

MINISTÉRIO DA EDUCAÇÃO E DO DESPORTO
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
Escola de Engenharia
Programa de Pós-Graduação em Engenharia Civil

Uma Implementação Geral de Elementos de Contorno e sua Aplicação a Problemas de Flexão de Placas

Rogério José Marczak

Tese para obtenção do título de Doutor em Engenharia

Porto Alegre
2002

MINISTÉRIO DA EDUCAÇÃO E DO DESPORTO
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
Escola de Engenharia
Programa de Pós-Graduação em Engenharia Civil

Uma Implementação Geral de Elementos de Contorno e sua Aplicação a Problemas de Flexão de Placas

por

Rogério José Marczak, M.Eng.Mec.

Trabalho realizado no Departamento de Engenharia Civil da Escola de Engenharia da UFRGS, dentro do Programa de Pós-Graduação em Engenharia Civil como requisito parcial para obtenção do título de Doutor em Engenharia.

Área de concentração: Estruturas

Porto Alegre
2002.

Esta tese foi julgada adequada para a obtenção do título de DOUTOR EM ENGENHARIA e aprovada em sua forma final pelo orientador e pelo Programa de Pós-Graduação em Engenharia Civil da Universidade Federal do Rio Grande do Sul.

Prof. Guillermo Juan Creus, D.Sc.

Orientador

Prof. Francisco P. S. L. Gastal, Ph.D.

Coordenador do PPGEC - EE - UFRGS

Banca examinadora:

Prof. Vilson Sérgio Venturini

Ph.D., University of Southampton

Prof. Marco Tullio Menna Barreto de Vilhena

Dr.Eng., Universidade Federal do Rio Grande do Sul

Prof^a. Carla Maria Dal Sasso Freitas

Dr^a. Ciências da Computação, Universidade Federal do Rio Grande do Sul

Sumário

Sumário	i
Agradecimentos	v
Resumo	ix
Abstract	xi
Simbologia	xiii
1 Introdução	1
1.1 Considerações preliminares	1
1.2 Revisão bibliográfica	3
1.3 Objetivos do trabalho	14
1.4 Organização do texto	16
2 Os Modelos de Placa de Mindlin e de Reissner	19
2.1 Notas históricas	19
2.2 Relações cinemáticas	22
2.3 Equações de equilíbrio e condições de contorno	27
2.4 Campo de tensões	33

2.5	Equações para um estado de equilíbrio auxiliar	37
2.6	Resumo do Capítulo 2	41
3	Formulação Integral	43
3.1	Formalismo geral	43
3.2	Formulação integral para o problema de membrana	46
3.3	Formulação integral para o problema de flexão	49
3.4	Equações integrais sobre o contorno	53
3.5	Equações integrais para flexão não-linear de placas	61
3.6	Resumo do Capítulo 3	72
4	Implementação Numérica	75
4.1	Interpolação de geometria, variáveis e carregamentos	76
4.2	Discretização das equações de membrana	83
4.3	Discretização das equações de flexão	88
4.4	Discretização das equações para cálculo dos gradientes dos deslocamentos translacionais	92
4.5	Forma geral dos sistemas de equações resultantes	97
4.6	Tratamento numérico de subregiões	103
4.7	Resumo do Capítulo 4	111
5	Tratamento das Integrais Singulares	113
5.1	Integração numérica de núcleos singulares	113
5.2	Aplicação do método direto aos tensores ${}^m\mathbf{T}$ e ${}^f\mathbf{T}$	126
5.3	Resumo do Capítulo 5	133

6	Proposta de uma Arquitetura Orientada a Objetos para o MEC.....	135
6.1	Conceitos de programação orientada a objetos	135
6.2	Características comuns de arquiteturas OO para elementos finitos e elementos de contorno	141
6.3	Algumas referências relevantes	144
6.4	A biblioteca mcBEM	158
6.5	Resumo do Capítulo 6	198
7	Resultados Numéricos.....	199
7.1	Problemas lineares	200
7.2	Problemas não-lineares	213
7.3	Aspectos numéricos	227
7.4	Resumo do Capítulo 7	233
8	Conclusões.....	235
8.1	Recomendações para continuidade da pesquisa	238
	Referências Bibliográficas	241
A	Soluções Fundamentais.....	265
A.1	Tensores fundamentais de deslocamentos	266
A.2	Tensores fundamentais de tração	267
A.3	Tensores derivados	267
A.4	Tensores envolvidos nos termos de carregamento	269
B	Diferenciação de Integrais com Núcleos Singulares.....	271
B.1	Cálculo de $\frac{\partial I_i^N}{\partial x_\gamma(P)}$	272

