

UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA CIVIL E AMBIENTAL

**ANÁLISE ORIENTADA A OBJETOS DE MÉTODOS
NUMÉRICOS DE DISCRETIZAÇÃO SEM MALHA**

JOSÉ MANOEL MORALES SÁNCHEZ

ORIENTADOR: ATHAIL RANGEL PULINO FILHO

TESE DE DOUTORADO EM ESTRUTURAS E CONSTRUÇÃO CIVIL

PUBLICAÇÃO: E.TD – 002 A/2003

BRASÍLIA / DF: DEZEMBRO - 2003

UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA CIVIL E AMBIENTAL
PROGRAMA DE PÓS-GRADUAÇÃO EM ESTRUTURAS E CONSTRUÇÃO CIVIL

ANÁLISE ORIENTADA A OBJETOS DE MÉTODOS
NUMÉRICOS DE DISCRETIZAÇÃO SEM MALHA

JOSÉ MANOEL MORALES SÁNCHEZ

TESE DE DOUTORADO SUBMETIDA AO DEPARTAMENTO DE ENGENHARIA CIVIL E AMBIENTAL DA FACULDADE DE TECNOLOGIA DA UNIVERSIDADE DE BRASÍLIA, COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA OBTENÇÃO DO GRAU DE DOUTOR EM ESTRUTURAS E CONSTRUÇÃO CIVIL.

APROVADO POR:



Athail Rangel Pulino Filho, D.Sc (UnB)
(Orientador)



Eldon Londe Mello, PhD (UnB)
(Examinador Interno)



Paul William Partridge, PhD (UnB)
(Examinador Interno)



Paulo Batista Gonçalves, D.Sc (PUC/Rio)
(Examinador Externo)



Nelson Francisco Favilla Ebecken, D.Sc (COPPE/UFRJ)
(Examinador Externo)

Brasília-DF, 12 de dezembro de 2003.

FICHA CATALOGRÁFICA

SÁNCHEZ, JOSÉ MANOEL MORALES

Análise Orientada a Objetos de Métodos Numéricos de Discretização sem Malha [Distrito Federal] 2003.

xix, 211 p., 210 x 297 mm (ENC/FT/UnB, Doutor, Estruturas e Construção Civil, 2003).

Tese de Doutorado – Universidade de Brasília. Faculdade de Tecnologia.

Departamento de Engenharia Civil e Ambiental.

- | | |
|---------------------------------|-----------------------------------|
| 1. Métodos sem Malha | 2. Análise Orientada a Objetos |
| 3. Método dos Elementos Finitos | 4. Método dos Resíduos Ponderados |
| I. ENC/FT/UnB | II. Título (série) |

REFERÊNCIA BIBLIOGRÁFICA

SÁNCHEZ, J. M. M. (2003). Análise Orientada a Objetos de Métodos Numéricos de Discretização sem Malha. Tese de Doutorado em Estruturas e Construção Civil, Publicação E.TD-002A/2003, Departamento de Engenharia Civil e Ambiental, Universidade de Brasília, Brasília, DF, 211 p.

CESSÃO DE DIREITOS

AUTOR: José Manoel Morales Sánchez.

TÍTULO: Análise Orientada a Objetos de Métodos Numéricos de Discretização sem Malha.

GRAU: Doutor

ANO: 2003

É concedida à Universidade de Brasília permissão para reproduzir cópias desta tese de doutorado e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte desta tese de doutorado pode ser reproduzida sem autorização por escrito do autor.



José Manoel Morales Sánchez.
SQN 208 Bloco B Apt. 304.
70853-020 Brasília – DF – Brasil.

À minha família e em especial a meu pai.

AGRADECIMENTO

Ao colegiado do Departamento de Tecnologia da Faculdade de Arquitetura e Urbanismo pela consideração, apreço e apoio durante o desenvolvimento deste trabalho.

Aos professores Athail, Luciano, Paul e William pelos ensinamentos nas disciplinas cursadas e no contato cotidiano.

Ao professor Guilherme pelo apoio em momento decisivo na conclusão do trabalho.

Ao professor Athail pela orientação paciente e tolerante e pela irrestrita confiança na elaboração do trabalho.

Aos professores Eldon e Nelson pela contribuição e incentivo durante a defesa do trabalho.

Aos colegas Buzar e Milton pela convivência e companherismo no desbravamento do doutorado.

Ao amigo Paulo Batista pelo ilimitado apoio acadêmico e fraterno ao longo de tantos anos.

Aos amigos José Humberto e Walter pela solidariedade e pelas reuniões para avaliação do mundo e de nossas vidas.

E, à minha família - Tomás, Teodora, Yeda, Socorro, Bernardo e Danilo – pelo amor fundamental e estímulo que motivou todo o esforço durante esse anos.

RESUMO

Este trabalho apresenta uma análise orientada a objetos de métodos de discretização sem malha. A análise estabelece o Método dos Resíduos Ponderados como arcabouço do modelo matemático que, em conjunto com funções de forma estabelecidas pelo Método de Mínimos Quadrados Móvel, permitem apresentar os seguintes Métodos sem Malha: Diferenças Finitas Generalizadas, Galerkin sem Malha, *Hp-Clouds* e Ponto Finito. Com o uso de funções de forma adequadas deduz-se também o Método dos Elementos Finitos.

O modelo computacional concebido dentro do paradigma da orientação a objetos propõe uma estrutura de classes que permitem a consideração em uma mesma estrutura de vários métodos de discretização e de diversos problemas, tais como: equação diferencial de segunda ordem, equação de Poisson e elasticidade bidimensional.

A arquitetura do sistema computacional é projetada com a utilização da linguagem de modelagem UML e é construída com base em padrões de projeto orientado a objetos. A implementação realizada com a linguagem de programação Java denominou-se de JFraMMe.

Um novo Método sem Malha é apresentado utilizando o critério de Petrov-Galerkin no Método dos Resíduos Ponderados e aplicando as funções de forma obtidas por expansão em série de Taylor no âmbito do Método de Mínimos Quadrados Móvel.

ABSTRACT

This work presents an object oriented analysis of several Meshless Methods. The mathematical model is based on the Method of Weighted Residuals with form functions determined by the Moving Least Square Method. This allows a unified presentation of the following Meshless Methods: Generalized Finite Difference, Element-Free Galerkin, *Hp-Clouds* and Finite Point Method. It is also shown how the use of appropriate form functions conducts to the traditional Finite Element Method formulation.

The computational model conceived here using an object-oriented paradigm proposes a framework that allows considering in the same structure several discretization methods and problems, such as, second order differential equation, Poisson equation and plain elasticity.

The software architecture is designed using the UML modeling language and built using object oriented design patterns. The implementation using the Java programming language is named JFraMME.

Finally, a new Meshless Method is proposed by applying the Petrov-Galerkin criterion with the Method of Weighted Residuals and form functions obtained by Taylor series expansion.

SUMÁRIO

1 - INTRODUÇÃO	1
1.1 - Contexto da Tese	1
1.2 - Métodos Numéricos de Discretização sem Malha	3
1.3 - Abordagem Orientada a Objetos de Métodos Numéricos	7
1.4 - Metodologia de Análise Orientada a Objetos	9
1.5 - Conteúdo da Tese	13
2 - MÉTODO DOS RESÍDUOS PONDERADOS	15
2.1 - Introdução	15
2.2 - Método dos Resíduos Ponderados	20
2.3 - Forma Discreta do Método dos Resíduos Ponderados	24
2.4 - Caráter Local dos Métodos de Discretização	31
2.5 - Exemplo de Aplicação – Barra Elástica	35
2.6 - Decomposição por Objetos Computacionais do Método dos Resíduos Ponderados	39
2.7 - Comentários Finais	42
3 - TÉCNICAS DE APROXIMAÇÃO E INTERPOLAÇÃO. FUNÇÕES DE FORMA	43
3.1 - Introdução	43
3.2 - Funções de Forma pelo Método de Mínimos Quadrados Móvel	46
3.3 - Base de Funções	55
3.4 - Funções de Ponderação	57
3.5 - Exemplo de Aplicação – Ajustamento de Superfícies	67
3.6 - Modelagem Computacional das Funções de Forma	72
3.7 - Comentários Finais	79

4 - MÉTODOS NUMÉRICOS DE DISCRETIZAÇÃO SEM MALHA	80
4.1 - Introdução	80
4.2 - Método das Diferenças Finitas Generalizadas	82
4.3 - Método de Galerkin sem Malha	96
4.4 - <i>Hp-Clouds</i> – Método sem Malha Hierárquico	101
4.5 - Método do Ponto Finito	115
4.6 - Exemplo de Aplicação	123
4.7 - Modelagem Computacional dos Métodos sem Malha	128
4.8 - Considerações Finais	133
5 - MÉTODO DE GALERKIN SEM MALHA EM ELASTICIDADE	135
5.1 - Introdução	135
5.2 - Formulação Discreta em Elasticidade	136
5.3 - Tratamento de Descontinuidade de Material	147
5.4 - Algoritmo do Método de Galerkin sem Malha	151
5.5 - Exemplo de Aplicação	158
5.6 - Modelagem das Classes de um Problema	163
5.7 - Considerações Finais	167
6 – UMA ESTRUTURA COMPUTACIONAL PARA OS MÉTODOS SEM MALHA	168
6.1 - Introdução	168
6.2 - Modelagem do Domínio e da Discretização	169
6.3 - Classe Analysis Baseada em Algoritmos	177
6.4 - Classes Matriciais – O Pacote JMatrix.	180
6.5 - JFraMMe: Uma estrutura computacional em Java para os Métodos sem Malha	184
6.6 - Considerações Finais	185
7 - MÉTODO SEM MALHA PETROV-GALERKIN COM APROXIMAÇÃO LOCAL EM SÉRIE DE TAYLOR	186
7.1 - Introdução	186
7.2 - Método sem Malha Petrov-Galerkin Taylor	187
7.3 - Implementação Computacional	191

7.4 - Considerações Finais	193
8 - CONCLUSÕES E TRABALHOS FUTUROS	194
8.1 - Introdução	194
8.2 - Resumo e Contribuições	195
8.3 - Conclusões	198
8.4 - Trabalhos Futuros	200
REFERÊNCIAS	202

LISTA DE TABELAS

Tabela 3.1 – Funções de Ponderação.	66
Tabela 4.1 – Algoritmo para a construção do traçado de funções de forma de um ponto nodal.	103
Tabela 4.2 – Algoritmo de recobrimento do domínio.	111
Tabela 5.1 – Algoritmo do Método de Galerkin sem Malha.	152

LISTA DE FIGURAS

Figura 1.1 – Simbologia básica UML para modelagem de classes.	11
Figura 2.1 – Espaço da solução de um domínio bi-dimensional.	16
Figura 2.2 – Aproximação do meio contínuo por domínio discreto.	16
Figura 2.3 – Formulações forte, integral e variacional, e métodos de discretização.	18
Figura 2.4 – Modelo de classes computacionais do Método dos Resíduos Ponderados.	19
Figura 2.5 – Domínio bidimensional regido pela equação de Poisson.	26
Figura 2.6 – Discretização do domínio em pontos nodais. Pontos nodais gêmeos em uma singularidade do contorno.	30
Figura 2.7 – Função de forma global sobre todo o domínio e função de forma local.	31
Figura 2.8 – Operação de montagem da contribuição local na matriz global com critério de colocação e Galerkin.	33
Figura 2.9 – Conectividade de pontos nodais por molécula, elemento e nuvem.	34
Figura 2.10 – Barra elástica submetida a esforço axial.	35
Figura 2.11 – Funções de forma lineares definidas localmente.	36
Figura 2.12 – Soluções analítica e discreta da barra elástica.	38
Figura 2.13 – Modelo Físico, Modelo Matemático e Modelo Computacional.	40
Figura 3.1 - Ponderação com suporte definido a partir de um ponto qualquer.	51
Figura 3.2 - Ponderação com suporte definido para cada ponto nodal.	52
Figura 3.3 - Função gaussiana truncada.	60
Figura 3.4 – Efeito da função de ponderação sobre o Método de Mínimos Quadrados Móvel.	61
Figura 3.5 - Influência do parâmetro β na função de ponderação gaussiana.	62
Figura 3.6 - Avaliação do efeito da dimensão do suporte.	62

Figura 3.7 - Variação da função de forma com a dimensão do suporte.	63
Figura 3.8 - Relevo Lancaster-Šalkauskas.	67
Figura 3.9 - Amostra de dados aleatórios do Relevo Lancaster-Šalkauskas.	68
Figura 3.10 - Função de ponderação para determinação do relevo de Lancaster e Šalkauskas.	70
Figura 3.11 - Superfície interpolada por Mínimos Quadrados Móveis.	71
Figura 3.12 – Interfaces para determinação das funções de forma.	72
Figura 3.13 – Modelo de classes para elementos isoparamétricos.	73
Figura 3.14 – Classes para modelagem do Método de Mínimos Quadrado Móvel.	74
Figura 3.15 – Classes para modelagem das bases de funções.	75
Figura 3.16 – Classes que modelam as funções de ponderação.	77
Figura 3.17 – Classes para modelagem da nuvem de pontos e estratégias de construção.	78
Figura 4.1 – Critério de mínimo de 6 pontos nodais mais próximos para construção da nuvem.	88
Figura 4.2 – Seleção da molécula pelo critério dos quatro quadrantes.	88
Figura 4.3 – Molécula do contorno com um ponto no contorno e quatro no domínio.	92
Figura 4.4 – Modelagem das classes para obtenção da função de forma por expansão em série de Taylor.	95
Figura 4.5 – Estrutura de células de integração independentes dos pontos nodais.	98
Figura 4.6 – Domínio discretizado por pontos nodais com suportes circulares.	101
Figura 4.7 – Funções de forma geradas a partir de um suporte e de suportes múltiplos.	102
Figura 4.8 – Função de ponderação <i>B-Spline</i> quártica.	109
Figura 4.9 – Algoritmo de recobrimento para distribuição regular de pontos nodais.	110
Figura 4.10 – Distribuição em <i>cluster</i> com algoritmo de recobrimento de mínima distância.	110
Figura 4.11 – Algoritmo de recobrimento baseado em triangulação.	111

Figura 4.12 – Algoritmo de busca multidirecional para determinação dos suportes que interceptam um ponto.	112
Figura 4.13 – Implementação das classes computacionais para produção das funções de forma do Método <i>Hp-Clouds</i> .	114
Figura 4.14 – Método de Mínimos Quadrados Móvel com ponto de avaliação igual a um dos pontos nodais da nuvem.	116
Figura 4.15 – Domínio finito em equilíbrio de fluxos (a), e equilíbrio na vizinhança de um contorno Neumann (b).	118
Figura 4.16 – Domínio regido pela equação de Laplace e solução analítica.	124
Figura 4.17 – Solução da equação de Laplace com o Método das Diferenças Finitas Generalizadas.	125
Figura 4.18 – Solução da equação de Laplace com o Método de Galerkin sem Malha.	125
Figura 4.19 – Solução da equação de Laplace pelo Método <i>Hp-Clouds</i> .	126
Figura 4.20 – Solução da equação de Laplace com o Método dos Elementos Finitos.	127
Figura 4.21 – Interface para definição de métodos de discretização.	129
Figura 4.22 – Classes para construção das matrizes globais.	131
Figura 4.23 – Classes para incorporação de comportamento de <i>assembler</i> a matrizes e vetores.	132
Figura 5.1 – Problema de elasticidade com diagrama de corpo livre.	137
Figura 5.2 – Funções descontínuas para discretização do contorno essencial Γ_u .	144
Figura 5.3 - Corpo não-homogêneo.	147
Figura 5.4 – Tratamento de interface entre dois materiais.	148
Figura 5.5 – Definição dos objetos do problema e do domínio.	153
Figura 5.6 – Critérios de discretização aleatório e grade ortogonal.	153
Figura 5.7 – Discretização de células de integração.	154
Figura 5.8 – Critério de construção poligonal da nuvem de pontos.	156
Figura 5.9 – Chapa submetida à tração axial.	158

Figura 5.10 – Solução analítica para a chapa submetida à tração axial.	159
Figura 5.11 – Discretização da chapa: pontos nodais e células de integração.	159
Figura 5.12 – Variação do erro relativo com o parâmetro α , para os casos de carregamento e de distribuição de pontos nodais.	160
Figura 5.13 – Erro relativo versus α em função da função de ponderação.	161
Figura 5.14 – Erro relativo em função da ordem da base.	161
Figura 5.15 – Solução aproximada e exata da chapa submetida à tração axial.	162
Figura 5.16 – Classes para modelagem de um problema.	165
Figura 5.17– Classes para vinculação de um problema a um método.	166
Figura 6.1 – Representação da geometria de um domínio através de pontos, contornos e caminhamentos.	169
Figura 6.2 – Classes para modelagem geométrica de um domínio.	170
Figura 6.3 – Classe para discretização em pontos nodais da geometria de um domínio.	171
Figura 6.4 – Procedimentos de discretização do domínio em pontos nodais.	172
Figura 6.5 – Classe para discretização do contorno.	173
Figura 6.6 - Representação da discretização do contorno com propriedades nodais.	174
Figura 6.7 – Classes para discretização das células de integração e elementos finitos.	175
Figura 6.8 – Algoritmos para geração de células de integração.	176
Figura 6.9 – Interface para classe de análise do problema.	178
Figura 6.10 – Projeto de classes do pacote JMatrix.	182
Figura 6.11 – Classe para modelagem do sistema linear de equações.	183
Figura 7.1 – Modelagem computacional do Método Petrov-Galerkin Taylor.	192

LISTA DE SÍMBOLOS

A	- área da seção transversal
$A()$	- operador diferencial
A_k	- operador de assembler
$A(x)$	- matriz da formulação das funções de forma por mínimos quadrados
$B()$	- operador diferencial
$B(x)$	- matriz da formulação das funções de forma por mínimos quadrados
B	- matriz com derivadas das funções de forma
C^n	- função de continuidade n
C	- matriz que relaciona parâmetros nodais com a base de funções
D	- matriz gradiente simétrico
Du	- vetor com derivadas da incógnita do problema na expansão por Taylor
Dw	- vetor com função de forma e derivadas da função de ponderação
E	- módulo de elasticidade
$E(u)$	- funcional de quadrados do erro
$E(u, x)$	- funcional ponderado de quadrados do erro
E	- matriz de elasticidade
F	- matriz da formulação do Método dos Resíduos Ponderados
G	- matriz da formulação do Método dos Resíduos Ponderados
H	- matriz da formulação do Método dos Resíduos Ponderados
K_{Ω_k}	- matriz local
K	- matriz da formulação do Método dos Resíduos Ponderados
L	- comprimento da barra
L^p	- base de funções hierárquicas
N	- norma ponderada do erro

N	- número de pontos nodais na célula
N_u, N_f	- número de pontos nodais nos contornos essencial e forçado
\bar{N}	- funções de forma nos contornos
N_i	- funções de forma
\mathbf{N}	- vetor com as funções de forma
\mathbf{P}	- matriz que relaciona incógnitas com coeficientes nodais
\mathbf{P}_n	- matriz que relaciona as derivadas normais dos pontos de contorno com a expansão por Taylor de um ponto nodal; matriz com co-senos diretores
$Q(x)$	- fonte distribuída
Q_0	- fluxo no contorno
\mathbf{U}	- matriz da formulação do Método dos Resíduos Ponderados
w_j, \bar{w}_j e $\bar{\bar{w}}_j$	- funções de ponderação
\mathbf{W}	- matriz diagonal com funções de ponderação
a_i	- parâmetro nodal
\mathbf{a}	- vetor com parâmetros nodais
b	- forças ou fontes externas
b_{ji}	- coeficientes na aproximação por partição da unidade
b_x, b_y	- forças de corpo por unidade de volume
\mathbf{b}	- vetor com forças de corpo
$d = \ x - x_i\ $	- distância entre pontos
dm_i	- dimensão do suporte no ponto
dm_x, dm_y	- dimensão do suporte em x e y
d_{\min}	- dimensão mínima do suporte para condicionamento do sistema local
h	- espaçamento nodal; dimensão do suporte na aproximação por reprodução do núcleo
n_c	- número de células
n_g	- número de pontos de integração gaussiana

n_p	- número total de pontos nodais no domínio
n_x, n_y	- co-senos diretores
p_i	- base de funções polinomiais
P	- vetor com base de funções
q_i	- base de funções
q	- vetor com o fluxo no contorno essencial
$r = d_i / dm_i$	- distância adimensional
r_x, r_y	- distâncias dimensionais em x e y
\bar{t}_x, \bar{t}_y	- tensões prescritas
t	- forças ou fontes externas
\mathbf{t}	- vetor com tensões impostas
u	- incógnita do problema; solução exata
u^h	- solução aproximada
u_p	- valor prescrito da incógnita do problema
\mathbf{u}	- vetor com deslocamentos incógnitos
u_n	- derivadas normais da incógnita do problema
$\bar{\mathbf{u}}$	- vetor com deslocamentos prescritos
$w(x - y, h)$	- núcleo (<i>kernel</i>) ou função peso
$w_i(x)$	- função de ponderação na aproximação por mínimos quadrados
\mathbf{w}	- vetor com funções de ponderação nas direções x e y
Γ	- contorno do problema
Γ_f	- contorno forçado ou de Neumann
Γ_u	- contorno essencial ou Dirichlet
Ω	- domínio do problema
Ψ_k	- função partição da unidade

α	- coeficiente que controla o número de pontos abrangidos pelo suporte
α_i	- coeficientes
\mathbf{a}	- vetor com os coeficientes
β	- parâmetro da função gaussiana
δ	- delta de Dirac
ε	- erro
$\boldsymbol{\varepsilon}$	- vetor deformação específica
$\phi(x)$	- funções de forma por mínimos quadrados
ϕ_i	- funções base
$\phi_s(x)$	- derivada da função de forma
$\phi_{sr}(x)$	- segunda derivada da função de forma
$\phi_a(x)$	- função de forma do ponto
$\boldsymbol{\varphi}$	- vetor com as funções base
$\boldsymbol{\varphi}_n$	- vetor com as funções de forma no contorno
ν	- coeficiente de Poisson
ρ	- constante da equação de Poisson
$\boldsymbol{\sigma}$	- vetor de tensões
$\boldsymbol{\sigma}_u$	- vetor tensões no contorno essencial
σ_x, σ_y	- tensões normais
τ_{xy}	- tensão cisalhante
ψ_j	- funções de forma com salto
$\frac{\partial u}{\partial n}$	- derivada normal
$\ \cdot\ $	- norma

1 - INTRODUÇÃO

1.1 - CONTEXTO DA TESE

O escopo desta Tese se insere na área de conhecimento denominada Mecânica Computacional, definida como campo da mecânica que soluciona problemas específicos através de um processo de simulação que utiliza métodos numéricos implementados em computadores digitais (Felippa, 2000).

Na segunda metade do século vinte os métodos numéricos da mecânica computacional se constituíram na base na qual se construíram as principais conquistas da engenharia e da física. E, não se pode deixar de reconhecer que o desenvolvimento do Método dos Elementos Finitos foi o mais importante avanço da simulação numérica desse período, sendo o seu uso amplamente disseminado por todos os campos da engenharia e da ciência em geral. Essa abrangência se deve à facilidade de modelagem de domínios complexos, à intuitiva natureza física de sua formulação, e à sua precisão (Zienkiewicz, 1992).

Em algumas áreas de aplicação o Método dos Elementos Finitos é inerentemente limitado, tais como problemas de fronteiras móveis, malhas complexas, auto-refinamento da solução, dentre outras. Nos últimos anos vem ocorrendo um esforço para se investigar e desenvolver novos métodos numéricos que sejam mais flexíveis, versáteis e robustos. A característica fundamental desses novos métodos reside numa discretização feita apenas através de pontos nodais, abolindo a malha (ou grade) de elementos. Estes métodos são usualmente chamados de Métodos sem Malha (*Meshless Methods*).

O esforço mencionado foi oportunamente documentado numa edição especial da *Computer Methods in Applied Mechanics and Engineering* intitulado *Meshless Methods*, editada por Liu, Belytschko e Oden (1996).

O avanço na simulação numérica não teria sido possível sem o notável desenvolvimento da ciência da computação e da tecnologia de construção de programas. A implementação (programação ou codificação) evoluiu de um paradigma procedural, onde a

concepção é funcionalista, para um paradigma orientado a objetos, onde a concepção é centrada na modelagem dos objetos do mundo real.

A adoção do paradigma de análise, projeto e programação orientada a objetos objetiva a criação de programas extensíveis e robustos, além de facilitar a construção de sistemas computacionais complexos a partir da grande reutilização de código pronto e bem testado.

Em artigo sobre avanços da mecânica computacional na direção de um ambiente automatizado, Tworzydło e Oden (1993), apontam que a programação orientada a objetos como uma das ferramentas que permitirão a construção futura de um ambiente inteligente que permita a tomada de decisão automatizada em engenharia.

O objetivo deste trabalho é o desenvolvimento de uma estrutura computacional orientada a objetos que facilite a criação de programas de aplicação para a resolução numérica de problemas modelados pelos Métodos sem Malha.

1.2 - MÉTODOS NUMÉRICOS DE DISCRETIZAÇÃO SEM MALHA

O objetivo dos Métodos sem Malha é a discretização de domínios através de *pontos nodais*, que são *pontos geométricos* associados a parâmetros da solução do problema do domínio. Para a classificação de um procedimento de discretização como sendo sem malha, Oñate *et al.* (1996), propõem que as seguintes condições sejam satisfeitas:

1. A discretização da função incógnita, e de suas derivadas, deve ser feita apenas com pontos nodais do domínio;
2. A função de ponderação para aproximação local deve ser definida somente com pontos nodais do domínio;
3. Não deve ser necessária integração no domínio ou a integração numérica não deve depender do procedimento de aproximação local.

Uma caracterização mais imediata seria definir os Métodos sem Malha como aqueles onde os procedimentos de solução não dependem da informação externa da conectividade entre pontos nodais do domínio, sendo essa conectividade determinada por algum procedimento matemático e / ou computacional.

Os métodos assim classificados utilizam três estratégias para definição da função de aproximação local para a função incógnita do problema (Belytschko *et al.*, 1996):

1. Mínimos Quadrados Móvel (Lancaster e Šalkauskas, 1981);
2. Aproximação por reprodução da função de ponderação (Lucy *apud* Belytschko *et al.*, 1994); e,
3. Partição da Unidade (Babuška e Melenk, 1997).

Na verdade as duas primeiras formas de aproximação local podem ser equivalentes, e a aproximação por Mínimos Quadrados Móvel pode ser considerada uma instância específica da partição da unidade.

As primeiras abordagens no sentido da criação dos Métodos sem Malha foram empreendidas no âmbito do Método das Diferenças Finitas com a adoção de esquemas que permitem a utilização de malhas arbitrárias de pontos nodais (Perrone e Kao, 1975). Esta abordagem denominada *Generalized Finite Difference Method* (GFD), estabelece uma molécula (*star* ou *stencil*) de pontos nodais utilizando expansão por série de Taylor. Uma implementação numericamente mais robusta utilizando um processo de mínimos quadrados ponderado foi realizada por Liszka e Orkisz (1980).

Um caminho alternativo utilizado nas primeiras experiências com métodos sem malha é a técnica conhecida como *Smooth Particle Hydrodynamics* (SPH) criada em 1977 e aplicada em problemas de astrofísica e cosmologia. Este método, também conhecido como *Free Lagrange*, depende somente de um conjunto disperso de pontos ou partículas. A aproximação local é obtida por reprodução da função de ponderação, também chamada função núcleo. O método conduz a bons resultados para problemas onde há ausência de contornos, embora em outras aplicações os resultados não sejam tão precisos. Uma apresentação deste método pode ser encontrada no artigo de Randles e Libersky (1996).

A utilização da aproximação local por mínimos quadrados ponderados móvel no Método de Galerkin foi investigada pela primeira vez por Nayroles *et al.* (1992). Sua técnica denominada de *Diffuse Element Method* (DEM) emprega uma nuvem dispersa de pontos nodais conformada pelo suporte da função de ponderação truncada, que é utilizada na obtenção da aproximação local. A integração das equações originadas pelo Método de Galerkin é feita com a utilização de uma grade auxiliar. Belytschko *et al.* (1994), propuseram uma extensão dessa técnica não desprezando nenhum termo da aproximação das derivadas e uma estrutura regular de células sem conexão com os pontos nodais da aproximação local. Outra consideração importante foi o tratamento adequado da condição de contorno essencial modelada por multiplicadores de Lagrange. Essa nova abordagem denominada *Element-Free Galerkin* (EFG) é sem dúvida o método sem malha mais pesquisado e com aplicações que abrangem diversas áreas da mecânica computacional, e até de outros campos da engenharia.

Liu *et al.* (1995), desenvolveram abordagem descendente da aproximação local por reprodução de núcleo (função de ponderação) do método SPH. As funções de aproximação foram construídas a partir da reprodução de uma função contínua do núcleo ao invés de uma função discreta. Liu aponta a transformada de Fourier como um exemplo

de reprodução de núcleo. O método denominado de *Reproducing Kernel Particle Method* (RKPM) é objeto de intensa pesquisa liderada pelo grupo de Liu e Belytschko.

Duarte e Oden (1996) apresentaram um método baseado na aproximação local construída a partir de funções que se comportam como uma partição da unidade. As funções de aproximação são organizadas em famílias de funções que permitem a implementação de esquemas auto-adaptáveis hierárquicos tipo *hp*. As funções de aproximação são enriquecidas com funções de ordem mais elevada de modo a permitir um controle localizado do erro por nuvem de pontos, de forma semelhante àquela utilizada no Método dos Elementos Finitos. A função partição da unidade empregada é a função de Shepard, obtida a partir do método dos mínimos quadrados móvel com função base constante igual à unidade. Este método, denominado de *Hp-Clouds*, vem sendo pesquisado por grupo brasileiro em conjunto com Duarte em aplicações em mecânica dos sólidos (Garcia *et al.*, 2000).

Oñate, Idelsohn *et al.* (1996), apresentaram um método sem malha baseado no critério de colocação do Método dos Resíduos Ponderados e com função de aproximação construída a partir do método dos mínimos quadrados ponderados. O procedimento é bastante semelhante ao Método das Diferenças Finitas Generalizadas. No método de Oñate, denominado de *Finite Point Method* (FPM), as derivadas a serem introduzidas nas equações diferenciais que regem o problema em estudo são obtidas após a obtenção da função de aproximação através do método dos mínimos quadrados ponderados. Nas Diferenças Finitas Generalizadas as derivadas são obtidas diretamente no processo de mínimos quadrados. Por utilizar o critério de colocação a integração é evitada, contudo os resultados apresentam oscilação numérica que é contornada com acréscimo de termo de estabilização nas equações diferenciais originais. Este método tem uso marcante em problemas de fluidos tendo sido aplicado em elasticidade por Oñate *et al.* (2001).

Atluri e Zhu (1999) e Zhu (2000) propuseram três métodos em busca de uma solução totalmente sem malha. Os três métodos utilizam uma formulação integral fraca que é obtida por integração por partes da expressão de resíduos ponderados e aplicada de forma local e de modo que a integração se efetue em um domínio de formato conhecido (no caso circular) definido pelo suporte da função de ponderação. As funções de aproximação local são as obtidas pelo método de mínimos quadrados móvel. Dos três métodos propostos o que parece mais promissor é o denominado *Meshless Regular Local*

Boundary Integral Equation (MRLBIE) uma vez que não envolve o tratamento numérico de integrais singulares.

Conforme se evidencia nesta resenha os Métodos sem Malha se constituem em fronteira promissora na modelação numérica em Mecânica Computacional. Em virtude da diversidade de abordagens este trabalho se aterá aos métodos que possuem função de aproximação local geradas a partir do método de mínimos quadrados móvel e dentre esses os seguintes:

1. Diferenças Finitas Generalizadas (*Generalized Finite Difference Method* –GFD).
2. Método de Galerkin sem Malha (*Element-Free Galerkin* - EFG).
3. Método sem Malha Hierárquico (*Hp-Clouds*).
4. Método do Ponto Finito (*Finite Point Method* - FPM).

1.3 - ABORDAGEM ORIENTADA A OBJETOS DE MÉTODOS NUMÉRICOS

A necessidade cada vez mais freqüente da construção de programas de computadores de alta complexidade exigiu uma abordagem com visão de engenharia tanto no processo quanto nos instrumentos de desenvolvimento dos aplicativos. Nesse sentido se estabelece como enfoque a modelação do objeto, ou melhor, dos conceitos do mundo real. Um objeto (conceito concreto) é compreensível a todos envolvidos na interação com um programa de computador. A seqüência de desenvolvimento, dentro desse paradigma (orientação a objetos), deve basear-se na análise conceitual exaustiva, no projeto (*design*) detalhado e documentado, e numa programação baseada em modularidade e reutilização de componentes de programação previamente construídos e testados.

No âmbito da modelação computacional dos métodos numéricos de discretização esse enfoque foi reportado pela primeira vez no início dos anos 1990. Uma das primeiras aplicações detalhadas da modelação orientada a objetos no Método dos Elementos Finitos foi empreendida por Forde *et al.* (1990).

Na última década diversos autores promoveram o uso da orientação a objetos a diversos problemas de engenharia modelados pelo Método dos Elementos Finitos, como análise dinâmica (Pidaparti e Hudli, 1993) e análise não-linear (Dubois-Pèlerin e Pegon, 1998). Uma área que aparece como promissora é a utilização de técnicas de computação paralela aplicada ao Método dos Elementos Finitos (Masters *et al.*, 1997).

Ainda nesse período Zimmermann empreendeu profícua pesquisa notadamente em análise não-linear e na programação simbólica do Método dos Elementos Finitos (Eyheramendy e Zimmermann, 2001).

A utilização de bibliotecas comerciais de classes de objetos foi reportada por Sampath e Zabarar (2000). Em seu trabalho mencionam a biblioteca *Diffpack* desenvolvida para modelação de equações diferenciais parciais discretizadas pelos Métodos dos Elementos Finitos e Diferenças Finitas.

Na literatura consultada não se observou a aplicação de orientação a objetos aos Métodos sem Malha em seu conjunto. Krysl e Belytschko (2001), apresentaram dentro de um enfoque procedural uma biblioteca para determinação de funções de forma para o Método de Galerkin sem Malha.

Em consonância com o enfoque de construção de programas de computador por orientação a objetos se empreenderá neste trabalho grande esforço na análise conceitual e no projeto do sistema computacional. Num sentido oposto ao que se depreende da literatura onde o foco é a programação orientada a objetos com a linguagem C++.

1.4 - METODOLOGIA DE ANÁLISE ORIENTADA A OBJETOS

A transformação da análise teórica dos Métodos sem Malha em classes e decorrentes objetos será empreendida através da utilização sistemática do paradigma da orientação a objetos computacionais (Rumbaugh *et al.*, 1991) .

Esse enfoque computacional objetiva uma implementação (programação) que seja robusta, adaptável e reutilizável. Em outras palavras uma implementação que seja resistente a situações críticas como entrada de dados inesperados, que tenha capacidade de evolução ou expansão e que seus componentes (classes) possam ser utilizados em várias aplicações conduzindo a uma economia de escala (Goodrich e Tamassia, 2002).

Para consecução desses objetivos alguns princípios devem ser seguidos como o da **abstração**, o **encapsulamento** e a **modularidade**. Por abstração entende-se a decomposição de um sistema em partes fundamentais e genéricas que possam ser repetidas em outros sistemas. No caso dos métodos numéricos em engenharia um exemplo desse tipo de abstração seriam as matrizes. O encapsulamento busca esconder a complexidade interna e a proteção dos dados de um objeto dos usuários desse objeto, facilitando a sua reutilização. Do ponto de vista computacional, um usuário não necessita saber como uma matriz será invertida mas como acessar o resultado desejado. A modularidade se refere à organização do sistema em partes funcionais que guardam uma hierarquia definida. No exemplo de matrizes a modularidade induz a uma separação das operações entre matrizes em rotinas específicas.

Esses princípios podem ser implementados em diversas linguagens tais como Ada, C++ e Java, dentre outras. Neste trabalho adotou-se a linguagem Java devido à sua forte aderência aos conceitos de orientação a objetos e aos princípios acima mencionados, além da motivação apresentada a seguir.

A aplicação do princípio da abstração em projetos de computação numérica leva ao conceito de **tipos abstratos de dados** (*abstract data type – ADT*). Um tipo abstrato é um modelo matemático com uma coleção de operações definida nesse modelo (Aho *et al.*, 1987). Um tipo abstrato é uma generalização dos tipos primitivos de dados (por exemplo: integer e double), sendo as operações generalizações das operações primitivas (por exemplo: adição e multiplicação).

A linguagem Java possui uma sintaxe específica para definição de tipos abstratos que é a *interface*. A interface não passa de uma lista de declaração de métodos que devem ser materializadas pelas classes que implementam essa interface. Dessa forma todas as classes que implementam uma interface possuem comportamento externo similar diferenciando-se pela forma interna que se comportam.

Esta abordagem permite que um programa de computador possa ser acrescido de novas classes concretas sem que o sistema existente tenha de ser alterado, desde que a especificação da interface, também chamada de protocolo, seja atendida.

A herança é um mecanismo que permite a construção modular e hierárquica de programas. Através deste mecanismo uma subclasse herda todos os dados e operações da superclasse a que está vinculada. Neste caso uma subclasse expande ou especializa o comportamento de uma classe. Por exemplo uma matriz pode ter uma subclasse para a construção de matrizes quadradas, que especializa o conceito impondo que as dimensões de uma matriz sejam iguais.

A nova classe pode sobre-carregar, ou substituir, uma operação de sua superclasse de forma a refinar ou especializar o seu comportamento. Em algumas linguagens o sobre-carregamento pode ser realizado nos operadores dos tipos primitivos ('+', '-', '*', '/'). Esta capacidade não foi implementada na linguagem Java.

A representação de um sistema computacional numa perspectiva de análise, projeto e mesmo de implementação é realizada através da notação UML (*Unified Modeling Language*). Esta linguagem de modelação de programas foi estabelecida em 1995 pelo OMG (*Object Management Group*) a partir da fusão dos três métodos de modelação mais proeminentes à época (Fowler e Scott, 2000; Furlan, 1998). Trata-se de uma notação intensamente visual e gráfica utilizada para especificar, visualizar, documentar e construir programas de computador.

Os recursos da notação UML a serem empregados no processo de análise, projeto e implementação dos Métodos sem Malha são os mais comuns, tais como a notação de interface, classe, herança, associação etc. A Figura 1.1 apresenta a simbologia comentada dos elementos da notação empregados neste trabalho. Os diagramas foram produzidos utilizando-se a versão comunitária da ferramenta CASE (*Computer Aided Software Engineering*) de construção de programas *Poseidon for UML* (2003). Esta

ferramenta gera automaticamente o código de programação Java correspondente ao projeto desenhado.

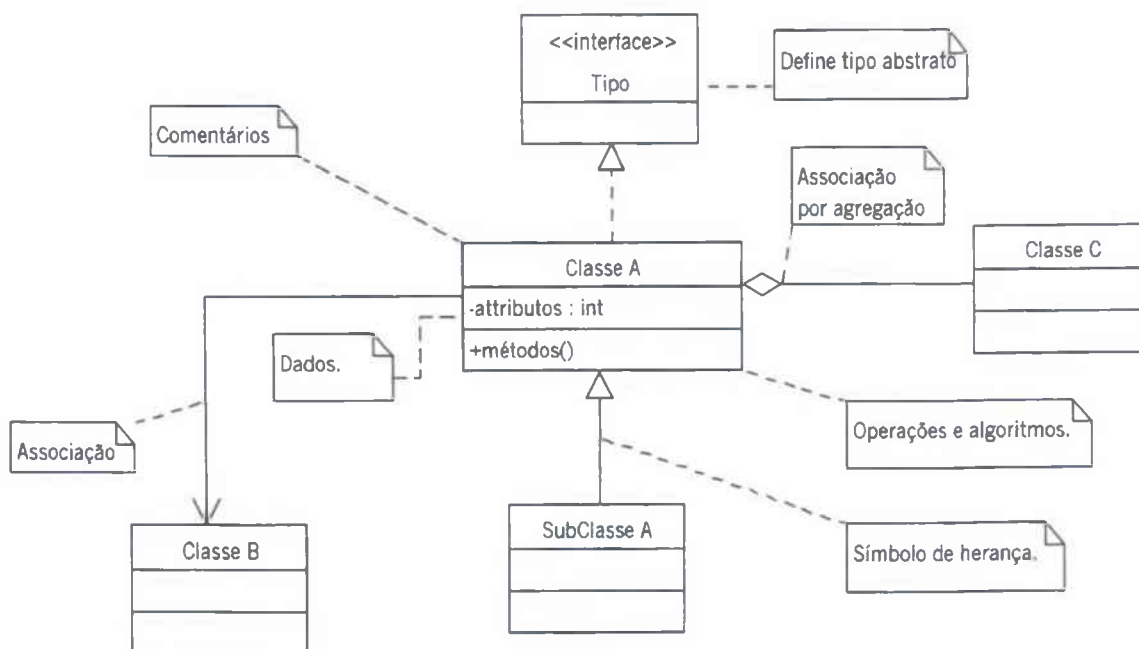


Figura 1.1 – Simbologia básica UML para modelagem de classes.

A utilização dos conceitos de orientação a objetos conduz a soluções de projetos de classes e de comunicação entre objetos que podem ser repetidas como padrão. Gamma *et al.* (2000) descrevem uma série de padrões de boas soluções reutilizáveis para construção de programas orientados a objetos. Estas descrições de objetos e classes que se comunicam, devem ser customizadas para resolver um problema específico em um contexto particular. O conhecimento dos padrões de projeto é fundamental para a construção de programas robustos fundados em soluções já testadas. Exemplos da aplicação de alguns desses padrões serão descritos ao longo deste trabalho.

Como referido acima, os experimentos das soluções de programação propostas serão realizados com a linguagem Java (Jandl, 2002; Lemay e Cadenhead, 2001). Mais do que uma linguagem, Java se constitui num grande ambiente de desenvolvimento e execução de programas. Daí resulta uma ampla possibilidade de utilização tanto para aplicações comerciais como científicas.

Além de ser uma linguagem concebida, e não adaptada, para ser orientada a objetos, o que facilita a compreensão desses conceitos e a sua implementação, Java possui extrema portabilidade sendo independente de plataforma e de sistema operacional.

Uma questão relevante que se coloca é a de *performance*. Como já mencionado uma característica chave de Java é a portabilidade. Para isto Java opera numa forma pseudo-interpretada. As técnicas de compilação *just-in-time* e os avanços da máquina virtual Java, criaram condições para performances aceitáveis e comparáveis em muitos casos aos obtidos por linguagens puramente compiladas como C++ e Fortran (Moreira *et al.*, 2000; Boisvert *et al.*, 1998) . Cabe mencionar a existência de compiladores Java para plataformas específicas, que aumentam a performance, com a conseqüente eliminação da portabilidade.

1.5 - CONTEÚDO DA TESE

A organização do trabalho segue o enfoque conceitual requerido pela análise orientada a objetos. Assim o Capítulo 2 descreve a formulação matemática e decorrente modelação computacional do Método dos Resíduos Ponderados. Como se trata da abstração mais abrangente na abordagem orientada a objetos dos Métodos sem Malha o capítulo conclui com a apresentação do mapa a ser trilhado para a construção da interface de programação de aplicativos, cujos pacotes de classes serão descritos nos capítulos seguintes.

No Capítulo 3 desenvolve-se o procedimento crítico na concepção dos métodos de discretização de maneira geral e dos Métodos sem Malha em particular que são as técnicas de aproximação e interpolação para construção de funções de aproximação local denominadas usualmente de funções de forma.

Os Métodos sem Malha analisados na tese estão detalhados no Capítulo 4. Como assinalado anteriormente, optou-se por estudar quatro métodos, dentre os mais significativos: Método das Diferenças Finitas Generalizadas, Método de Galerkin sem Malha, Método sem Malha Hierárquico e Método do Ponto Finito. Apresenta-se ainda um padrão para a construção de classes que modelam um método de discretização.

No Capítulo 5 apresenta-se o detalhamento da modelação computacional de um problema utilizando-se como exemplo o Método de Galerkin sem Malha, aplicado à elasticidade bidimensional.

O projeto orientado a objetos da interface de programação para implementação de aplicativos é apresentado no Capítulo 6, em complementação e detalhamento das classes específicas apresentadas nos capítulos anteriores. Constam, também, as classes para modelação do domínio e de sua discretização, de especificação do problema, dos métodos de discretização e de classes matriciais.

O Capítulo 7 ilustra como o projeto proposto permite a expansão para novos métodos, constituindo-se em bancada de experimentação numérica desses métodos. Apresenta-se um método novo que utiliza o critério de Petrov-Galerkin no Método dos Resíduos Ponderados e expansão em série de Taylor na definição das funções de forma da aproximação local como utilizado no Método das Diferenças Finitas Generalizadas.

Uma compilação das principais conclusões mencionadas ao longo do desenvolvimento do trabalho encontra-se no Capítulo 8. Que também relaciona algumas possibilidades de continuidade da pesquisa e de trabalhos futuros.

2 - MÉTODO DOS RESÍDUOS PONDERADOS

2.1 - INTRODUÇÃO

A formulação de problemas de meios contínuos em engenharia conduz, via de regra, a sistemas de equações diferenciais e condições de contorno impostas às variáveis incógnitas em um determinado domínio. Esses sistemas de equações são compostos pelas chamadas equações de campo (*governing equations*), e descrevem o modelo mecânico em estudo. Para um problema escalar essas equações têm a forma

$$A(u) = b \quad \text{em } \Omega \quad (2.1)$$

e condições de contorno

$$\begin{aligned} B(u) &= t & \text{em } \Gamma_t \\ u - u_p &= 0 & \text{em } \Gamma_u \end{aligned} \quad (2.2)$$

onde A e B são operadores diferenciais, b e t representam forças ou fontes externas atuando sobre o domínio Ω e sobre o contorno Γ_t , respectivamente. O valor u_p é o valor de u prescrito sobre Γ_u . O contorno Γ foi decomposto em um contorno Γ_t , chamado de Neumann ou forçado e um contorno Γ_u , denominado de Dirichlet ou essencial de modo que $\Gamma = \Gamma_t \cup \Gamma_u$.

A Figura 2.1 apresenta para um domínio bi-dimensional a representação do espaço da solução de um problema do tipo regido pelas equações (2.1) e (2.2).

Soluções analíticas são geralmente limitadas aos casos de configurações mais simples. A complexidade da maioria dos problemas (da natureza) leva à busca de soluções tratadas por métodos aproximados. Em engenharia um procedimento padrão é a transformação do meio contínuo em um meio discreto a partir do qual se estabelecem equações algébricas que permitem obter uma solução geral do problema com um número finito de parâmetros ou coeficientes incógnitos.

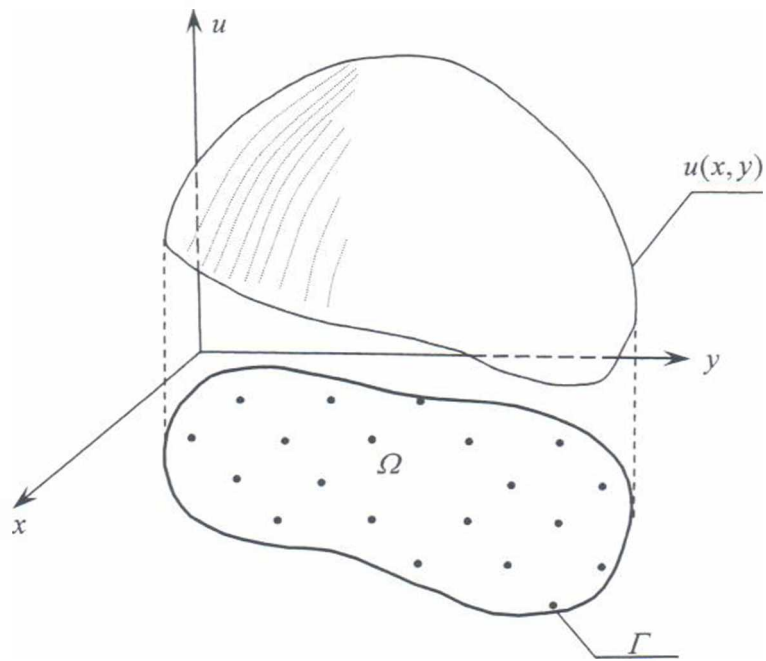


Figura 2.1 – Espaço da solução de um domínio bi-dimensional.

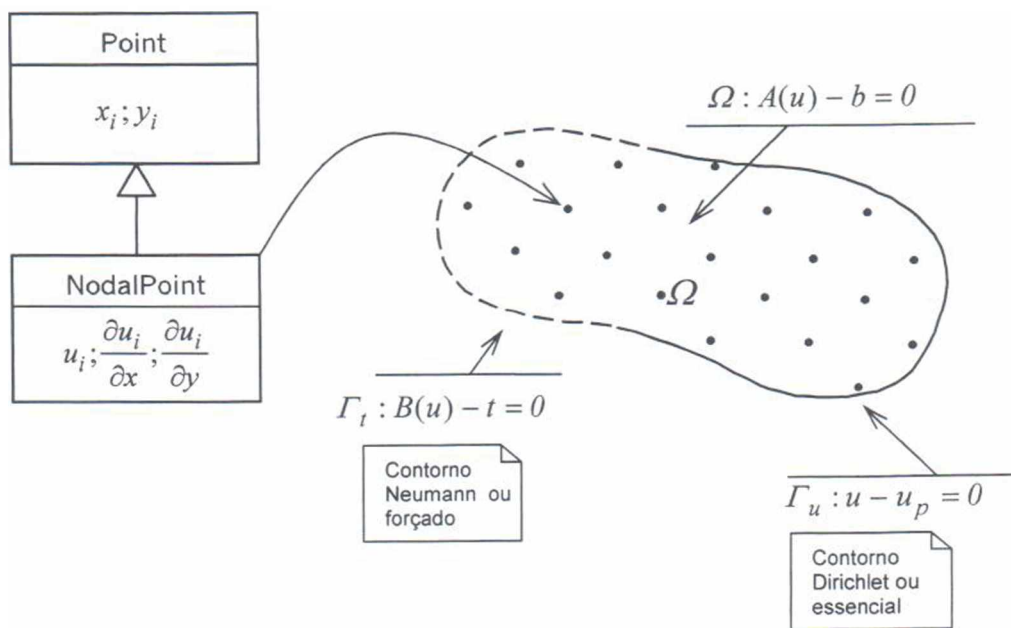


Figura 2.2 – Aproximação do meio contínuo por domínio discreto.

Na Figura 2.2 um domínio bi-dimensional contínuo foi substituído por um domínio discreto onde se atribui a cada *ponto geométrico* (x_i, y_i) propriedades do

problema tratado, como o valor da solução e de suas derivadas, chamado agora de *ponto nodal*.

A análise do problema discreto em pontos nodais e expresso pelas equações diferenciais (2.1) e condições de contorno (2.2), pode ser empreendida através das seguintes formulações (Felippa, 2000):

- **Forte** – abordagem direta das equações de governo do problema.
- **Integral** – transformação da formulação forte numa expressão integral média sobre o domínio,
- **Variacional** – formulação representada por um funcional cuja condição estacionária gera as formulações forte e integral.

Na *formulação forte* as equações diferenciais são aproximadas localmente por equações de diferenças. Este tipo de aproximação tende a magnificar erros locais, característica peculiar do processo de diferenciação (Hildebrand, 1956). Ao contrário, a *formulação integral* tende a suavizar de forma excessiva os erros locais. Neste caso é bastante comum o rebaixamento da ordem das equações diferenciais através de um procedimento de integração por partes, conduzindo à chamada *formulação integral fraca*. A abordagem através da *formulação variacional* permite a utilização do tratamento matemático rigoroso e poderoso do Cálculo Variacional. Infelizmente nem todos os problemas da mecânica são regidos por um princípio variacional conhecido.

Cada uma das formulações para análise do modelo matemático possui uma forma natural de método de discretização. Na formulação forte o *Método das Diferenças Finitas* substitui as derivadas por diferenças, obtidas por expansão em série de Taylor. Na formulação integral o *Método dos Resíduos Ponderados* define uma solução aproximada para todo o domínio. Em função da ponderação adotada surgem vários métodos: Colocação, Sub-Domínio, Mínimos Quadrados, Galerkin e Petrov-Galerkin. Na formulação variacional o *Método de Rayleigh-Ritz* é historicamente o primeiro método de discretização.

Na Figura 2.3 encontram-se representadas as três formulações com as possibilidades de transformação entre as mesmas e os métodos naturais de discretização. Apresenta ainda o *Método dos Elementos Finitos* que é uma aplicação local em elementos

do Método de Rayleigh-Ritz, na sua dedução por formulação variacional, podendo ser também apresentado através da variante Galerkin do Método dos Resíduos Ponderados.

Cabe mencionar, de modo a apresentar um quadro mais completo dos métodos de discretização, o Método dos Elementos de Contorno que usa uma formulação integro-diferencial, combinando no processo de discretização uma solução analítica com elementos finitos no contorno.

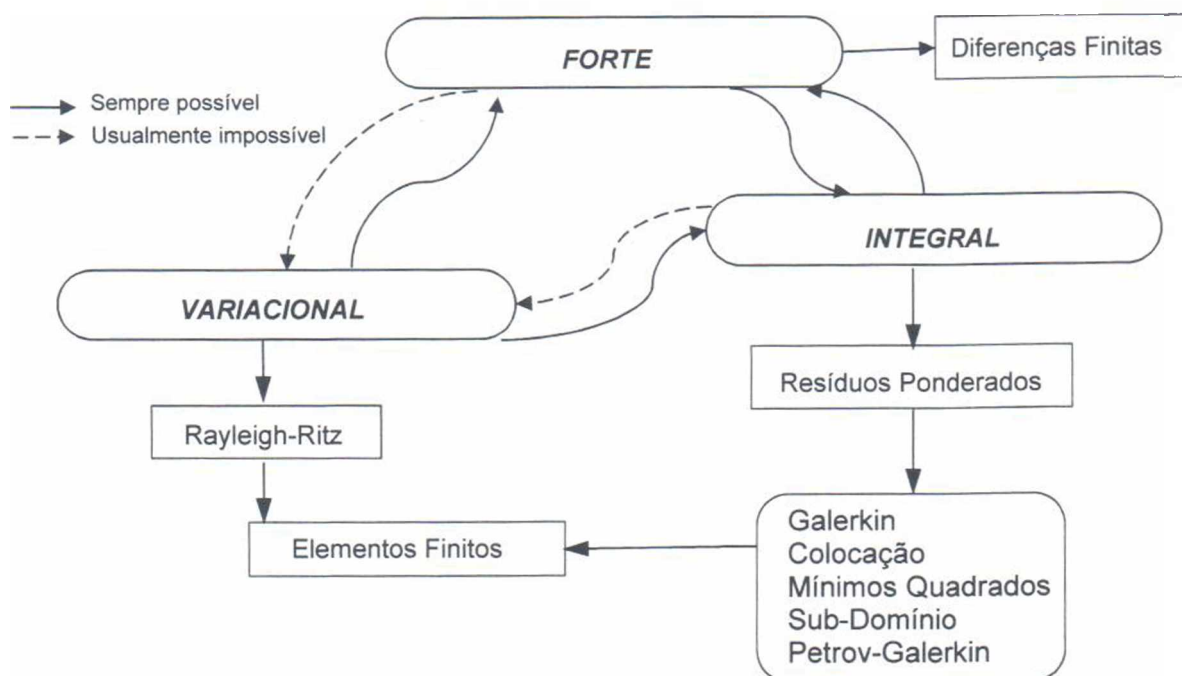


Figura 2.3 – Formulações forte, integral e variacional, e métodos de discretização.

Como será visto no Capítulo 4, os **Métodos sem Malha** utilizam a formulação integral fraca através de uma das subclasses do Método dos Resíduos Ponderados, distinguindo-se na forma da construção das funções locais de aproximação que prescindem de um elemento geometricamente simples como no caso do Método dos Elementos Finitos.

Neste trabalho será utilizado o Método dos Resíduos Ponderados como arcabouço matemático para geração dos Métodos sem Malha. Dado o seu caráter geral este método pode ser visto como superclasse computacional que permite através de subclasses especializadas implementar os principais métodos de discretização, além das diferentes especializações dos Métodos sem Malha, como mostrado na Figura 2.4.

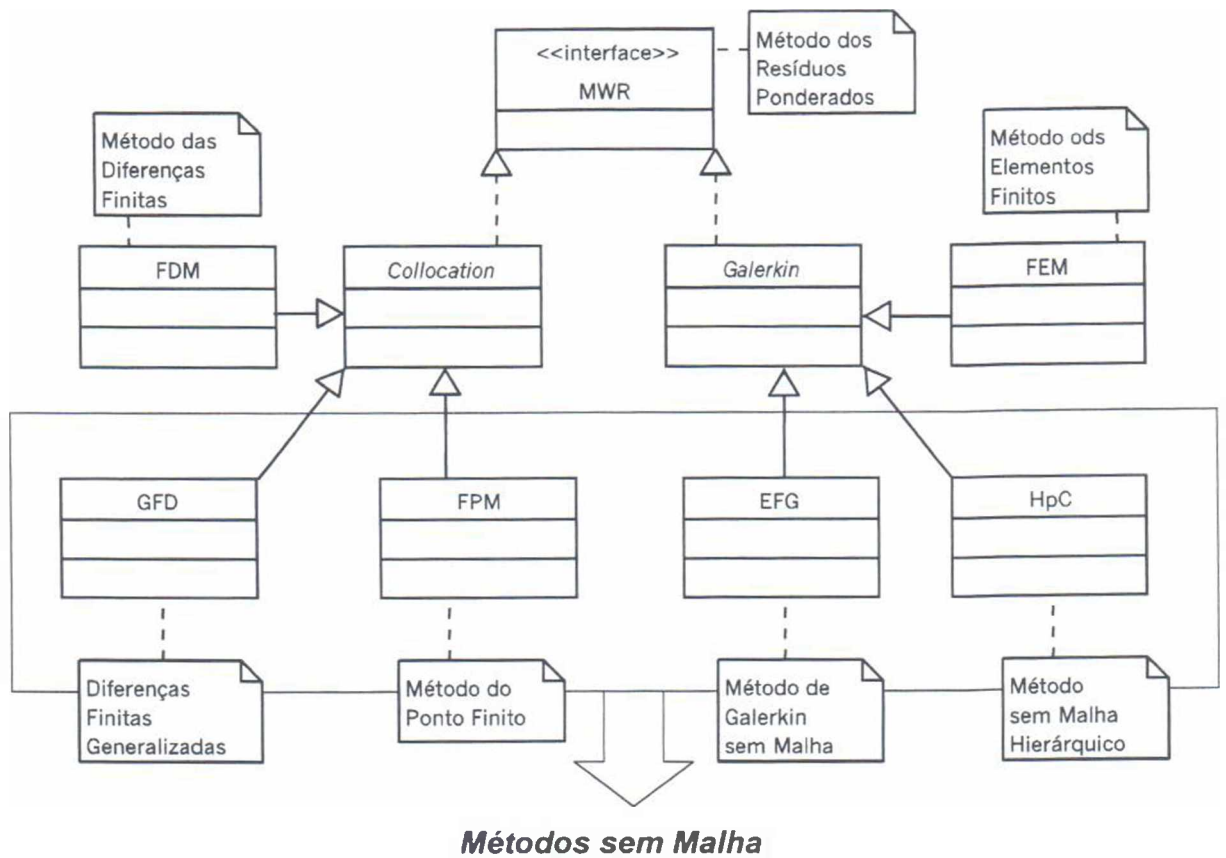


Figura 2.4 – Modelo de classes computacionais do Método dos Resíduos Ponderados.

2.2 - MÉTODO DOS RESÍDUOS PONDERADOS

De um modo geral, os métodos de aproximação buscam uma solução aproximada u^h para a solução exata u , através de uma seqüência na forma

$$u \doteq u^h = \sum_{i=1}^m \alpha_i \phi_i = \phi^T \alpha \quad (2.3)$$

onde α_i são coeficientes e ϕ_i são as funções base, que devem ser linearmente independentes e constituir uma seqüência completa.

Essas características matemáticas são de grande importância no processo de aproximação. Uma seqüência de funções é dita linearmente independente se $\alpha_1 \phi_1 + \alpha_2 \phi_2 + \dots + \alpha_m \phi_m = 0$, somente quando os α_i são zero. Assim, para cada nova função ϕ_i acrescida à solução u^h existirá um valor α_i diferente de zero. O emprego desta seqüência numa das formulações fraca, integral ou variacional é condição necessária para se obter um sistema de equações algébricas linearmente independente.

Uma seqüência linearmente independente é dita completa se um número m de constantes α_i pode ser encontrado, tal que:

$$\|u - u^h\| = \left\| u - \sum_{i=1}^m \alpha_i \phi_i \right\| < \varepsilon \quad (2.4)$$

onde $\|\cdot\|$ é uma norma e ε é uma quantidade pequena. Portanto, a seqüência no limite ($m \rightarrow \infty$) aproximará a função u . Ou seja, à medida que se agregam termos à seqüência se obtém uma melhor aproximação de u . Este critério de convergência estabelece, então, que dado um conjunto ϕ_i completo, uma função qualquer pode ser expandida em termos deste conjunto. Assim, u^h é inerentemente capaz de representar a solução exata u , desde que seja usado um número suficiente de termos.

A expressão (2.3) é conhecida na literatura como função tentativa, teste ou prova. A escolha dessa função atendendo a geometrias e a condições de contorno complexas é extremamente difícil, tendo sido sua utilização limitada a casos mais simples.

Levando-se a solução aproximada (2.3) ao sistema de equações diferenciais (2.1) resultará uma função erro ε , chamada de função residual

$$\varepsilon = [A(u^h) - b] \quad (2.5)$$

sendo ε igual a zero para a solução exata u .

Este resíduo pode ser forçado a zero por todo o domínio, em um sentido médio, fazendo-se a integral ponderada da função resíduo ε igual a zero

$$\int_{\Omega} W_j \cdot \varepsilon \, d\Omega = 0 \quad j = 1 \dots m \quad (2.6)$$

onde W_j é um conjunto de funções de ponderação.

Num sentido matemático, definindo o produto interno (a, b) por

$$(a, b) = \int_{\Omega} a \cdot b \, d\Omega \quad (2.7)$$

se $(a, b) = 0$, as funções a e b são ortogonais. Ou seja, a operação (2.6) procura parâmetros α_j , presentes em u^h , que ortogonalizem o resíduo com uma função de ponderação W_j . Uma função é zero se ela for ortogonal a cada membro de um conjunto completo de funções. Assim se W_j for um conjunto completo a função erro ε será nula no limite quando $m \rightarrow \infty$ (Connor e Brebbia, 1977).

O mesmo raciocínio se estende às condições de contorno, sendo então as equações de governo (2.1) e (2.2) substituídas pelas seguintes equações de resíduos

$$\int_{\Omega} W_j [A(u^h) - b] \, d\Omega + \int_{\Gamma_t} \bar{W}_j [B(u^h) - t] \, d\Gamma + \int_{\Gamma_u} \bar{\bar{W}}_j [u^h - u_p] \, d\Gamma = 0 \quad (2.8)$$

$$j = 1 \dots m$$

A expressão (2.8) constitui-se no *Método dos Resíduos Ponderados* que é o procedimento mais geral para a solução numérica do sistema de equações diferenciais que descrevem um dado problema físico (Zienkiewicz e Taylor, 1989).

Em geral empregam-se funções u^h e W_j que evitem que algum integrando se torne infinito, buscando-se utilizar funções finitas unívocas. Outra consideração importante refere-se à continuidade de u^h . Caso algum termo dos operadores $A()$ ou $B()$ possua derivada de ordem n , as funções de u^h deverão ter derivadas contínuas até a ordem $n - 1$, continuidade C^{n-1} . Pode-se verificar que esta condição garante um integrando finito.

