção de óleo e gás (PACHECO, 2007). Ao longo dos anos, três abordagens foram sendo desenvolvidas com o propósito de servir de amparo a previsão do comportamento de escoamentos multifásicos: as correlações empíricas, os modelos mecanicistas e mais recentemente os modelos de fluidodinâmica computacional.

A metodologia de correlações empíricas, consiste na observação de regras causais que visam determinar padrões macroscópicos do escoamento por meio da análise de medidas experimentais.

Os métodos mecanicistas visam o desenvolvimento de soluções analíticas descritivas do escoamento a partir do relacionamento apropriado dos conceitos físicos oriundos da análise dos mecanismos de transferência de calor e massa dos fluidos em escoamento.

A junção dos modelos empíricos, dos modelos mecanicistas conjuntamente com o advento do computador pessoal em meados de 1950 (FORTUNA, 2000), cria uma nova área de estudo da área de escoamentos multifásicos, a fluidodinâmica computacional, onde o comportamento macroscópico do escoamento pode ser previsto utilizando modelos mais realistas, a partir da resolução numérica de modelos microscópicos baseados nos princípios

CAPÍTULO I

de conservação de massa, de energia e da quantidade de movimento das partículas constituintes do meio.

No entanto, segundo Faccini (2008), os modelos físicos ainda requerem informações acerca das características locais e transientes dos escoamentos. Portanto, técnicas de determinação dos perfis de escoamentos com grande confiabilidade continuam sendo necessárias, tanto na pesquisa quanto na indústria, a fim de se ter um projeto de engenharia racional e seguro.

Dentre as técnicas disponíveis, as de visualização tomográfica tem se consolidado atraindo o interesse da indústria e da comunidade científica (PRADEEP *et al.*, 2015). A tomografia de escoamento de meios multifásicos também chamada de tomografia de processos tem por finalidade revelar as estruturas internas de um escoamento de maneira não invasiva e não intrusiva (ISMAIL *et al.*, 2005).

Em diversas situações é preponderante o conhecimento em detalhes do comportamento do escoamento interno no interior de dutos em um dado processo industrial. Muitas vezes, isto só é conseguido utilizando o processo de instrumentação tomográfica, pois a instrumentação de medição convencional pode ser inadequada para este fim, devido a exposição do sensor a condições internas adversas ou devido ao fato de que a sua presença pode perturbar a dinâmica do fluxo.

As principais características dos sensores de medição de processos devem ser:

- a) robustez, pois existe a possibilidade de ser apresentado ao sensor um ambiente agressivo;
- b) ser não invasivos e não intrusivo, para não interferir no padrão ao qual se quer visualizar;
- c) rápido, para que possa ser utilizado em situações de identificação de escoamentos transientes;
- d) simples, para possa ser operado sem a necessidade de muitos protocolos de manipulação;
- e) barato, custos reduzidos de confecção, manutenção e substituição (quando necessário).

A adoção de técnicas tomográficas, modifica radicalmente o projeto e a operação de processos que envolvem misturas de meios multifásicos, permitindo que as fronteiras dos diferentes componentes possam ser visualizadas em tempo real.

CAPÍTULO I

Informações sobre o regime de escoamento, o vetor velocidade e concentrações dos componentes em vasos e dutos serão determinados a partir da visualização de imagens. Os dados das imagens podem ser analisados de maneira qualitativa, indicando de maneira subjetiva o bom andamento do processo, sugerindo processos preventivos tais como retirada de depósitos de parafinas no interior da tubulação (WEBER, 2013) ou de maneira quantitativa, em uso para acionar estratégias de controle eficiente das bombas de sucção do petróleo em extração a partir de um processo de realimentação em malha fechada (ISMAIL *et al.*, 2005).

Devido as características do tipo de escoamento em estudo, dentre os métodos tomográficos mais indicados para este propósito encontramos a Tomografia via Capacitância Elétrica (TCE). A TCE baseia-se na resposta causal não linear de uma amostra multifásica a campos elétricos aplicados na periferia de uma dada secção transversal de uma tubulação, com a qual é possível inferir a distribuição geométrica de maneira espacial e temporal dos dielétricos que constituem o meio. A teoria e a instrumentação relacionada a TCE são bem sedimentadas no meio acadêmico e estão baseadas na solução das equações de Maxwell para meios dielétricos e no uso de sensores capacitivos aplicados a construção de imagens (AZEVEDO FILHO, 2002).

A técnica de TCE é relativamente nova no cenário industrial (meados da década de 80), onde ganha um interesse mais pronunciado da comunidade científica – industrial durante o decorrer da década de 90 com a criação do “European Concerted Action on Process Tomography” (ECAPT), supervisionado pela Universidade de Manchester (UMIST). Observando-se a partir daí diversos trabalhos voltados para o aprimoramento da técnica em termos de desenvolvimento da eletrônica de transdução, sistema de aquisição e na implementação de algoritmos de reconstrução de forma eficiente (BECK, 1995).

1.2 - OBJETIVOS

1.2.1 – Objetivo geral

O presente trabalho tem como objetivo geral resolver o problema direto e inverso associado a TCE utilizando técnicas numérico-computacionais que possam diminuir o tempo total de processamento das imagens obtidas pelo método.

CAPÍTULO I

1.2.2 – Objetivos específicos

Como objetivos específicos teremos:

- Analisar a aplicabilidade do algoritmo heurístico de Otimização por Enxame de Partículas (OEP) como algoritmo inverso aplicado a TCE;
- Aplicação do paradigma de programação paralela na solução do algoritmo OEP para reduzir o tempo computacional de reconstrução da imagem da TCE.
- Construção de uma arquitetura de hardware para computação de alto desempenho (Cluster Beowulf) para implementação dos algoritmos propostos.

1.3 - ORGANIZAÇÃO DOS CAPÍTULOS

No capítulo 2 será realizado um levantamento bibliográfico a respeito da TCE, onde procura-se discorrer a respeito do estado da arte da técnica aplicada ao estudo de meios multifásicos, no qual se envolve as etapas de descrição do sensor, aquisição e condicionamento do sinal de capacitância, multiplexagem e reconstrução da imagem a partir da utilização de algoritmos inversos.

Será apresentado a modelagem matemática do sensor e para o caso particular de perfis escoamento com simetria core-anular, uma solução analítica capaz de servir de comparação para a solução numérica necessária para o sensor. Será também apresentado uma revisão bibliográfica a respeito dos primeiros algoritmos inversos aplicados a TCE a partir de métodos projetivos que se utilizam de uma aproximação linear para a relação entre a capacitância mensurada no sensor e a distribuição de permissividade no interior da tubulação e os modernos algoritmos bioinspirados que buscam a solução do problema inverso aplicando processos de inteligência artificial resgatando a não linearidade entre a capacitância e a distribuição de permissividade na seção da tubulação em estudo.

No capítulo 3, serão apresentados os fundamentos da técnica do Método dos Elementos Finitos (MEF) e como ela pode ser aplicada a solução numérica do problema eletrostático que foi modelado no capítulo 2, serão discutidos os princípios e os passos para tal implementação, sendo discutido desde o processo de discretização do domínio, da derivação da equação governante que descreve o campo eletromagnético no interior do sensor, montagem da matriz global oriunda do processo de linearização da equação campo eletromagnético em cada elemento discretizado do domínio e por fim, o processo de

CAPÍTULO I

resolução do sistema linear obtido utilizando estratégias numéricas apropriadas para se aproveitar da estrutura esparsa do sistema linear obtido.

No capítulo 4, a OEP é apresentada como uma técnica robusta e eficiente para determinação da distribuição de dielétricos a partir dos valores de capacitância mensurados, também serão apresentados os problemas em relação ao custo computacional para realizar este fim.

Neste capítulo, serão discutidos os vários pontos necessários que definem o algoritmo computacional e como amplificar as suas potencialidades em relação a determinação da solução requerida e minimizar o máximo possível a problemática de um alto custo computacional. Inicialmente será realizado uma apresentação ontológica da técnica computacional escolhida para servir como algoritmo inverso da TCE empregada nesta tese, seguido de uma explanação da estrutura do algoritmo computacional com a definição de cada parâmetro de ajuste que define a convergência do mesmo, logo após exploraremos variantes do algoritmo original que visam ampliar a “inteligência do exame” reduzindo assim a necessidade de uma longa análise do domínio do problema em busca de uma solução apropriada o último passo a ser discutido é a paralelização do algoritmo OEP para ser executado em várias máquinas interligadas por uma estrutura de rede.

No capítulo 5, serão discutidos os princípios do que venha a ser programação paralela, sendo apresentadas as principais arquiteturas de hardware e software empregados para a montagem e utilização de um cluster Beowulf, além de discutir a análise de desempenho obtido pelos algoritmos desenvolvidos nesta tese.

No capítulo 6, serão apresentados os resultados obtidos das análises de três padrões de distribuições de dielétricos: o padrão anular o padrão em bolhas e o padrão estratificado. Serão discutidos os princípios de resolução espacial, repetitividade da solução, tempo serial e tempo paralelo de solução.

Nas conclusões será sintetizado os resultados obtidos além de apontar possíveis desdobramentos das investigações realizadas para trabalhos futuros.

Nos apêndices, serão apresentadas as rotinas serial e paralela da técnica OEP, conjuntamente com a biblioteca que resolve o problema direto.

No anexo, serão apresentadas as funções da biblioteca OPENMPI que foram utilizadas para implementar a solução paralela, elas são responsáveis por interligar os nodos do cluster, tornando transparente para o programa a passagem de informações processadas por processadores distintos.

CAPÍTULO II

TOMOGRAFIA CAPACITIVA ELÉTRICA

2.1 - INTRODUÇÃO

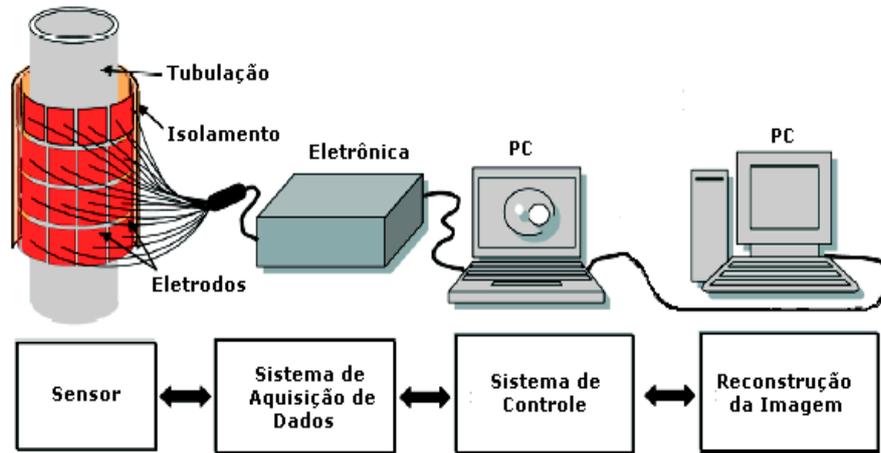
A TCE é um dos métodos tomográficos com melhor relação custo/benefício voltados a obtenção de informações aplicadas ao controle dos escoamentos presentes na indústria do petróleo. Comparativamente com outras modalidades de tomografia, a TCE oferece algumas vantagens, tais como a não utilização de radiação ionizante, uma resposta rápida, um custo relativamente baixo, resistente a condições adversas de temperatura e pressão, sendo também uma técnica não-invasiva e não-intrusiva.

Esta técnica se baseia na medição das mudanças de capacitância originadas pelas mudanças da distribuição dos dielétricos que constituem o fluxo entre as placas de capacitores de um sensor multi eletrodos (normalmente de 8 a 16) dispostos na periferia da tubulação, onde são realizadas as medidas de capacitância. A distribuição de permissividade da secção transversal é reconstruída a partir destas medidas de capacitância empregando convenientemente algum algoritmo de reconstrução (YANG, 2003).

As medidas de capacitância são realizadas por um sensor capacitivo, que é composto por placas de cobre dispostas de forma não paralelas entre si na periferia da tubulação. Além do sensor capacitivo, também são empregados um sistema de transdução eletrônica para converter o sinal de capacitância em um sinal de tensão, seguido de um sistema de controle e aquisição de sinal que converte o valor de tensão em um dado que será processado no último estágio do sistema, o de reconstrução da permissividade da seção imageada utilizando um algoritmo inverso. Detalhes dos sistemas empregados e do posicionamento do sensor podem ser vistos na Figura 2.1.

CAPÍTULO II

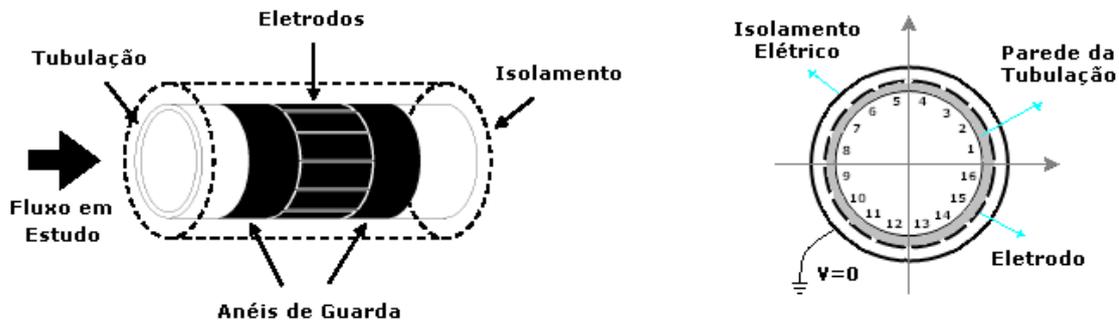
Figura 2.1 – Sistema Tomográfico via Capacitância Elétrica.



Fonte: Adaptado de Yusuf (2014)

Pode-se visualizar com um pouco mais de detalhes o sensor capacitivo nas Figura 2.2. Estes sensores se resumem em um conjunto de placas de cobre dispostas de forma não paralelas entre si, no intuito de que com isso possamos inferir a composição e a distribuição de dielétricos em toda a região interna da tubulação.

Figura 2.2 – Visão do sensor ao redor da tubulação a ser imageada



Fonte: Adaptado de Sun (2008)

Devido as suas características, a TCE apresenta alguns problemas intrínsecos:

- é considerada como uma tomografia do tipo soft-field, ou seja, o meio a ser imageado se modifica com a ação do campo de prospecção adotado (campo elétrico), tornando o processo de reconstrução da imagem um pouco mais complexa do que na tomografia convencional utilizando Raios X, devido à sua não-linearidade entre as medições de capacitância e a respectiva distribuição de permissividade elétrica.

CAPÍTULO II

- b) é considerado um problema mal condicionado, isto significa que a resposta do sensor responde de maneira diferente a uma variação de permissividade na parte interna (região menos sensível) do sensor em relação a uma variação próxima da parede da tubulação (região mais sensível).
- c) apresenta um número reduzido de medidas em relação ao número de incógnitas a serem determinadas (número de pixels na discretização do sensor).
- d) apresenta uma relação não linear entre a capacitância mensurada e a distribuição de permissividade existente, causando uma dificuldade intrínseca ao processo de reconstrução da imagem.

2.2 - AQUISIÇÃO DO SINAL ELÉTRICO

Durante o processo de aquisição das medidas de capacitância, cada eletrodo em um sensor ECT é excitado através da aplicação de um sinal de tensão de referência, enquanto executamos nos demais eletrodos a detecção da carga elétrica induzida sobre o mesmo. Tomemos por exemplo um sensor com 16 eletrodos. Primeiro, consideremos o eletrodo 01 como sendo o eletrodo de excitação (eletrodo fonte) e os eletrodos 02 a 16 como os eletrodos de detecção (eletrodo receptor). Em seguida, o eletrodo 02 é usado como eletrodo de excitação e os eletrodos 03 a 16 como a eletrodos de detecção, e assim por diante, até alcançarmos o eletrodo 15, o qual será o último eletrodo de excitação, e o eletrodo 16 como o de detecção. Para um sensor ECT com N eletrodos, existem $N(N - 1)/2$ combinações de pares de eletrodos, ou seja, $N(N - 1)/2$ medidas independentes de capacitância para uma imagem.

2.3 - ELETRÔNICA DE TRANSDUÇÃO

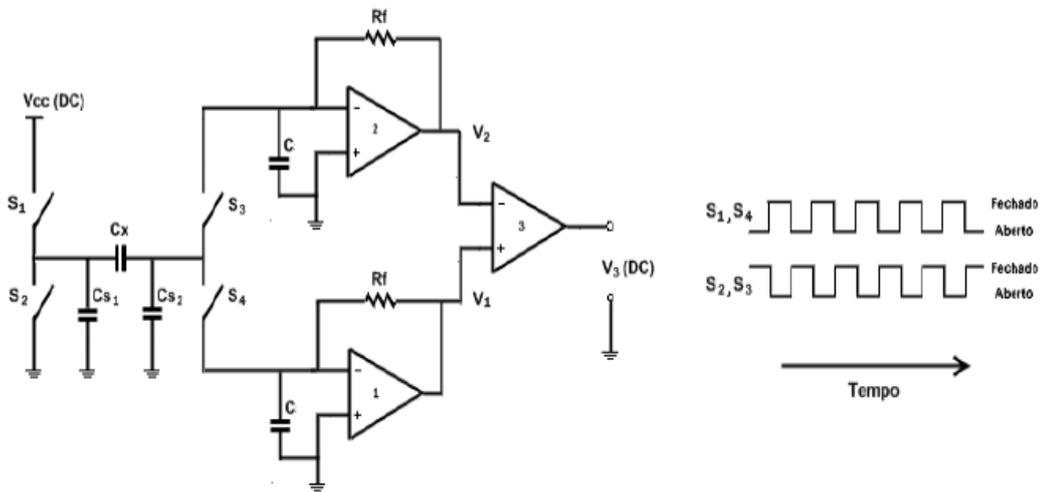
Circuitos específicos de transdução eletrônica são empregados na obtenção de medidas de capacitância nas aplicações que envolvem a TCE. Devido às características encontradas em tais aplicações, como no mapeamento do fluxo de gás-sólidos em leitos fluidizados (DYAKOWSKI, 1997) ou em transportadores pneumáticos (RAO, 2001), ou ainda gás-petróleo em oleodutos, os circuitos de transdução devem ter alto desempenho para que possam medir capacitâncias da ordem de 0,1 fF com acurácia. Imunidade a capacitâncias parasitas, boa relação sinal/ruído, alta sensibilidade e estabilidade térmica são características desejáveis nos circuitos empregados. Também, a frequência de operação dos circuitos é

CAPÍTULO II

importante já ela está relacionada com a sensibilidade das medidas e com a taxa de aquisição de dados necessários para a reconstrução de imagens do fluxo.

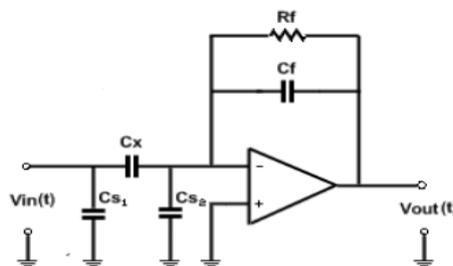
Revisando-se a literatura pertinente, pode-se destacar duas abordagens usualmente empregadas para a medida das capacitâncias (HUANG, 1992; YANG, 1996) percebidas pelos sensores capacitivos: o método de carga/descarga e o da ponte de corrente alternada. Sendo os circuitos mais utilizados para medição de capacitâncias e na TCE, seu estudo e a comparação de suas características são de grande relevância. O diagrama simplificado de cada um dos circuitos pode ser visto nas Figuras 2.3 e 2.4.

Figura 2.3 – Medida da capacitância C_x baseada no método de carga/descarga



Fonte: Adaptado de YANG (1996)

Figura 2.4 – Medida da capacitância C_x baseada em corrente alternada.



Fonte: Adaptado de YANG (1996)

Na Tabela 2.1 estão relacionadas as várias características, vantagens e desvantagens dos circuitos eletrônicos aplicados em cada um dos métodos mencionados.

CAPÍTULO II

Tabela 2.1 – Características, vantagens e desvantagens dos circuitos eletrônicos.

Características do circuito eletrônico	
<i>Carga/Descarga</i>	<i>Baseado em Sinal De Corrente Alternada</i>
1) Imune a capacitâncias parasitas. 2) resolução de até 0,3 fF. 3) Mede impedância. 4) Frequência de carga e descarga de até 2,5MHz.	1) Imune a capacitâncias parasitas. 2) Resolução de até 0,04 pF. 3) Mede impedância, ou com o demodulador sensível de fase mede capacitância e/ou condutância 4) Frequência de excitação de até 1,0 MHz.
Vantagens do circuito eletrônico	
<i>Carga/Descarga</i>	<i>Baseado em Sinal De Corrente Alternada</i>
1) Simples e de baixo custo.	1) Baixa deriva devido ao uso de amplificadores de corrente alternada em vez de amplificadores de corrente contínua. 2) Alto valor da relação sinal/ruído.
Desvantagens do circuito eletrônico	
<i>Carga/Descarga</i>	<i>Baseado em Sinal De Corrente Alternada</i>
1) Injeção de carga através de chaves CMOS causam problema. 2) Os amplificadores que usam a corrente contínua sofrem de problemas de deriva. 3) Afetado pela perda em condutância.	1) Complicado e caro, especialmente quando se trabalha com altas frequências.

Fonte: Adaptado de YANG (1996)

2.4 – RECONSTRUÇÃO DA IMAGEM

A TCE se baseia na aplicação de um perfil de potencial elétrico no contorno de uma tubulação e a respectiva leitura de capacitância obtida entre sensores capacitivos no seu contorno, a partir da relação entre os dados da excitação e da resposta, estima-se a distribuição espacial do material na seção transversal da tubulação, o que pode ser compreendido como uma imagem da seção.

CAPÍTULO II

A reconstrução de uma imagem via TCE é uma tarefa realizada em duas etapas: primeiro deve-se resolver o problema direto, que se resume na determinação dos potenciais elétricos no interior do domínio e das respostas no contorno a partir dos dados da excitação; segundo, deve-se resolver o problema inverso, que é a determinação da distribuição de dielétricos nos pontos internos do domínio a partir da relação entre os dados de excitação e resposta no contorno.

2.4.1 – Resolução do Problema Direto da TCE

O problema direto associado à TCE pode ser definido como: Dada a distribuição de dielétricos sobre o domínio de uma secção transversal de uma tubulação e um perfil apropriado de excitação dos eletrodos fontes, dispostos na periferia da tubulação, deve-se determinar a respectiva configuração do potencial elétrico no interior do sensor que induz o acúmulo de cargas no eletrodo sensor e conseqüentemente a respectiva capacitância entre estes eletrodos.

Para tal, Gatkowski (2014) observa que devesse conhecer a relação causal entre a distribuição do campo elétrico devido ao perfil de excitação do potencial elétrico no eletrodo fonte no meio a ser imageado e a respectiva indução de cargas a se depositar no eletrodo sensor, sendo assim necessário o conhecimento das leis físicas que modelam este fenômeno.

Considerando que o campo eletromagnético pode ser completamente descrito de forma qualitativa e quantitativa pelo conjunto de quatro equações básicas, onde

a) (Lei de Gauss)

$$\vec{\nabla} \cdot \vec{D} = \rho \quad (2.1 - a)$$

b) (Lei de Faraday)

$$\vec{\nabla} \times \vec{E} + \frac{\partial \vec{B}}{\partial t} = 0 \quad (2.1 - b)$$

c) (Não Existência de Monopolo Magnético)

$$\vec{\nabla} \cdot \vec{B} = 0 \quad (2.1 - c)$$

d) (Lei de Ampère)

$$\vec{\nabla} \times \vec{H} = \vec{J} + \frac{\partial \vec{D}}{\partial t} \quad (2.1 - d)$$

Juntamente com a lei de conservação da carga

$$\vec{\nabla} \cdot \vec{J} = -\frac{\partial \rho}{\partial t} \quad (2.2)$$

CAPÍTULO II

São então conhecidas como as equações de Maxwell, estas equações governam o comportamento fenomenológico do processo de determinação da distribuição do potencial dentro de um domínio Ω , obtida a partir do conhecimento de distribuição de material dentro deste domínio e a respectiva excitação sobre o contorno Γ .

Observando-se as relações constitutivas para meios lineares e isotrópicos

$$\vec{D} = \varepsilon \vec{E} \quad (2.3 - a)$$

$$\vec{B} = \mu \vec{H} \quad (2.3 - b)$$

$$\vec{J} = \sigma \vec{E} \quad (2.3 - c)$$

onde \vec{E} é definido como o campo elétrico, \vec{D} o deslocamento elétrico, \vec{H} o campo magnético, \vec{B} a indução magnética, \vec{J} a densidade de corrente elétrica e ρ a densidade de carga, ε a permissividade elétrica (é uma constante física que descreve como um campo elétrico afeta e é afetado por um meio), μ a permeabilidade magnética (é uma medida do grau de magnetização de um material em resposta a um campo magnético) e σ a condutividade elétrica (é uma constante física indicativa da facilidade com a qual um material é capaz de conduzir uma corrente elétrica). Tem-se então juntamente com as Equação (2.1) e (2.2), condições para caracterizar o meio em estudo em termos de sua interação entre o campo eletromagnético e a matéria que o constitui.

Sendo as equações de Maxwell um conjunto de quatro equações diferenciais parciais de 1ª ordem acopladas, relacionando-se o vetor campo elétrico e o magnético, a sua solução matemática pode ser de difícil obtenção. Observando-se então que sob algumas condições simplificadoras, consegue-se desacoplar os campos e solucioná-los para algumas situações particulares, mas, em sua grande maioria de problemas isto não é possível. Um tratamento conveniente para este tipo de problema se dá pela introdução de potenciais (escales e/ou vetoriais), que trasladam o problema de resolver um sistema de equações de 1ª ordem para o de resolver duas equações diferenciais desacopladas parciais de 2ª ordem que também satisfazem as equações de Maxwell identicamente. Introduziremos então os potenciais φ (potencial escalar) e \vec{A} (potencial vetorial).

Desde que $\nabla \cdot \vec{B} = 0$, para qualquer ponto do espaço, podemos definir \vec{B} em termos de um potencial vetorial da seguinte forma

$$\vec{B} = \nabla \times \vec{A} \quad (2.4)$$

CAPÍTULO II

Substituindo a Equação (2.4) na Equação (2.1 – b)

$$\begin{aligned}\vec{\nabla} \times \vec{E} + \frac{\partial(\nabla \times \vec{A})}{\partial t} &= 0 \\ \vec{\nabla} \times \vec{E} + \nabla \times \left(\frac{\partial \vec{A}}{\partial t} \right) &= 0 \\ \vec{\nabla} \times \left(\vec{E} + \frac{\partial \vec{A}}{\partial t} \right) &= 0\end{aligned}\tag{2.5}$$

Isto significa que a quantidade dentro do parêntesis da Equação (2.5) possui um rotacional nulo, podendo então ser expressa em termos de um gradiente de um potencial escalar, tal que

$$\begin{aligned}\vec{E} + \frac{\partial \vec{A}}{\partial t} &= -\vec{\nabla} \varphi \\ \vec{E} &= -\vec{\nabla} \varphi - \frac{\partial \vec{A}}{\partial t}\end{aligned}\tag{2.6}$$

A definição de \vec{B} e \vec{E} em termos do potencial \vec{A} e φ de acordo com a Equação (2.4) e Equação (2.6), satisfazem para qualquer configuração dos potenciais as duas equações de Maxwell que são homogêneas, Equação (2.1 – b) e (2.1 – c). O comportamento dinâmico de \vec{A} e φ deve ser então determinado pelas Equação (2.1 – a) e a Equação (2.1 – d). Substituindo a Equação (2.3 – a) na Equação (2.1 – a), teremos,

$$\begin{aligned}\vec{\nabla} \cdot \vec{D} &= \rho \\ \vec{\nabla} \cdot \varepsilon \vec{E} &= \rho \\ \vec{\nabla} \cdot \left[\varepsilon \left(-\vec{\nabla} \varphi - \frac{\partial \vec{A}}{\partial t} \right) \right] &= \rho \\ -\vec{\nabla}(\varepsilon \vec{\nabla} \varphi) - \vec{\nabla} \cdot \left[\varepsilon \left(\frac{\partial \vec{A}}{\partial t} \right) \right] &= \rho \\ \vec{\nabla}(\varepsilon \vec{\nabla} \varphi) + \varepsilon \frac{\partial(\vec{\nabla} \cdot \vec{A})}{\partial t} &= -\rho\end{aligned}\tag{2.7}$$

E substituindo a Equação (2.3 – a) e a Equação (2.3 – b) na Equação (2.1 – d),

$$\vec{\nabla} \times \vec{H} = \vec{j} + \frac{\partial \vec{D}}{\partial t}$$

CAPÍTULO II

$$\vec{\nabla} \times \vec{B} = \mu \vec{j} + \mu \varepsilon \frac{\partial \vec{E}}{\partial t}$$

$$\vec{\nabla} \times (\vec{\nabla} \times \vec{A}) = \mu \vec{j} + \mu \varepsilon \frac{\partial}{\partial t} \left(-\vec{\nabla} \varphi - \frac{\partial \vec{A}}{\partial t} \right)$$

$$\vec{\nabla}(\vec{\nabla} \cdot \vec{A}) - \nabla^2 \vec{A} = \mu \vec{j} - \mu \varepsilon \frac{\partial}{\partial t} (\vec{\nabla} \varphi) - \mu \varepsilon \frac{\partial}{\partial t} \left(\frac{\partial \vec{A}}{\partial t} \right)$$

$$\vec{\nabla}(\vec{\nabla} \cdot \vec{A}) - \nabla^2 \vec{A} = \mu \vec{j} - \mu \varepsilon \vec{\nabla} \left(\frac{\partial \varphi}{\partial t} \right) - \mu \varepsilon \frac{\partial^2 \vec{A}}{\partial t^2}$$

$$\vec{\nabla} \left(\vec{\nabla} \cdot \vec{A} + \mu \varepsilon \frac{\partial \varphi}{\partial t} \right) - \nabla^2 \vec{A} + \mu \varepsilon \frac{\partial^2 \vec{A}}{\partial t^2} = \mu \vec{j}$$

$$\nabla^2 \vec{A} - \nabla \left(\vec{\nabla} \cdot \vec{A} + \mu \varepsilon \frac{\partial \varphi}{\partial t} \right) - \mu \varepsilon \frac{\partial^2 \vec{A}}{\partial t^2} = \mu \vec{j} \quad (2.8)$$

Podendo-se então reduzir as quatro equações de Maxwell as Equações (2.7) e (2.8). Mesmo assim, ainda se tem duas equações acopladas, o desacoplamento pode ser obtido explorando a arbitrariedade envolvendo a escolha dos potenciais. Sabendo-se que \vec{B} pode ser definido a partir de uma escolha apropriada de \vec{A} , devemos então observar que esta escolha não é única e que podemos substituir o potencial vetorial \vec{A} pelo potencial vetorial $\vec{A} + \nabla \Lambda$ sem modificar o campo \vec{B} , implicando assim que o potencial escalar φ deve ser substituído pelo potencial escalar $\varphi - \partial \Lambda / \partial t$, ou seja

$$\vec{A} \rightarrow \vec{A}' = \vec{A} + \nabla \Lambda \quad (2.9 - a)$$

↓

$$\varphi \rightarrow \varphi' = \varphi - \frac{\partial \Lambda}{\partial t} \quad (2.9 - b)$$

As transformações acima são chamadas de transformações de gauge, e a invariância dos campos a uma transformação como essa é chamada de invariância de gauge.

Pode-se então estabelecer uma escolha que favoreça aos nossos objetivos. Sendo assim, será utilizado o gauge de Coulomb. Este gauge é expresso da seguinte forma,

$$\nabla \cdot \vec{A} = 0 \quad (2.10)$$

CAPÍTULO II

Substituindo a Equação (2.10) na Equação (2.7), tem-se que o potencial escalar φ satisfaz a equação,

$$\vec{\nabla}(\varepsilon\vec{\nabla}\varphi) = -\rho \quad (2.11)$$

e que substituindo a Equação (2.10) na Equação (2.8), tem-se o potencial vetorial satisfazendo a equação de onda inhomogênea,

$$\nabla^2\vec{A} - \mu\varepsilon\frac{\partial^2\vec{A}}{\partial t^2} = \mu\vec{J} + \mu\varepsilon\frac{\partial\varphi}{\partial t} \quad (2.12)$$

Considerando que os nossos meios em estudo são não magnéticos, tem-se que a Equação (2.12) se apresentará como

$$\nabla^2\vec{A} = 0 \quad (2.13)$$

onde a partir da Equação (2.13) juntamente com a Equação (2.10), tem-se como consequência direta que o potencial vetor \vec{A} se reduzirá a um potencial constante, aonde toda a dinâmica do sistema ficará determinada pelo comportamento do potencial escalar φ .

Se o sistema não for constituído por um meio condutor não haverá cargas livres no interior da região da seção imageada, então a relação entre a capacitância e a distribuição de permissividade é modelada pela seguinte expressão:

$$\vec{\nabla}(\varepsilon(x,y)\vec{\nabla}\varphi(x,y)) = 0 \quad \text{onde } x, y \in \Omega \quad (2.14)$$

sendo $\vec{\nabla}$ o operador gradiente, $\varepsilon(x,y)$ é a permissividade elétrica na região imageada e $\varphi(x,y)$ é o potencial elétrico.

A condição de contorno fica definida quando um eletrodo está excitado com uma tensão fixa V_0 (eletrodo fonte) e todos os outros eletrodos são mantidos aterrados, onde a partir daí é multiplexado o processo de leitura em cada um dos eletrodos que estão aterrados (eletrodos receptores) no intuito de mensurar as suas capacitâncias. Sendo então

$$\varphi(x,y) = \begin{cases} V_0 & \subseteq \Gamma_i \\ 0 & \subseteq \text{em todo } \Gamma_k (k \neq i) \end{cases} \quad (2.15)$$

onde Γ_i é a região do contorno da tubulação que compreende o eletrodo fonte e $\Gamma_k (k \neq i)$ é a região do contorno que compreende os eletrodos receptores. Tem-se então que

CAPÍTULO II

$$C = \frac{-\varepsilon_0}{V_c} \iint_{\Omega} \varepsilon(x, y) \vec{V} \varphi(x, y) dx dy \quad (2.16)$$

O problema proposto pelo problema direto da TCE pode ser considerado como um problema eletrostático conforme apresentado pelas Equações (2.14), (2.15) e (2.16), esta condição se deve ao fato de que a velocidade de obtenção dos sinais de capacitância pelo sistema de aquisição é muitas ordens de grandeza maior que a velocidade do fluxo no interior da região da tubulação observada pelo sensor, desacoplando o problema elétrico do problema de transporte de energia e massa associado ao escoamento, além de que o objeto de interesse do problema é a distribuição de material no interior da tubulação, isto pode ser convenientemente determinado pelo conhecimento do perfil de escoamento em uma dada secção, tornando-se assim um problema bidimensional.

Não existe solução analítica geral para a Equação (2.16), sujeita as condições impostas pelas Equação (2.14) e (2.15) para uma distribuição qualquer de permissividade no interior do domínio do sensor e sim apenas soluções particulares para distribuições com simetria axial como no caso de uma distribuição homogênea, de uma distribuição core-flow ou uma distribuição anular.

Belo (1995) apresenta uma solução analítica particular para o caso de simetria apresentado anteriormente, considerando um sensor capacitivo composto com n eletrodos sendo um o eletrodo fonte e os demais, eletrodos sensores aterrados. A expressão por ele obtida é determinada a partir do fornecimento das propriedades dos materiais envolvidos, da geometria da distribuição do escoamento e da geometria do sensor, conforme descrito na Figura 2.5.

O valor de capacitância é determinado por

$$C = \frac{2\varepsilon_0\varepsilon_3}{\pi} \left[\sum_{n=1}^{\infty} \frac{\left(AB \left(\frac{R_3}{R_1} \right)^n + CD \left(\frac{R_3}{R_2} \right)^n - AC \left(\frac{R_2}{R_1} \right)^n - BD \right) \text{sen}(n\theta_1) (\text{sen}(n\theta_2) - \text{sen}(n\theta_3))}{n \left(AB \left(\frac{R_3}{R_1} \right)^n + CD \left(\frac{R_3}{R_2} \right)^n + AC \left(\frac{R_2}{R_1} \right)^n + BD} \right] \quad (2.17)$$

Sendo

$$A = \varepsilon_2 + \varepsilon_1 \quad (2.5 - a)$$

$$B = \varepsilon_3 + \varepsilon_2 \quad (2.5 - b)$$

$$C = \varepsilon_3 - \varepsilon_2 \quad (2.5 - c)$$

CAPÍTULO II

$$D = \varepsilon_2 - \varepsilon_1 \quad (2.5 - d)$$

Onde

$\theta_1 \rightarrow$ Ângulo do eletrodo fonte;

$\theta_2 \rightarrow$ Ângulo inicial do eletrodo sensor;

$\theta_3 \rightarrow$ Ângulo final do eletrodo sensor.

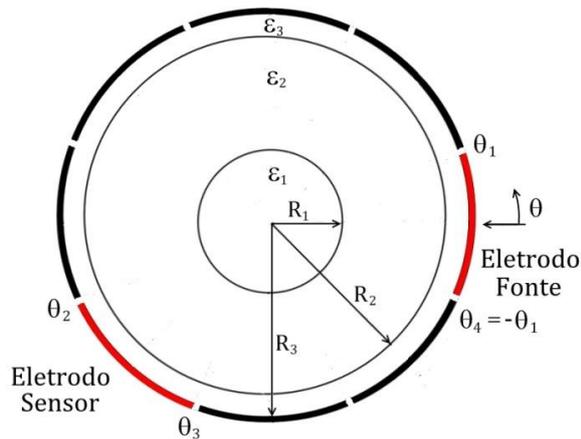
$\varepsilon_0 \rightarrow$ permissividade absoluta do vácuo;

ε_1 e $R_1 \rightarrow$ permissividade relativa e raio do meio mais interno;

ε_2 e $R_2 \rightarrow$ permissividade relativa e raio do meio mais externo;

ε_3 e $R_3 \rightarrow$ permissividade relativa e raio externas das paredes da tubulação;

Figura 2.5 – Sensor de capacitância com 8 eletrodos.



Fonte: Autoria própria (2017)

Considerando o caso geral, não existe uma solução analítica para a resolução do problema direto, sendo assim deve-se investir esforços em soluções numéricas eficientes que possam determinar a relação para qualquer distribuição de material no interior do sensor e a sua respectiva resposta.

Porém, para que a EDP que modela o problema direto da TCE possa ser resolvida computacionalmente, deve-se adequar o problema matemático que é definido sobre um domínio contínuo para um domínio discreto, isto devido à condição de que os autômatos necessários para a solução deste problema apresentarem-se como sistemas de tempo finito e memória finita para realização do computo (HUMPHRIES - 1997). Como não é possível obter soluções numéricas computacionais sobre uma região contínua, devido ao fato dos

CAPÍTULO II

infinitos pontos da mesma, o domínio então deve ser discretizado, isto é, dividido em unidades mais simples de representação e a EDP é formulada de maneira discreta sobre pontos do elemento, levando assim à obtenção de sistemas matriciais de dimensões elevadas em que as incógnitas são as variáveis dependentes nos pontos discretos do domínio. A última tarefa é a solução desse sistema.

SHARMA (2012) apresenta vários métodos que podem ser utilizados para esse propósito, dentre os quais se destacam o Método das Diferenças Finitas (MDF), Método dos Elementos Finitos (MEF) e o Método de Elementos de Contorno (MEC). Cada um dos métodos apresenta vantagens e desvantagens, e dependendo do problema uma das abordagens de solução se mostrará mais apropriada.

Nesta tese será utilizado o MEF, haja vista se tratar de um método robusto e versátil o suficiente para tratar geometrias complexas e meios inomogêneos. Neste método a ideia principal é a divisão do domínio do problema em um número finito de subdomínios, denominados elementos, que se ligam entre si em pontos chamados nós. Assim, após um procedimento conveniente é possível transladar o problema de solucionar uma EDP de maneira analítica sobre o domínio para se obter um sistema de equações algébricas que descreve o valor da grandeza em estudo nos pontos nodais (ALVES, 2007). No próximo capítulo será apresentado o desenvolvimento do procedimento numérico para solução do problema direto da TCE.

2.4.2 – Resolução do Problema Inverso da TCE

A solução do problema inverso associado à TCE pode ser definida como: Dado os valores de capacitância obtidos nos sensores capacitivos disposto na periferia de uma tubulação, determinar qual a distribuição espacial de dielétricos que gerou estes valores. Ou seja, determinar uma imagem da distribuição espacial da seção transversal do escoamento, a partir dos valores de capacitância obtidos por meio dos sensores no contorno da tubulação.