B.2	Cálculo de $\frac{\partial I_i^q}{\partial x_\gamma(P)}$	277
B.3	Cálculo de $\frac{\partial J_\alpha^N}{\partial x_\gamma(P)}$	278
B.4	Cálculo de $\frac{\partial J_\alpha^q}{\partial x_\gamma(P)}$	280
C	Expansões Assintóticas para os Tensores Fortemente Singulares.	281
C.1	Algumas relações úteis	282
C.2	Expansões assintóticas para $\frac{r;n}{r}$	285
C.3	Expansões assintóticas para ${}^m T_{\alpha\beta}$	286
C.4	Expansões assintóticas para ${}^f T_{ij}$	289

Agradecimentos

– Ao meu orientador, Prof. Guillermo Juan Creus, pela orientação deste trabalho e pela tolerância com que permitiu que este fosse realizado.

– Aos professores Milton Zaro, Jun Fonseca, Alberto Tamagna e Ignacio Iturrioz, cujos incentivos e cobranças constantes me mantiveram motivado para concluir este trabalho.

– A todos os colegas do Grupo de Mecânica Aplicada do Departamento de Engenharia Mecânica da UFRGS, pelo encorajamento, pelo ambiente de trabalho descontraído e pelas inúmeras discussões sobre mecânica computacional: Eduardo "Olho Esbugalhado" Lenz, Luiz "Gurtz" Gertz, Pablo Rojas, Carlos "Carne Seca" Guilherme, Rafael Laranja e André "Wursti" Beck.

– Ao meu amigo Prof. Tancredo "Tarsky" Westphal Jr., atualmente no Instituts für Technische Mechanik, da Universidade de Karlsruhe, pelas incontáveis referências e dicas enviadas. Também ao meu colega Lauro "Monstro" Nicolazzi, do Depto. Eng. Mecânica da UFSC, por todo apoio e amizade.

– Aos professores do Departamento de Engenharia Mecânica da UFRGS, por toda colaboração durante a realização deste trabalho.

– Ao Prof. Massimo Guiggiani, do Departamento de Engenharia Mecânica, Nuclear e de Produção da Universidade de Pisa, pelos artigos e comentários sobre cálculo direto de integrais singulares.

– Ao Dr. Robert Davies, da Statistics Research Associates Limited, Wellington, Nova Zelândia, por tantos e-mails esclarecedores a respeito de C++ e da biblioteca *Newmat*.

– Aos amigos plastimodelistas do Wonderland - Museu de Miniaturas, cujas reuniões mensais ajudaram a manter minha sanidade mental nos momentos mais difíceis.

– Ao meu antigo professor de Física e Desenho, Hiroki Oba, pela minha iniciação inadvertida na carreira científica e tecnológica.

– Ao Mr. Jack Daniel, pela companhia da sua invenção ao longo dos últimos anos.

– A meus pais e a meus irmãos, pelo amor e incentivo incondicionais.

– Finalmente, um agradecimento especial às minhas filhas Fernanda e Ana Cláudia, que por tantos períodos abdicaram de um pai para que esta jornada fosse cumprida; e à Ligia, minha incansável esposa que arcou sozinha com as responsabilidades da família quando eu estive ausente. A vocês eu espero poder retribuir todo amor e carinho.

À memória de meu avô, Bento.

Resumo

O presente trabalho aborda a aplicação do método dos elementos de contorno (MEC) para solução de problemas de flexão linear e geometricamente não-linear de placas semi-espessas. Os modelos de placa empregados consideraram a influência do cisalhamento através de teorias de primeira ordem, especificamente as de Mindlin e Reissner.