A expressão (2.8) pode ter a ordem de suas derivadas diminuída nos operadores $A()$ e $B()$, aumentando-se a ordem das derivadas das funções de ponderação W_j , através de um procedimento de integração por partes. Com esta operação de relaxamento da ordem dos operadores necessita-se de uma continuidade de menor ordem para as funções de u^h e, em contrapartida, uma maior continuidade para as funções de ponderação W_j .

Esta operação bastante típica nos métodos de discretização é chamada de *formulação fraca* das equações diferenciais e de contorno que regem o problema, como será apresentado na Seção 2.3.

Vários critérios levaram à escolha de diferentes funções de ponderação W_j , quase todos desenvolvidos na primeira metade do Século XX. Crandall (1956), em seu "*Engineering Analysis*" unificou a consideração dos diferentes critérios como sendo o Método dos Resíduos Ponderados. Os critérios mais usuais são:

• $W_j = \delta(x - x_j)$	Colocação	
• $W_j = \begin{cases} 1 & \forall x \in \Omega_j \\ 0 & \forall x \notin \Omega_j \end{cases}$	Sub-domínio	
• $W_j = \frac{\partial \varepsilon}{\partial a_j}$	Mínimos Quadrados	(2.9)
• $W_j = \frac{\partial u^h}{\partial a_j} = N_j$	Galerkin	
• $W_j \neq N_j$	Petrov-Galerkin	

onde δ é o delta de Dirac e $\varepsilon = [A(u^h) - b]$ é o resíduo no domínio.

Finlayson (1972) investigou de forma bastante completa o Método dos Resíduos Ponderados. Levando em conta a grande diversidade de problemas analisados concluiu que a escolha do critério não é crucial, obtendo resultados similares especialmente para aproximações mais elevadas. Uma possível exceção ocorre em problemas que possuem princípio variacional associado. Neste caso o Método de Galerkin é preferido dado à sua equivalência com o método variacional de discretização (Rayleigh-Ritz). Na maioria dos casos a escolha do critério pode ser estabelecida por conveniência computacional.

O Método dos Elementos Finitos e o Método das Diferenças Finitas podem ser considerados instâncias particulares do Método dos Resíduos Ponderados, com as funções de ponderação W_j , \bar{W}_j e $\bar{\bar{W}}_j$ definidas de diferentes maneiras (Oñate *et al.*, 1996). Mesmo o Método dos Elementos de Contorno através de um processo de integração por partes pode ser deduzido a partir do Método dos Resíduos Ponderados (Zienkiewicz e Taylor, 1989; Brebbia e Dominguez, 1989). Na verdade, todos os procedimentos de aproximação sem malha se enquadram, também, como uma forma particular do Método dos Resíduos Ponderados.

2.3 - FORMA DISCRETA DO MÉTODO DOS RESÍDUOS PONDERADOS

Como já mencionado o emprego da expressão aproximada na forma $u^h = \phi^T a$ está limitado aos casos de geometrias e condições de contorno simples. A transformação do meio contínuo em um meio discreto, como na Figura 2.2, permite reescrever a seqüência em função de algum parâmetro nodal na forma

$$u^h = \sum_{i=1}^{n_p} N_i(x) a_i = N^T a \quad (2.10)$$

onde n_p é o número total de pontos nodais no domínio; a_i são parâmetros nodais, todos ou alguns incógnitos, tais como o valor da função $u(x_i, y_i)$ ou sua derivada $u_x(x_i, y_i)$, sendo (x_i, y_i) um ponto geométrico conhecido; e $N_i(x)$ são funções expressas em termos de variáveis independentes.

As funções $N_i(x)$ são conhecidas como *funções de forma* e determinadas nos métodos tradicionais de discretização (Método dos Elementos Finitos e Diferenças Finitas) a partir de uma geometria simples definidas com n pontos nodais, já nos Métodos sem Malha os pontos não são definidos por uma geometria. Em qualquer dos casos o valor de n adotado é bem menor que n_p .

De um modo geral pode-se expressar as funções de forma N_i em termos das funções de base ϕ_i . Para tal pode-se tomar n pontos nodais e escrevendo os parâmetros nodais a_i em termos das funções ϕ_i , calculadas nas posições nodais a_i , e dos coeficientes α_i , assim

$$\begin{Bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{Bmatrix} = \begin{bmatrix} \phi_1^1 & \phi_2^1 & \cdots & \phi_n^1 \\ \phi_1^2 & \phi_2^2 & \cdots & \phi_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ \phi_1^n & \phi_2^n & \cdots & \phi_n^n \end{bmatrix} \begin{Bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{Bmatrix} \quad (2.11)$$

ou simplesmente

$$\mathbf{a} = \mathbf{C} \boldsymbol{\alpha} \quad (2.12)$$

Como as funções base ϕ_i são linearmente independentes, e como o número de pontos nodais escolhidos n é adotado da mesma dimensão da base m , a matriz \mathbf{C} pode ser invertida e os coeficientes $\boldsymbol{\alpha}$ podem ser expressos em função dos parâmetros nodais \mathbf{a}

$$\boldsymbol{\alpha} = \mathbf{C}^{-1} \mathbf{a} \quad (2.13)$$

Levando-se em (2.3) obtém-se

$$u = \boldsymbol{\varphi}^T \boldsymbol{\alpha} = \boldsymbol{\varphi}^T \mathbf{C}^{-1} \mathbf{a} = \mathbf{N}^T \mathbf{a} \quad (2.14)$$

ou seja

$$\mathbf{N}^T = \boldsymbol{\varphi}^T \mathbf{C}^{-1} \quad (2.15)$$

como apresentado em (2.10).

Substituindo a aproximação a parâmetros nodais na forma (2.10) na expressão integral de resíduos (2.8) obtém-se

$$\begin{aligned} \int_{\Omega} \bar{W}_j [A(\mathbf{N}^T \mathbf{a}) - b] d\Omega + \int_{\Gamma_t} \bar{W}_j [B(\mathbf{N}^T \mathbf{a}) - t] d\Gamma + \\ + \int_{\Gamma_u} \bar{\bar{W}}_j [\mathbf{N}^T \mathbf{a} - u_p] d\Gamma = 0 \quad j = 1 \dots n_p \end{aligned} \quad (2.16)$$

Neste formato o número de equações algébricas resultante não é mais igual à dimensão da base m mas sim igual ao número de pontos nodais do domínio discretizado n_p .

A expressão de resíduos na forma (2.16) pode ter a ordem das derivadas dos operadores reduzida a partir da integração por partes da integral sobre o domínio. Esta operação conduz à chamada *formulação fraca* que além de reduzir a ordem dos operadores, que em decorrência passa a exigir uma menor ordem de continuidade das funções de forma N_j , incorpora outras propriedades à formulação discreta do Método dos

Resíduos Ponderados. Com freqüência a forma resultante é fisicamente mais coerente do que a expressão (2.16), uma vez que permite a descontinuidade das derivadas, impedida pela excessiva suavização decorrente da operação que zera o erro num sentido médio através da integração no domínio (Zienkiewicz e Taylor, 1989).

A obtenção da forma discreta da formulação fraca será ilustrada utilizando-se a equação de Poisson sobre um domínio bidimensional mostrado na Figura 2.5. Seja o problema de campo

$$\begin{aligned}
 A(u) &= \frac{\partial}{\partial x} \left(\rho \frac{\partial u}{\partial x} \right) + \frac{\partial}{\partial y} \left(\rho \frac{\partial u}{\partial y} \right) = b & \text{em } \Omega \\
 B(u) &= q = \rho \frac{\partial u}{\partial n} = t & \text{em } \Gamma_t \\
 u &= u_p & \text{em } \Gamma_u
 \end{aligned} \tag{2.17}$$

onde ρ , b , t , e u_p são constantes e $\partial u / \partial n$ é a derivada normal a Γ .

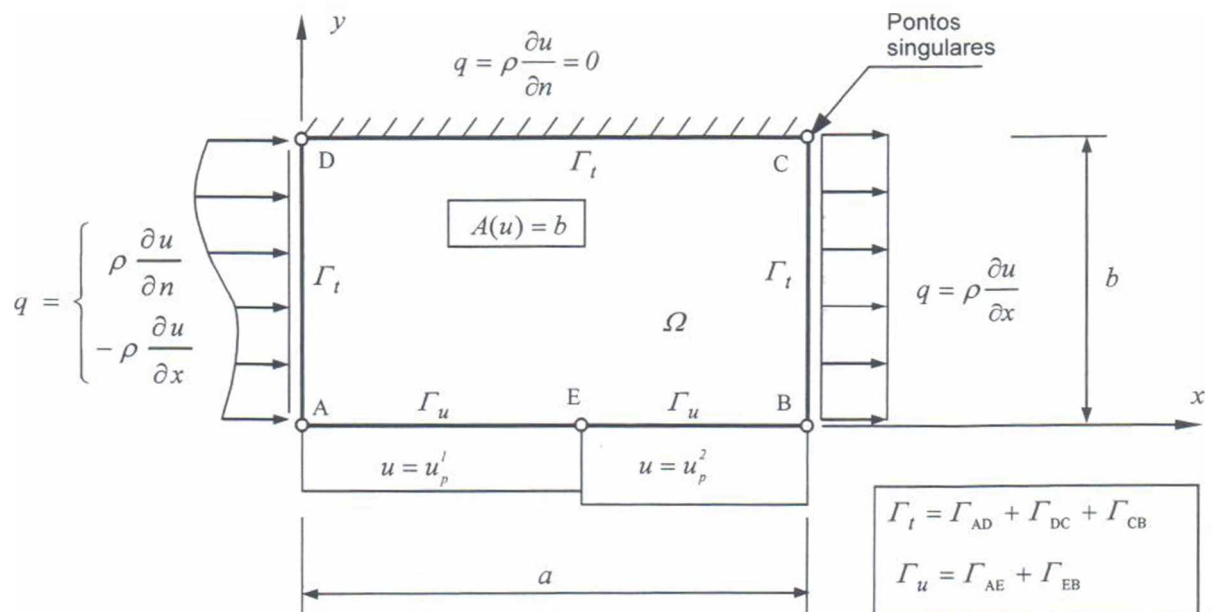


Figura 2.5 – Domínio bidimensional regido pela equação de Poisson.

Fazendo a integração ponderada do resíduo de cada equação em separado vem

$$\int_{\Omega} W_j \left[\frac{\partial}{\partial x} \left(\rho \frac{\partial u}{\partial x} \right) + \frac{\partial}{\partial y} \left(\rho \frac{\partial u}{\partial y} \right) - b \right] d\Omega = 0 \quad j = 1 \dots n_p \quad (2.18)$$

$$\int_{\Gamma_t} \bar{W}_j \left[\rho \frac{\partial u}{\partial n} - t \right] d\Gamma = 0 \quad j = 1 \dots n_p \quad (2.19)$$

$$\int_{\Gamma_u} \bar{\bar{W}}_j [u - u_p] d\Gamma = 0 \quad j = 1 \dots n_p \quad (2.20)$$

Integrando-se por partes o primeiro termo da integral sobre o domínio em (2.19) obtém-se

$$- \int_{\Omega} \left[\frac{\partial W_j}{\partial x} \left(\rho \frac{\partial u}{\partial x} \right) + \frac{\partial W_j}{\partial y} \left(\rho \frac{\partial u}{\partial y} \right) \right] d\Omega + \int_{\Gamma} W_j \left[\rho \frac{\partial u}{\partial n} \right] d\Gamma = 0 \quad (2.21)$$

Como as funções de ponderação são arbitrárias, será adotado $W_j = \bar{W}_j$. Levando (2.21) em (2.18) e subtraindo (2.19) as expressões resultantes são

$$\begin{aligned} & - \int_{\Omega} \left[\frac{\partial W_j}{\partial x} \left(\rho \frac{\partial u}{\partial x} \right) + \frac{\partial W_j}{\partial y} \left(\rho \frac{\partial u}{\partial y} \right) \right] d\Omega + \int_{\Gamma_u} W_j \left[\rho \frac{\partial u}{\partial n} \right] d\Gamma = \\ & = \int_{\Omega} W_j b d\Omega - \int_{\Gamma_t} W_j t d\Gamma \quad j = 1 \dots n_p \end{aligned} \quad (2.22)$$

$$\int_{\Gamma_u} \bar{\bar{W}}_j [u - u_p] d\Gamma = 0 \quad j = 1 \dots n_p \quad (2.23)$$

Como em Γ_u o valor de u é prescrito, verifica-se que a expressão $q_u = \rho \frac{\partial u}{\partial n}$, dita fluxo, será incógnita. Percebe-se então que no processo de enfraquecimento da continuidade na integral de resíduos o fluxo $\partial u / \partial n$ em Γ_u passou a integrar numa mesma expressão as incógnitas do problema.

Assim, adotando-se as formas discretas $u = N^T \mathbf{a}$, para a incógnita principal, $q_u = \bar{N}^T \mathbf{q}$ para a incógnita secundária e $u_p = \bar{N}^T \bar{\mathbf{u}}$, como ilustrado na Figura 2.6, e

substituindo-se em (2.22) e (2.23), chega-se ao seguinte sistema linear de equações algébricas

$$\begin{bmatrix} \mathbf{K} & \mathbf{H} \\ \mathbf{G}^T & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \mathbf{a} \\ \mathbf{q} \end{Bmatrix} = \begin{Bmatrix} \mathbf{F} \\ \mathbf{U} \end{Bmatrix} \quad (2.24)$$

onde

$$\mathbf{K} = - \int_{\Omega} \left[\frac{\partial W_j}{\partial x} \left(\rho \frac{\partial N^T}{\partial x} \right) + \frac{\partial W_j}{\partial y} \left(\rho \frac{\partial N^T}{\partial y} \right) \right] d\Omega \quad (2.25)$$

$$\mathbf{F} = \int_{\Omega} W_j b d\Omega - \int_{\Gamma_t} W_j t d\Gamma \quad (2.26)$$

$$\mathbf{H} = \int_{\Gamma_s} W_j \overline{N}^T d\Gamma \quad (2.27)$$

$$\mathbf{G} = \int_{\Gamma_u} \overline{\overline{W}}_j \overline{N}^T d\Gamma \quad (2.28)$$

$$\mathbf{U} = \int_{\Gamma_u} \overline{\overline{W}}_j \overline{N}^T u d\Gamma \quad (2.29)$$

Cabe observar que as funções de forma N_i devem possuir continuidade C^1 e as funções de forma \overline{N}_i devem possuir continuidade no mínimo de C^0 . Nada impede que se adote $\overline{N}_i = N_i$ com algum custo computacional extra.

Os métodos considerados neste trabalho, derivados do Método dos Resíduos Ponderados, são obtidos com a utilização de dois critérios: Colocação, onde $W_j = \delta(x - x_j)$ e Galerkin, onde $W_j = N_j$.

Aplicando o critério de Colocação às equações (2.25) a (2.29) e considerando-se as propriedades da função delta de Dirac (Sokolnikoff e Redheffer, 1966), que se comporta como uma função generalizada (Gel'Fand e Shilov, 1964), chega-se a

$$\mathbf{K} = \begin{cases} \frac{\partial}{\partial x} \left(\rho \frac{\partial \mathbf{N}^T}{\partial x} \right) + \frac{\partial}{\partial y} \left(\rho \frac{\partial \mathbf{N}^T}{\partial y} \right); & \text{para } j \in (\Omega - \Gamma) \\ \frac{\partial}{\partial x} \left(\rho \frac{\partial \mathbf{N}^T}{\partial x} \right) + \frac{\partial}{\partial y} \left(\rho \frac{\partial \mathbf{N}^T}{\partial y} \right) - \left(\rho \frac{\partial \mathbf{N}^T}{\partial x} n_x + \rho \frac{\partial \mathbf{N}^T}{\partial y} n_y \right); & \text{para } j \in \Gamma \end{cases} \quad (2.30)$$

$$\begin{aligned} \mathbf{F} &= b(x_j)|_{\Omega} - t(x_j)|_{\Gamma_u} & \mathbf{U} &= u_p(x_j)|_{\Gamma_u} \\ \mathbf{G} &= \mathbf{N}^T(x_j)|_{\Gamma_u} & \mathbf{H} &= \overline{\mathbf{N}^T}(x_j)|_{\Gamma_u} \end{aligned} \quad (2.31)$$

A matriz \mathbf{K} considera a descontinuidade que ocorre na função de ponderação delta de Dirac ao passar pelo contorno (Gel'Fand e Shilov, 1964). Essa expressão pode ser deduzida através de integração por partes aplicando-se as propriedades da função delta de Dirac ao final do procedimento.

O formato acima de aplicação do critério de Colocação difere da prática tradicional que impõe o critério na formulação forte da expressão de resíduos, quando não nas próprias equações diferenciais, e não na forma fraca como aqui. A exposição acima permite a inclusão dos pontos gêmeos no contorno e acrescenta o termo de contorno na matriz \mathbf{K} . Cabe lembrar que muitos procedimentos numéricos com critério de Colocação possuem flutuação numérica dos resultados em torno da solução exata que é estabilizada com termos adicionais quase empíricos (Oñate e Idelsohn, 1996). A formulação acima será relacionada como trabalho para estudos futuros.

Esta formulação permite uma abordagem dos *pontos singulares* do contorno, ou seja, pontos geométricos onde ocorrem variações dos valores conhecidos de u ou q no contorno e de mudança brusca da geometria. Nesses locais pode-se adotar dois pontos nodais distintos para a mesma coordenada geométrica (x_i, y_i) , em outras palavras *pontos "gêmeos"* como na Figura 2.6. Esses pontos são decorrência matemática da formulação integral fraca adotada. Na verdade esses pontos são os limites de integração das integrais do contorno.

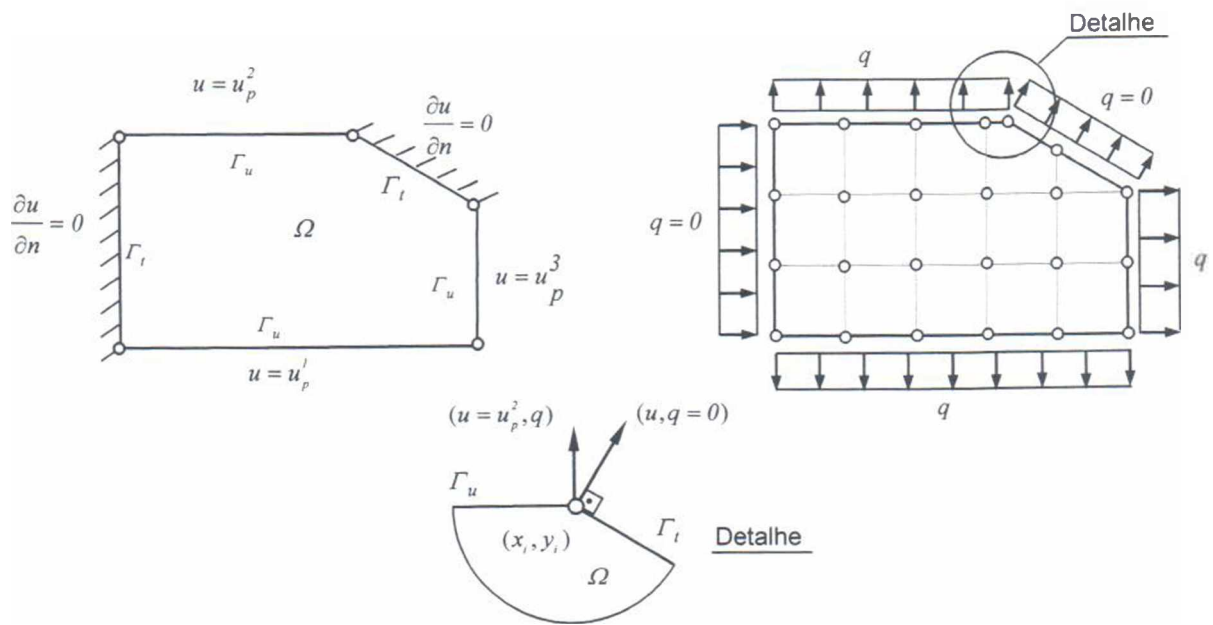


Figura 2.6 – Discretização do domínio em pontos nodais. Pontos nodais gêmeos em uma singularidade do contorno.

Para a aplicação do critério de Galerkin adotam-se as funções de ponderação $W_j = N_j$ e $\overline{W}_j = \overline{N}_j$ diretamente nas equações (2.25) a (2.29). As integrais resultantes são obtidas usualmente utilizando quadratura gaussiana nos moldes do realizado nos elementos finitos isoparamétricos.

As duas formulações apresentadas correspondentes aos critérios de Colocação e Galerkin, permitem estabelecer de forma bastante geral o sistema de equações algébricas dos métodos de discretização que serão detalhados no Capítulo 4. A dedução partiu de uma formulação fraca do Método dos Resíduos Ponderados que permite a utilização de funções de forma de ordem mais baixa, incorpora as condições de contorno essencial e forçada e surpreendentemente permite que a variável primária u possua derivada descontínua, possibilidade fisicamente compatível que a formulação forte (2.1) e (2.2) e mesmo a equação de resíduos original (2.8) não admitiam (Zienkiewicz e Taylor, 1989).

O procedimento ilustrado através da equação de Poisson é bastante geral, podendo ser utilizado em outros problemas como o de elasticidade bidimensional, como será visto no Capítulo 5.

2.4 - CARÁTER LOCAL DOS MÉTODOS DE DISCRETIZAÇÃO

A construção da expressão aproximada a parâmetros nodais

$$u^h = \sum_{i=1}^{n_p} N_i(x) a_i = \mathbf{N}^T \mathbf{a} \quad (2.32)$$

com contribuição de todos os pontos nodais n_p do domínio não considera que o valor da aproximação em um ponto geométrico (x_i, y_i) , $u^h(x_i, y_i)$, é mais influenciada pelos pontos nodais mais próximos desse ponto geométrico. Por outro lado o caráter global das funções de forma N_i na expressão (2.32) torna a sua determinação para diferentes geometria e condições de contorno bastante complexa. Para não mencionar que se a base de funções ϕ for composta de monômios, por exemplo, sistemas de ordem até mesmo pouco elevada conduzem a notória instabilidade numérica.

Desse modo as funções de forma N_i são construídas com um caráter local através da consideração de um número de pontos n muito menor que o número de pontos n_p de discretização do domínio, passando a possuir valores diferente de zero apenas em um subdomínio Ω_i de Ω , como na Figura 2.7.

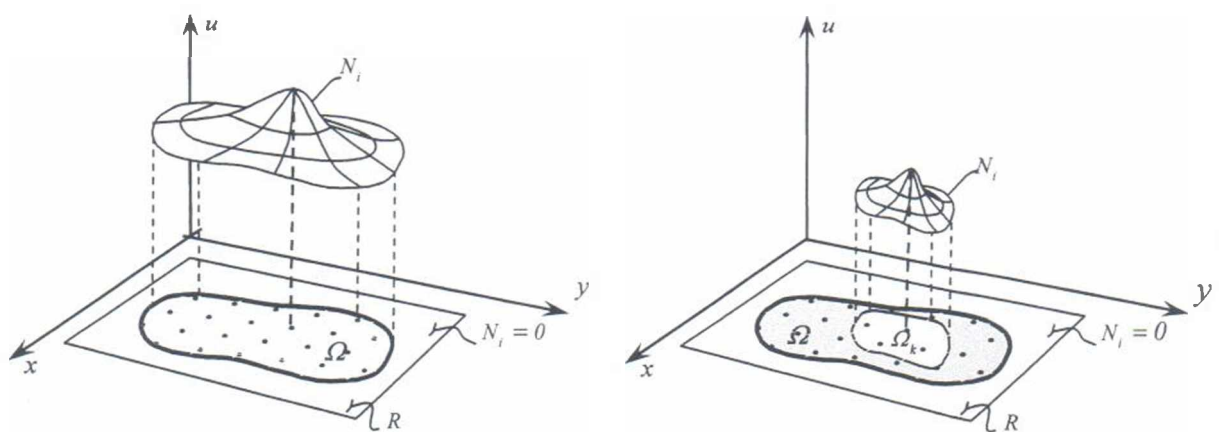


Figura 2.7 – Função de forma global sobre todo o domínio e função de forma local.

A consideração das funções de forma com caráter local no sistema de equações algébricas (2.24) do Método dos Resíduos Ponderados permite substituir as integrais por todo o domínio Ω pelo somatório das integrais dos subdomínios Ω_k , assim por exemplo a equação (2.25) torna-se

$$\begin{aligned} \mathbf{K} &= - \int_{\Omega} \left[\frac{\partial W_j}{\partial x} \left(\rho \frac{\partial \mathbf{N}^T}{\partial x} \right) + \frac{\partial W_j}{\partial y} \left(\rho \frac{\partial \mathbf{N}^T}{\partial y} \right) \right] d\Omega \\ &= - \sum_{k=1}^{n_{\Omega_k}} \int_{\Omega_k} \left[\frac{\partial W_j}{\partial x} \left(\rho \frac{\partial \mathbf{N}^T}{\partial x} \right) + \frac{\partial W_j}{\partial y} \left(\rho \frac{\partial \mathbf{N}^T}{\partial y} \right) \right] d\Omega \end{aligned} \quad (2.33)$$

onde n_{Ω_k} é o número de sub-domínios. Este procedimento estende-se às demais integrais.

Pode-se observar que neste formato as funções de ponderação W_j e de forma N_i terão muitos termos nulos, ou de outro modo, somente o conjunto de pontos nodais pertencentes ao subdomínio Ω_k possuirão valor diferente de zero. Assim pode-se definir uma matriz local \mathbf{K}_{Ω_k} cuja dimensão depende do número de pontos nodais em Ω_k e um operador de montagem (ou *assembler*) \mathbf{A}_k que mapeia os termos da matriz local \mathbf{K}_{Ω_k} do subconjunto de pontos nodais de Ω_k para a matriz \mathbf{K} dos n_{Ω} pontos nodais do domínio Ω , portanto

$$\mathbf{K} = \mathbf{A} \sum_{k=1}^{n_{\Omega_k}} \mathbf{K}_{\Omega_k} \quad (2.34)$$

Esse operador toma formas especializadas conforme o critério seja de Colocação ou Galerkin. No critério de Colocação \mathbf{K}_{Ω_k} toma a forma de um vetor linha e no critério de Galerkin \mathbf{K}_{Ω_k} é uma matriz. A Figura 2.8 ilustra o procedimento de montagem e as formas especializadas das classes computacionais **Assembler** que serão consideradas.

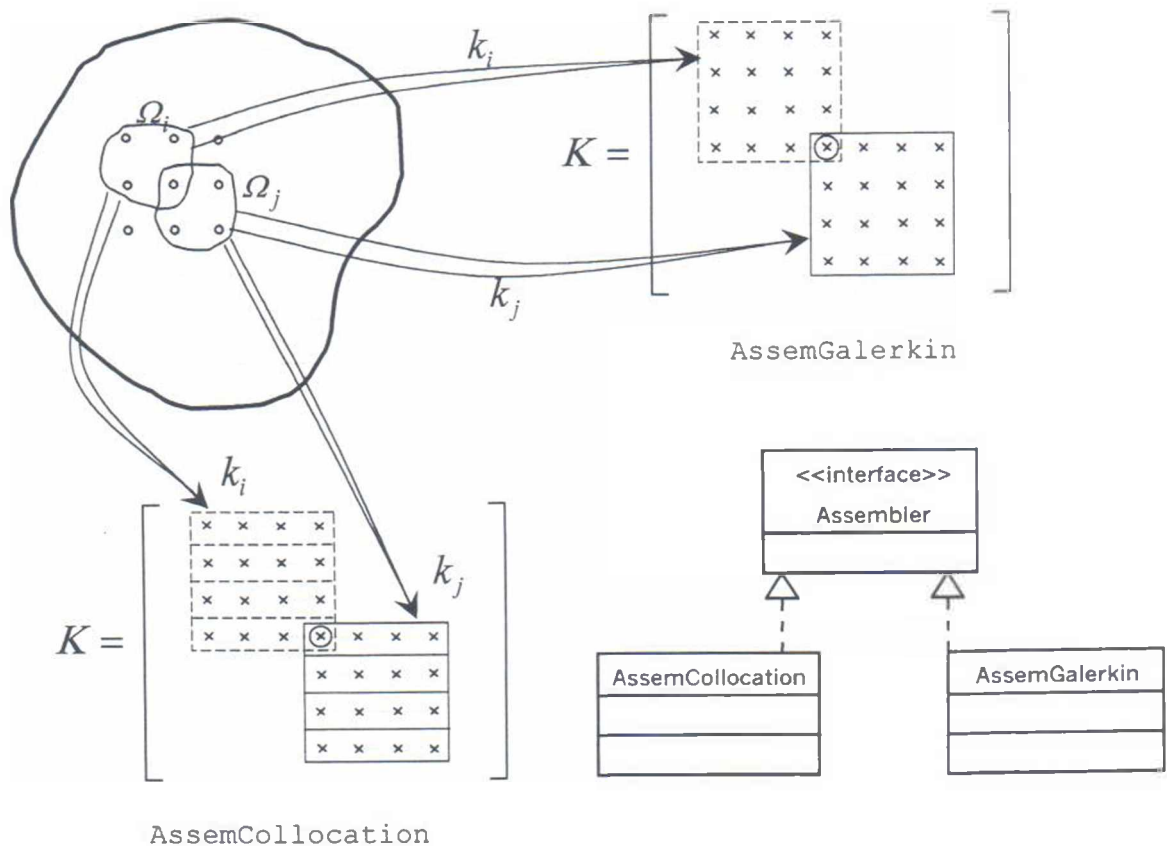


Figura 2.8 – Operação de montagem da contribuição local na matriz global com critério de colocação e Galerkin.

Como mostrado na Figura 2.8 um ponto nodal pode pertencer a mais do que um subdomínio existindo, neste caso, uma superposição das contribuições locais. Este acoplamento é bastante corrente nos métodos de discretização levando a uma concentração das contribuições em torno da diagonal num formato matricial em banda. Isto decorre do caráter local das funções de forma que não consideram a influência dos nós distantes sobre o ponto em consideração.

Em geral os métodos de discretização utilizam alguma forma de conectividade entre os nós do subdomínio Ω_i como forma de gerar as funções de forma N_i . Por exemplo, no método das Diferenças Finitas é utilizada uma conexão pontual denominada de molécula (ou *star*), no Método dos Elementos Finitos a conectividade entre os nós do subdomínio define o elemento. Este elemento que se confunde com o subdomínio é utilizado também como domínio de integração de K_{Ω_i} . Nos Métodos sem Malha as funções de forma são obtidas a partir dos nós pertencentes ao subdomínio sem a

consideração de nenhuma conectividade entre esses nós, tratados em seu conjunto, como nuvem (*cloud*). A Figura 2.9 ilustra.

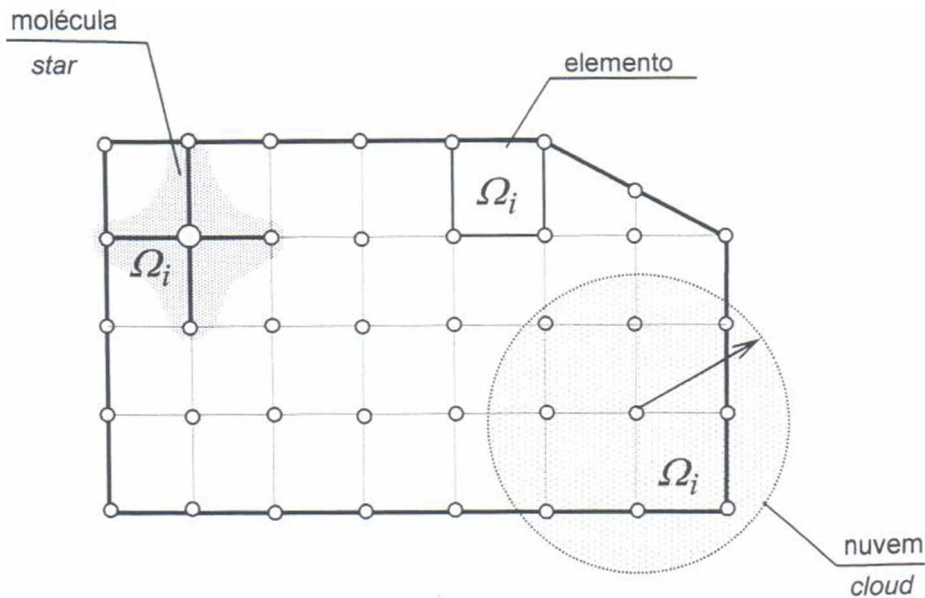


Figura 2.9 – Conectividade de pontos nodais por molécula, elemento e nuvem.

Na verdade o que se busca é a associação de um dado ponto do domínio com um conjunto de pontos nodais circunvizinhos de modo a permitir a construção de uma aproximação local. Assim pode-se dizer que tanto a organização em molécula como em elemento são casos particulares da organização por nuvem.

Na definição da matriz local K_{Ω_k} estão incorporados além do tipo de função de forma local N_i , e portanto da existência ou não de uma conectividade, o tipo de problema a ser resolvido (potencial, elasticidade etc), caracterizado pelo operador e propriedades incorporadas ao integrando de K_{Ω_k} . Assim, para cada problema a ser implementado computacionalmente, devem-se prover as diferentes matrizes locais que um determinado método requer.

2.5 - EXEMPLO DE APLICAÇÃO – BARRA ELÁSTICA

De modo a exemplificar a aplicação dos fundamentos teóricos de Método dos Resíduos Ponderados, acima expostos, será considerado o problema unidimensional da barra elástica submetida a esforço axial regido pelas equações

$$A(u) = \frac{d}{dx} \left(\rho \frac{du}{dx} \right) = -b \quad \text{para} \quad \Omega : 0 < x < L \quad (2.35)$$

$$B(u) = q = \rho \frac{dy}{dx} \Big|_{x=L} \quad n_x = Q_0 \quad \text{para} \quad \Gamma_t : x = L \quad (2.36)$$

$$u(0) = u_0 = 0 \quad \text{para} \quad \Gamma_u : x = 0 \quad (2.37)$$

onde $\rho = \rho(x)$, $b = b(x)$, u_0 , Q_0 são dados conhecidos e n_x é o co-seno diretor na direção normal. A Figura 2.10 ilustra um problema específico

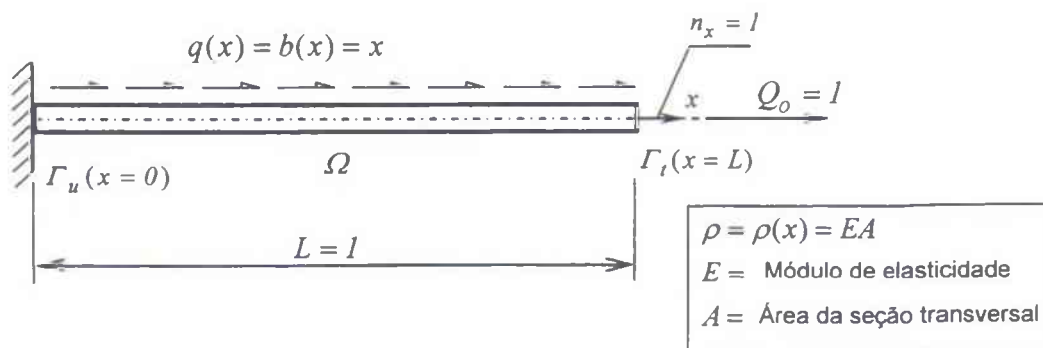


Figura 2.10 – Barra elástica submetida a esforço axial.

Outros problemas de valor de contorno elípticos de segunda ordem possuem formulação bastante semelhante à apresentada para a deformação axial de uma barra unidimensional (Reddy, 1993).

O problema será resolvido com a utilização do critério de Galerkin na formulação fraca desenvolvida na Seção 2.3. De forma bastante similar à equação de Poisson chega-se a

$$\begin{aligned}
 \mathbf{K} &= \int_{\Omega} N_{,x} \rho N_{,x}^T d\Omega & \mathbf{H} &= N \bar{N}^T \Big|_{\Gamma_u} & \mathbf{U} &= \bar{N}^T u_p \Big|_{\Gamma_u} \\
 \mathbf{F} &= \int_{\Omega} N b d\Omega + N Q_0 \Big|_{\Gamma_t} & \mathbf{G} &= \bar{N} N^T \Big|_{\Gamma_u}
 \end{aligned}
 \tag{2.38}$$

onde $N_{,x} = \frac{d}{dx} N$.

A função \bar{N} no contorno pontual Γ_u pode ser escolhida de modo relativamente arbitrário. No caso pode-se adotar como constante ou mais especificamente será adotado $\bar{N} = 1$. Na escolha das funções de forma N serão adotadas funções com caráter local, em subdomínios, como apresentado na Seção 2.4. A Figura 2.11 mostra as funções lineares com valor unitário no ponto nodal adotadas, ou seja $N_i(x_j) = 1$ para $i = j$ e $N_i(x_j) = 0$ para $i \neq j$. Portanto, com a chamada propriedade delta de Kronecker.

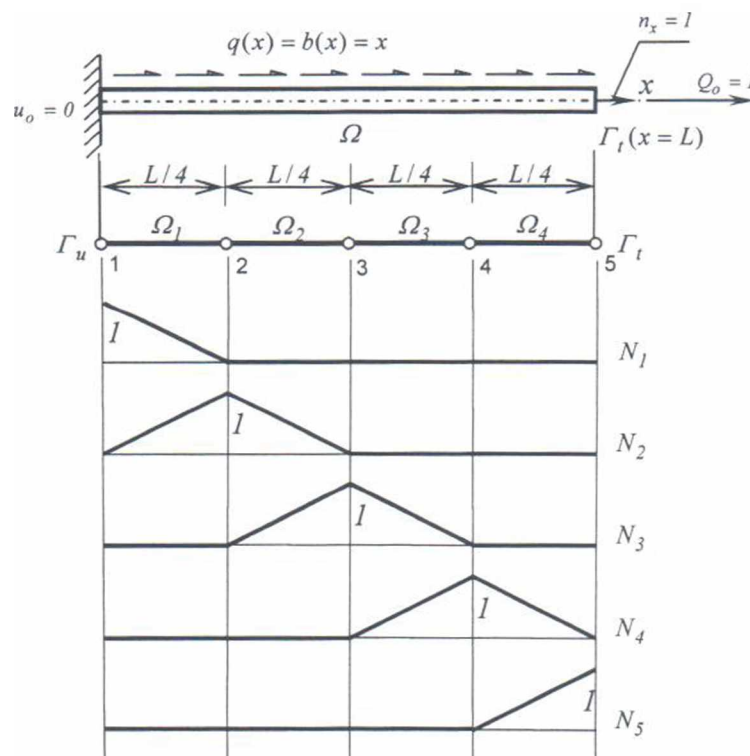


Figura 2.11 – Funções de forma lineares definidas localmente.

Ao aplicar os dados do problema e efetuando as integrações, bastante simples, chega-se

$$\mathbf{K} = \begin{bmatrix} 4 & -4 & 0 & 0 & 0 \\ -4 & 8 & -4 & 0 & 0 \\ 0 & -4 & 8 & -4 & 0 \\ 0 & 0 & -4 & 8 & -4 \\ 0 & 0 & 0 & -4 & 4 \end{bmatrix} \quad \mathbf{F} = \begin{bmatrix} 1/96 \\ 1/16 \\ 1/8 \\ 3/16 \\ 11/96 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad \mathbf{H} = \mathbf{G} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \mathbf{U} = \{0\} \quad (2.39)$$

Deve-se observar na equação (2.24) que a montagem de \mathbf{K} considerou as matrizes \mathbf{K}_{Ω} e sua superposição. Verifica-se também a característica de matriz em banda. Levando ao sistema (2.24), ou seja montado a equação matricial

$$\begin{bmatrix} \mathbf{K} & \mathbf{H} \\ \mathbf{G}^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{a} \\ \mathbf{q} \end{bmatrix} = \begin{bmatrix} \mathbf{F} \\ \mathbf{U} \end{bmatrix} \quad (2.40)$$

e resolvendo obtém-se

$$\mathbf{a}^T = \{ 0.0000 \quad 0.3724 \quad 0.7292 \quad 1.0547 \quad 1.3333 \} \quad (2.41)$$

$$\mathbf{q} = \{ 1.500 \} \quad (2.42)$$

A Figura 2.12 apresenta a solução analítica e a solução discreta determinada acima. Percebe-se que a variável primária u obtida possui bom ajustamento com a equação analítica. Já o esforço axial, variável secundária du/dx , apresenta o resultado escalonado devido à continuidade C^1 empregada nas funções de forma N_i .

Devido ao tipo de funções de forma escolhidas verifica-se que $a|_{r_u} = \bar{u}$, desta forma o sistema pode ser simplificado e resolvendo-se para os pontos nodais incógnitos o sistema (2.40) transforma-se

$$\begin{bmatrix} K_{11} & 0 \\ K_{21} & 1 \end{bmatrix} \begin{bmatrix} a_1 \\ q \end{bmatrix} = \begin{bmatrix} f_1 - K_{12}a_2 \\ f_2 - K_{22}a_2 \end{bmatrix} \quad (2.43)$$

onde a_i se refere aos pontos nodais incógnitos, a_j aos pontos nodais conhecidos em Γ_u . Como o sistema (2.43) está desacoplado e K_{11} é simétrica positiva definida chega-se

$$\{a_i\} = [K_{11}]^{-1} \{f_i - K_{12} a_j\} \quad (2.44)$$

$$\{q\} = \{f_2\} - \{K_{22} a_2\} - \{K_{21} a_1\} \quad (2.45)$$

Ou seja, nas condições específicas consideradas é possível resolver apenas o sistema com pontos nodais onde o parâmetro correspondente à variável primária u seja incógnita (Bathe, 1996; Cook *et al.*, 1989). Na verdade o exemplo acima apresenta a formulação Galerkin do Método dos Elementos Finitos.

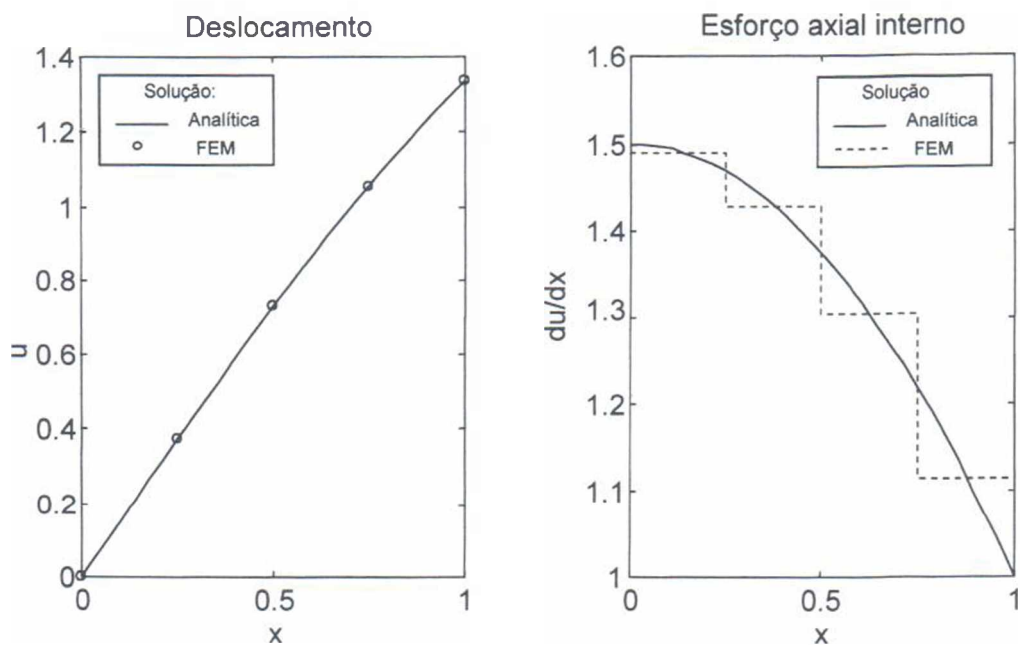


Figura 2.12 – Soluções analítica e discreta da barra elástica.

2.6 - DECOMPOSIÇÃO POR OBJETOS COMPUTACIONAIS DO MÉTODO DOS RESÍDUOS PONDERADOS

Nas seções anteriores foram apresentados os aspectos mais relevantes da modelação matemática, utilizando-se o Método dos Resíduos Ponderados, de problemas de valor de contorno elípticos. A abordagem apresentada propicia um arcabouço matemático geral para a dedução de diversos métodos de discretização do contínuo. O alto grau de abstração é um dos requisitos para a modelagem computacional orientada a objetos.

Assim a Figura 2.13 procura representar de forma bastante genérica a descrição geométrica de um domínio, suas condições de contorno e propriedades, bem como um modelo de discretização através de pontos nodais representando o modelo físico. Apresenta-se ainda uma representação do modelo matemático através da equação integral ponderada de resíduos, do sistema algébrico de equações lineares e da expressão de montagem da aproximação local no sistema global.

Um modelo computacional constituindo-se em um projeto orientado a objeto do Método dos Resíduos Ponderados está proposto através dos pacotes de classes apresentados, também, na Figura 2.13. Um *pacote* (*package*) agrupa diversas classes relacionadas permitindo uma decomposição organizada do modelo computacional. Na visão orientada a objetos os pacotes permitem a divisão de um sistema em unidades de alto nível reunindo as classes correlacionadas.

O pacote **Domain** permite criar um objeto que descreve as características geométricas do problema. Como será visto no Capítulo 6 a definição da geometria parte de uma abordagem de CAD (*Computer Aided Design*).

O pacote **Discret** modela a discretização do domínio, ou do objeto criado de **Domain**, para os diversos métodos implementados. Este pacote possui classes para a discretização em pontos nodais do contorno, e do domínio, e para discretização em células de integração (ou elementos). O seu detalhamento será apresentado no Capítulo 6.

O pacote **Problem** possui duas responsabilidades. A primeira é a modelação dos dados de um problema, tais como constantes, e funções que representam as fontes e as condições de contorno. A outra atribuição é a de fornecer as matrizes locais para a

montagem do sistema de equações lineares que caracterizam o Método dos Resíduos Ponderados. O Capítulo 5 apresenta o projeto deste pacote.

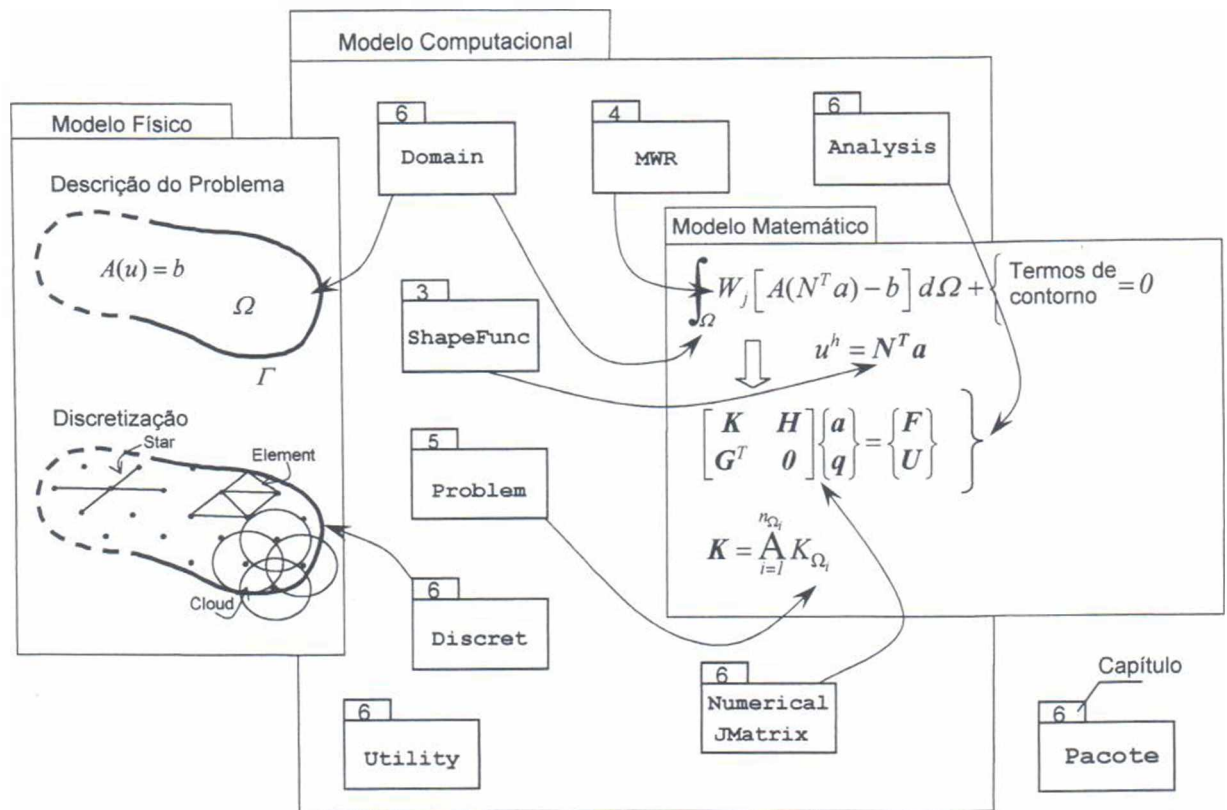


Figura 2.13 –Modelo Físico, Modelo Matemático e Modelo Computacional.

O pacote **MWR** possui classes para a definição dos dados de cada método de discretização implementado. Assim estas classes serão um container dos objetos que compõem um método, tais como qual tipo de função de forma, procedimento de integração se for o caso, dentre outros. Recebem também os objetos que caracterizam o domínio, a discretização e o problema, agindo como diretores que coordenarão a montagem do sistema de equações algébricas. Estas classes serão detalhadas nos Capítulos 4 e 6.

O pacote **ShapeFunc** possui classes para definição das funções de forma. Permite a definição de classes para os Métodos sem Malha e do Método dos Elementos Finitos. Para os Métodos sem Malha o pacote possui classes para definição da base de funções, da função de ponderação e da estratégia de formação das nuvens de pontos nodais. O detalhamento do pacote encontra-se no Capítulo 3.

O pacote **Analysis** concentra as classes e principalmente os métodos que possibilitam a construção de algoritmos para solução das equações algébricas. Das diferentes modelações existentes na literatura das classes de análise, uma das mais simples, extensíveis e robustas é a implementação feita por Zienkiewicz e Taylor (1989) através de macro-instruções. No Capítulo 6 a classe **Analysis** proposta é uma releitura orientada a objetos dessa abordagem.

O pacote **Numerical** reúne as classes que implementam os métodos numéricos básicos, tais como integração gaussiana, solução de sistemas de equações algébricas, bem como os objetos fundamentais na descrição de métodos de discretização que são os objetos vetores e matrizes. Neste caso se utiliza o pacote **JMatrix** (Sánchez e Pulino Filho, 2003).

2.7 - COMENTÁRIOS FINAIS

Na direção do desenvolvimento de uma estrutura computacional orientada a objetos que facilite a criação de programas de computador para problemas modelados pelos Métodos sem Malha optou-se como ponto de partida teórico a formulação matemática do Método dos Resíduos Ponderados. Do ponto de vista computacional esta abordagem permite projetar um conjunto de componentes que facilitarão a implementação das diferentes variantes dos Métodos sem Malha, como, também a implementação de outros métodos, como por exemplo, o Método dos Elementos Finitos.

A abordagem do critério de colocação em uma formulação fraca não foi observada na literatura. Como será visto no Capítulo 4 os métodos de colocação não seguem na direção apresentada. Um estudo mais aprofundado pode conduzir a um novo Método sem Malha.

A decomposição em objetos computacionais proposta visa a criação de um projeto (*design*) que possa ser ampliado e estendido de forma consistente e robusta. Não se constrói um aplicativo, mas um conjunto de classes que podem ser reaproveitadas e expandidas para a construção de aplicativos específicos. Busca-se, então, a construção de uma **estrutura computacional** (*framework*) que é um conjunto de classes que cooperam entre si e que constroem um projeto reutilizável para uma área de conhecimento específica (Gamma *et al.*, 2000). Uma estrutura computacional de classes que facilite a implementação de métodos e problemas discretizados por Métodos sem Malha.

3 - TÉCNICAS DE APROXIMAÇÃO E INTERPOLAÇÃO. FUNÇÕES DE FORMA

3.1 - INTRODUÇÃO

Na Seção 2.4 foi apresentada uma importante característica dos métodos de discretização que é o caráter local da aproximação da solução, conferido pela adoção de funções de forma N com valor diferente de zero num subdomínio Ω_i de Ω com um número de parâmetros nodais n_{Ω_i} , geralmente, bem menor que o número de pontos nodais n_p do domínio. Assim

$$\begin{aligned} N_i(x) &\neq 0 && \text{se } x \in \Omega_i \\ N_i(x) &= 0 && \text{se } x \notin \Omega_i \end{aligned} \quad (3.1)$$

Tanto no Método dos Elementos Finitos quanto no Método das Diferenças Finitas os pontos nodais do sub-domínio Ω_i são conectados numa estrutura geométrica previamente definida, chamada malha. Nos métodos ditos sem malha os pontos nodais são dispersos, como nuvem (*cloud*), sobre cada sub-domínio Ω_i sem nenhuma interconexão prévia.

Funções de aproximação baseadas num conjunto de pontos nodais dispersos e sem interconexão seguem na literatura três linhas de investigação. A mais antiga utiliza a aproximação de núcleo (*kernel*) para a aproximação $u^h(x)$ sobre o domínio Ω na forma

$$u^h(x) = \int_{\Omega} w(x-y, h) u(y) d\Omega_y \quad (3.2)$$

onde $w(x-y, h)$ é o núcleo (*kernel*) ou função peso e h é uma medida da dimensão do suporte, no qual $w(x-y)$ é diferente de zero (Lucy, 1977 *apud* Belytschko *et al.*, 1994).

As aproximações de Métodos sem Malha podem ser baseadas também em partições da unidade (Babuška e Melenk, 1997), que se constitui em um paradigma no qual

um domínio é coberto por sub-domínios Ω_i , cada um associado com uma função ϕ_i que é diferente de zero em Ω_i e possui a seguinte propriedade

$$\sum_i \phi_i(x) = 1 \text{ em } \Omega. \quad (3.3)$$

Neste caso a função de aproximação toma a seguinte forma

$$u^h(x) = \sum_i \phi_i(x) \left(u_i + \sum_{j=1}^m b_{j,i}(x) \right) \quad (3.4)$$

onde q_i é uma base de funções, geralmente polinomiais, e $b_{j,i}$ são coeficientes.

A terceira forma de se gerar funções de aproximação de Métodos sem Malha é a utilização do método de mínimos quadrados ponderados móvel. Lancaster e Šalkauskas (1981), utilizaram esta técnica na sua forma bi-dimensional, para a construção de superfícies interpoladas a partir de dados dispersos.

A técnica consiste na minimização do funcional do quadrado do erro ponderado na forma

$$E_x(u) = \sum_{i=1}^N w_i(x) (u(x_i) - u_i)^2 \quad (3.5)$$

onde $w_i(x) \equiv w(|x - x_i|)$ é uma função de ponderação positiva que decresce monotonicamente com o crescimento de $|x - x_i|$. A função de aproximação é

$$u(x) = \sum_{j=1}^m a_j(x) b_j(x) \quad (3.6)$$

onde $b_j(x)$ é uma base de funções linearmente independentes. Geralmente b_j são monômios.

Como se observa na equação (3.6) os coeficientes $a_j(x)$ são função da posição. Portanto, na construção da superfície aproximada será necessária a minimização

do funcional (3.5) para cada ponto x onde se pretenda obter o valor da função $u^h(x)$. Na prática a solução com valores pequenos de m (de 0 a 2) conduz a bons resultados. A superfície assim gerada pode ser descrita como um ajustamento de *mínimos quadrados ponderados "móvel"*.

Além do artigo de 1981, referência obrigatória na literatura dos Métodos de Discretização sem Malha, Lancaster e Šalkauskas publicaram, em 1986, livro bastante didático sobre ajustamento de curvas e superfícies.

Neste trabalho será utilizada como técnica para determinação de funções de aproximação o método de Lancaster e Šalkauskas de mínimos quadrados móvel, ficando o estudo dos Métodos de Discretização sem Malha limitado àqueles que utilizam esta técnica na geração de funções de forma.

3.2 - FUNÇÕES DE FORMA PELO MÉTODO DE MÍNIMOS QUADRADOS MÓVEL

A formulação a seguir busca a notação mais comum usada pelos diversos pesquisadores dos Métodos sem Malha.

Seja $u(x)$ uma função definida num domínio Ω . A função $u(x)$ pode ser aproximada através da combinação linear de funções linearmente independentes p_i de uma seqüência completa, na forma

$$u(x) \doteq u^h(x) = \sum_{i=1}^m p_i a_i = \mathbf{p}^T \mathbf{a}, \quad (3.7)$$

onde a_i são coeficientes. Geralmente a base de funções $\{p_i\}$ é adotada como sendo monômios em virtude das facilidades inerentes à análise numérica com polinômios. Para um problema 2D, por exemplo, a base linear é dada por

$$\mathbf{p}^T = [1 \quad x \quad y], \quad m = 3 \quad (3.8)$$

e a base quadrática por

$$\mathbf{p}^T = [1 \quad x \quad y \quad x^2 \quad xy \quad y^2], \quad m = 6 \quad (3.9)$$

onde m define a dimensão de cada base. A Seção 3.4 detalha a utilização da base de funções no âmbito do método dos mínimos quadrados.

A determinação dos coeficientes pode ser feita tomando-se um conjunto de n pontos nodais com coordenadas $x_i, i = 1 \dots n$ e valores nodais u_i . Substituindo em (3.7) a seguinte relação aparece

$$u_i = \mathbf{p}^T(x_i) \mathbf{a} = \mathbf{p}_i^T \mathbf{a}, \quad i = 1 \dots n \quad (3.10)$$

ou numa organização matricial

$$\mathbf{u} = \mathbf{P} \mathbf{a} \quad (3.11)$$

onde

$$\mathbf{u}^T = [u_1 \quad u_2 \quad \cdots \quad u_n]$$

$$\mathbf{P} = \begin{bmatrix} p_1(x_1) & \cdots & p_m(x_1) \\ \vdots & & \\ p_1(x_n) & \cdots & p_m(x_n) \end{bmatrix} \quad (3.12)$$

Caso o número de termos da base (dimensão) seja igual ao número de pontos nodais do sub-domínio, ou seja $m = n$, o sistema (3.12) pode ser resolvido, impondo-se certas condições à organização geométrica dos pontos nodais, obtendo-se assim os coeficientes a_i , que substituídos em (3.7) possibilitam relacionar a função de aproximação $u^h(x)$ com os valores nodais u_i , ou seja

$$\mathbf{a} = \mathbf{P}^{-1} \mathbf{u}$$

$$u^h(x) = \mathbf{p}^T(x) \mathbf{P}^{-1} \mathbf{u} \quad (3.13)$$

$$u^h(x) = \mathbf{N}^T(x) \mathbf{u}$$

onde $\mathbf{N}^T(x) = \mathbf{p}^T(x) \mathbf{P}^{-1}$ são as chamadas *funções de forma*. Se o conjunto de pontos $\{x_i\}$ delimitar um sub-domínio Ω_e , cuja geometria seja convexa, chega-se à definição de funções de forma do Método dos Elementos Finitos.

Se ao invés de um conjunto de pontos organizados geometricamente cujo número, n , é igual à dimensão da base, m , o conjunto de pontos for muito maior que a dimensão da base e disperso num sub-domínio Ω_e , como nuvem, a definição da função de aproximação deverá ser feita com base em algum critério de ajustamento, por exemplo, de minimização do funcional de quadrados do erro entre a função de aproximação e o valor da função

$$E(u) = \sum_{i=1}^n (u^h(x_i) - u_i)^2 \quad (3.14)$$

De um modo geral pode-se supor que alguns termos do funcional (3.14) possuem maior importância relativa no processo de otimização. Considerando fatores de ponderação $w_i > 0$, o funcional será agora

$$E(u) = \sum_{i=1}^n w_i (u^h(x_i) - u_i)^2 \quad (3.15)$$

Os valores u^h num ponto x devem ser mais influenciados pelos valores de u_i relativos a pontos x_i mais próximos de x . Isto implica que os pesos w_i dependem de x e devem possuir a propriedade de decrescer em magnitude à medida que a distância de x a x_i aumenta, ou seja, $w_i(x) \equiv w_i(x - x_i)$. Ao se estabelecer a dependência de w_i com x , o funcional (3.15) passa também a depender de x , fazendo com que os coeficientes da função de aproximação também dependam da posição x . Assim o funcional torna-se

$$E(u, x) = \sum_{i=1}^n w_i(x) (u^h(x_i, x) - u_i)^2 \quad (3.16)$$

com

$$u^h(x_i, x) = \mathbf{p}^T(x_i) \mathbf{a}(x) \quad (3.17)$$

Efetuada a substituição e organizando matricialmente o funcional assume a forma

$$E(u, x) = (\mathbf{P}\mathbf{a} - \mathbf{u})^T \mathbf{W}(x) (\mathbf{P}\mathbf{a} - \mathbf{u}) \quad (3.18)$$

onde $\mathbf{W}(x) = \text{diag}[w_1(x - x_1) \quad w_2(x - x_2) \quad \cdots \quad w_n(x - x_n)]$.

A função de ponderação $w_i(x) \equiv w_i(x - x_i)$ possui papel decisivo na formulação do método. Caso o seu valor seja diferente de zero apenas em um sub-domínio Ω_x de Ω , o número de pontos será função da posição x , portanto $n = n(x)$. Esse conjunto de pontos na vizinhança de x para os quais $w_i(x) \neq 0$, formam o seu domínio de influência, ou *suporte*, também chamado de *nuvem*.

Normalmente se consideram funções de ponderação que dependam somente da distância entre dois pontos, ou seja,

$$w_i(x - x_i) = w_i(d) \quad (3.19)$$

onde $d = \|x - x_i\|$ é a distância entre os pontos x e x_i .

A função de ponderação mais utilizada é a seguinte função gaussiana esférica truncada

$$w_i(x_i, x) = w_i(r) = \frac{\exp(-\beta^2 r^2) - \exp(-\beta^2)}{1 - \exp(-\beta^2)} \quad (3.20)$$

Na expressão acima, $r = d_i / dm_i = \|x - x_i\| / dm_i$, onde dm_i é a dimensão do suporte de w_i . O parâmetro β afeta a forma do sino da função de forma, e é usualmente adotado como sendo $\beta = 4$, como será visto na Seção 3.4.

A escolha da função ponderação é bastante arbitrária, outras funções são reportadas na literatura, como detalhado na Seção 3.4.

O processo de minimização do funcional (3.18) deverá ser feito para cada ponto x em que se deseja o valor da aproximação $u^h(x)$, daí a referência a Método dos Mínimos Quadrados Ponderados Móvel.