Os algoritmos de reconstrução de imagem voltados para a TCE podem ser categorizados em dois grandes grupos: algoritmos que utilizam um processo de reconstrução a partir das projeções de uma matriz de sensibilidade espacial do sensor e algoritmos que utilizam processos iterativos de busca da distribuição que minimiza o erro obtido entre uma resposta simulada obtida do modelo direto do sensor e os valores de capacitância obtidos a partir da resposta experimental do sensor.

CAPÍTULO II

Historicamente, os algoritmos projetivos foram os primeiros a serem utilizados para reconstruir imagens a partir de dados tomográficos (XIE, 1992), estes algoritmos procuram relacionar de maneira conveniente os resultados obtidos no sistema sensor com a sua distribuição espacial de material no interior do meio imageado evidenciando a sua estrutura interna.

O processo de registro da variação da intensidade incidente e a lida pelos sensores é o primeiro passo para determinação da imagem, este procedimento pode ser sintetizado pela Figura 2.6. O mecanismo de obtenção da imagem se dá pela ponderação de uma matriz de sensibilidade com a matriz de resposta fornecida pelos sensores, formando um mapa de intensidade bidimensional onde após uma normalização, é possível identificar os pontos de inhomogeneidades no interior do corpo em questão, este procedimento pode ser sintetizado pela Figura 2.7. Os principais algoritmos projetivos são descritos no Anexo A.

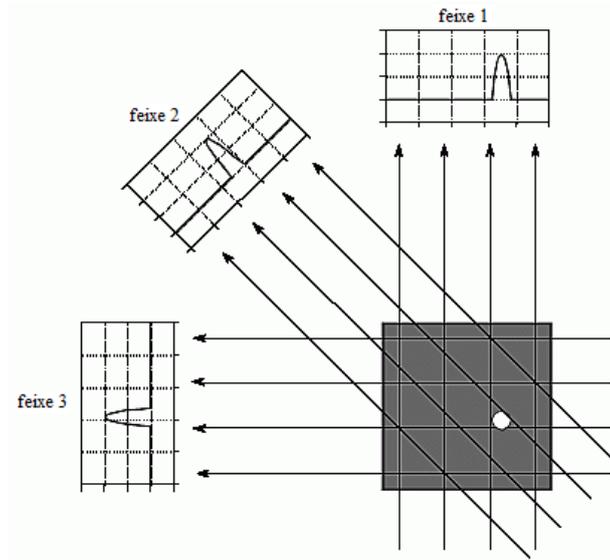
Diferentemente dos algoritmos projetivos, os algoritmos iterativos são formulados em termos de um problema de otimização, caracterizado por iteradas tentativas de minimização de um funcional de erro entre os valores de capacitância obtidos pela solução teórica do problema direto e os valores de capacitância experimentais obtidas pelo sistema sensor. Para isso, deve-se utilizar um algoritmo de busca que a partir de uma distribuição inicial, resolva o problema direto obtendo uma resposta numérica para o valor de capacitância que será então comparado com o valor experimental, gerando assim um valor para o funcional de erro. Em seguida, caso o valor do funcional erro seja maior que um valor pré-estabelecido a priori, existirá uma correção na distribuição proposta inicialmente e então o processo deverá ser repetido iteradas vezes até encontrar um mínimo global do funcional de erro. No final do processo, o valor encontrado numericamente deverá corresponder a àquela distribuição de dielétricos que mais se assemelha a distribuição real no sensor, representando assim uma "imagem da seção da tubulação em um dado momento", este procedimento está descrito na Figura 2.8.

Nos últimos anos, diversos algoritmos iterativos foram propostos no sentido de melhorar a qualidade das imagens empregadas na TCE (JANG, 2006). Ao escolher um algoritmo de reconstrução, as principais considerações que devemos levar em conta são: esforço computacional, rapidez e precisão. Algoritmos projetivos são muito rápidos, mas apresentam uma resolução espacial pobre, isto devido a sua condição de resposta linear entre a distribuição de dielétricos e a capacitância obtida, estes algoritmos oferecem imagens apenas de natureza qualitativa. Algoritmos iterativos podem aumentar a precisão, mas em

CAPÍTULO II

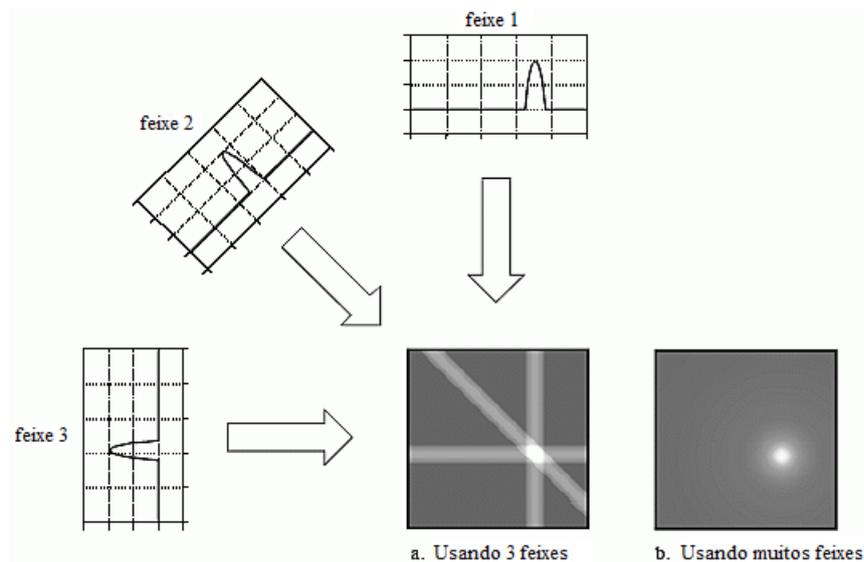
contrapartida podem tornar o processo de reconstrução mais lento. O processo de reconstrução de imagens utilizando um enfoque iterativo é demorado, pois para tal, necessitamos determinar o valor de capacitância resolvendo o problema direto iteradas vezes, tornando-se então uma técnica de resolução temporal pobre, apresentando dificuldades de aplicação em sistemas que careçam de imagens em tempo real.

Figura 2.6 – Representação do processo de obtenção do sinal nos sensores alocados na periferia do meio a ser imageado.



Fonte: Adaptado de Steiner (2007).

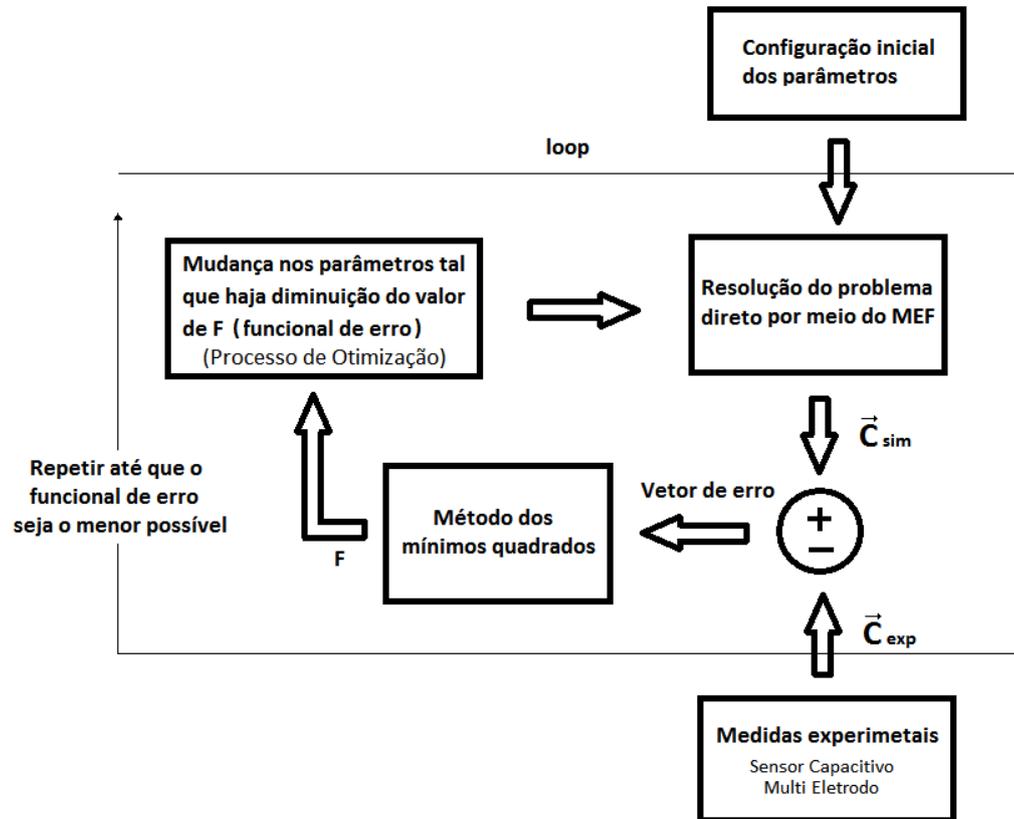
Figura 2.7 – Representação do processo de sobreposição algébrica dos resultados individuais de cada sensor para formação da imagem.



Fonte: Adaptado de Steiner (2007).

CAPÍTULO II

Figura 2.8 – Diagrama esquemático do procedimento utilizado pelos algoritmos iterativos durante o processo de reconstrução.



Fonte: Adaptado de Isaksen (1996)

CAPÍTULO III

SOLUÇÃO DO PROBLEMA DIRETO DA TCE VIA MEF

3.1 - INTRODUÇÃO

É comum na modelagem matemática de diversos problemas de interesse científico e tecnológico, surgirem equações diferenciais (ordinárias ou parciais), seja no equacionamento de problemas envolvendo sistemas eletromagnéticos, processos de transferência de calor e massa, estabilidade aerodinâmica de aeronaves, dispersão de poluentes em sistemas fluidos, engenharia financeira e etc. Kwon (1997) comenta que em muitas situações, estas equações apresentam-se de maneira tão complicada que as suas soluções por meios puramente analíticos são de difícil obtenção ou impossíveis, sendo necessário recorrer a aproximações numéricas da solução analítica desconhecida.

Para resolver tais dificuldades, há a necessidade da utilização de métodos numéricos, obtendo-se assim soluções aproximadas da equação diferencial que se deseja resolver, sujeitas a certas condições de contorno e/ou condições iniciais. Segundo Luersen (2000), a ideia básica dos métodos numéricos para solucionar tal classe de problemas é “discretizar” o problema contínuo, ou seja, o conjunto infinito de números que representam a função ou funções desconhecidas, é substituído por um conjunto finito de coeficientes desconhecidos, sendo que esse processo requer alguma forma de aproximação. Os coeficientes desconhecidos são encontrados através da solução algébrica, ou seja, obtém-se um sistema algébrico de equações do tipo

$$[A][x] = [b] \tag{3.1}$$

CAPÍTULO III

onde $[A]$ é uma matriz quadrada e não-singular, $[b]$ é o vetor independente e $[x]$ é o vetor que contém os coeficientes desconhecidos da solução aproximada, ou seja, é o vetor solução.

Dentre os métodos numéricos mais conhecidos e utilizados pode-se destacar o Método das Diferenças Finitas (MDF), o Método dos Volumes Finitos (MVF), o Método dos Elementos Finitos (MEF) e o Método dos Elementos de Contorno (MEC).

Segundo Silva (2014), dentre os métodos citados anteriormente o MEF possui características que o distingue dos demais métodos numéricos, pois

- a) é um método que possui grande versatilidade na sua fase de modelagem, especialmente à sua capacidade de capturar de forma precisa geometrias arbitrárias das fronteiras e do domínio;
- b) possui implementação genérica para diversos tipos de problemas;
- c) a solução obtida poder ser conhecida para qualquer ponto do domínio;
- d) o equacionamento é resolvido na forma integral (forma fraca), favorecendo maior flexibilidade e generalidade no processo de resolução;
- e) os sistemas matriciais envolvidos são em geral esparsos e simétricos, favorecendo o processo de resolução;
- f) possui facilidades para tratar problemas bidimensionais e tridimensionais constituídos por vários materiais diferentes e com fronteiras irregulares.

Devido as suas especificidades o MEF é indicado para resolução dos mais variados problemas do eletromagnetismo, seja para tratar problemas estáticos, quase-estáticos ou transientes. Estes problemas são formulados usando as equações de Maxwell satisfazendo condições de contorno e/ou condições iniciais apropriadas.

3.2 - UM BREVE HISTÓRICO DO MEF

Santos (2017) observar que as origens do método retomam ao início do cálculo variacional, com J. Bernoulli (1667 – 1748), passando pelos trabalhos de L. Euler (1707 – 1783), J. Lagrange (1736 – 1813), P. Laplace (1749 – 1827), chegando a W. Ritz (1878 – 1909), físico teórico suíço que, no início do século XX, ao se dedicar à solução do problema da deformação de uma membrana elástica submetida à uma força externa, cria um método para aproximar a solução por uma combinação de sequência de funções algébricas. Santos (2017) também comenta que o trabalho de Ritz, apesar de inicialmente não atrair muita atenção no ocidente, teve bastante destaque no então Império Russo, influenciando os trabalhos de S. Timoshenko (1878 – 1972), I. Bubnov (1872 – 1919) e em especial B.

CAPÍTULO III

Galerkin (1871 – 1945), o qual estendeu o entendimento e a validade do método de Ritz para uma classe de problemas mais ampla, todos pesquisadores do Instituto Politécnico de São Petesburgo.

Apesar de não se haver um consenso em se considerar nem uma data oficial e nem um criador específico para o MEF, Felippa (2004) chama a atenção para o fato de que mesmos os princípios matemáticos que norteiam o MEF já houvessem obtido maturidade dentro da matemática pura e aplicada, o MEF só prosperou a partir do desenvolvimento dos primeiros sistemas computadorizados que remontam a meados da década de 1940.

Segundo Felippa (2004), a indústria aeroespacial foi uma das primeiras áreas de interesse tecnológico que impulsionou o desenvolvimento e o aperfeiçoamento dado ao MEF como método numérico, sendo desenvolvido principalmente para a análise de tensões mecânicas nas superfícies das asas de aviões. Os principais responsáveis por esse desenvolvimento do MEF foram J. H. Argyris (1913-2004), R. W. Clough (1920 - 2016), H. C. Martin (1913 - 1970) e O. C. Zienkiewicz (1921 - 2009).

3.3 - DESENVOLVIMENTO MATEMÁTICO DO MEF

Os resultados obtidos pelo MEF podem ser derivados por dois caminhos distintos, um baseado no cálculo variacional e o outro relacionado a métodos residuais, a Figura 3.1 apresenta esta estrutura.

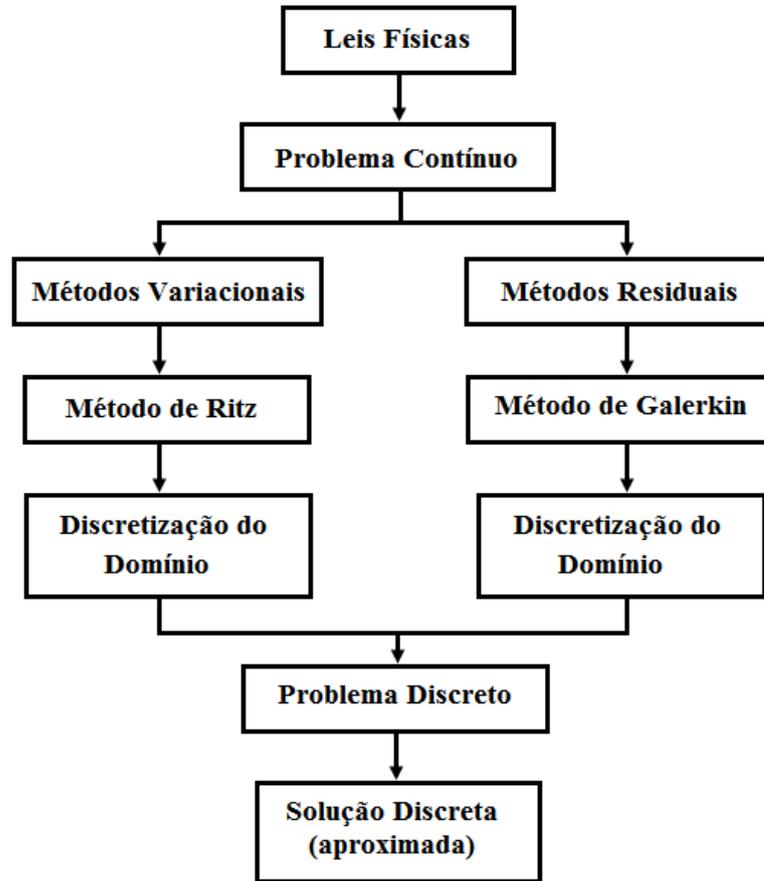
Na solução via métodos variacionais o problema consiste em transladar a solução da equação diferencial pela determinação da função que consiga minimizar ou maximizar a energia potencial (ψ) do sistema em relação à constantes indeterminadas, dentre os diversos métodos variacionais, o mais conhecido é o método desenvolvido por Ritz.

Segundo Rezende (2005), a ideia do método de Ritz consiste em se definir um subconjunto finito de funções admissíveis (ou seja, que satisfazem as condições de contorno) $\varphi_j(x)$ ($j = 1, 2, \dots, n$), geralmente esta escolha é dada por funções polinomiais ou trigonométricas, para então construir uma aproximação $\varphi^n(x)$ para a solução do problema por meio da combinação linear entre essas funções, sendo

$$\varphi^n(x) = \sum_{j=1}^n a_j \varphi_j(x) \quad (3.2)$$

CAPÍTULO III

Figura 3.1 – Representação esquemática do procedimento de solução via MEF.



Fonte: Adaptado de Pereira (2000)

$$\psi(\varphi^n(x)) = \psi(a_1, a_2, \dots, a_n) \quad (3.3)$$

Considerando que os sistemas físicos na condição estacionária procuram se estabelecer na configuração de menor energia, é possível então determinar os coeficientes a_j a partir da determinação dos valores que minimizam o funcional energia, obtendo-se dessa maneira um sistema algébrico em termo dos a_j ($j = 1, 2, \dots, n$) cuja solução será utilizada para determinar a solução $\varphi^n(x)$ na Equação (3.2)

$$\frac{\partial \psi}{\partial a_j} = 0, \quad j = 1, 2, \dots, n \quad (3.3)$$

A solução exata para o problema original pode ser obtida tomando o limite

$$\varphi(x) = \lim_{n \rightarrow \infty} \varphi^n(x) \quad (3.4)$$

CAPÍTULO III

O método de Ritz possui uma base física e matemática sólida que justifica o seu emprego como método de solução, mas uma limitação do método é que ele depende da escolha das funções de teste que devem ser conduzida a partir do conhecimento do sistema e suas propriedades gerais, além de que a expressão analítica da energia potencial do sistema deve ser conhecida a priori em termos da grandeza a ser determinada, sendo que esta situação nem sempre é possível.

Já na solução via métodos residuais, uma função candidata desconhecida é expandida em um conjunto de funções de teste, que são arbitrárias, mas com constantes ajustáveis, que são escolhidas para fornecer a melhor solução para a equação diferencial. Dentre os métodos residuais, Resende (2005) observa que o método empregado por Galerkin é um dos mais populares, isto é devido à sua acurácia e maior generalidade.

O procedimento formal do método de Galerkin pode ser enunciado da seguinte forma: Seja $\Omega \subset \mathbb{R}^n$ um domínio limitado, onde deve-se obter em Ω uma solução da equação diferencial

$$D[\varphi(x)] = 0 \tag{3.5}$$

Onde D é um operador diferencial de $L^2(\Omega)$ e $\varphi: \Omega \rightarrow \mathbb{R}$ uma função de várias variáveis, satisfazendo a condição de contorno

$$\varphi|_{\Gamma} = 0 \tag{3.6}$$

Se a solução $\varphi(x_1, x_2, \dots, x_n)$ for solução exata da Equação (3.5) em Ω então $D[\varphi(x_1, x_2, \dots, x_n)] \equiv 0$ em Ω . Conseqüentemente, a função $D[\varphi(x_1, x_2, \dots, x_n)]$ é ortogonal à toda função $\varphi_j(x) \in L^2(\Omega)$, isto é,

$$\int_{\Omega} D[\varphi(x)]\varphi_j(x)dx = 0 \tag{3.7}$$

A solução $\varphi(x)$ é então aproximada por

$$\varphi^n(x) = \sum_{i=1}^n a_i \varphi_i(x) \tag{3.8}$$

Onde $\{\varphi_i(x)\}$ é um conjunto de funções linearmente independentes definidas em Ω , satisfazendo a condição dada pela Equação (3.6). Os coeficientes a_i são escolhidos de modo que às n primeiras funções do conjunto $\{\varphi_i(x)\}$ sejam ortogonais entre si.

CAPÍTULO III

$$\iint_{\Omega} D \left[\sum_{i=1}^n a_i \varphi_i(x) \right] \varphi_j(x) dx dy = 0, \quad j = 1, 2, \dots, n \quad (3.9)$$

Desta forma as aproximações $\varphi^n(x)$ são projeções ortogonais da solução desejada $\varphi(x)$ em um subespaço de dimensão finita S tal que $\dim(S) = n$. A solução $\varphi(x)$ é obtida fazendo o limite

$$\varphi(x) = \lim_{n \rightarrow \infty} \varphi^n(x) \quad (3.10)$$

O método de Galerkin consiste então de um procedimento de determinação de aproximações de $\varphi^n(x)$ que são projeções ortogonais de $\varphi \in V$ em um subespaço de dimensão finita $S = \{\varphi_1, \varphi_2, \dots, \varphi_n\} \subset V$, gerado pelos n primeiros vetores da base. Desta forma os coeficientes a_i serão determinados pelo sistema algébrico obtido do produto interno definido pela Equação (3.7) .

Uma observação feita por Finlayson (2014) é de que os resultados obtidos pelo método de Galerkin possuem uma maior generalidade de aplicação e obtém de maneira idêntica aos resultados encontrados pelo método de Ritz. Nesta tese será desenvolvido o procedimento do MEF via resíduos ponderados utilizando método de Galerkin

3.4 – DESENVOLVIMENTO DO MEF UTILIZANDO O MÉTODO DE GALERKIN

Se o operador diferencial descrito na Equação (2.14) for substituído na Equação (3.9) obtém-se

$$\iint_{\Omega} [\vec{\nabla}(\varepsilon(x, y) \vec{\nabla} \varphi^n(x, y)) \varphi_j(x, y)] dx dy = 0 \quad (3.11)$$

utilizando o teorema de Green na Equação (3.11) para o plano, chega-se a

$$\iint_{\Omega} [\varepsilon(x, y) \vec{\nabla} \varphi^n(x, y) \cdot \vec{\nabla} \varphi_j(x, y)] dx dy - \oint_{\Gamma} [\vec{\nabla} \varphi(x, y) * \varphi_j(x, y)] d\vec{s} = 0 \quad (3.12)$$

CAPÍTULO III

Como existe uma determinada liberdade na escolha da função $\varphi_j(x, y)$, pode-se escolher esta função de tal maneira que ela se anule no contorno do domínio, sendo assim a Equação (3.12) torna-se

$$\iint_{\Omega} [\varepsilon(x, y) \vec{\nabla} \varphi^n(x, y) \cdot \vec{\nabla} \varphi_j(x, y)] dx dy = 0 \quad (3.13)$$

A forma da equação acima é conhecida como forma fraca da equação de Laplace. Assim, toda solução da equação original dada pela Equação (3.13) é simultaneamente uma solução da Equação (2.14). O termo "fraca" se refere ao fato de que as condições impostas à uma possível solução da Equação (3.13) são menos restritivas que às impostas à equação de Laplace original, dada pela Equação (2.14) (forma forte).

A função solução da Equação (2.14) deve ter as derivadas de segunda ordem contínuas, ao passo que uma função para a solução da Equação (3.13) só precisa possuir a derivada de primeira ordem contínua. A grande vantagem que se obtém com o uso da forma fraca, dada pela Equação (3.13), é o fato de que funções lineares poderem ser admitidas como solução.

3.4.1 – Discretização do domínio em um número finito de elementos

Para se determinar a distribuição do potencial $\varphi^n(x, y)$ no interior do domínio mostrado na Figura 3.2, será dividido a região de interesse em um número de subdomínios menores, este processo é chamado de discretização, como mostrado na Figura 3.3.

Na nomenclatura do MEF, cada subdomínio é denominado de elemento. O elemento pode possuir várias formas mas a mais simples é o triângulo, devido a suas características, ele pode ladrilhar domínios bidimensionais quaisquer com boa precisão. Em cada elemento são escolhidos pontos específicos nos quais a solução será determinada, sendo no caso de um elemento triangular linear os três vértices do mesmo, os quais são chamados de nós. O domínio discretizado é comumente designado por malha, sendo a mesma caracterizada por M nós (pontos discretos) e N elementos (subdomínios).

Devido à complexidade dos domínios em estudo em relação a distribuição geométrica e de materiais no perfil da tubulação, torna-se conveniente a adoção de um sistema de geração de malhas automática conforme mostrado em Fenner (2012).

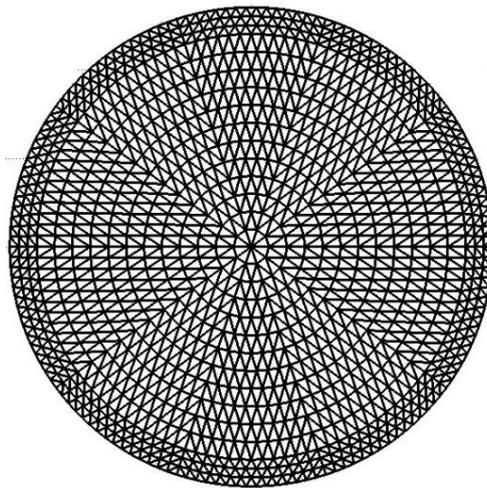
CAPÍTULO III

Figura 3.2 – Corte da Secção Transversal do Sensor.



Fonte: Autoria própria (2017)

Figura 3.3 – Discretização do Domínio do Sensor.



Fonte: Autoria própria (2017)

3.4.2 – Discretização das equações

Uma vez que o domínio em estudo foi discretizado, pode-se obter as equações aproximadas válidas para cada subdomínio designado por Ω_e , chamadas de equações discretas. Considerando-se a subdivisão do domínio original em M elementos, pode-se inicialmente escrever a Equação (3.13) na seguinte forma:

CAPÍTULO III

$$\sum_{e=1}^M \iint_{\Omega_e} [\varepsilon(x, y) \vec{\nabla} \varphi^n(x, y) \cdot \vec{\nabla} \varphi_j(x, y)] dx dy = 0 \quad (3.14)$$

A Equação (3.14) estabelece simplesmente que a integral foi dividida em uma soma de integrais parciais, abrangendo todos os M elementos que compõem o domínio. A fim de avaliar as integrais na Equação (3.14), é preciso determinar a forma da solução aproximada e da função de teste $\varphi_j(x, y)$.

A partir de agora aproximaremos o potencial em cada elemento e como sendo $\varphi_e(x, y)$ e então relacionaremos a distribuição de potencial em vários elementos tal que o potencial seja contínuo nas intersecções dos contornos de cada um. A solução discretizada pode ser expressa da seguinte maneira

$$\varphi^n(x, y) = \sum_{e=1}^N \varphi_e(x, y) \quad (3.15)$$

onde N é o número de elementos utilizados na discretização.

Será feito então uma aproximação linear para caracterizar a distribuição do potencial φ_e no interior de cada elemento, tal que

$$\varphi_e(x, y) = A + Bx + Cy \quad (3.16)$$

onde a , b e c são constantes a determinar. Uma consideração que deve ser feita é de que o potencial φ_e definido da forma da Equação (3.16) é geralmente não nulo para o elemento e , e nulo para qualquer outro elemento.

Sendo um elemento triangular típico tal como mostrado na Figura 3.4, tem-se que o potencial definido em cada vértice do triângulo (nós da malha) φ_{e1} , φ_{e2} e φ_{e3} são obtidos respectivamente utilizando a Equação (3.16), de tal forma que

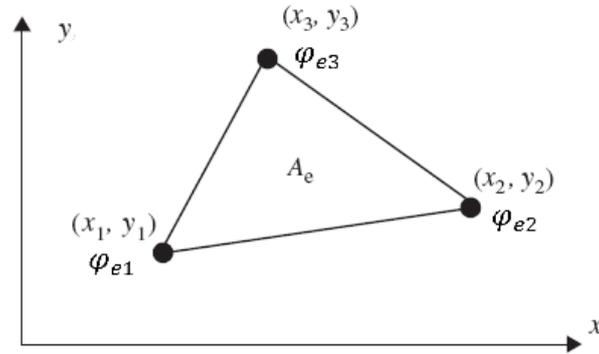
$$\begin{bmatrix} \varphi_{e1} \\ \varphi_{e2} \\ \varphi_{e3} \end{bmatrix} = \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_1 \\ 1 & x_3 & y_1 \end{bmatrix} \times \begin{bmatrix} A \\ B \\ C \end{bmatrix} \quad (3.17)$$

Os coeficientes A , B e C são determinados a partir da Equação (3.17) como

$$\begin{bmatrix} A \\ B \\ C \end{bmatrix} = \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_1 \\ 1 & x_3 & y_1 \end{bmatrix}^{-1} \times \begin{bmatrix} \varphi_{e1} \\ \varphi_{e2} \\ \varphi_{e3} \end{bmatrix} \quad (3.18)$$

CAPÍTULO III

Figura 3.4 – Elemento Triangular Linear



Fonte: Autoria própria (2017)

reescrevendo a Equação (3.16) em forma matricial, tem-se

$$\varphi_e(x, y) = A + Bx + Cy$$

⇓

$$\varphi_e(x, y) = [1 \quad x \quad y] \times \begin{bmatrix} A \\ B \\ C \end{bmatrix} \quad (3.19)$$

substituindo a Equação (3.18) na Equação (3.19)

$$\varphi_e(x, y) = [1 \quad x \quad y] \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{bmatrix}^{-1} \times \begin{bmatrix} \varphi_{e1} \\ \varphi_{e2} \\ \varphi_{e3} \end{bmatrix}$$

$$\varphi_e(x, y) = [1 \quad x \quad y] * \left(\frac{1}{2a} \begin{bmatrix} x_2y_3 - x_3y_2 & x_3y_1 - x_1y_3 & x_1y_2 - x_2y_1 \\ y_2 - y_3 & y_3 - y_1 & y_1 - y_2 \\ x_3 - x_2 & x_1 - x_3 & x_2 - x_1 \end{bmatrix} \right) * \begin{bmatrix} \varphi_{e1} \\ \varphi_{e2} \\ \varphi_{e3} \end{bmatrix}$$

$$\varphi_e(x, y) = \frac{1}{2a} \times [1 \quad x \quad y] * \begin{bmatrix} x_2y_3 - x_3y_2 & x_3y_1 - x_1y_3 & x_1y_2 - x_2y_1 \\ y_2 - y_3 & y_3 - y_1 & y_1 - y_2 \\ x_3 - x_2 & x_1 - x_3 & x_2 - x_1 \end{bmatrix} * \begin{bmatrix} \varphi_{e1} \\ \varphi_{e2} \\ \varphi_{e3} \end{bmatrix} \quad (3.20)$$

Podemos reescrever a Equação (3.20) convenientemente da forma

$$\varphi_e(x, y) = [\alpha_1(x, y) \quad \alpha_2(x, y) \quad \alpha_3(x, y)] * \begin{bmatrix} \varphi_{e1} \\ \varphi_{e2} \\ \varphi_{e3} \end{bmatrix} = \sum_{i=1}^3 \alpha_i(x, y) \varphi_{ei} \quad (3.21)$$

onde

$$\alpha_1(x, y) = \frac{1}{2a} [(x_2y_3 - x_3y_2) + (y_2 - y_3)x + (x_3 - x_2)y] \quad (3.22 - a)$$

CAPÍTULO III

$$\alpha_2(x, y) = \frac{1}{2a} [(x_3 y_1 - x_1 y_3) + (y_3 - y_1)x + (x_1 - x_3)y] \quad (3.22 - b)$$

$$\alpha_3(x, y) = \frac{1}{2a} [(x_1 y_2 - x_2 y_1) + (y_1 - y_2)x + (x_2 - x_1)y] \quad (3.22 - c)$$

e a é a área do elemento e , ou seja

$$a = \frac{1}{2} \begin{vmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{vmatrix} = \frac{1}{2} \{(x_2 y_3 - x_3 y_2) + (x_3 y_1 - x_1 y_3) + (x_1 y_2 - x_2 y_1)\} \quad (3.23)$$

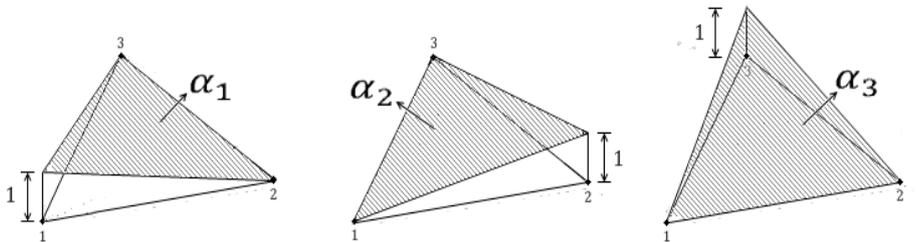
O valor de a é positivo se numerarmos os vértices do triângulo de forma horária (iniciando a numeração de qualquer vértice) como mostrado na Figura. 3.4. Note que a Equação (3.21) dá o potencial em algum ponto (x, y) no interior do elemento e , a partir de uma combinação linear dos potenciais em cada um dos vértices. Pode-se observar ainda que os termos $\alpha_i(x, y)$ são funções de interpolação lineares, elas são chamadas de equações de forma do elemento e possuem as seguintes propriedades

$$\alpha_i(x_j, y_j) = \begin{cases} 1 & \text{se } i = j \\ 0 & \text{se } i \neq j \end{cases} \quad (3.23 - a)$$

$$\sum_{i=1}^3 \alpha_i(x, y) = 1 \quad (3.23 - b)$$

as funções de forma $\alpha_1(x, y)$, $\alpha_2(x, y)$ e $\alpha_3(x, y)$ estão representadas na Figura 3.5.

Figura 3.5 – Representação gráfica das funções de forma sobre cada nó



Fonte: Autoria própria (2017)

Uma outra particularidade importante no desenvolvimento do MEF utilizando o método de Galerkin é que as funções de teste são as próprias funções de forma $\varphi_j(x, y) = \alpha_j(x, y)$. Com estas considerações e substituindo-se a Equação (2.21), a

CAPÍTULO III

Equação (3.15) e Equação (3.21) na Equação (3.14), resulta para a integral sobre o domínio Ω_e a seguinte expressão:

$$\iint_{\Omega_e} \left(\varepsilon(x, y) \vec{\nabla} \left[\sum_{i=1}^3 \alpha_i(x, y) \varphi_{ei} \right] \cdot \vec{\nabla} \alpha_i(x, y) \right) dx dy = 0, \quad i = 1, 2, 3 \quad (3.24)$$

Considerando-se a simplificação de que a distribuição de permissividade no interior do elemento é constante ($\varepsilon(x, y) = \varepsilon$) e que pela linearidade das funções de forma $\alpha_i(x, y)$, a avaliação da integral resulta em valores constantes, pode-se escrever a Equação (3.24) na forma:

$$\sum_{i=1}^3 \left[\varepsilon \iint_{\Omega_e} (\vec{\nabla} \alpha_i(x, y) \cdot \vec{\nabla} \alpha_j(x, y)) dx dy \right] \varphi_{ei} = 0, \quad j = 1, 2, 3 \quad (3.25)$$

Considerando-se a Equação (3.25) e variando-se os índices i e j de 1 a 3 obtém-se um sistema local de equações, onde os potenciais dos nós aparecem como incógnitas:

$$\sum_{i=1}^3 c_{ij}^e \varphi_{ei} = 0, \quad j = 1, 2, 3 \quad (3.26)$$

sendo

$$c_{ij}^e = \varepsilon \iint_{\Omega_e} (\vec{\nabla} \alpha_i(x, y) \cdot \vec{\nabla} \alpha_j(x, y)) dx dy, \quad i = 1, 2, 3 \quad e \quad j = 1, 2, 3 \quad (3.27)$$

reescrevendo a Equação (3.26) na forma matricial, tem-se

$$\underbrace{[c_{ij}^e]}_{3 \times 3} \cdot \underbrace{[\varphi_{ei}]}_{3 \times 1} = \underbrace{[0]}_{3 \times 1} \quad (3.28)$$

onde

$$[c_{ij}^e] = \begin{bmatrix} c_{11}^e & c_{12}^e & c_{13}^e \\ c_{21}^e & c_{22}^e & c_{23}^e \\ c_{31}^e & c_{32}^e & c_{33}^e \end{bmatrix}, \quad [\varphi_{ei}] = \begin{bmatrix} \varphi_{e1} \\ \varphi_{e2} \\ \varphi_{e3} \end{bmatrix} \quad e \quad [0] = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (3.29)$$

Calculando os valores dos coeficientes c_{ij}^e apresentados na Equação (3.27), é possível observar que a matriz resultante é uma matriz definida positiva e simétrica, onde

CAPÍTULO III

$$c_{11}^e = \frac{1}{4a}(y_2 - y_3)^2 + (x_3 - x_2)^2;$$

$$c_{22}^e = \frac{1}{4a}(y_3 - y_1)^2 + (x_1 - x_3)^2;$$

$$c_{33}^e = \frac{1}{4a}(y_1 - y_2)^2 + (x_2 - x_1)^2;$$

$$c_{21}^e = c_{12}^e = \frac{1}{4a}(y_2 - y_3)(y_3 - y_1) + (x_3 - x_2)(x_1 - x_3);$$

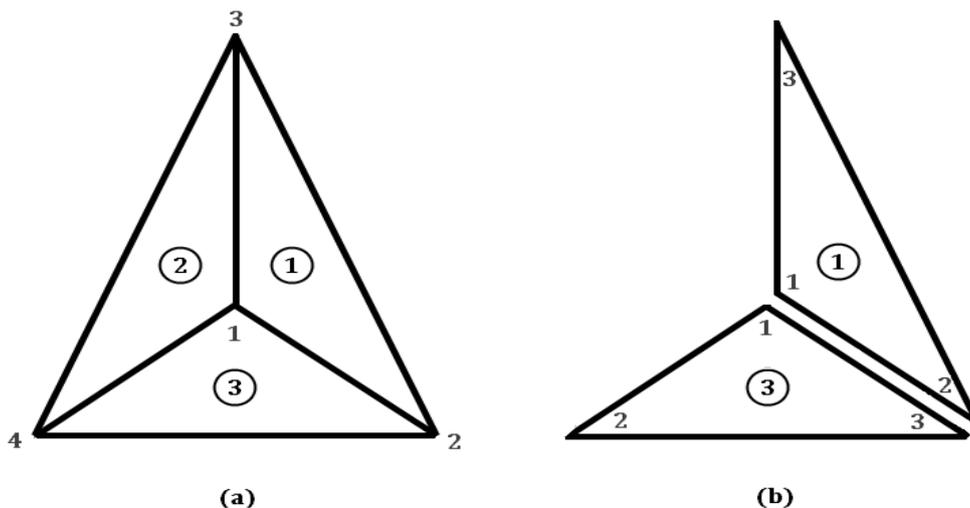
$$c_{31}^e = c_{13}^e = \frac{1}{4a}(y_2 - y_3)(y_1 - y_2) + (x_3 - x_2)(x_2 - x_1);$$

$$c_{32}^e = c_{23}^e = \frac{1}{4a}(y_3 - y_1)(y_1 - y_2) + (x_1 - x_3)(x_2 - x_1).$$

3.4.3 – Transladação da modelagem local para a modelagem global

Nesta seção é construído o sistema matricial envolvendo todos os pontos nodais, que também é denominado de sistema global. A dimensão das matrizes globais é determinada pelo número de nós da malha. Com o objetivo de descrever esse método, será apresentado uma malha simples, mostrada na Figura 3.6 – a, que é composta de quatro nós. Esses nós definem a numeração global da malha, enquanto que na Figura 3.6 – b, ilustra-se a padronização da numeração local.

Figura 3.6 – Discretização do domínio com 3 elementos triangulares linear.



Legenda: (a) Numeração global dos nodos da malha, (b) Padrão de numeração local dos nodos dos elementos.

Fonte: Autoria própria (2017)

CAPÍTULO III

A Tabela 3.1 ilustra a associação entre a identificação do elemento e as suas numerações locais e globais. Por exemplo, de acordo com a Figura. 3.6 – a, o elemento 3 cuja numeração local do vértice é 2, corresponde à numeração global 4.