Uma formulação integral unificada dos modelos de placa utilizados é desenvolvida para o operador de Navier do problema, onde foram mantidos alguns termos de ordem superior no tensor deformação de Green. A formulação integral do problema de membrana acoplado ao de flexão é igualmente desenvolvida, levando a um sistema de equações integrais não-lineares que descreve completamente problemas de placas que envolvem grandes deslocamentos. Estas equações podem ser particularizadas para problemas de flexão linear e estabilidade elástica. Tendo em vista a necessidade de se considerar derivadas dos deslocamentos translacionais, as equações integrais correspondentes ao gradiente dos deslocamentos também foram deduzidas, caracterizando uma formulação hipersingular.

O método empregado para solução numérica do sistema de equações integrais foi o método direto dos elementos de contorno. Um tratamento das integrais fortemente singulares presentes nas equações foi realizado, baseado em expansões assintóticas dos núcleos. Deste procedimento resulta uma abordagem regularizada que emprega apenas quadraturas padrão de Gauss-Legendre.

A formulação foi implementada em uma proposta de arquitetura orientada a objetos especialmente desenvolvida para o MEC. Esta arquitetura incorpora três grandes módulos de classes: auxiliares, de modelo computacional e de análise. O uso unificado destes módulos é particularmente adaptados para solução numérica de problemas descritos através de métodos integrais, e contempla eficientemente conceitos de extensibilidade e reusabilidade.

São apresentados resultados para problemas estáticos empregando elementos constantes, lineares e quadráticos para várias geometrias, condições de contorno e carregamentos.

Abstract

This work presents an application of the boundary element method (BEM) to linear and geometrically non-linear bending analysis of moderately thick plates. The plate models used account for shear influence by using the first order plate theories of Mindlin and Reissner.

An unified integral formulation for the plate models employed is derived for the corresponding Navier operator, and some higher order terms of the Green strain tensor are included. The integral formulation for membrane-bending coupling is also developed leading to a non-linear integral equation system that describes completely large displacement plate bending problems. The equations can be particularized to linear bending and elastic stability analysis. Because the coupling terms consider some derivatives of the translational displacements, the corresponding derivative integral equations were also derived, resulting an hypersingular approach.

The direct boundary element method was used for the numerical solution of the integral equation system. A treatment for the strongly singular integrals was carried out, based on asymptotic expansions of the kernels. This procedure results a regularized approach which uses only standard Gauss-Legendre quadrature rules.

The formulation was implemented through a proposed object-oriented framework specially developed to BEM applications. The design provides three basic class modules:

auxiliary, computational model and analysis. The unified application of these modules is particularly adapted for the numerical solution of problems through integral equation methods, and enables efficient extensibility and reusability of the code.

Results for static problems are presented using constant, linear and quadratic boundary elements, for various geometry, boundary conditions, and loading configurations.

Simbologia

Índices:

n_d	Dimensão do domínio sob análise (\mathfrak{R}^{n_d}).
n_v	Número de variáveis do problema.
n_n	Número de nós físicos de um elemento Γ_n ou de uma célula Ω_r .
n_p	Número de pontos geométricos de um elemento Γ_n ou de uma célula Ω_r .
\bar{n}_v	Número de derivadas consideradas para cada variável do problema.
a, b	Índices que variam de 1 a n_d .
i, j, k, l, m, n	Índices que variam de 1 a 3. Índices que variam de 1 a n_v .
p, q	Índices que variam de 1 a n_p .
r, s	Índices que variam de 1 a n_n .
I, J, K, L	Índices que variam de 1 a $n_n n_v$.
M, N, O, P	Índices que variam de 1 a $\bar{n}_v n_v$.
$\alpha, \beta, \gamma, \delta$	Índices que variam de 1 a 2.
g	Índice que varia de 1 a 3 e não implica em somatório.
ω	Índice que varia de 1 a 2 e não implica em somatório.

Sobrescritos à direita:

- i Referente à subregião i .
- l Indica parcela linear.
- n Indica parcela não-linear. Indica parcela regular ou quase singular.
- N Relativo ao acoplamento flexão-membrana.
- q Relativo aos termos de carregamento.
- s Indica parcela singular.