Efetuando a minimização $\partial E / \partial a(x) = 0$, chega-se ao conjunto de equações ditas normais, por produzirem uma ortogonalização entre os termos da base adotada

$$\begin{aligned} \frac{\partial E}{\partial a} &= P^T W(x) P a(x) - P^T W(x) u = 0 \\ &= A(x) a(x) - B(x) u = 0 \end{aligned} \quad (3.21)$$

onde

$$\begin{aligned} A(x) &= P^T W(x) P \\ B(x) &= P^T W(x) \end{aligned} \quad (3.22)$$

Resolvendo (3.21) chega-se

$$\mathbf{a}(x) = \mathbf{A}^{-1}(x) \mathbf{B}(x) \mathbf{u} \quad (3.23)$$

Para que a matriz \mathbf{A} possa ser invertida deve-se garantir, além de $w_i > 0$, que o número de pontos dentro da área de influência de x_i , vale dizer, dentro do suporte de definição de w_i , seja maior ou igual que a dimensão da base de funções. De modo a prevenir um mal condicionamento de \mathbf{A} adota-se, geralmente, um número bem maior de pontos do que a dimensão da base, quer dizer, $n \gg m$

Finalmente, a função de aproximação com este critério e condições passa a ser:

$$\begin{aligned} \mathbf{u}^h(x) &= \mathbf{p}^T(x) \mathbf{A}^{-1}(x) \mathbf{B}(x) \mathbf{u} \\ &= \boldsymbol{\phi}^T(x) \mathbf{u} \end{aligned} \quad (3.24)$$

onde

$$\boldsymbol{\phi}^T(x) = \mathbf{p}^T(x) \mathbf{A}^{-1}(x) \mathbf{B}(x) \quad (3.25)$$

são as funções de forma para um conjunto disperso de pontos (nuvem) em um subdomínio Ω_i .

A aproximação de Mínimos Quadrados Móvel acima depende substancialmente da maneira como se aplica a função de ponderação (Oñate *et al.*, 1996). Basicamente essa ponderação pode ser efetuada de duas maneiras.

Inicialmente admite-se a existência de um conjunto de pontos nodais, x_i , em um domínio, Ω . Para um ponto x_k qualquer, móvel sobre o domínio, determina-se o suporte ou sub-domínio Ω_k , função de algum critério como será visto na Seção 3.4, que garanta a existência da solução das equações normais, via de regra que garantam um número mínimo de pontos nodais dentro de Ω_k . A obtenção do valor da função de forma centrada em x_k será feita com a utilização dos valores da função de ponderação em cada ponto x_i , $W_k(x_i)$. A Figura 3.1 ilustra.

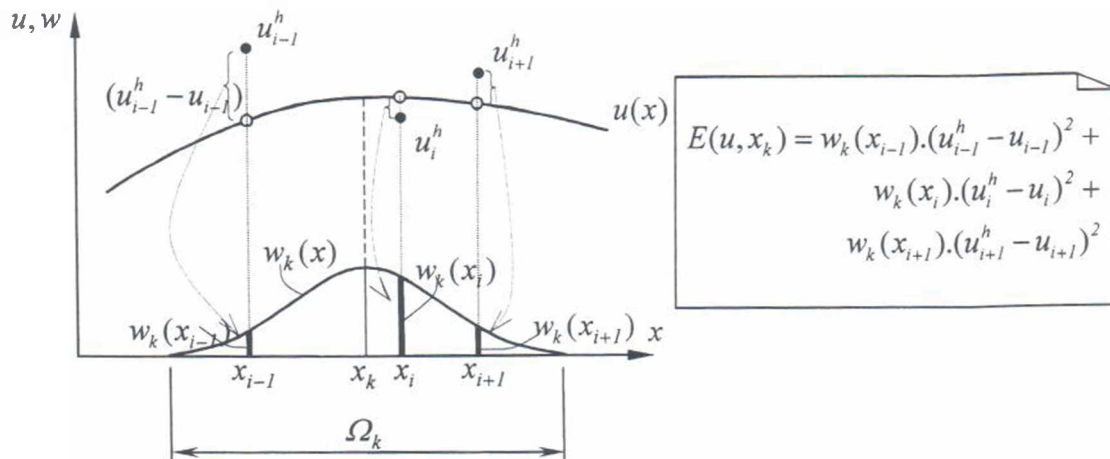


Figura 3.1 - Ponderação com suporte definido a partir de um ponto qualquer.

Esta abordagem é a adotada no Método sem Malha denominado de Método Galerkin sem Malha (*Element Free Galerkin*) como será visto no Capítulo 4. O caso particular que ocorre quando o ponto x_k , pólo da função de forma, coincide com um ponto nodal x_i é utilizado para a obtenção das funções de forma de dois Métodos sem Malha, o Método das Diferenças Finitas Generalizadas (*Generalized Finite Difference*) e o Método do Ponto Finito (*Finite Point Method*), também apresentados no Capítulo 4.

A outra possibilidade de ponderação é construída a partir da definição para cada ponto nodal x_i de um suporte Ω_i , escolhido de sorte que garanta a existência de solução das equações normais e em seu conjunto recobrendo o domínio Ω . A ponderação do quadrado das diferenças entre a função aproximada $u^h(x_i)$ e a função exata u_i é feita com a utilização do respectivo valor da função de ponderação determinado em um ponto móvel qualquer x_k , como representado na Figura 3.2.

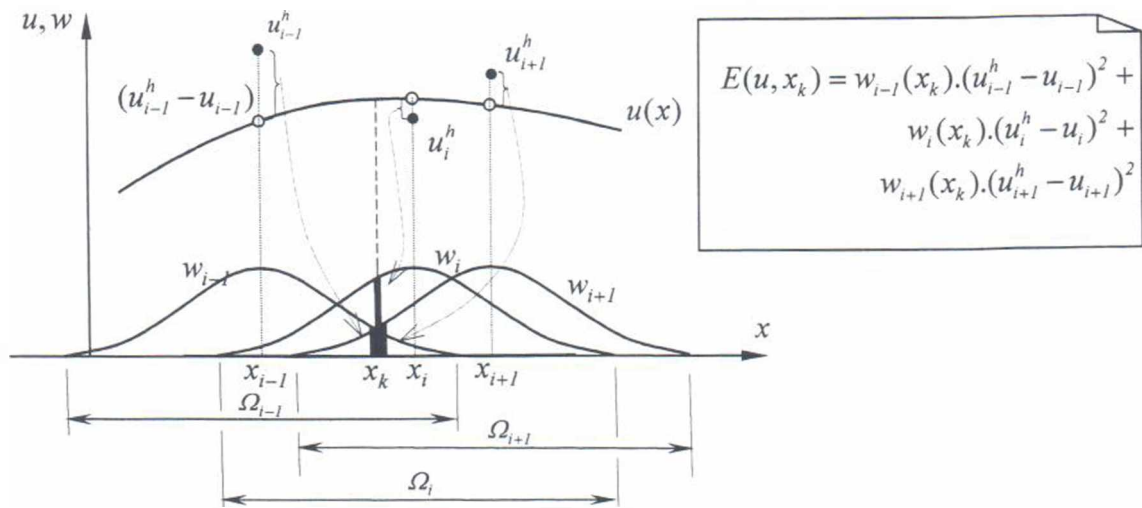


Figura 3.2 - Ponderação com suporte definido para cada ponto nodal.

Este enfoque de ponderação é o cerne do Método sem Malha Hierárquico (*Hp-Clouds*), apresentado no Capítulo 4. Outra utilização importante deste procedimento é no traçado da curva da função de forma centrada em um dado ponto nodal x_i . Vale enfatizar que as funções de forma oriundas do método dos mínimos quadrados móvel não possuem expressões analíticas, sendo o seu traçado obtido por exercício numérico. Neste caso o ponto x_k varre o domínio sendo feita a minimização do funcional para os pontos que se encontram no suporte do ponto nodal em consideração. As curvas do traçado das funções de forma apresentadas na Seção 3.4 foram obtidas com este procedimento.

As derivadas das funções de forma podem ser obtidas operando a equação (3.25). Assim a derivada em relação a s será dada por

$$\phi_s^T(x) = p_s^T(x) A^{-1} B + p^T \left(A_s^{-1} B + A^{-1} B_s \right) \quad (3.26)$$

onde as derivadas das matrizes valem

$$\begin{aligned} A_s &= P^T W_s(x) P \\ B_s &= P^T W_s(x) \end{aligned} \quad (3.27)$$

A derivada da inversa da matriz A pode ser obtida derivando-se ambos os membros da definição de matriz inversa $A A^{-1} = I$. Operando chega-se

$$A_s^{-1} = -A^{-1} A_s A^{-1} \quad (3.28)$$

A derivada da função de forma tomando uma segunda direção t é dada por

$$\begin{aligned} \phi_{st}^T &= p_{st}^T A^{-1} B + p_s^T (A_t^{-1} B + A^{-1} B_t) \\ &+ p_t^T (A_s^{-1} B + A^{-1} B_s) \\ &+ p^T (A_{st}^{-1} B + A_s^{-1} B_t + A_t^{-1} B_s + A^{-1} B_{st}) \end{aligned} \quad (3.29)$$

com

$$\begin{aligned} A_{st} &= P^T W_{st}(x) P \\ B_{st} &= P^T W_{st}(x) \end{aligned} \quad (3.30)$$

A matriz A_{st}^{-1} é obtida diferenciando-se expressão (3.28) com relação a t

$$A_{st}^{-1} = -A^{-1} [A_{st} A^{-1} + A_s A_t^{-1} + A_t A_s^{-1}] \quad (3.31)$$

Derivadas de ordem mais elevada podem ser obtidas por diferenciação sucessiva. Ficando a ordem de diferenciação limitada pela grau de continuidade da função de ponderação $w_i(x)$. As expressões deduzidas para as derivadas dependem da inversão da matriz A . As inversas das derivadas de A ficaram expressas em função de A^{-1} somente. Uma alternativa computacionalmente mais eficiente, não apresentada aqui, para a determinação das funções de forma e derivadas é obtida adotando-se um procedimento de decomposição tipo LU da matriz A e fazendo-se retro-substituições com termos independentes que permitem encontrar ϕ^T , ϕ_s^T e ϕ_{st}^T .

As funções de forma geradas pelo método de mínimos quadrados móvel apresentam características de reprodutividade e continuidade. Ela pode reproduzir qualquer das funções de sua base. Assim, caso a função constante unitária seja a única função incluída na base, as funções de forma são uma partição da unidade. Estas funções conhecidas como funções de Shepard tem a forma

$$\phi_i = \frac{W_i(x)}{\sum_{j=1}^n W_j(x)} \quad (3.32)$$

com a partição da unidade atendida por

$$\sum_{i=1}^n \phi_i(x) = 1 \quad (3.33)$$

Reproduz também as funções x e y caso incluídas, condição importante para se garantir convergência.

A outra propriedade mencionada é a alta continuidade obtida que depende somente do grau de continuidade da função de ponderação.

3.3 - BASE DE FUNÇÕES

Do exposto na seção anterior depreende-se que a escolha da base de funções é fundamental na definição de uma função aproximada $u^h(x)$ para uma função exata $u(x)$. Esta base de funções deve conter funções linearmente independentes de uma seqüência completa. As classes de funções mais encontradas são os monômios $\{x^i\}$ que geram aproximações polinomiais; senos e co-senos $\{\sin kx, \cos kx\}$ que geram funções de Fourier, e a classe $\{e^{ikx}\}$ que gera funções exponenciais.

No âmbito dos Métodos sem Malha a base de funções mais utilizada são monômios como já frisado. Assim para problemas unidimensionais utiliza-se a base

$$\mathbf{p}^T = [1 \quad x \quad x^2 \quad \dots \quad x^{m-1}] \quad (3.34)$$

onde m é a dimensão da base.

Para problemas bidimensionais a base pode ser representada pelo triângulo de Pascal. Para uma base de ordem quadrática, por exemplo, a base terá uma dimensão $m = 6$ e é dada por

$$\mathbf{p}^T = [1 \quad x \quad y \quad x^2 \quad xy \quad y^2] \quad (3.35)$$

Como mencionado acima, outras funções podem compor a base, selecionadas a partir da análise de problemas específicos de modo a acelerar a convergência do problema. Fleming *et al.* (1997) utilizaram em problemas de fratura em elasticidade estática bidimensional a seguinte base

$$\mathbf{p}^T = [1 \quad x \quad y \quad \sqrt{r} \cos \theta / 2 \quad \sqrt{r} \sin \theta / 2 \\ \sqrt{r} \sin \theta / 2 \sin \theta \quad \sqrt{r} \cos \theta / 2 \sin \theta] \quad (3.36)$$

com dimensão $m = 7$.

Outro exemplo de enriquecimento da base de funções foi realizado por Suleau *et al.* (2000), que utilizou funções seno e co-seno para análise de problemas regidos pela

equação de Helmholtz. Para problema de propagação de onda em uma dimensão empregou a seguinte base com $m = 3$

$$\mathbf{p}^T = [1 \quad \cos kx \quad \sin kx] \quad (3.37)$$

Lu *et al.* (1994) com o objetivo de evitar a solução do sistema de equações de dimensão m para cada ponto onde se deseja a minimização do funcional no método dos mínimos quadrados móvel, propuseram a utilização de uma base de funções ponderadas ortogonais obtida com a utilização do processo de ortogonalização de Smith. A base de funções \mathbf{p}^T é ortogonalizada com as funções de ponderação W . Este procedimento pode trazer alguma vantagem para sistemas de equações algébricas de ordem mais elevada que o usual.

3.4 - FUNÇÕES DE PONDERAÇÃO

Como apresentado na Seção 3.2 e frisado na Seção 3.3 a função de ponderação w_i possui papel fundamental na formulação do Método dos Mínimos Quadrados Ponderados Móvel. Por definição a função de ponderação deve atender

$$\text{i) } w_i(x) > 0 \quad \forall x;$$

$$\text{ii) } w_i \text{ deve ter decrescimento monotônico com o crescimento de } |x|;$$

Isto em conjunto com a base de funções linearmente independente garante que a matriz $A(x) = P^T W(x) P$ seja positiva definida e portanto inversível.

Outro efeito importante é a possibilidade de interpolação obtida caso as funções de ponderação atendam à condição

$$\text{iii) } \begin{cases} w_i \rightarrow \infty & \text{se } x \rightarrow 0, \text{ ou} \\ w_i(|x - x_i|) \rightarrow \infty & \text{se } x \rightarrow x_i \end{cases}$$

Esta característica não é bastante explorada nos Métodos sem Malha de um modo geral. Uma exceção é o Método das Diferenças Finitas Generalizadas onde a interpolação é fundamental, como será visto no Capítulo 4.

O decrescimento monotônico sugere que os pontos mais distantes são considerados de forma menos significativa. Deste modo deve-se definir uma função de ponderação truncada que atenda

$$\text{iv) } \begin{cases} w_i(d) > 0 & \text{para } d \leq dm_i \\ w_i(d) = 0 & \text{para } d > dm_i \end{cases}$$

onde $d = \|x - x_i\|$ é a distância entre os pontos x e x_i , e dm_i é a dimensão do suporte da função de ponderação e que determina o domínio de influência de x_i . Deste modo o número de pontos n dentro do domínio Ω , pode ser variável para cada ponto x_i . Contudo o número de pontos n deve ser no mínimo maior que a dimensão da base de funções m empregada, de modo a garantir que as equações normais tenham solução única. Assim,

$$v) n \in \Omega_i \quad | \quad n > m$$

Esta característica é fundamental nos Métodos sem Malha, pois através da definição do suporte se obtém um conjunto de pontos, ou vice-versa, que permite prescindir da tradicional conectividade nodal. Ou, de outra forma, pode-se dizer que a conectividade é estabelecida matematicamente, ou mais precisamente através de geometria computacional, a partir do suporte da função de ponderação.

A forma do suporte é em geral circular, para duas dimensões, podendo ser também retangular. Neste caso as funções de ponderação são geradas utilizando-se produto tensorial

$$vi) w(\mathbf{x} - \mathbf{x}_i) = w(x - x_i) \cdot w(y - y_i)$$

Estas funções de ponderações podem ser vantajosas para arranjos regulares de pontos nodais.

Outro aspecto importante na escolha de uma função de ponderação é a ordem de diferenciabilidade que se espera das funções de forma a serem obtidas. Lancaster e Šalkauskas (1981) demonstram que se a base de funções é de continuidade C^l e a função de ponderação é de classe C^m então as funções de forma serão de classe C^k , onde $k = \min\{l, m\}$. Na maior parte das vezes a ordem da base de funções é que controla a diferenciabilidade das funções de forma.

De um modo geral a escolha da função de ponderação é arbitrária desde que atenda às características listadas acima, mormente que seja uma função positiva e contínua com derivadas até o grau desejado. Na literatura encontram-se propostas diversas funções de ponderação aplicadas aos Métodos sem Malha. A Tabela 3.1 apresenta um resumo das principais funções encontradas com sua expressão e respectiva bibliografia.

A Figura 3.4 apresenta a influência das funções de ponderação no procedimento de aproximação resultante do Método dos Mínimos Quadrado Ponderados Móvel, como exemplificado por Beissel e Belytschko (1996). Utilizou-se a base de funções monomial linear, ou seja $\mathbf{p}^T = [1 \quad x]$. Para o caso da função de ponderação constante com suporte incluindo todo o domínio, a aproximação se reduz ao Método dos Mínimos

Quadrados tradicional, onde não se observa o efeito da ponderação. A utilização de uma função de ponderação na forma de degrau com suporte igual á metade do domínio, de forma que o número de nós no suporte fique sempre igual à dimensão da base adotada, ou seja $n = m = 2$, induz que o Método dos Mínimos Quadrados Móvel produza uma interpolação polinomial contínua por partes, equivalente à resultante do Método dos Elementos Finitos. A terceira função de ponderação, a Spline Quártica (vide Tabela 3.1) com suporte compacto e com derivadas, primeira e segunda contínuas, produz uma curva de ajustamento que aproxima e não interpola os pontos de dados.

A função de ponderação mais utilizada na literatura dos Métodos sem Malha, como mencionado na Seção 3.2, proposta inicialmente por Belytschko *et al.* (1994), é a função gaussiana com suporte circular (Figura 3.3)

$$w(r) = \begin{cases} \frac{\exp(-\beta^2 r^2) - \exp(-\beta^2)}{1 - \exp(-\beta^2)} & 0 \leq r \leq 1 \\ 0 & r > 1 \end{cases} \quad (3.38)$$

onde $r = \|x - x_i\| / dm_i$, com dm_i sendo o raio de suporte circular, no caso de duas dimensões, e β um parâmetro que afeta a forma de sino da função.

A função gaussiana apresenta uma descontinuidade na derivada quando $r = 1$ (Häussler-Combe e Korn, 1998), ou seja $d = \|x - x_i\| = dm_i$, neste caso

$$\left. \frac{dw(r)}{dr} \right|_{r=1} = -2\beta^2 \frac{e^{-\beta^2}}{1 - e^{-\beta^2}} \quad (3.39)$$

A referida descontinuidade é bastante reduzida para $\beta = 4$, como representado na Figura 3.5, sendo esse o valor comumente utilizado.

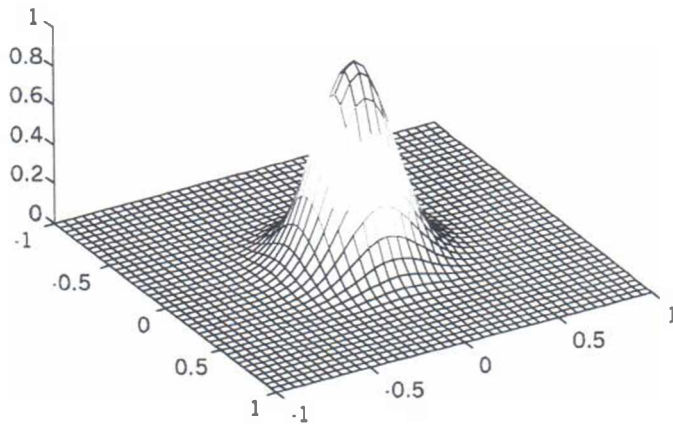


Figura 3.3 - Função gaussiana truncada.

A função gaussiana com $\beta = 4$ apresenta um decaimento muito pronunciado, por outro lado a definição do raio de suporte dm_i deve incluir um número adequado de pontos nodais que façam com que a matriz do sistema de equações normais seja bem condicionada. Assim, será adotado o seguinte critério para definição do raio de suporte

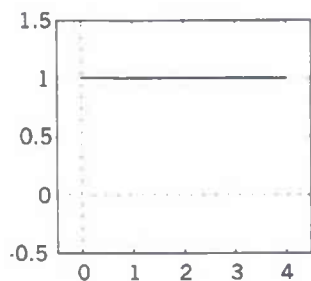
$$dm_i \geq d_{\min} + \alpha h \quad (3.40)$$

onde

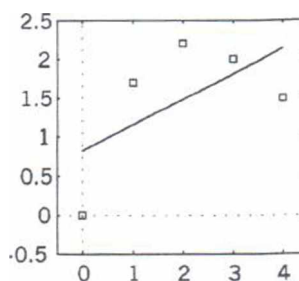
$$d_{\min} = \max_{j \in S_j} \|x_j - x_i\| \quad (3.41)$$

sendo S_j o conjunto dos m (dimensão da base) pontos nodais mais próximos de x_i , h o espaçamento nodal e α um coeficiente que controla o número de pontos nodais que serão englobados por dm_i .

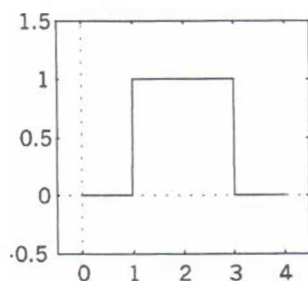
Como pode ser constatado no exemplo unidimensional da Figura 3.6, um valor de $\alpha \approx 1$ a 2 conduz a um sistema de equações normais mais bem condicionado.



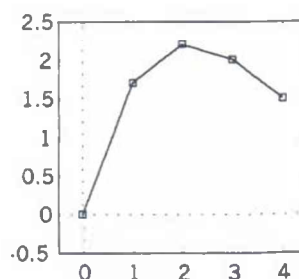
Função de Ponderação constante



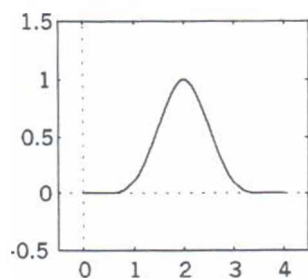
Aproximação tradicional de Mínimos Quadrados



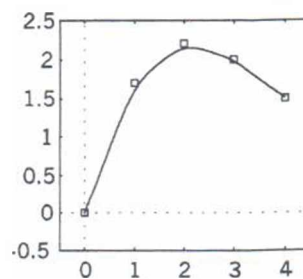
Função de Ponderação Degrau
Suporte metade do domínio



Mínimos Quadrados Móvel
Produz interpolação linear
por partes



Função *Spline* Quártica
Suporte = 1,5



Mínimos Quadrados Móvel
Produz aproximação

Figura 3.4 – Efeito da função de ponderação sobre o Método de Mínimos Quadrados Móvel.

Caso o espaçamento nodal h não seja regular pode-se adotar $h = dm_i$, ficando a equação (3.40) independente de h , resultando na expressão

$$dm_i \geq (1 + \alpha) d_{\min} \quad (3.42)$$

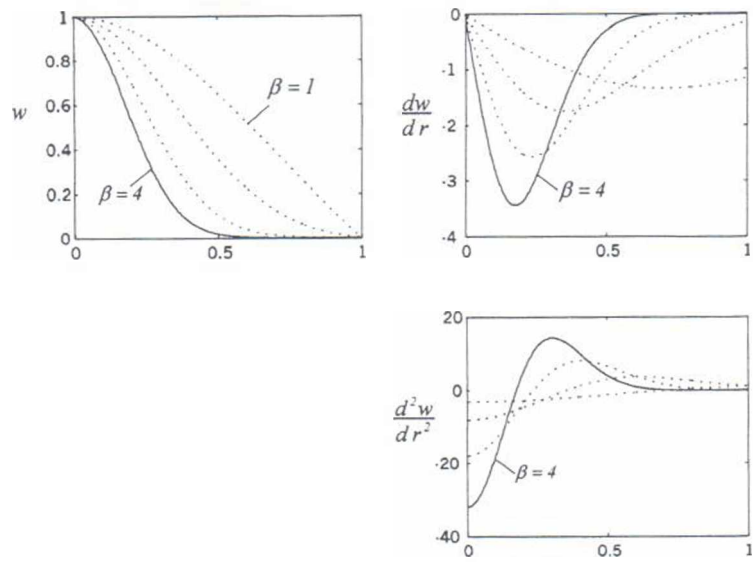


Figura 3.5 - Influência do parâmetro β na função de ponderação gaussiana.

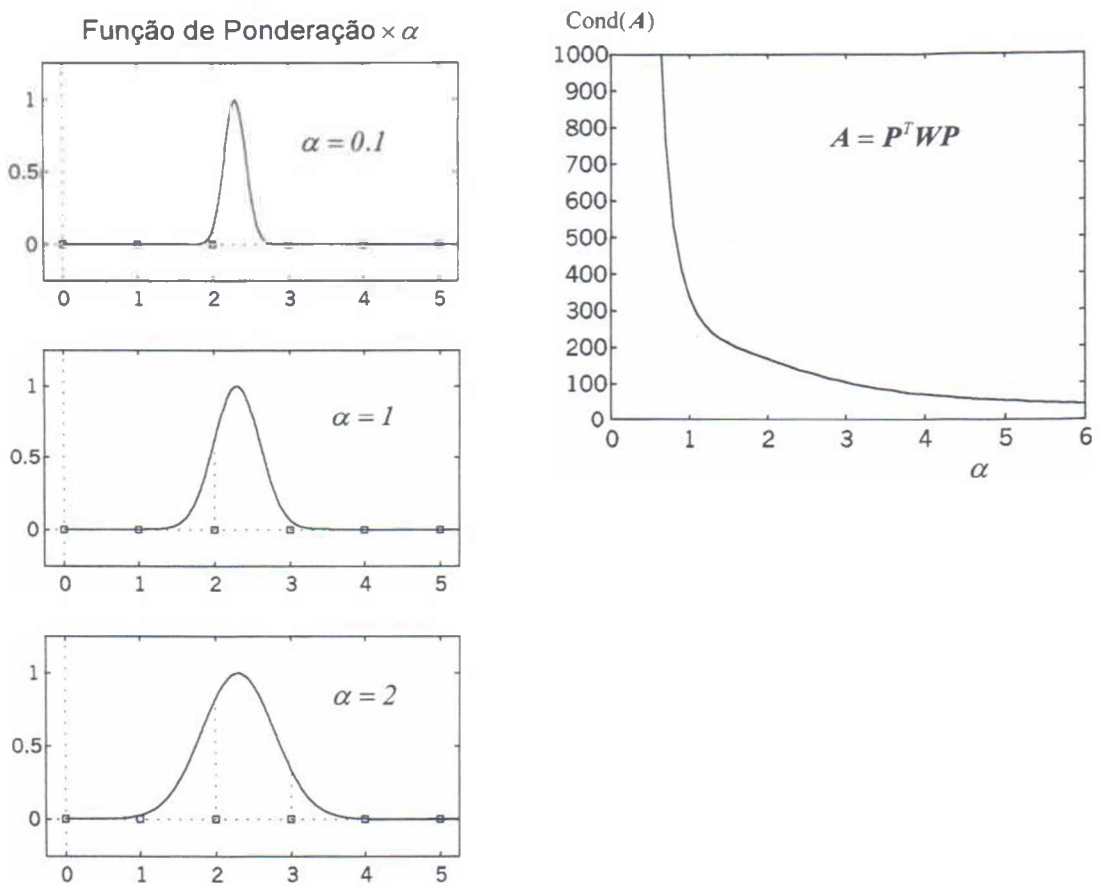


Figura 3.6 - Avaliação do efeito da dimensão do suporte.

A Figura 3.7 apresenta as funções de forma e sua derivada considerando α variando de um a três. Reproduzindo experimento numérico apresentado por Askes *et al.* (2000). Como se pode verificar, para valores baixos de α a função de forma assume grande semelhança com as funções de forma lineares utilizadas no Método dos Elementos Finitos. Para valores maiores do parâmetro α (dois ou três) a função de forma possui comportamento mais suave, como o de polinômios de ordem elevada.

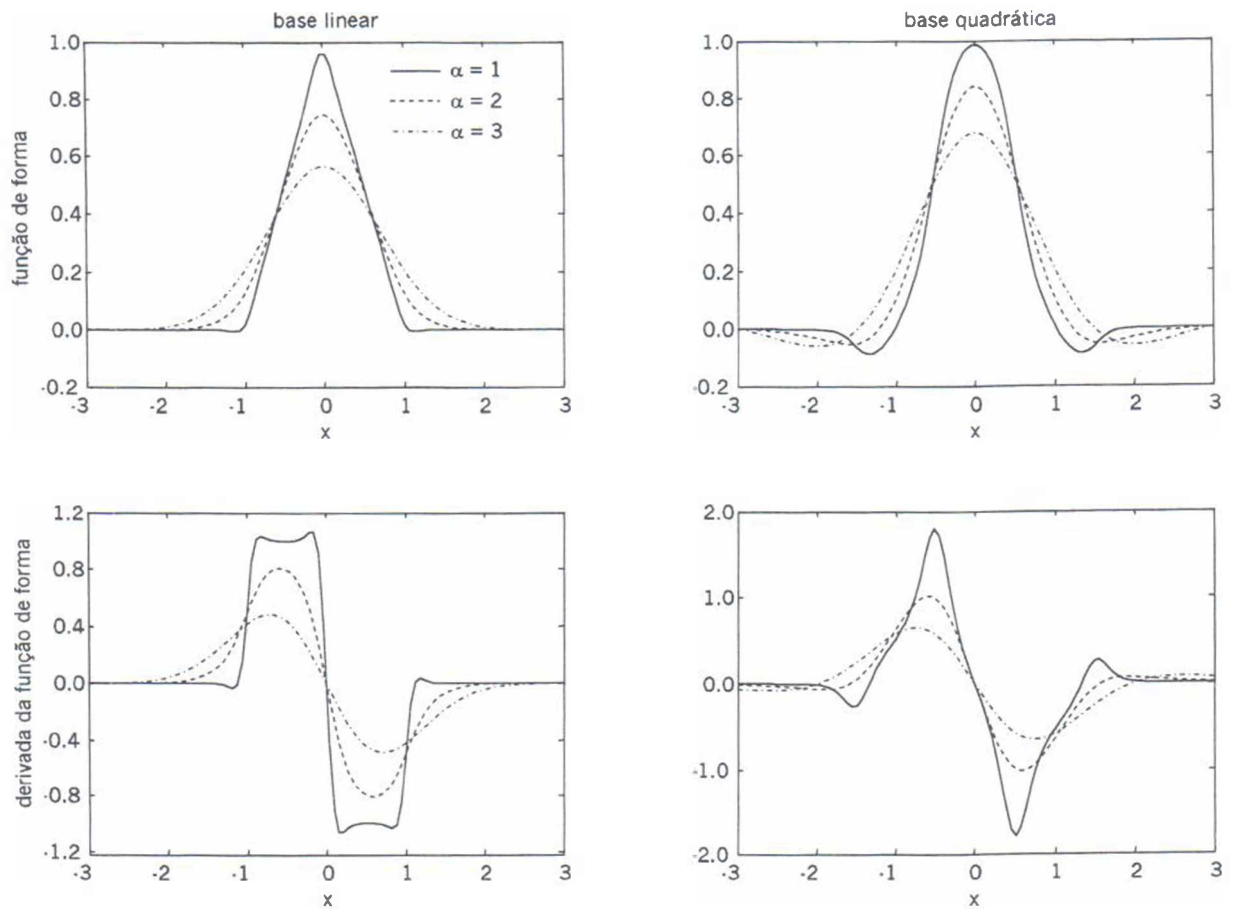


Figura 3.7 - Variação da função de forma com a dimensão do suporte.

Como mencionado no início desta seção, as funções de ponderação podem ser geradas por produto tensorial. Dolbow e Belytschko (1998 e 1999), utilizaram a seguinte função *spline* cúbica como função de ponderação

$$w(r) = \begin{cases} \frac{2}{3} - 4r^2 + 4r^3 & 0 \leq r \leq \frac{1}{2} \\ \frac{4}{3} - 4r + 4r^2 + \frac{4}{3}r^3 & \frac{1}{2} < r \leq 1 \\ 0 & r > 1 \end{cases} \quad (3.43)$$

onde r é o raio do domínio de influência.

A função de ponderação produto tensorial é definida em cada ponto como

$$w(\mathbf{x} - \mathbf{x}_i) = w(r_x) \cdot w(r_y) \quad (3.44)$$

onde $w(r_x)$ ou $w(r_y)$ são dados por (3.43) com o valor de r substituído por r_x ou r_y , respectivamente. Sendo esses parâmetros dados por

$$r_x = \frac{\|\mathbf{x} - \mathbf{x}_i\|}{dm_x}, \quad r_y = \frac{\|\mathbf{y} - \mathbf{y}_i\|}{dm_y} \quad (3.45)$$

onde

$$\begin{aligned} dm_x &\geq d_{\min}^x + \alpha h_x \\ dm_y &\geq d_{\min}^y + \alpha h_y \end{aligned} \quad (3.46)$$

sendo h_x e h_y o espaçamento nodal nas direções x e y ; e, d_{\min}^x e d_{\min}^y as componentes x e y da distância d_{\min} definida pela equação (3.42). Como antes pode-se adotar $h_x = d_{\min}^x$ e $h_y = d_{\min}^y$ ficando o critério para definição do raio de suporte

$$\begin{aligned} dm_x &\geq (1 + \alpha) d_{\min}^x \\ dm_y &\geq (1 + \alpha) d_{\min}^y \end{aligned} \quad (3.47)$$

Este critério difere ligeiramente do proposto por Dolbow e Belytschko com o propósito de se obter um critério uniforme para a determinação do raio de suporte.

Convém registrar que as derivadas da função de ponderação produto tensorial são calculadas por

$$\begin{aligned}w_x &= \frac{dw}{dr_x} \cdot \frac{dr_x}{dx} \cdot w(r_y) \\w_y &= \frac{dw}{dr_y} \cdot \frac{dr_y}{dy} \cdot w(r_x)\end{aligned}\tag{3.48}$$

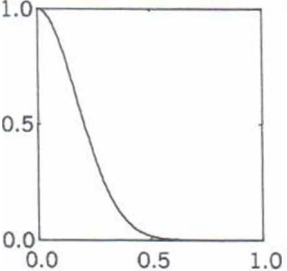
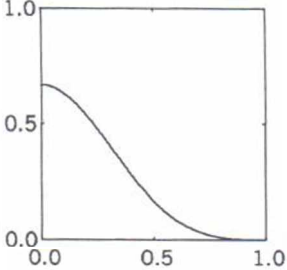
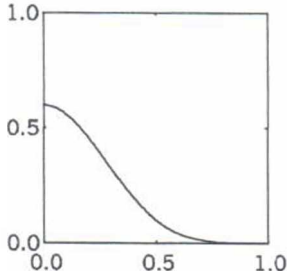
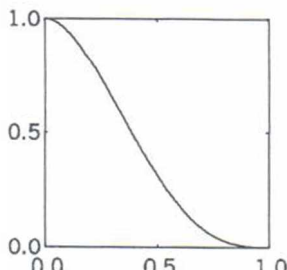
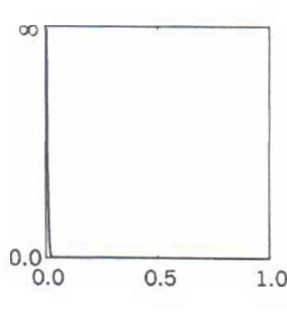
No caso da função spline cúbica as derivadas das funções de forma são obtidas da expressão

$$\frac{dw}{dr} = \begin{cases} -8r + 12r^2 & 0 \leq r \leq \frac{1}{2} \\ -4 + 8r - 4r^2 & \frac{1}{2} < r \leq 1 \end{cases}\tag{3.49}$$

substituindo por r_x e r_y , respectivamente; e sendo

$$\begin{aligned}\frac{dr_x}{dx} &= \frac{\text{sign}(x - x_i)}{dm_x} \\ \frac{dr_y}{dy} &= \frac{\text{sign}(y - y_i)}{dm_y}\end{aligned}\tag{3.50}$$

Tabela 3.1 – Funções de Ponderação.

Representação	Expressão	Bibliografia
	<p>Função Gaussiana Truncada</p> $w = \frac{\exp(-\beta^2 r^2) - \exp(-\beta^2)}{1 - \exp(-\beta^2)} \quad 0 \leq r \leq 1$ <p>$\beta = 4$</p>	<p>Belytschko <i>et al.</i> (1994)</p>
	<p>Função Spline de 3ª ordem</p> $w = \begin{cases} \frac{2}{3} - 4r^2 + 4r^3 & 0 \leq r \leq 0.5 \\ \frac{4}{3} - 4r + 4r^2 + \frac{4}{3}r^3 & 0.5 < r \leq 1 \end{cases}$	<p>Dolbow e Belytschko (1998)</p>
	<p>Função B-Spline de 4ª ordem</p> $w = \begin{cases} C[(r-1)^4 - 5(r-3/5)^4 + 10(r-1/5)^4] & 0 \leq r < 1/5 \\ C[(r-1)^4 - 5(r-3/5)^4] & 1/5 \leq r < 3/5 \\ C(r-1)^4 & 3/5 \leq r \leq 1 \end{cases}$ $C = \frac{1}{24} \frac{1}{(2/5)^4} = \frac{625}{384}$	<p>Duarte (1996) Garcia <i>et al.</i> (2000)</p>
	<p>Função Spline de 4ª ordem</p> $w = 1 - 6r^2 + 8r^3 - 3r^4 \quad 0 \leq r \leq 1$	<p>Beissel, Belytschko (1996)</p>
	<p>Função Potência Inversa</p> $w = \frac{1}{c r^\beta}$ <p>$c = 1$ e $\beta = 6$</p>	<p>Liszka <i>et al.</i> (1980) Liszka <i>et al.</i> (1996)</p>

3.5 - EXEMPLO DE APLICAÇÃO – AJUSTAMENTO DE SUPERFÍCIES

O procedimento apresentado na Seção 3.2 para geração de funções de forma pode ser utilizado para a construção de superfícies a partir de dados dispersos no plano.

Como aplicação seja o “relevo” definido pelas equações

$$\begin{array}{ll}
 z = 1, & \text{se } y - x \geq 1/2 \\
 z = 2(y - x), & \text{se } 0 \leq y - x < 1/2 \\
 z = \frac{1}{2} \left\{ \cos \left(4\pi \left[(x - 3/2)^2 + (y - 1/2)^2 \right]^{1/2} \right) + 1 \right\}, & \text{se } \left[(x - 3/2)^2 + (y - 1/2)^2 \right]^{1/2} < 1/16 \\
 z = 0 & \text{caso contrário}
 \end{array} \tag{3.51}$$

e com fronteira na região do retângulo $[0 \ 2] \times [0 \ 1]$, denominado aqui de “Relevo Lancaster-Šalkauskas”. A Figura 3.8 apresenta perspectiva e linhas de contorno desse relevo.

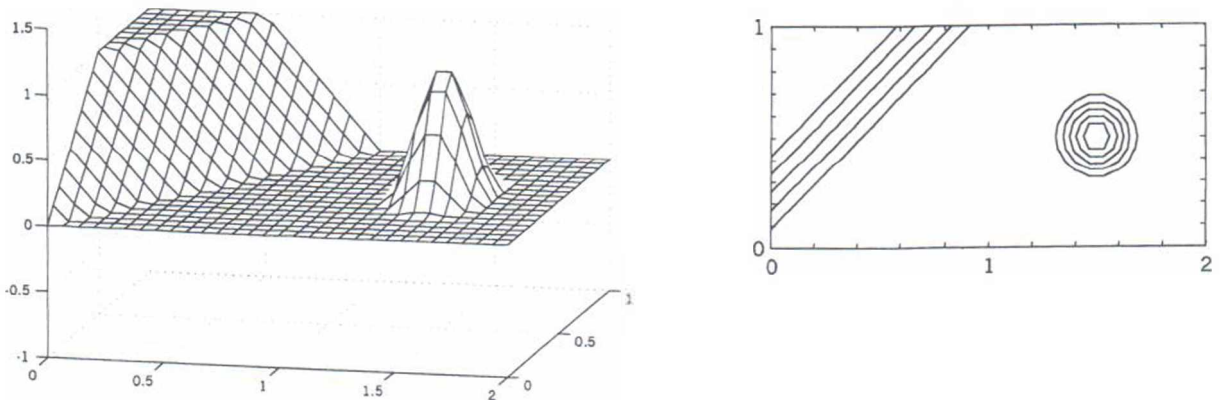


Figura 3.8 - Relevo Lancaster-Šalkauskas.

Pretende-se a construção da superfície do relevo a partir de um conjunto disperso de pontos $z_i = (x_i, y_i)$, $i = 1, N$, e correspondentes valores conhecidos f_i obtidos da expressão matemática do relevo (3.51).

A Figura 3.9 apresenta um conjunto de 150 pontos gerados aleatoriamente com respectivos valores amostrados da expressão (3.51). Como se pode verificar trata-se de um conjunto esparsos e que visualmente diz pouco sobre o relevo a construir.

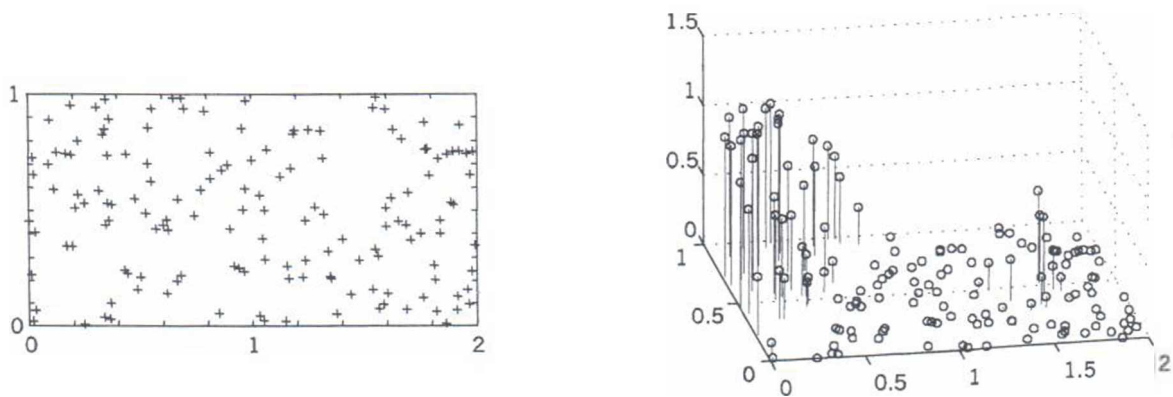


Figura 3.9 - Amostra de dados aleatórios do Relevo Lancaster-Šalkauskas.

Seja então u a função a ser determinada no domínio convexo fechado que contém z ; e sendo $u \in C^1$, ou seja uma vez diferenciável (continuidade das tangentes), nesse domínio, e ainda que atenda a condição de interpolação, vale dizer $u(z_i) = f_i$, para $i = 1 \dots N$.

A função u atendendo as imposições anteriores pode ser construída a partir da combinação linear de m funções linearmente independentes b_j na forma

$$u(z) = \sum_{j=1}^m a_j(z) b_j(z) \quad (3.52)$$

onde $a_j(z)$ são constantes que dependem de z .

Deve-se assumir que $m \leq N$, para que o sistema de equações algébricas possua solução única. Na prática m é bem menor que N . As funções b_j são arbitrárias, desde que sejam linearmente independentes e pertencentes a um conjunto completo, como estabelecido na Seção 3.2. Como o Método dos Mínimos Quadrados reproduz as funções da base e como se deseja a condição $u(z_i) = f_i$ e também se pretende reproduzir o plano,

as funções $\{1, x \text{ e } y\}$ devem ser incluídas na base de funções. Por outro lado como se deseja continuidade C^1 a base a ser utilizada na construção de $u(z)$ será o conjunto de $m = 6$ funções monômios

$$\begin{aligned} b_1(z) &= 1, & b_2(z) &= x, & b_3(z) &= y, \\ b_4(z) &= x^2, & b_5(z) &= xy, & b_6(z) &= y^2 \end{aligned} \quad (3.53)$$

Os coeficientes $a_j(z)$ de (3.82) são determinados a partir do funcional ponderado

$$\begin{aligned} E(u, z) &= \sum_{i=1}^N W_i(z) (u(z_i) - f_i)^2 \\ E(u, z) &= \sum_{i=1}^N W_i(z) \left(\sum_{j=1}^m a_j(z) b_j(z) - f_i \right)^2 \end{aligned} \quad (3.54)$$

Efetuada a minimização $\partial E / \partial a_i = 0$, para $i = 1 \dots N$, chega-se ao seguinte sistema normal de equações na forma matricial

$$\mathbf{B}^T \mathbf{W} \mathbf{B} \mathbf{a} = \mathbf{B}^T \mathbf{W} \mathbf{f} \quad (3.55)$$

onde \mathbf{W} é a matriz diagonal com os valores da função de ponderação, a ser detalhada adiante; \mathbf{a} é o vetor de coeficientes $a_j(z)$; \mathbf{f} é o vetor dos valores dos dados f_i ; e \mathbf{B} é a matriz $m \times N$ com os valores da base de funções coletadas na forma

$$\mathbf{B} = \begin{bmatrix} b_1(z_1) & \dots & b_m(z_1) \\ \vdots & \ddots & \vdots \\ b_1(z_N) & \dots & b_m(z_N) \end{bmatrix} \quad (3.56)$$

Como no caso a base de funções é constituída de monômios, verifica-se que $\mathbf{B} = \mathbf{P}$, onde \mathbf{P} está definida na Seção 3.2.

Uma vez resolvido o sistema (3.55), ficam determinados os coeficientes $a_j(z)$ que permitem encontrar o valor $u(z)$ através da equação (3.52). Como se percebe o

sistema de equações normais (3.55) deve ser resolvido para cada valor de z para o qual a altura da superfícies será calculada.

Para que o processo de minimização do funcional conduza a uma superfície interpolante, a função de ponderação deve possuir o comportamento de tender a infinito quando a distância entre o ponto z e o ponto amostrado z_i tender a zero. No caso a distância é dada por

$$d(z, z_i) = \sqrt{(x - x_i)^2 + (y - y_i)^2} \quad (3.57)$$

Lancaster e Šalkauskas (1981) propõem para a função de ponderação uma equação do tipo $W(d) = e^{-\alpha d^2} / d^2$ e de modo a se evitar problemas com a singularidade e com *overflow* sugerem a seguinte expressão

$$W(d) = \frac{e^{-\alpha d^2}}{(d^2 + \varepsilon)} \quad (3.58)$$

com $\varepsilon = 0.001$ e $\alpha = 1/16$, representada na Figura 3.10. Cabe observar que a função proposta não define um suporte de influência.

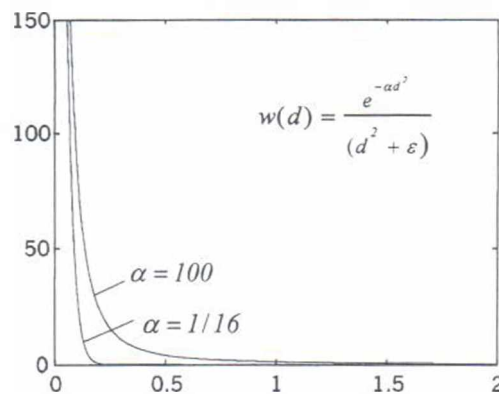


Figura 3.10 - Função de ponderação para determinação do relevo de Lancaster e Šalkauskas.

A Figura 3.11 apresenta o resultado para os 150 pontos gerados de forma aleatória com as respectivas curvas de nível.

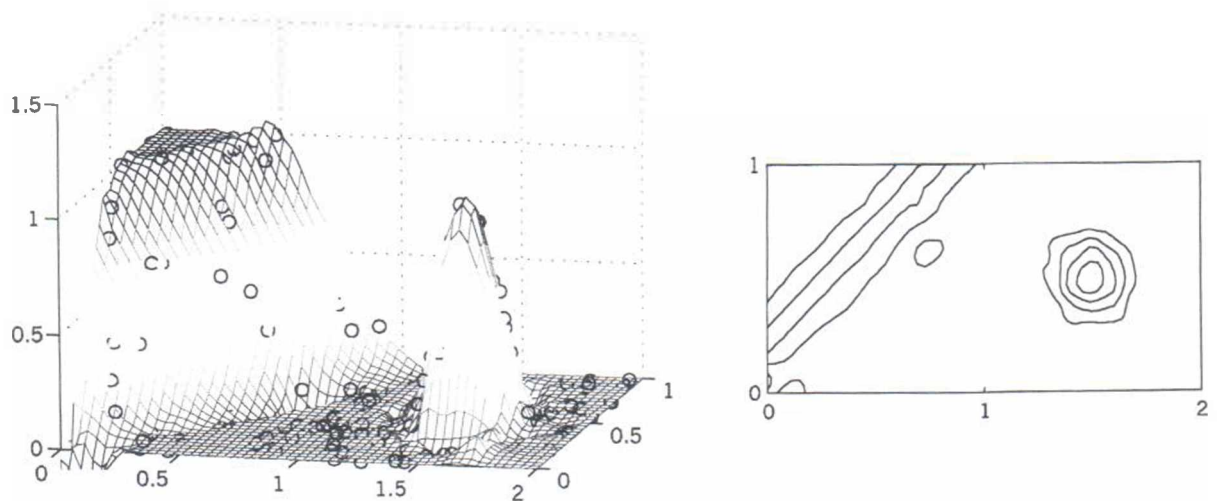


Figura 3.11 - Superfície interpolada por Mínimos Quadrados Móveis.

Comparando as Figuras 3.11 e 3.8 verifica-se que a superfície interpolada representa com boa qualidade o relevo acidentado proposto.

Zienkiewicz e Zhu (1992) utilizaram procedimento bastante similar ao apresentado acima na construção de indicadores de erro para processos auto-adaptativos no âmbito do Método dos Elementos Finitos, denominado de *Superconvergent Patch Recovery*.

3.6 - MODELAGEM COMPUTACIONAL DAS FUNÇÕES DE FORMA

Um princípio de projeto reutilizável orientado a objetos estabelece que se deve programar para uma interface e não para uma implementação (Gamma *et al.*, 2000). Uma interface é um protocolo com uma lista de declarações de métodos, sem código, que devem ser materializadas pelas classes que implementam essa interface. Com isto se estabelece dentro do mecanismo de herança uma importante decisão. Procura-se herdar o comportamento e não a implementação. A programação através de interfaces é a base de construção de programas de computador expansíveis. Assim todas as classes que implementam uma determinada interface terão comportamento externo similar diferenciando-se pela forma interna que se comportam.

As funções de forma serão construídas a partir da interface **ShapeFunction**. Na Figura 3.12 apresenta-se o diagrama UML com o protocolo desta interface. Uma classe que queira se comportar como uma **ShapeFunction** deverá implementar o código de todos os métodos listados. No caso, deve fornecer métodos para determinar as funções de forma e suas derivadas até a segunda ordem, que atende aos métodos de discretização deste trabalho. Deve informar ainda qual foi a nuvem de pontos e o pólo em coordenadas cartesianas utilizados para sua determinação.

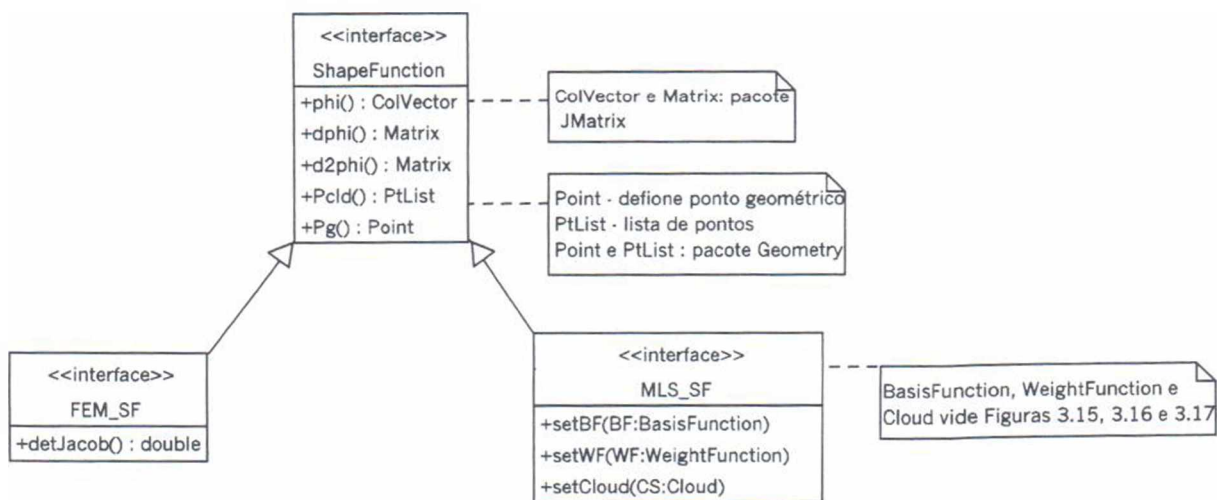


Figura 3.12 – Interfaces para determinação das funções de forma.

Na Figura 3.12 têm-se ainda duas especializações da interface **ShapeFunction** que são as interfaces **FEM_SF** e **MLS_SF**. A primeira será utilizada para definir as funções de forma de elementos isoparamétricos. Para isso incluiu-se um método para a determinação do determinante do Jacobiano. Todas as funções de forma de elementos finitos deverão implementar tanto o protocolo de **ShapeFunction** como o de **FEM_SF**. Estas funções de forma têm dupla finalidade, permite construir o Método dos Elementos Finitos e possibilita a integração gaussiana nos Métodos sem Malha com variante Galerkin.

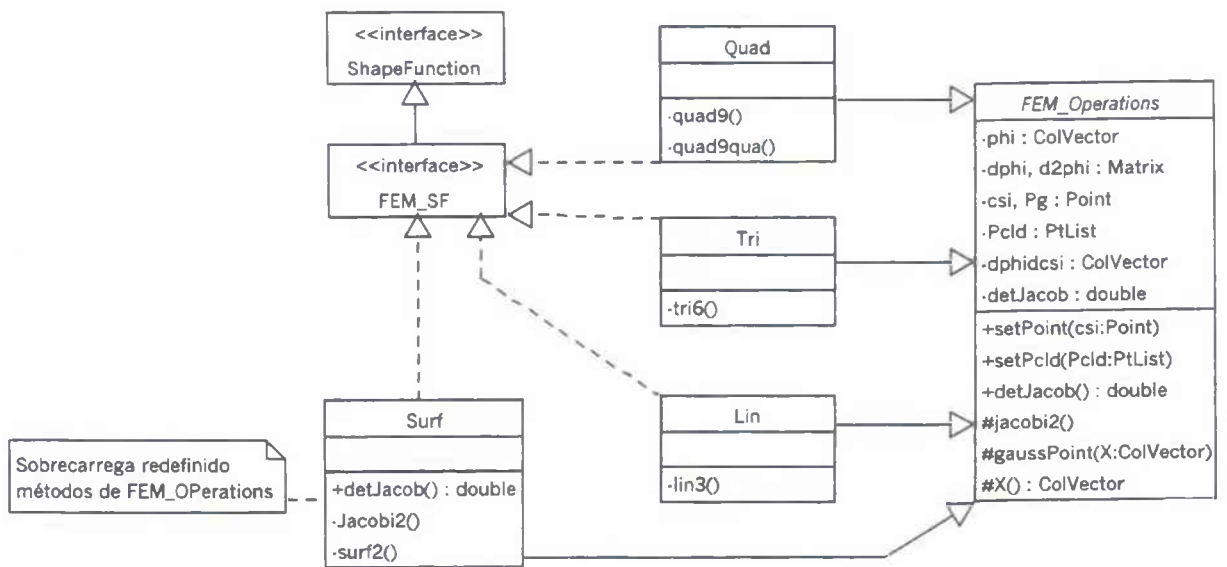


Figura 3.13 – Modelo de classes para elementos isoparamétricos.

A Figura 3.13 apresenta os elementos implementados neste trabalho que seguem a interface a interface **FEM_SF**: elementos quadriláteros, triangulares e lineares, que permitem a integração de domínios unidimensionais e bidimensionais, além de elemento de superfície para integração no contorno bidimensional. A construção agrega as operações e dados comuns aos diferentes elementos numa classe abstrata (que não gera objetos) denominada de **FEM_Operations**. Estas operações e dados são transmitidos por herança às classes que desejarem reaproveitar esse código ou parte dele. A classe **Quad**, por exemplo, implementa a interface **FEM_SF** e estende a classe abstrata **FEM_Operations**. Pode-se dizer que **Quad** herda sua genética, ou melhor, seu comportamento de **FEM_SF** e que herda ou não o patrimônio de código existente em

FEM_Operations. Por simplicidade não foram representados os métodos das interfaces nas classes concretas e tampouco os construtores das classes (métodos que criam os objetos das classes em tempo de execução). Como os elementos implantados são de classe C^1 os métodos que informam a derivada segunda das funções de forma retornam valor nulo.

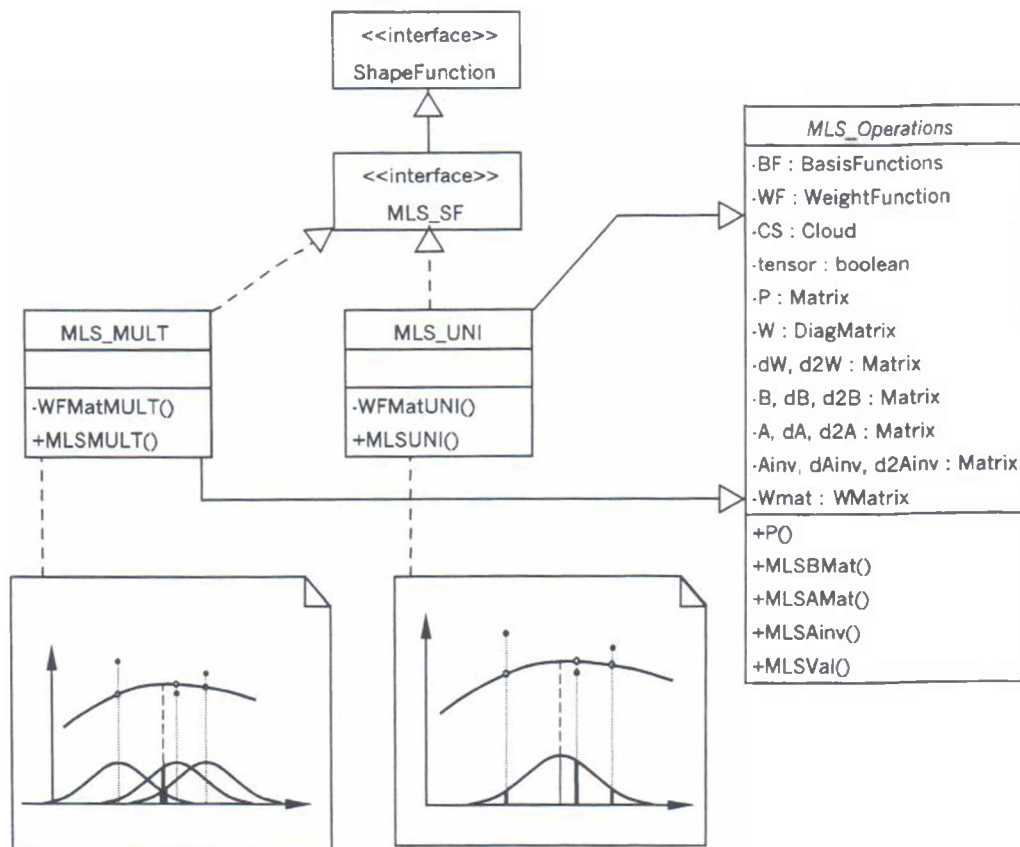


Figura 3.14 – Classes para modelagem do Método de Mínimos Quadrado Móvel.

A interface **MLS_SF**, representada na Figura 3.12 permite criar as classes necessárias à construção de funções de forma pelo Método dos Mínimos Quadrados Móvel. Sua interface agrega à interface herdada de **ShapeFunction**, métodos para a definição da base de funções, função de ponderação e estratégia de construção da nuvem de pontos, classes que serão detalhadas adiante. A Figura 3.14 apresenta as classes implementadas como embasamento da Seção 3.2. A classe **MLS_UNI** determina as funções de forma seguindo o modelo representado na Figura 3.1 onde a ponderação é realizada com suporte definido a partir de um ponto qualquer. Já a classe **MLS_MULT** segue a

estratégia de ponderação com suporte definido para cada ponto nodal. Aqui também foi criada uma classe abstrata que reúne os dados e métodos comuns que podem ser herdados, ou não, por outras classes. No Capítulo 4 uma subclasse de **MLS_UNI** será proposta de modo a permitir a determinação por especialização das funções de forma utilizadas no Método das Diferenças Finitas Generalizadas.

A modelagem computacional das bases de funções segue de forma semelhante ao apresentado para as funções de forma, através da definição de uma interface com o protocolo que especifica um comportamento (**BasisFunctions**) e da criação de uma classe abstrata (**BF_Operations**) container de dados e métodos que podem ser reaproveitados nas classes que implementam a interface. A Figura 3.15 apresenta esta estrutura bem como duas classes concretas denominadas de **Monomial** e **Crack**. A primeira implementa a base de funções de monômios como visto na Seção 3.4. Já a classe **Crack** representa a base proposta por Fleming *et al.* (1997) para problemas de fratura. No Capítulo 4 serão criadas duas classes que implementam bases de funções obtidas por série de Taylor no âmbito do Método das Diferenças Finitas Generalizadas.

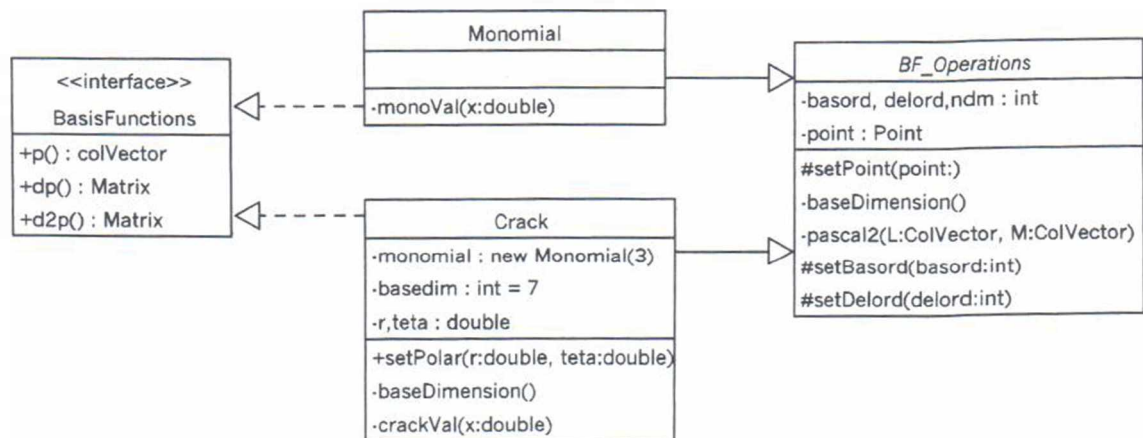


Figura 3.15 – Classes para modelagem das bases de funções.

A determinação da base de funções de monômios e de suas derivadas em duas dimensões é realizada através de produto tensorial utilizando o triângulo de Pascal (**pascal2()** na Figura 3.15). Assim após determinar a base de funções $L(x)$ e $M(y)$, e de suas derivadas $dL(x)$, $d2L(x)$, $dM(y)$ e $d2M(y)$ pode-se determinar utilizando a

rotina mencionada a base de funções em duas dimensões obtendo-se os resultados através dos métodos $p()$, $dp()$ e $d2p()$.