Tabela 3.1: Associação do elemento com sua numeração local e global.

Nº Elem.	Nº local			Nº Global
	1	2	3	
1	1	2	3	
2	1	3	4	
3	1	4	2	

Fonte: Autoria própria (2017)

Após o cálculo das matrizes $[C_{ij}^e]_{Local}$ para cada elemento conforme a Equação (3.29), devemos montar a matriz global $[C_{ij}^e]_{Global}$ alocando cada componente da matriz local na respectiva posição da matriz global, para isto deve ser determinar a correspondência entre as coordenadas locais com as coordenadas globais da matriz de capacitância. Como exemplo será tomado o elemento 3.

Tabela 3.2 – Mapeamento dos componentes locais nos componentes globais da matriz de capacitância relacionadas ao elemento 3.

		Numeração Global		
		1	4	2
Numeração Global	1	c_{11}^3	c_{12}^3	c_{13}^3
	4	c_{21}^3	c_{22}^3	c_{23}^3
	2	c_{31}^3	c_{32}^3	c_{33}^3

Fonte: Autoria própria (2017)

$$[C_{ij}^3]_{Global} = \begin{bmatrix} c_{11}^3 & c_{13}^3 & 0 & c_{12}^3 \\ c_{31}^3 & c_{33}^3 & 0 & c_{32}^3 \\ 0 & 0 & 0 & 0 \\ c_{21}^3 & c_{23}^3 & 0 & c_{22}^3 \end{bmatrix} \quad (3.30 - a)$$

CAPÍTULO III

Deve-se então agora proceder de maneira semelhante ao procedimento delineado pelo mapeamento dado na Tabela 3.2 aos elementos 1 e 2, alocando cada componente das matrizes locais $[C_{ij}^1]_{Local}$ e $[C_{ij}^2]_{Local}$ em suas respectivas posições nas matrizes globais $[C_{ij}^1]_{Global}$ e $[C_{ij}^2]_{Global}$. Tem-se então que

$$\begin{aligned}
 [C_{ij}^1]_{local} &= \begin{bmatrix} c_{11}^1 & c_{12}^1 & c_{13}^1 \\ c_{21}^1 & c_{22}^1 & c_{23}^1 \\ c_{31}^1 & c_{32}^1 & c_{33}^1 \end{bmatrix} \\
 &\Downarrow \\
 [C_{ij}^1]_{global} &= \begin{bmatrix} c_{11}^1 & c_{12}^1 & c_{13}^1 & 0 \\ c_{21}^1 & c_{22}^1 & c_{23}^1 & 0 \\ c_{31}^1 & c_{32}^1 & c_{33}^1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (3.30 - b)
 \end{aligned}$$

e

$$\begin{aligned}
 [C_{ij}^2]_{local} &= \begin{bmatrix} c_{11}^2 & c_{12}^2 & c_{13}^2 \\ c_{21}^2 & c_{22}^2 & c_{23}^2 \\ c_{31}^2 & c_{32}^2 & c_{33}^2 \end{bmatrix} \\
 &\Downarrow \\
 [C_{ij}^2]_{global} &= \begin{bmatrix} c_{11}^2 & 0 & c_{12}^2 & c_{13}^2 \\ 0 & 0 & 0 & 0 \\ c_{21}^2 & 0 & c_{22}^2 & c_{23}^2 \\ c_{31}^2 & 0 & c_{32}^2 & c_{33}^2 \end{bmatrix} \quad (3.30 - c)
 \end{aligned}$$

Logo após o processo de espalhamento da matriz local na matriz global, obtém-se a matriz $[C_{ij}]_{Global}$ representando todo o domínio Ω a partir da sobreposição das respectivas matrizes de cada elemento. Sendo então:

$$[C_{ij}] = \sum_{e=1}^3 [C_{ij}^e]_{Global} \quad (3.31)$$

Obtendo-se então

$$[C_{ij}] = \begin{bmatrix} c_{11}^1 + c_{11}^2 + c_{11}^3 & c_{12}^1 + c_{12}^2 + c_{12}^3 & c_{13}^1 + c_{13}^2 + c_{13}^3 & c_{13}^2 + c_{12}^3 \\ c_{21}^1 + c_{21}^2 + c_{21}^3 & c_{22}^1 + c_{22}^2 + c_{22}^3 & c_{23}^1 + c_{23}^2 + c_{23}^3 & c_{23}^2 + c_{23}^3 \\ c_{31}^1 + c_{31}^2 + c_{31}^3 & c_{32}^1 + c_{32}^2 + c_{32}^3 & c_{33}^1 + c_{33}^2 + c_{33}^3 & c_{33}^2 + c_{33}^3 \\ c_{31}^2 + c_{31}^3 & c_{32}^2 + c_{32}^3 & c_{33}^2 + c_{33}^3 & c_{33}^2 + c_{33}^3 \end{bmatrix} \quad (3.32)$$

CAPÍTULO III

obtendo-se então a matriz global resultante. O sistema linear global a ser resolvido é dado da forma

$$\underbrace{[C_{ij}]}_{4 \times 4} \cdot \underbrace{[\varphi_{ei}]}_{4 \times 1} = \underbrace{[0]}_{4 \times 1} \quad (3.33)$$

3.4.4 – Resolução do sistema de equações lineares

Se houver todos os nós livres e os nós prescritos numerados de forma ordenada, pode-se escrever a Equação (3.33) de tal forma que

$$[C_{ff} \quad C_{fp}] \begin{bmatrix} \varphi_f \\ \varphi_p \end{bmatrix} = 0 \quad (3.34)$$

onde os subscritos f e p , referem-se respectivamente a nós com potenciais livres ou prescrito (fixados). Para tal, a malha utilizada segue uma estrutura similar a apresentada na Figura 3.3 onde a numeração dos nós inicia no ponto central da malha e vai aumentando gradualmente de dentro para fora da malha, conforme a estruturação apresentada na Figura 3.6, o que garante que os nós do contorno sejam contíguos e com os maiores valores.

Considerando que φ_p são potenciais prescritos, pode-se observar que aplicando a Equação (3.20) a Equação (3.21) obtém-se

$$[C_{ff}][\varphi_f] + [C_{fp}][\varphi_p] = [0] \quad (3.35)$$

ou

$$[C_{ff}][\varphi_f] = -[C_{fp}][\varphi_p] \quad (3.36)$$

esta equação pode ser escrita como

$$[C][\varphi] = [Q] \quad (3.37)$$

onde $[\varphi] = [\varphi_f]$, $[C] = [C_{ff}]$, $[Q] = -[C_{fp}][\varphi_p]$.

Uma vez que $[C]$ é, em geral, não singular, o potencial nos nodos livres pode ser encontrado usando a Equação (3.37) (SPINK *et al.* - 1998).

Pode-se perceber facilmente que com o aumento do número de elementos a matriz $[C]$ torna-se demasiadamente complexa. Para se alcançar uma implementação numérica

CAPÍTULO III

eficiente deve-se então observar e explorar as características de simetria e esparsividade desta matriz (FANG, 2005).

Segundo Fulton (1995) e Spink *et al.* (1998), um dos métodos que melhor se adéquam ao perfil da matriz obtida pela Equação (3.37) é o método da decomposição de Cholesky, entretanto o Método dos Gradientes Conjugados (MGC) desenvolvido por Hestenes e Stiefel (1952), vem se mostrando como uma das técnicas mais promissoras de resolução de sistemas lineares para problemas dessa natureza (BARRET *et al.* - 1994). O sucesso do MGC para problemas com este tipo de perfil se deve ao fato de que as suas principais operações numéricas, o produto matriz-vetor e o produto vetor-vetor, podem ser realizados acessando apenas as posições não nulas da matriz de $[C]$, economizando assim tanto memória quanto processamento na hora da resolução do sistema linear.

O MGC é um método iterativo que parte de uma solução inicial x_0 e a cada iteração procura-se reduzir o erro cometido em se tomar x_0 como uma solução aproximada até que está se aproxime da solução exata. O nível de aproximação é definido a partir da implementação de alguma condição de parada, permitindo ao usuário uma escolha apropriada do nível de precisão da solução em relação ao tempo de execução da rotina. O ganho em se adotar o MGC fica mais evidente com o aumento do tamanho da malha e com o aumento do número de eletrodos, pois o tamanho da malha está associado ao tamanho da matriz e o número de eletrodos está associado ao número de vezes que a rotina será chamada para resolver o problema direto.

Encontra-se em Davis (2006) e em Saad (2003) um estudo minucioso dos principais algoritmos de resolução de sistemas lineares que citados anteriormente.

CAPÍTULO IV

SOLUÇÃO DO PROBLEMA INVERSO VIA OTIMIZAÇÃO POR ENXAME DE PARTÍCULAS

4.1 - INTRODUÇÃO

Desde longa data, o comportamento de certos animais tais como pássaros, peixes, formigas, abelhas e outros insetos, despertam a atenção de pesquisadores. Isto se deve ao fato de como estes seres conseguem solucionar problemas de alocações de energia e recursos necessários à sua sobrevivência, tal como observados em cardumes de peixes se defendendo de predadores ou de formigas em busca de alimento. O curioso é que, cada ser do enxame mesmo sem possuir um sistema desenvolvido de raciocínio, quando juntos, apresentam um padrão complexo de ações inteligentes onde conseguem resolver de maneira elegante os seus problemas. Este comportamento foi batizado de inteligência de enxame pela comunidade científica (BONABEAU, 1999).

A inteligência de enxame é um processo no qual o padrão a um nível global emerge apenas devido às numerosas interações entre os componentes do sistema. As regras que especificam as interações entre os componentes do sistema são executadas utilizando apenas informação local, sem referência ao padrão global.

Diversos pesquisadores da área de inteligência artificial se utilizam dessa filosofia para instruir unidades simples e autônomas de hardware (KAZADI, 2000) e software (GROSAN, 2006) a terem comportamentos complexos que possam emergir da interação dessas unidades, e assim solucionar algum problema específico.

Um dos primeiros trabalhos relacionado ao tema foi proposto por Reynolds (1987), o qual buscava reproduzir de maneira simulada em computador o comportamento emergente

CAPÍTULO IV

do movimento observado nos voos de certos pássaros (ou Boyds como o autor os denominou) em revoada.

No trabalho de Reynolds (1987) foi descrito um processo de animação comportamental que partia da suposição de que cada indivíduo é um agente autônomo que se comporta de acordo com sua percepção do ambiente gerando um sistema auto organizado, este comportamento poderia ser descrito utilizando apenas 3 regras simples para cada indivíduo:

- a) Controle de colisões, uma forma de garantir uma separação física entre os elementos do bando em revoada;
- b) Controle de velocidade, garantindo que a velocidade de um pássaro seja similar aos seus vizinhos mais próximos;
- c) Controle de posição, garantindo que os pássaros fiquem próximo uns aos outros.

As simulações desenvolvidas por Reynolds (1987), inspiraram Heppner e Grenander (1990) a desenvolverem um modelo mais geral que levasse em conta o comportamento social das aves em busca de comida e de um lugar seguro para construção do seu ninho.

Os pássaros do modelo exibido por Heppner e Grenander (1990) apresentavam o mesmo comportamento de revoada apresentado no modelo de Reynolds (1987), mas eles adicionaram algo diferente: os pássaros eram atraídos para uma área de pouso que pudesse favorecer a oferta de alimento e segurança ao bando.

Nesse modelo, os pássaros inicialmente começariam a voar sem um destino em particular e espontaneamente uma revoada estaria formada a partir do momento em que um dos pássaros identificasse uma área promissora de pouso e partilhasse esta informação com o restante do bando. No início do voo, eles não sabem onde se encontra o melhor local, mas a partir da análise individual de satisfação de cada pássaro em relação ao ambiente e dos mecanismos de comunicação social, o melhor local encontrado é difundido para todo o bando e rapidamente existe a convergência de consensos a partir da ponderação da análise individual de cada pássaro e da coletividade.

As regras que modelam o comportamento do bando de pássaros de Heppner e Grenander (1990) são simples:

- os pássaros escolhem a direção e a velocidade de voo (essencialmente, cada pássaro tenta permanecer próximo aos outros enquanto também tenta não colidir com os mesmos);

CAPÍTULO IV

- um dos pássaros se desvia do restante do bando com o intuito de pousar num poleiro, o que acarreta num movimento dos pássaros mais próximos para mesma direção;
- quando esses pássaros descobrem um poleiro, eles pousam conduzindo o restante do bando para o mesmo local.

Pelo fato de um pássaro encontrar o alimento, ou o local para o ninho, as chances de os outros pássaros encontrarem esse mesmo local aumenta consideravelmente, isto se deve à existência de uma inteligência social, ou seja, cada *indivíduo* aprende não só com os seus acertos, mas também com os acertos dos demais do bando.

Encontrar um poleiro é análogo a encontrar uma solução ótima em um espaço de soluções possíveis, e a maneira como uma das aves "conduz" seus vizinhos em direção a este ponto determina o sucesso do processo.

Kennedy e Eberhart (1995) perceberam o potencial da heurística sócio-comportamental envolvida na descrição do voo dos pássaros de Heppner e Grenander (1990) e desenvolveram um algoritmo para solução de problemas de otimização irrestrita, este algoritmo ficou conhecido como Otimização por Enxame de Partículas (OEP).

Portanto, em termos computacionais, a OEP consiste num algoritmo iterativo baseado numa população de indivíduos denominado *enxame*, no qual cada indivíduo, denominado *partícula*, sobrevoa o espaço de decisão interagindo entre si e com o meio ambiente em busca de soluções ótimas. Com base nas propriedades de auto-avaliação, comparação e imitação, os indivíduos são capazes de lidar com um número de possíveis situações que os ambientes lhes apresentam. Os comportamentos globais serão, portanto, resultados emergentes dessas interações.

Devido às suas muitas vantagens, incluindo a simplicidade, a implementação fácil, eficiência computacional e uma fácil adaptação a programação paralela, a OEP tem sido considerada uma técnica de otimização de propósito geral amplamente aplicada em diversas áreas, como otimização discreta e contínua (KAUR *et al.*, 2016), regressão e classificação (PRADEEPKUMAR *et al.*, 2017), reconhecimento de padrões (KUO *et al.*, 2011 e ALSWAITTI *et al.*, 2018), aprendizagem de máquina (QOLOMANY *et al.*, 2017) , treinamento de redes neurais (NING *et al.*, 2017) etc.

Para estimar a importância dada a técnica de OEP pela comunidade científica em geral foi realizada uma busca nas 5 principais bases de periódicos científicos disponibilizados no portal CAPES (SCOPUS, Google Scholar, Springer, Web of Science e

CAPÍTULO IV

IEEE Xplore) a respeito do tema utilizando a expressão “particle swarm optimization” (no título do artigo), revelando uma tendência crescente de publicações a respeito do tema desde a sua gênese até os dias atuais. O número de artigos publicados usando a técnica de OEP listadas nas bases de dados em questão estão mostradas na Tabela 4.1.

Tabela 4.1 – Evolução temporal do número de publicação anual a respeito da técnica de OEP pelas principais bases de dados listadas no portal CAPES

Ano	Scopus	Google scholar	Springer	Web of science	IEEE xplore
2017	1333	1600	286	606	1101
2016	1535	1980	219	1110	1773
2015	1521	2100	207	1074	1570
2014	1702	2300	220	1021	1513
2013	1624	2310	206	895	1390
2012	1827	2640	231	954	1494
2011	1625	2340	162	696	1492
2010	1745	2360	188	743	1950
2009	1557	2070	131	980	1827
2008	1140	1530	86	790	1301
2007	882	1170	89	619	842
2006	645	896	121	483	641
2005	358	459	59	238	325
2004	193	233	11	123	236
2003	57	92	5	58	105
2002	44	53	3	28	75
2001	8	15	0	8	17
2000	5	9	0	4	11
1999	5	10	0	1	10
1998	6	5	3	1	7
1997	3	1	0	1	3
1996	0	0	0	0	0
1995	1	2	0	1	4

Fonte: Autoria própria (2017)

A aplicação da OEP foi explorada em diversos artigos publicados em vários periódicos nas mais diversas áreas do conhecimento, demonstrando a plasticidade de utilização que esta técnica apresenta. A Tabela 4.2 mostra, segundo um levantamento na base Scopus, os periódicos que mais publicam artigos baseados na técnica de OEP. Quando os artigos são classificados de acordo com os periódicos em que são publicados, o “*Lecture Notes In Computer Science Including Subseries Lecture Notes In Artificial Intelligence And*

CAPÍTULO IV

Lecture Notes In Bioinformatics” é encontrado com o maior número de artigos (812 artigos). Os periódicos “*Applied Mechanics And Materials*”, “*Advanced Materials Research*”, “*Applied Soft Computing Journal*” e “*Expert Systems With Applications*” vem em sequência apresentando também um número expressivo de artigos relacionados a OEP. Pode-se observar que os periódicos em questão são fortemente inclinados a resolver problemas de engenharia com o auxílio do computador.

Tabela 4.2 – Número de publicações por periódico.

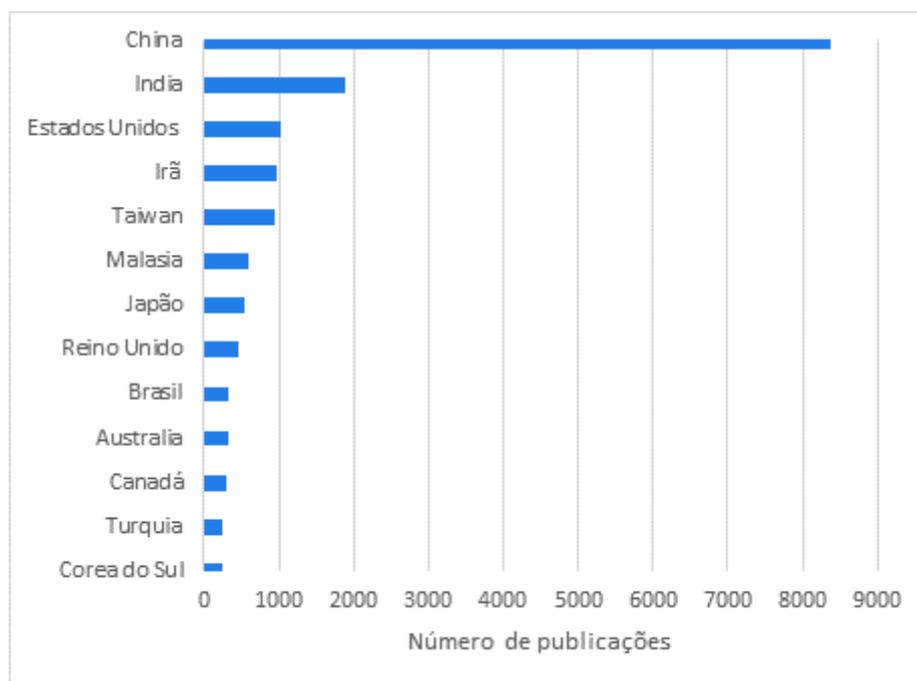
Periódico	Número de publicações
Lecture Notes In Computer Science Including Subseries Lecture Notes In Artificial Intelligence And Lecture Notes In Bioinformatics	812
Applied Mechanics And Materials	231
Advanced Materials Research	220
Applied Soft Computing Journal	196
Expert Systems With Applications	176
Kongzhi Yu Juece Control And Decision	155
Advances In Intelligent Systems And Computing	144
Proceedings Of The World Congress On Intelligent Control And Automation WCICA	131
Communications In Computer And Information Science	115
Lecture Notes In Electrical Engineering	111
Xitong Fangzhen Xuebao Journal Of System Simulation	106
Journal Of Computational Information Systems	103
Studies In Computational Intelligence	103
Mathematical Problems In Engineering	102

Fonte: Autoria própria (2017)

Executando um levantamento na base Scopus em termos do número de publicações por país, é possível perceber que a técnica de OEP recebeu uma maior atenção de pesquisadores chineses (com 8385 publicações) ao longo dos anos, sendo o Brasil o nono país aparecendo no ranking com 326 publicações, como descrito na Figura 4.1. Já se o critério for a área de atuação pode-se ver a partir da Tabela 4.3 que a técnica OEP se difundiu para diversas áreas, mas mantendo um forte laço com as áreas da computação aplicada, engenharias e matemática.

CAPÍTULO IV

Figura 4.1 – Número de publicações da técnica OEP por país.



Fonte: Autoria própria (2017)

Tabela 4.3 – Distribuição de publicações da técnica OEP por área do conhecimento.

Área	Número de publicações
Ciência da computação	9991
Engenharias	9402
Matemática	4246
Recursos Energéticos	1369
Física e Astronomia	1131
Ciências dos materiais	848
Processos decisórios	596
Ciências da terra	467
Ciências sociais	408
Ciências Ambientais	380
Engenharia química	330
Bioquímica e Genética	309
Negócios	284
Química	255
Medicina	218
Ciências Biológicas	166
Outras áreas	549

Fonte: Autoria própria (2017)

CAPÍTULO IV

4.2. ALGORITMO COMPUTACIONAL

O procedimento de otimização é descrito como a ação de determinar a melhor escolha em um conjunto de parâmetros disponíveis que quando substituídos em uma função específica possa aproximar uma solução da maneira mais próxima possível a um valor desejado. A técnica desenvolvida por Kennedy e Eberhart (1995) trata-se de um processo heurístico estocástico baseado no conceito de interação social, buscando a otimização de uma função de aptidão por meio da troca de informações entre indivíduos de uma população.

O algoritmo utiliza agentes (partículas) que constituem um enxame movendo-se em um espaço de busca procurando por uma solução ótima. Cada partícula é tratada como um ponto em um espaço n-dimensional que ajusta seu “voo” de acordo com sua própria experiência assim como com a experiência das outras partículas do enxame.

Sua posição é ajustada com base na própria experiência e na experiência do grupo. Sendo a nova posição das partículas calculadas a partir das suas posições atuais e o deslocamento realizado dado pelas velocidades obtidas no processo iterativo.

$$v_i^{k+1} = v_i^k + r_1 c_1 [p_i^k - x_i^k] + r_2 c_2 [g_{best}^k - x_i^k] \quad (4.1)$$

$$x_i^{k+1} = x_i^k + v_i^{k+1} \quad (4.2)$$

sendo v_i^k e x_i^k respectivamente a velocidade e a posição atual da partícula i na iteração k , c_1 e c_2 coeficientes comportamentais, r_1 e r_2 números aleatórios entre 0 e 1, p_i^k a melhor posição já alcançada pela partícula i e g_{best}^k a melhor posição encontrada pelo enxame até a iteração k .

O processo de otimização é movido pelo vetor velocidade, Equação (4.1), que reflete o consenso de conhecimento da partícula com base na sua própria experiência (fator cognitivo) e na experiência do grupo (fator social). A cada iteração, a posição de cada partícula é atualizada, conforme Equação (4.2).

Segundo Cleghorn (2013), a Equação (4.1) corresponde à soma de três termos distintos: inércia, memória e cooperação.

- ❖ O termo v_i^k é chamado componente de inércia, ele fornece uma tendência da direção de voo seguida pela partícula i . Este componente impede alterações drásticas na direção das partículas, impedindo assim a polarização das direções de propagação do enxame.

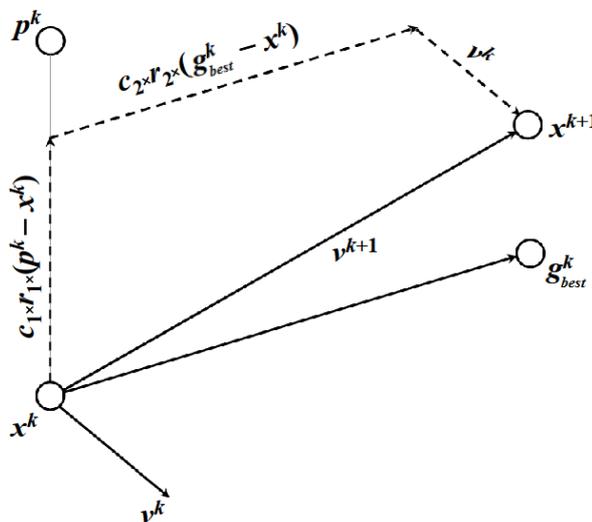
CAPÍTULO IV

- ❖ O termo $r_1 c_1 [p_i^k - x_i^k]$ é chamado de componente cognitivo e está relacionado ao aprendizado individual da partícula em relação ao melhor ponto que já encontrado no passado. O componente cognitivo representa uma memória das partículas em relação às posições mais promissoras obtidas no passado (atrator local). Kennedy e Eberhart (1995) batizaram este termo com o título de “termo nostálgico”, devido a analogia da ação desta parcela do algoritmo com a tendência psico-social de regressarmos aos locais que possuem mais associações positivas em nossa memória.
- ❖ O termo $c_2 r_2 [g_{best}^k - x_i^k]$ é chamado de componente social e representa a troca de experiências entre um grupo de partículas com os seus vizinhos. O efeito do componente social é que cada partícula compartilha com os seus vizinhos a melhor posição encontrada pelo exame (atrator global). Kennedy e Eberhart (1995) se referem ao componente social como "termo de imitação" associando a ideia de que as demais partículas querem desfrutar do mesmo sucesso obtido por aquela partícula que conseguiu localizar o mínimo.

Podemos perceber na Figura (4.2) uma representação pictórica do processo de determinação de uma nova posição de uma partícula (x_i^{k+1}) a partir da sua posição anterior (x_i^k) e do conhecimento difundido pelo enxame.

É importante ressaltar que no enfoque do algoritmo, os termos “cognitivo”, “social”, “nostalgia” e “inveja” implicam apenas metáforas, o que é diferente da modelagem formal de tais conceitos.

Figura 4.2 - Atualização de uma partícula.



Fonte: Adaptado de Viveros (2007)

CAPÍTULO IV

Os passos para a implementação do algoritmo OEP são descritos em Eberhart et. al. (1996) sintetizados no Pseudo-código (4.1):

Pseudocódigo 4.1 - Algoritmo OEP Original.

```
1 Inicialização das constantes  $n, m, c_1, c_2, k_{max}, \epsilon$ 
2 Inicialização randômica das variáveis  $x_i$  e  $v \in R^n$ ;
3 Inicialização do valor de  $f_g^1$ 
4 para  $i$  de 1 até  $n$  faça
5    $f_i^1 = f(x_i)$ ;
6    $p_i = x_i$ 
7   se  $f_i^1 \leq f_g^1$  então
8      $f_g^1 = f_i^1$ 
9      $g_{best}^1 = x_i^1$ 
10  fim se
11 fim para
12 saída  $\leftarrow$  "falso"
13 repita
14  para  $i$  de 1 até  $n$  faça
15    atualização da velocidade da partícula  $i$  de acordo com a Equação (4.1);
16    atualização da posição da partícula  $i$  de acordo com a Equação (4.2);
17     $f_i^k = f(x_i^k)$ ;
18    se  $f_i^k \leq f_p^k$  então
19       $f_p^k = f_i^k$ 
20       $p_i^k = x_i^k$ 
21    senão
22       $f_p^k = f_i^{k-1}$ 
23       $p_i^k = p_i^{k-1}$ 
24    fim se
25    se  $f_i^k \leq f_g^k$  então
26       $f_g^k = f_i^k$ 
27       $g_{best}^k = x_i^k$ 
28    senão
29       $f_g^k = f_g^{k-1}$ 
30       $g_{best}^k = g_{best}^{k-1}$ 
31    fim se
32  fim para
33  se condição de parada atendida então
34    saída = "verdade"
35     $k = k + 1$ 
36  fim se
37 até que saída
```

Resultado: assume-se que g_{best}^k é a melhor solução encontrada para o problema.

Fonte: Adaptado de Eberhart et. al. (1996)

CAPÍTULO IV

4.3. PARÂMETROS DO ALGORITMO

O algoritmo OEP possui parâmetros que podem afetar seu desempenho, onde devem ser ajustados para cada problema a ser otimizado. Alguns dos valores e escolhas desses parâmetros influenciam diretamente na sua eficiência. Os parâmetros básicos do OEP são o tamanho do enxame, o número de iterações, os coeficientes comportamentais além de outros que serão apresentados com os métodos de Melhoria da Convergência do algoritmo apresentados mais à frente.

4.3.1. Tamanho do enxame

Tamanho do enxame ou tamanho da população é o número de partículas no enxame. Quanto maior o enxame, mais partes do espaço de pesquisa podem ser cobertos por cada iteração.

O uso de muitas partículas pode reduzir a necessidade de um número elevado de iterações afim de se obter um bom resultado de otimização. Em contraste, quantidades enormes de partículas aumentam a complexidade computacional por iteração e consomem mais tempo.

Deve-se então equilibrar o número de partículas com o número de iterações para que se possa reduzir o custo computacional do algoritmo.

4.3.2. Número de iterações

Para se obter um bom resultado de convergência, deve-se ter um número de iterações apropriado pois o resultado será tão próximo do resultado esperado conforme o número de iterações realizadas. Um número muito baixo de iterações pode parar o processo de busca prematuramente, enquanto iterações muito grandes tem a consequência da complexidade computacional adicional desnecessária implicando em mais tempo de processamento.

4.3.3. Coeficientes comportamentais

Os coeficientes comportamentais c_1 e c_2 , juntamente com os valores aleatórios r_1 e r_2 , mantêm a influência estocástica dos componentes cognitivo e social da velocidade da partícula. A constante c_1 expressa quanta confiança uma partícula tem em si mesma, enquanto c_2 expressa quanta confiança uma partícula tem em seus vizinhos. Podemos

CAPÍTULO IV

observar algumas particularidades no comportamento do enxame a partir da variação de c_1 e c_2 :

- ❖ Quando $c_1 = c_2 = 0$, então, todas as partículas continuam a voar à sua velocidade atual até atingirem o limite do espaço de busca. Portanto, a partir da Equação (4.1), a equação de atualização de velocidade é calculada como

$$v_i^{k+1} = v_i^k \quad (4.3)$$

- ❖ Quando $c_1 > 0$ e $c_2 = 0$, todas as partículas são independentes. A equação de atualização de velocidade será

$$v_i^{k+1} = v_i^k + r_1 c_1 [p_i^k - x_i^k] \quad (4.4)$$

- ❖ Quando $c_1 = 0$ e $c_2 > 0$, todas as partículas são atraídas para um único ponto em todo o enxame, e a velocidade de atualização se tornará

$$v_i^{k+1} = v_i^k + r_2 c_2 [g_{best}^k - x_i^k] \quad (4.5)$$

- ❖ Quando $c_1 = c_2$, todas as partículas são atraídas na média para o valor de g_{best}^k .
- ❖ Quando $c_1 \gg c_2 > 0$, cada partícula é mais fortemente influenciada pela sua melhor posição pessoal, resultando em uma baixa agregação. Em contraste, quando $0 < c_1 \ll c_2$, então, todas as partículas são muito mais influenciadas pela melhor posição global, o que faz com que todas as partículas converjam prematuramente a um ótimo local.

Normalmente, c_1 e c_2 são estáticos, com seus valores otimizados sendo encontrados empiricamente. Uma inicialização errada de c_1 e c_2 pode resultar em comportamento divergente ou cíclico.

4.4. MELHORIA DA CONVERGÊNCIA DO ALGORITMO

No algoritmo OEP, a velocidade da partícula desempenha um papel muito importante, já que ela define o tamanho do passo e a direção do enxame. Em cada passo, as partículas movimentam-se de maneira coordenada em cada dimensão do espaço de busca a partir do ajuste adequado das suas próprias experiências e a das suas vizinhas em um

CAPÍTULO IV

processo emergente. Existem duas características a serem ressaltadas neste processo: a exploração global e a exploração local.

Exploração global é a capacidade de explorar diferentes áreas do espaço de busca para localizar pontos que possam indicar pontos promissores para o processo de otimização, enquanto a exploração local é a capacidade de concentrar a busca em torno de uma área de busca para refinar uma solução. Portanto, essas duas características têm que se equilibrar em um bom algoritmo de otimização. Quando se aumenta muito o módulo da velocidade, a posição da partícula aumenta proporcionalmente. Como resultado, as partículas deixam os limites do espaço de busca e divergem. Portanto, para controlar essa divergência, as velocidades das partículas são reduzidas para permanecerem dentro das restrições de limites.

Foram desenvolvidas diversas técnicas na literatura da área como podemos ver em Carlisle (2001) para equilibrar o processo de exploração global - exploração local, encontrando-se assim uma solução com as qualidades de convergência requerida e custo computacional limitado.

3.4.1. Peso de inércia

O equilíbrio entre a exploração global e local ao longo do processo iterativo é fundamental para o sucesso de um algoritmo de otimização. Quase todos os algoritmos estocásticos utilizam algum mecanismo para atingir esse objetivo. Com este propósito, Shi e Eberhart (1998a) propuseram uma variação no algoritmo original introduzindo um coeficiente chamado de peso de inércia (ω) o qual pondera a influência das velocidades obtidas na interação anterior na atual, controlando o deslocamento da partícula. A equação de velocidade da partícula passará a ser definida pela Equação (4.6)

$$v_i^{k+1} = \omega v_i^k + r_1 c_1 [p_j^k - x_j^k] + r_2 c_2 [g_{best}^k - x_i^k] \quad (4.6)$$

O peso de inércia foi inicialmente introduzido por Shi e Eberhart (1998a) para reduzir as magnitudes das velocidades das partículas ao longo do tempo (iterações), controlando as capacidades de exploração global e exploração local do enxame e assim criar condições de convergência com uma maior precisão e eficiência em comparação ao obtido com o uso da equação (4.4).

Se $\omega \geq 1$, então as magnitudes das velocidades aumentam ao longo do tempo e as partículas dificilmente podem mudar sua direção para retornar a uma região promissora

CAPÍTULO IV

dificultando o processo de exploração local, e o enxame então diverge. Se $\omega = 1$, caímos na mesma situação descrita pelo algoritmo clássico. Se $0 \leq \omega \leq 1$, então as magnitudes das velocidades tendem a diminuir ao longo do tempo e as partículas tendem a estagnar em regiões que possuam mínimos locais, dificultando o processo de exploração global, implicando em uma convergência prematura do enxame. Se $\omega = 0$, o termo inercial das partículas desaparece e todas as partículas se movem sem o conhecimento da velocidade anterior em cada passo.

O peso de inércia pode ser implementado como um valor fixo ou modificado dinamicamente como mostrado em Amoshahy *et al.* (2016). Shi e Eberhart (1998a) inicialmente propuseram implementações de ω com valor fixo para todas as partículas, mas trabalhos mais recentes, como por exemplo o de Silveira (2009), exploram a possibilidade de se utilizar valores dinâmicos para ω devido a influência deste parâmetro em controlar a exploração local e exploração global do espaço de busca.

Normalmente, inicia-se o algoritmo com um valor maior que a unidade para ω , o que permite que todas as partículas se movam livremente no espaço de busca durante as primeiras iterações e gradualmente vai-se diminuindo o seu valor ao longo do tempo dando a possibilidade ao algoritmo de refinar o estudo de regiões promissora facilitando assim a exploração local. Portanto, o processo muda o foco do modo exploratório global para o modo exploratório local controlando o equilíbrio entre a exploração global e local,

Um dos principais esquemas de atualização do peso de inércia é apresentado no trabalho de Eberhart e Shi, (2001), onde o valor de ω é reduzido de maneira linear com o aumento do número de iteração de acordo com a seguinte equação:

$$\omega^k = \omega_{max} - \left(\frac{\omega_{max} - \omega_{min}}{k_{max}} \right) k \quad (4.7)$$

Onde, w_{max} e w_{min} são os valores inicial e final do peso de inércia, respectivamente, k_{max} é o número máximo de iterações, k é o número da iteração atual.

3.4.2. Coeficiente de constrição

Esta técnica introduziu um novo parâmetro, “ χ ”, conhecido como fator de constrição. O coeficiente de constrição foi desenvolvido por Clerc e Kennedy (2002). Este coeficiente é extremamente importante para controlar a ação de exploração global e exploração local,

CAPÍTULO IV

para garantir o comportamento de convergência do algoritmo. A equação de atualização de velocidade proposta por Clerc e Kennedy (2002) é calculada da seguinte forma:

$$v_i^{k+1} = \chi(v_i^k + r_1 c_1 [p_i^k - x_i^k] + r_2 c_2 [g_{best}^k - x_i^k]) \quad (4.8)$$

Onde

$$\chi = \begin{cases} \frac{2\kappa}{|2 - \phi - \sqrt{\phi^2 - 4\phi}|} & p/ \quad \phi > 4 \\ \kappa & p/ \quad \phi < 4 \end{cases} \quad (4.9)$$

sendo $\phi = r_1 c_1 + r_2 c_2$ e $\in [0;1]$. A constante que controla a taxa de convergência. Para $\kappa \approx 0$, obtém-se convergência rápida para um ponto estável, enquanto que $\kappa \approx 1$ resulta em convergência lenta.

Clerc e Kennedy (2002) verificaram que quando $\phi < 4$, o enxame entra em um movimento lento e espiralado em torno de um possível mínimo local sem garantia de convergência, enquanto para $\phi > 4$ a convergência torna-se rápida e garantida devido a redução da amplitude da oscilação da partícula focalizando os melhores pontos locais e globais. Se a melhor posição da partícula e a melhor posição da vizinhança estiverem próximas uma da outra, então as partículas realizarão uma busca local. Por outro lado, se suas posições estiverem distantes umas das outras, então as partículas irão realizar uma busca global. O coeficiente de constrição garante convergência das partículas ao longo do tempo e também evita o colapso.

A desvantagem do coeficiente de constrição é que, se a melhor posição pessoal de uma partícula e a melhor posição global estiverem muito afastadas umas das outras, as partículas podem seguir ciclos mais amplos e não convergir (DAS, 2008).

3.4.3. Condições de fronteira

Os métodos mencionados anteriormente, constituem mecanismos de controle da evolução das velocidades com o intuito de evitar que as partículas se desagreguem e percam a capacidade de equilibrar o processo de busca global-local, mas eles não garantem que as partículas fiquem restritas a um domínio pré-definido anteriormente, para isso deve-se existir algum mecanismo que limite o espaço de busca, evitando a divergência do enxame. Em outras palavras, as partículas podem ocasionalmente voar para uma posição para além do espaço de busca definido e gerar uma solução fora do domínio físico (HUANG, 2005).

CAPÍTULO IV

Há duas dificuldades principais ligadas às técnicas de controle de velocidade apresentadas anteriormente: primeiro, a escolha de valores adequados para os parâmetros de controle não é uma tarefa trivial e em segundo lugar, as técnicas apresentadas anteriormente não garantem que as partículas permaneçam dentro do espaço viável de busca o tempo todo.

Portanto, o algoritmo deve ser codificado com instruções claras sobre como superar esta situação, e tais instruções são chamadas de condição de fronteira do algoritmo OEP. Para resolver este problema, diferentes propostas de condições de fronteiras foram propostas, Mikki (2005) e Xu (2007) discutem bem as características que distinguem cada condição e como implementá-las computacionalmente.

Estas condições de fronteira formam dois grandes grupos: fronteiras restritas (nomeadamente, absorção, reflexão e amortecimento) e fronteiras sem restrições (nomeadamente, invisíveis, invisíveis/refletoras, invisíveis/amortecedoras). As seis condições de fronteiras são discutidas abaixo:

- a) Na situação de **Fronteiras Restritas**, quando uma partícula vai para fora do espaço de solução em uma das suas dimensões, a partícula é deslocada para a parede do espaço que foi ultrapassada além disso, para cada condição de fronteira listada abaixo, é feita distinções a respeito da dinâmica da partícula quando este fato ocorre.
 - ❖ Condição de Fronteira Refletiva (CFR): A velocidade da partícula é alterada no sentido oposto dessa dimensão tal como ilustrado no pseudocódigo 02. Isso significa que, a partícula é refletida por uma parede dura e, em seguida, ela volta para o espaço de solução.
 - ❖ Condição de Fronteira Amortecida (CFAM): A velocidade da partícula é ajustada no sentido oposto dessa dimensão multiplicada por um coeficiente aleatório entre 0 e 1 como ilustrado no pseudocódigo 03. Este mecanismo é similar a condição anterior, exceto que parte da energia é determinada aleatoriamente por causa da reflexão imperfeita.
 - ❖ Condição de Fronteira Absortiva (CFA): A velocidade da partícula é ajustada a zero nessa dimensão como ilustrado no pseudocódigo 04. Isto significa que, nesta condição, a partícula realiza um choque totalmente inelástico e a sua energia cinética nesta dimensão é absorvida de modo a ficar aderida na parede do espaço de solução até ser atraída para o interior do espaço de solução.