Sobrescritos à esquerda:

- m Indica parcela correspondente ao problema de membrana.
- f Indica parcela correspondente ao problema de flexão.
- α Relativo aos deslocamentos translacionais paralelos ao plano da placa.
- β Relativo aos deslocamentos translacionais transversais ao plano da placa.

Letras arábicas:

- a Dimensão lateral de placas retangulares.
- $A(z)$ Combinação de funções de Bessel.
- \mathbf{A} Matriz de coeficientes do sistema resultante em problemas lineares.
- b Dimensão lateral de placas retangulares.

- $B(z)$ Combinação de funções de Bessel.
- b_i Forças de corpo atuantes na direção e_i .
- \mathbf{b}_r Vetor resultante da integração dos termos de carregamento sobre as células de domínio.
- \mathbf{B} Matriz resultante da sobreposição de \mathbf{B}_r e \mathbf{b}_r .
- \mathbf{B}_r Matriz resultante da integração dos termos não-lineares sobre as células de domínio.
- \mathbf{c} Matriz ou vetor de termos convectivos.
- C Módulo de rigidez de membrana; $C = \frac{Eh}{1-\nu^2}$.
- \mathbf{C} Matriz dos fatores geométricos do tensor \mathbf{T} .
- D Módulo de rigidez à flexão; $D = \frac{Eh^3}{12(1-\nu^2)}$.
- e_i Base canônica do sistema de coordenadas cartesiano global.
- E Módulo de elasticidade.
- $E_{\alpha\beta}$ Componentes de membrana do tensor deformação de Green.
- F_3, F_6 Coeficientes livres da solução fundamental dos modelos de placa.
- F_{ij} Componentes do tensor gradiente de deformações.
- \mathbf{f} Vetor resultante dos carregamentos $\mathbf{F}_r + \mathbf{v}$. Vetor excitação do sistema resultante em problemas lineares.
- f_i^* Forças de corpo generalizadas na direção i .
- F_i, F_i^* Forças concentradas na direção i .

- $\bar{\mathbf{F}}_r$ Matriz resultante da integração dos carregamentos externos sobre as células de domínio.
- $\mathbf{F}_{-1}, \mathbf{F}_{-2}$ Expansões assintóticas de um tensor em torno do ponto singular.
- G Módulo de cisalhamento; $G = \frac{E}{2(1+\nu)}$.
- \mathbf{G}_n Matriz resultante da integração de \mathbf{U} sobre os elementos de contorno.
- G_i^* Momentos concentrados na direção i .
- h Espessura da placa.
- \mathbf{H} Matriz resultante da sobreposição de \mathbf{H}_n .
- \mathbf{H}_n Matriz resultante da integração \mathbf{T} sobre os elementos de contorno adicionada de \mathbf{C} , no caso singular.
- $\bar{\mathbf{H}}_n$ Matriz resultante da integral de \mathbf{T} sobre os elementos de contorno.
- I_1, I_2 Integrais genéricas uni e bidimensionais, respectivamente.
- I^ϵ Integral definida sobre um círculo de raio ϵ .
- \mathbf{I} Resultado da integração de um núcleo genérico \mathbf{K} . Tensor identidade.
- $|J_n|$ Jacobiano referente a um elemento de contorno Γ_n .
- $|J_r|$ Jacobiano referente a uma célula de domínio Ω_r .
- K Número de pontos de integração utilizados por quadraturas.
- $K_0(z), K_1(z)$ Funções de Bessel de segunda espécie de ordem 0 e 1, respectivamente.
- $K_{\alpha\beta}$ Componentes de flexão do tensor deformação de Green.
- $K_{ij}(\xi, \eta)$ Núcleo singular genérico.