A modelagem dos objetos computacionais que representam as funções de ponderação no âmbito do Método dos Mínimos Quadrados Móvel deve abranger três abstrações. A primeira, mais imediata, é a determinação dos valores da função de ponderação e de suas derivadas. Todas as funções apresentadas na Tabela 3.1 foram parametrizadas com o raio adimensional r que varia de zero a um, correspondendo à dimensão do suporte da nuvem de pontos nodais dm_i . A Figura 3.16 apresenta a interface **WeightFunction** com a especificação do protocolo para obtenção da função e de suas derivadas até a segunda ordem em relação ao raio r . As funções de ponderação listadas na Tabela 3.1 estão representadas por uma classe correspondente na Figura 3.16. Devido à relativa complexidade da determinação das derivadas W_s e W_{sr} da matriz de ponderação em relação às coordenadas cartesianas, inclusive no caso tensorial, como visto na Seção 3.4, apresenta-se também na Figura 3.16 a classe **WFDeriv**. Esta classe é utilizada pelas classes que implementam **MLS_SF** para determinação da matriz diagonal de ponderação e de suas derivadas.

O segundo conceito ou abstração importante é a definição da nuvem de pontos (*cloud*) que permite definir o raio de suporte para cálculo dos valores da função de ponderação. Na literatura, a construção da nuvem de pontos é realizada com diversos critérios com respectivos algoritmos (ou estratégias). As diferentes estratégias se constituem no terceiro conceito ou abstração a considerar. Portanto os dois conceitos possuem grande acoplamento, sendo importante a possibilidade de se utilizar a mesma nuvem com diferentes algoritmos e seria conveniente a possibilidade de se expandir o número de algoritmos. Em computação orientada a objetos a construção que desacopla dois objetos (classes) possibilitando múltiplos algoritmos em um deles é conhecido como padrão de projetos **Strategy** (Gamma *et al.*, 2000). Assim a Figura 3.17 apresenta a classe **Cloud** que permite a construção de nuvens de pontos nodais e obtenção de seus dados tais como raio de suporte, o conjunto de pontos nodais (nuvem) para um dado ponto pólo no domínio, para diferentes estratégias.

A conexão entre a classe **Cloud**, que constrói nuvens e os algoritmos de construção é feita através da interface **CloudStrategy**. Esta interface estabelece o

protocolo que permite a interação entre a classe **Cloud** e os diferentes algoritmos concretos implementados. A operação **computeStrategy()**, que deve estar presente em todas as estratégias concretas, executa o algoritmo propriamente. Como comentado na Figura 3.17 a operação **buildCloud()** da classe **Cloud** executa o algoritmo correspondente à estratégia corrente.

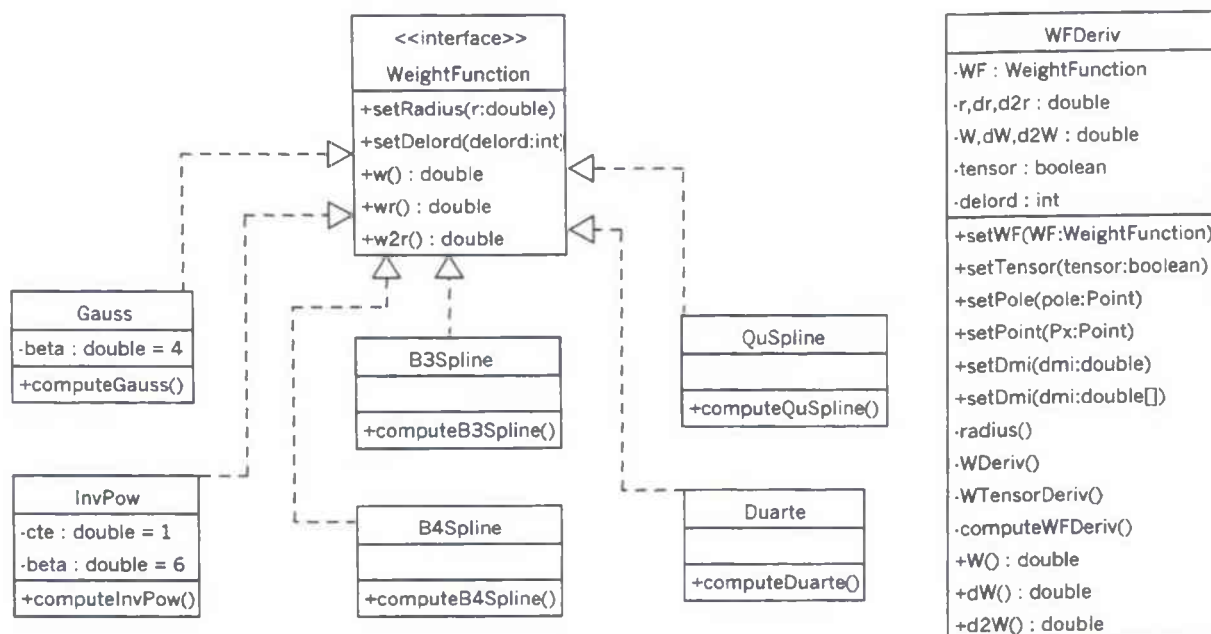


Figura 3.16 – Classes que modelam as funções de ponderação.

A Figura 3.17 apresenta três estratégias para configuração da nuvem de pontos. **Distance** determina a nuvem selecionando os m pontos nodais mais próximos do pólo, onde m é a dimensão da base de funções utilizada. Devido a problemas de instabilidade numérica o suporte da nuvem é aumentado através de um parâmetro empírico α como apresentado na Seção 3.4. O critério **FixedSupport** determina a nuvem de pontos partindo-se de um valor pré-estabelecido para a dimensão do suporte da nuvem. Ambos critérios permitem determinar uma nuvem com suporte retangular para utilização de funções de ponderação obtidas por produto tensorial. O terceiro critério apresentado é na verdade uma particularização para obtenção do conjunto de pontos presentes numa célula de integração, ou se for o caso em um elemento finito.

Outros critérios podem ser facilmente implementados sem alterar a estrutura existente. Por exemplo, um critério que determine a nuvem a partir de um número fixo de pontos nodais, procedimento semelhante a diferenças finitas. No Capítulo 4 serão apresentados outros critérios decorrentes da implementação dos Métodos sem Malha.

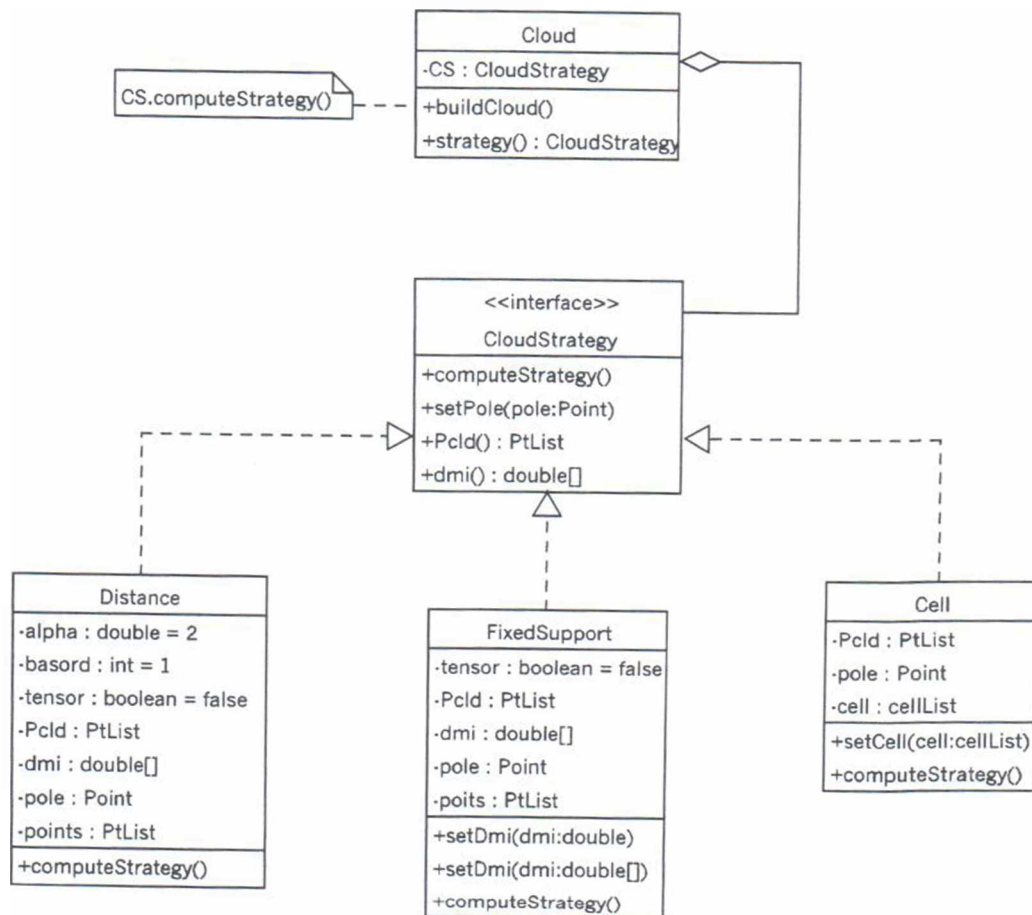


Figura 3.17 – Classes para modelagem da nuvem de pontos e estratégias de construção.

3.7 - COMENTÁRIOS FINAIS

As funções de forma definidas pelo Método de Mínimos Quadrados Móvel apresentadas neste capítulo são a base para a implementação dos Métodos sem Malha relacionados neste trabalho. Estas funções, construídas através de uma ponderação aplicada sobre uma nuvem esparsa de pontos nodais sem conexão prévia, conferem o caráter sem malha (sem conectividade) aos métodos que serão detalhados no próximo capítulo. O aprofundamento do estudo da estratégia de construção de nuvens é crucial para a determinação de uma solução estável nos diferentes problemas da mecânica.

A modelação computacional elaborada tem como foco a expansão e a reutilização do programa criado. A utilização de padrões de projeto, como na modelação das diversas estratégias de construção de nuvens, é uma tendência na concepção de estruturas computacionais de classes (*frameworks*). A utilização desses padrões de projeto não foi aprofundada na literatura orientada a objetos no âmbito do Método dos Elementos Finitos.

Como visto, a interface criada para tratamento das funções de forma foi desenhada de modo a permitir o emprego das funções de forma dos elementos isoparamétricos. Outras funções de forma podem ser acrescentadas de modo a atender à solução de problemas específicos.

4 - MÉTODOS NUMÉRICOS DE DISCRETIZAÇÃO SEM MALHA

4.1 - INTRODUÇÃO

Neste capítulo são apresentados os Métodos sem Malha listados para estudo neste trabalho, como assinalado no Capítulo 1:

1. Diferenças Finitas Generalizadas (*Generalized Finite Difference Method –GFD*).
2. Método de Galerkin sem Malha (*Element-Free Galerkin - EFG*).
3. Método sem Malha Hierárquico (*Hp-Clouds*).
4. Método do Ponto Finito (*Finite Point Method - FPM*).

Esses métodos foram escolhidos de modo a ilustrar diferentes facetas dos Métodos sem Malha. Dois deles seguem critério de colocação e dois seguem critério de Galerkin na formulação do Método dos Resíduos Ponderados. Todos obtêm as funções de forma de um modo peculiar a partir do Método de Mínimos Quadrados Móvel.

O Método das Diferenças Finitas Generalizadas possui cunho histórico. Uma variante denominada de Métodos dos Operadores Discretos (Gonçalves e Pamplona, 1980; Pulino, 1989) foi o ponto de partida deste trabalho.

O Método de Galerkin sem Malha é, num sentido lato, uma generalização do Método dos Elementos Finitos. Sua importância no âmbito dos Métodos sem Malha é bastante destacada na literatura. Além da seção introdutória apresentada neste capítulo, o Capítulo 5 é dedicado ao estudo mais detalhado do método através do problema de elasticidade bidimensional.

O Método sem Malha Hierárquico, denominado de *hp-Clouds* é na verdade uma variante do Método de Galerkin sem Malha, e que permite a incorporação de um enriquecimento das funções de forma do tipo- p , através do aumento da ordem da base polinomial por ponto nodal. O enriquecimento tipo- h é inerente a todos os Métodos sem Malha.

O Método do Ponto Finito tem origem ligada à aplicação em Mecânica dos Fluidos com a utilização de termos de estabilização numérica em problemas com número elevado de pontos nodais. A necessidade destes termos torna o método dependente do problema analisado.

A modelação computacional, dos Métodos sem Malha listados, dentro de um único arcabouço matemático representado pelo Método dos Resíduos Ponderados, como visto no Capítulo 2, evidencia a importância da escolha de abstrações, ou seja, de conceitos, de alto nível, na construção de programas de computador reutilizáveis. Assim outros métodos podem ser facilmente incorporados sendo o código pré-existente reaproveitado, como demonstra a modelação no final do capítulo.

4.2 - MÉTODO DAS DIFERENÇAS FINITAS GENERALIZADAS

Na literatura o Método das Diferenças Finitas com esquema de pontos nodais arranjados de forma não regular é conhecido como Método das Diferenças Finitas Generalizadas. A linha de investigação com implementação amplamente reconhecida como mais robusta segue a trilha de Perrone e Kao (1975), e Lyszka e Orkisz (1980).

A consideração da aproximação local é feita a partir do conceito de uma molécula (*molecule, star* ou *stencil*) de pontos nodais com geometria arbitrária. Trata-se na verdade de uma especialização do conceito de nuvem de pontos nodais, ocorrendo aqui uma organização por conectividade dos pontos nodais através das ligações do padrão gerado pela molécula. A aproximação das derivadas para cada ponto nodal central da molécula é estabelecida através de expansões em série de Taylor usando a informação provida pelo número e posição dos pontos nodais de cada molécula. Ou seja, as funções de forma e suas derivadas são obtidas simultaneamente como será visto.

Para uma função qualquer suficientemente diferenciável $u(x, y)$, em um dado domínio bidimensional, a expansão em série de Taylor em torno de um ponto (x_0, y_0) pode ser escrita como

$$u = u_0 + h \frac{\partial u_0}{\partial x} + k \frac{\partial u_0}{\partial y} + \frac{h^2}{2} \frac{\partial^2 u_0}{\partial x^2} + kh \frac{\partial^2 u_0}{\partial x \partial y} + \frac{k^2}{2} \frac{\partial^2 u_0}{\partial y^2} + O(\rho^3) \quad (4.1)$$

onde

$$u = u(x, y), \quad u_0 = u(x_0, y_0), \quad h = x - x_0$$
$$k = y - y_0, \quad e \quad \rho = \sqrt{h^2 + k^2}$$

A expressão (4.1) pode ser interpretada como uma aproximação polinomial com uma base de funções na forma

$$p = \left[1, x, y, \frac{x^2}{2}, xy, \frac{y^2}{2}, \dots \right]^T \quad (4.2)$$

com aplicação no ponto $x - x_0 = (h, k)$ onde $x = (x, y)$ e $x_0 = (x_0, y_0)$. Assim pode-se reescrever (4.1) como sendo

$$u(\mathbf{X}) = \mathbf{p}^T (\mathbf{X} - \mathbf{X}_0) \mathbf{D}u \quad (4.3)$$

$$\text{onde } \mathbf{D}u = \left[u_0, \frac{\partial u_0}{\partial x}, \frac{\partial u_0}{\partial y}, \frac{\partial^2 u_0}{\partial x^2}, \frac{\partial^2 u_0}{\partial x \partial y}, \frac{\partial^2 u_0}{\partial y^2} \right]^T$$

A base de funções de Taylor, expressão (4.2), pode conter mais termos desde que derivadas de ordem mais elevada sejam necessárias ou se aproximações mais precisas são desejadas em derivadas de ordem mais baixa. Considerações estas decorrentes do Teorema de Taylor, e que permite vislumbrar uma adaptabilidade hierárquica tipo p , como sugerido na literatura do Método dos Elementos Finitos.

Voltando à equação (4.1) verifica-se que são necessárias cinco equações independentes para se determinar as cinco derivadas diferentes no ponto (x_0, y_0) , ou seja é necessário conhecer os valores de u em cinco pontos vizinhos. Escrevendo (4.1) para cinco pontos no entorno do ponto u_0 chega-se

$$\mathbf{P} \mathbf{D}u - \mathbf{u} = \mathbf{0} \quad (4.4)$$

$$\text{onde } \mathbf{P} = \begin{bmatrix} h_1 & k_1 & h_1^2/2 & h_1 k_1 & k_1^2/2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ h_5 & k_5 & h_5^2/2 & h_5 k_5 & k_5^2/2 \end{bmatrix}$$

$$\mathbf{u} = [u_1 - u_0 \quad u_2 - u_0 \quad \cdots \quad u_5 - u_0]^T$$

$$\text{neste caso com } \mathbf{D}u = \left[\frac{\partial u_0}{\partial x}, \frac{\partial u_0}{\partial y}, \frac{\partial^2 u_0}{\partial x^2}, \frac{\partial^2 u_0}{\partial x \partial y}, \frac{\partial^2 u_0}{\partial y^2} \right]^T.$$

Esta é a formulação clássica do Método das Diferenças Finitas Generalizadas para moléculas de seis pontos e que considera o caráter interpolador da série de Taylor para o valor da função e de suas derivadas em um ponto. A opção por moléculas com um número maior de pontos nodais implica no aumento da ordem da série de Taylor, de tal forma que o número total de pontos que envolvem a molécula seja igual à dimensão da base de Taylor menos um.

A matriz \mathbf{P} é bastante mal condicionada e diversos esquemas foram propostos na literatura para a escolha dos pontos nodais que compõem a molécula (Liszka e Orkisz,

1980; Godoy, 1986), de modo a melhor condicionar a molécula e também de modo a automatizar a construção dessas moléculas. O mesmo fenômeno ocorre quando do ajustamento pelo polinômio interpolador que conduz à matriz de Vandermonde (Lancaster e Šalkauskas, 1986), também mal condicionada, mesmo para dimensões pequenas, e que possui alguma similaridade com a matriz P .

No caso de uma distribuição ortogonal quadrada de pontos nodais a matriz P para seis pontos nodais conduz a equações idênticas àquelas do esquema convencional de diferença central do Método das Diferenças Finitas (Pulino Filho, 1986), a menos da derivada mista (Perone e Kao, 1975).

Liszka e Orkisz (1980) propuseram, como forma de contornar o mal condicionamento da matriz P e melhorar a qualidade das derivadas Du , a utilização de um número maior de pontos nodais ($m > 5$) para a construção das aproximações, o que resulta em um sistema super-determinado. Para obtenção das derivadas estabeleceram a minimização da norma ponderada do erro

$$N = \sum_{i=1}^m \left[\left(u_0 - u_i + \frac{\partial u_0}{\partial x} h_i + \dots \right) \frac{1}{\rho_i^3} \right]^2 = \min \quad (4.5)$$

Escrevendo

$$\frac{\partial N}{\partial (Du)} = 0 \quad (4.6)$$

chega-se a um sistema linear de equações com cinco incógnitas.

Duarte (1995) demonstrou que a formulação de Liszka e Orkisz possui similaridade com o Método de Mínimos Quadrados Móvel descrito por Lancaster e Šalkauskas (1981). Adotando a expansão a partir da origem, por simplicidade, ou seja $(x_0, y_0) = (0, 0)$, a expressão (4.3) pode ser posta na forma

$$u(x_i) = p^T(x_i) Du(0) \quad (4.7)$$

Neste caso a norma do erro (4.5) fica

$$\mathbb{N}(\mathbf{Du}(0)) = \sum_{i=1}^m w_i(0) (u(x_i) - u_i)^2 \quad (4.8)$$

onde $w_i(0) = \frac{l}{\rho_i^6}$, são os valores das funções de ponderação tomados a partir da origem.

Colocando a expressão (4.8) em uma forma matricial chega-se a

$$\mathbb{N}(\mathbf{Du}(0)) = (\mathbf{P Du}(0) - \mathbf{u})^T \mathbf{W}(0) (\mathbf{P Du}(0) - \mathbf{u}) \quad (4.9)$$

onde

$$\mathbf{P} = \begin{bmatrix} 1 & h_1 & k_1 & h_1^2/2 & h_1 k_1 & k_1^2/2 \\ \vdots & & \vdots & & & \\ 1 & 0 & 0 & 0 & 0 & 0 \\ \vdots & & \vdots & & & \\ 1 & h_n & k_n & h_n^2/2 & h_n k_n & k_n^2/2 \end{bmatrix}$$

$$\mathbf{u} = [u_0 \quad u_1 \quad \dots \quad u_n]^T$$

$$\mathbf{Du}(0) = \left[u_0 \quad \frac{\partial u_0}{\partial x} \quad \frac{\partial u_0}{\partial y} \quad \frac{\partial^2 u_0}{\partial x^2} \quad \frac{\partial^2 u_0}{\partial x \partial y} \quad \frac{\partial^2 u_0}{\partial y^2} \right]^T$$

Efetuada a minimização obtém-se

$$\mathbf{Du}(0) = \mathbf{A}^{-1} \mathbf{B u} = \boldsymbol{\varphi u} \quad (4.10)$$

onde

$$\begin{aligned} \mathbf{A} &= \mathbf{P}^T \mathbf{W}(0) \mathbf{P} \\ \mathbf{B} &= \mathbf{P}^T \mathbf{W}(0) \end{aligned} \quad (4.11)$$

As funções de forma e suas derivadas são dadas por

$$\boldsymbol{\varphi} = \mathbf{A}^{-1} \mathbf{B} \quad (4.12)$$

onde $\boldsymbol{\varphi}$ coleta para cada termo de $\mathbf{Du}(0)$ as contribuições de cada ponto nodal. Ou, apresentando numa forma mais explícita

$$\mathbf{Du}(0) = \begin{Bmatrix} u_0 \\ \frac{\partial u_0}{\partial x} \\ \vdots \\ \frac{\partial^2 u_0}{\partial y^2} \end{Bmatrix} = \begin{bmatrix} \phi_1 & \dots & \phi_i & \dots & \phi_n \\ \phi_{1,x} & & \phi_{i,x} & & \phi_{n,x} \\ \vdots & & \vdots & & \vdots \\ \phi_{1,yy} & \dots & \phi_{i,yy} & \dots & \phi_{n,yy} \end{bmatrix} \begin{Bmatrix} u_1 \\ \vdots \\ u_i \\ \vdots \\ u_n \end{Bmatrix} \quad (4.13)$$

Como neste procedimento adotou-se função de ponderação com caráter interpolador ($w_i = 1/\rho_i^\beta$) é fácil verificar que se obterá

$$u_0 = \begin{bmatrix} 0 & \dots & 1 & \dots & 0 \end{bmatrix} \begin{Bmatrix} u_1 \\ \vdots \\ u_i \\ \vdots \\ u_n \end{Bmatrix} = u_i \quad (4.14)$$

ou seja

$$\phi_i = \begin{cases} 1 & \text{para } i = \text{pólo da molécula} \\ 0 & \text{para } i \neq \text{pólo da molécula} \end{cases}$$

Neste caso a função de forma obtida no Método das Diferenças Finitas Generalizadas segue o delta de Kronecker tão característico das funções de forma do Método dos Elementos Finitos.

Em suma, o procedimento apresentado por Liszka e Orkisz é uma especialização do Método de Mínimos Quadrados Móvel com interpolação, onde os parâmetros obtidos $\mathbf{Du}(x)$ são a função de forma e suas derivadas nesse ponto. A especialização é obtida através da utilização da base de função de Taylor, expressão (4.4), e a interpolação decorre da função de ponderação adotada ($w_i = 1/\rho_i^\beta$) que tende a infinito quando $x \rightarrow 0$, resultando $u(x_i) = u_i$. A singularidade pode ser tratada numericamente limitando-se w_i a um valor relativamente grande, como realizado na Seção 3.6, ou através de procedimento de normalização demonstrado por Lancaster e Šalkauskas (1986).

A função de ponderação utilizada não mantém correlação com o suporte definido pelos pontos adotados na nuvem da molécula. Em artigo de 1996, Liszka *et al.*, propõem função de ponderação com o seguinte aspecto

$$w_i = \frac{1}{c\rho^\beta} \quad (4.15)$$

onde c é uma constante e β um expoente no intervalo de 1 a 8. Como relatado na referência citada para $\beta = 6$ obtém-se moléculas com estabilidade pobre. Liszka *et al.* (1996), sugerem que após alguma experimentação os seguintes valores podem ser adotados: $\beta = 3$ e $c = 24$.

Convém observar que, ao contrário das funções de ponderação vistas na Seção 3.4, que são função do suporte através do parâmetro $r = \rho / d_m$, a expressão (4.15) não apresenta essa correlação. Mantendo o espírito de interpolação, outras funções de ponderação correlacionadas com o suporte poderiam ser sugeridas. Este assunto relaciona-se para futuras investigações.

A preocupação com a construção da nuvem de pontos nodais que compõem a molécula é central na formulação do Método das Diferenças Finitas Generalizadas, devido à instabilidade causada por moléculas desbalanceadas principalmente quando não se utiliza o procedimento de minimização de erro. Jansen (1972) propôs uma estratégia baseada no critério da menor distância que contém $m = 6$ pontos nodais. Este critério pode conduzir a moléculas muito assimétricas (Figura 4.1-a). No caso de uma malha ortogonal quadrada este critério conduz a equações de diferenças do Método das Diferenças Finitas tradicional, como ilustra a Figura 4.1-b, onde a numeração é decorrente da ordenação lexicográfica (menor distância em x e a seguir em y) dos pontos nodais.

Liszka e Orkisz (1980) propuseram uma estratégia baseada num critério denominado de *quatro quadrantes*, que seleciona os n pontos nodais mais próximos em cada quadrante. Como ilustra a Figura 4.2 a molécula fica mais balanceada, podendo, contudo, não utilizar alguns pontos nodais que se encontram dentro do suporte dm_i , definido como a máxima distância em relação ao pólo da nuvem dos pontos nodais que a compõem.

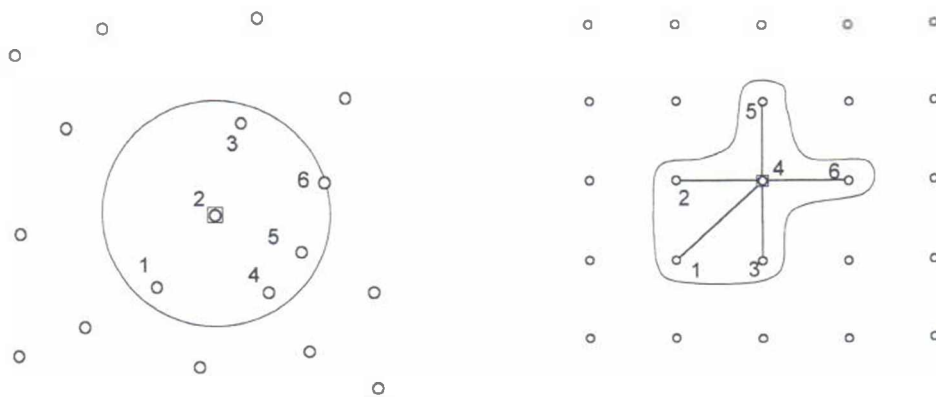


Figura 4.1 – Critério de mínimo de 6 pontos nodais mais próximos para construção da nuvem.

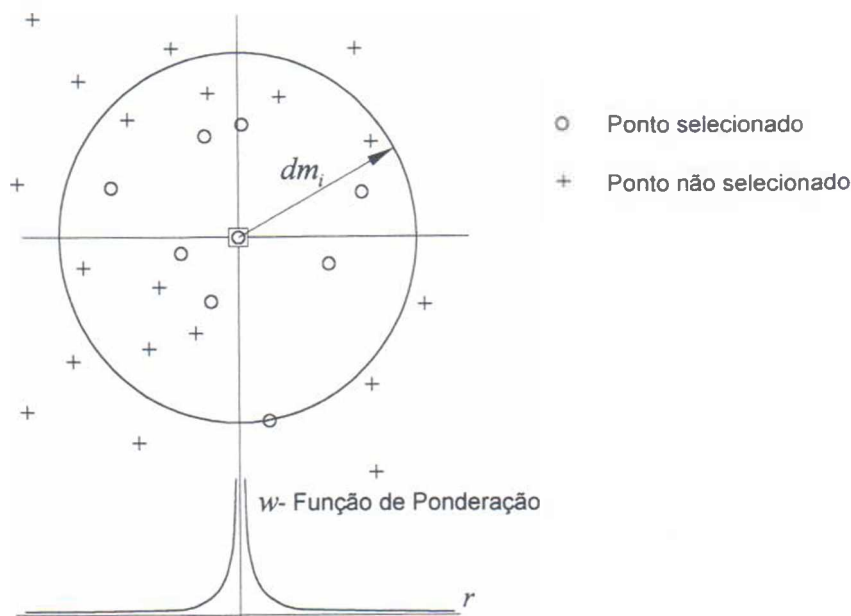


Figura 4.2 – Seleção da molécula pelo critério dos quatro quadrantes.

Convém perceber que na verdade o suporte definido pela função de ponderação $w_i = 1 / \rho^\beta$ é infinito. Na prática como o decaimento é muito pronunciado a maioria dos pontos com raio $\rho > dm_i$ possuem contribuição numérica próximo de zero. Talvez mais grave, como comenta Duarte (1995), é a não inclusão de alguns pontos pertencente ao suporte que poderiam levar a descontinuidade na função de aproximação definida pelo Método de Mínimos Quadrados Móvel. Contudo a experimentação numérica conduzida,

principalmente, por Lyszka *et al.* (1996), não permite detectar evidências de possíveis descontinuidades. Também este aspecto carece de novas investigações.

Lyszka *et al.* (1996) com o objetivo de melhorar o condicionamento das moléculas próximas do contorno propõem a inclusão de grau de liberdade rotacional referente à derivada normal como incógnita do problema.

As derivadas normais da função $u(x, y)$ são dadas por

$$\frac{\partial u(x, y)}{\partial n} = \frac{\partial u}{\partial x} n_x + \frac{\partial u}{\partial y} n_y \quad (4.16)$$

onde $n = (n_x, n_y)$ é o vetor normal ao contorno.

Como visto a função $u(x, y)$ expandida em série de Taylor a partir do ponto u_0 é dada por

$$u(x, y) = \mathbf{p}^T \mathbf{D}u(0) \quad (4.17)$$

Derivando (4.17) e substituindo em (4.16) chega-se a

$$\frac{\partial u}{\partial n} = \left[\frac{\partial \mathbf{p}^T}{\partial x} n_x + \frac{\partial \mathbf{p}^T}{\partial y} n_y \right] \mathbf{D}u(0) \quad (4.18)$$

com

$$\frac{\partial \mathbf{p}}{\partial x} = [0 \quad 1 \quad 0 \quad x \quad y \quad 0]^T$$

$$\frac{\partial \mathbf{p}}{\partial y} = [0 \quad 0 \quad 1 \quad 0 \quad x \quad y]^T$$

Substituindo obtém-se a expressão para as derivadas normais

$$\frac{\partial u}{\partial n} = \mathbf{p}_n^T \mathbf{D}u(0) \quad (4.19)$$

onde

$$p_n = \begin{bmatrix} 0 & n_x & n_y & xn_x & (yn_x + xn_y) & yn_y \end{bmatrix}^T \quad (4.20)$$

podendo ser interpretada como uma base de Taylor para as derivadas normais.

Aplicando a expressão anterior aos r pontos do contorno obtém-se

$$u_n = P_n Du(0) \quad (4.21)$$

onde

$$P_n = \begin{bmatrix} 0 & n_{1x} & n_{1y} & h_1 n_{1x} & (k_1 n_{1x} + h_1 n_{1y}) & k_1 n_{1y} \\ \vdots & & & & & \vdots \\ 0 & n_{rx} & n_{ry} & h_r n_{rx} & (k_r n_{rx} + h_r n_{ry}) & k_r n_{ry} \end{bmatrix}$$

$$u_n = \begin{bmatrix} \frac{\partial u_1}{\partial n} & \dots & \frac{\partial u_r}{\partial n} \end{bmatrix}^T$$

$$Du(0) = \begin{bmatrix} u_0 & \frac{\partial u_0}{\partial x} & \frac{\partial u_0}{\partial y} & \frac{\partial^2 u_0}{\partial x^2} & \frac{\partial^2 u_0}{\partial x \partial y} & \frac{\partial^2 u_0}{\partial y^2} \end{bmatrix}^T$$

O erro total será dado agora por duas parcelas. Uma referente aos pontos nodais do domínio $\mathbb{N}(Du(0))$ e outra referente aos pontos nodais do contorno expressa em termos matriciais

$$\mathbb{N}_n(Du(0)) = (P_n Du(0) - u_n)^T W_n(0) (P_n Du(0) - u_n) \quad (4.22)$$

onde W_n é a matriz diagonal com os valores das funções de ponderação adotada ligeiramente diferente como sendo $w_i = 1/c\rho^{\beta-1}$.

Efetuando a minimização do erro total, ou, em outras palavras, efetuando o procedimento de Mínimos Quadrados Ponderados chega-se

$$Du(0) = [A + A_n]^{-1} B u + [A + A_n]^{-1} B_n u_n \quad (4.23)$$

onde

$$\begin{aligned}
A &= P^T W(0) P \\
A_n &= P_n^T W_n(0) P_n \\
B &= P^T W(0) \\
B_n &= P_n^T W_n(0)
\end{aligned}$$

Colocando a expressão em termos das funções de forma vem

$$Du(0) = \varphi u + \varphi_n u_n \quad (4.24)$$

com

$$\begin{aligned}
\varphi &= [A + A_n]^{-1} B \\
\varphi_n &= [A + A_n]^{-1} B_n
\end{aligned}$$

Liszka *et al.* (1996) partem da seguinte expressão para realizar o procedimento de minimização

$$\begin{bmatrix} P \\ P_n \end{bmatrix} Du = \begin{bmatrix} u \\ u_n \end{bmatrix} \quad (4.25)$$

chegando a resultados idênticos aos apresentados.

A inclusão do grau de liberdade de rotação, chamado por Liszka ora de espectral, ora de Hermite, propicia uma melhora significativa nas moléculas que incluem pontos nodais do contorno.

A expressão anterior considerando um ponto nodal no contorno, pólo da molécula, e quatro pontos no domínio Figura 4.3, reduz a expressão anterior a

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & h_2 & \dots & k_2^2/2 & & \\ \vdots & & & \vdots & & \\ 1 & h_5 & \dots & k_5^2/2 & & \\ 0 & n_x & n_y & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} u_0 \\ \frac{\partial u_0}{\partial x} \\ \vdots \\ \frac{\partial^2 u_0}{\partial^2 y} \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_5 \\ \frac{\partial u_1}{\partial n} \end{bmatrix} \quad (4.26)$$

Como $u_0 = u_1$ a expressão anterior possui formato bastante similar com a expressão apresentada por Pulino (1989), e com resultados idênticos.

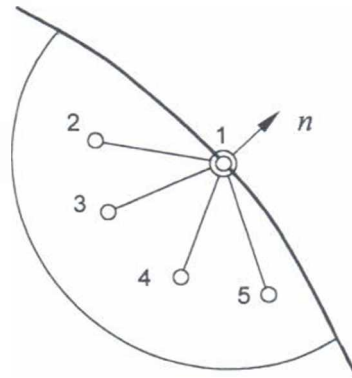


Figura 4.3 – Molécula do contorno com um ponto no contorno e quatro no domínio.

Apresenta-se a seguir uma variante da formulação de colocação do Método dos Resíduos Ponderados vista no Capítulo 2. Procedendo de forma similar com a equação de Poisson chega-se às seguintes expressões integrais

$$\begin{aligned}
 - \int_{\Omega} \frac{\partial w}{\partial x} \frac{\partial u}{\partial x} + \frac{\partial w}{\partial y} \frac{\partial u}{\partial y} d\Omega + \int_{\Gamma_u} w \frac{\partial u}{\partial n} d\Gamma &= \int_{\Omega} w u d\Omega - \int_{\Gamma_i} w \bar{q} d\Gamma \\
 \int_{\Gamma_u} \bar{w} u d\Gamma &= \int_{\Gamma_u} \bar{w} \bar{u} d\Gamma
 \end{aligned} \tag{4.27}$$

Utilizando a função delta de Dirac como função de ponderação, ou seja $w = \bar{w} = \delta$ e considerando a descontinuidade que ocorre quando esta função se depara com o contorno como visto no Capítulo 2, chega-se

$$\begin{aligned}
 - \left\{ - \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) \Big|_{\Omega} + \frac{\partial u}{\partial n} \Big|_{\Gamma} \right\} + \frac{\partial u}{\partial n} \Big|_{\Gamma_u} &= b|_{\Omega} - \bar{q}|_{\Gamma_i} \\
 u|_{\Gamma_u} &= \bar{u}|_{\Gamma_u}
 \end{aligned} \tag{4.28}$$

Simplificando vem

$$\begin{aligned} \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \Big|_{\Omega} - \frac{\partial u}{\partial n} \Big|_{\Gamma_i} &= b|_{\Omega} - \bar{q}|_{\Gamma_i} \\ u|_{\Gamma_u} &= \bar{u}|_{\Gamma_u} \end{aligned} \quad (4.29)$$

Observando-se que $\frac{\partial u}{\partial n} = \bar{q}$ em Γ_i , chega-se

$$\begin{aligned} \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \Big|_{\Omega} &= b|_{\Omega} \\ u|_{\Gamma_u} &= \bar{u}|_{\Gamma_u} \end{aligned} \quad (4.30)$$

Verifica-se que o contorno Γ_i , não aparece nas expressões acima. Considerando, agora, que as funções de forma do Método das Diferenças Finitas Generalizadas com termo incluindo a derivada normal vem da expressão

$$Du(0) = \varphi u + \varphi_n u_n \quad (4.31)$$

e portanto as derivadas tomam a forma

$$\begin{aligned} u_{,xx} &= \phi_{,xx} a + \phi_{n,xx} a_n \\ u_{,yy} &= \phi_{,yy} a + \phi_{n,yy} a_n \end{aligned} \quad (4.32)$$

onde, $a_n = \begin{Bmatrix} q_i \\ q_u \end{Bmatrix}$ são as derivadas normais no contorno Γ_i , q_i ; e no contorno Γ_u , q_u .

Introduzindo na expressão (4.30) obtém-se

$$\begin{aligned} [\phi_{,xx} + \phi_{,yy}] a \Big|_{\Omega} + [\phi_{n,xx} + \phi_{n,yy}] a_n \Big|_{\Omega} &= b|_{\Omega} \\ u|_{\Gamma_u} &= \bar{u}|_{\Gamma_u} \end{aligned} \quad (4.33)$$

ordenando

$$\begin{aligned} K_i &= \phi'_{,xx} + \phi'_{,yy} \Big|_{\Omega}, \quad f_i = b_i \Big|_{\Omega}, \quad U_i = \bar{u}_i \Big|_{\Gamma_u} \\ H_i &= H'_i + H''_i = \phi'_{n,xx} + \phi'_{n,yy} \Big|_{\Omega} \end{aligned} \quad (4.34)$$

Como q_i é o valor conhecido, prescrito, no contorno Γ_i , chega-se ao sistema matricial

$$\begin{aligned} \mathbf{K}\mathbf{a} + \mathbf{H}_u\mathbf{q}_u &= -\mathbf{H}_i\mathbf{q}_i + \mathbf{f} \\ \mathbf{G}^T\mathbf{a} &= \mathbf{U} \end{aligned} \quad (4.35)$$

ou seja

$$\begin{bmatrix} \mathbf{K} & \mathbf{H}_u \\ \mathbf{G}^T & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \mathbf{a} \\ \mathbf{q}_u \end{Bmatrix} = \begin{Bmatrix} -\mathbf{H}_i\mathbf{q}_i + \mathbf{f} \\ \mathbf{U} \end{Bmatrix} = \begin{Bmatrix} \mathbf{F} \\ \mathbf{U} \end{Bmatrix} \quad (4.36)$$

A formulação acima permite a construção do Método das Diferenças Finitas Generalizadas no mesmo arcabouço do Método dos Resíduos Ponderados. Como estão presentes as derivadas normais nesta formulação pode-se considerar a existência de pontos gêmeos nos pontos de singularidade do domínio.

A aplicação computacional das funções de forma deduzidas acima será realizada através de procedimento de especialização de análise orientada a objetos. Assim a Figura 4.4 apresenta as classes necessárias para a determinação das funções de forma baseadas em expansão de série de Taylor.

Taylor e **TaylorN** são as bases de funções correspondentes às expressões (4.2) e (4.20), respectivamente. **InvPow** é a classe que determina a função de ponderação inversa da potência expressão (4.15), que também consta da Figura 3.16. **FourQuadrants** é a estratégia de construção de nuvens dos quatro quadrantes representada na Figura 4.2.

As funções de forma por expansão em série de Taylor, denominadas de **MLS_Taylor** são, como dito, uma especialização do Método dos Mínimos Quadrados com nuvem centrada num pólo, ponto nodal do domínio discretizado, que foi definida como **MLS_UNI** na Figura 3.14. A classe **MLS_Taylor** herda de **MLS_UNI** todos os dados e métodos, substituindo alguns de modo a considerar a especialização proposta. Esta classe também está representada na Figura 4.4.

A modelagem do Método das Diferenças Finitas será vista mais adiante na seção 4.7.

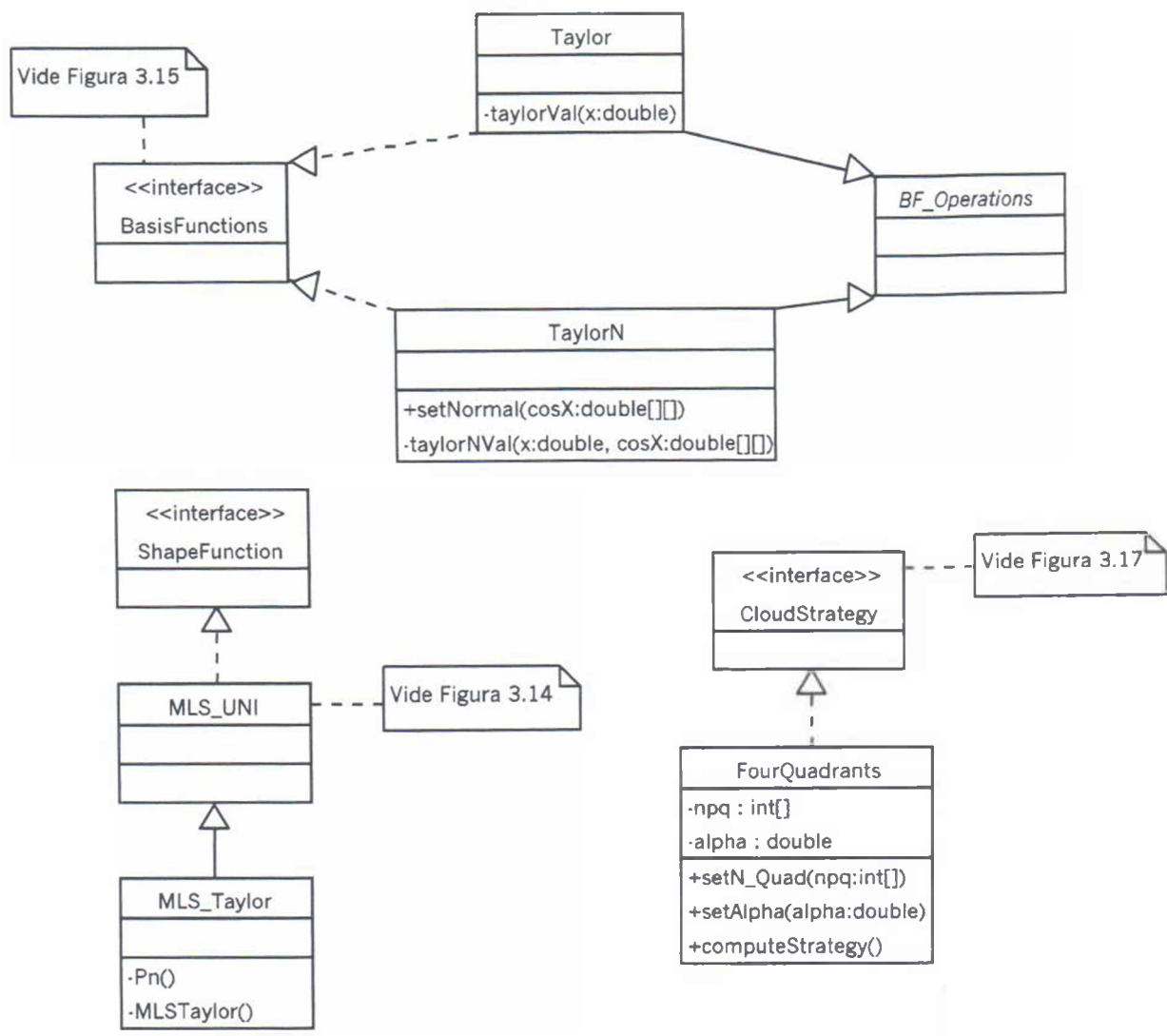


Figura 4.4 – Modelagem das classes para obtenção da função de forma por expansão em série de Taylor.

4.3 - MÉTODO DE GALERKIN SEM MALHA

Belytschko *et al.* (1994) apresentaram um método sem malha denominado “*Element Free Galerkin*” (Método de Galerkin sem Malha). O método emprega como procedimento para construção das aproximações locais o Método de Mínimos Quadrados Móvel, Seção 3.2, que determina funções de forma a partir de uma nuvem dispersa e sem conexão prévia delimitada por um suporte definido por uma função de ponderação. E como procedimento para discretização do domínio, emprega o critério de Galerkin numa expressão variacional (formulação fraca).

Como salientado por Belytschko, essas idéias foram implementadas inicialmente por Nayroles *et al.* (1992), *apud* Belytschko *et al.* (1994), como uma formulação para generalização do Método dos Elementos Finitos. O procedimento de Nayroles denominado de Método de Elemento Difuso (“*Diffuse Element Method*”) continha algumas limitações que foram superadas pela implementação de Belytschko.

Estes trabalhos desencadearam uma intensa pesquisa nos últimos anos com o objetivo de se superar o paradigma da construção de uma malha para se discretizar um domínio.

Embora não observado por Nayroles, as interpolações usadas são as obtidas pelo Método de Mínimos Quadrados Móvel de Lancaster e Šalkauskas (1981). Como visto estas funções de aproximação tem a forma

$$\phi^T(x) = p^T(x)A^{-1}(x)B(x) \quad (4.37)$$

Por se tratar de um procedimento de ajustamento por aproximação, as funções de forma não atendem à condição de Kronecker, ou seja, não alcançam o valor unitário sobre os pontos nodais. Somente com a utilização de funções de ponderação singulares o ajustamento torna-se interpolado, passando pelo valor atribuído a um ponto nodal. Isto é uma desvantagem para os métodos do tipo Galerkin já que dificulta a imposição da condição de contorno essencial e a aplicação de carregamentos pontuais. Diversos enfoques foram desenvolvidos para imposição da condição de contorno essencial.

Belytschko *et al.* (1994) no artigo de apresentação do Método de Galerkin sem Malha, adota o enfoque de utilizar uma formulação variacional impondo as condições de contorno através do método de multiplicadores de Lagrange. Essa abordagem conduz a um sistema de equações idêntico ao obtido pelo Método dos Resíduos Ponderados, sob determinadas condições. Esta formulação conduz a resultados bastante precisos quando comparado com outras formas de imposição das condições de contorno, embora conduza para um problema auto-adjunto linear a um sistema com matriz de coeficientes não positiva definida e nem apresenta configuração em banda.

Lu, *et al.* (1995) apresentaram formulação baseada num princípio variacional modificado, obtido a partir da dedução do significado físico do multiplicador de Lagrange, que são as trações ao longo do contorno Γ_u , para um problema de elasticidade. Com isto o sistema de equações torna-se simétrico e em banda. Este método mostrou-se menos preciso que o método de imposição baseado em multiplicadores de Lagrange.

Krongaux e Belytschko (1996) conduzem a imposição de condições de contorno através do acoplamento de elementos finitos na região do contorno essencial, que é aplicada do modo padrão desse método. Esta técnica pode ser utilizada em outros métodos de discretização sem malha.

Kaljević e Saigal (1997) utilizaram uma característica do método de mínimos quadrados móvel (Lancaster e Šalkauskas, 1981) que é a capacidade de gerar uma interpolação caso a função peso possua valor infinito para $x = x_i$. Como mencionado no Capítulo 3, essa singularidade pode ser contornada com alguma manipulação algébrica. Esta formulação possui boa precisão e conduz a um sistema de equações positivo definido e em banda.

Como visto no Capítulo 2, a construção das equações discretas é feita a partir do Método dos Resíduos Ponderados. Esta abordagem prescinde da existência de um funcional relativo às equações diferenciais que regem um problema sendo por isso um método mais geral no âmbito da mecânica computacional. Na forma apresentada aqui a imposição da condição de contorno essencial resulta num sistema de equações similar ao obtido pela via variacional com imposição de condição de contorno via multiplicadores de Lagrange

A solução das integrais necessárias à determinação dos coeficientes resultantes da discretização da primeira variação do funcional é efetuada geralmente por quadratura de Gauss-Legendre. Como o domínio Ω é discretizado apenas por pontos nodais, Belytschko estabeleceu uma grade subjacente de células que são totalmente independentes dos pontos nodais, como na Figura 4.5. Em cada célula a integração será efetivada se o ponto gaussiano estiver dentro do polígono que representa o contorno do domínio.

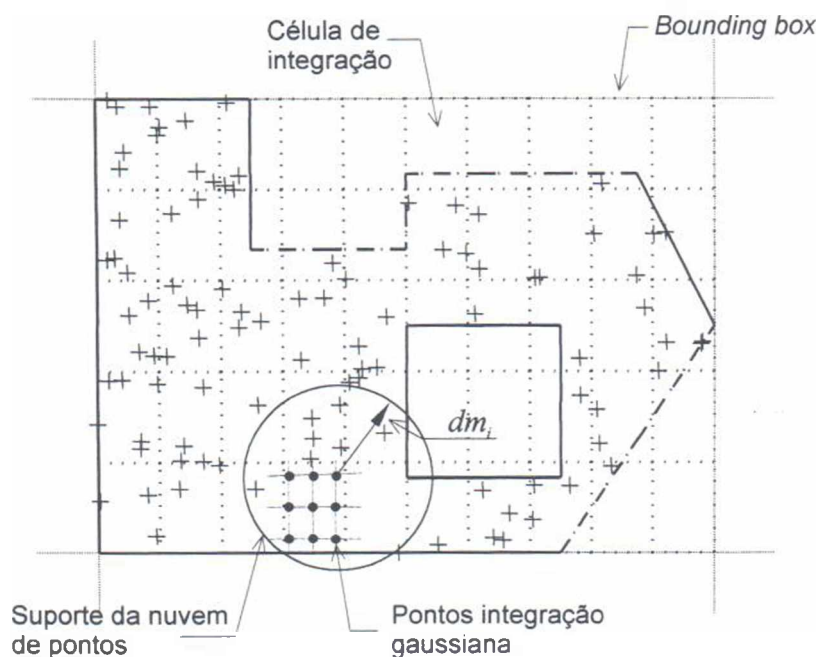


Figura 4.5 – Estrutura de células de integração independentes dos pontos nodais.

Em algumas apresentações esse arranjo de células é utilizado também para a identificação dos pontos nodais sob o suporte que configura o subdomínio de um ponto gaussiano. Na verdade a questão da identificação dos pontos sob domínio de influência de uma dada coordenada é crucial no tratamento de problemas de grande porte. Estruturas de dados em árvores (de Berg *et al.*, 2000) e ordenação lexicográfica (Goodrich e Tamassia, 2001) são possibilidades a estudar em futuros trabalhos.

Belytschko *et al.* (1994) propuseram para a estimativa do número de células, $n_c \times n_c$, a seguinte expressão

$$n_c = \sqrt{n_p} \quad (4.38)$$

onde n_p é o número total de pontos nodais da discretização do domínio.

Outro ponto relevante é o estabelecimento do número de pontos gaussianos por célula. Belytschko, ainda, propôs o número de pontos de integração gaussiana $n_g \times n_g$ adotados pela fórmula.

$$n_g = \sqrt{N} + 2 \quad (4.39)$$

onde N é o número de pontos nodais no interior da célula.

O número de pontos de integração gaussiana é geralmente alto, 4×4 ou mais, uma vez que as funções de forma do método dos Mínimos Quadrados Móvel possuem comportamento de polinômios de ordem elevada, em virtude do fato de que as funções de ponderação exercem influência marcante na sua construção. Dolbow e Belytschko (1998) estudaram a relação entre o suporte das funções de ponderação e as células de integração das integrais que formam o sistema de equações, concluindo que se obtém maior precisão na quadratura quando se faz coincidir a célula com o suporte.

Quanto às integrais do contorno são também obtidas por quadratura gaussiana em uma dimensão. As células são consideradas, geralmente, como limitadas pelos pontos nodais da discretização do polígono do contorno do domínio.

Alguma pesquisa vem sendo feita com o intuito de se evitar a estrutura de células, na busca de um procedimento absolutamente sem malha. Beissel e Belytschko (1996) propuseram um procedimento para avaliação das integrais somente nos pontos nodais, utilizando um funcional com termo adicional para estabilizar a solução numérica.

Duflot e Nguyen-Dang (2002) propuseram uma estratégia que prescinde da grade subjacente de células. A função a integrar é multiplicada por uma função partição da unidade e a quadratura é efetuada sobre todos os subdomínios que se sobrepõem cobrindo todo o domínio. Ou seja

$$\int_{\Omega} f(x) d\Omega = \sum_{k=1}^l \int_{\Omega \cap \Omega_k} \Psi(x) f(x) d\Omega \quad (4.40)$$

A função Ψ_k possui um subdomínio associado Ω_k de tal forma que $\Psi_k = 0$ para pontos fora de Ω_k . Além disso a função Ψ_k é uma partição da unidade, ou seja atende a $\sum_{k=1}^l \Psi_k(x) = 1$. A partição da unidade utilizada são funções de forma de Shepard obtidas a partir do método de mínimos quadrados móvel com base $\mathbf{p}^T = [1]$. O valor da integral de $f(x)$ sobre Ω é obtido através da quadratura de Gauss das funções $\Psi(x) f(x)$ sobre os sub-domínios $\Omega \cap \Omega_k$, ($k = 1 \dots l$) e adicionando essas contribuições. Esta estratégia exige um grande número de pontos nodais para a obtenção de uma boa precisão.

A nuvem de pontos sob o domínio de influência, ou seja sob o suporte, de um ponto gaussiano é construída utilizando-se uma estratégia de se escolher o conjunto mínimo de pontos nodais que constroem um polígono ao redor do ponto gaussiano. Este critério, proposto por Belytschko, garante alguma simetria à nuvem de pontos nodais. Essa construção utiliza-se das técnicas da geometria computacional (Preparata e Shamos, 1985). Obviamente os critérios citados na Seção 3.6 também podem ser utilizados, sendo de implementação mais simples.

Do ponto de vista computacional as classes já modeladas permitem a construção do Método de Galerkin sem Malha como será visto na Seção 4.7.

4.4 - HP-CLOUDS – MÉTODO SEM MALHA HIERÁRQUICO

Duarte e Oden (1995 e 1996) propuseram um método cujo principal objetivo é a incorporação de adaptabilidade hierárquica para uma nuvem dispersa de pontos. A apresentação do método *Hp-Clouds* é desenvolvida numa estrita linguagem formal matemática com demonstrações abundantes sobre consistência, convergência e tratamento de erro. Apesar do forte apelo do tratamento hierárquico semelhante ao utilizado pelo Método dos Elementos Finitos e sem o inconveniente da malha, o Método *Hp-clouds* tem sido pouco reportado na literatura. Talvez o formalismo matemático e uma abordagem bastante axiomática não facilite o entendimento de seus fundamentos. A apresentação a seguir, como de resto em todo este trabalho, buscará uma abordagem mais construtiva.

Um aspecto do Método *Hp-clouds* foge do escopo estabelecido para este trabalho. O estudo dos indicadores de erro que permitem a construção de métodos auto-adaptativos será listado como trabalho futuro.

Seja um domínio bidimensional discretizado com pontos nodais (com atributos geométricos do domínio e parâmetros do problema) espalhados de forma arbitrária. Centrado em cada ponto nodal será estabelecido um suporte (circular ou retangular) que devem, em seu conjunto recobrir todo o domínio, como ilustrado na Figura 4.6.

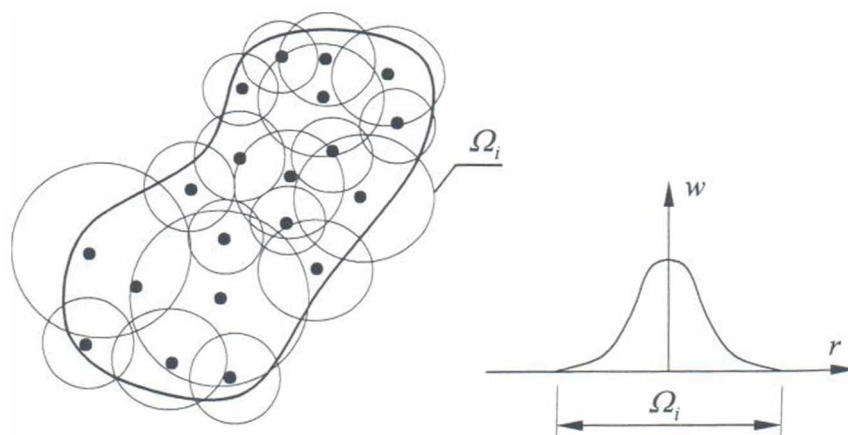


Figura 4.6 – Domínio discretizado por pontos nodais com suportes circulares.

Obviamente estabelecer a dimensão do suporte é crucial, devendo incorporar requisitos matemáticos, com sua construção guiada por algoritmos da Geometria

Computacional. No momento, busca-se estabelecer uma função que de forma aproximada solucione um problema de valor de contorno nesse domínio. Uma aproximação desse tipo em função de parâmetros do problema estabelecidos nos pontos nodais vem sendo apresentada pela expressão

$$u^h(x) = \sum_{i=1}^{n_p} \phi_i u_i = \boldsymbol{\phi}^T \mathbf{u} \quad (4.41)$$

Usualmente as funções ϕ_i são determinadas para um ponto x a partir de um suporte centrado no próprio ponto x , como visto no Capítulo 3. Neste caso busca-se uma aproximação em x construída a partir de suportes centrados nos pontos nodais x_i que discretizam o problema. Esta variante foi apenas ilustrada na Seção 3.2. Neste caso o funcional a minimizar será construído a partir dos suportes dos pontos nodais que são interceptados pelo ponto x , como rerepresentado na Figura 4.7.

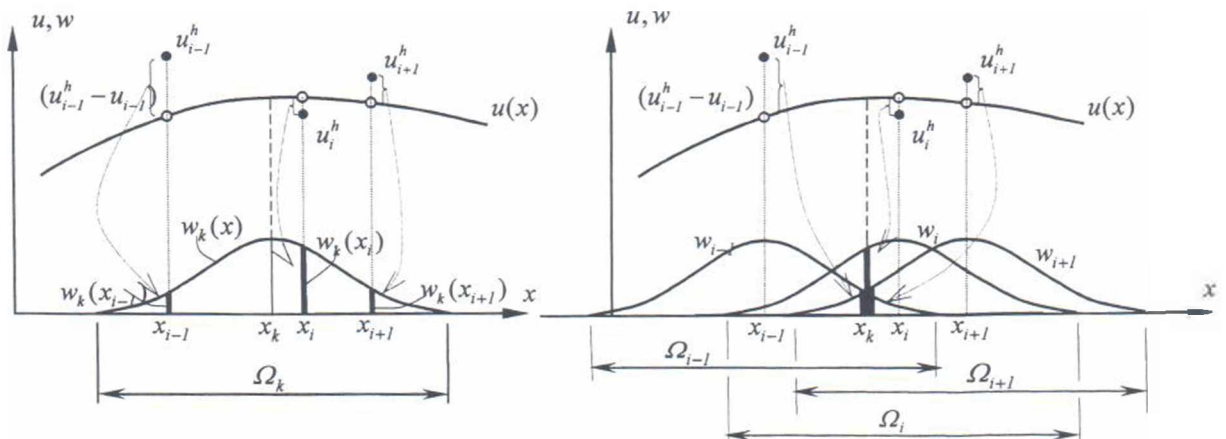


Figura 4.7 – Funções de forma geradas a partir de um suporte e de suportes múltiplos.

O funcional agora é dado pela expressão

$$E(u, x) = \sum_{i=1}^n w_\alpha(x) (u^h(x_i, x) - u_i)^2 \quad (4.42)$$

onde $w_\alpha(x) \equiv w_\alpha(x - x_\alpha)$ sendo a função de ponderação em $x_\alpha \in x_i$ e avaliada no ponto x . Antes, nas demais aplicações dos Métodos sem Malha, a função de ponderação era

estabelecida com suporte centrado no ponto x e com valores da função de ponderação avaliados nos x_i pontos nodais abrangidos pelo suporte de x .

A minimização do funcional (4.42) leva basicamente à mesma expressão para as funções de forma ϕ_i , dada por

$$\phi_\alpha(x) = \mathbf{p}^T \mathbf{A}^{-1}(x) \mathbf{p}(x_\alpha) w_\alpha(x - x_\alpha) \quad (4.43)$$

onde $\mathbf{A}(x) = \mathbf{P}^T(x_i) \mathbf{W}(x_\alpha - x_i) \mathbf{P}(x_i)$, os demais termos foram apresentados no Capítulo 3.

A expressão (4.43) permite o traçado da função de forma para um determinado ponto nodal α . De fato as funções de forma para pontos nodais apresentadas no Capítulo 3 foram traçadas com a utilização desta expressão através do algoritmo apresentado na Tabela 4.1.

Tabela 4.1 – Algoritmo para a construção do traçado de funções de forma de um ponto nodal.

<p>Input: x_i - pontos nodais que discretizam o domínio. x - ponto do domínio. α - índice do ponto x_i que se deseja traçar a curva.</p> <p>Caso $x \notin \text{suporte}(x_\alpha)$, $\phi_\alpha = 0$, retorne.</p> <p>Senão procure $S(x)$ conjunto dos pontos nodais x_i cujos suportes interceptam x.</p> <p>Calcule $\mathbf{A}(x) = \mathbf{P}^T(x_i) \mathbf{W}(x_\alpha - x_i) \mathbf{P}(x_i)$</p> <p>Calcule $\phi_\alpha(x) = \mathbf{p}^T \mathbf{A}^{-1}(x) \mathbf{p}(x_\alpha) w_\alpha(x - x_\alpha)$</p> <p>Output: valor de $\phi_\alpha(x)$</p>

Como pode-se perceber a viabilidade de inversão de $\mathbf{A} = \mathbf{P}^T \mathbf{W} \mathbf{P}$ está sujeita a que a ordem da base deve ser no mínimo igual ao número de suportes de x_i interceptados por x . Esta restrição deve ser considerada, portanto, no algoritmo de construção dos suportes que se verá adiante.

De modo a se obter um enriquecimento hierárquico tipo p (de ordem polinomial), Duarte e Oden propõem a substituição da função de forma ϕ_i por outra que toma a forma

$$\Psi_i = \phi_\alpha^k L^p \quad (4.44)$$

onde ϕ_α^k é a função oriunda do Método de Mínimos Quadrados Móvel com k sendo a ordem da base empregada; e L^p é a base de funções polinomiais hierárquicas. Duarte e Oden (1995) sugerem funções baseadas em polinômios de Legendre ou monômios.

Do ponto de vista matemático esta operação, de produto de funções linearmente independentes produzir um conjunto de novas funções também linearmente independentes, somente é possível devido a que as funções ϕ_i^k são o que se denomina de **partição da unidade**. E, portanto, o conjunto L^p pode ser qualquer conjunto de funções linearmente independentes.

Uma partição da unidade atende à seguinte condição

$$\sum_{\alpha=1}^n \Psi_\alpha(x) = 1 \quad \forall x \in \Omega \quad (4.45)$$

Lancaster e Šalkauskas (1981) demonstram que as funções produzidas pelo Método dos Mínimos Quadrados Móvel se constituem em uma partição da unidade.