Pseudocódigo 4.2 - Condição de Fronteira Refletivo.

```
1 se  $x_i^k \geq x_{max}$  então
2    $x_i^k \leftarrow x_{max}$ 
3    $v_{ij}^k \leftarrow -v_{ij}^k$ 
4 senão
5 se  $x_i^k \leq x_{min}$  então
6    $x_i^k \leftarrow x_{min}$ 
7    $v_{ij}^k \leftarrow -v_{ij}^k$ 
8 fim se
9 fim se
```

Fonte: Adaptado de Xu (2007)

Pseudocódigo 4.3 - Condição de Fronteira Amortecido.

```
1 se  $x_i^k \geq x_{max}$  então
2    $x_i^k \leftarrow x_{max}$ 
3    $v_{ij}^k \leftarrow -r * v_{ij}^k$ 
4 senão
5 se  $x_i^k \leq x_{min}$  então
6    $x_i^k \leftarrow x_{min}$ 
7    $v_{ij}^k \leftarrow -r * v_{ij}^k$ 
8 fim se
9 fim se
```

Fonte: Adaptado de Xu (2007)

Pseudocódigo 4.4 - Condição de Fronteira Absortivo.

```
1 se  $x_i^k \geq x_{max}$  então
2    $x_i^k \leftarrow x_{max}$ 
3    $v_{ij}^k \leftarrow 0$ 
4 senão
5 se  $x_i^k \leq x_{min}$  então
6    $x_i^k \leftarrow x_{min}$ 
7    $v_{ij}^k \leftarrow 0$ 
8 fim se
9 fim se
```

Fonte: Adaptado de Xu (2007)

CAPÍTULO IV

- b) Na situação de **Fronteiras sem Restrições**, considera-se uma partícula ainda no algoritmo mesmo está estando fora do espaço de solução viável, mais senso atribuído a ela um valor de aptidão ruim, além disso, para cada condição de fronteira listada abaixo, existem distinções a respeito dos mecanismos de regresso dessas partículas para o espaço de solução.
- ❖ **Condição de Fronteiras Invisíveis (CFI):** Mantem-se inalterado o valor da velocidade da partícula como ilustrado no pseudocódigo 0 e espera-se que a atração de melhores posições pessoais e globais possam neutralizar o avanço da partícula para fora dos limites de fronteira e, finalmente, puxá-la de volta para dentro do espaço da solução.
 - ❖ **Condição de Fronteira Invisível/Refletor (CFIR):** Altera-se o sinal da velocidade da partícula na direção oposta nessa dimensão de modo que o momento da partícula seja invertido para acelerá-la de volta para o espaço de solução viável.
 - ❖ **Condição de Fronteira Invisível/Amortecido (CFIA):** A velocidade da partícula é alterada no sentido oposto nessa dimensão multiplicada por um coeficiente aleatório entre 0 e 1, de modo que o momento inverso da partícula a acelere de maneira amortecida de volta para dentro do espaço de solução.

Pseudocódigo 4.5 - Condição de Fronteira Invisível.

```
1 se  $(x_{ij}^k \geq x_{max}$  ou  $(x_{ij}^k \leq x_{min}$  então  
2  $f_i^k \leftarrow \infty$   
3 fim se
```

Fonte: Adaptado de Xu (2007)

Pseudocódigo 4.6 - Condição de Fronteira Invisível/Refletor.

```
1 se  $(x_{ij}^k \geq x_{max}$  ou  $(x_{ij}^k \leq x_{min}$  então  
2  $f_i^k \leftarrow \infty$   
3  $v_{ij}^k \leftarrow -v_{ij}^k$   
4 fim se
```

Fonte: Adaptado de Xu (2007)

Pseudocódigo 4.7 - Condição de Fronteira Invisível/Amortecido.

```
1 se  $(x_{ij}^k \geq x_{max}$  ou  $(x_{ij}^k \leq x_{min}$  então
2    $f_i^k \leftarrow \infty$ 
3    $v_{ij}^k \leftarrow -r * v_{ij}^k$ 
4 fim se
```

Fonte: Adaptado de Xu (2007)

4.5. CRITÉRIOS DE CONVERGÊNCIA

Qualquer algoritmo de otimização que se utilize de processos iterativos deve estabelecer critérios de convergência eficientes. Atender ao(s) critério(s) de convergência significa retirar cálculos adicionais ao processo de otimização que não serão necessários após o algoritmo ter encontrado uma solução aceitável para o problema, ou seja, quando uma melhoria significativa na solução encontrada é improvável. Um critério de parada eficiente tanto serve para diminuir o custo computacional como para aumentar a confiança na solução encontrada (INNOCENTE 2006).

Observando que devido à natureza heurística do algoritmo OEP, este pode se deparar com um ponto do domínio que funcione como um atrator local, induzindo assim uma convergência prematura do algoritmo a um mínimo local que não atenda os pré-requisitos enquanto solução do problema, onde devido a estagnação da solução em torno desse ponto, não compensa continuar com o custo computacional para determinar uma solução melhor, ou quando a agregação das partículas em uma dada região do espaço de busca se torna tão alta que o enxame perde a diversidade necessária para explorar regiões promissoras do espaço de busca, sendo necessário a implementação de condições de parada eficientes que possam balancear o custo computacional da solução em relação ao grau de convergência do algoritmo.

Os principais critérios utilizados para analisar a convergência, envolvem:

- Analisar o número máximo de iterações. Deve-se ter cuidado ao escolher um número máximo de iterações, pois em demasia o algoritmo seguirá com um custo computacional muito grande e se por falta o algoritmo terminará prematuramente antes que uma solução ideal possa ser encontrada.
- Analisar quando uma solução aceitável foi encontrada. Esta terminação requer que a solução seja conhecida a priori e é amplamente utilizada como método para testar um algoritmo usando benchmarks.

CAPÍTULO IV

Contudo estes critérios são insuficientes para métodos de base populacional por muitas razões. Algumas destas razões são citadas por Innocente (2006):

- Métodos de base populacional apresentam inúmeros candidatos à solução por iteração;
- A melhor solução encontrada na iteração corrente poderia permanecer imutável por algum tempo antes de sofrer uma melhoria;
- A melhor solução encontrada na corrente iteração e a encontrada na iteração precedente poderia corresponder a partículas diferentes então a distância entre suas localizações não é exatamente uma medida de convergência.

Zielinski *et al* (2007) considera diversas propostas de implementação, onde discute critérios baseados na melhoria da aptidão, na agregação e na movimentação do enxame.

O primeiro procedimento é relacionado ao monitoramento da melhor aptidão do enxame, onde se avalia a possibilidade de melhoria no algoritmo após um número definido de iterações. Considerando a Equação (4.10) como um indicador de convergência

$$\Delta^k = \frac{f(g^{k-1}) - f(g^k)}{f(g^{k-1})} \quad (4.10)$$

se $\Delta^k < \epsilon$ para um número consecutivo de passos, então isso implica que o enxame tenha convergido para g^k . Este critério pode ser considerado uma boa condição de terminação devido ao fato de que se baseia no resultado obtido ao questionamento das partículas ao próprio espaço de pesquisa. No entanto, tem a lacuna de que a busca pode ser prematuramente terminada se algumas das partículas ficam presas em mínimos locais enquanto o resto do enxame ainda está explorando o espaço de busca. Também deve ser dito que esta condição também adiciona complexidade adicional ao algoritmo para avaliar Δ_k .

O segundo relaciona-se à análise da existência da agregação das partículas a um ponto de acumulação após um número definido de iterações. Considerando a Equação (4.11) como um indicador de agregação

$$S^k = \sqrt{\sum_{i=1}^n \left(\frac{(x_i^k - \bar{x}^k)^2}{n-1} \right)} \quad (4.11)$$

CAPÍTULO IV

se $S^k < \epsilon$ para um número consecutivo de passos, então isso implica que o enxame atingiu um elevado grau de agregação e que a taxa de melhoria da solução atingiu um limiar inferior admissível. Mesmo sabendo-se que a agregação de partículas não necessariamente significa a convergência do método, mas apenas uma indicação de que é pouco provável a melhoria na melhor solução encontrada devido à perda de diversidade, indicando uma condição de estagnação do algoritmo.

O terceiro é relacionado como as partículas se movimentam no espaço de busca, dando uma ideia de como as partículas realizam as atualizações da posição entre o passo k e o passo $k + 1$, esta análise dispende uma maior energia devido ao monitoramento das partículas a cada iteração e, portanto, não será considerada devido ao alto custo computacional de implementação.

Apesar das várias estratégias citadas anteriormente visando um aumento do desempenho do algoritmo OEP, a grande limitação deste algoritmo, assim como outros algoritmos heurísticos, é a necessidade de um número elevado de avaliações da função aptidão, dificultado a sua utilização em problemas de tempo quase real. Para contornar este tipo de problema será investigado no próximo capítulo os procedimentos necessários para paralelizar este algoritmo, obtendo com isso uma redução do tempo total de processamento.

CAPÍTULO V

PROGRAMAÇÃO PARALELA

5.1 - INTRODUÇÃO

Ao longo dos anos, os problemas de engenharia tornaram-se cada vez mais complexos e para solucioná-los, diversas técnicas numérico-computacionais foram desenvolvidas com a intenção de resolver os modelos matemáticos oriundos da descrição física destes. A técnica de tomografia capacitiva possui este perfil, ela inicia-se em meados da década de 1980, onde os primeiros algoritmos de reconstrução possuíam baixa resolução espacial devido a carência de unidades de processamento que pudessem estimular a busca de algoritmos que efetuassem a determinação de uma imagem com qualidade em um tempo computacional adequado.

Entretanto, a complexidade computacional desse problema ampliou-se além dos limites de computo das unidades convencionais, demandando a adoção de sistemas que possuam a capacidade de implementar novas estratégias tanto em termos de software quanto de hardware para ampliar a sua capacidade de armazenamento e o seu poder de processamento.

Moore (1965) previa um crescimento da capacidade de processamento dos computadores modernos, o qual dobrar-se-ia a cada 18 meses, isto devia-se a capacidade tecnológica de ampliar a densidade de transistores encapsulados em uma única pastilha de silício pelo mesmo custo unitário. Este crescimento traduzia-se em ganhos exponenciais de velocidade do Clock e no número de instruções executadas.

Porém este crescimento não poderia perpetua-se indefinidamente, pois devido aos limites impostos pela termodinâmica e os de natureza econômica (investimento em pesquisa

CAPÍTULO V

e produção), tornar-se-ia inviável construir uma super unidade computacional de processamento único.

Uma das saídas que a indústria, pesquisadores e desenvolvedores de software e hardware propuseram para superar esta limitação foi a reformulação do paradigma convencional de hardware e software que impunha uma unidade central de processamento (UCP) para uma nova que propunha uma sinergia tanto em termos de software quanto de hardware para implementar de maneira eficiente o paradigma de dividir, conquistar e combinar. Esta área de pesquisa e desenvolvimento ficou conhecida como computação paralela, demandando um grande esforço de implementações eficientes de hardware e software.

Segundo Almeida e Paiva (2005), programação paralela é um paradigma de programação no qual o computador faz uso da execução simultânea de várias tarefas computacionais interativas, que podem ser realizadas como programas separados ou como um conjunto de instruções criadas por um único programa. Essas instruções podem ser executadas por um único processador, vários processadores em um único equipamento ou em diversos processadores distribuídos por uma rede.

5.2. ARQUITETURAS PARALELAS

Segundo Jaquie (1999), pode-se entender por arquitetura paralela, aquelas máquinas que são capazes de executar mais de uma tarefa ao mesmo tempo. Devido ao rápido desenvolvimento das arquiteturas de hardware e software, hoje é disposto em mercado diversas configurações possíveis, para que se possa estudá-las convenientemente é necessário agrupá-las em termos de alguma característica comum. Uma das principais é a forma de como se dá o acesso a memória RAM (do inglês Random Access Memory).

5.2.1. Memória Compartilhada

Sistemas de memória compartilhada são aqueles em que a memória pode ser acessada simultaneamente por vários programas com a intenção de proporcionar a comunicação entre eles ou evitar cópias redundantes. Dependendo do contexto, os programas podem ser executados em um único processador ou em vários processadores separados. É mostrada na figura 5.1 a estrutura esquemática de como se dá o acesso a memória pelos processadores.

CAPÍTULO V

Vantagens:

- Eficiência - É a maneira mais rápida para dois processos efetuarem uma troca de dados. Os dados não precisam ser passados ao kernel para que este os repasse aos outros processos. O acesso à memória é direto.
- Acesso randômico - É possível acessar uma parte específica de uma estrutura de dados que está sendo comunicada.
- Não há necessidade de particionar o código ou dados, logo técnicas de programação convencionais podem ser facilmente adaptados para ambientes multiprocessadores.

Desvantagens:

- Não existe um mecanismo automático (implícito) de sincronização, podendo exigir, por exemplo, o uso de semáforos para controlar ou inibir condições de disputa por acesso ao espaço de memória.
- Apesar de permitir fácil programação, os sistemas deste tipo sofrem problemas de contenção e de alta latência para fazer acesso à memória compartilhada, o que degrada a performance e limita a escalabilidade das aplicações.

5.2.2. Memória Distribuída

Sistemas de memória distribuída são aqueles em que cada processador tem sua própria memória privada. Programas só podem operar localmente sobre os dados disponível na sua memória, se for necessário, deve-se estabelecer um processo de comunicação com um ou mais processadores remotos. Nessa arquitetura cada processador possui sua própria memória e a comunicação entre elas é feita por meio de passagem de mensagem utilizando a rede de comunicação. Sistemas de memória distribuída também são chamados de sistemas de passagem de mensagem. É mostrada na figura 5.2 a estrutura esquemática de como se dá o acesso a memória pelos processadores.

Vantagens:

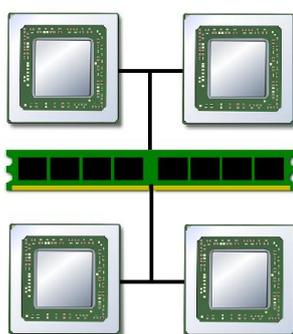
- Altamente escalável e permitem a construção de processadores maciçamente paralelos.
- A comunicação entre os processadores se dá através de troca de mensagens.
- A troca de mensagens resolve tanto o problema da comunicação processadores como o da sincronização.

CAPÍTULO V

Desvantagens:

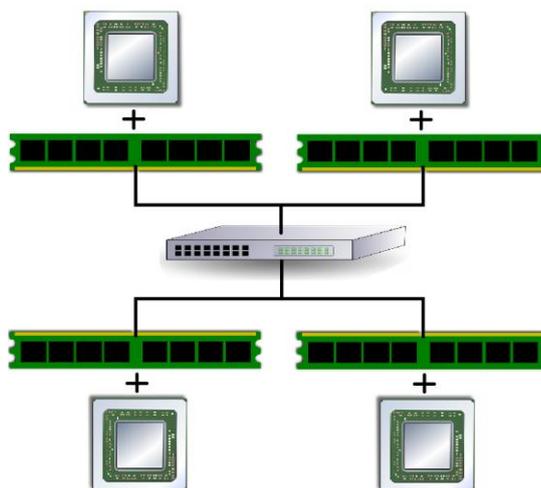
- Necessidade de fazer uma boa distribuição de carga entre os processadores, quer seja automaticamente, quer seja manualmente.
- É necessário evitar as situações de “deadlock” (situação em que um processo aguarda por um recurso que nunca estará disponível ou um evento que não ocorrerá), tanto no nível de aplicação como no nível de aplicação.
- Modelo de programação menos natural.

Figura 5.1 – Configuração esquemática de um sistema de memória compartilhada.



Fonte: Autoria própria (2017)

Figura 5.2 - Configuração esquemática de um sistema de memória distribuída.



Fonte: Autoria própria (2017)

Devido a característica de escalabilidade natural dos sistemas de memória distribuída em relação aos de memória compartilhada, estes sistemas permitem o desenvolvimento de soluções com um maior poder de computação. Entretanto, o projeto de software para este

CAPÍTULO V

tipo de arquitetura é mais complexo, porém o projeto de hardware é mais fácil que em um sistema de memória compartilhada.

A programação em sistemas com memória distribuída é realizada por meio de passagem de mensagens entre as unidades de computação, podendo ser feita com o uso de várias API's, sendo que a mais utilizada é a MPI (Message Passing Interface). O MPI controla todo o fluxo de mensagens (envio, recebimento, reenvio caso necessário, etc.), porém o programador é o grande responsável por definir o que será enviado e quando será enviado.

5.3. MPI

O MPI é uma biblioteca de interface de troca de mensagens para comunicação entre processos em sistemas de computadores paralelos com memória compartilhada ou distribuída (SNIR *et al*, 1996). Segundo Pacheco (1998), o MPI não é uma linguagem de programação e sim um conjunto de especificações que podem ser inseridas em códigos escritos em Fortran, C e C++ e amplamente utilizado no desenvolvimento de aplicativos para máquinas com memória distribuída.

Após várias reuniões com fabricantes, desenvolvedores e cientistas da computação em 1994 a primeira versão do padrão MPI-1 foi proposta. A especificação é mantida por um fórum (o MPI fórum <http://www.mpi-forum.org/>) e atualmente a versão mais recente é a MPI-3.1 lançada em 4 de Junho de 2015. O MPI-1 tratava-se basicamente de comunicação do tipo ponto a ponto, o MPI-2.0 apresentou novas áreas como Entrada/Saída Paralela, operações remotas de memória e gerenciamento dinâmico de processos, além de corrigir e adicionar novas funções ao MPI-1. O MPI-3.0 estendeu as operações sobre conjuntos para adicionar operações não bloqueantes e pequenas correções ao padrão proposto anteriormente. O MPI-3.1 é uma atualização menor para o padrão MPI, a maioria das atualizações são correções e esclarecimentos para o padrão, especialmente para as rotinas em Fortran.

No padrão MPI, uma aplicação é constituída por um ou mais processos que se comunicam entre si, movendo dados entre os seus respectivos espaços de endereçamento a partir da ação de funções de envio e recebimento de mensagens entre os processos. Cada processo roda exatamente o mesmo programa, sendo os dados divididos e distribuídos aos processos participantes, cada processo é identificado por um valor inteiro denominado rank, através de condições de teste sobre o rank dos processos, podemos identificar o remetente e

CAPÍTULO V

o destinatário da mensagem, bem como a porção de código que será executada por cada processador.

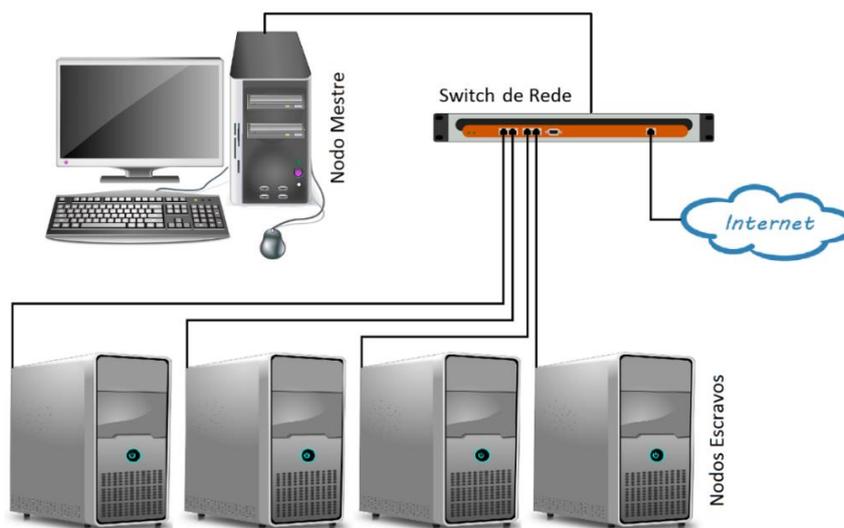
As principais vantagens do estabelecimento de um padrão de troca de mensagens são a portabilidade e a facilidade de utilização. Utilizando o MPI em um ambiente de memória distribuída, o programador tem a possibilidade de escrever as suas rotinas em um nível de abstração mais alto, abrindo mão da necessidade de conhecimento e controle das rotinas de passagem de mensagens.

O maior objetivo do MPI é fornecer um grau de portabilidade através de máquinas diferentes, ou seja, o mesmo código pode ser executado de forma eficiente em uma variedade de máquinas, além de sua habilidade em rodar de forma transparentemente em máquinas heterogêneas.

5.3. CLUSTER BEOWULF

Cluster é um conjunto de computadores individuais conectados por meio de uma rede de comunicação que trabalham em conjunto trocando informações entre si em torno de uma única tarefa, como descrito na Figura 5.3. Estes autômatos dividem entre si as tarefas de processamento e executam este trabalho de maneira simultânea

Figura 5.3 - Configuração típica de um cluster Beowulf.



Fonte: Autoria própria (2017)

Segundo Braga (2003), Beowulf é o nome de um poema que narra as aventuras de um herói lendário da literatura anglo-saxônica cujo poema leva seu nome. Nele Beowulf luta e vence o gigante Grendel em defesa do seu reino. A história serviu de inspiração para os pesquisadores

CAPÍTULO V

Thomas Sterling e Donald Becker batizassem o projeto de cluster no qual trabalhavam de Beowulf.

O cluster Beowulf resultou de um projeto iniciado pela NASA no Verão de 1994 no centro de pesquisas CESDIS, localizado no Goddard Space Flight Center, no âmbito do projeto Earth and Space Sciences, cujo objetivo primário consistia em determinar a aplicabilidade de arquiteturas de processamento paralelo a problemas do domínio das ciências espaciais, nomeadamente para o tratamento de dados recolhidos por satélite, a preços acessíveis (STERLING, 2001).

Na versão literária "Beowulf" enfrenta um poderoso adversário, e vence. Na versão computacional, enfrenta o monopólio das grandes fornecedoras de hardware e software oferecendo uma solução barata à computação de alto desempenho para o meio acadêmico e científico (BRAGA, 2003).

O Beowulf é um cluster baseado no paradigma de memória distribuída, onde processos são particionados por uma unidade central (nodo mestre) em partes menores e enviados às demais máquinas do cluster (nodos escravos) para serem resolvidos e logo após devolvido os resultados ao nodo mestre que irá reunir tudo e apresentar um resultado único como se todos os processos tivessem sido executados em uma única máquina. Todos os processos são realizados por meio de passagem de mensagens, tornando a rede de comunicação um fator importante para o sucesso do sistema.

Segundo Alecrim (2013), apesar dos benefícios, os clusters não são a solução perfeita para todo e qualquer problema computacional, a implementação de um cluster Beowulf apresenta vantagens e desvantagens, dentre muitas podem-se citar:

- a) ótima relação custo-benefício (pode-se obter resultados tão bons quanto ou até superiores que um servidor sofisticado a partir de máquinas mais simples e mais baratas);
- b) não é necessário depender de um único fornecedor ou prestador de serviço para reposição de componentes;
- c) é escalável, sendo possível aumentar a capacidade de um cluster com a adição de nós ou remover máquinas para reparos sem interromper a aplicação;
- d) com um sistema operacional "open-source", é fácil o processo de debug de erros e correções;
- e) a facilidade de expansão do cluster pode ser uma "faca de dois gumes": a quantidade de máquinas pode aumentar tanto que a manutenção se torna mais trabalhosa, o espaço

CAPÍTULO V

físico pode ficar impróprio, necessitar de um sistema de refrigeração devido ao aquecimento dos processadores etc;

- f) A grande maioria do hardware de rede não foi projetada para trabalhar em processamento paralelo;
- g) um cluster tem como base uma rede local, logo, não se pode acrescentar máquinas que estejam muito distantes geograficamente.

5.4. MONTAGEM DO CLUSTER PITÁGORAS

Para o desenvolvimento (prototipagem e teste de execução) dos algoritmos numéricos (versões serial e paralela) foi necessário a montagem de um cluster Beowulf que pudesse atender as necessidades de robustez, disponibilidade, baixo custo de aquisição e manutenção. Desta necessidade nasce o Cluster Pitágoras (CP), conforme descrito na Figura 5.4, cuja proposta é constituir-se como um ambiente de simulação numérica de problemas de multifísica em geral.

Figura 5.4 – Cluster Pitágoras.



Fonte: Autoria própria (2017)

Todo o procedimento de montagem foi realizado no Laboratório de Modelagem Computacional de Alto Desempenho (LAMCAD) localizado no Centro de Ciências da Saúde (CCS) da UFPB, que gentilmente forneceu subsídios para a realização desta tese. Segue-se abaixo as ações de configuração de hardware e software empregados para a construção do CP.

CAPÍTULO V

5.4.1. Configuração de hardware

Para constituição do CP foram utilizados 7 computadores all in one modelo HB-A206 da HP, todos com a mesma configuração, contendo cada um 01 processador Intel(R) Core(TM) i7-4790 com 8 GB de RAM, interligados por um Switch Fast Ethernet de 24 portas da D-LINK modelo DES-1024D conforme pode ser visto na Figura 5.5 e Figura 5.6 respectivamente. Foi adicionado ao nodo mestre uma placa de rede usb modelo realtek 8152, para comunicação com a internet, enquanto a placa de rede integrada serviu para o acesso a rede local intermediada pelo switch.

Figura 5.5 – Máquina utilizada.



Fonte: Autoria própria (2017)

Figura 5.6 – Roteador utilizado.



Fonte: Autoria própria (2017)

5.4.2. Configuração de software

Para implementação do CP utilizou-se como sistema operacional o Linux com a distribuição Ubuntu versão 17.10. Após a sua instalação, foi iniciado a configuração da rede, a habilitação do firewall, a instalação dos pacotes necessários para comunicação, compilação

CAPÍTULO V

e execução dos códigos desenvolvidos nesta tese. Segue-se abaixo a sequência de ações executadas:

5.4.2.1. Configuração da rede

Inicialmente todas as máquinas foram configuradas para que os seus ip's permanecessem estáticos. Para tal, foi acessado o arquivo interfaces contido em cada máquina, conforme comando abaixo, esta alteração é necessária para que o MPI possa gerenciar o envio de mensagens para cada elemento integrante da rede de maneira precisa e inequívoca. Segue abaixo os comandos para configuração.

```
$ sudo nano /etc/network/interfaces
```

(Arquivo interfaces editado)

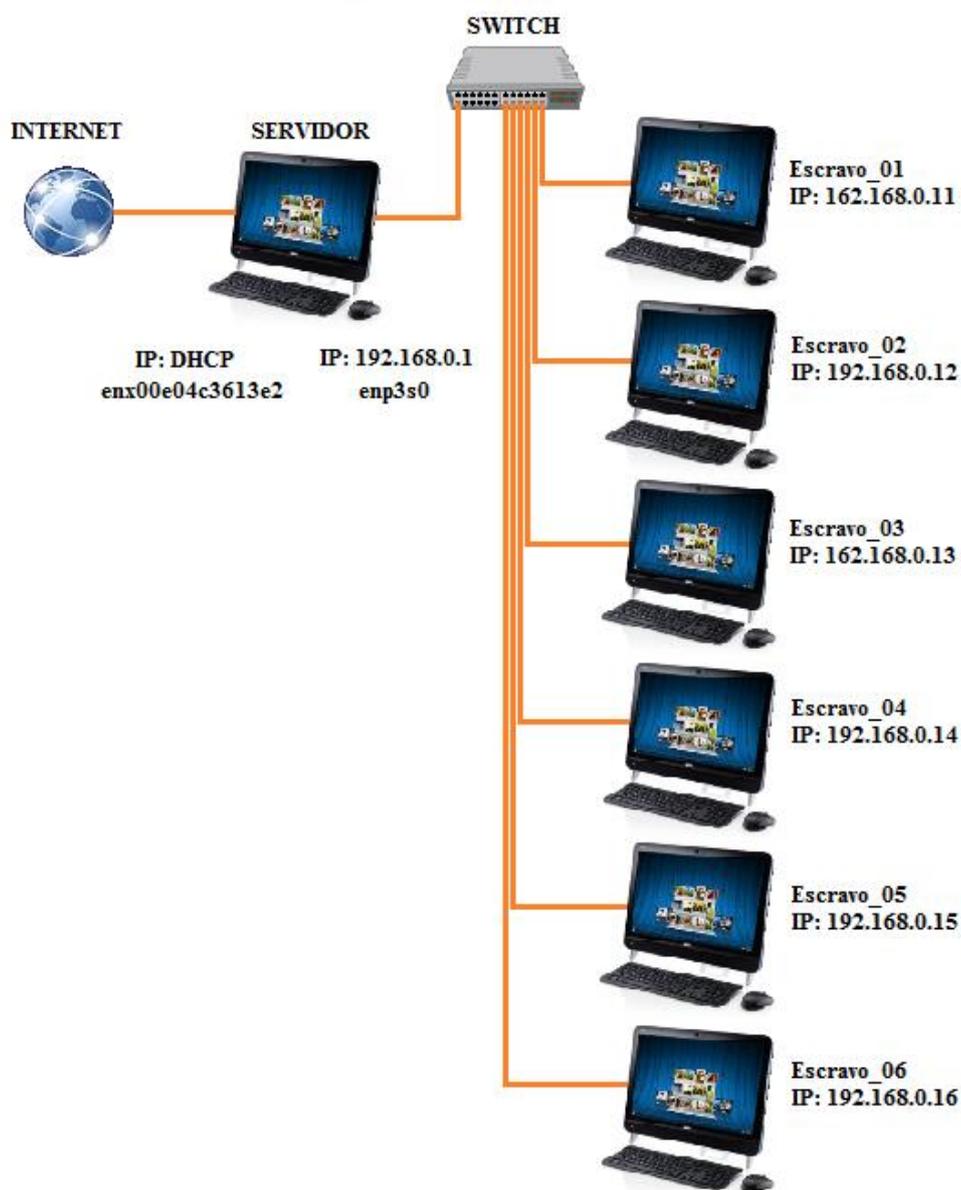
```
a) Nó mestre
auto lo
iface lo inet loopback
# configuração da placa de rede usb
auto enx00e04c3613e2
allow-hotplug enx00e04c3613e2
iface enx00e04c3613e2 inet dhcp
# configuração da placa de rede integrada
auto enp3s0
allow-hotplug enp3s0
iface enp3s0 inet static
address 192.168.0.1
netmask 255.255.255.0
network 192.168.0.0
broadcast 192.168.0.255

b) Nó escravo_01
auto lo
iface lo inet loopback
# configuração da placa de rede integrada
auto enp3s0
allow-hotplug enp3s0
iface enp3s0 inet static
address 192.168.0.11
netmask 255.255.255.0
network 192.168.0.0
broadcast 192.168.0.255
gateway 192.168.0.1
```

CAPÍTULO V

Todos os demais nós escravos possuem uma configuração similar para habilitar a rede com endereçamento estático, devendo-se apenas observar que os nós escravos 01, 02, 03, 04, 05 e 06 possuem terminação 11, 12, 13, 14, 15 e 16 respectivamente para o final dos seus ip's, conforme a descrito na Figura 5.7.

Figura 5.7 – Estrutura da rede utilizada.



Fonte: Autoria própria (2017)

5.4.2.2. Ativação do encaminhamento de pacotes IP no gateway (IP forwarding):

Para que se possa configurar o compartilhamento de Internet é necessário habilitar o encaminhamento de ip's no gateway, habilitando um parâmetro do kernel do sistema

CAPÍTULO V

operacional. Esse parâmetro está localizado no arquivo `sysctl.conf`, acessado conforme o comando:

```
$ sudo nano /etc/sysctl.conf
```

Com o arquivo apto a edição, deve-se então localizar a linha que contém o parâmetro `# net.ipv4.ip_forward = 0`, onde será removido o símbolo de comentário (`#`) e ajustado o seu valor para `1`.

5.4.2.3. Configuração do acesso remoto utilizando um firewall

Para que os nodos da rede interna possam acessar a internet para instalar e/ou atualizar os seus pacotes de software, deve-se habilitar os recursos do Netfilter (firewall que roda a nível do kernel e que pode manipular as pilhas de redes linux) no nodo master, sendo necessário configurar o Iptables. O Iptables é um front-end para utilizar os recursos do Netfilter. O Iptables é utilizado para controlar quais pacotes podem entrar ou sair pela rede interna ou externa, assim ele poderá bloquear o tráfego indesejado de pacotes.

A função primária do Iptables é realizar a análise do tráfego de rede recebido pelo computador, no qual é feita uma comparação com o conjunto de regras preestabelecidas nas suas configurações. Ou seja, ele inspeciona todos os pacotes, verifica o enquadramento de cada uma das regras e aplica uma determinada ação. O processamento dos pacotes é feito pelo Iptables a partir de uma estrutura que contém suas camadas (tabelas) e cadeias (chains).

Para isso, será utilizado duas tabelas do Iptables: Filter e Nat. Primeiramente, é necessário limpar todas as regras vigentes no firewall para nos certificarmos de que não haverá interferência das regras atuais. Segue abaixo os comandos para tal configuração.

```
$ iptables -X  
$ iptables -F  
$ iptables -t nat -X  
$ iptables -t nat -F
```

Agora é necessário configurar as regras nas cadeias INPUT e FORWARD da tabela Filter para que os pacotes que atravessam o firewall sejam aceitos e processados corretamente. Segue abaixo os comandos para tal configuração.

```
$ iptables -A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT  
$ iptables -A FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT
```

CAPÍTULO V

E então será criada a regra na tabela Nat, cadeia POSTROUTING para habilitar o mascaramento de IP (IP masquerading) na interface WAN (conectada à Internet). Em nosso caso, a interface WAN é a enx00e04c3613e2. Segue abaixo os comandos para tal configuração.

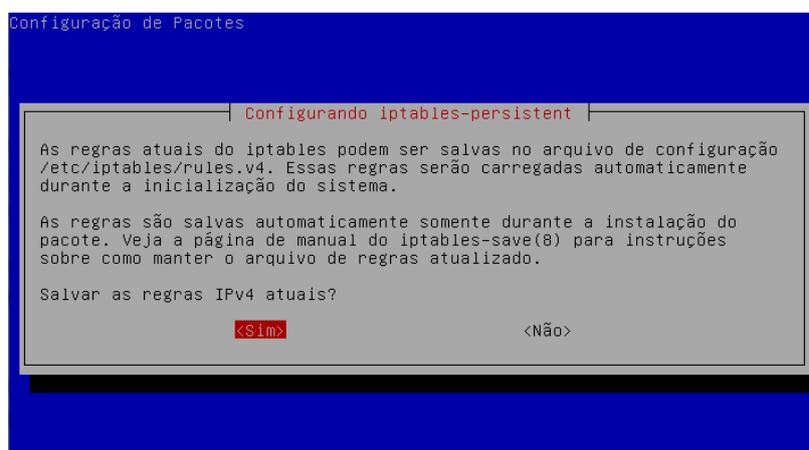
```
$ iptables -t nat -A POSTROUTING -o enx00e04c3613e2 -j MASQUERADE
```

Após o término desta etapa é necessário criar um mecanismo de persistência das configurações do Iptables, pois este conjunto de regras é reinicializado juntamente com o computador quando necessário reinicia-lo para algum propósito de manutenção ou atualização. Para tal, será instalado o pacote iptables-persistent. Segue abaixo os comandos para esta instalação.

```
$ sudo apt-get install iptables-persistent
```

Será apresentado a tela conforme descrito na Figura 5.8, sendo necessário confirmar a escolha da opção “Sim” e pressionar enter para salvar automaticamente as regras atuais do firewall iptables. Se o processo de instalação transcorreu como esperado, será apresentada uma confirmação indicando que o sistema está apto a criar a persistência das configurações.

Figura 5.8 – Tela de confirmação de instalação do iptables-persistent.



Fonte: Autoria própria (2017)

Agora, ao reiniciar o servidor, as regras do firewall existentes atualmente serão preservadas (na verdade, recarregadas). As regras são salvas no arquivo rules.v4 e lidas durante a reinicialização do sistema.

CAPÍTULO V

Caso seja necessário criar novas regras ou as regras atuais sejam alteradas, deve-se atualizar o arquivo de regras do iptables-persistent, para garantir que as novas regras também sejam carregadas durante a inicialização do sistema. Segue abaixo os comandos para salvar as configurações modificadas.

```
$ iptables-save > /etc/iptables/rules.v4
```

5.4.2.4. Instalação da biblioteca OpenMPI em todos os nós

Para que haja a troca de mensagens entre as máquinas, é necessário a instalação da biblioteca do MPI, devido a uma boa performance e boa documentação foi escolhido a distribuição OpenMPI. Segue abaixo os comandos para sua instalação.

```
$ sudo apt-get install openmpi-bin openmpi-common libopenmpi2 libopenmpi-dev
```

Uma das coisas a serem observadas na instalação das máquinas é que a versão do OpenMPI deve ser a mesma em todas as máquinas, para que não haja conflito de versão.

5.4.2.5. Instalação do compilador gfortran em todos os nós

Para rodar os códigos é necessário um compilador que suporte o Fortran 90, para tal foi instalado o GNU fortran. Segue abaixo os comandos para tal instalação.

```
$ sudo apt-get install gfortran
```

5.4.2.6. Instalação e configuração do openssh em todos os nós

a) Nós escravos

Para que o nó mestre possa controlar os nós escravos, cada nó escravo precisa de um servidor SSH instalado, para tal foi instalado o pacote openssh-server. Segue abaixo os comandos para sua instalação.

```
$ sudo apt-get install openssh-server
```

b) Nó mestre

Para poder controlar os nós escravos a partir do nó mestre é necessário um cliente SSH instalado, para tal foi instalado o pacote openssh-client. Segue abaixo os comandos para sua instalação.

CAPÍTULO V

```
$ sudo apt-get install openssh-client
```

Após a instalação do cliente, é necessário criar pares de chaves públicas/privadas para que o SSH possa habilitar a comunicação entre as máquinas. Digitando os comandos abaixo, foi seguido o fluxo de natural de instalação, aceitando a sugestão de localização das pastas de instalação das chaves pública e privada, deixando sem preencher o campo de senha.

```
$ ssh-keygen -t rsa
```

5.4.2.7. Tornando a comunicação transparente entre as máquinas

Seguindo então com a configuração do SSH, foi modificado o arquivo hosts contido na pasta etc, este arquivo encontra-se no nó mestre. Para tal foi aberto o arquivo, listando os valores de ip's e o nome de cada máquina no cluster.

```
$ sudo nano /etc/hosts
```

(Arquivo hosts editado)

```
127.0.0.1    localhost
127.0.1.1    pitagoras
192.168.86.1 mestre
192.168.86.11 escravo_01
192.168.86.12 escravo_02
192.168.86.13 escravo_03
192.168.86.14 escravo_04
192.168.86.15 escravo_05
192.168.86.16 escravo_06
```

Para que o nó mestre possa acessar outra máquina ele deve estar autorizado a logar-se sem senha. Para isso é necessário enviar do nó mestre para cada nodo escravo a chave pública do nó mestre, como exemplo:

```
$ ssh-copy-id -i ~/.ssh/id_rsa.pub pitagoras@escravo_01
```

Para saber se este processo funcionou, basta tentar acessar alguma máquina a partir do nó mestre utilizando o SSH:

```
$ ssh escravo_01
```

CAPÍTULO V

5.4.2.8. Configuração da biblioteca OpenMPI

Para permitir que o MPI saiba quais máquinas irão participar da execução dos programas, deve-se criar um arquivo para armazenar esta informação. Deve-se então criar o arquivo `/home/pitagoras/.mpi_hostfile`, no nó mestre. Segue abaixo os comandos para tal.

```
$ sudo nano /home/pitagoras/.mpi_hostfile
```

contendo os seguintes comandos:

```
# Arquivo contendo os nodos que serão utilizados no cluster
# Nodo mestre
localhost
# Se os nodos escravos possuírem 4 processadores
escravo_01 slots=4 max-slots=4
escravo_02 slots=4 max-slots=4
escravo_03 slots=4 max-slots=4
escravo_04 slots=4 max-slots=4
escravo_05 slots=4 max-slots=4
escravo_06 slots=4 max-slots=4
```

5.4.2.9. Teste de execução de um Programa no Cluster

Para testar a comunicação entre as máquinas foi utilizado um programa simples (`hello_world.f90`), que apenas escreve na tela uma mensagem de “alô mundo” e indica de qual nodo o comando de escrita na tela se originou. Este programa se encontra no anexo C.