l	Comprimento de um elemento de contorno.
m_α	Momentos distribuídos correspondentes à direção e_i .
m_f	Fator dependente do modelo de placa.
M_α	Momentos concentradas na direção i .
$M_{\alpha\beta}$	Esforços generalizados de flexão (momentos resultantes).
$\overline{M}_{\alpha\beta}$	Valores prescritos de $M_{\alpha\beta}$.
\mathbf{n}	Vetor canônico normal do sistema de coordenadas local sobre o contorno da placa.
$N_{\alpha\beta}$	Esforços generalizados de membrana.
$\overline{N}_{\alpha\beta}$	Valores prescritos de $N_{\alpha\beta}$.
\mathbf{N}^*	Matriz de carregamentos críticos de membrana.
p	Ponto fonte sobre Γ .
P	Ponto fonte sobre Γ ou Ω .
P_{ij}	Matriz de transformação de coordenadas associada a um nó de Γ_n ou Ω_r .
q	Ponto campo sobre Γ .
q_i	Componentes de carregamento distribuído atuante na direção e_i .
\hat{q}_i	Componentes de pseudo-carregamento distribuído atuante na direção e_i .
\mathbf{q}	Vetor de carregamentos nodais.
\mathbf{q}^j	Vetor de carregamentos nodais correspondente ao j -ésimo elemento de contorno ou célula de domínio.

- Q Ponto campo sobre Γ ou Ω .
- Q_α Esforços generalizados de cisalhamento (esforços cortantes).
- \bar{Q}_α Valores prescritos de Q_α .
- r Distância do ponto-fonte ao ponto-campo; $r = \|Q - P\|$. Parâmetro de deslocamento.
- $r_{,n}$ Derivada de r segundo a direção normal.
- r_α Projeção de r segundo a direção x_α .
- $r_{,\alpha}$ Derivada de r segundo a direção x_α .
- R** Matriz de transformação de coordenadas associada a um elemento de contorno ou célula de domínio. Parâmetro de carregamento.
- S^i Ponto genérico sobre o qual atua uma carga concentrada.
- s** Vetor canônico transversal do sistema de coordenadas local sobre o contorno da placa.
- S_{ij} Componentes do 2º tensor tensão de Piola-Kirchhoff.
- t_i Forças de superfície atuantes na direção e_i .
- \bar{t}_i Valores prescritos de t_i .
- t** Vetor canônico tangencial do sistema de coordenadas local sobre o contorno da placa.
- \mathbf{t}^j Vetor de trações nodais correspondente ao j -ésimo elemento de contorno ou célula de domínio.

- T** Tensor representativo do campo de trações da solução fundamental.
- u_i Componentes de deslocamento translacional generalizado da placa.
- \bar{u}_i Valores prescritos de u_i .
- \tilde{u} Solução aproximada de um problema.
- u** Vetor de deslocamentos nodais.
- \mathbf{u}^j Vetor de deslocamentos nodais correspondente ao j -ésimo elemento de contorno ou célula de domínio.
- U_i Componentes do campo de deslocamentos tridimensional em um ponto qualquer da placa.
- U** Tensor representativo do campo de deslocamentos da solução fundamental.
- V Volume compreendido pela placa.
- V** Tensor das integrais de carregamento distribuído sobre o domínio.
- v** Vetor dos termos de carregamento concentrado sobre o domínio.
- x** Vetor de coordenadas. Vetor de incógnitas do sistema resultante em problemas lineares.
- x_i Coordenadas cartesianas associadas ao sistema de coordenadas global.
- x_j^i Coordenada cartesiana referente à posição do j -ésimo nó de Γ_n ou Ω_r .
- y_i Coordenadas do sistema local de Γ_n ou Ω_r .
- Y** Tensor para transformação de integrais de carregamento de domínio em integrais sobre o contorno.

z	Parâmetro característico das soluções fundamentais de placa; $z = \lambda r$.
w_n	Deslocamento normalizado.
$u_{3_{\max}}$	Deslocamento transversal máximo de uma placa.

Letras gregas:

$\gamma_{\alpha i}$	Componente scisalhantes do tensor deformação de Green.
Γ	Contorno do domínio em estudo.
Γ_n	Elemento de contorno.
Γ_u	Parcela do contorno sobre a qual são prescritos deslocamentos generalizados.
Γ_t	Parcela do contorno sobre a qual são prescritas trações generalizadas.
Γ'_1	Circunferência de raio unitário centrada no ponto fonte.
δ_{ij}	Delta de Kronecker.
$\delta(P, Q)$	Função delta de Dirac.
ε_{ij}	Componentes normais do tensor deformação de Green.
ϵ, ε	Raios genéricos.
λ	Fator de carga. Constante de Lamé; $\lambda = \frac{\nu E}{(1+\nu)(1-2\nu)}$.
λ^2	Parâmetro característico dos modelos de placa estudados; $\lambda^2 = \frac{12\kappa^2}{h^2}$.
κ^2	Fator de correção das tensões cisalhantes.
κ_t	Fator de concentração de tensões.