Nos experimentos numéricos realizados observou-se uma relativa independência do erro com a ordem k da base utilizada. Assim, adotou-se $k = 0$, ou seja a base unitária, o que leva o Método de Mínimos Quadrados Móvel a produzir as funções de Shepard. Portanto a partição da unidade que será implementada é a função de Shepard dada por

$$\phi_\alpha^0 = \frac{w_\alpha(x)}{\sum_{\beta=1}^n w_\beta(x)} \quad (4.46)$$

onde $w_\alpha(x) = w_\alpha(x - x_\alpha)$ é uma função de ponderação com valor diferente de zero e positivo num suporte de raio h_α . Note-se que em um dado ponto x devem se superpor n suportes de modo a ocorrer um fracionamento da unidade entre os pontos nodais. Caso contrário, se um dado ponto do domínio for coberto por apenas um suporte, não ocorrerá a divisão da unidade e a função de forma ϕ_α terá valor unitário nessa região. Assim, para que a função partição da unidade produza efeito deverá ocorrer cobertura em cada ponto do domínio por dois suportes, no mínimo.

Esta observação, não enfatizada por Duarte e Oden, foi percebida por Mendonça *et al.* (2000) numa aplicação do Método *Hp-Clouds* à viga de Timoshenko. Embora computacionalmente muito fáceis de produzir as funções de Shepard podem produzir a resultados inesperados, conduzindo a derivada nula sobre os pontos nodais.

Na construção das funções $\Psi_i = \phi_\alpha^k L^P$, onde ϕ_α^k é a função partição da unidade do Método de Mínimos Quadrados Móvel, e L^P é um conjunto de funções linearmente independentes requer algum cuidado de modo que a nova base, produto das anteriores, conserve independência linear. Seja por exemplo o seguinte grupo de funções

$$\begin{array}{ll} \phi^k \equiv \phi^0 : \{1\} & \text{Funções de Shepard} \\ L^P \equiv L^2 : \{1 \ x \ x^2\} & \text{Monômios} \end{array} \quad (4.47)$$

Como as funções de Shepard são geradas por uma base unitária, então a base linearmente independente que gera as funções $\Psi_i = \phi_\alpha^k L^P$ deverá ser

$$\phi^0 L^2 : \{1 \ x \ x^2\} \quad (4.48)$$

Assim a base de monômios deverá excluir o termo $\{1\}$. Essa análise deve ser realizada para cada grupo de família de funções utilizadas. Tendo sido feita aqui para um caso simples, mas suficientemente geral.

Pertinências matemáticas à parte, para o caso das funções de Shepard o conjunto de funções que gera Ψ_i não deverá incluir em L^P o termo $\{1\}$, uma vez que este

já consta da base de funções ϕ^0 (Funções de Shepard). Assim uma aproximação da solução pelo Método *Hp-Clouds* toma uma forma ligeiramente diferente, isto é

$$u^{hp}(x) = \sum_{\alpha=1}^n \sum_{i=1}^{m(\alpha)} a_{\alpha i} \left(\phi_{\alpha}^0(x) L_{i\alpha}^p(x) \right) \quad (4.49)$$

onde α é o índice que percorre os n pontos nodais; i é o índice dos termos da função base hierárquica; $a_{\alpha i}$ são coeficientes em número de i para cada ponto nodal; $m(\alpha)$ é o número de termos variável da base para cada ponto nodal α ; $\phi_{\alpha}^0(x)$ é a função de Shepard, adotada como partição da unidade, para o ponto nodal α ; e $L_{i\alpha}^p$ é o conjunto de funções hierárquicas a empregar.

Alguns comentários são oportunos. Primeiro, deve-se notar que cada ponto pode ter um número diferente de termos da base $L_{i\alpha}^p$, garantindo-se uma variação de ordem p por ponto nodal. Segundo, a função de Shepard, ϕ_{α}^0 , partição da unidade foi adotada por conveniência computacional, podendo outras ordens serem utilizadas. Poderia ser adotada, inclusive, funções lineares como as usadas nos elementos finitos isoparamétricos que também são uma partição da unidade. E, terceiro, não é possível com a expressão acima aproximar uma função com a utilização de uma restrição de interpolação. O número abundante de coeficientes, maior do que um, por ponto nodal requer outro critério. Por exemplo, o erro em relação à solução exata pode ser ortogonalizado com um conjunto de funções (Prenter, 1975)

$$(f - \tilde{f}, \Psi_i) = \int_a^b [f(t) - \tilde{f}(t)] \Psi_i(t) dt = 0 \quad 1 \leq i \leq n \quad (4.50)$$

onde f é a função conhecida, $\tilde{f} = c_1 \phi_1(t) + \dots + c_n \phi_n(t)$. Usualmente adota-se $\Psi_i = \phi_i$. Esta formulação corresponde à projeção da norma $L_2 = \|f - \tilde{f}\| = \int_a^b [f(t) - \tilde{f}(t)]^2 dt$ no espaço definido pela base $\Psi_i = \phi_i$. Ora, esta expressão é bastante similar à apresentada no Capítulo 2 para o critério de Galerkin.

Retomando a expressão das funções de forma do Método *Hp-Clouds*, $\Psi_i = \phi_\alpha^k L^p$, suas derivadas são dadas por

$$\begin{aligned} (\Psi_i)_s &= (\phi_\alpha^0)_s \cdot (L^p) + (\phi_\alpha^0) \cdot (L^p)_s \\ (\Psi_i)_{st} &= (\phi_\alpha^0)_{st} \cdot (L^p) + (\phi_\alpha^0)_s \cdot (L^p)_t + (\phi_\alpha^0)_t \cdot (L^p)_s + (\phi_\alpha^0) \cdot (L^p)_{st} \end{aligned} \quad (4.51)$$

A primeira derivada da função de Shepard é dada por

$$\begin{aligned} \phi_{\alpha,s} &= A^{-1} B_s + A_s^{-1} B \\ \phi_{\alpha,s} &= \frac{w_{\alpha,s}}{\sum_{\beta=1}^n w_\beta} - \frac{\sum_{\beta=1}^n w_{\beta,s}}{\left(\sum_{\beta=1}^n w_\beta\right)^2} w_\alpha \end{aligned} \quad (4.52)$$

Convém alertar que caso $n = 1$, na região recoberta por apenas um suporte a derivada será nula.

A derivada da função de Shepard em relação a uma segunda direção t é dada por

$$\begin{aligned} \phi_{\alpha,sts} &= A_{st}^{-1} B + A_s^{-1} B_t + A_t^{-1} B_s + A^{-1} B_{st} \\ \phi_{,st} &= 2 \frac{\sum w_s \sum w_t}{(\sum w)^3} w - \frac{\sum w_{st}}{(\sum w)^2} w - \\ &\quad - \frac{\sum w_s}{(\sum w)^2} w_t - \frac{\sum w_t}{(\sum w)^2} w_s + \frac{1}{\sum w} w_{st} \end{aligned} \quad (4.53)$$

De novo é fácil perceber que caso em um determinado ponto ocorra somente o suporte de um ponto nodal, o valor da segunda derivada da função de Shepard será nula, ou seja, $\phi_{\alpha,sts} = 0$. Neste caso as derivadas da função de forma do Método *Hp-Clouds* serão dadas por

$$\begin{aligned} \Psi_{i,s} &= \phi_\alpha L_{,s}^p \\ \Psi_{i,st} &= \phi_\alpha L_{,st}^p \end{aligned} \quad (4.54)$$

Portanto de modo a preservar a qualidade numérica do método deve-se utilizar funções hierárquicas L^p com $p \geq 1$, eliminando-se assim, a possibilidade de derivada nula decorrente do fato de um ponto estar recoberto por apenas um suporte.

No presente trabalho adotou-se a base de monômios, já definida no Capítulo 3, como conjunto de funções hierárquicas L^p . Assim a avaliação da função e de suas derivadas é realizada pelos mesmos objetos, e classes definidas naquele capítulo.

A aplicação das funções de forma hierárquicas, definidas a partir do produto de uma partição da unidade por uma base de funções com ordem crescente p , a problemas de valor de contorno é realizada no âmbito do critério de ortogonalização de Galerkin. Duarte e Oden manifestam que se trata de conveniência, outros critérios podem ser empregados, como o de colocação.

A única diferença na implementação do procedimento de Galerkin, em relação aos demais métodos apresentados neste trabalho, é que o resultado da solução do sistema de equações não são as incógnitas dos pontos nodais, mas sim parâmetros que quando combinados linearmente com a partição da unidade e a base hierárquica resultam nas procuradas incógnitas.

Na determinação da função de Shepard qualquer das funções de ponderação listadas no Capítulo 3 poderia ser utilizada. Duarte (1996), e Garcia *et al.* (2000) apresentam uma representação gráfica de função de ponderação *B-spline* quártica, portanto C^3 , e que possui a vantagem de ter derivadas nulas quando a função se anula, além de ser polinomial. A Figura 4.8 apresenta o gráfico da função *B-spline* bem como das expressões polinomiais que permitem a sua construção. A dedução foi realizada a partir de exposição sobre o tema, constante de Prenter (1975), com a utilização de cinco pontos nodais equidistantes e com um fator de normalização a soma dos valores da *B-spline* quártica nos pontos nodais.

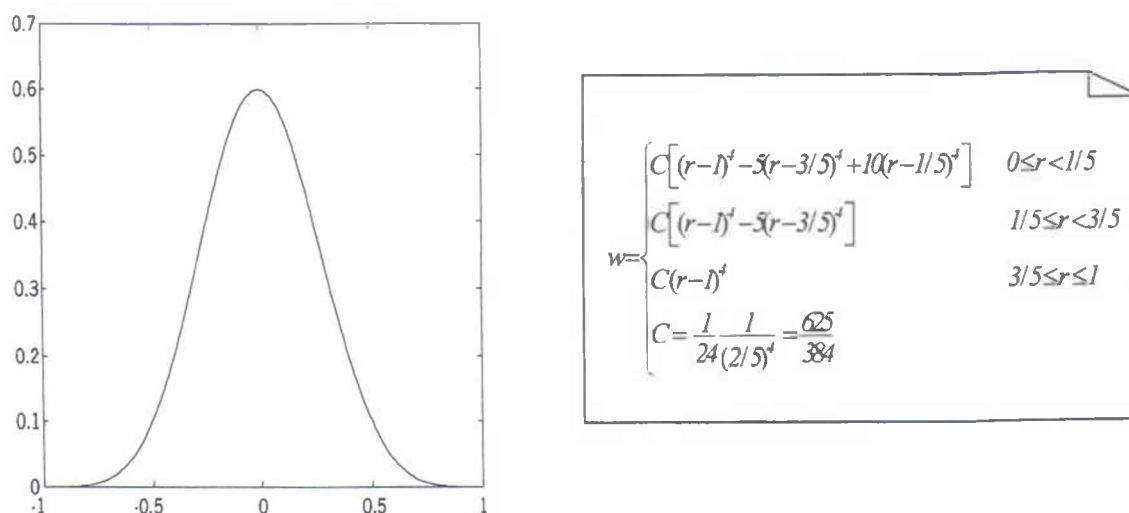


Figura 4.8 – Função de ponderação *B-Spline* quártica.

O Método *Hp-Clouds* faz uma extensa utilização de técnicas e algoritmos de Geometria Computacional. Dois algoritmos são essenciais para a implementação numérica e computacional do método. O primeiro trata de um procedimento de recobrimento de todo o domínio com os suportes dos pontos nodais. Além de garantir o recobrimento do domínio que é antes de tudo uma imposição matemática o algoritmo deve estabelecer de forma automática a dimensão do suporte de cada ponto nodal.

Para o caso de distribuição regular ortogonal, em grelha, dos pontos nodais o algoritmo é bastante simplificado, bastando estabelecer como dimensão do suporte metade da distância diagonal entre os pontos nodais. Neste caso boa parte da área junto ao ponto nodal será recoberta por apenas um suporte, resultando em uma função de Shepard unitária. Outra possibilidade seria arbitrar para cada ponto nodal um recobrimento igual à distância ao ponto mais próximo, o que garante uma sobreposição mais elevada dos suportes. A Figura 4.9 ilustra esse algoritmo onde a dimensão do suporte é dada por $h_\alpha = \beta \max(d_x, d_y)$.

Para uma distribuição irregular ou aleatória, em duas ou três dimensões, um algoritmo dessa natureza não é tão simples. Duarte e Oden (1996) propõem um algoritmo baseado em busca binária que estabelece a dimensão do suporte como a distância ao ponto nodal mais próximo, no caso de $k = 0$, ou seja partição da unidade função de Shepard. Este algoritmo pode falhar se a distribuição ao acaso redundar em forte aglomeração de pontos

nodais (*cluster*), como mostrado na Figura 4.10. Neste trabalho optou-se por utilizar um algoritmo baseado na construção de uma triangulação válida restringida pelo domínio e adotando-se como raio do suporte h_α a distância ao ponto nodal mais distante. Com isto evita-se a falha apontada e chega-se a uma sobreposição de suportes bastante elevada.

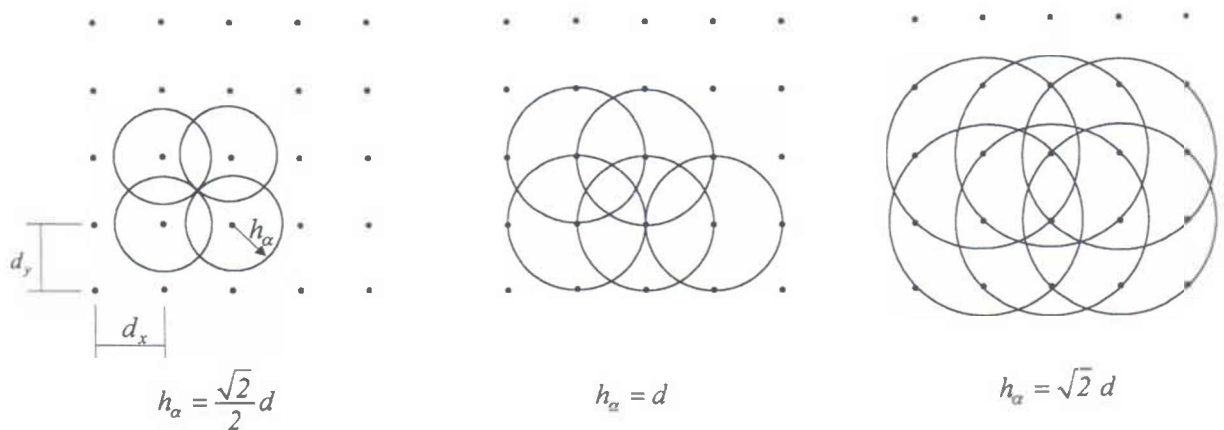


Figura 4.9 – Algoritmo de recobrimento para distribuição regular de pontos nodais.

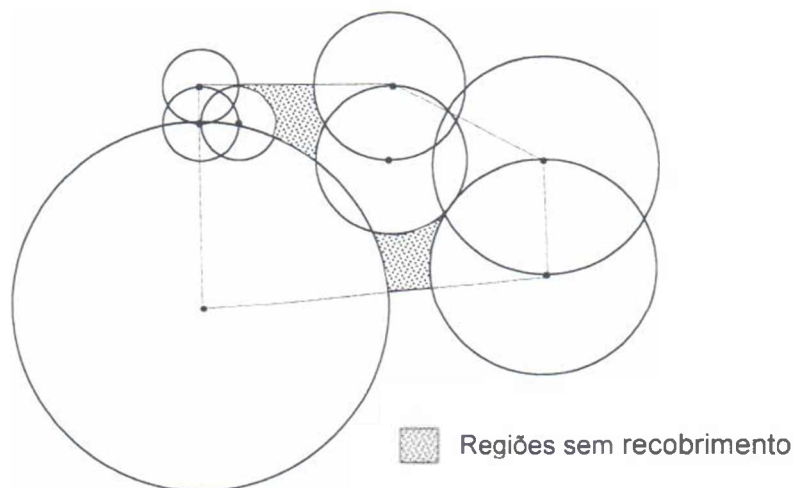


Figura 4.10 – Distribuição em *cluster* com algoritmo de recobrimento de mínima distância.

O algoritmo de triangulação gulosa (*greedy triangulation*) com restrição, apresentado na Tabela 4.2, foi elaborado baseado nos conceitos existentes em Preparata e Shamos (1985), e em de Berg *et al.* (2000). A verificação de interceptação de segmentos utiliza algoritmo baseado em procedimento proposto por Sunday (2003), assim como a verificação se um segmento está dentro de um polígono. Após a triangulação é construída a

estrutura de dados para cada triângulo com sua respectiva conectividade. Finalmente, busca-se o maior segmento dentre os que partem de um ponto nodal para dimensão do suporte. A Figura 4.11 ilustra duas fases do algoritmo.

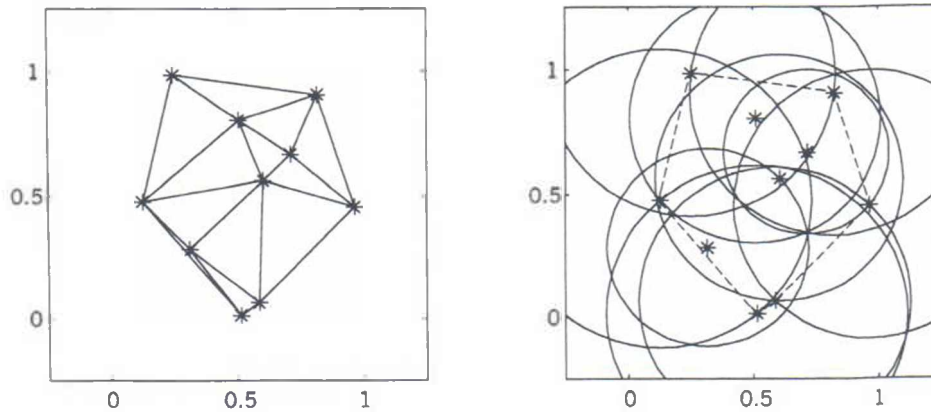


Figura 4.11 – Algoritmo de recobrimento baseado em triangulação.

Tabela 4.2 – Algoritmo de recobrimento do domínio.

<p>Input : - pontos nodais. - geometria de restrição.</p> <p>Construa pool de lados com combinação $\binom{n}{2}$ lados e ordenar crescente pela dimensão.</p> <p>Inclua lados da geometria de restrição na estrutura de dados da triangulação.</p> <p>Loop sobre o pool de lados a partir do menor lado Caso não intercepte nenhum lado Inclua na triangulação. Senão descartar lado</p> <p>End.</p> <p>Loop sobre a triangulação Caso algum lado esteja fora da geometria de restrição (furos e reentrâncias) descartar lado</p> <p>End.</p> <p>Loop sobre os pontos nodais Busque triângulos que incidem no ponto nodal Determine o suporte como sendo o maior segmento que parte do ponto nodal</p> <p>End.</p> <p>Output : suporte de cada ponto nodal.</p>

O outro algoritmo fundamental deve responder à seguinte questão: dado um ponto no domínio (para integração gaussiana, por exemplo) quais pontos nodais possuem suportes que interceptam esse ponto. Com isto as funções de forma e derivadas podem ser determinadas para construção das matrizes que modelam a discretização. O algoritmo desenvolvido foi concebido a partir do algoritmo de buscas binária, de Swegle (1994), utilizado no âmbito do método SPH (*Smooth Particle Hydrodynamic*), e que também foi empregado por Duarte. Em essência o algoritmo procura para cada direção cartesiana quais suportes interceptam o ponto e a seguir faz uma verificação cruzada com as demais direções sobre a possibilidade de novas intersecções. A resposta buscada são as intersecções existentes em todas as direções pesquisadas. A Figura 4.12 esclarece o procedimento.

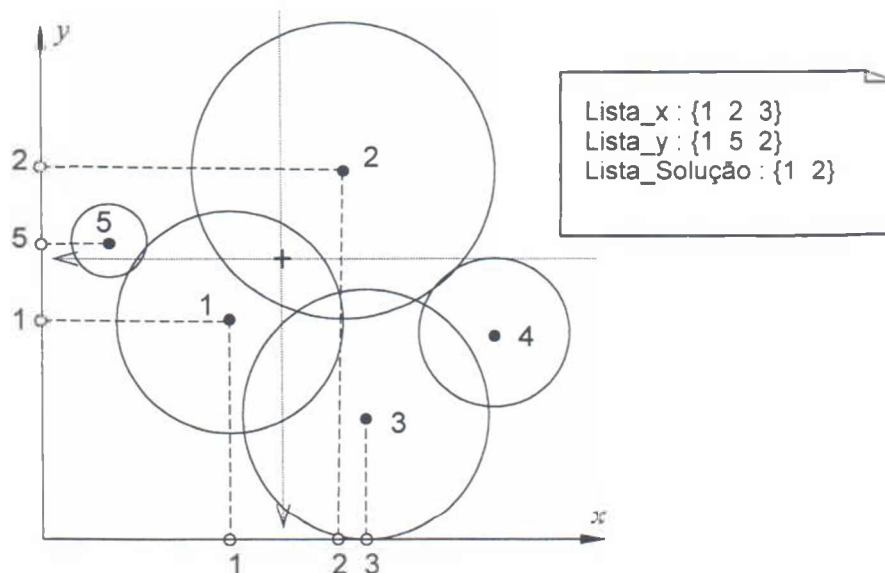


Figura 4.12 – Algoritmo de busca multidirecional para determinação dos suportes que interceptam um ponto.

A implementação computacional das funções de forma do Método *Hp-Clouds* segue os passos já apresentados. A classe **B4Spline** na Figura 3.16 implementa a função de ponderação B-Spline de 4ª ordem (vide Tabela 3.1 e Figura 4.8).

A estratégia de construção da nuvem de pontos é realizada em duas etapas. Inicialmente são definidos os suportes dos pontos nodais. Na classe **MultSupport** a definição pode ser realizada através da imposição direta dos valores, ou pelos algoritmos

de distância (operação `setDmiDistance()`) que fixa o suporte em função de uma constante β que multiplica a distância do pólo ao k-ésimo vizinho mais próximo selecionado; ou ainda, pelo algoritmo de triangulação (operação `setDmiTriangul()`) descrito na Tabela 4.2. A seguir uma consulta a classe `MultiSupport` informa os suportes que um determinado pólo intercepta, utilizando o algoritmo de busca multidirecional ilustrado na Figura 4.12.

Como as funções hierárquicas aqui apresentadas são produzidas a partir das funções de Shepard que são uma particularização do Método de Mínimos Quadrados Móvel para a base unitária, uma classe específica que implementa a formulação baseada em simples somatório, evitando-se o processo de inversão de matrizes, resulta bastante eficiente. Assim, a Figura 4.13 apresenta a classe `MLS_MULT_Shepard` herdeira de `MLS_MULT`.

As funções de forma do método *Hp-Clouds* são produzidas a partir de uma partição da unidade representada por uma classe que segue a interface `MLS_SF` (do Método de Mínimos Quadrados Móvel) com uma base polinomial. Deste modo a classe `HPC_SF` segue a interface `ShapeFunction` diretamente agrupando por composição um objeto que segue a interface `MLS_SF` com outro objeto oriundo da classe `Monomial`.

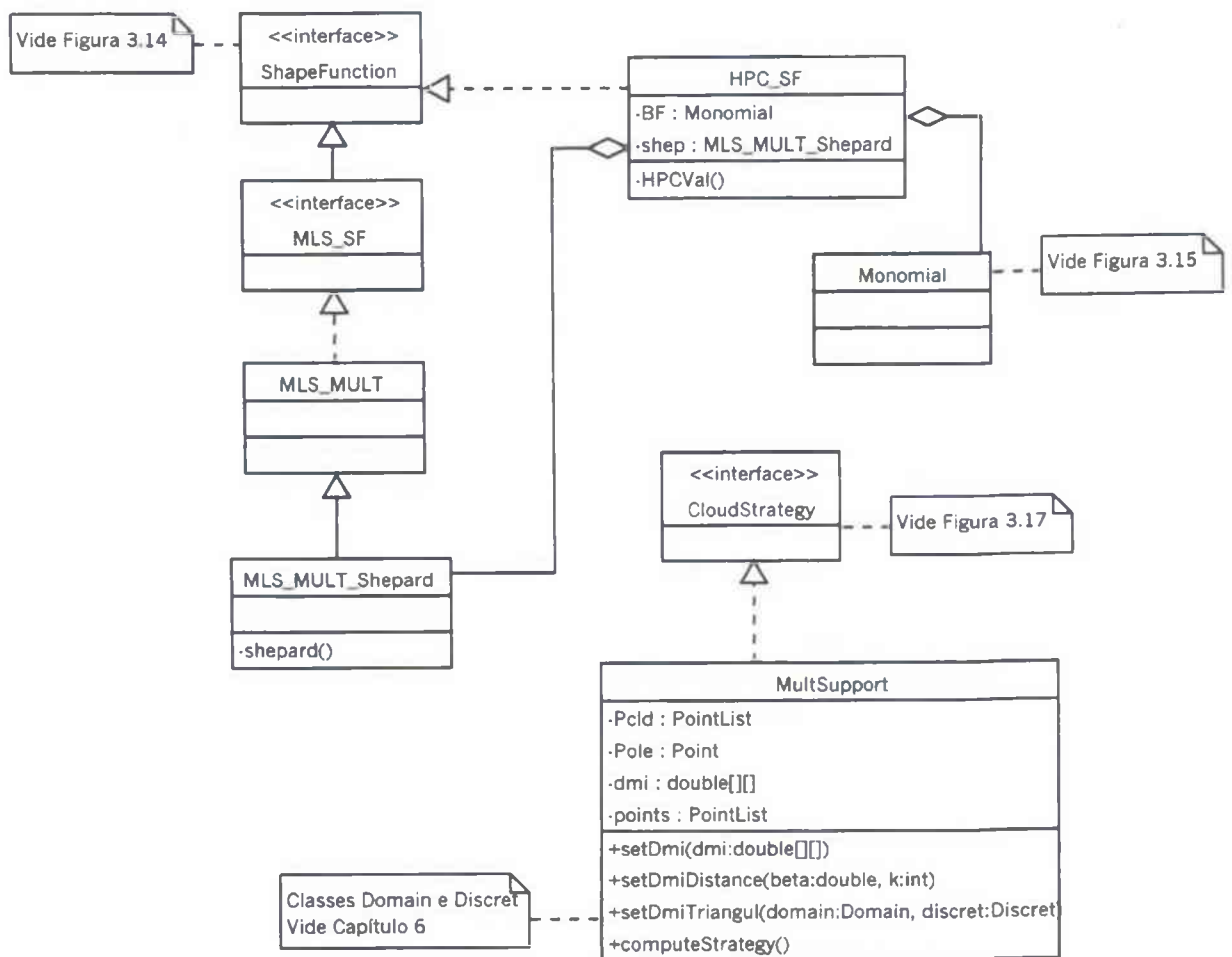


Figura 4.13 – Implementação das classes computacionais para produção das funções de forma do Método *Hp-Clouds*.

Finalizando sobre aspectos computacionais, a implementação do Método *Hp-Clouds* utilizando o Método dos Resíduos Ponderados, com critério de Galerkin, deve considerar que a solução do sistema de equações produz parâmetros ou coeficientes que quando combinados com a base hierárquica resultam nos valores procurados, inclusive nodais. Em síntese a Figura 4.13 resume os diagramas de classe, numa perspectiva de análise, ou seja, sem os detalhes de implementação, envolvidos na produção do Método *Hp-clouds*, ou Método sem Malha Hierárquico.

4.5 - MÉTODO DO PONTO FINITO

O Método do Ponto Finito (*Finite Point Method*) foi descrito em artigos publicados em 1996 por Oñate tendo como colaboradores dentre outros Zienkiewicz e R. L. Taylor. Trata-se de um procedimento de colocação na formulação forte onde as funções de forma são obtidas pelo Método de Mínimos Quadrados Móvel e com as derivadas necessárias à satisfação das equações que regem um problema, obtidas a seguir por diferenciação.

Deve-se destacar um aspecto do método bastante salientado em todos os artigos que é a necessidade de se agregar um termo de estabilização às equações que regem um problema de modo a diminuir a oscilação da solução numérica em torno da solução exata.

As aplicações mais marcantes foram realizadas em problemas de mecânica dos fluidos (Oñate *et al.*, 1996; Oñate, Idelsohn *et al.*, 1996) tendo sido recentemente realizada aplicação em elasticidade (Oñate *et al.*, 2001).

As funções de forma são obtidas como já mencionado utilizando-se o Método de Mínimos Quadrados Móvel, com o ponto móvel coincidindo com um dos pontos nodais chamado de pólo da nuvem (*star node*), como mostra a Figura 4.14. Assim, como visto na Seção 3.2 obtém-se

$$\varphi^T(x_i) = \mathbf{p}^T(x_i) \mathbf{A}^{-1} \mathbf{B} \quad (4.55)$$

A função de ponderação utilizada é a gaussiana truncada apresentada na Seção 3.4. Como estratégia de construção das nuvens de pontos nodais é citado o critério dos quatro quadrantes (Oñate, Idelsohn *et al.*, 1996) apresentado na Seção 4.2, por ocasião da descrição do Método das Diferenças Finitas Generalizadas. A base de funções são monômios, geralmente de ordem um ou dois, como na Seção 3.3.

As equações discretizadas do domínio são obtidas fazendo uso do critério de colocação no Método de Resíduos Ponderados. Assim, tomando $W_i = \bar{W}_i = \bar{\bar{W}}_i = \delta_i$ na equação de resíduos ponderados, onde δ_i é o delta de Dirac, obtém-se o seguinte conjunto de equações

$$\begin{aligned}
[A(\hat{u}_j)]_p - b_j &= 0 & p &= 1, \dots, N_r \\
[\hat{u}_j]_s - \bar{u}_j &= 0 & s &= 1, \dots, N_u \\
[B(\hat{u}_j)]_r - t_j &= 0 & r &= 1, \dots, N_t
\end{aligned}
\tag{4.56}$$

onde N_u e N_t são o número de pontos nodais localizados nos contornos Γ_u e Γ_t , respectivamente e N_r é o número de pontos restante no domínio Ω não pertencentes a nenhum contorno, Γ_u e Γ_t . Sendo, ainda, a aproximação da solução utilizada acima dada por

$$\hat{u}(x) = \varphi^T \mathbf{u}^h \tag{4.57}$$

onde u^h é o valor da solução nos pontos nodais.

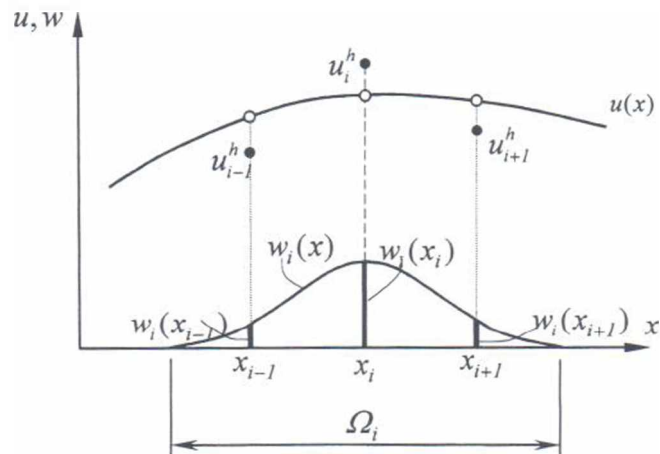


Figura 4.14 – Método de Mínimos Quadrados Móvel com ponto de avaliação igual a um dos pontos nodais da nuvem.

Substituindo (4.57) nas equações (4.56), determinando as derivadas contidas nos operadores $A(\hat{u}_j)$ e $B(\hat{u}_j)$ como na Seção 3.2 chega-se ao sistema de equações algébricas na forma

$$K \mathbf{u}^h = \mathbf{f} \tag{4.58}$$

com $K_{ij} = [A(\phi_i)]_j + [B(\phi_i)]_j$, sendo que K não é usualmente simétrica. Na expressão f é um vetor contendo as contribuições dos termos b e t das equações que regem o problema e dos valores prescritos \bar{u} .

Como, salientado na Seção 3.2, as funções de forma obtidas pelo Método de Mínimos Quadrados Móvel não atendem à condição delta de Kronecker, ou seja, não alcançam o valor unitário sobre o pólo da nuvem com valor nulo nos demais pontos nodais. Isto se deve ao caráter de aproximação e não de interpolação dos procedimentos de mínimos quadrados.

A forma (4.58) guarda grande semelhança com a formulação do Método dos Elementos Finitos onde aparecem como incógnitas apenas a variável u , como mostrado na Seção 2.5. De modo a manter esta formulação no Método do Ponto Finito, Oñate e colaboradores utilizam o artifício de no contorno Γ_u realizarem uma aproximação com base unitária, vale dizer $p^T = \{1\}$, e portanto com apenas um ponto nodal na nuvem. Neste caso é imediato que

$$\phi_i(x_i) = p^T (P^T W P)^{-1} (P^T W) = 1 \quad (4.59)$$

e então

$$a_2 = \bar{u}_i = u_p(x_j) \Big|_{\Gamma_u} \quad (4.60)$$

conduzindo a um sistema como apresentado na Seção 2.5

$$\begin{aligned} \{a_1\} &= [K_{11}]^{-1} \{f_1 - K_{12} a_2\} \\ \{q\} &= \{f_2\} - \{K_{22} a_2\} - \{K_{21} a_1\} \end{aligned} \quad (4.61)$$

De forma similar a condição de contorno Γ_t pode ser aproximada com um ponto nodal como

$$B(u(x_i)) = t(x_i) \quad (4.62)$$

A formulação conduz a dois inconvenientes. Primeiro o operador do domínio $A(\hat{u}_j)$ não é aplicado no contorno, como apresentado em (4.56), violando a condição de que o contorno pertence ao domínio. E, em segundo lugar a aproximação pobre dos pontos de contorno, na busca de uma formulação semelhante à do Método dos Elementos Finitos, pode gerar uma solução não tão precisa nas proximidades do contorno.

Como forma de reduzir os problemas de instabilidade numérica e mau condicionamento do sistema de equações algébricas, devido principalmente à incapacidade dos métodos de colocação em satisfazer precisamente as equações que regem um problema em todo o domínio da nuvem de pontos, utilizando apenas o pólo da nuvem como representativo, Oñate e colaboradores introduziram uma técnica de estabilização que agrega às equações que regem o problema novos termos derivados, geralmente, de uma expansão em série de Taylor ou a sentimento. Estas técnicas de estabilização são de senso comum na solução numérica de equações não auto-adjuntas de modo a se evitar oscilações da solução (Zienkiewicz e Taylor, 1989).

Para o Método do Ponto Finito, Oñate e colaboradores, desenvolveram uma técnica baseada na imposição de leis de equilíbrio da mecânica sobre um domínio de dimensão finita. As variáveis de campo incógnitas são aproximadas dentro do domínio finito utilizando-se expansão em série de Taylor e tomando-se termos de ordem mais elevada do que é usual na formulação infinitesimal.

Para o caso da equação de fluxo em uma dimensão a Figura 4.15 apresenta um segmento típico AB de comprimento h onde o equilíbrio de fluxos devem estar satisfeitos.

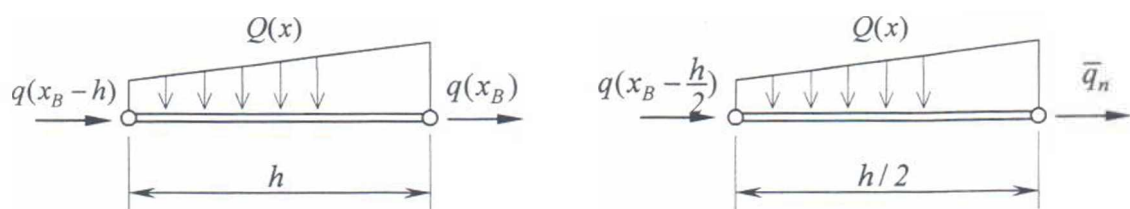


Figura 4.15 – Domínio finito em equilíbrio de fluxos (a), e equilíbrio na vizinhança de um contorno Neumann (b).

Expandindo o fluxo em série de Taylor de terceira ordem, tem-se

$$q(x_B - h) = q(x_B) - h \left. \frac{dq}{dx} \right|_B + \frac{h^2}{2} \left. \frac{d^2q}{dx^2} \right|_B + O(h^3) \quad (4.63)$$

Escrevendo o equilíbrio para o caso estacionário

$$\sum_{i=1}^n q_i = q(x_B) - q(x_B - h) - \frac{1}{2} [Q(x_B) + Q(x_B - h)] h = 0 \quad (4.64)$$

onde foi assumido que a fonte Q tem uma distribuição linear sobre o segmento AB , e expandindo também em série de Taylor de segunda ordem vem

$$Q(x_B - h) = Q(x_B) - h \left. \frac{dQ}{dx} \right|_B + O(h^2) \quad (4.65)$$

Substituindo (4.61) e (4.63) em (4.62) e adotando $x_B = x$ como posição arbitrária obtém-se

$$-\frac{dq}{dx} + Q + \frac{h}{2} \frac{d}{dx} \left[-\frac{dq}{dx} + Q \right] = 0 \quad (4.66)$$

A taxa de fluxo se relaciona com a variável incógnita u através da Lei de Fourier, ou seja, $q = -kdu/dx$, onde k é a constante de condutibilidade térmica. Substituindo em (4.64) chega-se finalmente

$$r - \frac{h}{2} \frac{dr}{dx} = 0 \quad (4.67)$$

com

$$r = \frac{d}{dx} \left(k \frac{du}{dx} \right) + Q \quad (4.68)$$

Fazendo agora a equação de equilíbrio para um domínio finito de comprimento $h/2$ próximo a um contorno Neumann cujo valor de fluxo prescrito vale \bar{q}_n tem-se

$$\bar{q}_n - q(x_B - h/2) - \frac{h}{2} Q = 0 \quad (4.69)$$

Expandindo em série de Taylor e substituindo vem

$$\bar{q}_n - q - \frac{h}{2} \left[\frac{dq}{dx} + Q \right] = 0 \quad (4.70)$$

Aplicando a lei de Fourier chega-se

$$k \frac{du}{dx} + \bar{q}_n - \frac{h}{2} r = 0 \quad (4.71)$$

onde r é dado por (4.66).

As equações que regem o fluxo unidimensional são modificadas como os termos de estabilização na forma

$$\begin{aligned} A - \frac{1}{2} h \frac{dA}{dx} &= 0 && \text{em } \Omega \\ u - \bar{u} &= 0 && \text{em } \Gamma_u \\ B - \frac{1}{2} h A &= 0 && \text{em } \Gamma_t \end{aligned} \quad (4.72)$$

com

$$\begin{aligned} A &= \frac{d}{dx} \left(k \frac{du}{dx} \right) + Q \\ B &= k \frac{du}{dx} + \bar{q}_n \end{aligned} \quad (4.73)$$

Como pode-se verificar o termo de estabilização na primeira equação de (4.72) necessita de derivadas terceiras impostas a aproximação u . Na maioria dos casos se utiliza uma aproximação no máximo de ordem quadrática, fazendo com que o termo de estabilização no domínio seja nulo. Nas palavras de Oñate *et al.* (2001): “Isto embasa o fato de que os termos de estabilização chave são aqueles advindos das condições de contorno Neumann.”

A seguir apresenta-se a discretização do problema de Poisson com o termo de estabilização, seguindo dedução acima feita para uma dimensão. Explicitando-se as equações (4.72) e (4.73) chega-se

$$\begin{aligned} \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + Q - \frac{1}{2} h_x \left[\frac{\partial^3 u}{\partial x^3} + \frac{\partial^3 u}{\partial x \partial y^2} + \frac{\partial Q}{\partial x} \right] - \frac{1}{2} h_y \left[\frac{\partial^3 u}{\partial y^3} + \frac{\partial^3 u}{\partial y \partial x^2} + \frac{\partial Q}{\partial y} \right] &= 0 \quad \text{em } \Omega \\ \frac{\partial u}{\partial n} + \bar{q} - \frac{1}{2} h_x n_x \left[\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + Q \right] - \frac{1}{2} h_y n_y \left[\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + Q \right] &= 0 \quad \text{em } \Gamma_t \\ u - \bar{u} &= 0 \quad \text{em } \Gamma_u \end{aligned} \quad (4.74)$$

Desprezando-se, como sugere Oñate, os termos de terceira ordem na equação do domínio e procedendo a ortogonalização de resíduos com função de ponderação delta de Dirac, padrão em colocação, resulta

$$\begin{aligned} \left[\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) \right]_{\Omega} - \frac{\partial u}{\partial n} \Big|_{\Gamma} \left[-\frac{1}{2} (h_x n_x + h_y n_y) \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) \right]_{\Gamma} - q_u \Big|_{\Gamma} &= \\ = -\bar{q} \Big|_{\Gamma} - \left(Q - \frac{1}{2} h_x \frac{\partial Q}{\partial x} - \frac{1}{2} h_y \frac{\partial Q}{\partial y} \right) \Big|_{\Omega} + \frac{1}{2} (h_x n_x + h_y n_y) Q \Big|_{\Gamma} &= \end{aligned} \quad (4.75)$$

$$u - \bar{u} = 0$$

Oñate não inclui o termo $\partial u / \partial n$ em Γ já que não utiliza a formulação fraca para produzir as equações discretas. Aplicando as funções de forma e agrupando na forma padrão, apresentada do Método dos Resíduos Ponderados, chega-se

$$\begin{aligned} \mathbf{K}_i &= \phi_{,xx}^i + \phi_{,yy}^i \Big|_{\Omega} - \frac{1}{2} (h_x n_x + h_y n_y) (\phi_{,xx}^i - \phi_{,yy}^i) \Big|_{\Gamma} \\ \mathbf{f}_i &= -\bar{q} \Big|_{\Gamma} - \left(Q - \frac{1}{2} h_x \frac{\partial Q}{\partial x} - \frac{1}{2} h_y \frac{\partial Q}{\partial y} \right) \Big|_{\Omega} + \frac{1}{2} (h_x n_x + h_y n_y) Q \Big|_{\Gamma} \\ \mathbf{H} = \mathbf{G} &= [\mathbf{0} \mid \mathbf{1}]^T \end{aligned} \quad (4.76)$$

Num formato matricial e considerando-se que ocorre um desacoplamento devido à consideração peculiar do contorno com nuvem de pontos composta de ponto nodal único resulta

$$\begin{bmatrix} \mathbf{K}_{11} & \mathbf{K}_{12} \\ \mathbf{K}_{21} & \mathbf{K}_{22} \end{bmatrix} \begin{Bmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \end{Bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{1} \end{bmatrix} \{ \mathbf{q}_u \} = \begin{Bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \end{Bmatrix} \quad (4.77)$$

onde, \mathbf{a}_1 incógnitas nos pontos do domínio (sem contorno) e no contorno Γ_t ; e $\mathbf{a}_2 = \bar{u}$ são os valores prescritos no contorno Γ_u . Como resultado vem

$$\begin{aligned} [\mathbf{K}_{11}]\{\mathbf{a}_1\} &= \{\mathbf{f}_1\} - [\mathbf{K}_{12}]\{\bar{\mathbf{u}}\} \\ \{\mathbf{q}_u\} &= \{\mathbf{f}_2\} - [\mathbf{K}_{21}]\{\mathbf{a}_1\} - [\mathbf{K}_{22}]\{\bar{\mathbf{u}}\} \end{aligned} \quad (4.78)$$

As expressões são similares às do Método dos Elementos Finitos, como Oñate almejou.

As classes modeladas no Capítulo 3 permitem a implementação do Método do Ponto Finito como desenhado na Seção 4.7.

4.6 - EXEMPLO DE APLICAÇÃO

O exemplo proposto a seguir, regido pela equação de Poisson, ilustra a aplicação dos Métodos sem Malha. Não é uma análise numérica exaustiva, mesmo porque alguns dos artigos citados realizam essa tarefa. Não se pretende também estabelecer comparações de performance entre os métodos, mesmo porque, a escolha de um método, para a solução de um problema específico, é ditada muitas vezes por razões de conveniência computacional ou até mesmo por tradição.

Sem dúvida o sistema computacional proposto permite à exaustão a realização de ambos os tipos de análises citadas acima. Neste momento serão realçadas as diferenças na configuração dos métodos, como apresentadas neste capítulo. Os resultados foram obtidos a partir da modelagem computacional que se completará nos Capítulos 5 e 6.

Seja o problema de valor de contorno regido pela equação de Poisson com fonte nula, conhecida como equação de Laplace, e correspondentes condições de contorno

$$\begin{aligned} \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} &= 0 & \text{para } \begin{cases} 0 < x < 5 \\ 0 < y < 10 \end{cases} \\ u(x, 0) &= 0 & 0 < x < 5 \\ u(y, 0) &= 0 & 0 < y < 10 \\ u(x, 10) &= 100 \sin\left(\frac{\pi}{10} x\right) & 0 < x < 5 \\ \frac{\partial u(5, y)}{\partial x} &= 0 & 0 < y < 10 \end{aligned} \quad (4.79)$$

A solução analítica é dada por

$$u(x, y) = 100 \frac{\sinh(\pi / 10 y) \sin(\pi / 10 x)}{\sinh(\pi)} \quad (4.80)$$

A Figura 4.16 ilustra o domínio com a representação das condições de contorno bem como gráfico com as curvas de nível geradas pela solução analítica correspondente à equação (4.80).

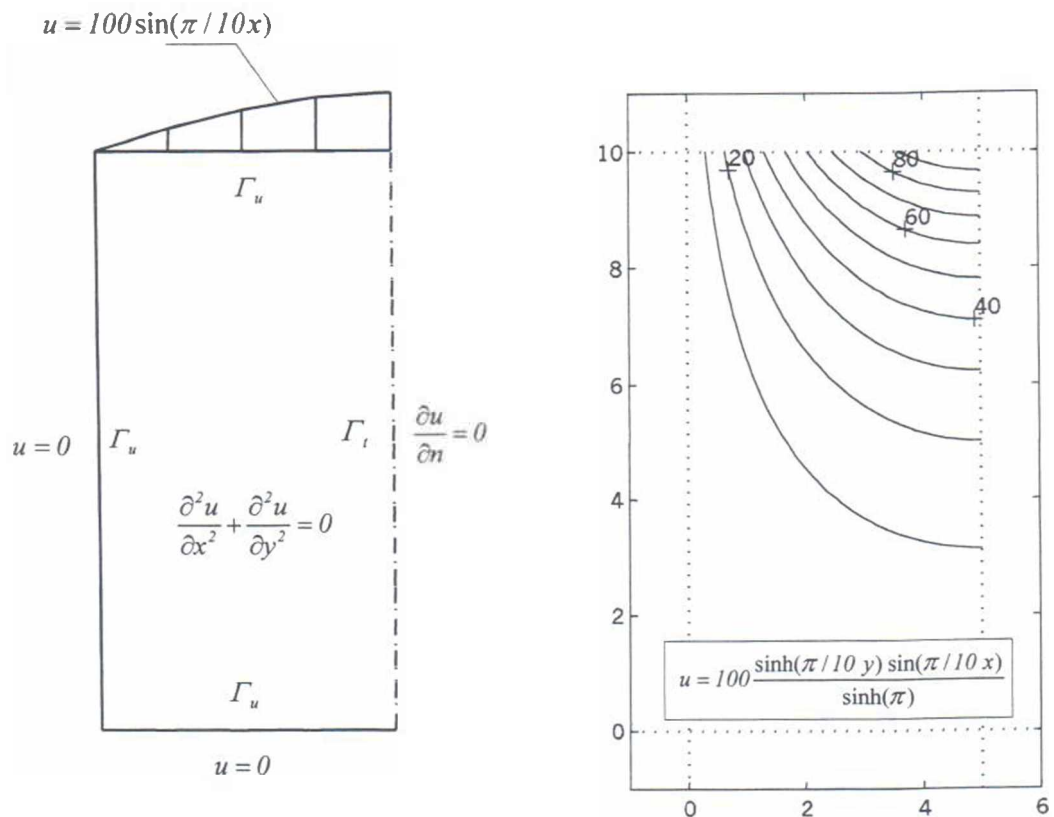


Figura 4.16 – Domínio regido pela equação de Laplace e solução analítica.

Na solução pelo Método das Diferenças Finitas Generalizadas deve-se estabelecer uma discretização como mostra a Figura 4.17 e um critério para a construção da aproximação local. No caso foi estabelecida uma distância fixa dm_i para suporte de todos os pontos nodais. A figura também apresenta a solução discreta, desenhada em conjunto com a solução analítica, revelando uma boa qualidade.

Cabe observar que foram utilizados pontos gêmeos nos cantos do domínio, como indicam as normais no contorno da Figura 4.17. Uma consideração de ordem prática importante se refere à colocação sobre um dos pontos gêmeos. Nesse caso o outro ponto gêmeo não deverá estar presente na nuvem de pontos. Caso contrário se produzirá singularidade na matriz global do sistema. Quando a colocação se dá nos demais pontos a existência dos pontos gêmeos se revela importante na precisão do resultado junto aos cantos (Pulino, 1989).

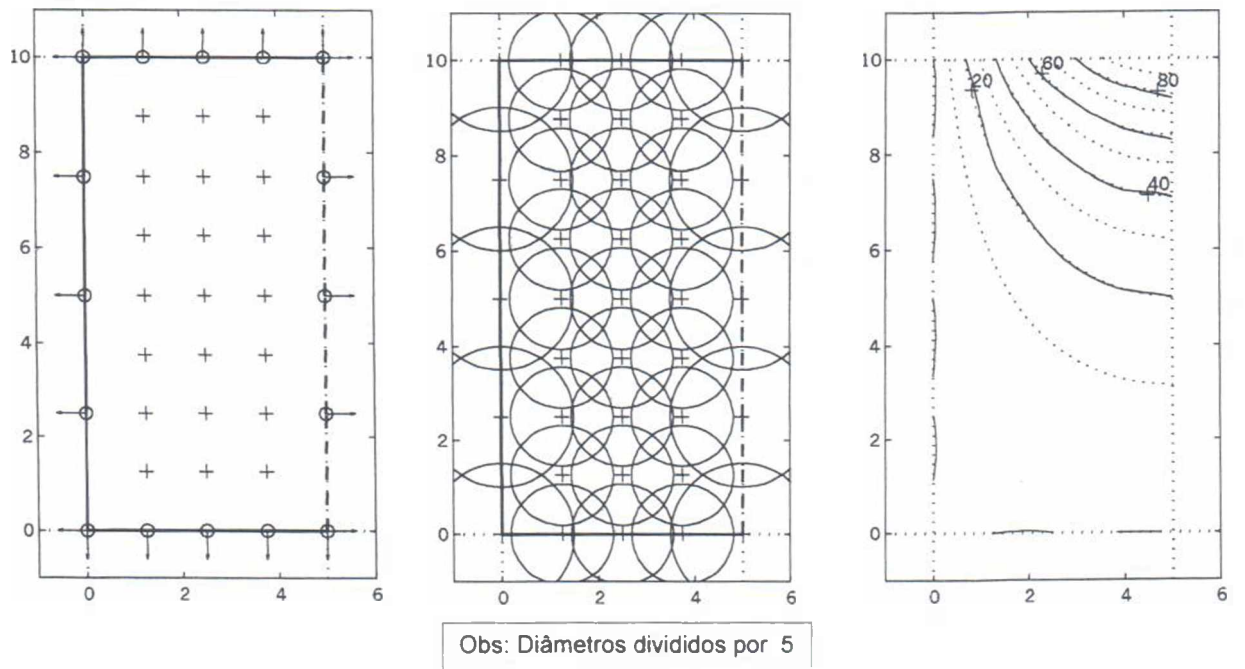


Figura 4.17 – Solução da equação de Laplace com o Método das Diferenças Finitas Generalizadas.

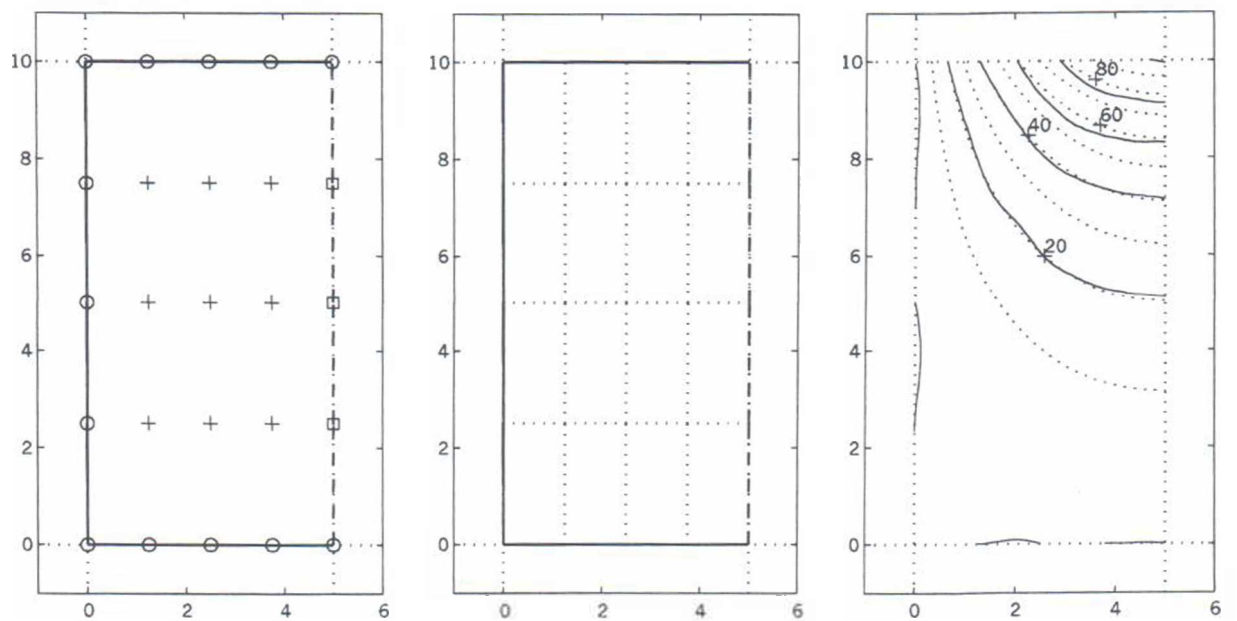


Figura 4.18 – Solução da equação de Laplace com o Método de Galerkin sem Malha.

A Figura 4.18 apresenta discretização para solução pelo Método de Galerkin sem Malha. A função de ponderação utilizada foi a gaussiana truncada, com critério da

distância para estabelecer a nuvem de pontos com parâmetro $\alpha = 2$. Utilizou-se a base linear na determinação das funções de forma. A figura mostra além da discretização dos pontos nodais a discretização de células de integração, realizada com 3×3 pontos gaussianos.

Como relatado, a aplicação do Método *Hp-Clouds* requer que de forma preliminar seja imposta a dimensão do suporte para cada ponto nodal. Na Figura 4.19 os suportes foram definidos através do algoritmo de triangularização exposto na Seção 4.4. A solução foi obtida com os mesmos parâmetros de integração utilizados acima. As funções de forma foram geradas com polinômios de primeira ordem.

A utilização da característica hierárquica do Método *Hp-Clouds* depende da avaliação de indicadores de erro para cada ponto nodal. Duarte (1996), apresenta um tratamento hierárquico que pode ser aplicado em outros Métodos sem Malha. Pelo momento se relaciona esta investigação para trabalhos futuros.

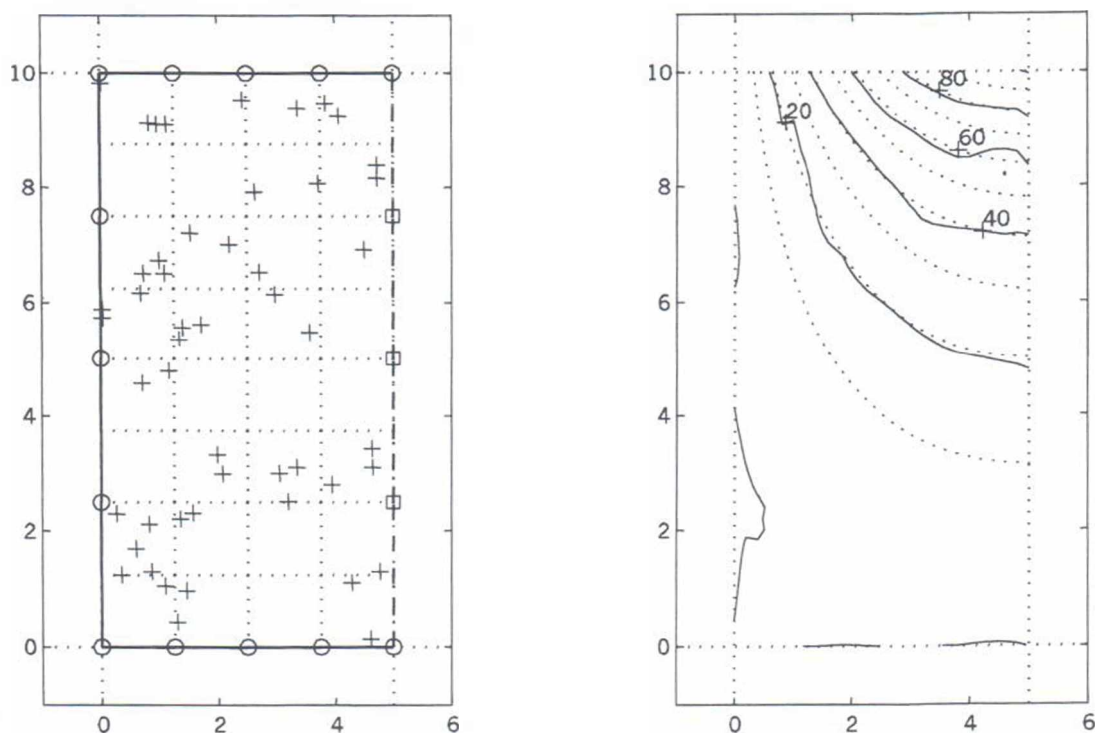


Figura 4.19 – Solução da equação de Laplace pelo Método *Hp-Clouds*.

Por fim apresenta-se a solução pelo Método dos Elementos Finitos. Neste caso as células são subdomínio de integração e seus pontos nodais geram as funções de forma para a aproximação local. A Figura 4.20 apresenta a solução obtida com integração 2×2 .

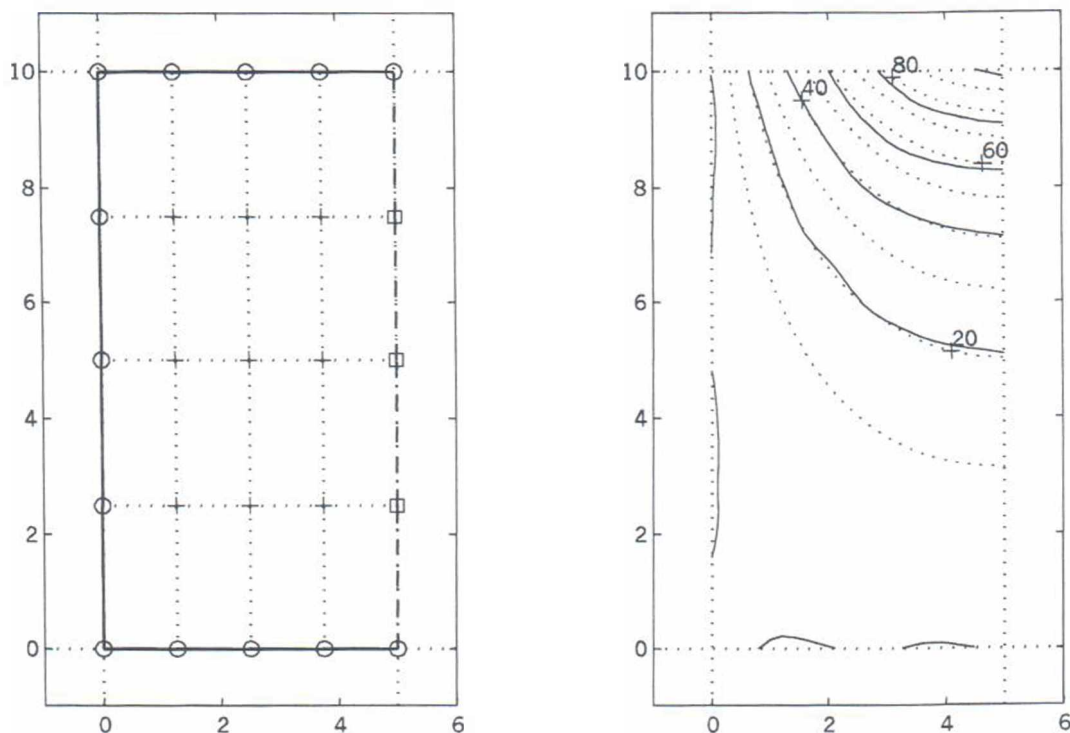


Figura 4.20 – Solução da equação de Laplace com o Método dos Elementos Finitos.

De modo geral uma observação qualitativa permite concluir que todos os métodos produziram boa solução com a discretização bastante modesta proposta. Um adensamento das discretizações produziria soluções de qualidade superior.

4.7 - MODELAGEM COMPUTACIONAL DOS MÉTODOS SEM MALHA

Uma vez realizada a modelagem das funções de forma, núcleo central dos procedimentos de discretização sem malha relatados, pode-se passar à definição computacional de cada método em si, e como cada método monta as equações algébricas que permitirão a solução discreta do problema. Essas duas abstrações serão tratadas por grupos de classes distintas.

A Figura 4.21 apresenta a interface **Method**, bem como classes para cada um dos métodos detalhados neste capítulo, além do Método dos Elementos Finitos. A interface estabelece e induz um protocolo comum de comunicação com as classes dos métodos concretos implementados. Basicamente são parâmetros para a definição das funções de forma. Alguns atributos são invariáveis e caracterizam um método, não podendo ser alterados, sendo declarados no interior da classe como estático. Outros podem ser modificados até mesmo em tempo de execução.

Como exemplo de flexibilidade proporcionada pela programação através de interfaces, a obtenção das primeiras derivadas das funções de forma de um método qualquer segue a seguinte expressão

```
currentMethod.SF().dphi()
```

onde, **currentMethod** é uma referência à instância (objeto) de uma das classes concretas que definem os métodos.

Além dos parâmetros que a interface **Method** define outros atributos específicos podem constituir o corpo de um método. Nem todos os parâmetros devem ser implementados. Por exemplo, os métodos de colocação não devem retornar valor para os acessos ao número de pontos gaussianos para integração do domínio e do contorno.

A função **getEquation()** chama a atenção. As equações da discretização devem ser enumeradas de modo a permitir a montagem das matrizes locais no sistema global de equações lineares. Os métodos de discretização implementados seguem um mesmo padrão de numeração, onde todos os pontos nodais do domínio e do contorno são pontos para discretização. De forma diferente o Método dos Elementos Finitos numera os pontos nodais das células. De qualquer modo optou-se por encapsular o conceito numa classe

denominada de **Equation**, vide Figura 4.21, deixando para cada método o acesso aos dados das equações. Desse modo a classe pode evoluir permitindo outros padrões de numeração de equações não implementados aqui.