Para compilar o programa “`hello_world.f90`”

```
$ mpifort teste.f90 -o hello_world
```

Para executar “teste” em dois processos na máquina local:

```
$ mpirun -np 2 ./hello_world
```

Para executar “`hello_world`” em cinco processos no cluster usando o arquivo de endereçamento “`mpi_hostfile`” criado anteriormente (deve-se observar que o executável “`hello_world`” deve estar na mesma pasta local em cada máquina):

```
$ mpirun -np 2 --hostfile /home/pitagoras/.mpi_hostfile ./hello_world
```

CAPÍTULO V

5.5. IMPLEMENTAÇÃO PARALELA

Como já discutido no capítulo anterior, a função aptidão é modelada como um funcional de medida de verossimilhança entre as medidas experimentais de capacitância simulada e os valores de capacitância óbitos pelo modelo numérico, implicando na solução do problema direto para cada avaliação da função aptidão, tornando este cálculo demasiadamente custoso. Devido as características inerentes do algoritmo OEP, o uso da computação paralela apresenta-se como uma alternativa viável para redução do tempo computacional necessário para solução do problema inverso da técnica de TCE.

Um dos primeiros trabalhos de paralelização do algoritmo OEP foi desenvolvido por Schutte *et al.* (2004), o qual propôs uma implementação síncrona utilizando a estratégia Mestre/Escravo (ME), denominada por eles de PPSO (do inglês Parallel Synchronous Particle Swarm Optimization).

Neste tipo de estratégia existe uma unidade de processamento chamada de Processador Mestre (PM), que distribui, controla e reagrupa a ação de cálculo realizada por outros processadores, estes processadores são denominados Processadores Escravos (PEs). Sendo assim, a comunicação entre os PEs e o PM acontece em modo síncrono, existido a necessidade de pontos de espera para que o PM possa analisar e logo após partilhar as informações sobre a análise da função aptidão a cada ciclo de iteração do algoritmo OEP.

A implementação do algoritmo OEP paralelo utilizando a estratégia ME é bastante simples. Inicialmente o processador mestre (PM) cria um conjunto de posições e velocidades iniciais igualmente distribuídas no espaço de busca (a qual é feita, normalmente, de maneira aleatória), onde logo após, divide este conjunto em subconjuntos igualmente balanceados e então envia-os para que suas funções de aptidão sejam analisadas pelos PEs.

Os PEs resolvem o problema direto e compararam seus resultados com os valores de capacitância obtidos experimentalmente em um funcional de aptidão. Assim, para cada configuração de posição a função de aptidão é avaliada e o valor resultante é devolvido ao PM.

Ao final de cada iteração é realizado um teste sobre o valor atual do ótimo global encontrado pelo enxame. Sempre que esse valor se repetir dentro de uma determinada tolerância o valor do contador (k) é incrementado. Se o valor do ótimo se repetir por N_{min} vezes (denominado de número de permanência), então o critério de convergência do algoritmo é testado.

CAPÍTULO V

As ações executadas pelo PM e pelos PEs são descritas no Pseudocódigo 5.1. Para a comunicação entre os processadores mestre e escravos são utilizadas funções de comunicação ponto a ponto do MPI, particularmente as funções *MPI_Send()* e *MPI_Recv()*, detalhadas no Anexo B.

Pseudocódigo 5.1 - Algoritmo OEP Paralelo.

```
1 se meu_id = 0 então // Nodo Mestre
2   inicialização das constantes  $iter_{max}$ ,  $n$ ,  $c_1$ ,  $c_2$ ,  $w$ ;
3   inicialização aleatória de  $n$  posições e velocidades no espaço  $R^n$ ;
4   saída  $\leftarrow$  "falso"
5 fim se
6 repita
7   se meu_id = 0 então // Nodo Mestre
8     para alvo de 1 até  $n_{processador}$  faça
9       envio das posições  $x_i$  das  $m$  partícula para cada nodo escravo
10    fim para
11    para fonte de 1 até  $n_{processador}$  faça
12      recebimento das posições  $x_i$  das  $m$  partículas oriundas de cada nodo escravo;
13      recebimento da avaliação  $f(x_i)$  de  $m$  partículas oriundas de cada nodo escravo;
14    fim para
15    realização de uma barreira de sincronização para todas partículas;
16    para  $i$  de 1 até  $n$  faça
17      se  $f_k^i \leq f_p^i$  então
18         $f_p^i = f_k^i$ 
19         $p_{best} = x_k^i$ 
20      senão
21         $f_p^k = f_i^{k-1}$ 
22         $p_i^k = p_i^{k-1}$ 
23      fim se
24      se  $f_k^i \leq f_g$  então
25         $f_g = f_k^i$ 
26         $g_{best} = x_k^i$ 
27      senão
28         $f_g^k = f_g^{k-1}$ 
29         $g_{best}^k = g_{best}^{k-1}$ 
30      fim se
31    fim para
32    atualização da velocidade de acordo com a Equação (4.4);
33    atualização da posição de acordo com a Equação (4.5);
34    atualização do peso de inércia de acordo com a Equação (4.6);
35    se condição de parada atendida então
36      saída = "verdade"
37       $k=k+1$ 
38    fim se
39  senão // Nodo Escravo
40    recebimento das posições  $x_i$  das  $m$  partículas oriundas do nodo mestre;
41    para  $i$  de 1 até  $m$  faça
```

CAPÍTULO V

```
42     resolução do problema direto para cada partícula  $x_i$ ;  
43     avaliação  $f(x_i)$ ;  
44     fim para  
45     envio das posições  $x_i$  das  $m$  partícula de volta para o nodo mestre;  
46     envio da avaliação  $f(x_i)$  das  $m$  partícula de volta para o nodo mestre;  
47     fim se  
48 até que saída
```

Resultado: Assume-se g_{best} como a solução do problema;

Fonte: Adaptado de Schutte et. al. (2004)

Pode-se perceber pelo exposto que um ponto fraco dessa estratégia é a necessidade de sincronização, tornando o código limitado pelo escravo mais lento empregado no ciclo de iteração.

Além do inconveniente citado anteriormente, de acordo com KOH *et al.* (2006), a estratégia síncrona de paralelização necessita que alguns critérios sejam atendidos para que a sua utilização seja eficiente no uso dos recursos computacionais disponíveis:

- a) *Uso de uma rede de conexão de alta performance:* A estratégia Mestre/Escravo apresenta a condição de que a comunicação de dados entre os nodos é realizada por um protocolo de rede cabeada, formando um gargalo de desempenho devido à alta latência desse tipo de arquitetura de interconexão entre os processadores. Para este propósito utilizou-se uma rede gigabit ethernet para a montagem do cluster.
- b) *Tempo de cálculo da função objetivo é constante e independente das coordenadas no espaço de busca:* Como o tempo de cálculo para a função aptidão é praticamente o mesmo cálculo da solução do problema direto, e como o esforço computacional é invariável a distribuição essa situação naturalmente se observa.
- c) *Número de tarefas paralelas seja igualmente dividida entre os processadores:* Implementou-se uma estrutura apropriada no algoritmo favorecendo a divisão de tarefas em grandes blocos de dados, balanceando a carga atribuída a cada PEs, diminuindo a quantidade de solicitações inter-nodais entre os processadores.

CAPÍTULO VI

RESULTADOS

6.1 - INTRODUÇÃO

Neste capítulo serão apresentados os resultados obtidos a partir do processo investigativo de determinação de parâmetros ótimos a serem adotados como padrão para a adoção do algoritmo de OEP como solucionador do problema inverso da TCE. Para tal será apresentado os experimentos numéricos empregados neste processo.

Na execução dos testes numéricos foram utilizados apenas dados sintéticos de distribuição de material no interior do sistema sensor, mas que ainda assim, mantem compromisso em termos de sua magnitude e distribuição geométrica de permissividade com os comportamentos encontrados em escoamentos reais nas secções de tubulações empregadas na indústria de petróleo.

Para analisar a resolução espacial obtida pelos algoritmos de reconstrução aplicados a TCE, foi implementado numericamente a resposta de um sensor capacitivo com oito eletrodos de placas paralelas, onde os valores de capacitâncias foram obtidos a partir da solução do problema direto para valores simulados de permissividade.

Foram admitidos três materiais distintos para a composição do sensor e do meio imageado, o acrílico da tubulação, o óleo e a água contida no interior da tubulação simulando 3 diferentes distribuições de uma mistura bifásica (óleo-água): uma distribuição *core flow*, uma distribuição homogênea de água contendo bolhas de óleo e por último uma distribuição estratificada.

Para as execuções das simulações numéricas, foram utilizados os valores de permissividade relativa e as dimensões da tubulação como descritos na Tabela 6.1 e na Tabela 6.2.

CAPÍTULO VI

Todos os algoritmos implementados no processamento de obtenção de imagens, tanto os seriais quanto os paralelos, foram codificados em linguagem Fortran 90, utilizando-se o compilador gfortran da GNU. Para o pré e pós-processamento foram implementadas rotinas desenvolvidas na linguagem MATLAB utilizando o pacote EIDORS (Electrical Impedance and Diffuse Optical Tomography Reconstruction Software) (ADLER et al. - 2006) no intuito de fornecer uma percepção gráfica para os resultados obtidos.

Tabela 6.1 – Valores da permissividade relativa dos materiais empregados no sensor.

Ácrilico	Água	Óleo
4,1	80	3

Fonte: Autoria própria (2017)

Tabela 6.2 – Dimensões físicas do tubo simulado.

Raio Interno	Raio Externo	Parede do Tubo	Comprimento do Sensor
92,7 mm	100,0 mm	7,3 mm	182 mm

Fonte: Autoria própria (2017)

Durante o processo de reconstrução da imagem, utilizou-se uma malha estruturada, subdividindo a região em estudo em duas sub-regiões: a parte interna e a parede da tubulação, como pode ser observado na Figura 6.1.

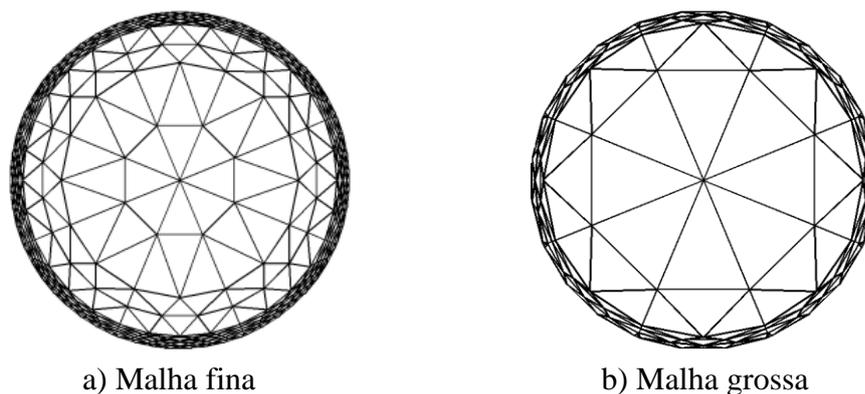
Em relação a escolha da malha, Azevedo Filho (2002) observa que uma maior discretização (refinamento) da malha sugere um melhor resultado do processo de obtenção da solução do problema direto implicando em um melhor resultado no processo de reconstrução das imagens, mas também aponta para o desafio de trabalhar com um número maior de incógnitas a serem determinadas pelo algoritmo inverso, ficando-se então no compromisso de se determinar uma quantidade ótima que possa ser empregada oferecendo a melhor resolução possível mas sem perder a capacidade de convergência do método.

Um outro ponto que também deve-se ser considerado no processo de reconstrução é o chamado “crime de inversão” como descrito em Wirgin (2004), que consiste na utilização dos mesmos (ou quase os mesmos) ingredientes teóricos para sintetizar, bem como para inverter dados em um problema inverso, sendo assim, utilizou-se uma malha mais refinada no interior da região interna do sensor para calcular os valores de capacitância simulados

CAPÍTULO VI

(com 128 elementos) e uma malha menos refinada (com 32 elementos) empregada na solução do problema inverso, como descrito na Figura 6.1 – a e 6.1 – b. A justificativa de se utilizar uma malha com poucos elementos advém da condição imposta pela dificuldade de resolução do problema inverso com o aumento do espaço de busca.

Figura 6.1 - Discretização do domínio (a) com 128 elementos e (b) com 32 elementos na parte interior da tubulação.



Fonte: Autoria própria (2017)

6.2 - ESTUDO DOS PARÂMETROS RELEVANTES PARA O ALGORITMO OEP

Para a realização desta análise foram realizadas 100 execuções para cada conjunto de partículas, onde foram escolhidos os valores de $\omega_{max} = 0,9$ e $\omega_{min} = 0,4$ para o peso de inércia máximo e mínimo respectivamente, conjuntamente com o processo de atualização para o peso de inércia descrito pela Equação (4.7). Os coeficientes comportamentais de cognição e socialização foram atribuídos com os valores $c_1 = 2.1$ e $c_2 = 2.1$, o número máximo de iterações foi dimensionado como sendo 20 vezes o valor da dimensão do problema, neste caso 640 iterações, e foi atribuído um valor de 0,01 como limite de erro entre o valor de capacitância sintético e o valor calculado pelo procedimento inverso.

No intuito de melhorar a performance do algoritmo OEP, foi proposto um mecanismo de atualização dinâmica do tamanho do enxame durante o processo de execução do algoritmo OEP, este procedimento aumenta de maneira gradual e em múltiplos inteiros do valor inicial do enxame o número de partículas em execução até o máximo de 2000 iterações, isto só ocorre quando o algoritmo encontra uma certa dificuldade para atualizar o valor de fitness global do enxame, indicando uma estagnação prematura do algoritmo em algum mínimo local, e diminui este valor gradualmente na mesma proporção quando o algoritmo consegue escapar deste ponto e consegue localizar um outro ponto mais promissor que o anterior no espaço de soluções, atualizando assim o valor do fitness global.

CAPÍTULO VI

O procedimento delineado anteriormente fornece ao algoritmo OEP, quando necessário, a possibilidade de ampliar o número de partículas no domínio, criando assim uma maior capacidade de exploração global do espaço de otimização, mas também possibilita quando não mais necessário, diminuir o custo computacional desnecessário no processo de procura do mínimo global do problema. Segue abaixo os resultados obtidos pelo algoritmo OEP quando é variado o número inicial de partículas.

Tabela 6.3 – Solução do algoritmo OEP utilizando um mecanismo auto adaptativo para o número de partículas no enxame.

Tamanho inicial do enxame		10	20	50	100	200
Quantidade de avaliações	μ	13834,65	8053,33	12085,35	18347,47	24753,54
	σ_s	47981,03	19639,77	14859,28	18162,17	16469,34
	Max	412730	134060	120550	142300	81800
	Min	480	540	750	1100	10200
Quantidade de iterações	μ	119,40	106,70	95,13	91,90	80,10
	σ_s	63,45412	49,30	36,97	35,46	22,2612
	Max	441	246	134	141	150
	Min	42	26	27	18	39
Valor de fitness encontrado	μ	0,009	0,009	0,009	0,009	0,009
	σ_s	0,001	0,001	0,001	0,001	0,001
	Max	0,010	0,010	0,010	0,010	0,010
	Min	0,006	0,005	0,007	0,005	0,006
Tempo de solução (s)	μ	14,040	8,086	12,106	18,368	24,795
	σ_s	49,184	19,690	14,912	18,127	16,340
	Max	425,465	134,379	129,217	113,570	81,370
	Min	0,499	0,546	1,235	1,548	10,327
Índice de sucesso (%)		100	100	100	100	100

Fonte: Autoria própria (2017)

Pode-se observar pelos resultados apresentados na Tabela 6.3 que o algoritmo OEP apresentou um desempenho semelhante em relação ao valor de fitness para todas as situações propostas, mas já em termos do tempo médio de execução apresentou um melhor desempenho com a quantidade inicial de 20 partículas, onde percebe-se que para um

CAPÍTULO VI

tamanho inicial muito maior ou muito menor do que este o algoritmo desperdiça tempo executando cálculos desnecessários ou fica muito tempo preso em mínimos locais.

Considerando agora o tipo de fronteira do espaço de solução, utilizou-se os mesmos parâmetros descritos anteriormente, mas agora trabalhando com o valor de 20 partículas para inicialização do enxame, com esta análise, buscou-se determinar a melhor configuração de atualização das partículas quando elas deixam o espaço de busca viável que se encontra no intervalo de 3,0 e 80.

Tabela 6.4 – Solução do algoritmo OEP utilizando diferentes tipos de condições de fronteiras.

Tipo de condição de fronteira		CFR	CFAM	CFA	CFI	CFIR	CFIA
Quantidade de avaliações	μ	2140,5	8966,8	1201,84	31118,45	5024,92	3375,29
	σ_s	70029,89	33342,67	268,7006	93460,23	1123,48	754,65
	Max	76040	246940	3700	692178,24	15470,32	9780,29
	Min	420	860	420	2410,61	1756,09	1110,21
Quantidade de iterações	μ	62,15	112,86	55,92	316,35	233,81	143,19
	σ_s	34,36	52,31	11,31371	146,63	47,30	28,97
	Max	232	323	123	640	514,28	325,13
	Min	23	41	20	120,53	83,62	52,87
Valor de fitness encontrado	μ	0,007	0,009	0,007	0,025	0,030	0,019
	σ_s	0,002	0,001	0,000	0,002	0,001	0,001
	Max	0,004	0,005	0,001	0,015	0,024	0,026
	Min	0,000	0,001	0,003	0,008	0,009	0,007
Tempo de solução (s)	μ	2,176	9,065	1,210	25,409	5,052	3,795
	σ_s	10,171	33,723	0,298	94,526	1,243	0,787
	Max	76,675	249,477	3,744	532,792	15,654	9,897
	Min	0,218	0,5148	0,4212	1,443	1,761	1,322
Índice de sucesso (%)		100	100	100	92	100	100

Fonte: Autoria própria (2017)

Pode-se observar pela Tabela 6.4 que a melhor escolha de condição de fronteira foi definida pelo algoritmo CFA, este algoritmo se mostrou capaz de explorar de maneira eficiente o espaço de busca equilibrando a capacidade de exploração global e local, isto pode

CAPÍTULO VI

ser percebido devido aos baixos valores das quantidades de avaliações em contraste com o alto índice de sucesso encontrado.

Analisando o tipo de condição de parada, utilizou-se os mesmos parâmetros descritos anteriormente, escolhendo convenientemente a quantidade de 20 partículas para inicialização do enxame, conjuntamente com a condição de fronteira CFA.

Como descrito na seção 4.5, o processo de análise das condições de parada possui um alto custo computacional, por isso fica inviável a sua análise a cada iteração, sendo necessário um intervalo predeterminado entre as avaliações, este intervalo será chamado de índice de amostragem e o parâmetro de comparação que indicará que o algoritmo não mais apresenta-se promissor em determinar uma solução melhor será chamado de limiar de estagnação. Segue abaixo o ajuste adequado dos parâmetros descritos na seção 4.5.

Tabela 6.5 – Solução do algoritmo OEP em relação as condições de parada

Índice de estagnação		10		25		50	
Limiar de estagnação		0,10	0,15	0,10	0,15	0,10	0,15
Quantidade de avaliações	μ	547,2	494	1062,6	1135	1251,2	1343
	σ_s	255,64	154,20	481,32	514,04	755,75	748,24
	Max	1520	1120	4200	3860	7540	4560
	Min	400	400	520	540	560	600
Quantidade de iterações	μ	24,2	23	48,01	51,1	57,56	59,47
	σ_s	6,99	5,41	11,12	11,54	19,11	21,81
	Max	40	40	100	100	105	124
	Min	20	20	25	26	29	29
Valor de fitness encontrado	μ	16,482	18,060	0,525	0,367	0,008	0,007
	σ_s	6,273	5,598	1,922	1,181	0,008	0,003
	Max	31,548	32,356	13,82	9,720	0,077	0,012
	Min	3,8703	5,855	0,000	0,000	0,000	0,000
Tempo de solução (s)	μ	0,545	0,493	1,063	1,141	1,255	1,342
	σ_s	0,243	0,154	0,479	0,510	0,788	0,744
	Max	1,451	1,108	2,5428	3,697	7,909	4,586
	Min	0,390	0,374	0,515	0,515	0,662	0,577
Índice de sucesso (%)		0	0	77	60	99	100

Fonte: Autoria própria (2017)

CAPÍTULO VI

Pode-se observar pela Tabela 6.5 que a melhor escolha do valor do limiar de estagnação e do índice de estagnação foi dada respectivamente pelo par 0,15 e 50, os quais apresentaram para o algoritmo um equilíbrio entre a qualidade da obtenção da imagem sem comprometer o seu desempenho temporal.

6.3 - ALGORITMOS CONVENCIONAIS VERSUS ALGORITMO OEP

Foram empregados dois métodos de avaliação da qualidade das imagens obtidas: um de natureza qualitativa e outra de natureza quantitativa. A avaliação qualitativa envolve comparações visuais observando o padrão geral de similaridade dos resultados obtidos pelos diferentes métodos de reconstrução com as distribuições de referência, que estão descritas na Figura 6.2, com os resultados obtidos a partir da resposta de cada algoritmo, como descrito nas Figuras 6.3, 6.4 e 6.5. Já a análise quantitativa pretende analisar as imagens obtidas a partir do uso de estimadores de discrepância pixel a pixel listados abaixo, onde cada um possui capacidades distintas de caracterização das imagens obtidas.

- Erro quadrático normalizado;

$$\Delta\varepsilon = \frac{\sum_{i=1}^N (\varepsilon_i^{sim} - \varepsilon_i^{cal})^2}{\sum_{i=1}^N (\varepsilon_i^{sim} - \bar{\varepsilon}^{cal})^2} \quad (6.1)$$

- Erro absoluto normalizado;

$$\delta\varepsilon = \frac{\sum_{i=1}^N |\varepsilon_i^{sim} - \varepsilon_i^{cal}|}{\sum_{i=1}^N |\varepsilon_i^{sim} - \bar{\varepsilon}^{cal}|} \quad (6.2)$$

- Coeficiente de correlação.

$$R_{xy} = \frac{\sum_{i=1}^N (\varepsilon_i^{sim} - \bar{\varepsilon}^{sim})(\varepsilon_i^{cal} - \bar{\varepsilon}^{cal})}{\sqrt{(\sum_{i=1}^N (\varepsilon_i^{sim} - \bar{\varepsilon}^{sim})^2)(\sum_{i=1}^N (\varepsilon_i^{cal} - \bar{\varepsilon}^{cal})^2)}} \quad (6.3)$$

onde ε_i^{sim} e ε_i^{cal} são respectivamente o valor de permissividade do elemento i da distribuição de dielétrico simulada e da distribuição de dielétrico calculada pelo algoritmo TRL, TRA, TAIL e OEP, $\bar{\varepsilon}^{sim}$ e $\bar{\varepsilon}^{cal}$ são seus respectivos valores médios.

O melhor algoritmo é aquele que possui pequenos valores para $\delta\varepsilon$ e $\Delta\varepsilon$ e valores de R_{xy} próximos da unidade.

CAPÍTULO VI

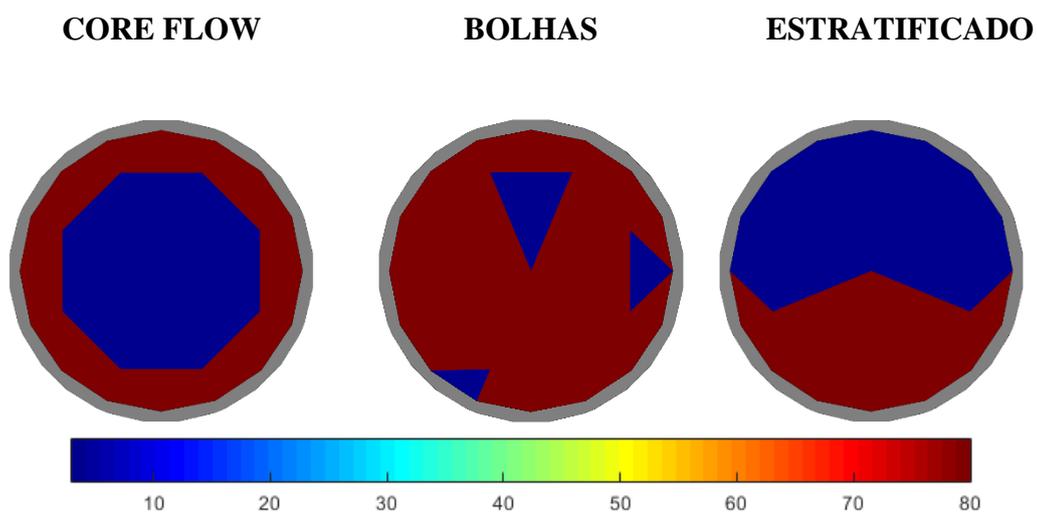
O erro quadrático normalizado $\Delta\varepsilon$, é sensível a grandes erros de alguns elementos, enquanto o erro absoluto normalizado $\delta\varepsilon$, é sensível a pequenos erros de muitos elementos, já o coeficiente de correlação indica a similaridade espacial entre a imagem de referência e a imagem reconstruída.

Além dos parâmetros apresentados na seção 6.2, foram utilizadas 20 partículas para inicialização dinâmica do enxame e 200 partículas para o tamanho máximo permitido e um máximo de 200 iterações, além disso utilizou-se a condição de fronteira CFA e um limiar de estagnação igual a 1,5 e um índice de estagnação igual a 50, estes parâmetros foram empregados na solução do algoritmo OEP, tanto na sua forma serial quanto na paralela, com o objetivo de comparação com a solução encontrada pelos demais algoritmos empregados na solução do problema inverso.

Para determinação dos valores médios de erro de permissividade e tempo médio de processamento do algoritmo OEP, foram realizadas 100 execuções para cada configuração de distribuição de permissividade.

Avaliação quantitativa dos resultados estão apresentados nas Tabelas 6.6, 6.7, 6.8, 6.9, 6.10 e 6.11 e a avaliação qualitativa das imagens estão listados respectivamente nas Figuras 6.3, 6.4 e 6.5.

Figura 6.2 – Distribuições de dielétricos simuladas.



Fonte: Autoria própria (2017)

CAPÍTULO VI

Tabela 6.6 – Valores de erro médio de permissividade relativa referente ao processo de reconstrução de imagens sem ruído numérico.

Geometria	Método	Erro Quadrático Relativo (%)	Erro Absoluto Relativo (%)	Coefficiente de Correlação
CORE-FLOW	TRL	0,000	0,000	1,000
	TRA	0,000	0,000	1,000
	TAIL	4,792	6,412	0,977
	OEP	1,50e-04	4,67e-02	1,000
BOLHAS	TRL	33,753	48,294	0,821
	TRA	2,154	13,153	0,994
	TAIL	35,148	44,911	0,822
	OEP	3,41e-04	1,26e-02	1,000
EXTRATIFICADO	TRL	11,956	11,539	0,938
	TRA	23,562	12,972	0,876
	TAIL	9,191	8,581	0,953
	OEP	2,780e-06	4,203e-03	1,000

Fonte: Autoria própria (2017)

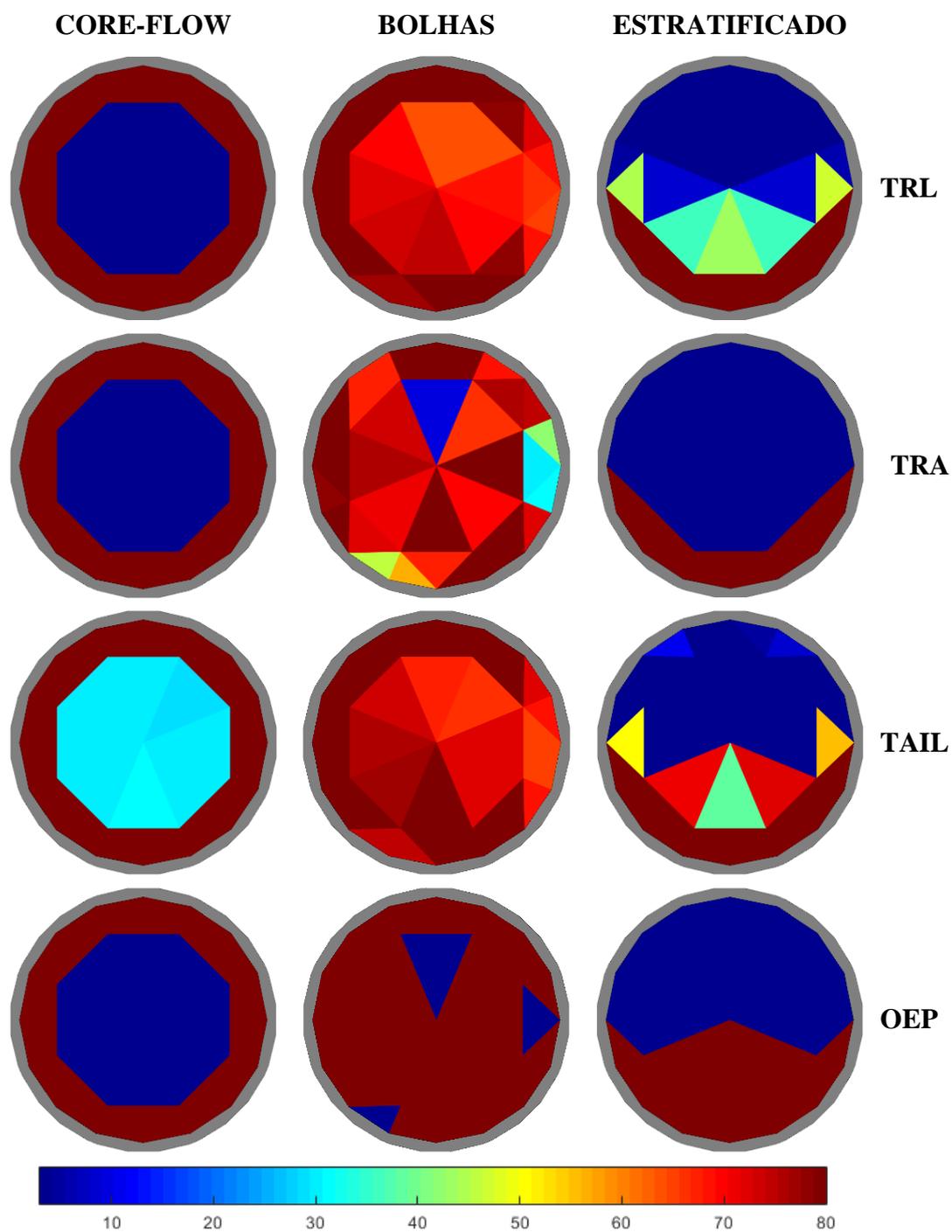
Tabela 6.7 – Valores do tempo médio de processamento referente ao processo de reconstrução de imagens sem ruído numérico.

	CORE-FLOW	BOLHAS	EXTRATIFICADO
TRL	0,000	0,000	0,000
TRA	0,640	0,639	2,106
TAIL	0,000	0,000	0,000
OEP	1,310	1,391	1,036

Fonte: Autoria própria (2017)

CAPÍTULO VI

Figura 6.3 – Imagens reconstruídas a partir dos valores de capacitância obtidas das distribuições de dielétricos simuladas sem adição de ruído numérico.



Fonte: Autoria própria (2017)

CAPÍTULO VI

Tabela 6.8 – Valores de erro médio de permissividade relativa referente ao processo de reconstrução de imagens para um ruído numérico de 1%.

Geometria	Método	Erro Quadrático Relativo (%)	Erro Absoluto Relativo (%)	Coefficiente de Correlação
CORE-FLOW	TRL	0,000	0,000	1,000
	TRA	24,453	16,590	0,872
	TAIL	2,472	4,635	0,988
	OEP	0,034	0,634	1,000
BOLHAS	TRL	10,034	8,342	0,949
	TRA	42,321	26,928	0,768
	TAIL	6,573	5,556	0,967
	OEP	0,041	0,482	1,000
EXTRATIFICADO	TRL	11,902	11,484	0,939
	TRA	47,283	23,348	0,764
	TAIL	8,652	8,334	0,956
	OEP	0,040	0,551	1,000

Fonte: Autoria própria (2017)

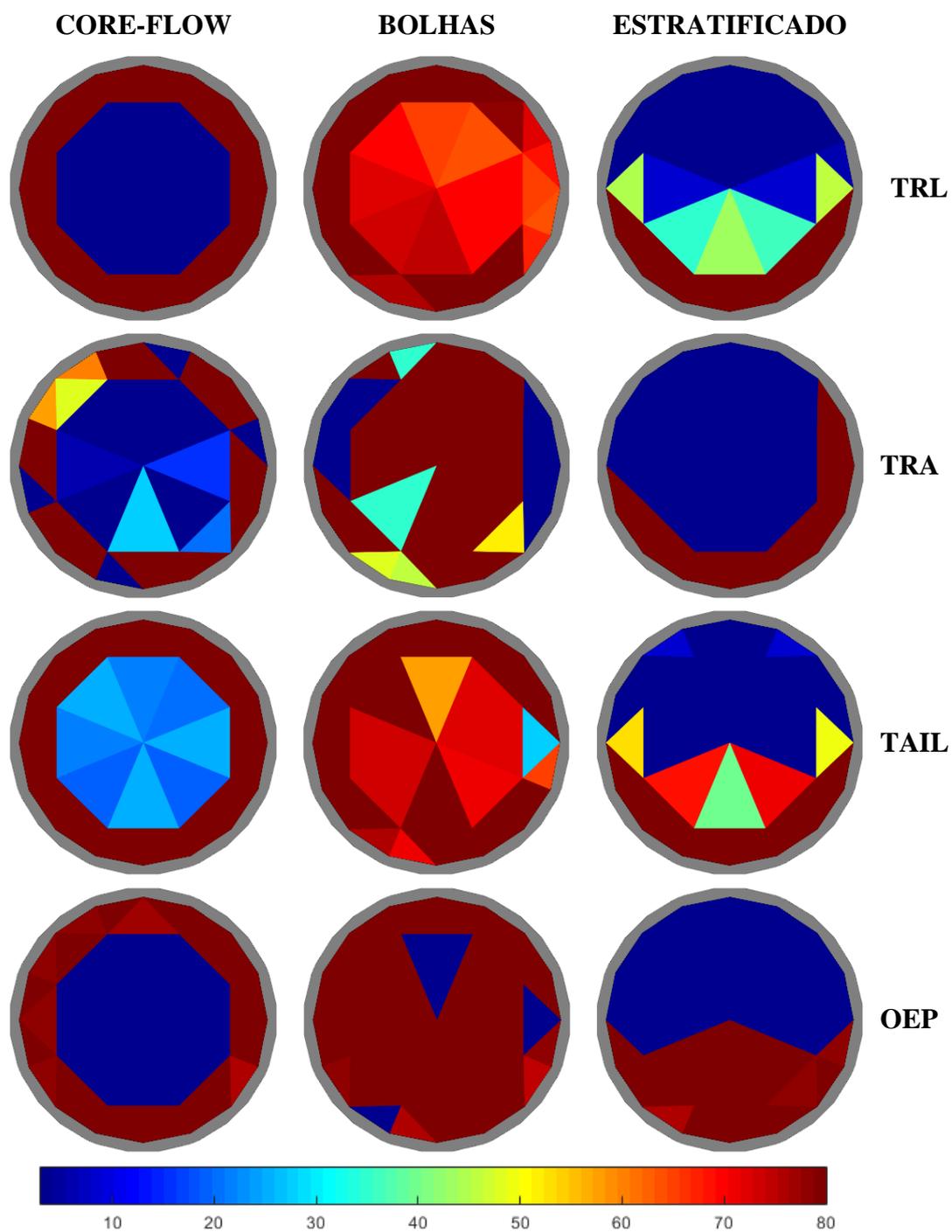
Tabela 6.9 – Valores do tempo médio de processamento referente ao processo de reconstrução de imagens para um ruído numérico de 1%.

	CORE-FLOW	BOLHAS	EXTRATIFICADO
TRL	0,000	0,000	0,000
TRA	0,686	0,639	2,099
TAIL	0,000	0,000	0,000
OEP	20,537	22,235	20,361

Fonte: Autoria própria (2017)

CAPÍTULO VI

Figura 6.4 – Imagens reconstruídas a partir dos valores de capacitância obtidas das distribuições de dielétricos simuladas com adição de 1% de ruído numérico.



Fonte: Autoria própria (2017)

CAPÍTULO VI

Tabela 6.10 – Valores de erro médio de permissividade relativa referente ao processo de reconstrução de imagens para um ruído numérico de 3%.

Geometria	Método	Erro Quadrático Relativo (%)	Erro Absoluto Relativo (%)	Coefficiente de Correlação
CORE-FLOW	TRL	0,000	0,000	1,000
	TRA	0,003	0,001	1,000
	TAIL	2,692	4,724	0,987
	OEP	0,874	3,282	0,997
BOLHAS	TRL	10,010	8,405	0,949
	TRA	37,145	19,853	0,804
	TAIL	7,378	6,096	0,963
	OEP	9,067	6,608	0,954
EXTRATIFICADO	TRL	11,831	11,451	0,939
	TRA	23,569	2,974	0,880
	TAIL	7,914	7,893	0,960
	OEP	0,292	1,174	0,999

Fonte: Autoria própria (2017)

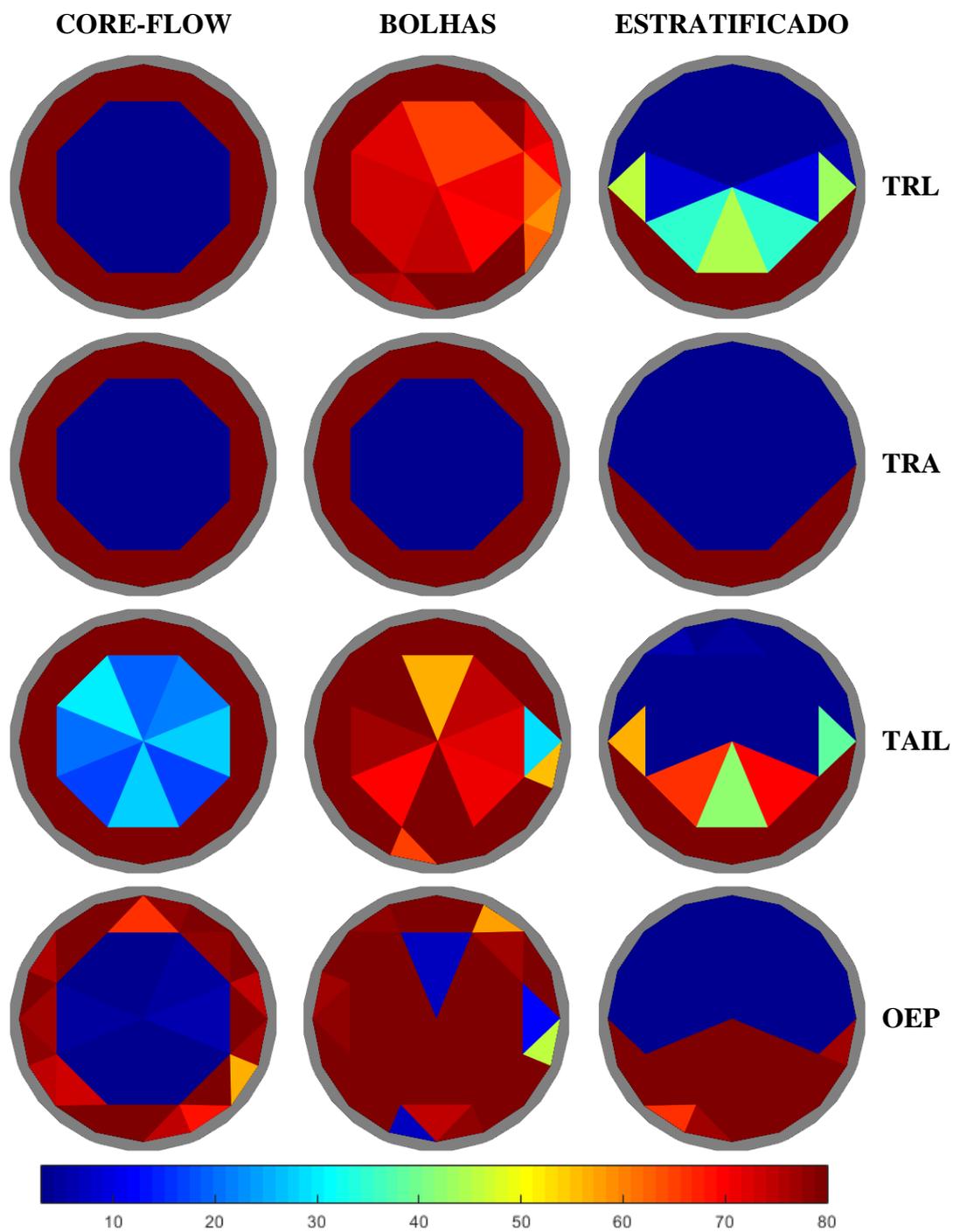
Tabela 6.11 – Valores do tempo médio de processamento referente ao processo de reconstrução de imagens para um ruído numérico de 3%.

	CORE-FLOW	BOLHAS	EXTRATIFICADO
TRL	0,000	0,000	0,000
TRA	0,842	1,498	1,966
TAIL	0,000	0,016	0,002
OEP	22,269	42,697	24,501

Fonte: Autoria própria (2017)

CAPÍTULO VI

Figura 6.5 – Imagens reconstruídas a partir dos valores de capacitância obtidas das distribuições de dielétricos simuladas com adição de 3% de ruído numérico.



Fonte: Autoria própria (2017)

CAPÍTULO VI

Pode-se perceber pelos resultados quantitativos apresentados nas Tabelas 6.6, 6.8 e 6.10 e pelos resultados qualitativos apresentados nas Figuras 6.3, 6.4 e 6.5 que o algoritmo OEP conseguiu recuperar todos os perfis de escoamentos propostos mesmo com a adoção de ruído numérico aos dados, já os demais algoritmos ou não conseguem determinar o perfil de escoamento de maneira precisa ou não possuem sensibilidade para distinguir diferentes tipos de padrões, significando que estes algoritmos possuem limitações impeditivas para aplicações gerais.

Para se determinar o tempo de processamento foi utilizado a função interna CPU_TIME() do FORTRAN, esta função possui a limitação de não medir tempos inferiores a milissegundos, sendo assim, devido à uma baixa complexidade algorítmica dos algoritmos TRL e o TAIL, estes apresentaram um tempo de processamento igual a zero, onde apenas pode-se inferir com este resultado que as rotinas foram executadas em uma escala de tempo inferior a milissegundos. Além disso pode-se observar que o tempo de processamento aumenta proporcionalmente ao nível de ruído numérico introduzido nos dados, um outro ponto que deve também ser considerado é a dependência do tempo de processamento com o padrão a ser determinado, onde o padrão de bolhas apresentou-se como uma configuração mais difícil a ser determinada pelo algoritmo OEP.

É possível perceber a partir dos resultados obtidos através dos indicadores de erro apresentados anteriormente, que a técnica OEP apresenta desempenho superior as demais técnicas em termos de resolução espacial, mas apresenta uma resolução temporal muito inferior, sendo proibitiva para aplicações em tempo real.

6.4 – PARALELIZAÇÃO DO ALGORITMO OEP

Já em relação a medida do desempenho obtido pelo processamento paralelo podemos utilizar algumas métricas tais como: largura de banda, latência de acesso a memória, número de operações de ponto flutuante e etc. Sendo que as mais utilizadas são:

- Speed-up (escalabilidade) – é uma medida do grau de desempenho de um conjunto de processadores quando comparado a um único processador (SILVA, 2013). O speed-up mede a relação entre o tempo de execução sequencial e o tempo de execução em paralelo. Pode ser calculado por meio da equação

$$S(p) = \frac{T_s}{T_p} \quad (6.4)$$

CAPÍTULO VI

onde T_s e T_p correspondem, respectivamente, ao tempo gasto por um processador para executar de maneira serial uma determinada tarefa e ao tempo gasto por vários processadores rodando em paralelo para executar esta mesma tarefa. Quanto mais próximo este valor estiver do número de processadores utilizados, melhor será o desempenho.

- Eficiência paralela – Corresponde a uma medida do grau de aproveitamento dos recursos computacionais por unidade de processador (SILVA, 2013). Esta grandeza estima o quão bem os processadores estão sendo utilizados para resolver o problema

$$E(p) = \frac{S(p)}{p} = \frac{T_s}{p * T_p} \quad (6.5)$$

Quanto mais próximo o valor de $E(p)$ de 1, mais eficiente está sendo o sistema paralelo.

- Overhead – Corresponde a um parâmetro que indica uma melhora de performance do cluster como se fosse uma só máquina (SILVA, 2013). Com o overhead também podemos entender o quanto de tempo está sendo perdido com o tráfego de rede.

$$V(p) = \frac{p * T_p - T_s}{T_1} = \frac{1 - E(p)}{E(p)} \quad (6.6)$$

Esse valor, como podemos perceber tende a zero para uma configuração paralela ideal.

Para determinação das métricas citadas anteriormente para o algoritmo OEP paralelo, foram realizadas 100 execuções do código para cada quantidade de processadores alocados com fim de se determinar um valor de tempo médio a ser utilizado para caracterização do sistema paralelo proposto. Foram empregados 28 processadores

Foi escolhida a configuração de bolhas como perfil a ser determinado, pois esta configuração se mostrou como perfil de maior demanda de tempo de processamento, como pode ser observado nas Tabelas 6.7, 6.9 e 6.11. Para execução do código foram utilizados os mesmos ajustes de parâmetros utilizados na seção 6.3, sendo que o erro numérico introduzido foi de 3%.

Avaliação quantitativa dos resultados encontram-se apresentados nas Figuras 6.6, 6.7 e 6.8.

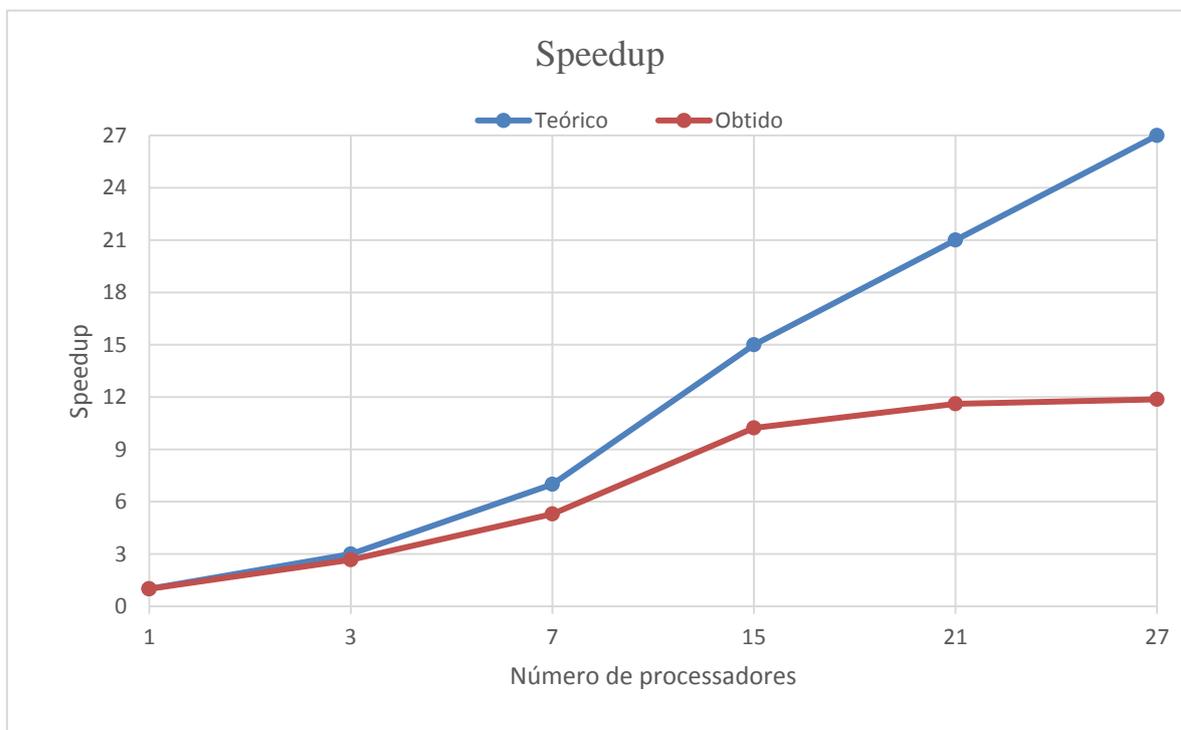
CAPÍTULO VI

Tabela 6.12 – Valores obtidos para as métricas de caracterização do sistema paralelo durante o processo de obtenção da imagem de um perfil de bolhas com um ruído numérico de 3%.

Número de processadores escravos	Tempo médio	Speedup	Eficiência	Overhead
1	37,637	1,000	1,000	0,000
3	14,103	2,669	0,890	0,124
7	7,112	5,292	0,756	0,323
15	3,681	10,225	0,682	0,467
21	3,242	11,609	0,553	0,809
27	3,171	11,869	0,440	1,275

Fonte: Autoria própria (2017)

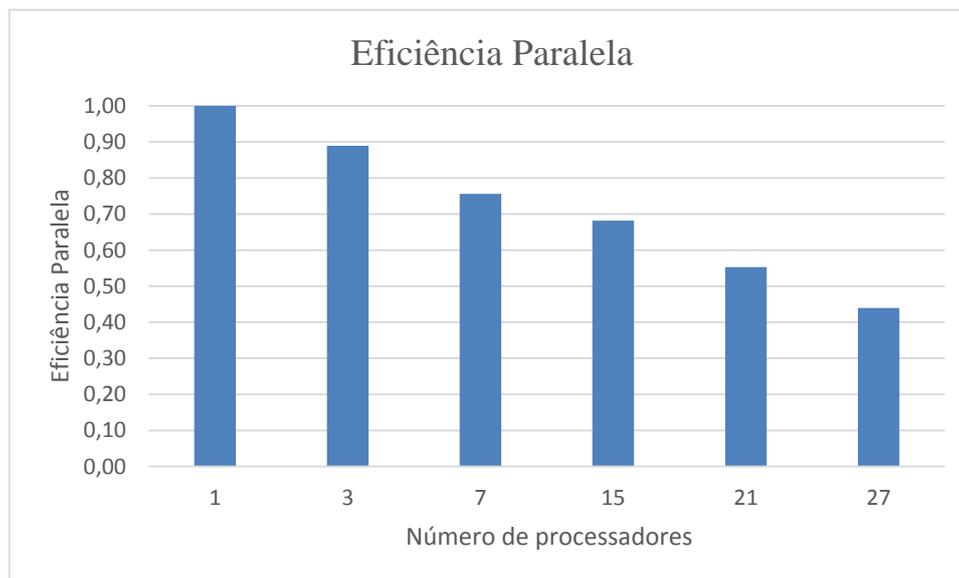
Figura 6.6 – Speedup obtido para um padrão de bolhas.



Fonte: Autoria própria (2017)

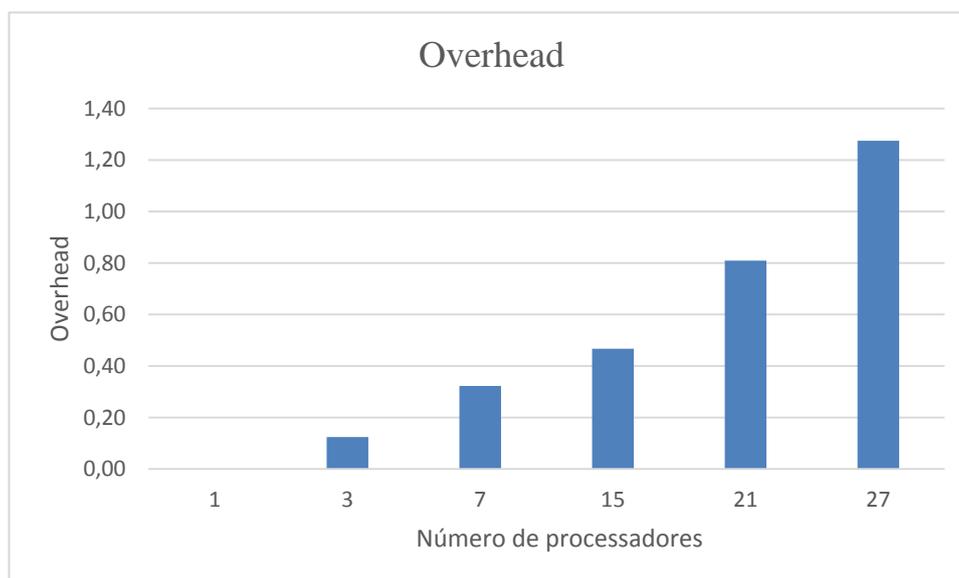
CAPÍTULO VI

Figura 6.7 – Eficiência paralela obtida para uma configuração de bolhas.



Fonte: Autoria própria (2017)

Figura 6.8 – Overhead obtido para uma configuração de bolhas.



Fonte: Autoria própria (2017)

Analisando o tempo de execução de cada porção do cálculo do algoritmo OEP serial é possível determinar um percentual de ~95% do código como sendo passível de ser paralelizável, sendo este um parâmetro crítico para determinação das métricas de caracterização do sistema proposto.

CAPÍTULO VI

A estimativa teórica da capacidade de paralelização de um código é descrita pela lei de Amdahl (POPOV et al., 2010), a qual é utilizada para determinar a máxima melhora esperada para um sistema em geral quando apenas uma única parte do mesmo é melhorada. Esta lei pode ser descrita matematicamente como:

$$S(p)_{amdahl} = \frac{1}{1 + \left(\frac{1}{p} - 1\right) \times par\%} \quad (6.7)$$

Onde $S(p)_{amdahl}$ é o valor teórico para o speedup dado pela lei de Amdahl e $par\%$ é a fração paralelizável do código em execução.

Tabela 6.13 – Valores obtidos para o speedup teórico de um sistema paralelo a partir da lei de Amdahl.

	0,94%	0,95%	0,96%	0,97%	0,98%	0,99%
1	1,00	1,00	1,00	1,00	1,00	1,00
3	2,68	2,73	2,78	2,83	2,88	2,94
7	5,15	5,38	5,65	5,93	6,25	6,60
15	8,15	8,82	9,62	10,56	11,72	13,16
21	9,55	10,50	11,67	13,13	15,00	17,50
27	10,55	11,74	13,24	15,17	17,76	21,43
50	12,69	14,49	16,89	20,24	25,25	33,56
100	14,41	16,81	20,16	25,19	33,56	50,25
200	15,46	18,26	22,32	28,69	40,16	66,89
500	16,16	19,27	23,85	31,31	45,54	83,47
1000	16,41	19,63	24,41	32,29	47,66	90,99
10000	16,64	19,96	24,94	33,23	49,76	99,02

Fonte: Autoria própria (2017)

Observando os resultados obtidos nos experimentos numéricos descritos na Tabela 6.12 e confrontando-os com os encontrados na Tabela 6.13, pode-se perceber uma boa concordância entre o valor teórico e o obtido para o sistema paralelo desenvolvido.

CONCLUSÃO

Neste trabalho foi desenvolvido o ferramental numérico-computacional responsável pela reconstrução de imagens provindas de um tomógrafo via capacitância elétrica com o intuito de observar de maneira não invasiva e não intrusiva o comportamento estático e dinâmico de escoamentos de meios multifásicos em tubulações.

Foi apresentado o modelo matemático que rege a propagação de campos elétricos no interior do sensor (problema direto), e como a variação de material influencia a resposta do sensor, além disso foi apresentado como os algoritmos convencionais e o proposto nesta tese executam ações para inferir a partir da resposta de capacitância nos sensores a distribuição de material com a sua respectiva distribuição geométrica (problema inverso).

Devido à dificuldade de resolução numérico computacional do problema direto e inverso da TCE, foi proposta a adoção de um sistema computacional (cluster Beowulf) a fim de processar estes problemas em um tempo hábil, além disso foi proposto um algoritmo que pudesse ser simples, robusto, preciso e que pudesse se apropriar da capacidade de paralelismo para ser utilizado no cluster desenvolvido.

Foram apresentados os resultados obtidos, mostrando que a ação de um algoritmo baseado em técnicas de inteligência de enxame e o da adoção de programação paralela conseguem atender ao aumento da qualidade sem onerar o tempo de obtenção da imagem.

Os resultados obtidos são encorajadores em relação ao potencial da técnica utilizada para obtenção das imagens via TCE. Mesmo em situações com ruído de sinal, consegue-se observar seja pela inspeção visual ou pelos indicadores discutidos anteriormente uma boa concordância com o modelo proposto de permissividade para as diversas configurações propostas de padrões de escoamentos e os resultados obtidos. Entretanto os valores de tempo de processamento encontrados ainda estão longe dos valores requeridos para o processo de reconstrução on-line, mas este tipo de abordagem para a solução do problema possui um futuro promissor para aplicações que não necessitem de uma alta frequência de amostragem do sinal de capacitância, tal como no problema de determinação de deposição de parafina

nas paredes das tubulações e no processo de controle de escoamentos de óleos pesados com lubrificação a água, requerendo assim investigações futuras que possam melhorar ainda mais os resultados aqui obtidos.

Um dos pontos a serem observados na continuidade deste trabalho é a criação de um gerador de malha adaptativo, que possa ampliar a discretização da malha nas regiões onde as linhas de campo elétrico sejam mais densas e reduzir em pontos que o campo seja suave. Outro ponto também a ser considerado é a implementação do MEF utilizando elementos triangulares quadráticos em vez de elementos lineares, considerando que com esta ação possa-se obter uma aproximação mais precisa do potencial no domínio em estudo, sem a necessidade da adoção de uma discretização muito fina.

Em relação ao aumento do tempo de processamento com os refinamentos da malha apresentados anteriormente, uma das propostas de solução deste problema pode ser a implementação de um estudo de complexidade algorítmica em cada etapa da rotina buscando assim mudanças em partes do código que apresentem uma relação aceitável entre a acurácia da solução e o tempo de processamento otimizando a solução em custo de processamento, dentre algumas possibilidades podemos citar: reordenação da conectividade dos elementos no intuito de diminuição da largura de banda da matriz de influência, redução da discretização da malha na região do tubo e mudanças na configuração da malha na região da tubulação, em vez do uso de elementos de área, utilizar elementos de barra e etc.

REFERÊNCIAS BIBLIOGRÁFICAS

ADAIME, L. M., “Aplicação do visualization toolkit para pós-processamento de análises pelo método dos elementos finitos”. Dissertação (mestrado), Setores de Tecnologia e Ciências Exatas, Programa de Pós-graduação em Métodos Numéricos em Engenharia, Universidade Federal do Paraná, Curitiba, 2005

ALECRIM, E., “Cluster: principais conceitos” - Disponível em: <http://www.infowester.com/cluster.php> - Publicado em 22/03/2013, Acessado em: 24/01/2017.

ALMEIDA, V. S.; PAIVA, J. B. “Uma adaptação do MEF para análise em multi computadores: aplicações em alguns modelos estruturais”, *Cadernos de Engenharia de Estruturas*, São Carlos, v.7, n. 23, p. 01-32, 2005.

ALSWAITTI, M., ALBUGHDADI, M., ISA, N. A. M., “Density-based particle swarm optimization algorithm for data clustering”, In *Expert Systems with Applications*, Volume 91, 2018, Pages 170-186.

AMOSHAHY, M. J., SHAMSI, M., SEDAAGHI, M. H., “A Novel Flexible Inertia Weight Particle Swarm Optimization Algorithm”. *PLoS ONE* 11(8): e0161558. doi:10.1371/journal.pone.0161558. 2016.

ANDRADE, C., MUREN, M., SCHNEIDER, F., & VICENCIO, F. “Projeto de bancada experimental para analisar o escoamento bifásico líquido-gás em uma tubulação horizontal”. *Anais do EVINCI - UniBrasil*. 2016.

ADLER, A. and LIONHEART, W. R. B., “Uses and abuses of EIDORS: An extensible software base for EIT” *Physiol. Meas.*, 27:S25-S42, 2006.

ATKINSON, I.; THEUVENY, B.; BERNARD, M.; CONORT, G.; GROVES, J.; LOWE, T.; MCDIARMID, A.; MEHDIZADEH, P.; PERCIOT, P.; PINGUET, B.; SMITH, G.; WILLIAMSON, K. A New Horizon in Multiphase Flow Measurement. *Oilfield Review* 16 n° 4, p.52-63, 2004.

AZEVEDO FILHO, A. C., “Desenvolvimento de um Sistema de Tomografia Capacitiva para Processos Multifásicos”, Tese (Doutorado), 306p, Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas, Campinas, São Paulo, 2002.

BARBOSA JR., JADER R. “Aspectos fenomenológicos e modelagem de escoamentos bifásicos gás-líquido”. In: Oscar M. H. Rodriguez. (Org.). Escoamento Multifásico. 1ª ed. Rio de Janeiro: ABCM, 2011, v. 1, p. 1-71.

BARRET, R. *et al*, “Templates for the solution of linear systems: Building blocks for iterative methods”, Philadelphia: SIAM, 1994, 105p.

BECK, M. S., and WILLIAMS, R. A., “Process Tomography—Principles, Techniques and Applications”, (Oxford: Butterworth-Heinemann), 1995.

BELO, F. A. “Aplicação da análise eletrônica ao estudo do escoamento multifásico”. Tese (Doutorado), Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas, Campinas, São Paulo, 1995.

BONABEAU, E., DORIGO, M., and THERAULAZ, G., “Swarm Intelligence: From Natural to Artificial Systems”, Oxford University Press, New York, USA, 1999.

BRAGA, A. A. C., “Aspectos técnicos envolvidos na construção de um cluster Beowulf”, Química Nova, Vol. 26, No. 3, 401-406, 2003.

CARLISLE, A. AND DOZIER, G., “An Off-The-Shelf PSO”, in Workshop Particle Swarm Optimization, Indianapolis, 2001.

CLEGHORN, C. W., “A Generalized theoretical deterministic particle swarm model”, MSc dissertation, University of Pretoria, Pretoria, 2013.

CLERC, M. and KENNEDY, J.: The particle swarm-explosion, stability and convergence in a multidimensional complex space. IEEE Trans. Evol. Comput. Vol.6, no.2, pp.58-73, Feb. 2002.

COUTINHO, F. R., “Um novo método ultrassônico para detecção da posição da interface em escoamentos bifásicos ar-água”. Tese (Doutorado), Universidade Tecnológica Federal do Paraná, Curitiba, 2014.

CRABTREE, M. A., “Industrial flow measurement”, Masters thesis, University of Huddersfield, 2009.

DAS, S. AND ABRAHAM, A. AND KONAR, A., “Particle Swarm Optimization and Differential Evolution Algorithms”: Technical Analysis, Applications and Hybridization Perspectives, Advances of Computational Intelligence in Industrial Systems. SCI, vol. 116, pp. 1–38. Springer, Berlin (2008)

DAVIS, T. A., “Direct methods for sparse linear systems”, SIAM, Philadelphia, sept. 2006, part of the SIAM book series on the fundamentals of algorithms.

DYAKOWSKI, T., EDWARDS, R. E., XIE, C. G., WILLIAMS, R. A., “Applications of capacitance tomography to gas-solid flow,” Chem. Eng. Sci., vol. 52, no. 13, pp. 2099–2110, 1997.

EBERHART, R. C., SIMPSON, P. K., & DOBBINS, R. W., “Computational intelligence PC tools”. Boston: Academic Press. 1996.

- EBERHART, R. C., and SHI, Y., “Particle swarm optimization: developments, applications and resources”. Proc. Congress on Evolutionary Computation 2001, Seoul, Korea. Piscataway, NJ: IEEE Service Center.
- FACCINI, J. L. H. “Estudo de Escoamentos Bifásicos Usando Técnicas Ultra-Sônicas”, Tese (Doutorado), 177f, Universidade Federal do Rio de Janeiro, 2008.
- FANG, W. and CUMBERBATCH, E., “Matrix properties of data from electrical capacitance tomography”, *Journal of Engineering Mathematics* (2005) 51: 127–146.
- FELIPPA, C. A., “Introduction to Finite Element Methods”, Notas de aula da disciplina Introduction to Finite Elements Methods (ASEN 5007), Aerospace Engineering Sciences Department, University of Colorado at Boulder, USA, 2004, Disponível em: <<https://www.colorado.edu/engineering/CAS/courses.d/IFEM.d/Home.html>>. Acesso em: 16/11/2017.
- FENNER, ROGER T. “Finite Element Methods for Engineers”, Imperial College press. London, 2012.
- FERREIRA, Aurélio Buarque de Holanda. Novo Aurélio Século XXI: o dicionário da língua portuguesa. 3 ed. totalmente rev. e ampl. Rio de Janeiro: Nova Fronteira, 1999.
- FINLAYSON, B. A., “The Method of Weighted Residuals and Variational Principles”, *Classics in Applied Mathematics*, University of Washington, Seattle, Washington, 2014.
- FORTUNA, A. O., “Técnicas Computacionais para Dinâmica dos Fluidos: Conceitos Básicos e Aplicações”, São Paulo: Editora da Universidade de São Paulo. 2000
- FULTON, W. S., “Optimising the time to solution in electrical impedance tomography”, *IEEE Proc. Sci. Meas. Technol.*, Vol. 142 N°6, November, 1995.
- GARCÍA-BERROCAL, A.; MONTALVO, C.; BLAZQUEZ, J.; BALBAS, M., "Flow measurement of liquid hydrocarbons with positive displacement meters: The correction for slippage", *Measurement Science and Technology*, 2013, Vol. 24(5).
- GATKOWSKI, M. J., BUCHNER, T., NIEWINSKI, G., MAZGAJ, P., "Development of a measurement and reconstruction system for determining the phase distribution in a two-phase flow vertical tube using Electrical Impedance Tomography", *JOURNAL OF POWER TECHNOLOGIES*, vol 94, N° 5, 2014.
- GROSAN, C. *et al.*: “Swarm Intelligence in Data Mining”, *Studies in Computational Intelligence (SCI)* 34, 1–20, 2006.
- HEPPNER, H., & GRENANDER, U., "A stochastic non-linear model for coordinated bird flocks. In S. Krasner (Ed.), *The ubiquity of chaos* (pp. 233–238). Washington: AAAS., 1990.
- HESTENES, M. R., STIEFEL, E. “Methods of Conjugate Gradients for Solving Linear Systems”, *Journal of Research of the National Bureau of Standards*, v. 49, n. 6, dez. 1952.
- HUANG, S.M., XIE, C.G., THORN, R., SNOWDEN, D., AND BECK, M. S., "Design of sensor electronics for electrical capacitance tomography", *IEE Proceedings-g*, vol. 139, n° I, February 1992.

- HUANG, T. AND MOHAN, A. S., "A hybrid boundary condition for robust particle swarm optimization", in IEEE Antennas and Wireless Propagation Letters, vol. 4, pp. 112-117, 2005.
- HUMPHRIES, S. "Field Solutions on Computers", CRC Press, NY, 1997.
- INNOCENTE, M. S., "Population based methods: Particle swarm optimization - Development general purpose optimizer and applications", Master's Thesis, Universitat Politècnica de Catalunya, Barcelona, 2006.
- ISAKSEN, Ø. (1996), "A review of reconstruction techniques for capacitance tomography", Measurement Science and Technology, Vol. 7 No. 3, pp. 325-337
- ISMAIL, I., GAMIO, J. C., BUKHARI, S. F. A., AND YANG, W. Q., "Tomography for multi-phase flow measurement in the oil industry", Flow Measurement and Instrumentation, vol 16, no. 2-3, pp. 145-155, 2005.
- JANG, J. D., *et al*, "Modified iterative Landweber method in electrical capacitance tomography", 2006, Meas. Sci. Technol., 17, 1909-1917.
- JAQUIE, Kalinka Regina Lucas. "Extensão da Ferramenta de Apoio à Programação Paralela (F.A.P.P.) para ambientes paralelos virtuais". Dissertação, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 1999.
- KAUR, H., PRABAHAKAR, G., "An advanced clustering scheme for wireless sensor networks using particle swarm optimization", Proceedings on 2016 2nd International Conference on Next Generation Computing Technologies, NGCT 2016, art. no. 7877447, pp. 387-392.
- KAZADI, S., "Swarm engineering". Ph.D. thesis, California Institute of Technology, Pasadena, CA, USA, 2000.
- KENNEDY, J. and EBERHART, R. C. (1995). Particle swarm optimization. Proc. IEEE Int'l. Conf. on Neural Networks, IV, 1942–1948. Piscataway, NJ: IEEE Service Center.
- KOH, B.-I., GEORGE, A. D., HAFTKA, R. T., *et al*. "Parallel asynchronous particle swarm optimization", International Journal for Numerical Methods in Engineering, v. 67, n. 4, pp. 578–595, JUL 23 2006.
- KUO, R.J., WANG, M.J. & HUANG, T.W., "An application of particle swarm optimization algorithm to clustering analysis", Soft Comput (2011) 15: 533.
- KWON, Y. W., "The finite element method using MATLAB", CRC Press LLC, 1997
- LUERSEN, M. A. "Métodos numéricos para a engenharia - Introdução ao método dos elementos finitos", Centro Federal de Educação Tecnológica do Paraná, Departamento Acadêmico de Mecânica, 2000
- MIKKI, S. M., KISHK, A., "Improved particle swarm optimization technique using hard boundary conditions", Microwave and Optical Technology Letters 46(5):422 – 426, 2005

MOORE, G. E., "Cramming more components onto integrated circuits", *Electronics Magazine*, Volume 38, Number 8, April 19, 1965.

NING, X., XU, Y., LI, Y., LI, Y., "Particle swarm optimization-based time series data prediction", *Smart Innovation, Systems and Technologies*, 82, 2017, pp. 327-334.

OLIEMANS, R.V.A., POTS, B.F.M., e Trompé, N., "Modeling of Annular Dispersed Two-Phase Flow in Vertical Pipes", *Int. J. Multiphase Flow*, 12, 711-732, 1986.

OLIVEIRA, J. L. G., "Medição de vazão de escoamentos bifásicos utilizando tubo de Venturi ou placa de orifício associados a um sensor de fração de vazão com campo elétrico girante" Dissertação (Mestrado), programa de Pós-Graduação, Engenharia Mecânica, Universidade Federal de Santa Catarina, Santa Catarina, 2007.

PACHECO, S. P., "A User's Guide to MPI", 1998. Disponível em <https://www.researchgate.net/publication/2487381_A_User's_Guide_to_MPI> acessado em 16/11/2017

PACHECO, F., CAVALCANTE, C.C.P., MENDES, J.R.P., BANNWART, A.C., SERAPIÃO, A.B.S., "Identificação de padrões de escoamento multifásico e determinação da perda da carga com o auxílio de rede neurais", 4º PDPETRO, São Paulo, Brasil, 2007.

PALADINO, E. E., "Estudo do escoamento multifásico em medidores de vazão do tipo pressão diferencial", Tese (Doutorado), Universidade Federal de Santa Catarina, Centro Tecnológico. Programa de Pós-Graduação em Engenharia Mecânica, 2005.

PEREIRA, L. A., "Método dos Elementos Finitos Aplicado ao Eletromagnetismo", Notas de Aula, Escola de Engenharia, Universidade Federal do Rio Grande do Sul, 2000.

POPOV, G., MASTORAKIS, N. and MLADENOV, V., "Calculation of the Acceleration of Parallel Programs as a Function of the Number of Threads", In: *Proceeding ICCOMP'10 Proceedings of the 14th WSEAS international conference on Computers - Volume II*, 2010. - pp. 411-414. - ISBN: 978-960-474-213-4.

PRADEEPKUMAR, D., RAVI, V., "Forecasting financial time series volatility using Particle Swarm Optimization trained Quantile Regression Neural Network", *Applied Soft Computing*, Vol 58, 2017, pp 35-52.

QOLOMANY, B., MAABREH, M., AL-FUQAHA, A., GUPTA, A., BENHADDOU, D., "Parameters optimization of deep learning models using Particle swarm optimization", 2017 13th International Wireless Communications and Mobile Computing Conference, IWCMC 2017, art. no. 7986470, pp. 1285-1290.

RAO, S. MADHUSUDANA, KEWU ZHU, CHI-HWA WANG, AND SUNDARESAN SANKARAN, "Electrical Capacitance Tomography Measurements on the Pneumatic Conveying of Solids", *Ind. Eng. Chem. Res.* 2001, 40, 4216-4226.

REINECKE, N. and MEWES, D., "Recent developments and industrial/research applications of capacitance tomography", *Meas. Sci. Technol.* 7 (1996) 233–246.

REYNOLDS, C. W., "Flocks, Herds, and Schools: A Distributed Behavioral Model, in Computer Graphics", 21(4) (SIGGRAPH '87 Conference Proceedings), 1987 pages 25-34.

REZENDE, V., "O método de Galerkin", Dissertação (Mestrado), Departamento de Matemática, Universidade Estadual de Maringá, Maringá, PR, Brasil, 2005.

SAAD, Y., "Iterative methods for sparse linear systems", Second Edition, Paperback, April 30, 2003.

SANTOS, B. M. O., "Simulação de Sistemas Eletromagnéticos Usando Método de Elementos Finitos e Método de Diferenças Finitas no Tempo", Rio de Janeiro: UFRJ/Escola Politécnica, 2017.

SCHUTTE, J., REINBOLT, J., FREGLY, B., *et al.* "Parallel global optimization with the particle swarm algorithm", International Journal for Numerical Methods in Engineering, v. 61, n. 13, pp. 2296–2315, 2004.

SHARMA, R., "Analysis of Electromagnetic Field Using FEM: A Review". International Journal of Advanced Electrical Engineering, North America, 1, oct. 2012.

SHI Y., EBERHART R. C., "A modified particle swarm optimizer". The Evolutionary Computation Proceedings, 1998 IEEE World Congress on Computational Intelligence, 1998 IEEE International Conference on; 1998.

SHI, Y. and EBERHART, R. C. "Parameter selection in particle swarm optimization. In Evolutionary Programming" VII: Proc. EP98, New York: Springer-Verlag, pp. 591-600. 1998.

SILVA, C. B. C., FILHO, M. J. B., PINHEIRO, J. A., "Medição De Vazão E Propriedades Em Escoamento Multifásico: Solução Econômica Para Diferentes Atividades Industriais". Boletim Técnico PETROBRAS, Rio Janeiro, vol. 43 (1) 45-61, jan./mar. 2000.

SILVA, A. L. F., FILHO, J. E. S., RAMALHO, J. B. V. S., MELO, M. V., LEITE, M. M., BRASIL, N. Í., JUNIOR, O. A. P., OLIVEIRA, R. C. G., ALVES, R. P., COSTA, R. F. D., KUNERT, R., GOMES, W., "Processamento Primário de Petróleo", Universidade Petrobrás. Escola de Ciências e Tecnologias E&P. Rio de Janeiro: Ed. Rosana Kunert, 2007.

SILVA, K. G., "Análise de escalabilidade de uma implementação paralela do simulated annealing acoplado". Dissertação (Mestrado), Universidade Federal do Rio Grande do Norte, Rio Grande do Norte, 2013.

SILVA, C. R. S., "Desenvolvimento de Formulações de Elementos Finitos para Problemas de Transferência de Calor", Dissertação (Mestrado), Engenharia Civil, FCT/UNL, Lisboa, 2014

SILVEIRA, T., OLIVEIRA, H., SILVA, L., SALGADO, R., "Controle de Inércia Não Monotônico na Otimização por Enxame de Partículas", Interdisciplinary Studies in Computer Science 69-82. ,July/December, 2009.

SNIR, M., OTTO, S., HUSS LADERMAN, S., WALKER, D., DONGARRA, J. "MPI - The complete reference". The MIT Press, Massachusetts, USA, 1996.

- SPINK, D. M., NORAST, J. M., “Developments and improvements in the solution of the forward problem in capacitance and impedance tomography”, *Transactions of the Institute of Measurement and Control*.1998; Vol. 20: pp. 186-194
- STEINER, G., “Application and data fusion of different sensor modalities in tomographic imaging”, *Elektrotechnik & Informationstechnik*, 124/7/8: 232–239, 2007.
- STERLING, T. “Beowulf Cluster Computing with Linux”, Massachusetts: Mit Press, 2001.
- SUN, M., LIU, S., LEI, J., LI, Z., "Concentration and angular velocity measurement in a cyclone separator dipleg using electrical capacitance tomography", *Science Bulletin*, 53, 2, 2008.
- TAITEL, Y., DUKLER A. E., “A model for predicting flow regime transitions in horizontal and near horizontal gas-liquid flow”. *AICHE Journal*, v. 22, p. 47-55, 1976.
- THORN, R., JOHANSEN, G. A., HJERTAKER, B. T., “Three-phase flow measurement in the petroleum industry” *Measurement Science and Technology*, Volume 24, Number 1, 2013
- VIVEROS, E. R. C., *Ajuste Coordenado de Controladores de Sistemas de Potência usando Metaheurísticas*, COPPE/UFRJ, Engenharia Elétrica, Rio de Janeiro, 2007.
- WEBER, G. H., LONGO, J. P. N., MURAKAMI, P. H. W., “Aplicação de sensores capacitivos para monitoramento da formação de parafina em oleodutos” *Trabalho de Conclusão de Curso (Graduação) – Universidade Tecnológica Federal do Paraná, Curitiba*, 2013.
- WIRGIN, A., “The inverse crime”, arXiv preprint arXiv:mathph/0401050, 2004.
- XIE, C. G., HUANG, S. M., HOYLE, B. S., THORN, R., LENN, C. AND BECK, M. S., "Electrical capacitance tomography for flow imaging – system model for development of reconstruction algorithms and design of primary sensors" *IEE Proc. G* 139 89–98, 1992.
- XU, S., and RAHMAT-SAMII, Y. , "Boundary Conditions in Particle Swarm Optimization Revisited," in *IEEE Transactions on Antennas and Propagation*, vol. 55, no. 3, pp. 760-765, March 2007.
- YANG, W. Q., “Hardware design of electrical capacitance tomography systems”, *Meas. Sci. Technol.* 7 (1996) 225–232.
- YANG, W.Q. AND PENG, L. “Image reconstruction algorithms for electrical capacitance tomography”, *Meas. Sci. Tech.* 14, 2003, pp. 1-13.
- YUSUF, A., MUTTAKIN, I., WIDADA, W., TARUNO, W. P., "Analysis of single excitation signal for high speed ECVT data acquisition system", *Information Technology and Electrical Engineering (ICITEE) 2014 6th International Conference on*, pp. 1-6, 2014, October.
- ZIELINSKI, K. & LAUR, R., “Stopping Criteria for a Constrained Single-Objective Particle Swarm Optimization Algorithm”. *Informatika (Slovenia)*, 31, 51-59, 2007.

ZUBER, N., FINDLAY, J.A. "Average Volumetric Concentration in Two-Phase Flow Systems", J. Heat Transfer, v. 87, p. 453-468, 1965.