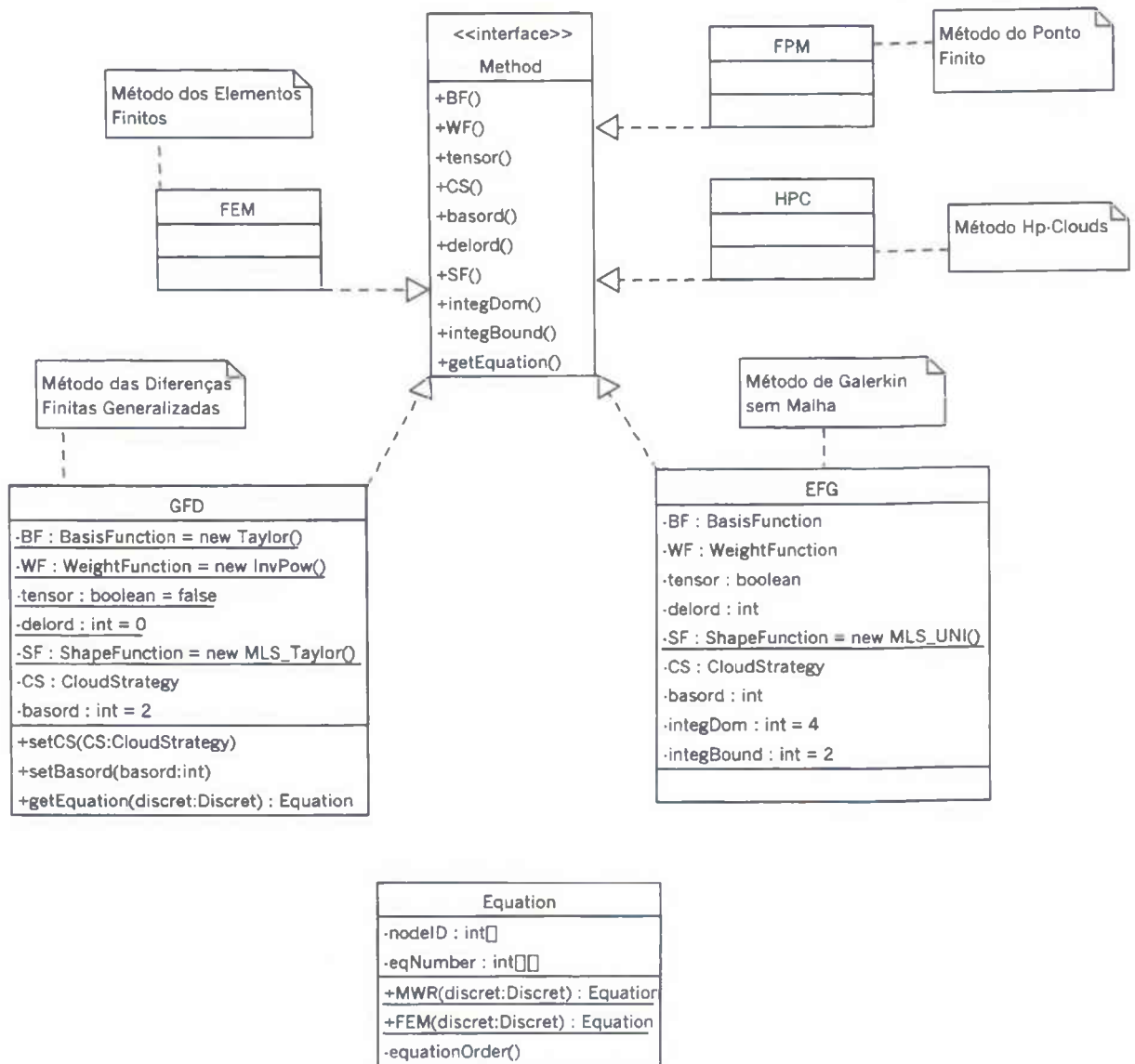


Figura 4.21 – Interface para definição de métodos de discretização.

As classes que implementam a interface **Method** conformam os métodos de discretização implementados. A classe abstrata **MWR** na Figura 4.22 é o engenho. Nesta classe é executado o procedimento de montagem das matrizes globais. Uma classe abstrata não pode criar objetos, apenas classes que a estendem por herança criam objetos. Assim, caso um método siga o critério de Galerkin basta instanciar um objeto da classe **Galerkin**

informando o tipo de problema (vide o Capítulo 5), uma discretização (vide o Capítulo 6) e um dos métodos como os definidos na Figura 4.21. Claro está que a inconsistência entre critério e método acarretará um procedimento de exceção.

Toda a funcionalidade de execução da montagem do sistema global reside na classe abstrata **MWR**. As classes derivadas devem se incumbir de implementar dois métodos denominados de **domain()** e **boundary()**, que são operações abstratas (sem código) em **MWR**. Essa construção é denominada de padrão de projeto *Template Method* (Gamma *et al.*, 2000). Novos motores podem ser criados implementando o código das funções abstratas. Por esse padrão o esqueleto de um algoritmo é definido num método da classe abstrata, no caso na operação **XEQ()**, sendo postergados alguns de seus passos para subclasses, no caso **domain()** e **boundary()**.

Por exemplo o algoritmo **domain()** em **Galerkin** realiza a integração gaussiana no domínio construindo as matrizes **K** e **F**. Já **boundary()** integra o contorno resultando as matrizes **F_t**, **U**, **G**, e **H**. No caso **F_t** agrupa os valores prescritos em Γ_t . Esta construção permite grande flexibilidade permitindo a construção de engenhos específicos para métodos particulares.

Uma questão relevante na construção das matrizes globais é o procedimento de montagem, ou de *assembler*. Dependendo do critério, Galerkin ou colocação, as matrizes serão montadas com o acréscimo de outras matrizes, de vetor coluna, ou de um vetor linha, e podendo ainda ser um escalar, como mencionado na Seção 2.4. Na verdade trata-se de incorporar um procedimento particular de adição a um objeto do tipo matriz (vide **MatrixType** no Capítulo 6). Essa incorporação se dará ao tempo de execução já que nem todas as matrizes utilizadas necessitam de *assembler*. Incorporar em tempo de execução um comportamento é tratado pelo padrão de projetos denominado de *Decorator* (Jandl, 2003). Normalmente esse padrão é usado para dar e tirar em tempo de execução alguma decoração a um objeto gráfico.

A Figura 4.23 apresenta a classe **Assembler** que implementa a interface **MatrixType** (vide Capítulo 6). Ou seja, um objeto do tipo **Assembler** irá emular o comportamento de uma matriz. Foram criadas duas subclasses, uma para realizar o *assembler* (adição) em matrizes retangulares denominada de **MatrixAssembler**, e outra para realizar a operação em vetores colunas, denominada de **ColVectorAssembler**.

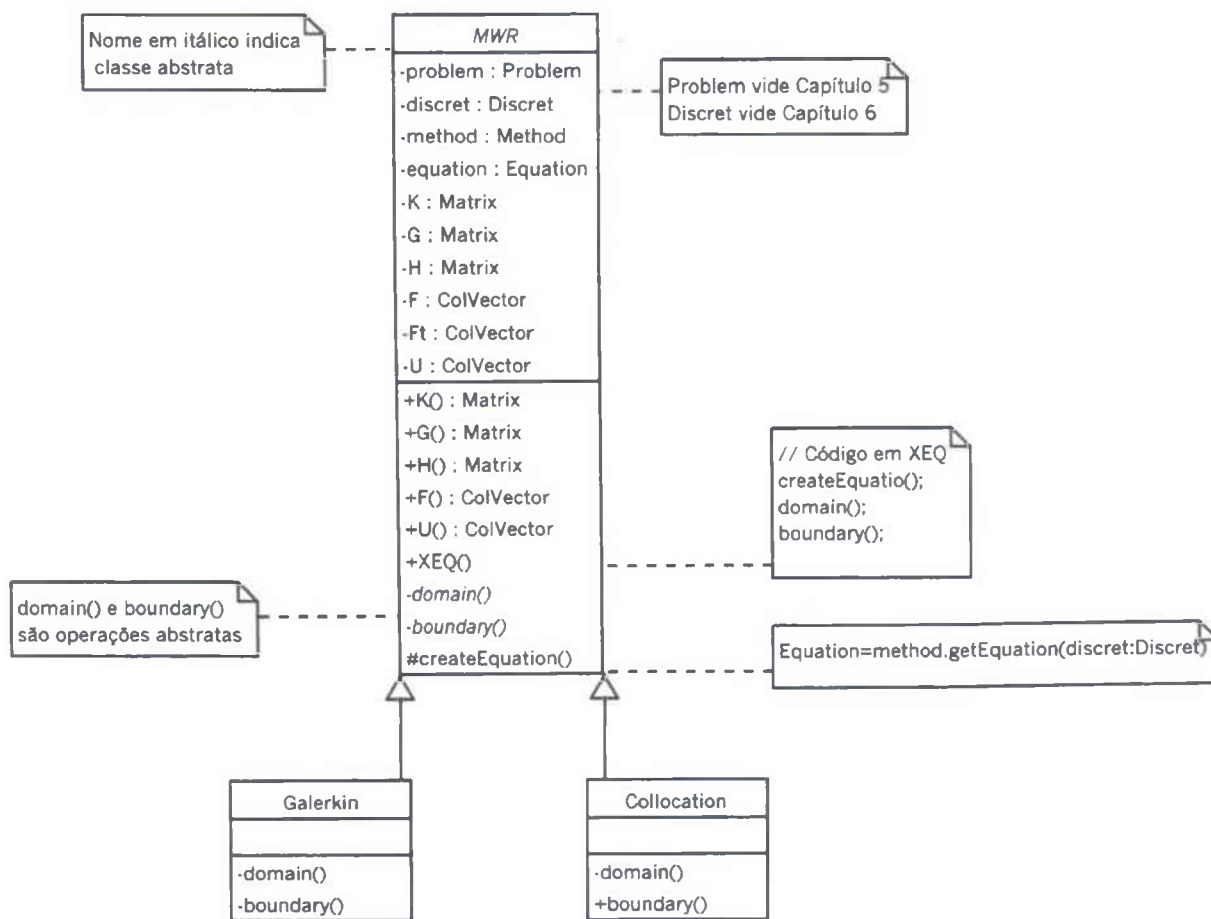


Figura 4.22 – Classes para construção das matrizes globais.

Uma classe **Matrix** que assume o comportamento de **MatrixAssembler** terá uma operação denominada **add()** que realiza a montagem em função de uma lista de pontos nodais. Por polimorfismo pode-se adicionar uma matriz, como em Galerkin, ou uma linha (**RowVector**), como em colocação. Do mesmo modo em um vetor coluna (**ColVector**), pode-se adicionar por *assembler* outro vetor coluna ou um escalar (**Scalar**).

Embora a classe **Assembler** se relacione de forma marcante com o pacote **JMatrix**, que será apresentado no Capítulo 6, optou-se pela exposição mais oportuna junto à classe **MWR**. Essa construção é bastante flexível, e ao que parece não reportada na literatura, já que incorpora o comportamento de *assembler* a uma matriz somente em tempo de execução e àquelas matrizes que o necessitam.

As classes propostas cobrem os casos previstos para os critérios de colocação e Galerkin.

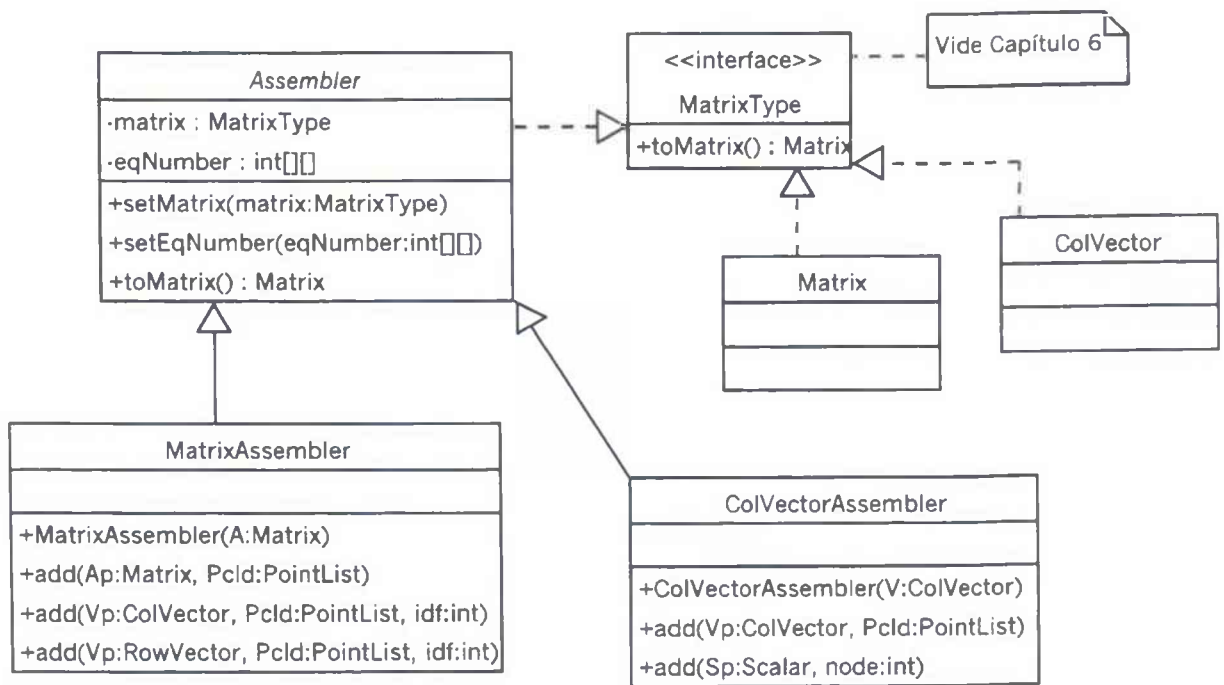


Figura 4.23 – Classes para incorporação de comportamento de *assembler* a matrizes e vetores.

4.8 - CONSIDERAÇÕES FINAIS

A exposição do Método das Diferenças Finitas Generalizadas buscou reunir as suas principais características de forma a apresentar um quadro completo do recolhido em diversos artigos. Além disso a apresentação objetivou justificar ou demonstrar teoricamente alguns aspectos relevantes. No âmbito do Método dos Resíduos Ponderados os pontos gêmeos são naturalmente justificados como limites dos intervalos de integração no contorno. Demonstrou-se, através da minimização da soma do erro no domínio e no contorno, como são geradas as funções de forma considerando as derivadas normais no contorno. Na literatura a apresentação é mais axiomática. Cabe destacar que a apresentação matricial da construção do sistema global de equações, resultante da abordagem do Método dos Resíduos Ponderados com formulação fraca, não se encontra na literatura, onde se emprega uma abordagem com viés algorítmico. O estudo da função de ponderação e da geração da nuvem de pontos deve ser relacionado como objeto de trabalhos futuros.

Visto como uma generalização do Método dos Elementos Finitos, o Método de Galerkin sem Malha possui grande possibilidade de evolução e aplicação em diversos problemas da mecânica computacional. A exposição iniciada aqui é detalhada no próximo capítulo, onde os aspectos práticos da construção do método são apresentados.

A apresentação do Método *Hp-Clouds* seguiu um caminho mais construtivo ao invés da apresentação axiomática presente na literatura. Devido à necessidade de concisão, não foram incluídos alguns exemplos que dariam maior didática aos conceitos. A função de ponderação construída a partir de uma base de funções *splines*, apenas citada na literatura, teve sua formulação apresentada. O algoritmo de recobrimento do domínio baseado em triangularização restringida apresentado garante uma construção segura da nuvem de pontos embora o algoritmo *greedy* utilizado não seja muito eficiente. A implementação desse método requer uma forte aproximação conceitual com a Geometria Computacional, como fica evidente nos algoritmos descritos, sendo este um caminho natural de novas investigações.

O Método do Ponto Finito possui forte ligação com um problema específico devido aos termos de estabilização numérica empregados. Estes termos conduzem a alguma dificuldade na implementação computacional. O procedimento de colocação na formulação fraca, mencionado no Capítulo 2, pode conduzir a uma implementação mais

simples de um método de colocação que se utiliza diretamente das funções de forma geradas pelo Método de Mínimos Quadrados Móvel. Esta deve ser como já lembrado uma direção de trabalhos futuros.

A modelagem dos objetos computacionais proposta permite grande flexibilidade na escolha do método de resolução de um problema, passível de realizar-se até mesmo em tempo de execução. Como a literatura consultada sobre orientação a objetos aplicada a métodos de discretização se restringe ao Método dos Elementos Finitos, a possibilidade de alternância de método não possui referência para comparação. Cabe destacar que a modelagem da abstração, método de discretização, em uma classe de conformação do método e outra classe com os engenhos de resolução, levou a uma arquitetura que beneficia a clareza e a expansibilidade.

O procedimento de se incorporar uma operação especial de adição indexada a uma matriz para a realização da montagem (assembler) das matrizes globais em tempo de execução demonstra uma das características fundamentais da análise orientada a objetos, qual seja, a viabilidade de se aproveitar código existente em grande escala. Não foi necessário o acréscimo de nenhuma linha de código ao pacote JMatrix para que seus objetos adquirissem a operação de assembler.

5 - MÉTODO DE GALERKIN SEM MALHA EM ELASTICIDADE

5.1 - INTRODUÇÃO

No Capítulo 4 foram apresentados os Métodos sem Malha objeto de análise neste trabalho. Dos métodos apresentados o Método de Galerkin sem Malha é atualmente o mais pesquisado. Este método propicia a abordagem de uma grande gama de aplicações em diferentes campos da física e da engenharia. Dessas aplicações o problema mais fundamental no âmbito da mecânica é o da elasticidade bidimensional.

O objetivo aqui é o estudo da formulação de um novo problema, elasticidade no caso, dentro de uma estrutura computacional que se pretende expansível, inclusive quanto à modelação de novas aplicações.

Via de regra o caminho adotado para a apresentação deste problema na literatura é o da formulação variacional. Neste capítulo a apresentação da formulação discreta será realizada como no Capítulo 2 com a utilização do Método dos Resíduos Ponderados.

Além do problema de elasticidade, exposto a seguir, o Método de Galerkin sem Malha vem sendo aplicado em diferentes problemas de mecânica computacional. Cabe destacar os trabalhos de Belytschko e colaboradores em dinâmica da fratura (Belytschko e Tabbara, 1996). Muitos outros pesquisadores vem abordando o método ocorrendo na literatura vasta gama de aplicações, dentre elas pode-se citar: análise modal de estruturas (Ouatouati e Johnson, 1999), análise elastoplástica (Barry e Saigal, 1999), acústica (Suleau *et al.*, 2000), fluxo de águas rasas em rios (Du, 2000), e análise de cascas e estruturas espaciais (Nogushi, 2000).

5.2 - FORMULAÇÃO DISCRETA EM ELASTICIDADE

Seja um problema de elasticidade bidimensional, com pequenos deslocamentos e material isotrópico elástico linear, num domínio Ω delimitado por um contorno Γ , como o representado na Figura 5.1. Um elemento infinitesimal desse domínio esta submetido a forças que produzem o diagrama de corpo livre, também representado na Figura 5.1, cujo equilíbrio é dado por

$$\begin{aligned}\sum F_x &= (\sigma_x + \frac{\partial \sigma_x}{\partial x} dx)dy - \sigma_x dy + (\tau_{xy} + \frac{\partial \tau_{xy}}{\partial y} dy)dx - \tau_{xy} dx + b_x dx dy = 0 \\ \sum F_y &= (\tau_{xy} + \frac{\partial \tau_{xy}}{\partial x} dx)dy - \tau_{xy} dy + (\sigma_y + \frac{\partial \sigma_y}{\partial y} dy)dx - \sigma_y dx + b_y dx dy = 0\end{aligned}\quad (5.1)$$

onde b_x e b_y são as forças atuantes no corpo por unidade de volume nas direções x e y , respectivamente, positivas nas direções desse eixos. σ_x e σ_y são as tensões normais nas direções dos eixos e τ_{xy} a tensão cisalhante mostrada na direção positiva na Figura 5.1.

Simplificando as expressões (5.1) obtém-se as equações diferenciais de equilíbrio

$$\begin{aligned}\frac{\partial \sigma_x}{\partial x} + \frac{\partial \tau_{xy}}{\partial y} + b_x &= 0 \\ \frac{\partial \tau_{xy}}{\partial x} + \frac{\partial \sigma_y}{\partial y} + b_y &= 0\end{aligned}\quad (5.2)$$

As condições de contorno sobre Γ podem ser tanto essenciais Γ_u como naturais Γ_f . A representação desses contornos não é tão simples como no caso da equação de Poisson, uma vez que agora não se tem um problema com incógnita escalar e sim um problema com duas incógnitas por ponto geométrico do domínio. Assim, as condições de contorno essenciais resultam em equações de compatibilidade na superfície devido a deslocamentos prescritos, também chamadas de condições de contorno de deslocamento

$$\begin{aligned}u_x &= \bar{u}_x & \text{em } \Gamma_u \\ u_y &= \bar{u}_y\end{aligned}\quad (5.3)$$

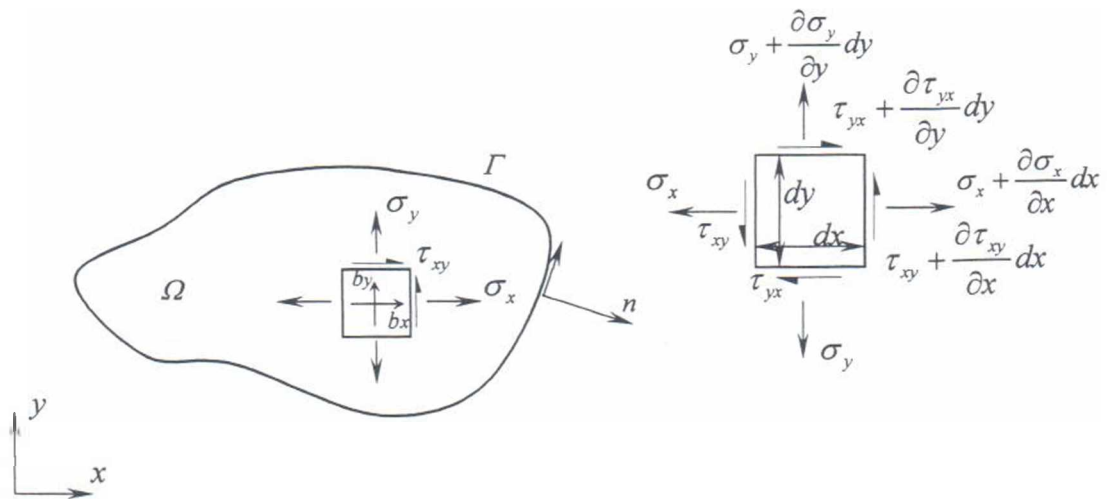


Figura 5.1 – Problema de elasticidade com diagrama de corpo livre.

As condições de contorno naturais são resultantes das equações de equilíbrio devidas a tensões prescritas na superfície, são também chamadas de condições de contorno de tensão e são dada por

$$\begin{aligned} \sigma_x n_x + \tau_{xy} n_y &= \bar{t}_x & \text{em } \Gamma_t \\ \tau_{xy} n_x + \sigma_y n_y &= \bar{t}_y \end{aligned} \quad (5.4)$$

onde n_x e n_y são os co-senos diretores do vetor unitário normal à superfície; e, \bar{t}_x e \bar{t}_y são as tensões prescritas.

De modo a adequar a formulação do problema à sua implementação computacional, as equações acima serão representadas na notação matricial. Assim as expressões acima se tornam

$$\begin{aligned} D\bar{\sigma} + \mathbf{b} &= \mathbf{0} & \text{em } \Omega \\ \sigma_n &= P_n \sigma = \mathbf{t} & \text{sobre } \Gamma_t \\ \mathbf{u} &= \bar{\mathbf{u}} & \text{sobre } \Gamma_u \end{aligned} \quad (5.5)$$

onde $\sigma = \{\sigma_x \quad \sigma_y \quad \tau_{xy}\}^T$ é o vetor de tensões; $\mathbf{b} = \{b_x \quad b_y\}^T$ é vetor das forças de corpo; $\mathbf{t} = \{\bar{t}_x \quad \bar{t}_y\}^T$ são as tensões impostas; $\mathbf{u} = \{u_x \quad u_y\}^T$ são os deslocamentos incógnitos; e,

$\bar{u} = \{\bar{u}_x \quad \bar{u}_y\}^T$ são os deslocamentos prescritos. A matriz D , denominada na literatura de gradiente simétrico, é dada por

$$D = \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix} \quad (5.6)$$

A matriz P_n é dada por

$$P_n = \begin{bmatrix} n_x & 0 & n_y \\ 0 & n_y & n_x \end{bmatrix} \quad (5.7)$$

O Método dos Resíduos Ponderados será aplicado às expressões (5.5) que representam as equações diferenciais que regem o problema de elasticidade, considerando-se que agora o problema possui duas incógnitas por ponto nodal

$$\int_{\Omega} w^T [D^T \sigma + b] d\Omega = 0 \quad (5.8)$$

$$\int_{\Gamma_t} \bar{w}^T [P_n \sigma - t] d\Gamma = 0 \quad (5.9)$$

$$\int_{\Gamma_u} \bar{\bar{w}}^T [u - \bar{u}] d\Gamma = 0 \quad (5.10)$$

onde $w = \{w_x \quad w_y\}^T$, $\bar{w} = \{\bar{w}_x \quad \bar{w}_y\}^T$ e $\bar{\bar{w}} = \{\bar{\bar{w}}_x \quad \bar{\bar{w}}_y\}^T$ são funções de ponderação nas direções x e y .

Integrando-se por partes o primeiro termo de (5.8) chega-se a

$$\int_{\Omega} w^T (D^T \sigma) d\Omega = - \int_{\Omega} (Dw)^T \sigma d\Omega + \int_{\Gamma} w^T P_n \sigma d\Gamma \quad (5.11)$$

Levando a expressão acima em (5.8), adotando, como no problema de Poisson, $\bar{\bar{w}}^T = w^T$, e subtraindo (5.9) de (5.8) chega-se às seguintes expressões de resíduos

$$-\int_{\Omega} (\mathbf{D}\mathbf{w})^T \boldsymbol{\sigma} d\Omega + \int_{\Omega} \mathbf{w}^T \mathbf{b} d\Omega + \int_{\Gamma_s} \mathbf{w}^T \mathbf{P}_n \boldsymbol{\sigma} d\Gamma + \int_{\Gamma_t} \mathbf{w}^T \mathbf{t} d\Gamma = 0 \quad (5.12)$$

$$\int_{\Gamma_u} \bar{\mathbf{w}}^T [\mathbf{u} - \bar{\mathbf{u}}] d\Gamma = 0 \quad (5.13)$$

Na expressão (5.12) a integral sobre o contorno essencial Γ_u é feita considerando-se a grandeza incógnita $\boldsymbol{\sigma}_n = \mathbf{P}_n \boldsymbol{\sigma}$, já que em Γ_u os deslocamentos \mathbf{u} são prescritos. No domínio vale a relação tensão-deformação da Lei de Hooke dada por

$$\boldsymbol{\sigma} = \mathbf{E} \boldsymbol{\varepsilon} \quad (5.14)$$

onde \mathbf{E} é a matriz de elasticidade função das propriedades do material que para o estado plano de tensão vale

$$\mathbf{E} = \frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & (1-\nu)/2 \end{bmatrix} \quad (5.15)$$

e para o estado plano de deformação é dada por

$$\mathbf{E} = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} (1-\nu) & \nu & 0 \\ \nu & (1-\nu) & 0 \\ 0 & 0 & (1-2\nu)/2 \end{bmatrix} \quad (5.16)$$

onde E é o módulo de Elasticidade e ν o coeficiente de Poisson. Ainda na equação (5.14), $\boldsymbol{\varepsilon} = \{\varepsilon_x \quad \varepsilon_y \quad \gamma_{xy}\}^T$ é a deformação específica que pode ser colocada em função dos deslocamentos

$$\boldsymbol{\varepsilon} = \mathbf{D}\mathbf{u} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix} \begin{Bmatrix} u_x \\ u_y \end{Bmatrix} \quad (5.17)$$

Em decorrência, as tensões postas em termos dos deslocamentos se transformam em

$$\boldsymbol{\sigma} = \mathbf{E} \boldsymbol{\varepsilon} = \mathbf{E} \mathbf{D} \mathbf{u} \quad (5.18)$$

onde todos os termos já foram definidos. Substituindo nas equações de resíduos chega-se, após separação das grandezas incógnitas das grandezas prescritas, a

$$\int_{\Omega} (\mathbf{D}\mathbf{w})^T \mathbf{E} \mathbf{D} \mathbf{u} d\Omega + \int_{\Gamma_u} \mathbf{w}^T \boldsymbol{\sigma}_n d\Gamma = \int_{\Omega} \mathbf{w}^T \mathbf{b} d\Omega + \int_{\Gamma_t} \mathbf{w}^T \mathbf{t} d\Gamma \quad (5.19)$$

$$\int_{\Gamma_u} \bar{\mathbf{w}}^T \mathbf{u} d\Gamma = \int_{\Gamma_u} \bar{\mathbf{w}}^T \bar{\mathbf{u}} d\Gamma \quad (5.20)$$

A discretização será realizada expandindo-se os deslocamentos do domínio, Ω , e as trações no contorno Γ_u pelos pontos nodais adotados para a solução discreta com a utilização de funções de forma adequadas. Assim

$$\mathbf{u} = \begin{Bmatrix} u_x \\ u_y \end{Bmatrix} = \boldsymbol{\varphi}(x) \mathbf{a} \quad (5.21)$$

$$\boldsymbol{\sigma}_n = \begin{Bmatrix} \sigma_n^x \\ \sigma_n^y \end{Bmatrix} = \mathbf{N}(x) \boldsymbol{\sigma}_u \quad (5.22)$$

Os vetores de deslocamentos nodais e as trações no contorno essencial são definidos por

$$\mathbf{a} \equiv \begin{Bmatrix} a_{lx} \\ a_{ly} \\ \vdots \\ a_{nx} \\ a_{ny} \end{Bmatrix} \quad \boldsymbol{\sigma}_u \equiv \begin{Bmatrix} \sigma_{lx} \\ \sigma_{ly} \\ \vdots \\ \sigma_{rx} \\ \sigma_{ry} \end{Bmatrix} \quad (5.23)$$

onde o primeiro sobrescrito se refere ao ponto nodal e o segundo indica a componente. Convém perceber que o número de pontos nodais que discretiza \mathbf{a} e $\boldsymbol{\sigma}_u$ são indicados como diferentes.

As matrizes com as funções de forma são dadas por

$$\begin{aligned}\varphi(x) &= [\varphi_1 \quad \varphi_2 \quad \cdots \quad \varphi_n] \\ N(x) &= [N_1 \quad N_2 \quad \cdots \quad N_r]\end{aligned}\tag{5.24}$$

onde $\varphi_l = \phi_l \mathbf{I}_2$, $N_l = N_l \mathbf{I}_2$, ou de forma mais explícita

$$\begin{aligned}\varphi(x) &= \left[\begin{array}{c|c|c|c} \phi_1 & \cdot & \phi_2 & \cdot & \cdots & \phi_n & \cdot \\ \cdot & \phi_1 & \cdot & \phi_2 & \cdots & \cdot & \phi_n \end{array} \right] \\ N(x) &= \left[\begin{array}{c|c|c|c} N_1 & \cdot & N_2 & \cdot & \cdots & N_r & \cdot \\ \cdot & N_1 & \cdot & N_2 & \cdots & \cdot & N_r \end{array} \right]\end{aligned}\tag{5.25}$$

A notação para as funções de ponderação foi considerada no âmbito da formulação infinitesimal. Colocando agora do mesmo modo que as funções de forma em uma formulação discreta, as funções de ponderação se transformam em

$$\mathbf{w}(x) = \left[\begin{array}{c|c|c|c} w_1 & \cdot & w_2 & \cdot & \cdots & w_n & \cdot \\ \cdot & w_1 & \cdot & w_2 & \cdots & \cdot & w_n \end{array} \right]\tag{5.26}$$

Substituindo essas matrizes com a formulação discreta nas equações de resíduos ponderados (5.19) e (5.20) chega-se a

$$\int_{\Omega} (\mathbf{D}\mathbf{w})^T \mathbf{E} \mathbf{D}\varphi \mathbf{a} \, d\Omega + \int_{\Gamma_u} \mathbf{w}^T \mathbf{N} \sigma_u \, d\Gamma = \int_{\Omega} \mathbf{w}^T \mathbf{b} \, d\Omega + \int_{\Gamma_t} \mathbf{w}^T \mathbf{t} \, d\Gamma\tag{5.27}$$

$$\int_{\Gamma_u} \bar{\bar{\mathbf{w}}}^T \varphi \mathbf{a} \, d\Gamma = \int_{\Gamma_u} \bar{\bar{\mathbf{w}}}^T \bar{\mathbf{u}} \, d\Gamma\tag{5.28}$$

Substituindo a forma discreta dos deslocamentos na relação tensão deformação obtém-se

$$\boldsymbol{\sigma} = \mathbf{E} \mathbf{D} \varphi \mathbf{a} = \mathbf{E} \mathbf{B} \mathbf{a}\tag{5.29}$$

onde \mathbf{B} é definida por

$$\mathbf{B} = [\mathbf{B}_1 \mid \mathbf{B}_2 \mid \cdots \mid \mathbf{B}_n]$$

com

$$B_l = \begin{bmatrix} \phi_{l,x} & 0 \\ 0 & \phi_{l,y} \\ \phi_{l,y} & \phi_{l,x} \end{bmatrix} \quad (5.30)$$

Formato tradicional de apresentação no Método dos Elementos Finitos.

Substituindo B nas expressões discretas de resíduos ponderados e observando que as incógnitas discretas a e σ_u são constantes no processo de integração, a seguinte expressão resulta

$$\begin{bmatrix} K & H \\ G^T & 0 \end{bmatrix} \begin{Bmatrix} a \\ \sigma_u \end{Bmatrix} = \begin{Bmatrix} f \\ f_u \end{Bmatrix} \quad (5.31)$$

onde

$$K = \int_{\Omega} (Dw)^T E B d\Omega \quad (5.32)$$

$$H = - \int_{\Gamma_u} w^T N d\Gamma \quad (5.33)$$

$$G = \int_{\Gamma_u} \bar{w}^T \phi d\Gamma \quad (5.34)$$

$$f = \int_{\Omega} w^T b d\Omega + \int_{\Gamma_i} w^T t d\Gamma \quad (5.35)$$

$$f_u = \int_{\Gamma_u} \bar{w}^T \bar{u} d\Gamma \quad (5.36)$$

Neste ponto será aplicado um dos critérios que caracterizam o Método dos Resíduos Ponderados. Neste caso será empregado o critério de Galerkin que permitirá resolver um problema de elasticidade por qualquer dos métodos baseados em Galerkin nomeados neste trabalho, vale dizer, Método de Galerkin sem Malha, objeto deste capítulo;

Método sem Malha Hierárquico (*Hp-Clouds*); e até mesmo o Método dos Elementos Finitos.

Fazendo, então, $w = \varphi$, e, como as funções de ponderação são arbitrárias, adotando $\bar{w} = -N$ chega-se

$$\begin{bmatrix} \mathbf{K} & \mathbf{G} \\ \mathbf{G}^T & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \mathbf{a} \\ \sigma_u \end{Bmatrix} = \begin{Bmatrix} \mathbf{f} \\ \mathbf{f}_u \end{Bmatrix} \quad (5.37)$$

com

$$\mathbf{K} = \int_{\Omega} \mathbf{B}^T \mathbf{E} \mathbf{B} d\Omega \quad (5.38)$$

$$\mathbf{G} = - \int_{\Gamma_u} \varphi^T \mathbf{N} d\Gamma \quad (5.39)$$

$$\mathbf{f} = \int_{\Omega} \varphi^T \mathbf{b} d\Omega + \int_{\Gamma_t} \varphi^T \mathbf{t} d\Gamma \quad (5.40)$$

$$\mathbf{f}_u = - \int_{\Gamma_u} \mathbf{N}^T \bar{\mathbf{u}} d\Gamma \quad (5.41)$$

Seria possível empregar o critério de Colocação do Método dos Resíduos Ponderados, que permitiria a solução do problema de elasticidade pelo Método das Diferenças Finitas Generalizadas ou pelo Método do Ponto Finito, a partir de uma formulação fraca, como realizado no Capítulo 2. Este caminho será relacionado para trabalho futuro.

As funções de forma N sobre o contorno essencial Γ_u poderiam ser adotadas iguais às funções de forma empregadas no domínio φ . Contudo deve-se observar que no contorno os requisitos de continuidade não são tão severos quanto no domínio. Assim basta adotar no contorno funções contínuas C^0 , sem necessidade de continuidade das derivadas, uma vez que basicamente trata-se de obter a aproximação da função ao longo do contorno. Nestas condições serão adotadas funções de forma polinomiais lineares, como as utilizadas nos elementos isoparamétricos, listadas no Capítulo 3.

Para se chegar à formulação do Método dos Elementos Finitos, a partir da formulação Galerkin apresentada, algumas considerações acerca das funções de forma devem ser postas. Primeiro, as funções do domínio pertencentes a φ devem ser adotadas de modo que seus valores se anulem ao longo de Γ_u . Segundo, as funções φ e N devem atender à condição delta de Kronecker, vale dizer, devem ter valor unitário no ponto nodal sob consideração, e valor nulo em todos os demais. E, terceiro as funções N que discretizam o contorno Γ_u devem ser descontínuas, C^{-1} , com valor unitário sobre o ponto nodal e zero em todo o restante. Uma função dessa natureza está desenhada na Figura 5.2.

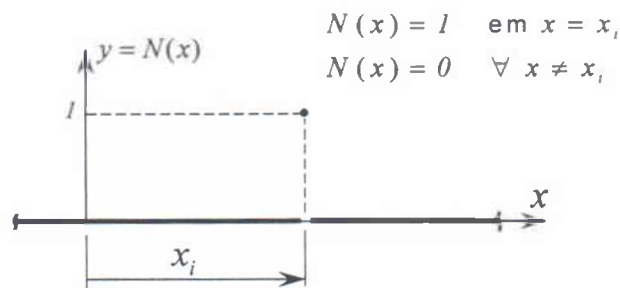


Figura 5.2 – Funções descontínuas para discretização do contorno essencial Γ_u .

De modo a melhor entender o resultado das três restrições das funções de forma sobre o sistema discreto do Método dos Resíduos Ponderados, seja o vetor dos deslocamentos nodais particionado $\mathbf{a} = \{\mathbf{a}_1 \quad \mathbf{a}_2\}$ de tal modo que em \mathbf{a}_1 se localizem as incógnitas que não estão sobre Γ_u (ou seja aquelas que estão em Γ_f e aquelas que estão no domínio sem o contorno), e, em \mathbf{a}_2 as incógnitas referentes ao contorno Γ_u . Dessa maneira o sistema (5.37) pode ser reorganizado como a seguir

$$\begin{bmatrix} \mathbf{K}_{11} & \mathbf{K}_{12} & \mathbf{G}_1 \\ \mathbf{K}_{21} & \mathbf{K}_{22} & \mathbf{G}_2 \\ \mathbf{G}_1^T & \mathbf{G}_2^T & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \\ \sigma_u \end{Bmatrix} = \begin{Bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \\ \mathbf{f}_u \end{Bmatrix} \quad (5.42)$$

A aplicação da primeira restrição, qual seja que as funções do domínio possuem valor nulo quando avaliadas em Γ_u , faz com que a matriz \mathbf{G}_1 seja identicamente nula, como abaixo

$$\mathbf{G}_1 = - \int_{\Gamma_u} \varphi_i^T(\Gamma_u) N d\Gamma \equiv \mathbf{0} \quad (5.43)$$

A adoção da função de ponderação descontínua (Figura 5.2) para discretização do contorno essencial faz com que a matriz \mathbf{G}_2 assuma a forma

$$\mathbf{G}_2 = - \int_{\Gamma_u} \varphi_i^T(x_i) N(x_j) d\Gamma = \begin{cases} -I & \text{para } x_i = x_j \\ 0 & \text{para } x_i \neq x_j \end{cases} \quad (5.44)$$

ou seja transforma-se numa matriz identidade, a menos do sinal

$$\mathbf{G} = -\mathbf{I}_r \quad (5.45)$$

onde r é o número de pontos nodais no contorno Γ_u .

Da mesma forma o vetor com deslocamentos prescritos equação (5.41) torna-se

$$\begin{aligned} \mathbf{f}_u &= \left(- \int_{\Gamma_u} N^T(x_i) N(x_j) d\Gamma \right) \bar{\mathbf{u}}_p \\ \mathbf{f}_u &= -\mathbf{I}_r \bar{\mathbf{u}}_p \end{aligned} \quad (5.46)$$

onde $\bar{\mathbf{u}}_p$ são os valores nos pontos do vetor com deslocamentos prescritos $\bar{\mathbf{u}} = \mathbf{N} \bar{\mathbf{u}}_p$.

Assim o sistema (5.42) resulta

$$\begin{bmatrix} \mathbf{K}_{11} & \mathbf{K}_{12} & \mathbf{0} \\ \mathbf{K}_{21} & \mathbf{K}_{22} & -\mathbf{I}_r \\ \mathbf{0} & -\mathbf{I}_r & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \\ \sigma_u \end{bmatrix} = \begin{bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \\ -\mathbf{I}_r \bar{\mathbf{u}}_p \end{bmatrix} \quad (5.47)$$

Portanto ocorre um desacoplamento do sistema, com \mathbf{a}_2 , deslocamentos no contorno, assumindo o seguinte valor em decorrência da terceira equação em (5.47)

$$\begin{aligned} -\mathbf{I}_r \mathbf{a}_2 &= -\mathbf{I}_r \bar{\mathbf{u}}_p \\ \mathbf{a}_2 &= \bar{\mathbf{u}}_p \end{aligned} \quad (5.48)$$

Substituindo o valor de $\mathbf{a}_2 = \bar{\mathbf{u}}_p$ na primeira equação de (5.47) vem

$$\mathbf{a}_1 = \mathbf{K}_{11}^{-1}(\mathbf{f}_1 - \mathbf{K}_{12}\bar{\mathbf{u}}_p) \quad (5.49)$$

e a partir da segunda equação de (5.47) tem-se

$$\boldsymbol{\sigma}_u = -(\mathbf{f}_2 - \mathbf{K}_{21}\mathbf{a}_1 - \mathbf{K}_{22}\bar{\mathbf{u}}_p) \quad (5.50)$$

Assim no Método dos Elementos Finitos trabalha-se somente com as incógnitas que não estão sobre o contorno essencial, facilitando a solução do sistema de equações. Entretanto a adoção de uma função descontínua para a discretização das tensões no contorno essencial resulta, de uma maneira geral, em baixa qualidade da aproximação. Sem mencionar que os resultados nas variáveis secundárias (tensões) no domínio são obtidas por derivação de funções de ordem de continuidade C^0 resultando em aproximações C^{-1} , portanto funções constantes, oferecendo valores de aproximação bastante sofríveis em grandezas bastante relevantes na prática da engenharia.

5.3 - TRATAMENTO DE DESCONTINUIDADE DE MATERIAL

É comum em problemas de elasticidade a existência de meios não-homogêneos, ou seja, com mais de um material constituindo o corpo. Devido a essa importância serão apresentadas a seguir as linhas de ataque, vistas na literatura, para abordagem desse problema, ficando a sua implementação para trabalhos futuros.

Cordes e Moran (1996) apresentam um método baseado no tratamento da interface com enfoque variacional. O meio não-homogêneo é separado em meios homogêneos e a seguir é imposta uma condição de interface de modo à re-conectar as partes. Formato semelhante ao visto no Método dos Elementos de Contorno (Brebbia e Dominguez, 1989). A Figura 5.3 apresenta uma representação dessa abordagem.

A formulação variacional de Cordes e Moran pode ser deduzida com bastante similaridade pelo Método dos Resíduos Ponderados, seguindo os passos apresentados na seção anterior.

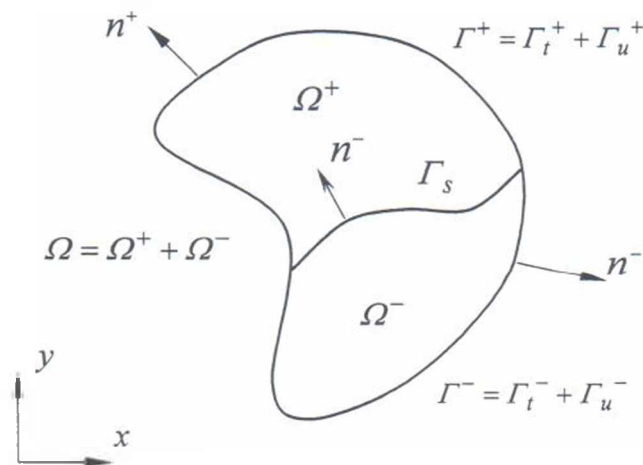


Figura 5.3 - Corpo não-homogêneo.

A questão crucial na aplicação do Método de Galerkin sem Malha a problemas de descontinuidade de material reside na escolha adequada do domínio de influência de cada nó. Assim, o algoritmo de construção da nuvem de pontos deve ter algum critério de visibilidade que impeça a inclusão de um ponto nodal não pertencente ao subdomínio homogêneo em análise.

Os pontos que definem a interface foram tratados na implementação de Cordes e Moran como pontos gêmeos, ou duplos, um para cada lado do meio que a interface separa. Procedimento já comentado no Capítulo 2. A Figura 5.4 ilustra modelo desta implementação destacando-se ainda a estrutura de células de integração utilizadas, que emprega os pontos nodais para a sua definição. De modo a melhor integrar, próximo da interface ocorre um adensamento das células de integração construídas de forma totalmente independente dos pontos nodais.

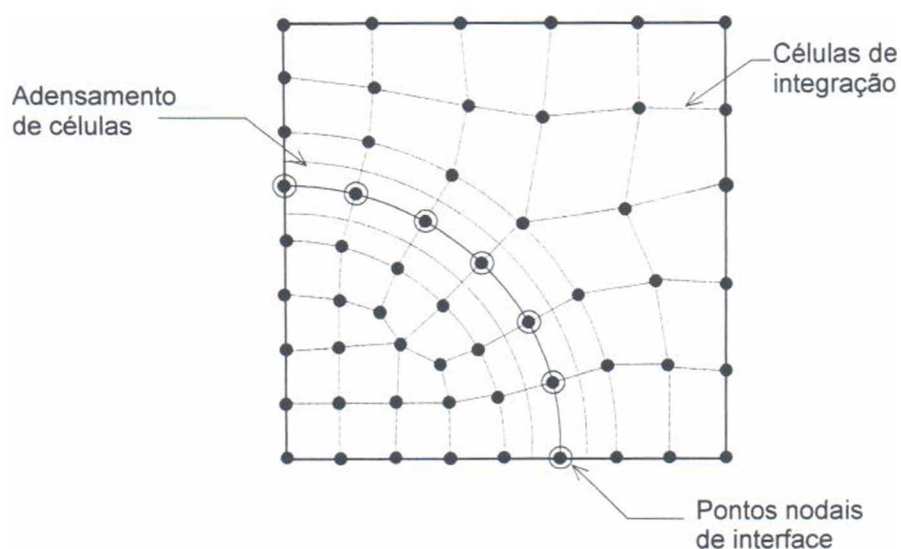


Figura 5.4 – Tratamento de interface entre dois materiais.

Os resultados apresentados são bastante precisos, embora alguma oscilação seja observada nos valores baseados nas derivadas. Alguma explicação pode residir no fato de ter sido utilizada uma forma variacional modificada ao invés de técnicas baseadas em multiplicadores de Lagrange.

Krongauz e Belytshko (1998) apresentam, uma técnica alternativa para tratamento de meios não-homogêneos ou casos de descontinuidades nas derivadas de forma geral. A técnica enriquece as funções de aproximação através da adição de funções de forma especiais que contém descontinuidades nas derivadas. A função de aproximação transforma-se em

$$u^h(x) = u^{\text{EFG}} + \sum_{j=1}^{nd} q_j \Psi_j(x) \quad (5.51)$$

onde q_j são parâmetros de amplitude; $\Psi_j(x)$ são funções de forma com salto; e nd é o número de pontos nodais onde ocorre descontinuidade.

O termo u^{EFG} é a aproximação padrão do Método de Galerkin sem Malha

$$u^{\text{EFG}}(x) = \sum_{i=1}^n \phi_i u_i \quad (5.52)$$

Os parâmetros de amplitude q_j são considerados como incógnitas no processo de discretização, resultando em equações adicionais. Quando a localização da descontinuidade pode ser inferida a partir do problema proposto, como no caso de meios não-homogêneos em elasticidade, as equações adicionais são então lineares em relação às incógnitas do problema.

As funções Ψ_j que geram a descontinuidade das derivadas são desenhadas de tal sorte que garanta a existência de um suporte compacto; que sejam linearmente independentes com relação às funções de forma ϕ_i geradas pelo método dos mínimos quadrados ponderados móvel; e, que sejam capazes de reproduzir a função salto de Heaviside que reproduz a descontinuidade das derivadas.

Duas funções foram utilizadas por Krongauz e Belytshko. A primeira é uma função *spline* cúbica na forma

$$\Psi(r) = \begin{cases} -\frac{1}{6}r^3 + \frac{1}{2}r^2 - \frac{1}{2}r + \frac{1}{6} & r \leq 1 \\ 0 & r > 1 \end{cases} \quad (5.53)$$

onde r é a distância normalizada a partir da descontinuidade. A segunda função é

$$\Psi(x) = \langle x - x_a \rangle - \sum \phi_i(x) \langle x_i - x_a \rangle \quad (5.54)$$

onde a função $\langle x \rangle$ é

$$\langle x \rangle = \begin{cases} 0 & x < 0 \\ x & x \geq 0 \end{cases} \quad (5.55)$$

Os resultados dos experimentos numéricos, confrontados com soluções analíticas, são bastante precisos tanto no valor da descontinuidade quanto na extensão de sua influência. Ao contrário da formulação de subdomínio homogêneos conectados numa formulação variacional, referida acima, oscilações dos resultados na superfície da descontinuidade são evitadas.

Embora apresentado para o Método de Galerkin sem Malha, esta técnica pode ser utilizada por outros métodos cujas funções de forma sejam obtidas por mínimos quadrados ponderados móveis. Neste caso a aproximação u^{EFG} deve ser adequadamente substituída.

5.4 - ALGORITMO DO MÉTODO DE GALERKIN SEM MALHA

O procedimento de solução empregado no Método de Galerkin sem Malha segue, de forma geral, os passos decorrentes da aplicação do critério de Galerkin dentro do Método de Resíduos Ponderados. De fato a implementação é bastante similar à organização do Método dos Elementos Finitos.

Vários autores apresentam um fluxograma para o Método de Galerkin sem Malha (Belytscko *et al.*, 1994; Lu *et al.*, 1994; Hegen, 1996; Dolbow e Belytscko , 1998; Du, 2000). As concepções das implementações são bastante similares. Na maioria desses algoritmos a montagem das matrizes globais é feita a partir da aproximação local de cada ponto de integração gaussiano. Hegen (1996) realiza a montagem das matrizes globais a partir da contribuição de cada célula. Como o número de pontos da nuvem pode variar em cada ponto gaussiano, deve-se realizar um *assembler* local. Ambos procedimentos são equivalentes. A opção é mais de ordem computacional de forma a permitir que o formato tradicional de *assembler* por elementos tenha objetos compatíveis.

A Tabela 5.1 apresenta o fluxograma do Método de Galerkin sem Malha implantado. Na verdade é o mesmo para todos os Métodos sem Malha deste trabalho, diferindo no procedimento de integração que no critério de Galerkin é feito por quadratura gaussiana e no critério de colocação a integração é a soma das parcelas de cada ponto nodal, decorrência da ponderação com a função delta de Dirac.

Além da marcha de cálculo para obtenção da solução, o algoritmo da Tabela 5.1 também apresenta as etapas de definição dos objetos que modelam um problema específico. O algoritmo foi desenvolvido com pseudocomandos, marcados em negrito. A seguir são comentadas as etapas do fluxograma ficando a aplicação numérica para a Seção 5.5.

Inicialmente é definido o tipo de problema, que no caso é de elasticidade bidimensional com pequenas deformações, e fornecidas as propriedades dos materiais a serem empregados. A estrutura computacional que permite a criação de novos problemas será apresentada na Seção 5.6.

Tabela 5.1 – Algoritmo do Método de Galerkin sem Malha.

```

defina o Tipo de Problema e as Propriedades dos Materiais.
defina o Domínio e introduza a descrição geométrica e condições de contorno.
defina a Discretização do contorno, do domínio e das células de integração.
defina a Base de Funções.
defina a Função de Ponderação.
defina o Critério de Construção das Nuvens de Pontos.
defina as Funções de Forma.
loop sobre as células de integração  $C$  do Domínio.
    loop sobre os pontos de quadratura  $x_q$  da célula  $C$ .
        caso  $x_q$  esteja fora do Domínio, então próximo  $x_q$ .
        construa nuvem de pontos de  $x_q$  com o critério definido.
        calcule as funções de forma  $\phi_i(x_q)$  e suas derivadas  $\phi_{i,s}(x_q)$ .
        construa a aproximação local  $(\mathbf{K}^h, \mathbf{G}^h, \mathbf{f}^h, \mathbf{f}_u^h)$ .
        assembler matrizes globais  $\mathbf{K}, \mathbf{G}, \mathbf{f}, \mathbf{f}_u$ .
    end integração.
end loop sobre as células.
resolva sistema 
$$\begin{bmatrix} \mathbf{K} & \mathbf{G} \\ \mathbf{G}^T & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \mathbf{a} \\ \boldsymbol{\sigma}_u \end{Bmatrix} = \begin{Bmatrix} \mathbf{f} \\ \mathbf{f}_u \end{Bmatrix}$$

determine grandezas secundárias: tensões, erros etc
    
```

Em seguida é definida a geometria do domínio. A descrição é realizada com técnicas de CAD (*Computer Aided Design*) com criação de pontos nodais e segmentos de reta que definem o corpo. Outras curvas podem ser implementadas, tais como parábolas e *splines* cúbicas. Além da seqüência do caminho do contorno externo, outros caminhos internos podem ser introduzidos, permitindo a definição de furos. Outros detalhes serão apresentados no Capítulo 6. A Figura 5.5 representa o estado do objeto domínio até esta etapa do algoritmo.

Problema : Elasticidade
Constantes: E, ν

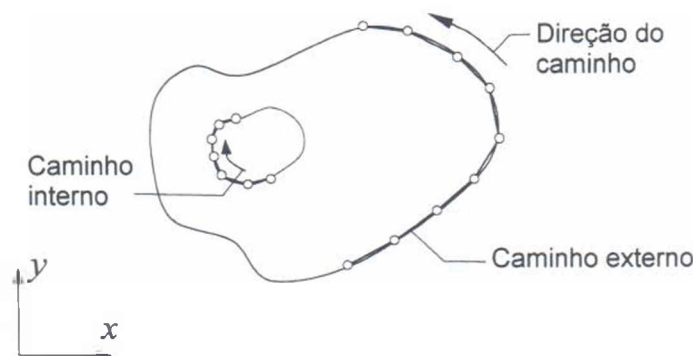


Figura 5.5 – Definição dos objetos do problema e do domínio.

Neste ponto deve ser realizada a discretização do domínio definido na etapa anterior. Optou-se por discretizar de forma separada o contorno do objeto de seu domínio (região interna ao contorno). Assim garante-se uma melhor definição do contorno do problema, inclusive com a criação se necessário de pontos gêmeos. A discretização pode ser realizada por vários critérios: geração a partir de uma grade ortogonal; aleatória; dentre outros que podem ser implementados. A Figura 5.6 representa um problema nesta etapa. Cabe comentar que pode ser criada nesta etapa uma discretização por elementos finitos, por exemplo, evitando a separação contorno-domínio.

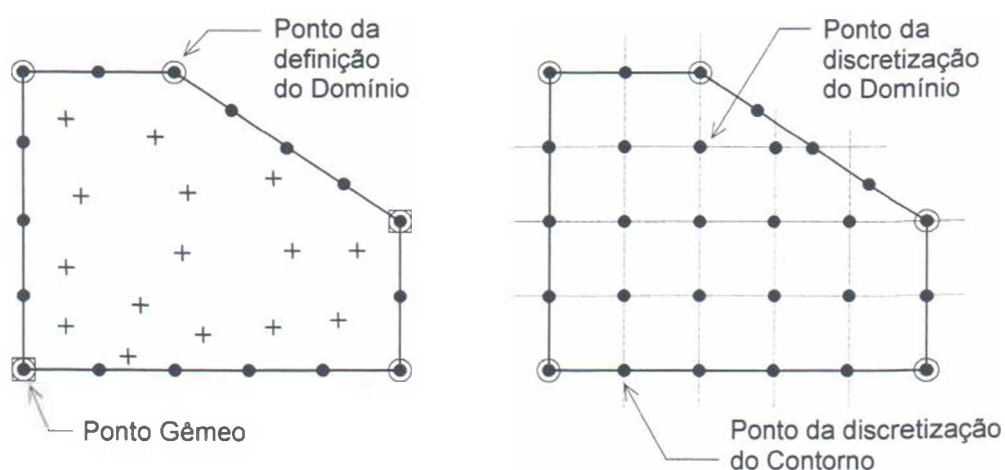


Figura 5.6 – Critérios de discretização aleatório e grade ortogonal.

Outra discretização que deve ser definida sobre o domínio é a criação de células ou zonas de integração. Belytschko *et al.* (1994), no trabalho seminal do Método de Galerkin sem Malha, utilizaram uma grade ortogonal definida a partir de um retângulo que envolve todo o domínio (*bounding box*), como mostra a Figura 5.7. Assim, caso um ponto de integração esteja fora do polígono que define a região do domínio é simplesmente abandonado. Kaljević e Saigal (1997) utilizaram um critério para as células próximas a contornos irregulares que se assemelha ao procedimento de *clipping* de figuras utilizado em CAD (Foley *et al.*, 1990). Uma alternativa seria realizar a integração com células triangulares geradas a partir de uma triangulação válida restringida pelo domínio e seus furos. Um algoritmo desta natureza foi implementado seguindo Preparata e Shamos (1985). A Figura 5.7 apresenta duas discretizações de células de integração.

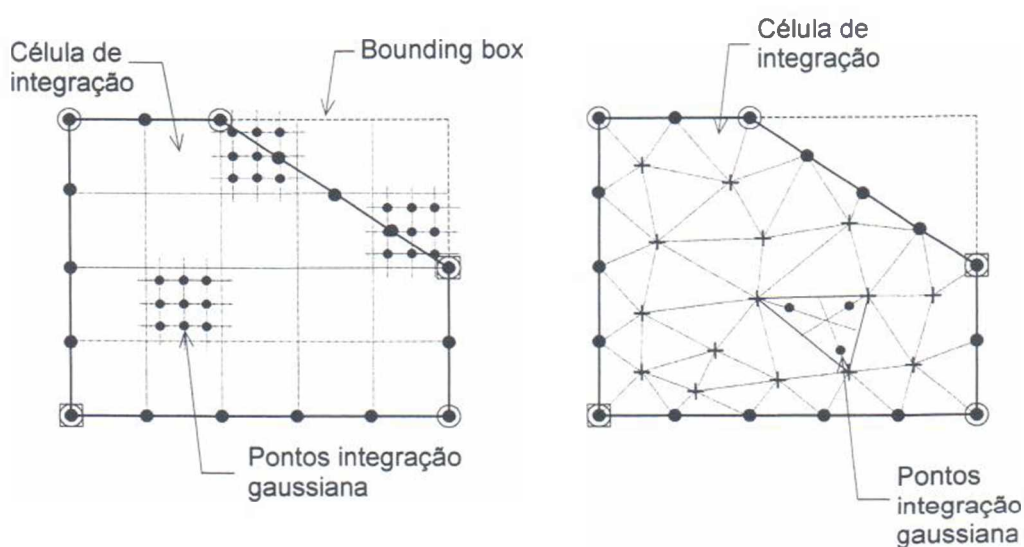


Figura 5.7 – Discretização de células de integração.

As integrações são realizadas utilizando-se a quadratura gaussiana. Como as funções de forma e suas derivadas se comportam como polinômios de ordem elevada é necessário que se utilize ordem de integração mais elevadas, iniciando-se com 3×3 pontos gaussianos. Deve-se mencionar que um refinamento das células de integração é mais utilizado do que o uso desmesurado do número de pontos gaussianos.

O procedimento de integração é bastante semelhante ao empreendido nos elementos finitos isoparamétricos. Dada a liberdade na definição das células, devem-se

estabelecer proporções que evitem um determinante jacobiano negativo. Neste caso deve-se utilizar subdivisão das células, principalmente para células triangulares.

A próxima etapa deve definir a construção das funções de forma. Como visto no Capítulo 3 alguns objetos devem ser criados para a definição das funções de forma que utilizam o método dos mínimos quadrados ponderados móveis. A base de funções empregada em problemas de elasticidade é a base linear que atende aos critérios de reprodução do movimento de corpo rígido. Já quanto às funções de ponderação além da função gaussiana truncada, também pode ser utilizada a função spline cúbica, por exemplo, apresentada no Capítulo 3.

Mais laborioso na definição das funções de forma é a definição de um critério para a construção da nuvem de pontos sob o domínio de influência de um ponto de integração gaussiano. Como referido no Capítulo 3, várias estratégias ou algoritmos podem ser estabelecidos. O mais simples, talvez seja o que define um número mínimo de pontos nodais, no caso $m = 3$ (polinômio linear). De modo que haja alguma eficiência na busca dos m pontos mais próximos deve ser utilizado algum algoritmo de busca binária com ordenação lexicográfica dos pontos nodais (de Berg *et al.*, 2000).

Outro algoritmo, sugerido por Belytschko *et al.* (1994), é a construção a partir da escolha dos m pontos mais próximos que constroem um polígono contornando o ponto de integração. Neste caso o polígono é um triângulo que pode ser obtido com o algoritmo de triangulação mencionado na discretização das células de integração triangulares. O ponto mais distante dos três é o escolhido para diâmetro do suporte do ponto de integração. A Figura 5.8 representa os pontos nodais onde se busca a solução e os pontos gaussianos com a definição do suporte.

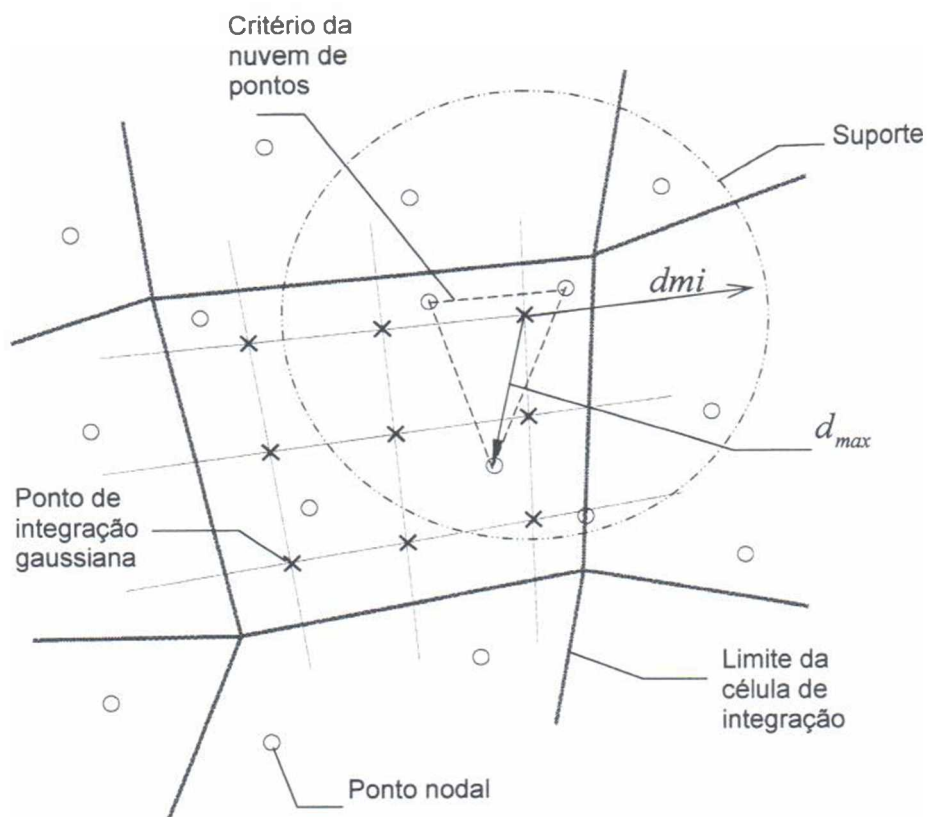


Figura 5.8 – Critério de construção poligonal da nuvem de pontos.

A partir da definição do diâmetro de suporte d_{mi} de um ponto gaussiano x_q deve-se construir a lista com pontos nodais dentro desse suporte para determinação das funções de forma, que, com o critério estabelecido, são obtidas sem singularidade na matriz de equações normais.

No processo de integração numérica, para montagem das matrizes globais que representam a discretização, alguns comentários são necessários. Primeiro, caso seja utilizada a grade de células definida pelo *bounding box*, pode ocorrer que o ponto gaussiano se localize fora do polígono que define o domínio. A determinação da localização de um ponto em relação a um polígono pode ser empreendida com o algoritmo que determina se o número de vezes que um raio horizontal partindo desse ponto corta o polígono é par ou ímpar (Preparata e Shamos, 1985). A implementação realizada baseia-se na apresentação de Harrington (1987).

Outro comentário é sobre a construção de matrizes com a aproximação local $(\mathbf{K}^h, \mathbf{G}^h, \mathbf{f}^h, \mathbf{f}_u^h)$ de uma célula. Cada ponto gaussiano pode ter uma lista de pontos nodais diferentes em seu domínio de influência. Assim a lista de pontos nodais que constroem a célula é o conjunto união das listas dos pontos gaussianos. Deve-se observar que o número total de pontos nodais que influenciam cada célula é geralmente diferente. Isto também é comum no *assembler* de elementos finitos com número variável de nós por elemento.

5.5 - EXEMPLO DE APLICAÇÃO

O exemplo a seguir apresenta alguns resultados do Método de Galerkin sem Malha aplicado à elasticidade como descrito no decorrer deste capítulo.

A Figura 5.9 mostra uma chapa solicitada por dois casos de carregamento. O problema é de longa data utilizado para verificação numérica no âmbito do Método dos Elementos Finitos (Belytschko *et al.*, 1994; Taylor *et al.*, 1976). A solução analítica é fornecida por Timoshenko e Goodier (1970). Para o caso de $E=1$ e $\nu=1/4$, os deslocamentos valem para o caso de carregamento 1 (vide Figura 5.9)

$$\begin{aligned} u_x &= x \\ u_y &= -y/4 \end{aligned} \quad (5.56)$$

e para o caso de carregamento 2

$$\begin{aligned} u_x &= xy \\ u_y &= (-4x^2 - y^2)/8 \end{aligned} \quad (5.57)$$

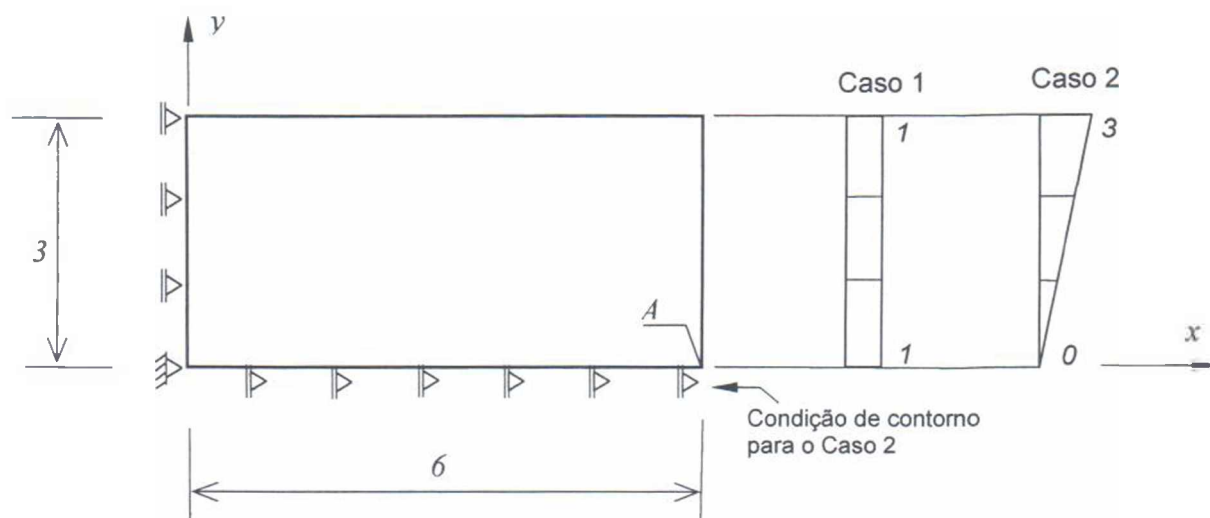


Figura 5.9 – Chapa submetida à tração axial.

A Figura 5.10 representa as soluções analíticas para os dois casos de carregamento.

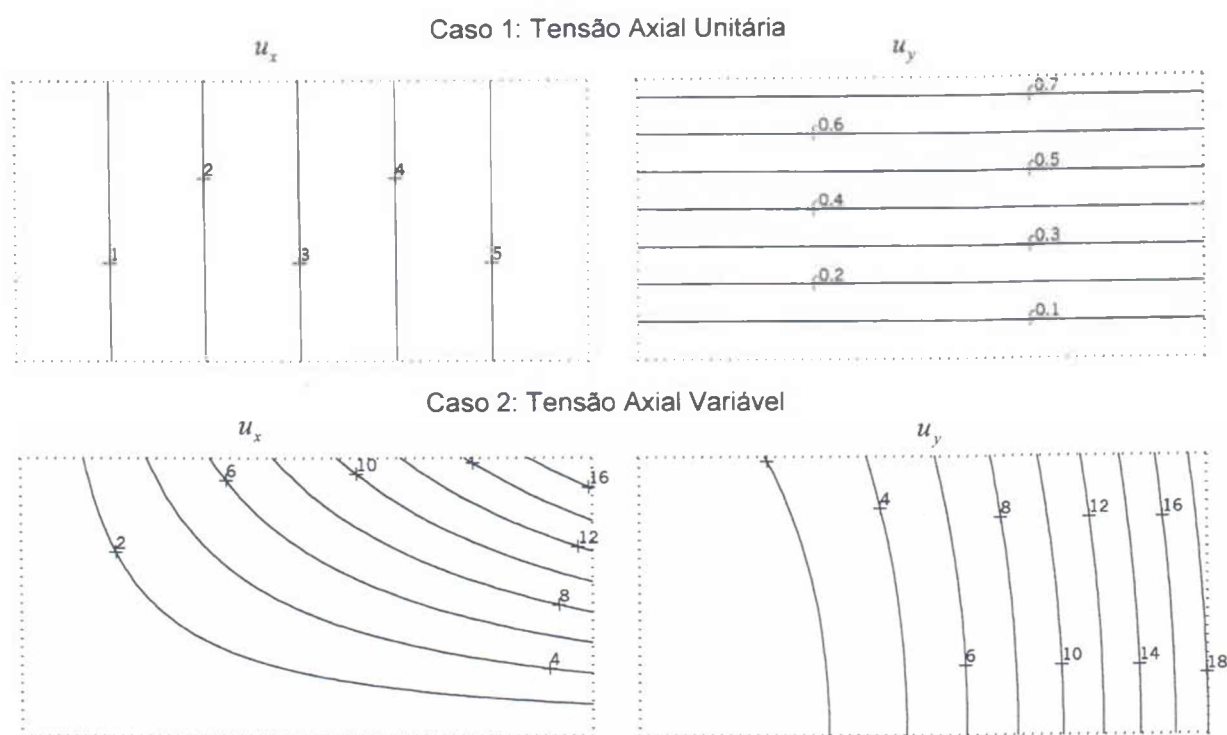


Figura 5.10 – Solução analítica para a chapa submetida à tração axial.

A Figura 5.11 apresenta a discretização adotada para o contorno, para o domínio e as células de integração utilizadas. Estão representadas duas distribuições de pontos no domínio: uma uniforme e outra como pontos gerados de forma aleatória.

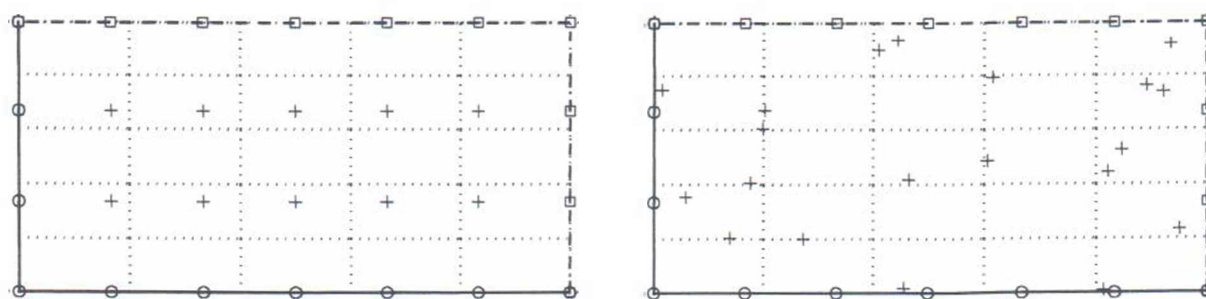


Figura 5.11 – Discretização da chapa: pontos nodais e células de integração.

A Figura 5.12 mostra a variação do erro relativo no ponto A (vide Figura 5.9) em função do parâmetro α , que controla a dimensão do suporte. Utilizou-se base de

funções linear e função de ponderação gaussiana. Foram traçadas curvas para cada caso de carregamento e de distribuição de pontos no domínio.

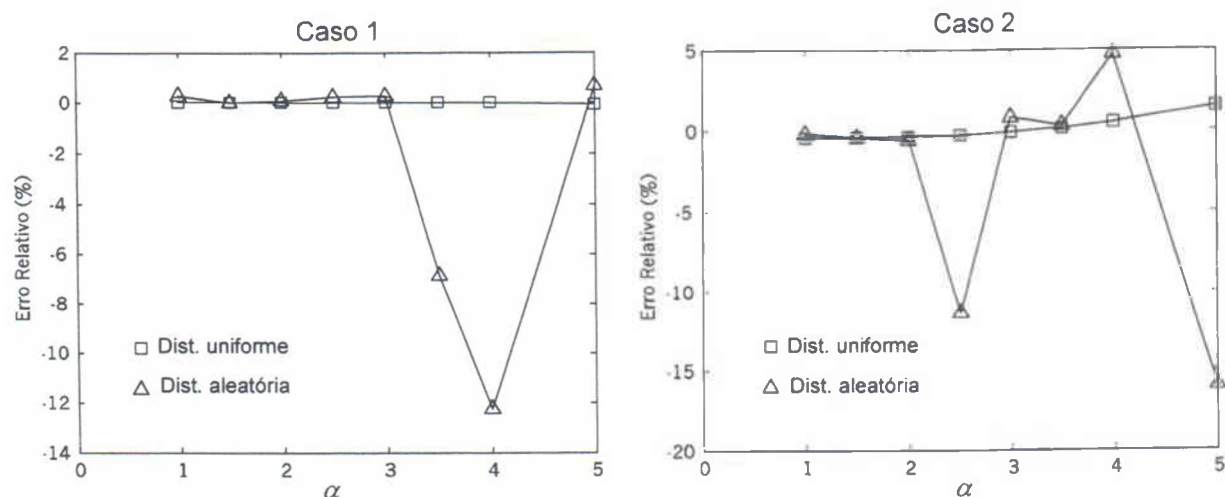


Figura 5.12 – Variação do erro relativo com o parâmetro α , para os casos de carregamento e de distribuição de pontos nodais.

A seguir é realizada uma avaliação da influência da escolha da função de ponderação sobre a solução. A Figura 5.13 apresenta o erro relativo no ponto A em função do parâmetro α para as funções de ponderação denominadas de Gauss, B3Spline e B4Spline conforme representação na Tabela 3.1. Utilizou-se o caso 1 de carregamento e distribuição uniforme de pontos nodais.

Na Figura 5.14 consta um estudo da influência da ordem da base. Utilizou-se base linear e quadrática. A base constante não produz convergência no exemplo apresentado. Novamente fez-se uso do caso 1 de carregamento distribuição uniforme e função de ponderação gaussiana.

De forma a averiguar o comportamento da solução discreta em todo o domínio a Figura 5.15 apresenta as curvas de nível com a solução exata e discreta, sendo utilizada base linear, função de ponderação de Gauss, distribuição uniforme e $\alpha = 2$.

Os resultados mostram boa aproximação com a solução analítica. Mais ainda: uma característica do Método de Galerkin sem Malha é a alta taxa de convergência que se observa em estudos de análise de erro (Belytschko *et al.*, 1994).

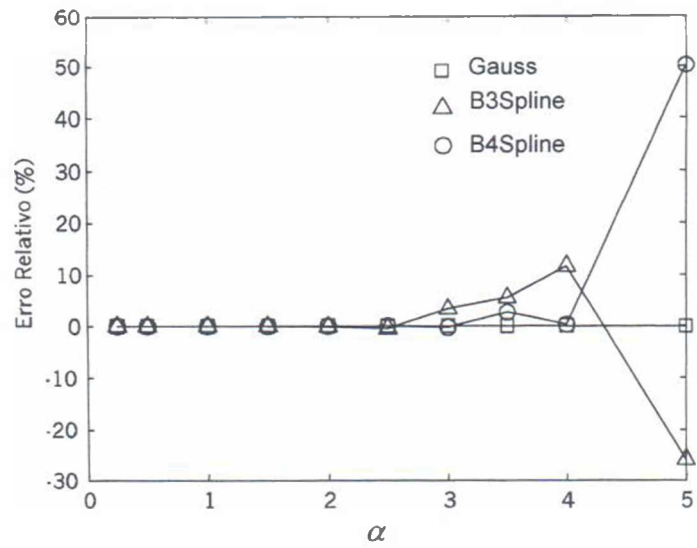


Figura 5.13 – Erro relativo versus α em função da função de ponderação.

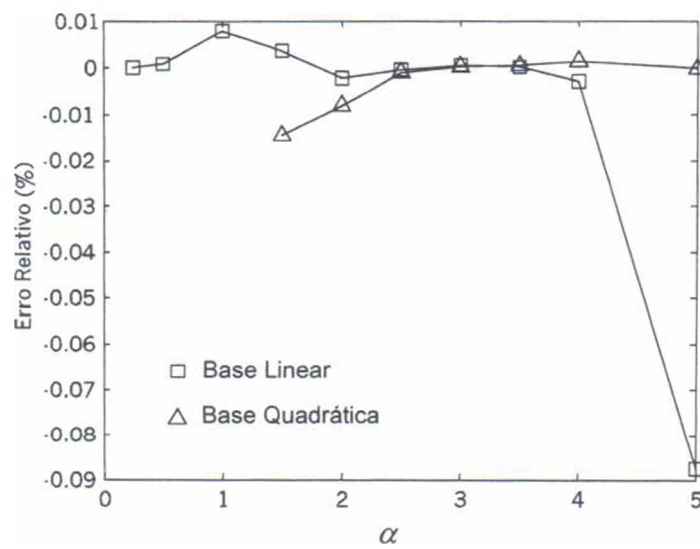


Figura 5.14 – Erro relativo em função da ordem da base.

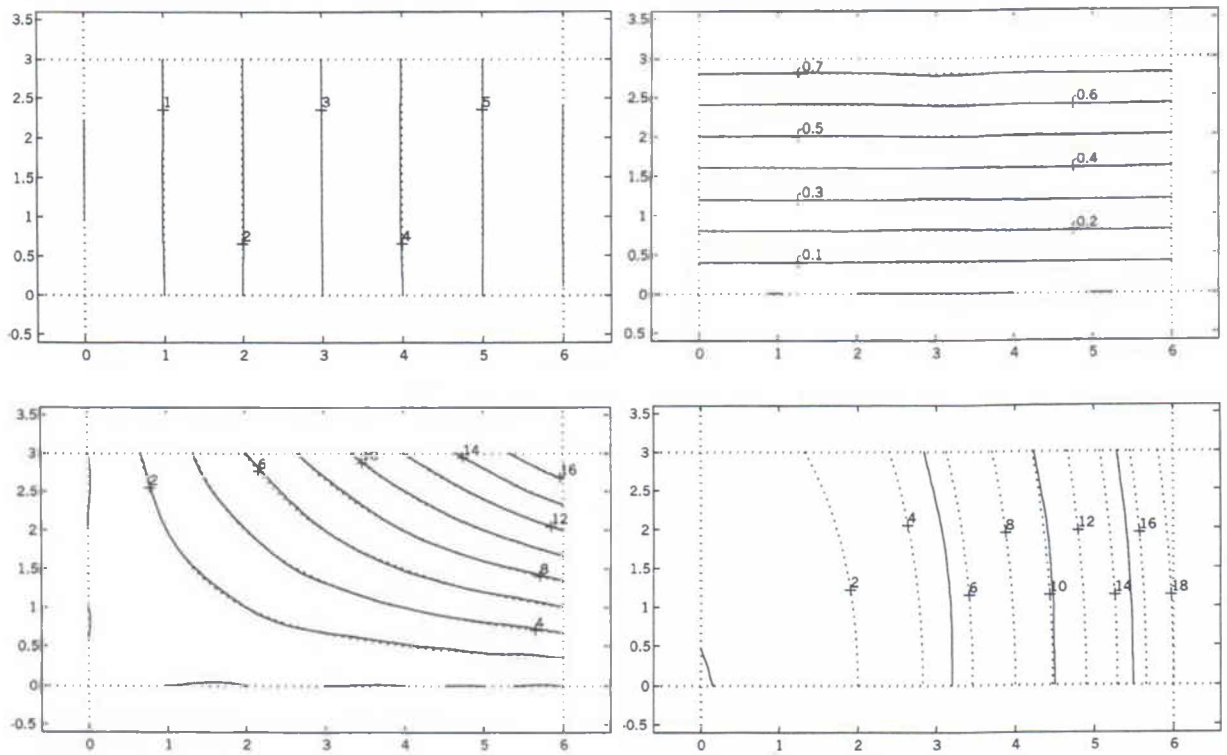


Figura 5.15 – Solução aproximada e exata da chapa submetida à tração axial.

5.6 - MODELAGEM DAS CLASSES DE UM PROBLEMA

No Capítulo 3 foram modeladas as classes para descrição das funções de forma. No Capítulo 4 as classes necessárias para a definição de um método, sua configuração e execução foram detalhadas. Neste capítulo, a propósito da apresentação do problema de elasticidade bidimensional, será empreendida a modelagem das classes que permitem a consideração das equações diferenciais que regem um problema de valor de contorno.

Como já discorrido no Capítulo 2, Seção 2.4, a consideração de um determinado tipo de problema é feita principalmente quando se consideram as matrizes locais com a aproximação por um número limitado de pontos nodais dos operadores diferenciais que regem um problema. Esta consideração é obviamente aderente ao tipo de método utilizado para a montagem das matrizes discretas globais, ou seja, se o método utiliza o critério de colocação ou de Galerkin, por exemplo.

Como a expansibilidade do sistema computacional é princípio basilar, a arquitetura a seguir busca assegurar que a implementação de novos problemas, equações diferenciais, ou método de solução sejam estanques em relação ao sistema já existente. Mais ainda, busca-se a possibilidade de alteração do método durante o tempo de execução o que facilitaria a implementação de interfaces gráficas.

Inicialmente a Figura 5.16, apresenta a interface **Problem** que permite a definição de um problema. Na verdade, parte da definição, uma vez que a caracterização de um problema necessita o estabelecimento de uma geometria que será atribuição da classe **Domain**, no Capítulo 6. Como ilustração seja a equação diferencial de segunda ordem

$$\begin{aligned} a(x) \frac{d^2 u}{dx^2} + b(x) \frac{du}{dx} + c(x) u &= f(x) \\ u(x) &= g(x) \quad \text{em } x = 0 \\ \frac{du}{dx} &= h(x) \quad \text{em } x = L \end{aligned}$$

onde: $f(x)$ - fonte

$a(x), b(x), c(x)$ - constantes

$g(x), h(x)$ - funções no contorno

Portanto para definição do problema linear acima, será necessário o estabelecimento de uma classe fonte (**Source**), de uma classe para definição dos coeficientes (**Coefficient**) e de outra para caracterização dos valores e tipos de contorno (**Edge**). A estrutura de dados que implementa a classe **Coefficient** permite sua utilização para a definição de outros valores, como por exemplo o módulo de elasticidade e o coeficiente de Poisson no problema de elasticidade bidimensional. Neste caso os coeficientes que multiplicam os termos das equações diferenciais no formato de tensões são unitários, como mostrado na Seção 5.2

A interface **Problem** requer ainda a definição de um método, através da operação **setMethod()** e a definição de um objeto denominado **ProblemMethod** cuja representação se dará adiante. Por fim, a interface requer a implementação de métodos de acesso a matrizes e vetores locais que permitirão a montagem do sistema global de equações, tais como **Kp()**, para a matriz de “rigidez” local, **Fp()**, vetor com a discretização da fonte, e assim por diante.

Como referido acima o acoplamento entre um problema e um método é realizado através da interface **ProblemMethod**, como mostrado na Figura 5.17. Esta interface emula os pedidos das matrizes locais direcionados à interface **Problem**. Assim os pedidos a **Problem** são redirecionados ao objeto da interface **ProblemMethod**, este sim responsável por atender à solicitação. Visto de outro modo, isto significa que um mesmo objeto **Problem** responderá de forma diferente dependendo do estado definido por um determinado método. Esse comportamento variável com um determinado estado configura o padrão de projeto orientado a objetos denominado *State* (Gamma *et al.*, 2000). Sua arquitetura é bastante similar ao padrão *Strategy*, que visava à alteração do algoritmo e não do estado como agora.

A escolha do objeto específico que acopla um método a um problema é realizada através da classe **ProblemMethodFactory**, que implementa a construção de um objeto através de uma operação chamada de fábrica. Este é um padrão de projeto bastante utilizado na construção de estruturas computacionais (*frameworks*), denominado de

FactoryMethod (Jandl, 2003). Como apresenta o comentário na Figura 5.17, junto à classe **ProblemMethodFactory**, trata-se simplesmente de uma seqüência de decisões que são tomadas dependendo do tipo de problema e de método que estão correntemente em uso. Esta construção permite isolar do problema o método de discretização empregado na construção da aproximação local.

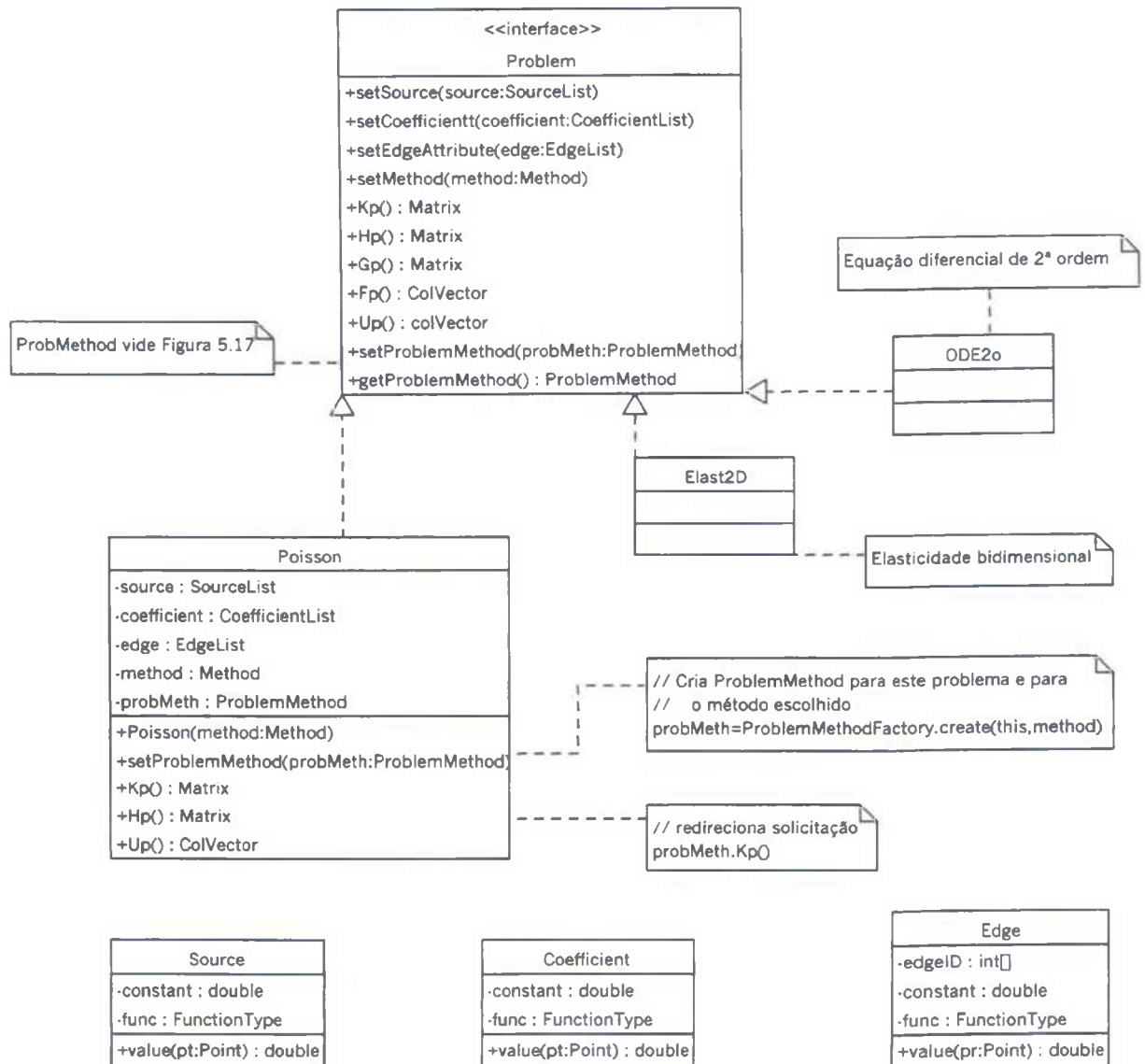


Figura 5.16 – Classes para modelagem de um problema.

Com esta arquitetura a implementação de um novo método de discretização para um problema requer uma intervenção bastante reduzida. Primeiro a construção da

classe que implementa a interface **ProblemMethod** e segundo a introdução de uma condição no método fábrica **create()** da classe **ProblemMethodFactory**.

Como estas classes são demandadas pelos algoritmos da classe MWR (vide Capítulo 5), deverá ocorrer uma consistência entre ambas. No caso da subclasse Galerkin o acesso às matrizes locais é feito diretamente através da classe **Problem** instanciada. Algumas vezes um método pode apresentar uma matriz local particular cujo acesso não está previsto na interface **Problem**. Neste caso o acesso se dará através da chamada de **getProblemMethod()**. Se for o caso poderá ser criada uma subclasse de **MWR** específica. Isto ocorreu no Método das Diferenças Finitas Generalizadas que monta a matriz **H** a partir do domínio, resultando numa alteração do algoritmo de colocação refletida na criação da classe **CollocationGFD**.

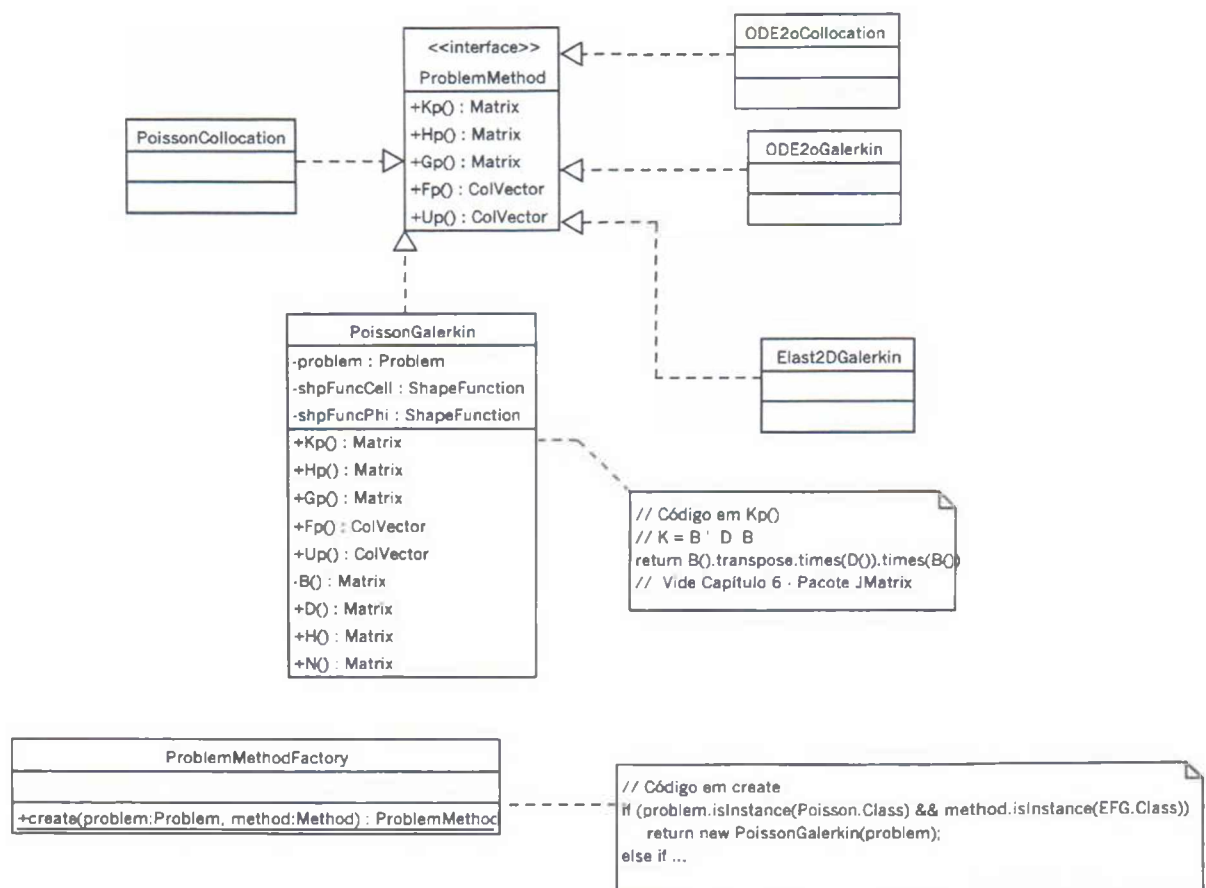


Figura 5.17– Classes para vinculação de um problema a um método.

5.7 - CONSIDERAÇÕES FINAIS

A apresentação da aproximação discreta do Método de Galerkin sem Malha aplicado a elasticidade linear seguiu, como de resto em todo este trabalho, formulação do Método dos Resíduos Ponderados. Esta abordagem, mais à álgebra linear, é bastante direta sem a necessidade de se partir de um funcional, nem sempre de existência e compreensão simples, e sem a utilização do cálculo variacional. A análise de erro, tanto no âmbito do Método dos Elementos Finitos como nos Métodos sem Malha, segue pelo caminho aqui sugerido, percorrendo os espaços normados da álgebra, como base de sua estratégia de avaliação.

Dentro desse enfoque matemático é que se insere a apresentação da dedução do Método dos Elementos Finitos. Como visto somente a adoção de funções descontínuas, de classe C^{-1} , permitem chegar a formulação tradicional, mostrando a precariedade imposta à aproximação no contorno Γ_u . Esse formato não é visto na literatura mas permite perceber com clareza a natureza do erro introduzido na dedução do método. No entanto a simplicidade obtida no sistema de equações e o relativamente baixo valor do erro, na maioria das vezes, justificam a tradição e a amplitude de seu uso na engenharia.

Como mostra a modelagem computacional a extensão do problema de elasticidade aos outros métodos se dá facilmente através da inclusão de classes que façam a ponte entre o problema de elasticidade e os respectivos métodos.

A estrutura computacional (*framework*) está agora preparada para enfrentar grande gama de problemas que devem ser devidamente vinculados a métodos oportunamente escolhidos. Na introdução deste capítulo foram relacionadas diversas áreas de aplicação do Método de Galerkin sem Malha que podem ser implementadas tendo esta estrutura computacional como embrião.

6 - UMA ESTRUTURA COMPUTACIONAL PARA OS MÉTODOS SEM MALHA

6.1 - INTRODUÇÃO

Nos capítulos precedentes empreendeu-se a análise das diferentes classes para modelagem computacional dos Métodos sem Malha. Neste capítulo são apresentados os passos que completam o mapa geral do Modelo Computacional ilustrado na Figura 2.13 do Capítulo 2. Nas seções seguintes constam as classes para a definição geométrica de um domínio (**Domain**), para a discretização em pontos nodais (**Discret**) da representação geométrica do domínio, para a análise do problema (**Analysis**), e para a criação de objetos matrizes (**JMatrix**).

A concepção computacional utilizada visa a criação de classes que possam ser expandidas e especializadas por novas classes deixando intacto o núcleo central previamente programado. Para isto utilizou-se sobremaneira o conceito de interface presente na análise orientada a objetos e como metodologia buscou-se o emprego sistemático de padrões de projeto como forma de melhor conceber e documentar a solução de programação apresentada.

Nortear a modelagem a construção de um arcabouço de classes (*framework*) que de forma consistente apresentem uma solução para um domínio específico de aplicação, no caso os Métodos sem Malha, e que de forma flexível possa vir a incorporar novos problemas, métodos de discretização e procedimentos de solução. Ao final do capítulo é apresentado em resumo o sistema computacional denominado de JFraMMe (*Java Framework for Meshless Methods*).

6.2 - MODELAGEM DO DOMÍNIO E DA DISCRETIZAÇÃO

As classes para modelagem geométrica do domínio (**Domain**) e para construção da discretização em pontos nodais possuem grande interação. A separação em classes distintas visa sobretudo possibilitar a interface futura com classes que procedam à descrição e representação num ambiente CAD (*Computer Aided Design*).

A descrição geométrica de um domínio é realizada através da criação de uma lista de pontos (**PointList**), uma lista de contornos (**EdgeList**) e de uma lista de caminhos, ou caminhamentos, denominada de **PathList**. Por simplicidade adotou-se que o primeiro caminho é externo com sentido anti-horário e possibilitando quantos caminhamentos internos de sentido horário que se deseje. Com isto pode-se representar um domínio com furos. A Figura 6.1 ilustra o resultado da imagem desta estrutura de dados.

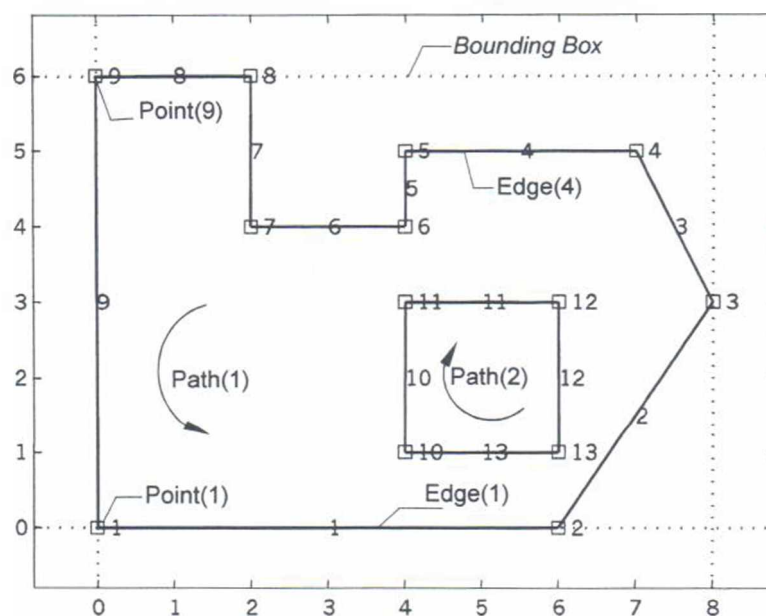


Figura 6.1 – Representação da geometria de um domínio através de pontos, contornos e caminhamentos.

A utilização de lista encadeada de pontos é suficiente para a construção do modelo pretendido. Outras estruturas mais adequadas ao CAD, tais como lista duplamente encadeada (Preparata e Shamos, 1985; de Berg *et al.*, 2000), podem ser implementadas e desde que as classes que as implementem sigam a interface **Domain**, os objetos

decorrentes poderão ser usados no restante do sistema computacional deste trabalho. Neste caso essas novas classes se comportarão como adaptadoras (padrão de projeto *Adapter*). A Figura 6.2 ilustra duas classes concretas que utilizam a interface **Domain**. A classe **DomainGeometry** implementa a estrutura em lista encadeada e a classe **DomainCAD**, não implementada, é um exemplo de possibilidade para uma estrutura de dados mais adequada à computação gráfica interativa. O método de classe **plotDomain()** induz a que todas as implementações possuam alguma forma de estabelecer uma representação gráfica do domínio.

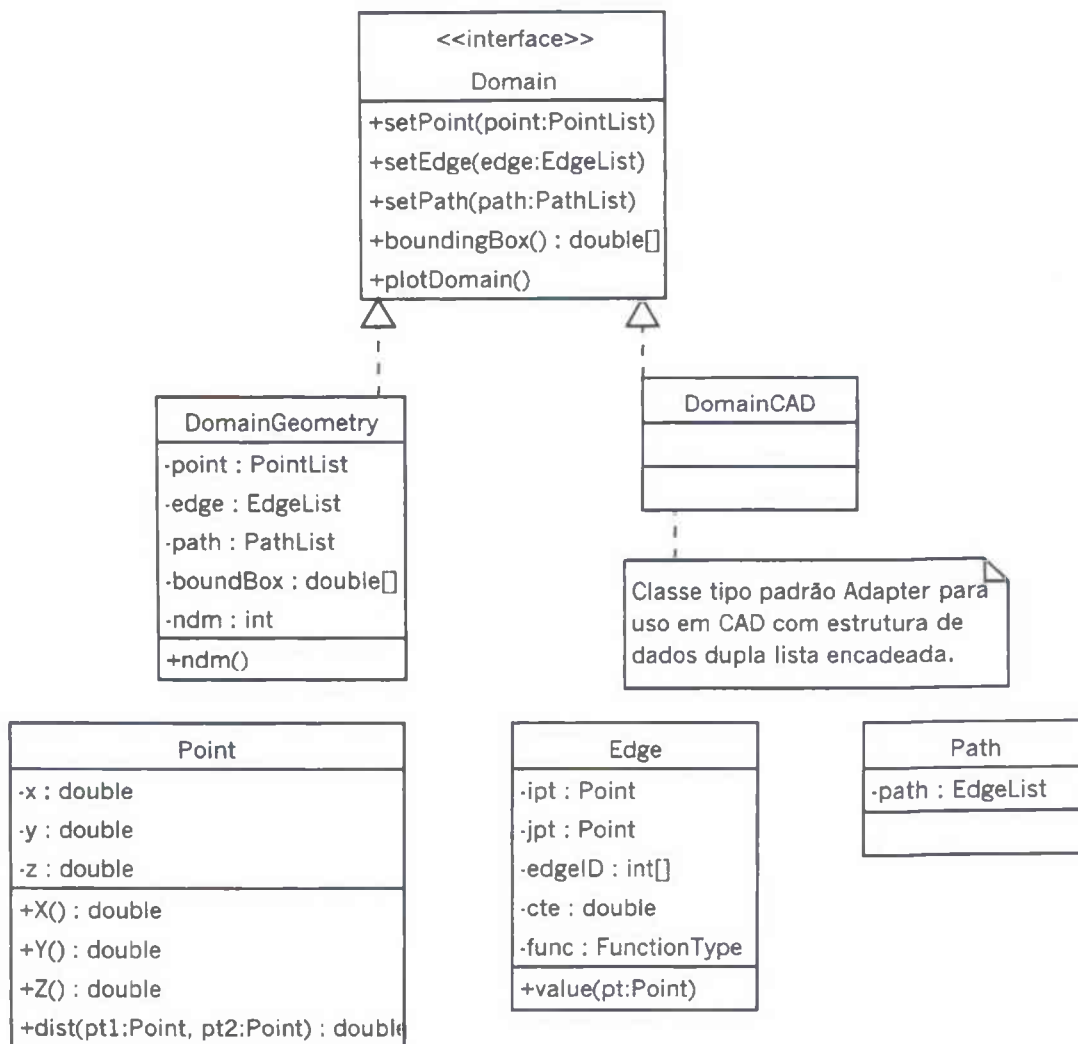


Figura 6.2 – Classes para modelagem geométrica de um domínio.

Na Figura 6.2 constam ainda as classes que em composição constroem o objeto **Domain**. A classe **Point** baseada na implementação de Sunday (2003), permite a

modelação de pontos em qualquer dimensão. A classe **Edge** é um refatoramento da classe apresentada na Figura 5.16, com a inclusão do atributo representando os pontos inicial e final de um segmento de contorno. A classe **Path** modela um caminharmento, representando uma lista de contornos.

A classe **Discret** conforme a Figura 6.3 possibilita a discretização do domínio, do contorno e das células de integração. Optou-se por separar a discretização do contorno e do domínio em duas classes distintas de modo a se garantir uma maior precisão na definição do modelo discreto. Assim a classe **Discret** se incumbe de criar as três discretizações e provê os meios de acesso aos componentes dessas discretizações.

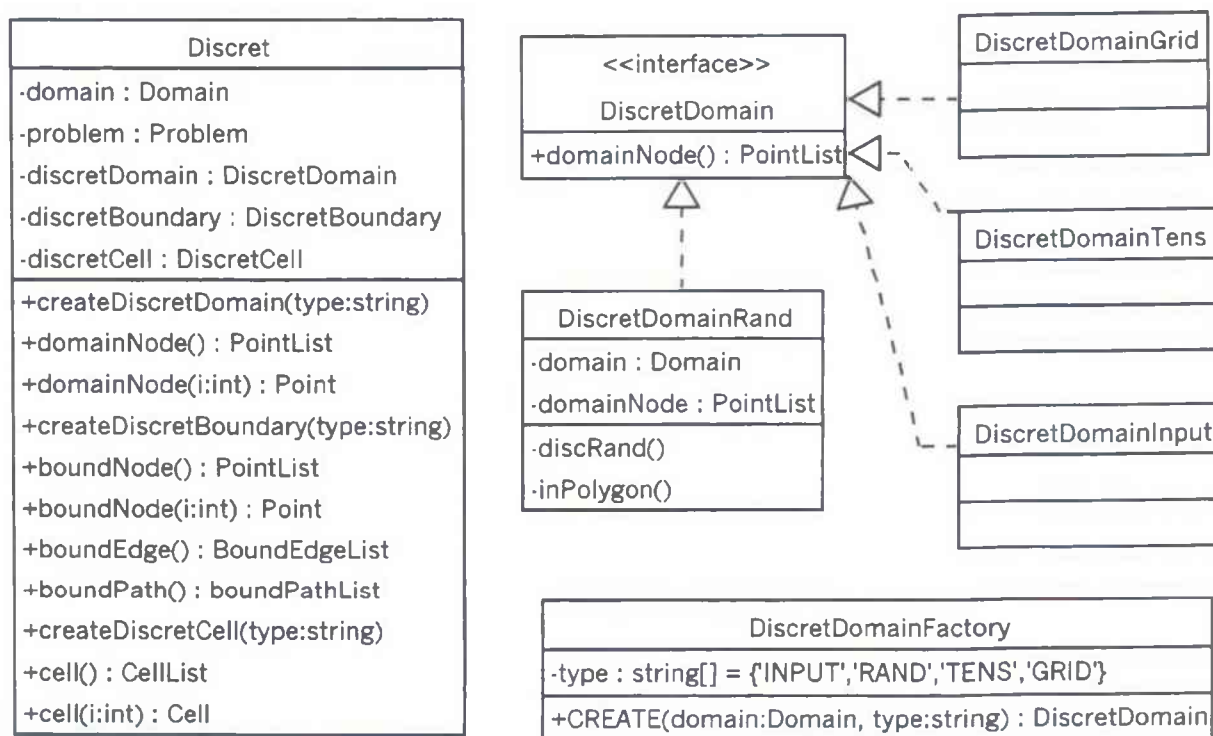


Figura 6.3 – Classe para discretização em pontos nodais da geometria de um domínio.

A interface **DiscretDomain** permite a criação de diversos procedimentos de discretização do domínio. A Figura 6.4 ilustra quatro possibilidades: geração randômica, em grelha, por produto tensorial e imposição de pontos diretamente. Cabe ressaltar que através de algoritmo de geometria computacional (Harrington, 1987; Preparata e Shamos, 1985). evita-se a inclusão de pontos fora do domínio. Outros procedimentos podem ser

acrescentados ao sistema seguindo a interface **DiscretDomain** e alterando-se o método fábrica **DiscretDomainFactory**.

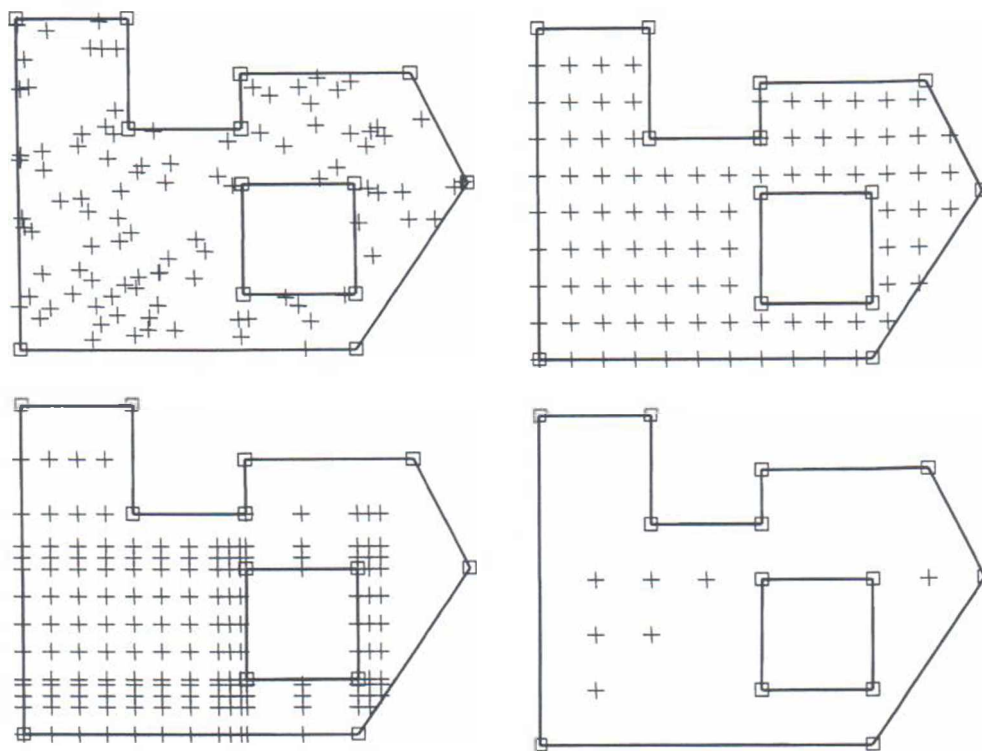


Figura 6.4 – Procedimentos de discretização do domínio em pontos nodais.

A classe **DiscretBoundary** Figura 6.5 é a responsável pela discretização do contorno. Na verdade no processo de discretização a estrutura de dados do contorno do domínio geométrico é substituída por outra incorporando os dados do problema nesses contornos. Assim, os pontos nodais do contorno discretizado possuem agora informação sobre o tipo de contorno, o co-seno diretor da normal externa, valor da condição de contorno para todos os graus de liberdade, além das coordenadas cartesianas do ponto.

Como esse ponto nodal de contorno possui atributos que permite diferenciar dois pontos geométricos, fica estabelecida a estrutura de dados que leva em conta a existência de pontos gêmeos. Caso um método de discretização não permita a consideração desses pontos, na verdade dos métodos implementados no Capítulo 4, somente o Método das Diferenças Finitas Generalizadas permite pontos gêmeos, então se deve empregar a

operação **discardTwinNode()**. O descarte do ponto gêmeo segue sugestão de Reddy (1993) que estabelece o predomínio do contorno Γ_u (essencial) e do maior valor absoluto.

Uma possibilidade que pode comprometer todo o processo de solução da discretização é a ocorrência de pontos nodais clones, que acarretaria singularidade no sistema de equações. O método de classe **checkCloneNode()**, elimina os pontos clones que porventura uma entrada de dados inapropriada tenha produzido.

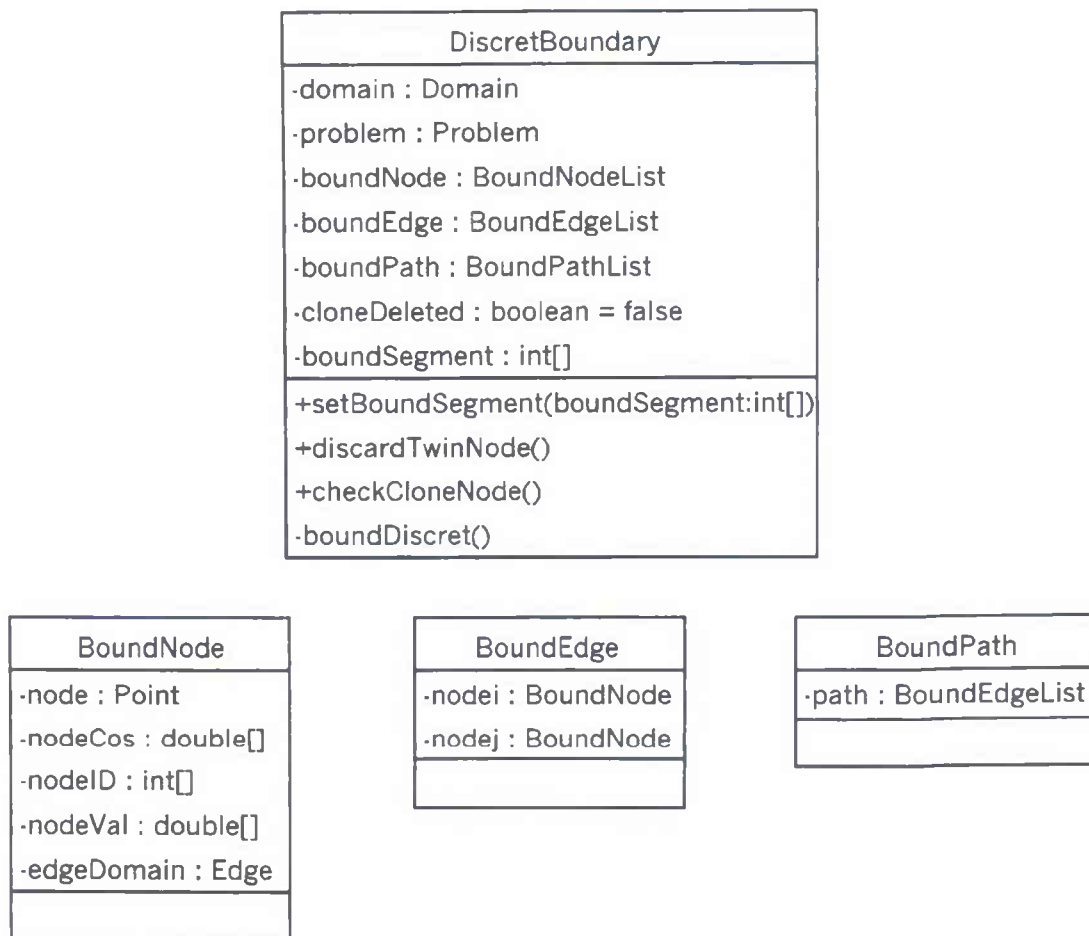


Figura 6.5 – Classe para discretização do contorno.

A Figura 6.6 apresenta a saída que a classe **DiscretBoundary** produz com o propósito de permitir a verificação dos dados discretizados no contorno. Para geração dos pontos nodais de contorno a única entrada de dados necessária é a informação da quantidade de segmentos que se deseja em cada face do contorno do domínio geométrico original.

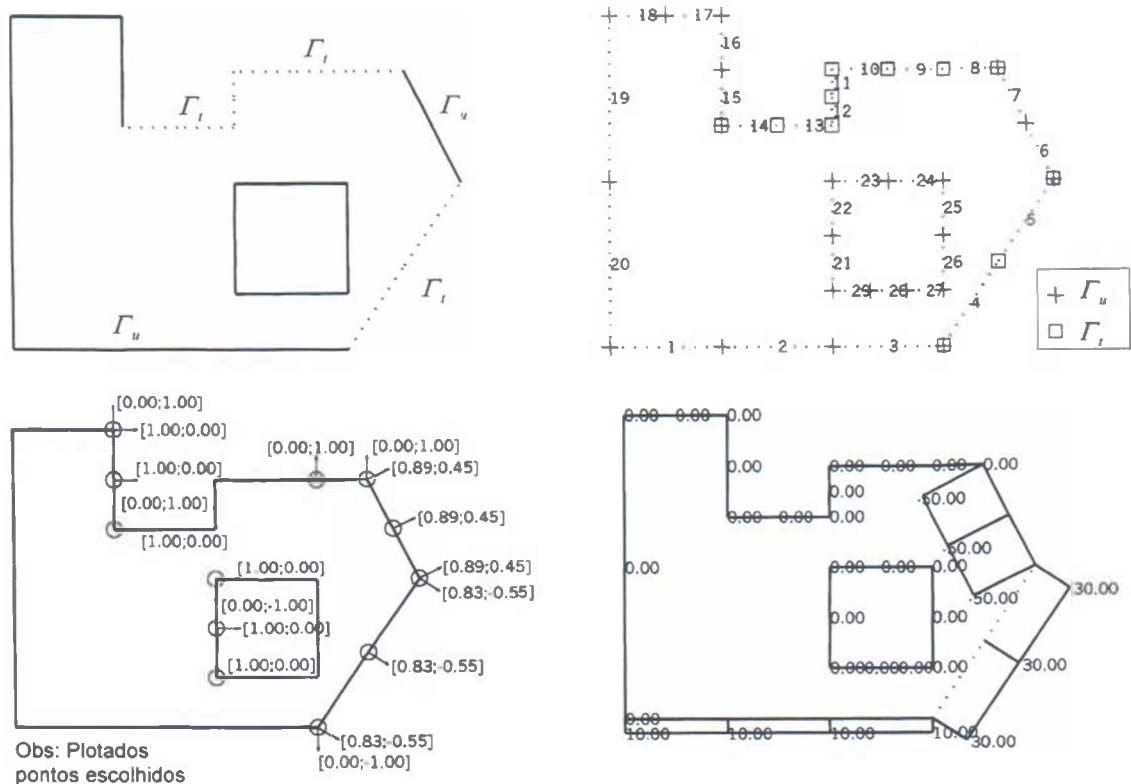


Figura 6.6 - Representação da discretização do contorno com propriedades nodais.

A discretização das células para integração numérica ou geração do Método dos Elementos Finitos é realizada com classes que implementam a interface **DiscretCell**, Figura 6.7. Como os pontos nodais para definição das células são totalmente independentes dos pontos gerados na discretização do domínio e do contorno, foi estabelecida uma classe específica para os pontos de definição das células. Para consistência no caso do Método dos Elementos Finitos cada ponto de célula possui atributo para identificar o tipo de ponto nodal. Ou seja, se pertence ao domínio, ou a um dos contornos essencial ou natural.

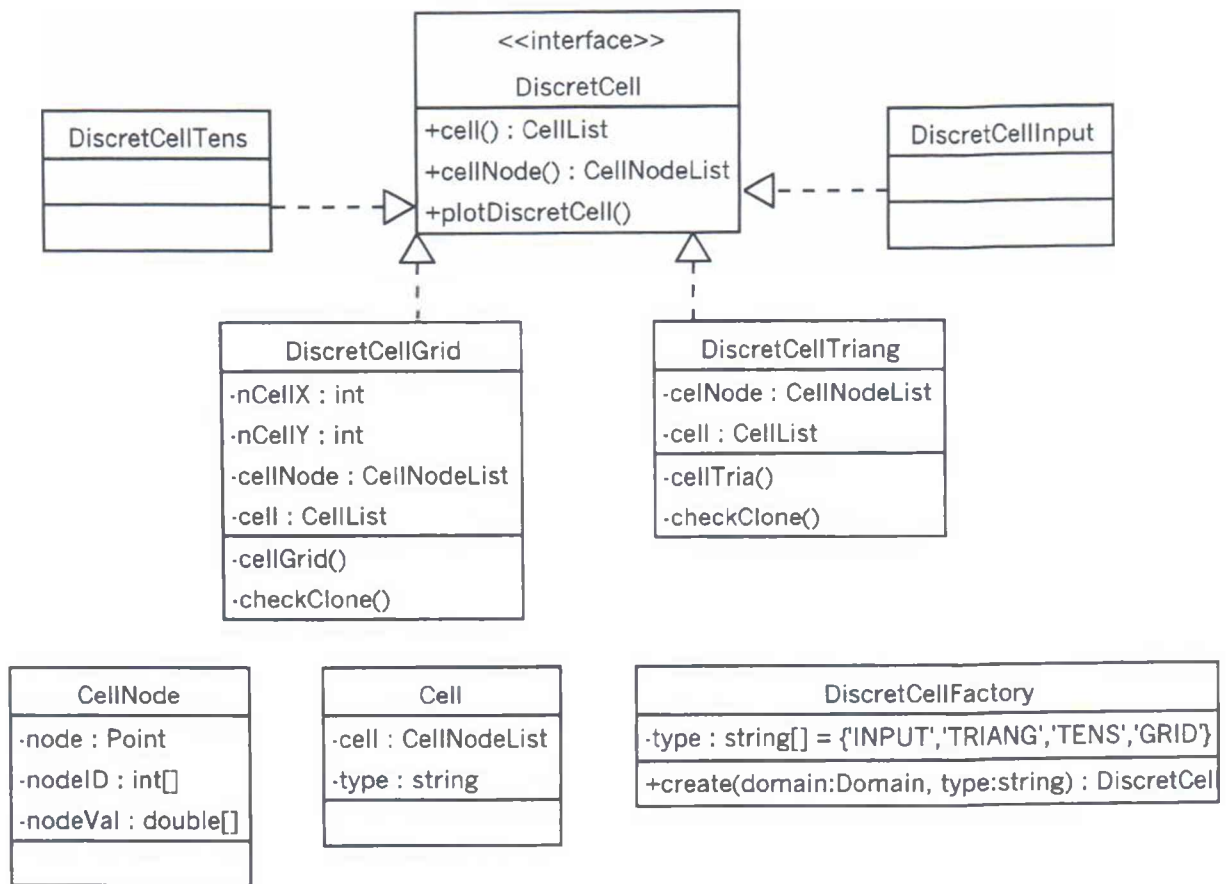


Figura 6.7 – Classes para discretização das células de integração e elementos finitos.

A Figura 6.8 apresenta algumas possibilidades de geração de células: grelha, produto tensorial e triangularização. Cabe destacar a geração de células com o emprego do algoritmo de triangularização restringida seguindo os passos apresentados no algoritmo da Tabela 4.2.

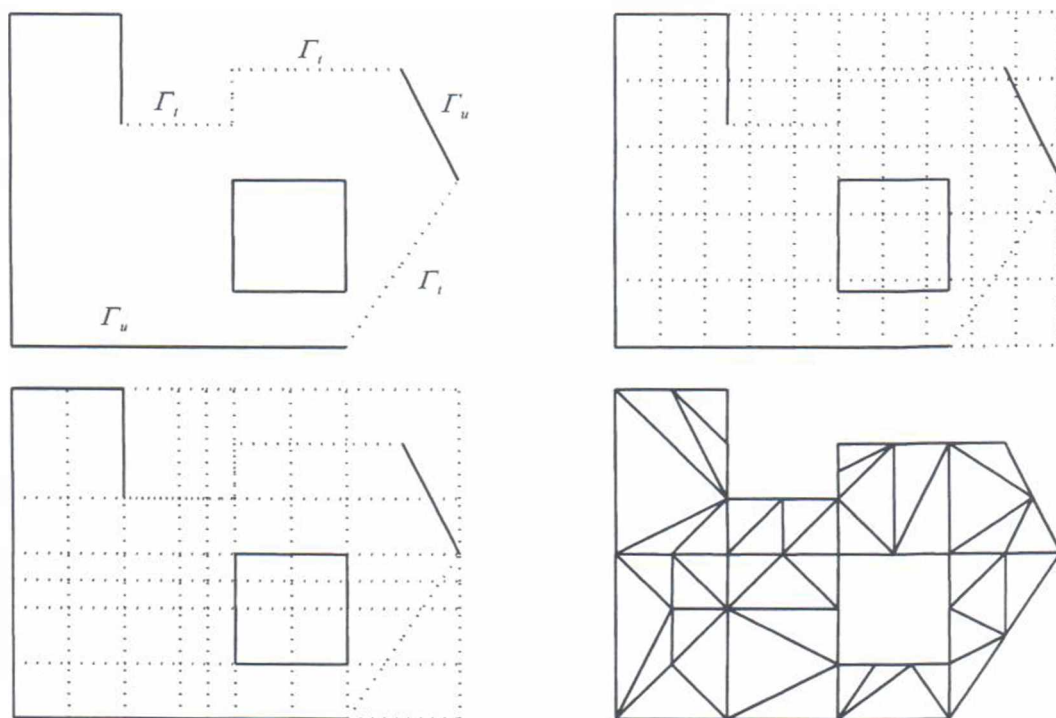


Figura 6.8 – Algoritmos para geração de células de integração.

6.3 - CLASSE **ANALYSIS** BASEADA EM ALGORITMOS

A classe **Analysis** tem como atribuição a formação e resolução do sistema de equações algébricas discretas que substituem as equações diferenciais que regem o problema. O enfoque típico adotado em diversos trabalhos (Forde *et al.*, 1990; Zimmermann *et al.*, 1992; Pidaparti e Hudl, 1993; Archer, 1996; McKenna, 1997), é a abordagem tradicional de caixa-preta presente na programação do Método dos Elementos Finitos. O algoritmo de solução seja de uma análise estática ou dinâmica é executado com pouco controle externo sobre as classes e etapas que o compõem.

Os modelos de classe **Analysis** vistos na literatura citada se baseiam numa hierarquia de classes que encapsulam um determinado algoritmo de classe específica. Dada a grande diversidade de tipos de análises as árvores da hierarquia de classes possuem uma grande quantidade de ramificações levando a uma dificuldade de expansão e compreensão desses modelos.

Zienkiewicz e Taylor (1989 e 1994) apresentam abordagem de solução dos algoritmos através de uma linguagem de macro-programação ou macro comandos. Esse enfoque, embora implementado numa linguagem procedural, revela grande flexibilidade e modularidade, responsáveis pela sua longevidade na literatura. A primeira apresentação data de 1977.

A abordagem de macro-comandos permite realizar diversos tipos de análise utilizando-se um número relativamente restrito de instruções. Portanto, o objetivo aqui deixa de ser como encapsular um algoritmo por classe mas sim como encapsular numa classe as ações necessárias a diferentes tipos de algoritmos.

Como habitual, de modo a permitir expansibilidade e flexibilidade, definiu-se a interface **Analysis** e uma classe abstrata container de dados e operações comuns, denominada de **AnalysisOperations**. A Figura 6.9 apresenta esses componentes além de duas classes concretas que implementam a interface **Analysis**.

A classe FEAP é baseada nas instruções de macro-comandos de Zienkiewicz e Taylor e contempla diversos algoritmos como análise linear estática, dinâmica e não-linear incremental. As instruções existentes na implementação original para execução de laços de

instrução e desvios não são necessárias, uma vez que estes comandos devem ser realizados com a própria linguagem de programação utilizada, Java.