APÊNDICE A

ROTINA OEP SERIAL

```
PROGRAM OEP_SERIAL_CFA
!
! Este Programa se Propõe a Resolver o Problema inverso da Técnica de Tomografia Capacitiva Elétrica
! Utilizando o Método de Enxame de Partículas na Forma Serial
!
! Programa Desenvolvido por Bruno Cesar B. N. De Souza
! Email: brunocbns@gmail.com
! Universidade Federal Da Paraíba (UFPB)
! Programa De Pós-Graduação Em Engenharia Mecânica (PPGEM)
! Grupo de Pesquisa em Instrumentação e Controle em Estudo de Energia e Meio Ambiente (GPICEEMA)
!
USE VARIAVEL
USE PROINVERSO
IMPLICIT NONE
INTEGER :: N, MINN, MAXN, MAXITER, MAXITER1, ITER, DIMEN, I, J, K, MG_POS, CONT1
INTEGER :: COND1, COND2
INTEGER, DIMENSION(:), ALLOCATABLE :: CONT
DOUBLE PRECISION :: AREA_MAX
DOUBLE PRECISION :: ERRO, C1, C2, R3, MAXW, MINW, W, CHI, PHI, MG_FITNESS
DOUBLE PRECISION :: ENERG, ENTROP, F_RAIO, F_FITNESS, SOMA, SOMAQUAD, RAI0
DOUBLE PRECISION, DIMENSION(:), ALLOCATABLE :: FITNESS, ML_FITNESS, STDDEV
DOUBLE PRECISION, DIMENSION(:), ALLOCATABLE :: R, R1, R2, EPSILON, AREA
DOUBLE PRECISION, DIMENSION(:, :), ALLOCATABLE :: POSICAO, ML, VELOCIDADE
DOUBLE PRECISION, DIMENSION(:, :), ALLOCATABLE :: CAP2
DOUBLE PRECISION :: INICIO, INICIO2, FIM
LOGICAL :: QUIT
!-----!
!----- INICIO DO ALGORITMO INVERSO -----!
!-----!
CALL INPUT
OPEN(UNIT=4,FILE='RESULTADO_PERMISSIVIDADE.dat')
OPEN(UNIT=10,FILE='RESULT_CAPACITANCIA.dat')
!-----!
!----- Inicialização dos parâmetros -----!
!-----!
DIMEN=NIND                                ! Dimensão do problema
MAXN=2000; MINN=20;   N=MAXN              ! Tamanho máximo e mínimo do enxame
MAXITER = 20*DIMEN                        ! Número máximo de iterações
ERRO=1e-2                                  ! Erro admissível
C1=2.1; C2=2.1                             ! Parâmetros de cognição e interação social
PHI=C1+C2; CHI=2/ABS(PHI-2+SQRT(PHI**2-4*PHI)) ! Parâmetro de constrição
```

```

MAXW=0.9; MINW=0.4; W=MAXW           ! Parâmetro de inércia máximo e mínimo
!-----!
!----- Alocação Dinâmica de Variáveis -----!
!-----!
ALLOCATE(VELOCIDADE(DIMEN,N))
ALLOCATE(POSICAO(DIMEN,N))
ALLOCATE(ML(DIMEN,N))
ALLOCATE(ML_FITNESS(N))
ALLOCATE(FITNESS(N))
ALLOCATE(R(DIMEN))
ALLOCATE(R1(DIMEN))
ALLOCATE(R2(DIMEN))
ALLOCATE(CONT(N))
ALLOCATE(EPSILON(NEL))
ALLOCATE(AREA(NEL))
ALLOCATE(STDDEV(DIMEN))
ALLOCATE(CAP2(NCEL,NCEL))
!-----!
!----- Inicialização da posição e velocidade das partículas -----!
!-----!
CALL CPU_TIME(INICIO)
CALL INIT_RANDOM_SEED ()
DO J = 1, N
    CALL RANDOM_NUMBER(R)
    VELOCIDADE(:,J) = R
    POSICAO(:,J) = 3.0 + (80-3)*R
    ML(:,J) = POSICAO(:,J)
    ML_FITNESS(J) = 1000000
    CONT(J) = 0
END DO
MG_FITNESS = 1000000
MG_POS = 1
F_FITNESS = 1000000
F_RAIO = 1000000
RAIO = -10000000
COND1 = 0
COND2 = 0
CALL CPU_TIME(INICIO2)
N=MINN
ITER = 1
QUIT = .FALSE.
DO WHILE ( ITER <= MAXIT )
    W=MAXW-(MAXW-MINW)*ITER/MAXITER
    DO J = 1, N
        CALL RANDOM_NUMBER(R1)
        CALL RANDOM_NUMBER(R2)
        VELOCIDADE(:,J) = W*VELOCIDADE(:,J)+C1*R1*(ML(:,J)-POSICAO(:,J))+C2*R2*(ML(:,MG_POS)-POSICAO(:,J))
        POSICAO(:,J) = POSICAO(:,J)+CHI*VELOCIDADE(:,J)
!-----!
!----- Condição de fronteira -----!
!-----!
    DO K = 1, DIMEN
        IF (POSICAO(K,J) < 3.0) THEN
            POSICAO(K,J) = 3.0
            VELOCIDADE(K,J) = 0.0
        END IF
    END DO

```

```

        IF (POSICAO(K,J) > 80.0) THEN
            POSICAO(K,J) = 80.0
            VELOCIDADE(K,J) = 0.0
        END IF
    END DO
END DO

!----- Avaliação da população -----!

DO J = 1, N
    CALL ENERGIA(DIMEN,POSICAO(:,J),2,ENERG)
    FITNESS(J) = ENERG
END DO

!----- Atualização do melhor individuo da população -----!

DO J = 1, N
    IF ((ML_FITNESS(J)-FITNESS(J))/FITNESS(J) > 0.01) THEN
        ML(:,J) = POSICAO(:,J)
        ML_FITNESS(J) = FITNESS(J)
        CONT(J) = 0
    ELSE
        CONT(J) = CONT(J)+1
    END IF
    IF ((MG_FITNESS-FITNESS(J))/FITNESS(J) > 0.10) THEN
        MG_POS = J
        MG_FITNESS = FITNESS(J)
        CONT1 = 0
    ELSE
        CONT1 = CONT1 + 1
    END IF
END DO

!----- Procedimento de fuga de mínimos locais -----!

DO J = 1, N
    IF (CONT(J) >= 10) THEN
        CALL RANDOM_NUMBER(R3)
        K = floor(R3*(MAXN-N+1))+N
        POSICAO(:,J)=POSICAO(:,K)
    END IF
END DO

!----- Atualização dinâmica de partículas -----!

IF (CONT1 > 5*N) THEN
    IF (N < MAXN) THEN
        N = N+MINN
    ELSE
        N = MAXN
    END IF
ELSE
    IF (N > MINN) THEN
        N = N-MINN
    ELSE
        N = MINN
    END IF
END IF

```

```

        END IF
    END IF

!-----!
!----- Condição de parada 01 -----!
!-----!

    IF (MOD(ITER, 50) .EQ. 0) THEN
        IF ((F_FITNESS-MG_FITNESS)/F_FITNESS > 0.15) THEN
            F_FITNESS = MG_FITNESS
        ELSE
            COND1 = 1
        END IF
    END IF

!-----!
!----- Condição de parada 02 -----!
!-----!

    IF (MOD(ITER, 50) .EQ. 0) THEN
        DO K = 1, DIMEN
            SOMA = 0
            SOMAQUAD = 0
            DO I = 1, N
                SOMA = SOMA + POSICAO(K,I)
                SOMAQUAD = SOMAQUAD + (POSICAO(K,I))**2
            END DO
            STDDEV(K) = ABS(SQRT((SOMAQUAD - SOMA*SOMA/N)/(N-1)))
            IF (STDDEV(K) > RAI0) THEN
                RAI0 = STDDEV(K)
            END IF
        END DO
        IF (ABS(F_RAIO-RAIO)/F_RAIO > 0.15) THEN
            F_RAIO = RAI0
        ELSE
            COND2 = 1
        END IF
    END IF

!-----!
!----- Análise do resultado -----!
!-----!

    IF ((ITER >= MAXITER) .OR. (MG_FITNESS <= ERRO) .OR. (COND1 == 1) .OR. (COND2 == 1)) THEN
        QUIT = .TRUE.
        CALL CPU_TIME(FIM)
        WRITE(4,*) 'TAMANHO FINAL DO ENXAME = ', N
        WRITE(4,*) 'TEMPO TOTAL = ', FIM - INICIO
        WRITE(4,*) 'TEMPO PARCIAL 1 = ', INICIO - INICIO2
        WRITE(4,*) 'TEMPO PARCIAL 2 = ', FIM - INICIO2
        WRITE(4,*) 'ITERACAO = ', ITER
        WRITE(4,*) 'VALOR DO MINIMO = ', MG_FITNESS
        WRITE(4,*) 'CONDIÇÃO DE PARADA 01', COND1
        WRITE(4,*) 'CONDIÇÃO DE PARADA 02', COND2
        WRITE(4,*) 'MELHOR ESTIMATIVA = ', ML(:,MG_POS)
        WRITE(10,'(8F12.4)') ((CAP1(I,J),J=1,NCEL),I=1,NCEL)
        WRITE(10,*)'-----'
        EPSILON = EPS
        EPSILON(IND) = ML(:,MG_POS)
        CALL PRODIRETO(EPSILON,CAP2,AREA_MAX,AREA)
        WRITE(10,'(8F12.4)') ((CAP2(I,J),J=1,NCEL),I=1,NCEL)
    ELSE

```

```
        ITER = ITER + 1
    END IF
END DO
CLOSE(UNIT=4)
CLOSE(UNIT=10)
STOP
END PROGRAM OEP_SERIAL
```

APÊNDICE B

ROTINA OEP PARALELO

```
PROGRAM OEP_PARALELO_CFA
!
! Este programa se propõe a resolver o problema inverso da técnica de tomografia capacitiva
! elétrica utilizando o método de enxame de partículas com a subdivisão de tarefas utilizando o
! paradigma mestre-escravo.
!
! Programa Desenvolvido por Bruno Cesar B. N. De Souza
! Email: brunocbns@gmail.com TEL:(83) 3216-7514
! Universidade Federal Da Paraíba (UFPB)
! Programa De Pós-Graduação Em Engenharia Mecânica (PPGEM)
! Laboratório De Energia Solar (LES)
!
USE VARIAVEL
USE PROINVERSO
USE MPI
IMPLICIT NONE
INTEGER, PARAMETER :: MASTER = 0, TAG1=10, TAG2=20
INTEGER :: NUMTASKS, NUMWORKERS, TASKID, DEST, INDEX, ARRAYMSG
INTEGER :: INDEXMSG, SOURCE, CHUNKSIZE, EXTRA, NCOL, MPIERR
INTEGER, DIMENSION(MPI_STATUS_SIZE) :: STAT
INTEGER :: N, MINN, MAXN, MAXITER, MAXITER1, ITER, DIMEN, I, J, K, MG_POS, CONT1
INTEGER :: COND1, COND2, QUIT
INTEGER, DIMENSION(:), ALLOCATABLE :: CONT
DOUBLE PRECISION :: ERRO, C1, C2, R3, MAXW, MINW, W, CHI, PHI, G_FITNESS
DOUBLE PRECISION :: ENERG, ENTROP, F_RAIO, F_FITNESS, SOMA, SOMAQUAD, RAIO
DOUBLE PRECISION, DIMENSION(:), ALLOCATABLE :: G, FITNESS, P_FITNESS, STDDEV
DOUBLE PRECISION, DIMENSION(:), ALLOCATABLE :: R, R1, R2, EPSILON
DOUBLE PRECISION, DIMENSION(:,:), ALLOCATABLE :: POSICAO, VELOCIDADE, P
DOUBLE PRECISION, DIMENSION(:,:), ALLOCATABLE :: CAP2
DOUBLE PRECISION :: INICIO, FIM
!-----!
!                               INICIO DO ALGORITMO INVERSO -----!
!-----!
CALL INPUT
!-----!
!----- Inicialização do ambiente paralelo -----!
!-----!
CALL MPI_INIT(MPIERR)
CALL MPI_COMM_RANK(MPI_COMM_WORLD,TASKID,MPIERR)
CALL MPI_COMM_SIZE(MPI_COMM_WORLD,NUMTASKS,MPIERR)
!-----!
```

```

!----- Inicialização dos parâmetros -----!
!-----!
DIMEN=NIND                                ! Dimensão do problema
MAXN=2000; MINN=20;  N=MAXN                ! Tamanho máximo e mínimo do enxame
MAXITER = 20*DIMEN                        ! Número máximo de iterações
ERRO=1e-2                                  ! Erro admissível
C1=2.1;  C2=2.1                            ! Parâmetros de cognição e interação social
PHI=C1+C2; CHI=2/ABS(PHI-2+SQRT(PHI**2-4*PHI)) ! Parâmetro de constrição
MAXW=0.9;  MINW=0.4;  W=MAXW              ! Parâmetro de inércia máximo e mínimo
!-----!
!----- Alocação Dinâmica de Variáveis -----!
!-----!
ALLOCATE(VELOCIDADE(DIMEN, N))
ALLOCATE(POSICAO(DIMEN, N))
ALLOCATE(P(DIMEN, N))
ALLOCATE(G(DIMEN))
ALLOCATE(FITNESS(N))
ALLOCATE(P_FITNESS(N))
ALLOCATE(R(DIMEN))
ALLOCATE(R1(DIMEN))
ALLOCATE(R2(DIMEN))
ALLOCATE(CONT(N))
ALLOCATE(EPSILON(NEL))
ALLOCATE(STDDEV(DIMEN))
!-----!
!----- Inicialização da posição e velocidade das partículas -----!
!-----!
IF (TASKID .eq. MASTER) THEN
  OPEN(UNIT=4,FILE='RESULTADO_PERMISSIVIDADE.dat')
  OPEN(UNIT=10,FILE='RESULTADO_CAPACITANCIA.dat')
  INICIO = MPI_Wtime()
  CALL INIT_RANDOM_SEED()
  DO J = 1, N
    CALL RANDOM_NUMBER(R)
    VELOCIDADE(:,J) = 0.3*R
    POSICAO(:,J) = 3.0 + (80.0-3.0)*R
    P(:,J) = POSICAO(:,J)
    CONT(J) = 0
  END DO
!-----!
!----- Envio para cada escravo de um conjunto de partículas -----!
!-----!
NUMWORKERS = NUMTASKS - 1
CHUNKSIZE = (N / NUMWORKERS)
EXTRA = mod(N, NUMWORKERS)
INDEX = 1
DO DEST=1, NUMWORKERS
  IF (DEST .le. EXTRA) THEN
    NCOL = CHUNKSIZE + 1
  ELSE
    NCOL = CHUNKSIZE
  END IF
  CALL MPI_SEND(INDEX, 1, MPI_INTEGER, DEST, TAG1, MPI_COMM_WORLD, MPIERR)
  CALL MPI_SEND(NCOL, 1, MPI_INTEGER, DEST, TAG1, MPI_COMM_WORLD, MPIERR)
  CALL MPI_SEND(POSICAO(1,INDEX), DIMEN*NCOL, MPI_DOUBLE_PRECISION, DEST, TAG1, MPI_COMM_WORLD,
MPIERR)

```

```

        INDEX = INDEX + NCOL
    END DO
!-----!
!----- Recebimento do valor de fitness de cada escravo -----!
!-----!

    DO SOURCE = 1, NUMWORKERS
        CALL MPI_RECV(INDEX, 1, MPI_INTEGER, SOURCE, TAG2, MPI_COMM_WORLD, STAT, MPIERR)
        CALL MPI_RECV(NCOL, 1, MPI_INTEGER, SOURCE, TAG2, MPI_COMM_WORLD, STAT, MPIERR)
        CALL MPI_RECV(FITNESS(INDEX), NCOL, MPI_DOUBLE_PRECISION, SOURCE, TAG2, MPI_COMM_WORLD, STAT,
MPIERR)
    END DO
    G_FITNESS = 1000000
    DO J=1, N
        P_FITNESS(J) = FITNESS(J)
        IF (FITNESS(J) < G_FITNESS) THEN
            G_FITNESS = FITNESS(J)
            G = P(:,J)
        END IF
    END DO
    F_FITNESS = 1000000
    F_RAIO = 1000000
    RAIO = -10000000
    COND1 = 0
    COND2 = 0
ELSE
!-----!
!----- Recebimento do conjunto de partículas por cada escravo -----!
!-----!

    CALL MPI_RECV(INDEX, 1, MPI_INTEGER, MASTER, TAG1, MPI_COMM_WORLD, STAT, MPIERR)
    CALL MPI_RECV(NCOL, 1, MPI_INTEGER, MASTER, TAG1, MPI_COMM_WORLD, STAT, MPIERR)
    CALL MPI_RECV(POSICAO, DIMEN*NCOL, MPI_DOUBLE_PRECISION, MASTER, TAG1, MPI_COMM_WORLD, STAT,
MPIERR)
    DO J = 1, NCOL
        CALL ENERGIA(DIMEN, POSICAO(:,J), 2, ENERG)
        CALL ENTROPIA(DIMEN, POSICAO(:,J), 2, ENTROP)
        FITNESS(J) = ENERG + ENTROP
    END DO
!-----!
!----- Envio de volta para o mestre do valor de fitness de cada partícula -----!
!-----!

    CALL MPI_SEND(INDEX, 1, MPI_INTEGER, MASTER, TAG2, MPI_COMM_WORLD, MPIERR)
    CALL MPI_SEND(NCOL, 1, MPI_INTEGER, MASTER, TAG2, MPI_COMM_WORLD, MPIERR)
    CALL MPI_SEND(FITNESS, NCOL, MPI_DOUBLE_PRECISION, MASTER, TAG2, MPI_COMM_WORLD, MPIERR)
END IF
N=MINN
ITER = 1
QUIT = 1
DO WHILE (QUIT.EQ. 1)
    IF (TASKID == MASTER) THEN
        W=MAXW-(MAXW-MINW)*ITER/MAXITER
        DO J = 1, N
            CALL RANDOM_NUMBER(R1)
            CALL RANDOM_NUMBER(R2)
            VELOCIDADE(:,J) = W*VELOCIDADE(:,J)+C1*R1*(P(:,J)-POSICAO(:,J))+C2*R2*(G-POSICAO(:,J))
            POSICAO(:,J) = POSICAO(:,J)+CHI*VELOCIDADE(:,J)
        END DO
    END IF
!-----!

```

!----- Condição de fronteira -----!
!-----!

```
DO K = 1, DIMEN
  IF (POSICAO(K,J) < 3.0) THEN
    POSICAO(K,J) = 3.0
    VELOCIDADE(K,J) = 0.0
  END IF
  IF (POSICAO(K,J) > 80.0) THEN
    POSICAO(K,J) = 80.0
    VELOCIDADE(K,J) = 0.0
  END IF
END DO
END DO
```

!----- Partição do enxame em subpopulações para cada escravo -----!
!-----!

```
NUMWORKERS = NUMTASKS-1
CHUNKSIZE = (N / NUMWORKERS)
EXTRA = mod(N, NUMWORKERS)
INDEX = 1
DO DEST=1, NUMWORKERS
  IF (DEST .le. EXTRA) THEN
    NCOL = CHUNKSIZE + 1
  ELSE
    NCOL = CHUNKSIZE
  END IF
  CALL MPI_SEND(INDEX, 1, MPI_INTEGER, DEST, TAG1, MPI_COMM_WORLD, MPIERR)
  CALL MPI_SEND(NCOL, 1, MPI_INTEGER, DEST, TAG1, MPI_COMM_WORLD, MPIERR)
  CALL MPI_SEND(POSICAO(1,INDEX), DIMEN*NCOL, MPI_DOUBLE_PRECISION, DEST, TAG1,
MPI_COMM_WORLD, MPIERR)
  INDEX = INDEX + NCOL
END DO
```

!----- Recebimento dos resultados de avaliação das subpopulações -----!
!-----!

```
DO SOURCE = 1, NUMWORKERS
  CALL MPI_RECV(INDEX, 1, MPI_INTEGER, SOURCE, TAG2, MPI_COMM_WORLD, STAT, MPIERR)
  CALL MPI_RECV(NCOL, 1, MPI_INTEGER, SOURCE, TAG2, MPI_COMM_WORLD, STAT, MPIERR)
  CALL MPI_RECV(FITNESS(INDEX), NCOL, MPI_DOUBLE_PRECISION, SOURCE, TAG2, MPI_COMM_WORLD, STAT,
MPIERR)
END DO
```

!----- Atualização do melhor individuo da população -----!
!-----!

```
DO J = 1, N
  IF ((P_FITNESS(J)-FITNESS(J))/FITNESS(J) > 0.05) THEN
    P_FITNESS(J) = FITNESS(J)
    P(:,J) = POSICAO(:,J)
    CONT(J) = 0
  ELSE
    CONT(J) = CONT(J)+1
  END IF
  IF ((G_FITNESS-FITNESS(J))/FITNESS(J) > 0.05) THEN
    G_FITNESS = FITNESS(J)
    G = P(:,J)
    CONT1 = 0
  END IF
END DO
```

```

ELSE
    CONT1 = CONT1 + 1
END IF
END DO
!-----!
!----- Procedimento de fuga de mínimos locais -----!
!-----!

DO J = 1, N
    IF (CONT(J) .GT. 10) THEN
        CALL RANDOM_NUMBER(R3)
        K = floor(R3*(MAXN-N+1))+N
        POSICAO(:,J)=POSICAO(:,K)
    END IF
END DO
!-----!
!----- Atualização dinâmica de partículas -----!
!-----!

IF (CONT1 > 5*N) THEN
    IF (N < MAXN) THEN
        N = N+MINN
    ELSE
        N = MAXN
    END IF
ELSE
    IF (N > MINN) THEN
        N = N-MINN
    ELSE
        N = MINN
    END IF
END IF
!-----!
!----- Condição de parada 01 -----!
!-----!

IF (MOD(ITER, 100) .EQ. 0) THEN
    IF ((F_FITNESS-G_FITNESS)/F_FITNESS > 0.1) THEN
        F_FITNESS = G_FITNESS
    ELSE
        COND1 = 1
    END IF
END IF
!-----!
!----- Condição de parada 01 -----!
!-----!

IF (MOD(ITER, 100) .EQ. 0) THEN
    DO K = 1, DIMEN
        SOMA = 0
        SOMAQUAD = 0
        DO I = 1, N
            SOMA = SOMA + POSICAO(K,I)
            SOMAQUAD = SOMAQUAD + (POSICAO(K,I))**2
        END DO
        STDDEV(K) = ABS(SQRT((SOMAQUAD - SOMA*SOMA/N)/(N-1)))
        IF (STDDEV(K) > RAI0) THEN
            RAI0 = STDDEV(K)
        END IF
    END DO
END IF

```

```

        IF (ABS(F_RAIO-RAIO)/F_RAIO > 0.1) THEN
            F_RAIO = RAIO
        ELSE
            COND2 = 1
        END IF
    END IF
ELSE
!-----!
!----- Recebimento das subpopulações de partículas em cada nodo escravo -----!
!-----!
    CALL MPI_RECV(INDEX, 1, MPI_INTEGER, MASTER, TAG1, MPI_COMM_WORLD, STAT, MPIERR)
    CALL MPI_RECV(NCOL, 1, MPI_INTEGER, MASTER, TAG1, MPI_COMM_WORLD, STAT, MPIERR)
    CALL MPI_RECV(POSICAO(1,INDEX), DIMEN*NCOL, MPI_DOUBLE_PRECISION, MASTER, TAG1, MPI_COMM_WORLD,
STAT, MPIERR)
!-----!
!----- Avaliação da população -----!
!-----!

    DO J = INDEX, INDEX + NCOL
        CALL ENERGIA(DIMEN,POSICAO(:,J),2,ENERG)
        FITNESS(J) = ENERG
    END DO

!-----!
!----- Envio dos resultados de avaliação das subpopulações -----!
!-----!
    CALL MPI_SEND(INDEX, 1, MPI_INTEGER, MASTER, TAG2, MPI_COMM_WORLD, MPIERR)
    CALL MPI_SEND(NCOL, 1, MPI_INTEGER, MASTER, TAG2, MPI_COMM_WORLD, MPIERR)
    CALL MPI_SEND(FITNESS(INDEX), NCOL, MPI_DOUBLE_PRECISION,MASTER, TAG2,MPI_COMM_WORLD,
MPIERR)
    END IF
    CALL MPI_BARRIER(MPI_COMM_WORLD, MPIERR)

!-----!
!----- Análise do resultado -----!
!-----!
    IF ( TASKID == MASTER ) THEN
        IF ((ITER .GE. MAXITER) .OR. (G_FITNESS .LE. ERRO) .OR. (COND1 .EQ. 1) .OR. (COND2 .EQ. 1)) THEN
            QUIT = 0
            DO DEST=1, NUMWORKERS
                CALL MPI_SEND(QUIT, 1, MPI_INTEGER, DEST, TAG1, MPI_COMM_WORLD, MPIERR)
            END DO
            FIM = MPI_WTIME()
            WRITE(*,*) 'ITERACAO = ', ITER
            WRITE(*,*) 'TAMANHO FINAL DO ENXAME = ', N
            WRITE(*,*) 'TEMPO TOTAL = ', FIM - INICIO
            WRITE(*,*) 'VALOR DO MINIMO = ', G_FITNESS
            WRITE(*,*) 'CONDIÇÃO DE PARADA 01 = ', COND1
            WRITE(*,*) 'CONDIÇÃO DE PARADA 02 = ', COND2
            WRITE(*,*) 'MELHOR ESTIMATIVA = ', G
            WRITE(4,*) 'ITERACAO = ', ITER
            WRITE(4,*) 'TAMANHO FINAL DO ENXAME = ', N
            WRITE(4,*) 'TEMPO TOTAL = ', FIM - INICIO
            WRITE(4,*) 'VALOR DO MINIMO = ', G_FITNESS
            WRITE(4,*) 'CONDIÇÃO DE PARADA 01 = ', COND1
            WRITE(4,*) 'CONDIÇÃO DE PARADA 02 = ', COND2
            WRITE(4,*) 'MELHOR ESTIMATIVA = ', G
            WRITE(10,'(8F12.4)') ((CAP1(I,J)),J=1,NCOL),I=1,NCOL)
        END IF
    END IF

```

```

WRITE(10,*)'-----'
ALLOCATE(CAP2(NCEL,NCEL))
EPSILON = EPS
EPSILON(IND) = G
CALL PRODIRETO(EPSILON,CAP2)
WRITE(10,'(8F12.4)') ((CAP2(I,J),J=1,NCEL),I=1,NCEL)
ELSE
QUIT = 1
DO DEST=1, NUMWORKERS
CALL MPI_SEND(QUIT, 1, MPI_INTEGER, DEST, TAG1, MPI_COMM_WORLD, MPIERR)
END DO
ITER = ITER + 1
WRITE(*,*) 'ITERACAO = ', ITER
WRITE(*,*) 'TAMANHO DO ENXAME = ', N
WRITE(*,*) 'FITNNES = ', G_FITNESS
WRITE(*,*) 'CONDIÇÃO DE PARADA 01 =', COND1
WRITE(*,*) 'CONDIÇÃO DE PARADA 02 =', COND2
WRITE(*,*) 'MELHOR ESTIMATIVA = ', G(1), G(5), G(12), G(25)
WRITE(*,*) '-----'
END IF
ELSE
CALL MPI_RECV(QUIT, 1, MPI_INTEGER, MASTER, TAG1, MPI_COMM_WORLD, STAT, MPIERR)
END IF
END DO
!-----!
!----- Fim do ambiente paralelo -----!
!-----!
CALL MPI_FINALIZE(MPIERR)
CLOSE(UNIT=4)
CLOSE(UNIT=10)
STOP
END PROGRAM OEP_ECT_PARALELO

```

APÊNDICE C

ROTINA TRL

PROGRAM TRL

!

! Este Programa se Propõe a resolver o Problema inverso da Técnica de
! Tomografia Capacitiva Elétrica Utilizando o Método de Retroprojeção Linear

!

! Programa Desenvolvido por Bruno Cesar B. N. De Souza

! Email: brunocbns@gmail.com TEL:(83) 3216-7514

! Universidade Federal Da Paraíba (UFPB)

! Programa De Pós-Graduação Em Engenharia Mecânica (PPGEM)

! Grupo de Pesquisa em Instrumentação em Controle em Estudos de Energia e

! Meio Ambiente (GPICEEMA)

!

!-----!

! INICIO DO ALGORITMO TRL !

!-----!

USE VARIAVEL

USE PROINVERSO

IMPLICIT NONE

INTEGER I, J, K, N, P, Q

DOUBLE PRECISION :: e_LOW, e_HIGH

DOUBLE PRECISION :: SOMA_TEMP, AREA_MAX, SOMA, SOMA1, SOMA2

DOUBLE PRECISION, DIMENSION(:), **ALLOCATABLE** :: AREA

DOUBLE PRECISION, DIMENSION(:), **ALLOCATABLE** :: EPSILON

DOUBLE PRECISION, DIMENSION(:,:), **ALLOCATABLE** :: CAP, S, CAP3

DOUBLE PRECISION, DIMENSION(:), **ALLOCATABLE** :: CAP2, CAP2_LOW, CAP2_HIGH

DOUBLE PRECISION, DIMENSION(:), **ALLOCATABLE** :: LAMBDA, CAP2_PERTUB, TEMP

DOUBLE PRECISION, DIMENSION(:), **ALLOCATABLE** :: G

OPEN(UNIT=4, FILE='RESULT_PERMISSIVIDADE.dat')

OPEN(UNIT=10, FILE='RESULT_CAPACITANCIA.dat')

CALL INPUT

N = NCEL*(NCEL-1)/2

e_LOW = 3.0

e_HIGH = 80.0

!-----!

! ALOCAÇÃO DINÂMICA DE VARIÁVEIS !

!-----!

ALLOCATE(CAP(NCEL,NCEL))

ALLOCATE(EPSILON(NEL))

ALLOCATE(AREA(NEL))

ALLOCATE(CAP2(N))

ALLOCATE(CAP2_LOW(N))

```

ALLOCATE(CAP2_HIGH(N))
ALLOCATE(LAMBDA(N))
ALLOCATE(CAP2_PERTUB(N))
ALLOCATE(TEMP(N))
ALLOCATE(S(N,NIND))
ALLOCATE(G(NIND))

```

```

!-----!
!           CAPACITÂNCIA OBTIDA DA DISTRIUIÇÃO A SER DETERMINADA           !
!-----!

```

```

K = 1
DO I = 1, NCEL - 1
  DO J = I + 1, NCEL
    CAP2(K) = CAP1(I,J)
    K = K + 1
  END DO
END DO

```

```

!-----!
!           CAPACITÂNCIA COM O SENSOR PREENCHIDO COM ELEMENTO DE MENOR PERMISSIVIDADE           !
!-----!

```

```

EPSILON = EPS
DO I = 1, NIND
  K = IND(I)
  EPSILON(K) = e_LOW
END DO
CALL PRODIRETO(EPSILON,CAP,AREA_MAX,AREA)
K = 1
DO I = 1, NCEL - 1
  DO J = I + 1, NCEL
    CAP2_LOW(K) = CAP(I,J)
    K = K + 1
  END DO
END DO

```

```

!-----!
!           CAPACITÂNCIA COM O SENSOR PREENCHIDO COM ELEMENTO DE MAIOR PERMISSIVIDADE           !
!-----!

```

```

EPSILON = EPS
DO I = 1, NIND
  K = IND(I)
  EPSILON(K) = e_HIGH
END DO
CALL PRODIRETO(EPSILON,CAP,AREA_MAX,AREA)
K = 1
DO I = 1, NCEL - 1
  DO J = I + 1, NCEL
    CAP2_HIGH(K) = CAP(I,J)
    K = K + 1
  END DO
END DO

```

```

!-----!
!           CÁLCULO DO VETOR DE CAPACITÂNCIA NORMALIZADO           !
!-----!

```

```

DO I = 1, N
  LAMBDA(I) = (CAP2(I) - CAP2_LOW(I)) / (CAP2_HIGH(I) - CAP2_LOW(I))
END DO

```

```

!-----!
!           CÁLCULO PARA MONTAGEM DA MATRIZ DE SENSIBILIDADE           !
!-----!

```

```
!-----!
CALL MATRIZ_SENSIBILIDADE(N,e_LOW,e_HIGH,CAP2_LOW,CAP2_HIGH,S)
WRITE(10,'(32F12.4)') ((S(I,J),J=1,NIND),I=1,N)
!-----!
```

```
!
! DETERMINAÇÃO DO VETOR DE PERMISSIVIDADE
!
!-----!
```

```
DO I = 1, NIND
  SOMA1 = 0
  DO J = 1, N
    SOMA1 = SOMA1 + S(J,I)
  END DO
  SOMA2 = 0
  DO J = 1, N
    SOMA2 = SOMA2 + S(J,I)*LAMBDA(J)/SOMA1
  END DO
  G(I) = SOMA2
END DO
DO I = 1, NIND
  IF (G(I) < 0.0) THEN
    G(I) = 0
  END IF
  IF (G(I) > 1.0) THEN
    G(I) = 1
  END IF
END DO
DO I = 1, NIND
  G(I)=(e_HIGH-e_LOW)*G(I)+e_LOW;
END DO
```

```
!-----!
!
! SAIDA DE DADOS
!
!-----!
```

```
WRITE(4,*) 'MELHOR ESTIMATIVA = ', G
WRITE(10,'(8F12.4)') ((CAP1(I,J),J=1,NCEL),I=1,NCEL)
WRITE(10,*)'-----'
ALLOCATE(CAP3(NCEL,NCEL))
EPSILON = EPS
EPSILON(IND) = G
CALL PRODIRETO(EPSILON,CAP3,AREA_MAX,AREA)
WRITE(10,'(8F12.4)') ((CAP3(I,J),J=1,NCEL),I=1,NCEL)
CLOSE(UNIT=4)
CLOSE(UNIT=10)
STOP
END PROGRAM TRL
```

APÊNDICE D

ROTINA TRA

```
PROGRAM TRA
!
! Este Programa se Propõe a resolver o Problema inverso da Técnica de Tomografia Capacitiva
! Elétrica Utilizando o Método de Reconstrução Algébrica
!
! Programa Desenvolvido por Bruno Cesar B. N. De Souza
! Email: brunocbns@gmail.com TEL:(83) 3216-7514
! Universidade Federal Da Paraíba (UFPB)
! Programa De Pós-Graduação Em Engenharia Mecânica (PPGEM)
! Grupo de Pesquisa em Instrumentação em Controle em Estudos de Energia e
! Meio Ambiente (GPICEEMA)
!
!-----!
!                               INICIO DO ALGORITMO TRA                               !
!-----!

USE VARIAVEL
USE PROINVERSO
IMPLICIT NONE
INTEGER :: I, J, K, P, N
DOUBLE PRECISION :: e_LOW, e_HIGH
DOUBLE PRECISION :: AREA_MAX, SOMA1, SOMA2
DOUBLE PRECISION, DIMENSION(:), ALLOCATABLE :: AREA
DOUBLE PRECISION, DIMENSION(:), ALLOCATABLE :: EPSILON
DOUBLE PRECISION, DIMENSION(:, :), ALLOCATABLE :: CAP, S, CAP3
DOUBLE PRECISION, DIMENSION(:), ALLOCATABLE :: CAP2, CAP2_LOW, CAP2_HIGH
DOUBLE PRECISION, DIMENSION(:), ALLOCATABLE :: LAMBDA, LAMBDK
DOUBLE PRECISION, DIMENSION(:), ALLOCATABLE :: CAP2K
DOUBLE PRECISION, DIMENSION(:), ALLOCATABLE :: G
OPEN(UNIT=4, FILE='RESULT_PERMISSIVIDADE.dat')
OPEN(UNIT=10, FILE='RESULT_CAPACITANCIA.dat')
CALL INPUT
N = NCEL*(NCEL-1)/2
e_LOW = 3.0
e_HIGH = 80.0
!-----!
!                               ALOCAÇÃO DINÂMICA DE VARIÁVEIS                               !
!-----!

ALLOCATE(CAP(NCEL,NCEL))
ALLOCATE(EPSILON(NEL))
ALLOCATE(AREA(NEL))
ALLOCATE(CAP2(N))
```

```

ALLOCATE(CAP2_LOW(N))
ALLOCATE(CAP2_HIGH(N))
ALLOCATE(CAP2K(N))
ALLOCATE(LAMBDA(N))
ALLOCATE(LAMBDAK(N))
ALLOCATE(S(N,NIND))
ALLOCATE(G(NIND))

```

```

!-----!
!           CAPACITÂNCIA OBTIDA DA DISTRIUIÇÃO A SER DETERMINADA           !
!-----!

```

```

K = 1
DO I = 1, NCEL - 1
  DO J = I + 1, NCEL
    CAP2(K) = CAP1(I,J)
    K = K + 1
  END DO
END DO

```

```

!-----!
!           CAPACITÂNCIA COM O SENSOR PREENCHIDO COM ELEMENTO DE MENOR PERMISSIVIDADE           !
!-----!

```

```

EPSILON = EPS
DO I = 1, NIND
  K = IND(I)
  EPSILON(K) = e_LOW
END DO
CALL PRODIRETO(EPSILON,CAP,AREA_MAX,AREA)
K = 1
DO I = 1, NCEL - 1
  DO J = I + 1, NCEL
    CAP2_LOW(K) = CAP(I,J)
    K = K + 1
  END DO
END DO

```

```

!-----!
!           CAPACITÂNCIA COM O SENSOR PREENCHIDO COM ELEMENTO DE MAIOR PERMISSIVIDADE           !
!-----!

```

```

EPSILON = EPS
DO I = 1, NIND
  K = IND(I)
  EPSILON(K) = e_HIGH
END DO
CALL PRODIRETO(EPSILON,CAP,AREA_MAX,AREA)
K = 1
DO I = 1, NCEL - 1
  DO J = I + 1, NCEL
    CAP2_HIGH(K) = CAP(I,J)
    K = K + 1
  END DO
END DO

```

```

!-----!
!           CÁLCULO DO VETOR DE CAPACITÂNCIA NORMALIZADO           !
!-----!

```

```

DO I = 1, N
  LAMBDA(I) = (CAP2(I) - CAP2_LOW(I)) / (CAP2_HIGH(I) - CAP2_LOW(I))
END DO

```

```

!           CÁLCULO PARA MONTAGEM DA MATRIZ DE SENSIBILIDADE           !
!-----!
CALL MATRIZ_SENSIBILIDADE(N,e_LOW,e_HIGH,CAP2_LOW,CAP2_HIGH,S)
!-----!
!           DETERMINAÇÃO DO VETOR DE PERMISSIVIDADE           !
!-----!
DO I = 1, NIND
  SOMA1 = 0
  DO J = 1, N
    SOMA1 = SOMA1 + S(J,I)
  END DO
  SOMA2 = 0
  DO J = 1, N
    SOMA2 = SOMA2 + S(J,I)*LAMBDA(J)/SOMA1
  END DO
  G(I) = SOMA2
END DO
EPSILON = EPS
DO I = 1, NIND
  K = IND(I)
  EPSILON(K) = (e_HIGH-e_LOW)*G(I)+e_LOW
END DO
CALL PRODIRETO(EPSILON,CAP,AREA_MAX,AREA)
K = 1
DO I = 1, NCEL - 1
  DO J = I + 1, NCEL
    CAP2K(K) = CAP(I,J)
    K = K + 1
  END DO
END DO
DO I = 1, N
  LAMBDK(I)=(CAP2K(I)-CAP2_LOW(I))/(CAP2_HIGH(I)-CAP2_LOW(I))
END DO
DO I = 1, NIND
  SOMA1 = 0
  DO J = 1, N
    SOMA1 = SOMA1 + S(J,I)
  END DO
  SOMA2 = 0
  DO J = 1, N
    SOMA2 = SOMA2 + S(J,I)*(LAMBDA(J)-LAMBDK(J))/SOMA1
  END DO
  G(I) = G(I) + SOMA2
END DO
DO I = 1, NIND
  IF (G(I) < 0.0) THEN
    G(I) = 0.0
  END IF
  IF (G(I) > 1.0) THEN
    G(I) = 1.0
  END IF
END DO
DO I = 1, NIND
  G(I)=(e_HIGH-e_LOW)*G(I)+e_LOW;
END DO
!-----!

```

! SAIDA DE DADOS !

```
WRITE(4,*) 'MELHOR ESTIMATIVA = ', G
WRITE(10,'(8F12.4)') ((CAP1(I,J),J=1,NCEL),I=1,NCEL)
WRITE(10,*)'-----'
ALLOCATE(CAP3(NCEL,NCEL))
EPSILON = EPS
EPSILON(IND) = G
CALL PRODIRETO(EPSILON,CAP3,AREA_MAX,AREA)
WRITE(10,'(8F12.4)') ((CAP3(I,J),J=1,NCEL),I=1,NCEL)
CLOSE(UNIT=4)
CLOSE(UNIT=10)
STOP
END PROGRAM TRA
```

APÊNDICE E

ROTINA TAIL

PROGRAM TAIL

!

! Este Programa se Propõe a resolver o Problema inverso da Técnica de
! Tomografia Capacitiva Elétrica Utilizando o Método de Retroprojeção Linear

!

! Programa Desenvolvido por Bruno Cesar B. N. De Souza
! Email: brunocbns@gmail.com TEL:(83) 3216-7514
! Universidade Federal Da Paraíba (UFPB)
! Programa De Pós-Graduação Em Engenharia Mecânica (PPGEM)
! Grupo de Pesquisa em Instrumentação em Controle em Estudos de Energia e
! Meio Ambiente (GPICEEMA)

!

!-----!
! INICIO DO ALGORITMO TAIL !
!-----!

USE VARIAVEL

USE PROINVERSO

IMPLICIT NONE

INTEGER :: I, J, K, N, ITER, MAX_ITER

DOUBLE PRECISION :: e_LOW, e_HIGH

DOUBLE PRECISION :: AREA_MAX, SOMA1, SOMA2, NORM, RNORM, ALPHA

DOUBLE PRECISION, DIMENSION(:), **ALLOCATABLE** :: AREA

DOUBLE PRECISION, DIMENSION(:), **ALLOCATABLE** :: EPSILON

DOUBLE PRECISION, DIMENSION(:,:), **ALLOCATABLE** :: CAP, S, CAP3

DOUBLE PRECISION, DIMENSION(:), **ALLOCATABLE** :: CAP2, CAP2_LOW, CAP2_HIGH

DOUBLE PRECISION, DIMENSION(:), **ALLOCATABLE** :: LAMBDA, LAMBDK

DOUBLE PRECISION, DIMENSION(:), **ALLOCATABLE** :: G, X, R, Q

OPEN(UNIT=4, FILE='RESULT_PERMISSIVIDADE.dat')

OPEN(UNIT=10, FILE='RESULT_CAPACITANCIA.dat')

CALL INPUT

N = NCEL*(NCEL-1)/2

e_LOW = 3.0

e_HIGH = 80.0

!-----!
! ALOCAÇÃO DINÂMICA DE VARIÁVEIS !
!-----!