A classe **JFraMMe**, na Figura 6.9, representa o estado implantado neste trabalho contemplando a análise linear estática.

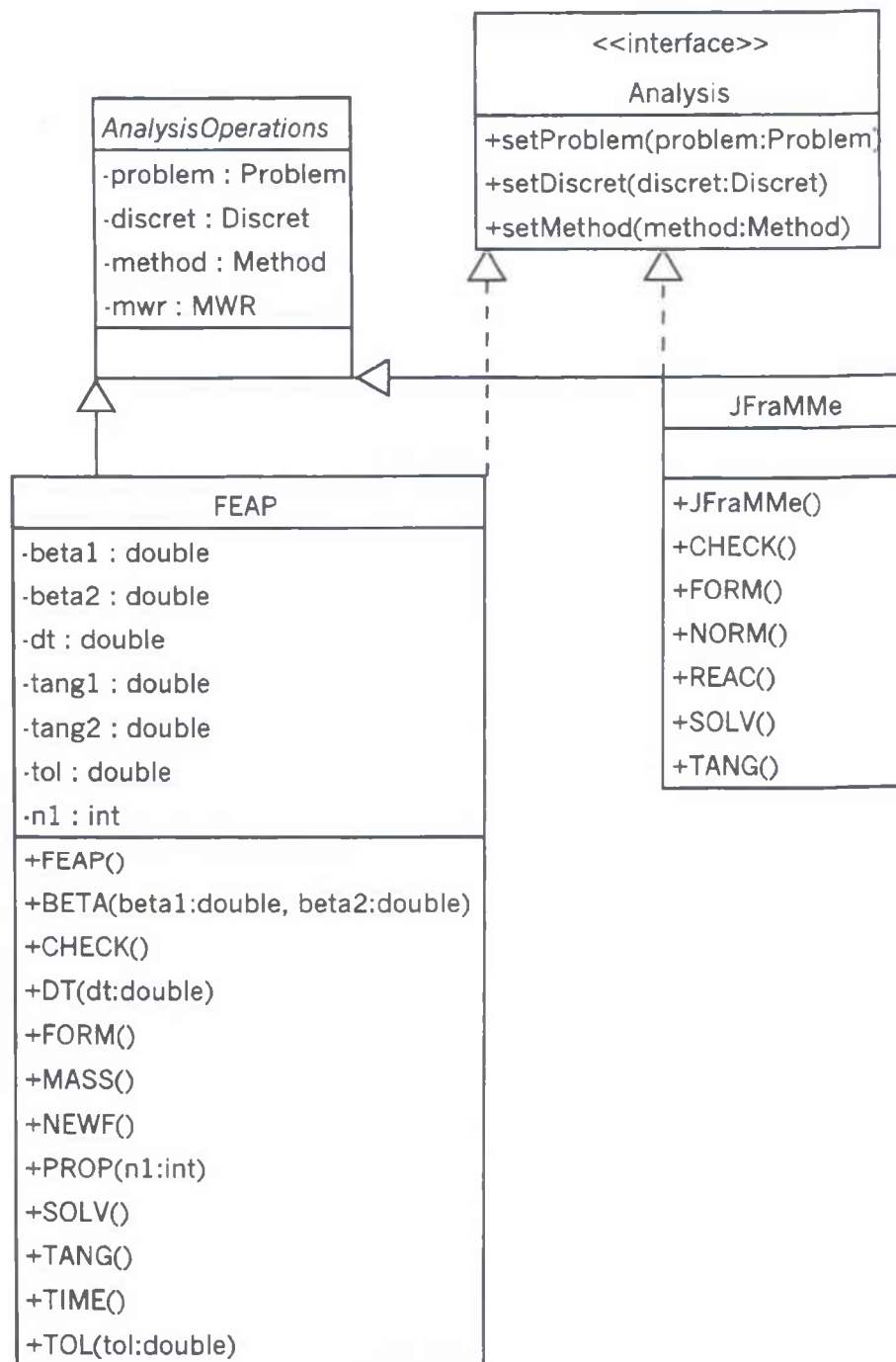


Figura 6.9 – Interface para classe de análise do problema.

A implementação dos algoritmos como apresentado na Figura 6.9 requer a modificação dos componentes da interface **Problem**, com a inclusão da matriz de massa local e para determinação de resíduos, por exemplo. Bem como um fracionamento das operações existentes na classe MWR para montagem das matrizes globais. Essas intervenções se acomodam com facilidade na arquitetura proposta. O formato obtido para a classe **Analysis** permite grande controle e versatilidade do processo de análise de um problema, conforme testemunha o próprio enfoque original de análise através de macro-instruções.

6.4 - CLASSES MATRICIAIS – O PACOTE JMATRIX.

Os objetos matriciais são os tipos primitivos mais utilizados na implementação de procedimentos numéricos em mecânica computacional. A utilização de matrizes e da álgebra matricial na apresentação de teorias e algoritmos é consagrada e habitual notadamente na engenharia. Como assinalado por Besset (2001), o paradigma da orientação a objetos possibilita a manipulação de conceitos matemáticos de tal maneira que os limites entre o objeto real e de seu modelo computacional sejam quase inexistentes, facilitando a conversão da linguagem matemática, empregada na exposição teórica, para a linguagem computacional.

A abordagem computacional da álgebra matricial deve considerar os seguintes aspectos: a definição matemática dos principais tipos de matrizes com suas operações básicas; formas de armazenamento das matrizes (completo, banda, perfil e esparsa, por exemplo); e algoritmos de solução de sistemas de equações e problemas de autovalores. Os modelos de classes reportados na literatura atendem de forma parcial aos aspectos citados, muitas vezes função de problemas específicos tratados (Forde *et al.*, 1990; Scholz, 1992; Zeglinski *et al.*, 1994; Lu *et al.*, 1995; Devloo, 1997; Jeremič e Sture, 1992; McKenna, 1997; Java Matrix Package, 1999; Besset, 2001).

A Figura 6.10 apresenta o diagrama das classes que compõem o pacote JMatrix (Sánchez e Pulino, 2003). A construção do pacote buscou como requisito básico uma especificação de interface simples e mínima. A interface **MatrixType** é a chave para definição de um *tipo abstrato de dados* (Aho *et al.*, 1987) que modele objetos matrizes. Deve-se observar dentre as operações de **MatrixType** a denominada de **toMatrix()**, que deve ser implementada em cada tipo de matriz de modo a permitir a coerção (transformação) entre os diversos tipos de matrizes. Com isto não se viola uma característica de algumas linguagens orientadas a objetos que se denomina de *tipagem forte* (Goodrich e Tamassia, 2002). As operações de interesse comum ficam reunidas na classe abstrata **MatrixOperations**.

No mesmo sentido de preservar a tipagem forte, ou seja, de restringir a mistura de tipos, foram criadas as classes **ColVector** e **RowVector**, de modo a modelar com precisão o objeto matemático, seguindo a solução de Lu *et al.* (1995). Assim, operações

inapropriadas entre os diferentes tipos, são verificadas ainda em tempo de compilação, reduzindo a possibilidade de temíveis erros de execução.

De modo que a sintaxe das operações de matrizes com escalares fique o mais próximo possível da sintaxe matemática foi criada a classe **Scalar**, Figura 6.10, que na realidade é apenas uma classe adaptadora (seguindo o padrão de projeto *Adapter*) que envolve um tipo numérico primitivo da linguagem em uso, por exemplo um *double*. Assim as seguintes sintaxes são compatíveis

```
scalar.times(matrix);
```

```
matrix.times(double);
```

```
matrix.times(scalar).
```

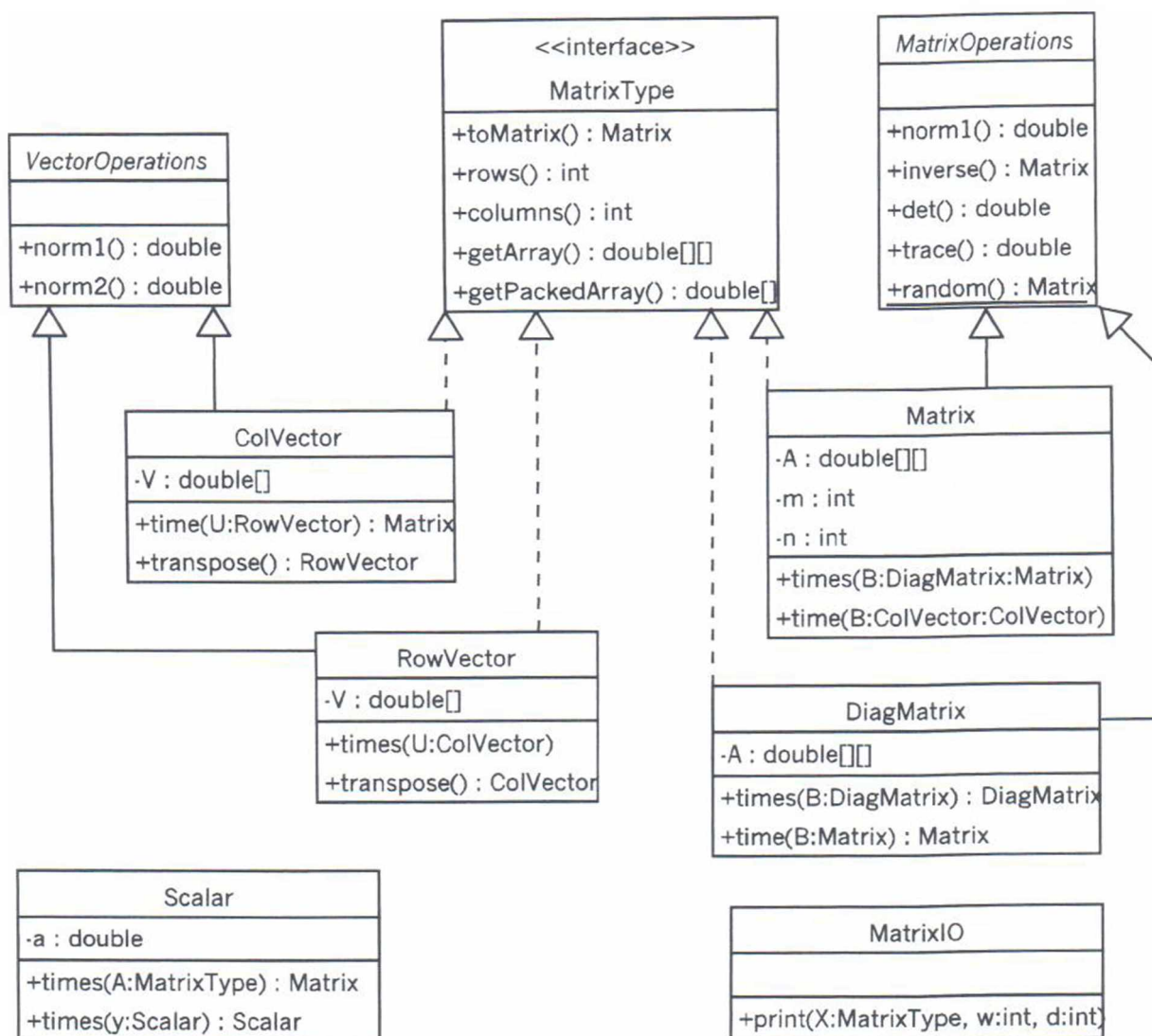


Figura 6.10 – Projeto de classes do pacote JMatrix.

O tratamento de sistemas de equações lineares poderia ser considerado diretamente dentro da classe **Matrix**, como realizado por diversas implementações citadas. Optou-se como usual na matemática por uma separação destes conceitos numa classe denominada de **LinearEquations** na direção dos trabalhos de McKenna (1997) e Besset (2001). Essa separação permite uma maior clareza, no tratamento computacional, do objeto matemático sistema linear de equações.

O modelo computacional da Figura 6.11 utiliza o padrão de projeto **Strategy** para permitir que o tipo de solver, ou seja da estratégia de solução do sistema possa ser alterada com facilidade e novos algoritmos possam ser acrescentados sem que a classe de contexto **LinearEquations** seja alterada.

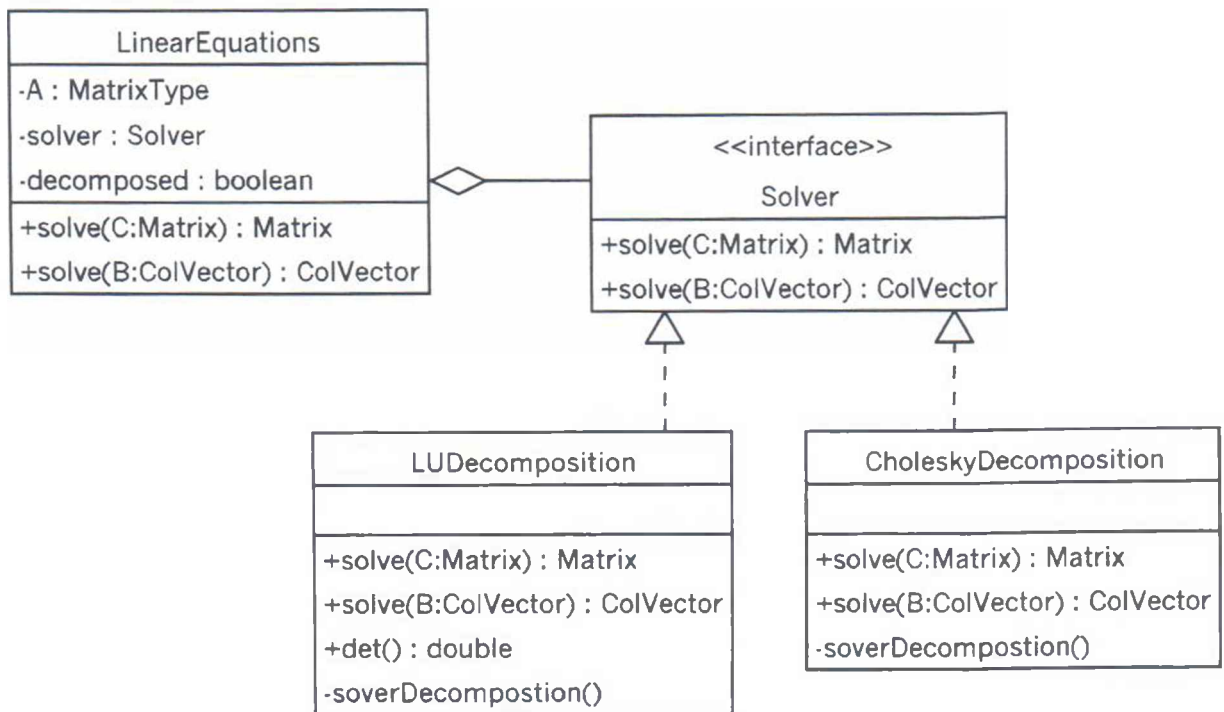


Figura 6.11 – Classe para modelagem do sistema linear de equações.

Um aspecto que será tratado em futuras versões do pacote JMatrix é o armazenamento de matrizes esparsas. Boa parte da arquitetura foi concebida tendo em vista esta possibilidade.

Nesta implementação fez-se uso dos algoritmos do pacote JAMA – *Java Matrix Package* (1999), tendo em vista a sua qualidade e seu caráter de domínio público. O pacote JAMA foi desenvolvido com vistas à futura introdução de um padrão de classes matriciais na linguagem Java.

6.5 - JFraMMe: UMA ESTRUTURA COMPUTACIONAL EM JAVA PARA OS MÉTODOS SEM MALHA

A análise computacional empreendida em seção específica de cada capítulo precedente e completada neste capítulo teve como mote a construção de uma estrutura computacional que facilite a implementação de aplicativos para a resolução de problemas da mecânica computacional através dos Métodos sem Malha.

Uma estrutura computacional (*framework*) pode ser definida como um conjunto de classes que cooperam entre si e que constroem um projeto reutilizável para um tipo específico de programa de computador (Gamma *et al.*, 2000). Nesse sentido foi projetada a arquitetura apresentada. A flexibilidade de expansão proporcionada pelo desacoplamento através de interfaces e classes abstratas permite que a criação de uma aplicação específica seja feita através de classes e subclasses que se inserirão no contexto do projeto elaborado. Como consequência o desenvolvimento de novos aplicativos ou aplicações de novos métodos de discretização e problemas terão uma construção mais célere.

O emprego extensivo de padrões de projeto orientado a objetos propicia uma arquitetura adequada a diferentes aplicações sem necessidade de reformulações do núcleo central da estrutura computacional (*framework*). Por outro lado, facilitam a documentação, uma vez que, seu uso intensivo é uma tendência na boa prática da análise orientada a objetos.

Os diagramas UML foram elaborados numa perspectiva dita de análise (Fowler e Scott, 2000), ou seja com poucos detalhes de implementação e sem referências a linguagens de programação, a não ser por alguns comentários. Convém frisar que a análise efetuada é independente da linguagem de programação.

O próximo passo na construção da estrutura computacional para os Métodos sem Malha é o detalhamento e a implementação em uma linguagem computacional específica. Pelas razões já apresentadas no capítulo de introdução, adotou-se a linguagem Java para implementação computacional, daí JFraMMe (*Java Framework for Meshless Methods*).

6.6 - CONSIDERAÇÕES FINAIS

Este capítulo encerra a apresentação da modelagem computacional desenvolvida neste trabalho. A análise orientada a objetos não conduz à solução única. Buscou-se através do estudo de abstrações de alto nível da teoria dos Métodos sem Malha aquelas soluções de programação mais próximas do modelo teórico.

Os diagramas UML, com a estrutura de interfaces e classes, foram apresentados numa perspectiva de análise, sem muitos detalhes e com simbologia resumida. Contudo, o desdobramento em diagramas de projeto, vinculados a uma linguagem de programação, é de decorrência natural.

A modelagem computacional do domínio e da sua discretização requer a incorporação de conceitos da Computação Gráfica e da Geometria Computacional. Nas classes propostas foram utilizados conceitos básicos dessas áreas suficientes para o objetivo aqui pretendido. O aprofundamento da aplicação dessas áreas na modelagem computacional dos Métodos sem Malha é fundamental para o desenvolvimento de programas em ambiente gráfico e para problemas em três dimensões.

O enfoque dado aos procedimentos de solução através de macro-comandos visa à futura incorporação com facilidade dos diferentes algoritmos para solução dos problemas da engenharia. A solução adotada é uma releitura orientada a objetos da concepção de Zienkiewicz e Taylor (1989).

O pacote matricial apresentado foi desenvolvido como sugestão de padrão para a linguagem Java. A incorporação de um ambiente de trabalho interativo resultaria em um programa para solução de problemas matriciais de baixo custo.

A estrutura computacional (*framework*) resultante da modelagem realizada encontra-se em uma fase inicial de desenvolvimento se adotado como novo objetivo o emprego por analistas modeladores de métodos de discretização. Questões como manual de usuário, ambiente gráfico para modelagem, ferramentas gráficas para entradas de dados e apresentação de resultados, são tarefas pendentes.

7 - MÉTODO SEM MALHA PETROV-GALERKIN COM APROXIMAÇÃO LOCAL EM SÉRIE DE TAYLOR

7.1 - INTRODUÇÃO

Um dos principais objetivos deste trabalho é a construção de um sistema computacional que se convertesse em verdadeira bancada de experimentação numérica, facilitando a comprovação computacional de desenvolvimentos teóricos realizados.

Neste capítulo desenvolve-se um Método sem Malha que utiliza o critério de Petrov-Galerkin na escolha das funções de ponderação no Método dos Resíduos Ponderados e para aproximação local emprega a expansão em série de Taylor à semelhança do Método das Diferenças Finitas Generalizadas. Como visto no Capítulo 4, essas funções de aproximação incorporam termo com a derivada direcional normal ao contorno, possibilitando maior consistência da modelação domínio-contorno.

Portanto pretende-se utilizar funções de aproximação baseada em série de Taylor, que possuem grande qualidade de aproximação em torno de um ponto. Essas funções, obtidas no âmbito do Método de Mínimos Quadrados Móvel, apresentam característica de interpolação seguindo o delta de Kronecker. E, no procedimento de integração da formulação fraca será utilizada função de ponderação distinta da função de aproximação (critério de Petrov-Galerkin).

A modelagem computacional será feita com a extensão da estrutura computacional (*framework*) desenvolvida criando-se as classes específicas para o método proposto. Basicamente somente devem ser implementadas classes para funções de forma, método e problema. Assim este capítulo é um exemplo de aplicação da estrutura computacional JFraMMe desenvolvida neste trabalho.

7.2 - MÉTODO SEM MALHA PETROV-GALERKIN TAYLOR

A apresentação utilizará o exposto no Capítulo 2 tomando como exemplo a equação de Poisson. Após aplicar procedimento de enfraquecimento, ou seja integrando-se por partes, chega-se ao seguinte sistema de equações

$$\begin{aligned} - \int_{\Omega} \left[\left(\frac{\partial w}{\partial x} \frac{\partial u}{\partial x} + \frac{\partial w}{\partial y} \frac{\partial u}{\partial y} \right) \right] d\Omega + \int_{\Gamma_n} w \frac{\partial u}{\partial n} d\Gamma &= \int_{\Omega} w b d\Omega - \int_{\Gamma_t} w t d\Gamma \\ \int_{\Gamma_n} \bar{w} u d\Gamma &= \int_{\Gamma_n} \bar{w} u_p d\Gamma \end{aligned} \quad (7.1)$$

onde, w é a função de ponderação, u é a variável primária e $\partial u / \partial n$ a variável secundária. Demais termos com significado detalhado no Capítulo 2.

Adotando a seguinte discretização para a função u

$$Du = \varphi a + \varphi_n \frac{\partial a}{\partial n} \quad (7.2)$$

ou como visto no Capítulo 4

$$Du = [A + A_n]^{-1} B a + [A + A_n]^{-1} B_n \frac{\partial a}{\partial n} \quad (7.3)$$

com significado já exposto.

O vetor Du coleta a função de forma e suas derivadas até a ordem estabelecida. As derivadas da incógnita principal u necessárias à expressão (7.1) são dadas por

$$\begin{aligned} u_{,x} &= \frac{\partial u}{\partial x} = \varphi_{,x} a + \varphi_{n,x} \frac{\partial a}{\partial n} \\ u_{,y} &= \frac{\partial u}{\partial y} = \varphi_{,y} a + \varphi_{n,y} \frac{\partial a}{\partial n} \end{aligned} \quad (7.4)$$

Logicamente se o conjunto de pontos utilizados (nuvem) não inclui pontos no contorno o segundo termo de (7.4) se anula. A derivada no contorno, $\partial u / \partial n$, é prescrita no

contorno natural, Γ_l , e incógnita no contorno essencial, Γ_u . Consideração idêntica já menciona no Método das Diferenças Finitas Generalizadas.

As funções de ponderação no âmbito do Método dos Resíduos Ponderados devem ser linearmente independentes e constituir uma seqüência completa, conforme assinalado no Capítulo 2. No critério de Galerkin adotam-se funções de ponderação idênticas às funções de aproximação local. No presente caso devido à consideração do grau de liberdade correspondente à derivada normal no contorno, a utilização do critério de Galerkin acarretaria um acoplamento entre as incógnitas primária e secundária, além de levar a uma formulação mais trabalhosa. Como a escolha é relativamente arbitrária, serão utilizadas funções baseadas em série de Taylor sem a expansão da derivada normal. Portanto

$$\begin{aligned} D\mathbf{w} &= \varphi_w a \\ D\mathbf{w} &= (A^{-1} B) a \end{aligned} \tag{7.5}$$

onde $D\mathbf{w}$ coleta a função de forma e derivadas da função de ponderação w .

Neste caso pode-se dizer que o método será de Petrov-Galerkin como apresentado no Capítulo 2. Caso não se utilize a expansão das derivadas normais, o método seguiria o critério de Galerkin.

A escolha acima, expressão (7.5) prevê a utilização das mesmas matrizes já empregadas para determinação da aproximação local. No caso do Método das Diferenças Finitas Generalizadas, por se tratar de critério de colocação, teve-se que prover as derivadas segundas. No presente caso isto não é necessário já que apenas a primeira derivada comparece na formulação. Contudo, quanto maior a ordem da base melhor a precisão dos termos de menor ordem na aproximação de um ponto, como prevê o Teorema da Fórmula de Taylor (Kaplan, 1973; Stewart, 1989). Assim será mantida a determinação até a segunda ordem utilizando-se de forma preliminar de uma certa hierarquia tipo- p .

Levando as expressões das funções acima na primeira expressão (7.1) e adotando para aproximação no contorno funções isoparamétricas chega-se

$$\begin{aligned}
& \left(- \int_{\Omega} \left[\left\{ \frac{\partial w}{\partial x} \right\} \left\{ \frac{\partial H}{\partial x} \right\}^T + \left\{ \frac{\partial w}{\partial y} \right\} \left\{ \frac{\partial H}{\partial y} \right\}^T \right] d\Omega \right) \{ \mathbf{a} \} + \\
& + \left(- \int_{\Omega} \left[\left\{ \frac{\partial w}{\partial x} \right\} \left\{ \frac{\partial Q}{\partial x} \right\}^T + \left\{ \frac{\partial w}{\partial y} \right\} \left\{ \frac{\partial Q}{\partial y} \right\}^T \right] d\Omega \right) \left\{ \frac{\partial \mathbf{a}}{\partial \mathbf{n}} \right\} \Big|_{\Gamma} + \\
& + \left(\int_{\Gamma_u} \{ \mathbf{w} \} \{ \mathbf{N} \}^T d\Gamma \right) \left\{ \frac{\partial \mathbf{a}}{\partial \mathbf{n}} \right\} \Big|_{\Gamma_u} = \int_{\Omega} \{ \mathbf{w} \} b d\Omega + \left(- \int_{\Gamma_i} \{ \mathbf{w} \} \{ \mathbf{N} \}^T d\Gamma \right) \left\{ \frac{\partial \mathbf{a}}{\partial \mathbf{n}} \right\} \Big|_{\Gamma_i}
\end{aligned} \tag{7.6}$$

Adotando $\bar{\mathbf{w}} = \mathbf{N}$ e introduzindo na segunda expressão de (7.1) obtém-se

$$\left(\int_{\Gamma_u} \{ \mathbf{N} \} \{ \mathbf{N} \}^T d\Gamma \right) \{ \mathbf{a} \} = \int_{\Gamma_u} \{ \mathbf{N} \} u_p d\Gamma \tag{7.7}$$

onde

$$\begin{aligned}
D\mathbf{w} &= \mathbf{A}^{-1} \mathbf{B} \\
DH &= (\mathbf{A} + \mathbf{A}_n)^{-1} \mathbf{B} \\
DQ &= (\mathbf{A} + \mathbf{A}_n)^{-1} \mathbf{B}_n \\
N &\text{- funções de forma isoparamétricas}
\end{aligned} \tag{7.8}$$

O termo de domínio produz uma matriz que multiplica as derivadas normais de contorno, parte delas prescritas, parte incógnita. Chamando

$$\left\{ \frac{\partial \mathbf{a}}{\partial \mathbf{n}} \right\} \Big|_{\Gamma} = \begin{Bmatrix} \mathbf{q}_i \\ \mathbf{q}_u \end{Bmatrix}; \quad \left\{ \frac{\partial \mathbf{a}}{\partial \mathbf{n}} \right\} \Big|_{\Gamma_i} = \mathbf{q}_i; \quad \left\{ \frac{\partial \mathbf{a}}{\partial \mathbf{n}} \right\} \Big|_{\Gamma_u} = \mathbf{q}_u \tag{7.9}$$

chega-se a

$$\begin{cases} \mathbf{K} \mathbf{a} + \mathbf{Q} \{ \mathbf{q}_i, \mathbf{q}_u \}^T + \mathbf{H}_u \mathbf{q}_u = \mathbf{F}_b + \mathbf{F}_i \mathbf{q}_i \\ \mathbf{G}^T \mathbf{a} = \mathbf{U} \end{cases} \tag{7.10}$$

particionando \mathbf{Q} em $\mathbf{Q}_u + \mathbf{Q}_i$ e reagrupando vem

$$\begin{cases} \mathbf{K} \mathbf{a} + (\mathbf{Q}_u + \mathbf{H}_u) \mathbf{q}_u = \mathbf{F}_b + (\mathbf{F}_i + \mathbf{Q}_i) \mathbf{q}_i \\ \mathbf{G}^T \mathbf{a} = \mathbf{U} \end{cases} \tag{7.11}$$

ou

$$\begin{bmatrix} \mathbf{K} & \mathbf{H} \\ \mathbf{G}^T & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \mathbf{a} \\ \mathbf{q} \end{Bmatrix} = \begin{Bmatrix} \mathbf{F} \\ \mathbf{U} \end{Bmatrix} \quad (7.12)$$

onde

$$\begin{aligned} \mathbf{K} &= - \int_{\Omega} \left[\left(\left\{ \frac{\partial w}{\partial x} \right\} \left\{ \frac{\partial H}{\partial x} \right\}^T + \left\{ \frac{\partial w}{\partial y} \right\} \left\{ \frac{\partial H}{\partial y} \right\}^T \right) \right] d\Omega \\ F_b &= \int_{\Omega} \{w\} b \, d\Omega \\ F_t &= - \int_{\Gamma_t} \{w\} \{N\}^T \, d\Gamma \\ \mathbf{Q} &= \mathbf{Q}_u + \mathbf{Q}_t = - \int_{\Omega} \left[\left(\left\{ \frac{\partial w}{\partial x} \right\} \left\{ \frac{\partial Q}{\partial x} \right\}^T + \left\{ \frac{\partial w}{\partial y} \right\} \left\{ \frac{\partial Q}{\partial y} \right\}^T \right) \right] d\Omega \\ \mathbf{G} &= \int_{\Gamma_u} \{N\} \{N\}^T \, d\Gamma \\ H_u &= \int_{\Gamma_u} \{w\} \{N\}^T \, d\Gamma \\ U &= \int_{\Gamma_u} \{N\} u_p \, d\Gamma \\ H &= \mathbf{Q}_u + H_u \\ F &= F_b + (F_t + \mathbf{Q}_t) \mathbf{q}_t \end{aligned} \quad (7.13)$$

7.3 - IMPLEMENTAÇÃO COMPUTACIONAL

A implementação do método proposto utilizando-se a estrutura computacional JFraMMe requer a modelagem de três classes. Inicialmente deve-se criar uma classe que implemente a interface **Method** de modo a configurar os dados relativos ao método, tais como funções de forma, de ponderação, dentre outros, como visto no Capítulo 4. Esta classe será denominada PGT (Petov-Galerkin Taylor). Como pode ser visto na Figura 7.1, esta classe possui dados semelhantes à classe GFD (Diferenças Finitas Generalizadas), complementados com dados para integração.

O método proposto não utiliza os critérios de colocação ou Galerkin já definidos como subclasse de **MWR** (Método dos Resíduos Ponderados). Desse modo será necessária à criação de uma classe denominada de **PetrovGalerkin** que se incumbirá de realizar as integrações considerando funções distintas para ponderação (no âmbito do Método dos Resíduos Ponderados) e para aproximação local. O algoritmo de integração dessa nova classe possui grande semelhança com a classe **Galerkin**, sendo as alterações no algoritmo bastante simples. Numa visão orientada a objetos se poderia dizer que a classe **Galerkin** é na verdade uma especialização da classe **PetrovGalerkin**.

Finalmente deve-se criar uma classe que forneça as aproximações locais para o problema que se soluciona denominada de **ODE2oPetrovGalerkin**, como consta da Figura 7.1.

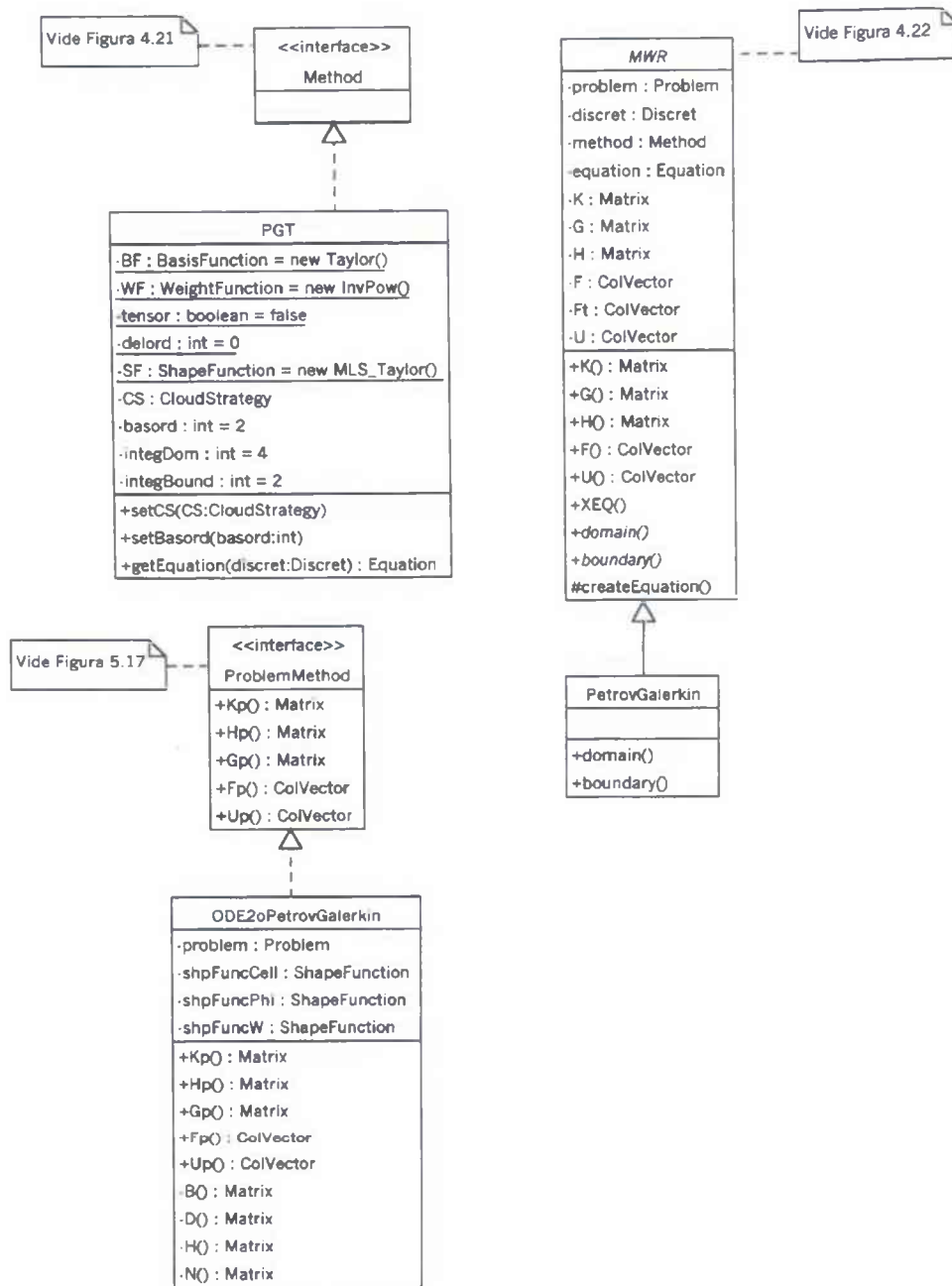


Figura 7.1 – Modelagem computacional do Método Petrov-Galerkin Taylor.

7.4 - CONSIDERAÇÕES FINAIS

Neste capítulo foi apresentado um Método sem Malha denominado de Petrov-Galekin Taylor (PGT). A inclusão do método, no corpo deste trabalho, tem como objetivo ilustrar a utilização da estrutura computacional JFraMMe na implementação de novos Métodos sem Malha. Por outro lado, visa documentar resultado atingido em decorrência do estudo, análise e implementação dos Métodos sem Malha escolhidos.

Fica para trabalhos futuros a exploração do método inclusive com oportuna análise de erro e comparativa.

8 - CONCLUSÕES E TRABALHOS FUTUROS

8.1 - INTRODUÇÃO

Neste capítulo de encerramento apresenta-se um resumo destacando as principais contribuições presentes no trabalho, as conclusões e uma seleção de possíveis rotas de investigação futura.

A consecução do objetivo inicialmente delineado, qual seja, o desenvolvimento de uma estrutura computacional orientada a objetos que facilite a criação de programas de computador para resolução de problemas modelados pelos Métodos sem Malha, resultou no enfrentamento de tarefa de complexidade multidisciplinar. Para a compreensão e implementação dos Métodos sem Malha foi necessário o estudo da Geometria Computacional e de Computação Gráfica na busca de um tratamento eficiente de uma nuvem dispersa de pontos nodais. Sem mencionar a incursão computacional que a implementação da orientação a objetos requer: estrutura de dados, algoritmos, UML, linguagens de programação.

A análise orientada a objetos mais do que uma ferramenta computacional é valorosa metodologia de análise de problemas. A dedução dos Métodos sem Malha a partir do Método dos Resíduos Ponderados e daí a construção de um arcabouço comum foi decorrência do princípio da busca da abstração de maior nível na modelação de um fenômeno.

A apresentação dos Métodos sem Malha selecionados procurou reunir de forma didática fragmentos coletados em uma bibliografia relativamente extensa. Com isto pretende-se que a monografia possa se constituir em texto para compreensão desses métodos e permita a frutificação em novos trabalhos.

8.2 - RESUMO E CONTRIBUIÇÕES

As principais contribuições foram assinaladas nas considerações finais presentes em cada capítulo. Devido ao tema proposto e ao seu desenvolvimento, as contribuições foram de natureza computacional, como pretendido, ocorrendo também contribuições aos Métodos sem Malha.

As principais contribuições aos Métodos sem Malha relacionam-se a seguir:

- A abordagem do critério de colocação em uma formulação fraca resulta em termo de contorno não observado nos métodos de colocação tradicionais. Alguns resultados apontam que um estudo mais aprofundado pode conduzir a um Método sem Malha baseado em funções de forma oriundas do Método de Mínimos Quadrados Móvel, sem a dependência marcante com o tipo de problema como ocorre com o Método do Ponto Finito e seu termo de estabilização.
- A exposição feita do Método das Diferenças Finitas Generalizada permite registrar algumas contribuições:
 - Justificativa da existência dos pontos gêmeos como sendo os limites das integrais, com colocação, realizadas no âmbito do Método dos Resíduos Ponderados.
 - Demonstração através de minimização do erro no domínio e contorno da formulação para a derivada normal.
 - Apresentação em um formato matricial, deduzida a partir do Método dos Resíduos Ponderados com colocação na formulação fraca, ao invés da abordagem tradicional mais algorítmica.
- A apresentação do Método sem Malha Hierárquico, *Hp-Clouds*, conduziu às seguintes contribuições:

- Exposição mais conceitual e didática com reduzido enfoque matemático, normalmente exacerbado na literatura.
 - Dedução da função de ponderação construída com uma base de funções *spline*, que é apenas citada na literatura.
 - Proposição de um algoritmo de recobrimento do domínio baseado em triangulação restringida a partir de conceitos da Geometria Computacional.
- O estudo do Método de Galerkin sem Malha resultou nas contribuições:
 - Formulação dos problemas baseada no Método dos Resíduos Ponderados que prescinde de um funcional natural e evita a imposição do contorno essencial via multiplicadores de Lagrange.
 - Dedução do Método dos Elementos Finitos com imposição de funções descontínuas no contorno essencial permitindo perceber com clareza a natureza da aproximação realizada.
 - Proposta de um *novo* Método sem Malha que utiliza o critério de Petrov-Galerkin para ponderação na formulação do Método dos Resíduos Ponderados e para função de forma emprega a expansão em série de Taylor.

No escopo da estrutura computacional orientada a objetos proposta, podem-se registrar as seguintes contribuições:

- Aplicação da análise orientada a objetos aos Métodos sem Malha tema não registrada na literatura consultada.
- Utilização generalizada do Método dos Resíduos Ponderados como modelo matemático base do modelo computacional.

- Construção de uma estrutura computacional (*framework*) para o desenvolvimento de aplicativos e de novos métodos de discretização.
- Utilização disseminada de padrões de projeto orientado a objetos na análise e concepção do modelo computacional e a utilização sistemática de diagramas UML na construção de programas. Prática de escassa referência na orientação a objetos aplicada a métodos de discretização.
- Apresentação de classes para modelagem de vários métodos de discretização e de classe baseada em delegação de algoritmo para execução da solução do sistema. Na literatura se resume a um método, quase sempre o Método dos Elementos Finitos.
- Generalização do conceito de *assembler* como adição indexada de matrizes e a atribuição em tempo de execução dessa operação a um objeto matriz específico.
- Estrutura de classes que permitem a consideração de diferentes tipos de problema e da sua solução por método específico que pode ser atribuído em tempo de execução.
- Utilização de classes para a descrição e discretização do domínio, comuns a todos os métodos e problemas, baseadas em CAD e Geometria Computacional.
- Releitura orientada a objetos, da análise de macro-comandos que constroem um algoritmo, proposta original de Zienkiewicz e Taylor (1989).
- Proposta de padrão de classes matriciais para utilização como objetos primitivos na modelação computacional de problemas de engenharia.
- Apresentação sintética de todo o projeto computacional através de diagramas UML permitindo a implementação em diferentes linguagens de programação.

8.3 - CONCLUSÕES

A estrutura computacional orientada a objetos construída para modelagem dos Métodos sem Malha permite as seguintes principais conclusões:

- O desenho através de diagramas da linguagem de modelagem UML é independente da linguagem de programação. O modelo foi implementado no ambiente científico MATLAB, em um programa denominado *Meshless*, e com a linguagem Java na estrutura computacional JFraMMe. A primeira versão facilitou a verificação dos desenvolvimentos teóricos numa programação semi-procedural. A segunda versão totalmente orientada a objetos explora os conceitos e princípios que demarcam a orientação a objetos.
- A viabilidade de modelagem sob uma mesma estrutura computacional de diversos métodos e diferentes problemas é atestada pela implementação de vários métodos de discretização, inclusive do Método dos Elementos Finitos. Quanto aos problemas foram testadas implementações para solução de equação diferencial ordinária de segunda ordem, da equação de Poisson e de elasticidade bidimensional.
- A utilização da análise orientada a objetos deve ser vista além da modelagem computacional, trata-se de metodologia de análise útil no trato da complexidade dos sistemas da natureza. A determinação do Método dos Resíduos Ponderados como ponto de partida do estudo foi decorrência do princípio da maior abstração preconizada pela análise orientada a objetos, como já mencionado.

O estudo e implementação dos Métodos sem Malha no âmbito da análise empreendida permitem algumas conclusões:

- Os Métodos sem Malha permitem a utilização em grande escala de desenho assistido por computador (CAD), tanto na

modelação geométrica do domínio do problema, como na discretização por pontos nodais. A inexistência de uma malha fixa, restringida a um contorno, viabiliza o uso sistemático dessa ferramenta.

- No atual estágio de desenvolvimento a utilização dos Métodos sem Malha requer um acúmulo de experiência numérica no seu trato. Principalmente no que se refere à definição da nuvem de pontos e a sua função de ponderação associada. Será necessária ainda alguma investigação para se alcançar a construção de um programa de caráter comercial.
- A grande facilidade de adensamento dos pontos nodais torna os Métodos sem Malha uma ferramenta importante na abordagem de problemas com singularidades, geométricas e de carregamento, ou com fronteiras móveis, como em fraturas e contato de superfícies. A utilização de indicadores de erro por ponto nodal com adensamento automático parece ser um caminho natural de investigação.
- O tratamento de vários materiais não é tão simples em comparação com o Método dos Elementos Finitos, por exemplo. As alternativas apresentadas requerem implementação cuidadosa tanto na definição dos atributos dos pontos nodais, como na escolha da nuvem de pontos.

Oden, Belytscko, Babuska e Hughes (2003) em artigo sobre perspectivas de investigação em mecânica computacional, enfatizam que a computação paralela de alto desempenho em conjunto com o desenvolvimento de programas criará uma revolução na análise de problemas em engenharia. Nesse sentido os Métodos sem Malha e análise orientada a objetos de programas de computador conforme apresentado são alternativas na busca da construção de ferramentas computacionais de alta performance.

8.4 - TRABALHOS FUTUROS

O objetivo estabelecido neste trabalho foi concebido de modo a lastrear investigação nos dois ramos aboradados da mecânica computacional. Assim, a monografia buscou um tratamento didático de modo a se constituir em texto que possa disseminar os Métodos sem Malha, e a estrutura computacional (*framework*) foi concebida de modo a abarcar novos problemas e métodos em uma construção que prioriza a flexibilidade e a expansão.

Em ambas as linhas mestras mencionadas as possibilidades de futura investigação foram assinaladas ao longo do texto. No que se refere aos Métodos sem Malha pode-se vislumbrar como novos trabalhos, dentre outros:

- Investigação do critério de colocação na formulação fraca, exposto no Capítulo 2, cujos resultados preliminares sugerem novo método de colocação baseado em funções de forma oriunda do Método de Mínimos Quadrados Móvel.
- Exploração do novo Método sem Malha, apenas proposto. A utilização de funções de forma baseadas em série de Taylor e obtidas no âmbito do Método de Mínimos Quadrados Móvel permite uma maior flexibilidade como na imposição do contorno essencial e carregamentos pontuais, características não presentes no Método de Galerkin sem Malha, por exemplo.
- Aplicação a outros tipos de problemas como os de mecânica da fratura e de contato de superfícies.
- Extensão a problemas tridimensionais sem grande esforço técnico ou teórico.
- Investigação da construção automática de nuvem de pontos e de funções de ponderação que induzam a um bom condicionamento da aproximação local. O uso da Geometria Computacional parece promissor.

Dentro de uma perspectiva mais computacional alguns novos trabalhos podem ser relacionados:

- Desenvolvimento de versão documentada do sistema computacional JFraMMe, com manuais de sistema e usuário, e construção de classes que isolem o núcleo central de código, mas permitindo o acoplamento de novas classes com a expansão do sistema.
- Projeto de ambiente gráfico de desenvolvimento de programas utilizando a linguagem de modelagem UML, tornando mais rápida a construção e o conhecimento da estrutura interna do sistema JFraMMe.
- Desenvolvimento de ferramentas gráficas para entrada de dados e visualização de resultados.
- Incorporação de um ambiente de trabalho interativo ao pacote JMatrix, viabilizando um programa para solução de problemas matriciais de baixo custo.

REFERÊNCIAS

- AHO, A. V.; HOPCROFT, J. E. **Data Structures and Algorithms**. Reading: Addison-Wesley, 1987. 427p.
- ATLURI, S. N.; ZHU, T. **New Concepts in Meshless Methods**. *Int. J. Numer. Meth. Engng.*, v. 47, p. 537-556. 2000.
- ARCHER, G. **Object-Oriented Nonlinear Dynamic Finite Element Analysis**. PhD thesis, University of California at Berkeley, May, 1996.
- ASKES, H.; PAMIN, J.; de BORST, R. **Dispersion analysis and element-free Galerkin solution of second- and fourth-order gradient-enhanced damage models**. *Int. J. Numer. Meth. Engng.*, v. 49, p. 811-832. 2000.
- BABUŠKA, I.; MELENK, J. M. **The Partition of Unity Method**. *Int. J. Numer. Meth. Engng.*, v. 40, p. 727-758. 1997.
- BARRY, W.; SAIGAL, S. **A Three-Dimensional Element-Free Galerkin Elastic and Elastoplastic Formulation**. *Int. J. Numer. Meth. Engng.*, v. 46, p. 671-693. 1999.
- BATHE, K. **Finite Element Procedures**. London: Prentice Hall International, 1996. 1037 p.
- BEISSEL, S.; BELYTSCHKO, T. **Nodal Integration of the Element-Free Galerkin Method**. *Comput. Methods Appl. Mech. Engng.*, v. 139, p. 49-74. 1996.
- BELYTSCHKO, T.; LU, Y. Y.; GU, L. **Element-Free Galerkin Method**. *Int. J. Numer. Meth. Engng.*, v. 37, p. 229-256. 1994.
- BELYTSCHKO, T.; KRONGAUZ, Y.; ORGAN, D.; FLEMING, M.; KRYSL, P. **Meshless Methods : An Overview and Recent Developments**. *Comput. Methods Appl. Mech. Engng.*, v. 139, p. 3-47. 1996.
- BELYTSCHKO, T.; TABBARA, M. **Dynamic Fracture Using Element-Free Galerkin Methods**. *Int. J. Numer. Meth. Engng.*, v. 39, p. 923-938. 1996.

- BESSET, D. H. **Object-Oriented Implementation of Numerical Methods – An Introduction with Java and Smalltalk**. San Francisco: Morgan Kaufmann Publishers, 2001. 766 p.
- BOISVERT, R. B.; DONGARRA, J. J.; POZO, R.; REMINGTON, K. A.; STEWART, G. W. **Developing numerical libraries in Java**. ACM 1998 Workshop on Java for High-Performance Network Computing, March 1998. Também disponível em <http://math.nist.gov/javanumerics/>. Acesso em: 24 out. 2003.
- BREBBIA, C. A.; DOMINGUEZ, J. **Boundary Elements – An Introductory Course**. Southampton: Computational Mechanics Publications, 1989. 292 p.
- CONNOR, J. J.; BREBBIA, C. A. **Finite Element Techniques for Fluid Flow**. London: Newnes-Butterworths, 1977. 310 p.
- COOK, R. D.; MALKUS, D. S.; PLESHA, M. P. **Concepts and Applications of Finite Element Analysis**. New York: John Wiley & Sons, 1989. 630 p.
- CORDES, L. W.; MORAN, B. **Treatment of material discontinuity in the Element-Free Galerkin method**. *Comput. Methods Appl. Mech. Engrg.*, v. 139, p. 75-89. 1996.
- CRANDALL, S. H. **Engineering Analysis: A Survey of Numerical Procedures**. New York: MacGraw-Hill, 1956. 417 p.
- de BERG, M.; van KREVELD, M.; OVERMARS, M.; SCHWARZKOPF, O. **Computational Geometry**. Berlin: Springer-Verlag, 2000. 367p.
- DEVLOO, P. **PZ: An Object Oriented Environment for Scientific Programming**. *Comput. Methods Appl. Mech. Engrg.*, v. 150, p. 133-153. 1997.
- DOLBOW, J.; BELYTSCHKO, T. **An Introduction to Programming the Meshless Element Free Galerkin Method**. Archives in Computational Mechanics and Engineering.
- DU, C. **An Element-Free Galerkin Method for Simulation of Stationary Two-Dimensional Shallow Water Flows in Rivers**. *Comput. Methods Appl. Mech. Engrg.*, v. 182, p. 89-107. 2000.

- DUARTE, C. A. **A Review of Some Meshless Methods to solve Partial Differential Equations.** Austin: Texas Institute for Computational and Applied Mathematics. TICAM Report 95-06. 1995. 37 p.
- DUARTE, C. A. **The HP Cloud Method.** PhD Dissertation. The University of Texas at Austin, dec, 1996.
- DUARTE, C. A.; ODEN, J. T. **Hp Clouds – A Meshless Method to Solve Boundary-Value Problems.** Austin: Texas Institute for Computational and Applied Mathematics. TICAM Report 95-05. 1995. 66 p.
- DUARTE, C. A.; ODEN, J. T. **An hp Adaptive Method Using Clouds.** *Comput. Methods Appl. Mech. Engrg.*, v. 139, p. 237-262. 1996.
- DUBOIS-PÉLERIN, Y.; PEGON P. **Object-Oriented Programming in Nonlinear Finite Element Analysis.** *Comp. Structures*, v.67, p. 225-241. 1998.
- DUFLOT, M.; NGUYEN-DANG, H. **A Trully Meshless Galerkin Method Based on a Moving Least Square Quadrature.** *Commun. Numer. Meth. Engng.*, v. 18, p. 1-9. 2002.
- EYHERAMENDY, D.; ZIMMERMANN, T. **Object-Oriented Finite Elements: IV. Symbolic Derivations and Automatic Programming of Nonlinear Formulations.** *Comput. Methods Appl. Mech. Engrg.*, v. 190, p. 2729-2751. 2001.
- FELIPPA, C. A. **Advanced Finite Element Methods.** Boulder: 2000. Disponível em <http://caswww.colorado.edu/Felippa.d/FelippaHome.d/>. Acesso em: 24 out. 2003.
- FINLAYSON, B. A. **The Method of Weighted Residuals and Variational Principles.** New York: MacGraw-Hill, 1972. 412 p.
- FLEMING, M.; CHU, Y. A.; MORAN, B.; BELYTSCHKO, T. **Enriched Element-Free Galerkin Methods for Crack Tip Fields.** *Int. J. Numer. Meth. Engng.*, v. 40, p. 1483-1504. 1997.
- FOLEY, J. D.; van DAM, A.; FEINER, S. K.; HUGHES, J. H. **Computer Graphics Principles and Practice.** Reading: Addison-Wesley, 1990. 1174p.

- FORDE, B. W.; FOSCHI, R. O.; STIEMER, S. F. **Object-Oriented Finite Element Analysis**. *Comp. Structures*, v. 34, p. 355-374. 1990.
- FOWLEY, M.; SCOTT, K. **UML Essencial: Um breve guia para a linguagem-padrão de modelagem de objetos**. 2ª Edição. Porto Alegre: Bookman, 2000. 169 p.
- FURLAN, J. D. **Modelagem de Objetos Através da UML – The Unified Modeling Language**. , São Paulo: Mackron Books, 1998. 329 p.
- GAMMA, E.; HELM, R.; JOHNSON, R.; VLISSIDES, J. **Padrões de Projeto – Soluções Reutilizáveis de Software Orientado a Objetos**. Porto Alegre: Bookman, 2000. 364p.
- GARCIA, O.; FANCELLO, E. A.; BARCELLOS, C. S.; DUARTE, C. A. **hp-Clouds in Midlin's Thick Plate Model**. *Int. J. Numer. Meth. Engng.*, v. 47, p. 1381-1400. 2000.
- GEL'FAND, I. M.; SHILOV, G. E. **Generalized Functions**. New York: Academic Press, 1964. Vol 1, 423 p.
- GODOY, L. A. **Ill-Conditioned Stars in the Finite Difference Method for Arbitrary Meshes**. *Comp. Structures*, v. 22, p. 469-473.1986.
- GONÇALVES, E. M.; PAMPLONA , D. C. **Uma Nova Técnica de discretização Numérica de Problemas de Contorno**. In: II CONGRESSO LATINO AMERICANO SOBRE MÉTODOS COMPUTACIONAIS EM ENGENHARIA, Curitiba, Brasil. 1980. p. 755-766.
- GOODRICH, M. T.; TAMASSIA, R. **Estruturas de Dados e Algoritmos em Java**. São Paulo: Bookman, 2002. 584 p.
- HARRINGTON, S. **Computer Graphics. A Programming Approach**. New York: MacGraw-Hill, 1987. 466 p.
- HEGEN, D. **Element-Free Galerkin methods in combination with finite element approaches**. *Comput. Methods Appl. Mech. Engrg.*, v. 135, p. 143-166. 1996.

- HILDEBRAND, F. B. **Introduction to Numerical Analysis**. New York: MacGraw-Hill, 1956. 511 p.
- HÄUSSLERR-COMBE, U.; KORN, C. **An adaptive approach with the Element-Free-Galerkin method**. *Comput. Methods Appl. Mech. Engrg.*, v. 162, p. 203-222. 1998.
- JANDL Jr., P. **Introdução ao Java**. São Paulo: Berkeley, 2002. 518p.
- JANDL Jr., P. **Mais Java**. São Paulo: Berkeley, 2003. 635p.
- JANSEN, P. S. **Finite Difference Technics for Variable Grids**. *Comp. Structures*, v. 2, p. 17-29. 1972.
- Java Matrix Package**. The MathWorks e National Institute of Standards and Technology (NIST). 1999. Disponível em <http://math.nist.gov/javanumerics/>.
- JEREMIČ, B.; STURE, S. **Tensor Objects in Finite Element Programming**. *Int. J. Numer. Meth. Engrg.*, v. 41, p. 113-126. 1998.
- KAPLAN, W.; LEWIS, D. **Cálculo e Álgebra Linear**. Rio de Janeiro: LTC, vol 4, 1973.
- KALJEVIĆ, I.; SAIGAL, S. **An Improved Element Free Galerkin Formulation**. *Int. J. Numer. Meth. Engrg.*, v. 40, p. 2953-2974. 1997.
- KRONGAUX, Y.; BELYTSCHKO, T. **Enforcement of Essential Boundary Conditions in Meshless Approximation Using Finite Elements**. *Comput. Methods Appl. Mech. Engrg.*, v. 131, p. 133-145. 1996.
- KRYSL, P.; BELYTSCHKO, T. **ESFLIB: An Library to Compute the Element Free Galerkin Shape Functions**. *Comput. Methods Appl. Mech. Engrg.*, v. 190, p. 2181-2205. 2001.
- LANCASTER, P.; ŠALKAUSKAS, K. **Surface Generated by Moving Least Squares Methods**. *Math. Comp.*, v. 37, p. 141-158. 1981.
- LANCASTER, P.; ŠALKAUSKAS, K. **Curve and Surface Fitting: An Introduction**. London: Academic Press. 1986.

- LEMAY, L.; CADENHEAD, R. **Aprenda em 21 Dias JAVA 2 – Professional Reference**. Rio de Janeiro: Campus, 2001. 531 p.
- LISZKA, T.; ORKISZ, J. **The Finite Difference Method at Arbitrary Irregular Grids and its Application in Applied Mechanics**. *Comp. Structures*, v. 11, p. 83-95.1980.
- LISZKA, T.; DUARTE, C. A.; TWORZYDLO, W. W. **hp-Meshless cloud method**. *Comput. Methods Appl. Mech. Engrg.*, v. 139, p. 263-288. 1996.
- LIU, W. K.; JUN, S.; LI, S.; BELYTSCHKO, T. **Reproducing Kernel Particle Methods for Structural Dynamics**. *Int. J. Numer. Meth. Engrg.*, v. 38, p. 1655-1679. 1995.
- LIU, W. K.; BELYTSCHKO, T.; ODEN, J. T. **Meshless Methods**. *Comput. Methods Appl. Mech. Engrg.*, v. 139, n. 1-4, p. 1-440, dez. 1996.
- LU, J.; WHITE, D. W.; CHEN, W.; DUNSMORE, H. E. **A Matrix Class Library in C++ for Structural Engineering Computing**. *Comp. Structures*, v. 55, p. 95-111.1995.
- LU, Y. Y.; BELYTSCHKO, T.; TABBARA, M. **Element-Free Galerkin Method for Wave Propagation and Dynamic Fracture**. *Comput. Methods Appl. Mech. Engrg.*, v. 126, p. 131-153. 1995.
- LUCY, L. B. **A Numerical Approach to the Testing of the Fission Hypothesis**. *The Astron. J.*, v. 8, n. 12, p. 1013-1024. 1977.
- MASTERS, I.; USMANI, A. S.; CROSS, J. T.; LEWIS, R. W. **Finite Element Analysis of Solidification Using Object-Oriented and Parallel Techniques**. *Int. J. Numer. Meth. Engrg.*, v. 40, p. 2891-2909. 1997.
- McKENNA, F. T. **Object-Oriented Finite Element Programming: Framework for Analysis, Algorithms and Parallel Computing**. PhD thesis, University of California at Berkley, Fall, 1997.
- MENDONÇA, P.; BARCELLOS, C. S.; DUARTE, C. A. **Investigation on the hp-Cloud Method by Solving Timoshenko Beam Problems**. *Computational Mechanics*, v. 25, p. 286-295. 2000.

- MOREIRA, J. E.; MIDKIFF, S. P.; GUPTA, M. **From flop to megaflops: Java for technical computing.** ACM Transactions on Programming Languages and Systems Volume 22, Issue 2 (2000), pp. 265-295. Também disponível em <http://math.nist.gov/javanumerics/>.
- NAYROLES, B.; TOUZOT, G.; VILLON, P. **Generalizing the Finite Element Method: Diffuse Approximation and Diffuse Elements.** Comput. Mech., v. 10, p. 307-318. 1992.
- NOGUCHI, H., KAWASHIMA, T.; MIYAMURA, T. **Element- Free Analyses of Shell and Spatial Structures.** Int. J. Numer. Meth. Engng., v. 47, p. 1215-1240. 2000.
- ODEN, J. T.; BELYTSCHKO, T.; BABUSKA, I.; HUGHES, T. J. R. **Research directions in computational mechanics.** Comput. Methods Appl. Mech. Engrg., v. 192, p. 913-922. 2003.
- OÑATE, E.; IDELSOHN, S.; ZIENKIEWICZ, O. C.; TAYLOR, R. L. **A Finite Point Method in Computational Mechanics. Applications to Convective Transport and Fluid Flow.** Int. J. Numer. Meth. Engng., v. 39, p. 3839-3866. 1996a.
- OÑATE, E.; IDELSOHN, S.; ZIENKIEWICZ, O. C.; TAYLOR, R. L.; SACCO, C. **A Stabilized Finite Point Method for Analysis of Fluid Mechanics Problems.** Comput. Methods Appl. Mech. Engrg., v. 139, p. 315-346. 1996b.
- OÑATE, E.; PERAZZO, F.; MIQUEL, J. **A Finite Point Method for Elasticity Problems.** Comp. Structures, v. 79, p. 2151-2163. 2001.
- OUATOUATI, A.; JOHNSON, D. A. **A New Approach for Numerical Modal Analysis Using the Element-Free Method.** Int. J. Numer. Meth. Engng., v. 46, p. 1-27. 1999.
- PERRONE, N.; KAO, R. **A General Finite Difference Method for Arbitrary Meshes.** Comp. Structures, v. 5, p. 45-58. 1975.
- PIDAPARTI, R. M.; HUDLI, A. V. **Dynamic Analysis of Structures Using Object-Oriented Techniques.** Comp. Structures, v. 49, p. 149-156. 1993.

- Poseidon for UML.** Gentleware AG: 2003. Disponível em <http://www.gentleware.com/>. Acesso em 15 jun. 2003.
- PRENTER, P. M. **Splines and Variational Methods.** New York: John Wiley & Sons, 1975. 323 p.
- PREPARATA, F. P.; SHAMOS, M. I. **Computational Geometry. An Introduction.** Berlin: Springer-Verlag, 2000. 367p.
- PULINO Fº, A. R. **Diferenças Finitas para Malhas Arbitrárias (Via Série de Taylor).** 1989. 66 p. Tese (Mestrado em Engenharia Mecânica) – Departamento de Mecânica Computacional, Faculdade de Engenharia, Universidade Estadual de Campinas, Campinas, São Paulo.
- RANDLES, P. W.; LIBERSKY, L. D. **Smoothed Particle Hydrodynamics: Some Recent Improvements and Applications.** *Comput. Methods Appl. Mech. Engrg.*, v. 139, p. 375-408. 1996.
- REDDY, J. N. **An Introduction to the Finite Element Method.** New York: MacGraw-Hill, 1993. 684 p.
- RUMBAUGH, J.; BLAHA, M.; PREMERLANI, W.; EDDY, F.; LORENSEN, W. **Object-Oriented Modeling and Design.** Englewood Cliffs: Prentice Hall, 1991. 500 p.
- SAMPATH, R.; ZABARAS, N. **An Object-Oriented Framework for the Implementation of Adjoin Techniques in the Design and Control of Complex Continuum Systems.** *Int. J. Numer. Meth. Engng.*, v. 48, p. 239-266. 2000.
- SÁNCHEZ, J.; PULINO FILHO; A. R. **Padrão Orientado a Objetos de Classes Matriciais Utilizando a Linguagem Java.** In CILAMCE - XXIV Iberian Latin-American Congress on Computational Methods in Engineering., Ouro Preto, MG, Brasil, 2003.
- SCHOZ, S. P. **Elements of an Object-Oriented FEM++ Programming C++.** *Comp. Structures*, v. 43, n. 3, p. 517-529. 1992.