ALLOCATE(CAP(NCEL,NCEL))

ALLOCATE(EPSILON(NEL))

ALLOCATE(AREA(NEL))

ALLOCATE(CAP2(N))

ALLOCATE(CAP2_LOW(N))

```

ALLOCATE(CAP2_HIGH(N))
ALLOCATE(LAMBDA(N))
ALLOCATE(LAMBDAK(N))
ALLOCATE(S(N,NIND))
ALLOCATE(G(NIND))
ALLOCATE(X(NIND))
ALLOCATE(R(NIND))
ALLOCATE(Q(N))

```

```

!-----!
!           CAPACITÂNCIA OBTIDA DA DISTRIUIÇÃO A SER DETERMINADA           !
!-----!

```

```

K = 1
DO I = 1, NCEL - 1
    DO J = I + 1, NCEL
        CAP2(K) = CAP1(I,J)
        K = K + 1
    END DO
END DO

```

```

!-----!
!           CAPACITÂNCIA COM O SENSOR PREENCHIDO COM ELEMENTO DE MENOR PERMISSIVIDADE       !
!-----!

```

```

EPSILON = EPS
DO I = 1, NIND
    K = IND(I)
    EPSILON(K) = e_LOW
END DO
CALL PRODIRETO(EPSILON,CAP,AREA_MAX,AREA)
K = 1
DO I = 1, NCEL - 1
    DO J = I + 1, NCEL
        CAP2_LOW(K) = CAP(I,J)
        K = K + 1
    END DO
END DO

```

```

!-----!
!           CAPACITÂNCIA COM O SENSOR PREENCHIDO COM ELEMENTO DE MAIOR PERMISSIVIDADE       !
!-----!

```

```

EPSILON = EPS
DO I = 1, NIND
    K = IND(I)
    EPSILON(K) = e_HIGH
END DO
CALL PRODIRETO(EPSILON,CAP,AREA_MAX,AREA)
K = 1
DO I = 1, NCEL - 1
    DO J = I + 1, NCEL
        CAP2_HIGH(K) = CAP(I,J)
        K = K + 1
    END DO
END DO

```

```

!-----!
!           CÁLCULO DO VETOR DE CAPACITÂNCIA NORMALIZADO           !
!-----!

```

```

DO I = 1, N
    LAMBDA(I) = (CAP2(I) - CAP2_LOW(I)) / (CAP2_HIGH(I) - CAP2_LOW(I))
END DO

```

```

!-----!
!           CÁLCULO PARA MONTAGEM DA MATRIZ DE SENSIBILIDADE           !
!-----!
CALL MATRIZ_SENSIBILIDADE(N,e_LOW,e_HIGH,CAP2_LOW,CAP2_HIGH,S)
!-----!
!           DETERMINAÇÃO DO VETOR DE PERMISSIVIDADE           !
!-----!
DO I = 1, NIND
  SOMA1 = 0
  DO J = 1, N
    SOMA1 = SOMA1 + S(J,I)
  END DO
  SOMA2 = 0
  DO J = 1, N
    SOMA2 = SOMA2 + S(J,I)*LAMBDA(J)/SOMA1
  END DO
  G(I) = SOMA2
END DO
MAX_ITER = 20
ITER = 1
RNORM = 1E-2
NORM = 1E+5
DO WHILE ((NORM >= RNORM) .AND. (ITER <= MAX_ITER))
  X = G
  SOMA2 = 0
  DO I = 1, N
    SOMA1 = 0
    DO J = 1, NIND
      SOMA1 = SOMA1 + S(I,J)
    END DO
    SOMA2 = 0
    DO J = 1, NIND
      SOMA2 = SOMA2 + S(I,J)*X(J)/SOMA1
    END DO
    LAMBDAK(I) = SOMA2
  END DO
  DO I = 1, NIND
    SOMA1 = 0
    DO J = 1, N
      SOMA1 = SOMA1 + S(J,I)*(LAMBDAK(J)-LAMBDA(J))
    END DO
    R(I) = SOMA1
  END DO
  DO I = 1, N
    SOMA1 = 0
    DO J = 1, NIND
      SOMA1 = SOMA1 + S(I,J)*R(J)
    END DO
    Q(I) = SOMA1
  END DO
  SOMA1 = 0
  DO I = 1, NIND
    SOMA1 = SOMA1 + R(I)*R(I)
  END DO
  SOMA2 = 0
  DO I = 1, N

```

```

        SOMA2 = SOMA2 + Q(I)*Q(I)
    END DO
    ALPHA = SOMA1/SOMA2
    X = X - ALPHA*R
    SOMA1 = 0
    DO I = 1, NIND
        SOMA1 = SOMA1 + (X(I)-G(I))**2
    END DO
    SOMA2 = 0
    DO I = 1, NIND
        SOMA2 = SOMA2 + (X(I))**2
    END DO
    NORM = SQRT(SOMA1)/SQRT(SOMA2)
    G = X;
    ITER = ITER + 1;
END DO
DO I = 1, NIND
    IF (G(I) < 0.0) THEN
        G(I) = 0.0
    END IF
    IF (G(I) > 1.0) THEN
        G(I) = 1.0
    END IF
END DO
DO I = 1, NIND
    G(I)=(e_HIGH-e_LOW)*G(I)+e_LOW;
END DO
!-----!
!                               SAIDA DE DADOS                               !
!-----!
WRITE(4,*) 'MELHOR ESTIMATIVA = ', G
WRITE(10,'(8F12.4)') ((CAP1(I,J),J=1,NCEL),I=1,NCEL)
WRITE(10,*)'-----'
ALLOCATE(CAP3(NCEL,NCEL))
EPSILON = EPS
EPSILON(IND) = G
CALL PRODIRETO(EPSILON,CAP3,AREA_MAX,AREA)
WRITE(10,'(8F12.4)') ((CAP3(I,J),J=1,NCEL),I=1,NCEL)
CLOSE(UNIT=4)
CLOSE(UNIT=10)
STOP
END PROGRAM TAIL

```

APÊNDICE F

ROTINAS ASSOCIADAS A RESOLUÇÃO DO PROBLEMA DIRETO

```
SUBROUTINE PRODIRETO(NCEL,NNP,NEL,NNL,XY,NL,NLCODE,EPSILON,CONTORNO,VAL,K,W,V,Q,CAP)
!
! Esta rotina se propõe a resolver o problema direto da TCE
!
! Programa Desenvolvido por Bruno César B. N. De Souza
! Email: brunocbns@gmail.com
! Universidade Federal Da Paraíba (UFPB)
! Programa De Pós-Graduação Em Engenharia Mecânica (PPGEM)
! Grupo de Pesquisa em Instrumentação e Controle em Estudo de Energia e Meio Ambiente (GPICEEMA)
!
IMPLICIT NONE
INTEGER :: NCEL, NNP, NEL, NNL, NNZ
INTEGER :: K, W
INTEGER, DIMENSION(NEL,3) :: NL
INTEGER, DIMENSION(NNP) :: NLCODE
INTEGER, DIMENSION(NNL) :: CONTORNO
INTEGER, DIMENSION(NNP+1) :: IC, IWK
INTEGER, DIMENSION(:), ALLOCATABLE :: JC
DOUBLE PRECISION :: AREA_MAX
DOUBLE PRECISION, DIMENSION(NEL) :: EPSILON
DOUBLE PRECISION, DIMENSION(NNP,2) :: XY
DOUBLE PRECISION, DIMENSION(NNP,NCEL) :: VAL
DOUBLE PRECISION, DIMENSION(NEL) :: AREA
DOUBLE PRECISION, DIMENSION(:), ALLOCATABLE :: C
DOUBLE PRECISION, DIMENSION(NNP,NCEL) :: V
DOUBLE PRECISION, DIMENSION(NNP-NNL,NCEL) :: Q
DOUBLE PRECISION, DIMENSION(NCEL,NCEL) :: CAP
  CALL MONTAGEM_IC(NNP,NEL,NL,NLCODE,IWK,IC,NNZ) ! MONTAGEM DO VETOR IC
  ALLOCATE(JC(NNZ))
  ALLOCATE(C(NNZ))
  CALL MONTAGEM_C(NNP,NEL,XY,NL,EPSILON,NNZ,IWK,IC,JC,C,AREA,AREA_MAX) ! MONTAGEM DA MATRIZ
  CALL SISLINEAR(NNL,NCEL,NNP,NNP+1,NNZ,IC,JC,C,CONTORNO,VAL,V,Q) ! CALCULO DO POTENCIAL
  CALL CAPACITANCIA(NCEL,K,W,Q,CAP) ! CALCULO DA CAPACITANCIA
  CAP=-0.182*CAP
RETURN
END SUBROUTINE PRODIRETO

SUBROUTINE MONTAGEM_IC(NNP,NEL,NL,NLCODE,IWK,IC,NNZ)
!
! Esta subrotina determina o valor de nnz (quantidade total de elementos não nulos na matriz de capacitância)
```

! e constrói o vetor IC (vetor que contém a informação da quantidade de elementos não nulos em uma dada
! linha da matriz de capacitância).

```
!
IMPLICIT NONE
INTEGER :: NNP, NEL, NNZ
INTEGER :: I, J
INTEGER :: NOD, KSAV, KSAVN
INTEGER, DIMENSION(NNP) :: NLCODE
INTEGER, DIMENSION(NNP+1) :: IC, IWK
INTEGER, DIMENSION(NEL,3) :: NL
DO I = 1, NNP+1
    IC(I) = 1
END DO
DO I = 1, NEL
    DO J = 1, 3
        NOD = NL(I,J)
        IC(NOD) = IC(NOD) + 1
    END DO
END DO
DO I = 1, NNP
    IF (NLCODE(I)>=1) IC(I)=IC(I)+1
END DO
KSAV = IC(1)
IC(1) = 1
DO I = 2, NNP+1
    KSAVN = IC(I)
    IC(I) = IC(I-1)+ KSAV
    IWK(I-1) = IC(I-1)-1
    KSAV = KSAVN;
END DO
NNZ=IC(NNP+1)-1
RETURN
END SUBROUTINE MONTAGEM_IC
```

```
SUBROUTINE MONTAGEM_C(NNP,NEL,XY,NL,EPSILON,NNZ,IWK,IC,JC,C,AREA,AREA_MAX)
```

!
! Esta subrotina determina o vetor JC (vetor que contém a informação da posição dos elementos não
! nulos em uma dada linha da matriz de capacitância) e o vetor C (vetor que contém os valores não
! nulos da matriz de capacitância)

```
!
IMPLICIT NONE
INTEGER :: NNP, NEL, NNZ
INTEGER :: I, J, K, M, N, KA, KB
INTEGER :: II, JJ, IROWST, ILAST, W
INTEGER, DIMENSION(NNP+1) :: IC, IWK
INTEGER, DIMENSION(NNZ) :: JC, JWK
INTEGER, DIMENSION(NEL,3) :: NL
DOUBLE PRECISION :: EO, AREA_MAX
DOUBLE PRECISION, DIMENSION(3) :: XL, YL, P, Q
DOUBLE PRECISION, DIMENSION(3,3) :: CE
DOUBLE PRECISION, DIMENSION(NEL) :: EPSILON
DOUBLE PRECISION, DIMENSION(NEL) :: AREA
DOUBLE PRECISION, DIMENSION(NNZ) :: C
DOUBLE PRECISION, DIMENSION(NNP,2) :: XY
DO I = 1, NNZ
```

```

        JWK(I)=0
        JC(I)=0
        C(I)=0
    END DO
    E0=8.854
    AREA_MAX=0
    DO I = 1, NEL
        DO J=1, 3
            K=NL(I,J)
            XL(J)=XY(K,1)
            YL(J)=XY(K,2)
        END DO
        P(1)=YL(2)-YL(3)
        P(2)=YL(3)-YL(1)
        P(3)=YL(1)-YL(2)
        Q(1)=XL(3)-XL(2)
        Q(2)=XL(1)-XL(3)
        Q(3)=XL(2)-XL(1)
        AREA(I)=0.5*abs(P(2)*Q(3)-Q(2)*P(3))
        IF (AREA_MAX<AREA(I)) AREA_MAX=AREA(I)
    !-----!
    !   DETERMINAÇÃO DA MATRIZ DE CAPACITÂNCIA LOCAL   !
    !-----!
        DO M = 1, 3
            DO N = 1, 3
                CE(M,N)=E0*EPSILON(I)*(P(M)*P(N)+Q(M)*Q(N))/(4.0*AREA(I)) !
            END DO
        END DO
    !-----!
    !   PROCEDIMENTO DE MONTAGEM DOS VETORES JC E C A PARTIR DA MATRIZ LOCAL   !
    !-----!
        DO KA = 1, 3
            II = NL(I,KA)
            IROWST = IC(II)
            ILAST = IWK(II)
            DO K = IROWST, ILAST
                W = JC(K)
                JWK(W) = K
            END DO
            DO KB = 1, 3
                JJ = NL(I,KB)
                IF (JWK(JJ)== 0) THEN
                    ILAST = ILAST+1
                    JWK(JJ) = ILAST
                    JC(ILAST) = JJ
                    C(ILAST) = CE(KA,KB)
                ELSE
                    C(JWK(JJ)) = C(JWK(JJ)) + CE(KA,KB) !MATRIZ GLOBAL
                END IF
            END DO
            DO K = IROWST, ILAST
                JWK(JC(K)) = 0
            END DO
            IWK(II) = ILAST
        END DO
    END DO

```

```

RETURN
END SUBROUTINE MONTAGEM_C

```

```

SUBROUTINE SISLINEAR(NNL,NCEL,NNP,NI,NJ,IC,JC,C,CONTORNO,VAL,V,Q)

```

```

!
! Esta subrotina prepara a resolução do sistema linear, acessando os elementos da matriz de capacitância
! relacionados com os nodos livres da malha, e resolvendo-os em função dos nodos prescritos, logo após,
! é calculada a carga elétrica nos eletrodos seguindo o desenvolvimento proposto por Spink et al (1998)
!

```

```

IMPLICIT NONE

```

```

INTEGER :: I, J, II

```

```

INTEGER :: NNL, NCEL, NNP

```

```

INTEGER :: NI, NI1, NI2, NI3, NI4

```

```

INTEGER :: NJ, NJ1, NJ2, NJ3, NJ4

```

```

INTEGER :: MAX_ITER

```

```

INTEGER, DIMENSION(NNL) :: CONTORNO

```

```

INTEGER, DIMENSION(NI) :: IC, IC1, IC2, IC3, IC4

```

```

INTEGER, DIMENSION(NJ) :: JC, JC1, JC2, JC3, JC4

```

```

DOUBLE PRECISION :: TOL, RNORM

```

```

DOUBLE PRECISION, DIMENSION(NJ) :: C, C1, C2, C3, C4

```

```

DOUBLE PRECISION, DIMENSION(NNP,NCEL) :: VAL

```

```

DOUBLE PRECISION, DIMENSION(NNL,NCEL) :: B, VF

```

```

DOUBLE PRECISION, DIMENSION(NNL) :: X

```

```

DOUBLE PRECISION, DIMENSION(NNP-NNL,NCEL) :: VB, Q1, Q2, Q

```

```

DOUBLE PRECISION, DIMENSION(NNP,NCEL) :: V

```

```

!-----!
!                APLICAÇÃO DOS VALORES PRESCRITOS DE POTENCIAL                !
!-----!

```

```

DO I= 1, NNP-NNL

```

```

    VB(I,:)=VAL(CONTORNO(I,:))

```

```

END DO

```

```

CALL CORTE_MATRIZ(1,NNL,1,NNL,IC,JC,C,NI,NJ,IC1,JC1,C1,NI1,NJ1)

```

```

CALL CORTE_MATRIZ(1,NNL,NNL+1,NNP,IC,JC,C,NI,NJ,IC2,JC2,C2,NI2,NJ2)

```

```

DO I=1, NNL

```

```

    B(I,:)=0.0

```

```

END DO

```

```

DO J=1, NNL

```

```

    DO I=IC2(J), IC2(J+1)-1

```

```

        II=JC2(I);

```

```

        B(J,:)=B(J,:)-C2(I)*VB(II,:)

```

```

    END DO

```

```

END DO

```

```

!-----!
!-----                CÁLCULO DO CAMPO PRIMÁRIO (POTENCIAL)                -----!
!-----!

```

```

DO I=1, NNL

```

```

    VF(I,:)=5.0

```

```

END DO

```

```

MAX_ITER = 500

```

```

TOL = 0.00001

```

```

DO I=1, NCEL

```

```

    X(:)=VF(:,I)

```

```

    CALL SERIAL_CONJ_GRAD(NI1,NJ1,IC1,JC1,C1,B(:,I),X,MAX_ITER,TOL,RNORM)

```

```

    VF(:,I)=X(:)

```

```

END DO

```

```

DO I=1, NNL
    V(I,:)=VF(I,:)
END DO
DO I=NNL+1, NNP
    V(I,:)=VB(I-NNL,:)
END DO
CALL CORTE_MATRIZ(NNL+1,NNP,1,NNL,IC,JC,C,NI,NJ,IC3,JC3,C3,NI3,NJ3)
CALL CORTE_MATRIZ(NNL+1,NNP,NNL+1,NNP,IC,JC,C,NI,NJ,IC4,JC4,C4,NI4,NJ4)
DO I=1, NNP-NNL
    Q(I,:)=0.0
END DO
DO I=1, NNP-NNL
    Q1(I,:)=0.0
END DO
DO I=1, NNP-NNL
    Q2(I,:)=0.0
END DO
!-----!
!-----  --- CÁLCULO DO CAMPO SECUNDÁRIO (CARGAS ELÉTRICAS)  -----!
!-----!
DO J=1, NNP-NNL
    DO I=IC3(J), IC3(J+1)-1
        II=JC3(I)
        Q1(I,:)=Q1(J,:)+C3(I)*VF(II,:)
    END DO
END DO
DO J=1, NNP-NNL
    DO I=IC4(J), IC4(J+1)-1
        II=JC4(I)
        Q2(I,:)=Q2(J,:)+C4(I)*VB(II,:)
    END DO
END DO
Q=Q1+Q2
RETURN
END SUBROUTINE SISLINEAR

SUBROUTINE CORTE_MATRIZ(I1,I2,J1,J2,IC,JC,C,NI,NJ,IC1,JC1,C1,NI1,NJ1)
!
! Esta subrotina acessa as posições de uma matriz esparsa e a separa
!
INTEGER :: I, I1, I2, J, J1, J2, NI, NJ, NI1, NJ1
INTEGER :: K, W, T
INTEGER, DIMENSION(NI) :: IC, IC1
INTEGER, DIMENSION(NJ) :: JC, JC1
DOUBLE PRECISION, DIMENSION(NJ) :: C, C1
K=0
W=1
IC1(1)=1
DO I = I1,I2
    W=W+1
    T=0
    DO J=IC(I),IC(I+1)-1
        IF ((JC(J)>=J1).AND.(JC(J)<=J2)) THEN
            K=K+1
            T=T+1
        END IF
    END DO
    IC1(W)=I
    JC1(K)=J
    C1(T)=C(J)
END DO

```

```

                JC1(K)=JC(J)-J1+1
                C1(K)=C(J)
            END IF
        END DO
        IC1(W)=IC1(W-1)+T
    END DO
    NI1=W
    NJ1=K
    RETURN
END SUBROUTINE CORTE_MATRIZ

```

```

SUBROUTINE SERIAL_CONJ_GRAD(NIA, NJA, IA, JA, A, B, X, MAX_ITER, TOL, RNORM)

```

```

!
! Esta subrotina soluciona o sistema linear utilizando o método do gradiente conjugado
!

```

```

IMPLICIT NONE
INTEGER :: I, J
INTEGER :: ITER, MAX_ITER
INTEGER :: NIA, NJA
INTEGER, DIMENSION(NIA) :: IA
INTEGER, DIMENSION(NJA) :: JA
DOUBLE PRECISION :: TOL, D, RHO, RHO0, ALPHA, BETA, SUM, SUM1, RNORM
DOUBLE PRECISION, DIMENSION(NJA) :: A
DOUBLE PRECISION, DIMENSION(NIA-1) :: B, R, Z, P, Q, X!, M
DO J = 1, NIA-1
    SUM1=0.0
    DO I = IA(J), IA(J+1)-1
        SUM1 = SUM1 + A(I)*X(JA(I))
    END DO
    R(J)=B(J)-SUM1
END DO

```

```

!-----!
!                               INÍCIO DO LOOP
!-----!

```

```

ITER=1
RNORM=1000
DO WHILE ((ITER <= MAX_ITER).and.(RNORM>=TOL))
    DO J = 1, NIA-1
        Z(J) = R(J) !/M(J)
    END DO
    RHO=0.0
    DO J = 1, NIA-1
        RHO = RHO + R(J)*Z(J)
    END DO
    IF (ITER == 1) THEN
        DO J = 1, NIA-1
            P(J) = Z(J)
        END DO
    ELSE
        BETA = RHO/RHO0
        DO J = 1, NIA-1
            P(J) = Z(J) + BETA*P(J)
        END DO
    END IF
END WHILE

```

```

!-----!

```

```

! q = A.p
!-----!
      DO J = 1, NIA-1
          SUM1 = 0.0
      DO I = IA(J), IA(J+1) - 1
          SUM1 = SUM1 + A(I)*P(JA(I))
      END DO
      Q(J)=SUM1
    END DO
!-----!
! p.q
!-----!
      D = 0.0
      DO J = 1, NIA-1
          D = D + P(J)*Q(J)
      END DO
!-----!
! ALPHA = RHO / (p.q)
!-----!
      ALPHA = RHO/D
!-----!
! z = z + ALPHA*p
! r = r - ALPHA*q
!-----!
      DO J = 1, NIA-1
          X(J) = X(J) + ALPHA*P(J)
          R(J) = R(J) - ALPHA*Q(J)
      END DO
!-----!
! ||r|| = ||b - A.x||
!-----!
      SUM = 0.0
      DO J = 1, NIA-1
          SUM = SUM + R(J)*R(J)
      END DO
      RNORM = SQRT(SUM)
      RHO0 = RHO
      ITER = ITER + 1
    END DO
    RETURN
  END SUBROUTINE SERIAL_CONJ_GRAD

SUBROUTINE CAPACITANCIA(NCEL,K,W,Q,CAP)
!
! Esta subrotina soma a quantidade de cargas elétricas depositadas em cada eletrodo e divide pelo
! respectivo potencial elétrico.
!
  IMPLICIT NONE
  INTEGER :: NCEL, K, W, I, J
  DOUBLE PRECISION, DIMENSION(NCEL,NCEL) :: CAP
  DOUBLE PRECISION, DIMENSION(W,NCEL) :: Q

  DO J = 1, NCEL
      DO I = 1, NCEL
          CAP(I,J)=0.0
      END DO
  END DO

```

```

        END DO
    END DO
    DO J = 1, NCEL
        DO I = 1, NCEL
            CAP(I,J)=sum(Q(2+(I-1)*K:I*K,J),1)/10
        END DO
    END DO
    RETURN
END SUBROUTINE CAPACITANCIA

```

```

SUBROUTINE ENERGIA(N,X,TIPO,F)

```

```

!
```

```

! Esta subrotina determina uma medida de similaridade entre os valores da resposta do sensor
! com a resposta do sistema simulado.

```

```

!
```

```

USE VARIAVEL

```

```

USE PROINVERSO

```

```

IMPLICIT NONE

```

```

INTEGER :: N, I, J, TIPO

```

```

DOUBLE PRECISION :: F, SIGMA, SIGMA_MIN, SIGMA_MAX

```

```

DOUBLE PRECISION, DIMENSION(N) :: X

```

```

DOUBLE PRECISION, DIMENSION(NEL) :: EPSILON

```

```

DOUBLE PRECISION, DIMENSION(NCEL,NCEL) :: CAP2

```

```

EPSILON = EPS

```

```

EPSILON(IND)=X

```

```

CALL PRODIRETO(EPSILON,CAP2)

```

```

SELECT CASE (TIPO)

```

```

    CASE (0) ! Norma L0

```

```

        SIGMA_MAX=1000

```

```

        SIGMA_MIN=0.001

```

```

        SIGMA=SIGMA_MAX-(SIGMA_MAX-SIGMA_MIN)*1/1000

```

```

        F = 0.0

```

```

        DO I = 1, NCEL - 1

```

```

            DO J = I + 1, NCEL

```

```

                F = F + 1 - DEXP(-(CAP2(I,J)-CAP1(I,J))**2/SIGMA**2)

```

```

            END DO

```

```

        END DO

```

```

    CASE (1) ! Norma L1

```

```

        F = 0.0

```

```

        DO I = 1, NCEL - 1

```

```

            DO J = I + 1, NCEL

```

```

                F = F + ABS((CAP2(I,J)-CAP1(I,J))/CAP1(I,J))

```

```

            END DO

```

```

        END DO

```

```

    CASE (2) ! Norma L2

```

```

        F = 0.0

```

```

        DO I = 1, NCEL - 1

```

```

            DO J = I + 1, NCEL

```

```

                F = F + (100*(CAP2(I,J)-CAP1(I,J))/CAP1(I,J))**2

```

```

            END DO

```

```

        END DO

```

```

        F=SQRT(F)

```

```

    CASE (3) ! Norma infinita

```

```

        F = 0.0

```

```

        DO I = 1, NCEL - 1

```

```

        DO J = I + 1, NCEL
            IF (F < (ABS(CAP2(I,J)-CAP1(I,J)))) THEN
                F = ABS(CAP2(I,J)-CAP1(I,J))
            END IF
        END DO
    END DO
CASE DEFAULT
    PRINT *, 'Opção Invalida!'
END SELECT
END SUBROUTINE ENERGIA

```

```

SUBROUTINE INIT_RANDOM_SEED()

```

```

!
```

```

! Esta subrotina reinicia a semente da função de geração de números pseudo-aleatórios

```

```

!
```

```

INTEGER :: I, N, CLOCK
INTEGER, DIMENSION(:), ALLOCATABLE :: SEED
CALL RANDOM_SEED(SIZE=N)
ALLOCATE(SEED(N))
CALL SYSTEM_CLOCK(COUNT=CLOCK)
SEED = CLOCK + 37 * (/ (I - 1, I = 1, N) /)
CALL RANDOM_SEED(PUT=SEED)
DEALLOCATE(SEED)
END SUBROUTINE

```

ANEXO G

ROTINA DE TESTE DE COMUNICAÇÃO

```
PROGRAM alo_mundo
!  
! Esta rotina testa a comunicação de rede entre os computadores contidos no cluster  
!  
USE mpi  
INTEGER :: SIZE, RANK, LEN, IERR  
CHARACTER(MPI_MAX_PROCESSOR_NAME) :: HOSTNAME  
CALL MPI_INIT(IERR)  
CALL MPI_COMM_SIZE(MPI_COMM_WORLD, SIZE, IERR)  
CALL MPI_COMM_RANK(MPI_COMM_WORLD, RANK, IERR)  
CALL MPI_GET_PROCESSOR_NAME(HOSTNAME, LEN, IERR)  
PRINT*, 'tarefa = ', SIZE, ' meu rank = ', RANK, ' versao = ', HOSTNAME  
CALL MPI_FINALIZE(IERR)  
END PROGRAM alo_mundo
```

ANEXO A

ALGORITMOS PROJETIVOS

Considerando que o valor de capacitância descrito pela Equação 2.3, é determinado como uma função não linear da distribuição de permissividade elétrica no interior da tubulação, esta não pode ser explícita de maneira simples.

Entretanto a relação causal entre a distribuição de permissividade (ε e os valores de capacitância C entre os sensores é descrita por um funcional não linear ξ , sendo:

$$C = \xi(\varepsilon) \quad (A.1)$$

Reescrevendo a Equação (A.1) em termos da sua série de Taylor, é possível determinar a variação na capacitância ΔC em resposta a uma perturbação da distribuição de permissividade ($\Delta\varepsilon$), onde será dada por:

$$\Delta C = \frac{d\xi}{dx}(\Delta\varepsilon) + O((\Delta\varepsilon)^2) \quad (A.2)$$

onde $d\xi/dx$ pode ser compreendida como uma relação de sensibilidade do sensor e $O((\Delta\varepsilon)^2)$ o erro de ordem quadrática. No entanto, se a diferença entre as constantes dielétricas dos materiais constituintes for pequena, é possível realizar uma aproximação linear simples da Equação (A.2), negligenciando os termos de mais altas ordens. Assim obtemos:

$$\Delta C = d\xi/dx \Delta\varepsilon \quad (A.3)$$

A Equação (A.3) necessita ser discretizada para ser implementada, para tal, a área interna do sensor deve ser fragmentada em N pequenas áreas (ou píxeis). Considerando um sensor com $M = n(n-1)/2$ leituras independentes de capacitância (par eletrodo fonte-

receptor) em um sensor com n eletrodos, obtém-se M equações na forma da Equação A.3. A forma linearizada e discretizada da equação (A.3) pode ser escrita como:

$$\underbrace{\Delta C}_{M \times 1} = \underbrace{S}_{M \times N} \underbrace{\Delta \varepsilon}_{N \times 1} \quad (A.4)$$

Onde S é a matriz jacobiana, isto é, a matriz das derivadas, que indica a influência da distribuição geométrica dos dielétricos no interior do domínio, contribuindo para a resposta do sensor.

Comumente a Equação A.4 pode ser apresentada na forma

$$\lambda = Sg \quad (A.5)$$

onde λ é um vetor de capacitância normalizada e g é o vetor de permissividade normalizada, isto é, o vetor utilizado para representar a imagem obtida no sensor.

Sendo i o índice relacionado ao eletrodo fonte, j o índice relacionado ao eletrodo detector, a matriz de capacitância normalizada λ_{ij} é calculada a partir de

$$C_{ij} = \frac{C_{ij} - C_{ij}^{menor}}{C_{ij}^{maior} - C_{ij}^{menor}} \quad (A.6)$$

onde $C_{i,j}$ é o valor da capacitância entre o eletrodo $i - j$ obtida pelo sistema sensor, C_{ij}^{maior} e C_{ij}^{menor} são respectivamente os valores de capacitância quando a tubulação se encontra preenchida com o material de maior e menor permissividade relativa.

A matriz de sensibilidade S_{ij}^p associada ao pixel p para o par de eletrodos $i - j$ é dada por

$$S_{ij}^p = \frac{1}{\varepsilon^{maior} - \varepsilon^{menor}} \frac{C_{ij}(p) - C_{ij}^{menor}}{C_{ij}^{maior} - C_{ij}^{menor}} \frac{A_p}{\max(A_p)} \quad (A.7)$$

onde ε^{maior} e ε^{menor} representam os valores de permissividade relativa para o material de maior permissividade relativa e o de menor permissividade relativa respectivamente, $C_{i,j}(p)$ é a capacitância entre o par de eletrodo $i - j$ considerando que este é o único pixel de maior permissividade em toda a seção da tubulação sendo todos os demais preenchidos com o menor valor de permissividade, A_p é a área do pixel em questão e $\max(A_p)$ é a área do maior pixel da discretização.

O procedimento numérico para construção da matriz de sensibilidade S_{ij}^p está descrito em Azevedo Filho (2002).

Um elemento de S_{ij}^p pode ser interpretado como um elemento ponderador em uma média no qual pixels com maior sensibilidade contribuem em maior proporção para o resultado de capacitância no sensor. A matriz de sensibilidade S_{ij}^p pode ser descrita como um mapa da sensibilidade da distribuição espacial da permissividade elétrica no interior do sensor.

01 – Tomografia por Retroprojeção Linear (TRL)

Se a inversa de S existisse, a Equação A.5 poderia ser resolvida diretamente por

$$g = S^{-1}\lambda \quad (A.8)$$

No entanto, a inversa de S não existe e outros métodos para a solução do problema devem ser utilizados. O algoritmo de TRL, apresentado em Xie et al. (1992), considera a matriz S como um mapeamento linear desde o espaço vetorial das permissividades relativas normalizadas para o espaço vetorial das capacitâncias normalizadas. Desta forma, a transposta de S pode ser considerada como um mapeamento linear do espaço vetorial das capacitâncias normalizadas para o espaço vetorial das permissividades relativas normalizadas. Assim, a solução aproximada para a distribuição de permissividade relativa normalizada, \hat{g} pode ser dada por

$$\hat{g} = S^T \lambda \quad (A.9)$$

Uma forma normalizada da Equação A.8 pode ser expressa como

$$\hat{g} = \frac{S^T \lambda}{S^T u_\lambda} \quad (A.10)$$

onde u_λ é um vetor identidade.

Azevedo Filho (2002) discute a respeito dos problemas causados pela não linearidade da Equação 2.3 e sua influência nos valores de \hat{g} apresentados pela Equação (A.10) implicando na ocorrência de valores acima de 1 (over-shooting) e valores abaixo de 0 (under-shooting). Para melhorar a convergência do método, ele apresenta uma operação de filtragem projetada, que é dada por

$$P[\hat{g}] = \begin{cases} 0 & \text{se } \hat{g} < 0 \\ \hat{g} & \text{se } 0 < \hat{g} < 1 \\ 1 & \text{se } \hat{g} > 1 \end{cases}$$

Entre os algoritmos projetivos, a TRL é o mais simples e mais rápido (YANG 2003). Embora tenha algumas limitações em termos de precisão e resolução espacial, é adequada para análise de processos industriais devido a sua velocidade de processamento.

02 – Tomografia por Reconstrução Algébrica (TRA)

Esse algoritmo é baseado na minimização da diferença entre as capacitâncias normalizadas, calculadas e medidas. Inicialmente, uma imagem é calculada usando-se o algoritmo de TRL, onde essa imagem é melhorada alterando a distribuição do meio através de um processo iterativo segundo a expressão abaixo.

$$\hat{g}^{(k+1)} = \hat{g}^k + \frac{S^T(\lambda - \lambda^k)}{S^T u_\lambda}$$

onde

$$\lambda_{ij}^k = \frac{C_{ij}^k - C_{ij}^{menor}}{C_{ij}^{maior} - C_{ij}^{menor}}$$

sendo C_{ij}^k o valor de capacitância calculada a partir da k -ésima configuração de dielétricos e λ_{ij}^k a sua respectiva matriz capacitância normalizada simulada.

Reinecke e Mewes (1996) mostraram em seu trabalho que a utilização desse algoritmo reduz para aproximadamente 40% o efeito das manchas ocasionadas pela utilização do algoritmo de TRL.

03 – Tomografia por Algoritmo Iterativo de Landweber (TAIL)

Utilizando uma ideia similar ao empregado na TRL, podemos realizar uma aproximação linear para a Equação 2.3 dada por:

$$C = Sg + E \tag{A.12}$$

Onde C é o vetor da capacitância medida, g é o vetor da distribuição de constante dielétrica, S é a matriz da distribuição de sensibilidade dentro dos sensores de capacitância, E representa os erros entre os dados medidos e os dados estimados.

O problema tomográfico é então deslocado para o problema de determinarmos o valor de g que minimiza o erro encontrado entre os valores experimentais e os obtidos pela aproximação linear considerada, este processo pode ser determinado aplicando o princípio de mínimos quadrados. A Equação A. 1.3 representa a função que será minimizada

$$f(g) = \frac{1}{2}(\lambda - Sg)^T(\lambda - Sg) \quad (A. 13)$$

Derivando a Equação (A. 13), obtém-se uma equação linear em relação a g que após resolvida pode oferecer uma aproximação de 1ª ordem para a solução da Equação (A. 1).

$$S^T Sg = S^T \lambda \quad (A. 14)$$

Para obter-se a solução da Equação (A. 14), utiliza-se o algoritmo iterativo do gradiente descendente, sendo:

$$g^{(0)} = S^T \lambda \quad (A. 15 - a)$$

$$g^{(k+1)} = g^k - \alpha_k(S^T Sg^{(k)} - S^T \lambda) \quad (A. 15 - b)$$

onde α_k representa um escalar (fator de ganho) positivo que determina o tamanho do passo na iteração k . O valor de α_k ótimo é determinado pela expressão

$$\alpha_k = \frac{(S^T Sg^k - S^T \lambda)^T (S^T Sg^k - S^T \lambda)}{(S^T Sg^k - S^T \lambda)^T S^T S (S^T Sg^k - S^T \lambda)}$$

Um problema com o algoritmo TAIL é a sua semi-convergência. O erro de imagem diminui rapidamente, mas após alcançar o ponto mínimo, o erro de imagem cresce caso o processo de iteração continue. Para tal, foi implementado duas condições de parada de modo a tornar o processo iterativo eficiente, sendo o primeiro associado ao número máximo de iterações e um segundo focado na estagnação da solução conforme a equação abaixo

$$\frac{\|g^{(k+1)} - g^k\|}{\|g^{(k+1)}\|} < erro \quad (A. 14 - b)$$

A função `MPI_Comm_rank()` fornece no argumento *my_id* a identificação de cada processo dentro de um grupo definido no comunicador *comm*, este identificador deve ter valor entre 0 e n-1, onde n é o número total de processos.

Neste trabalho, o comunicador padrão do MPI, dado pela macro pré-definida `MPI_COMM_WORLD`, foi utilizado. Este comunicador, disponível após a inicialização do MPI, é constituído de todos os processos que participam da computação paralela, os quais são definidos nas opções de execução do comando `mpirun`. A função retorna um número inteiro, com um código de erro associado (SNIR *et al.* [43]).

MPI_Send(msg, count, datatype, dest, tag, comm)

msg	Entrada.	Endereço inicial do " <i>buffer</i> " de envio;
count	Entrada.	Número de " <i>entradas</i> " para enviar;
datatype	Entrada.	Tipo de dado de cada entrada enviada para o " <i>buffer</i> ";
dest	Entrada.	Processador de destino da mensagem;
tag	Entrada.	Código de identificação da mensagem;
comm	Entrada.	Comunicador (<code>MPI_COMM_WORLD</code>);

A função `MPI_Send()` é bloqueada, retornando apenas quando o conteúdo da mensagem é completamente copiado no *buffer* de envio. Ela envia para o processo de *my_id* igual a *dest* (pertencente ao comunicador `MPI_COMM_WORLD`) *count* dados do tipo *datatype* contido no argumento *msg*. A função guarda em um "*buffer*" do sistema o endereço do primeiro elemento da variável *msg* e aloca espaço na memória para *count* dados do tipo *datatype*. O "*buffer*" de envio consiste, portanto, de um espaço na memória do sistema suficiente para armazenar *count* dados do tipo *datatype*.

A mensagem é enviada com um respectivo *tag*, o qual fornece um código de identificação para a mesma. A função `MPI_Send()` retorna um número inteiro com o código de erro associado.

MPI_Recv(msg, count, datatype, source, tag, comm, status)

msg	Saída.	Endereço inicial do " <i>buffer</i> " de recebimento;
count	Entrada.	Número de " <i>entradas</i> " para receber;

<code>datatype</code>	Entrada.	Tipo de dado de cada entrada recebida para o " <i>buffer</i> ";
<code>source</code>	Entrada.	Processador de origem da mensagem;
<code>tag</code>	Entrada.	Código de identificação da mensagem;
<code>comm</code>	Entrada.	Comunicador (MPI_COMM_WORLD);
<code>status</code>	Saída.	Status do processador de origem.

A função `MPI_Recv()` recebe do processador de *my_id* igual a *source*, pertencente ao comunicador MPI_COMM_WORLD, *count* dados do tipo *datatype* contido no "*buffer*" de recebimento. O endereço inicial do "*buffer*" é armazenado na variável *msg*. A mensagem é recebida com um determinado *tag*, que a caracteriza.

O argumento *source* pode ser especificado como MPI_ANY_SOURCE, uma macro do MPI que permite que a função receba a mensagem de qualquer processo contido no comunicador.

O Status do processo é retornado na estrutura *status*, de tipo MPI_Status. Essa estrutura contém três campos: o MPI_SOURCE, o MPI_TAG e o MPI_ERROR, os quais contém, respectivamente, a *fonte*, o *tag* e o *código de erro* da função.

A função `MPI_Recv()` também é uma função *bloqueada*. Ela retorna apenas quando o conteúdo da mensagem foi completamente copiado no "*buffer*" de recebimento do sistema.

MPI_Barrier(comm, mpierr)

<code>comm</code>	Entrada.	Comunicador (MPI_COMM_WORLD);
<code>mpierr</code>	Saída	Variável inteira de retorno com o status da rotina

A função `MPI_Barrier()`, sincroniza de maneira explícita todos os processos de um grupo. Um processo de um grupo que utilize `MPI_Barrier()`, para de executar, até que todos os processos do mesmo grupo também executem um `MPI_Barrier()`.

MPI_Finalize(mpierr)

<code>mpierr</code>	Saída	Variável inteira de retorno com o status da rotina
		Mpierr = 0, Sucesso (MPI_SUCCESS)
		Mpierr < 0, Erro

Esta função finaliza o ambiente necessário para executar o MPI. A partir de sua chamada, todas as definições, macros e funções do MPI definidos em “mpi.h” deixaram de ter validade.

Importante: Essa rotina só deve ser chamada após se ter certeza da finalização de qualquer comunicação entre os processos.