η	Imagem do ponto singular no domínio normalizado.
ν	Coefficiente de Poisson.
ξ_i	Coordenadas do sistema normalizado.
Π	Funcional genérico. Energia potencial total.
ρ	Fator de relação. Imagem de r no domínio normalizado.
σ_{ij}	Componentes normais do tensor tensão de Cauchy.
$\hat{\sigma}_{\alpha\beta}$	Tensões cisalhantes corrigidas.
ϕ_j	Função de interpolação associada às variáveis físicas referente ao j -ésimo nó de Γ_n ou Ω_r .
$\bar{\phi}_j$	Função de interpolação associada à geometria referente ao j -ésimo nó de Γ_n ou Ω_r .
φ_{kl}^j	Matriz de funções de interpolação associadas às variáveis físicas referente ao j -ésimo nó de Γ_n ou Ω_r .
Φ	Matriz das funções de interpolação de um elemento de contorno ou célula de domínio.
$\psi_{\alpha\beta}$	Componentes de deslocamentos rotacionais generalizados de placa.
Ω	Domínio em estudo.
Ω_r	Célula de domínio.

Símbolos matemáticos:

$ x $	Módulo da variável x .
$\ \mathbf{x}\ $	Norma do vetor \mathbf{x} .
$\ \mathbf{X}\ $	Determinante da matriz \mathbf{X} .
$(\cdot)_{,\alpha}$	Indica diferenciação em relação a x_α ; $\frac{\partial(\cdot)}{\partial x_\alpha}$.
$(\cdot)^*$	Variável que a variável (\cdot) se refere à um estado de equilíbrio auxiliar.
$\overline{(\cdot)}, \widehat{(\cdot)}$	Indicam derivada primeira de um tensor ou vetor (\cdot) .
$\overline{\overline{(\cdot)}}$	Indica derivada primeira de um tensor ou vetor (\cdot) .
$d(\cdot)$	Operador diferencial ordinário aplicado sobre a variável (\cdot) .
$\mathcal{D}(\cdot)$	Operador traço generalizado de Dirichlet.
$\mathcal{N}(\cdot)$	Operador traço generalizado de Neumann.
\mathcal{R}	Resíduo de uma solução aproximada em relação à solução exata.
$\mathcal{H}, \mathcal{H}'$	Espaço de Hilbert e seu dual, respectivamente.
$\mathcal{F}_{ij}(\partial_Q)$	Componentes do operador diferencial da excitação aplicado às coordenadas do ponto Q .
$\mathcal{L}_{ij}(\partial_Q)$	Componentes do operador diferencial em relação às coordenadas do ponto Q .
$\delta(\cdot)$	Operador variacional aplicado sobre a variável (\cdot) .
$\partial(\cdot)$	Operador diferencial parcial aplicado sobre a variável (\cdot) .
$\Delta(\cdot)$	Operador de Laplace; $\Delta(\cdot) = \nabla^2(\cdot) = \frac{\partial^2(\cdot)}{\partial x_\alpha \partial x_\alpha}$.
$\Delta^2(\cdot)$	Operador bi-harmônico; $\Delta^2(\cdot) = \Delta\Delta(\cdot)$.

Capítulo 1

Introdução

1.1 Considerações preliminares

Quando um engenheiro, um matemático ou um físico se depara com uma equação diferencial a ser resolvida, ele dispõe, atualmente, de diversas alternativas de solução, exatas ou aproximadas, analíticas ou numéricas. Uma destas alternativas é o método dos elementos de contorno (MEC), um método de solução aproximado, e que parte das equações do problema escritas na forma integral.

O MEC teve suas raízes matemáticas estabelecidas ainda no final do século XIX, mas apenas no final da década de 1960 passou a receber maior atenção por parte da comunidade científica. Após intensa contribuição dos matemáticos soviéticos na primeira metade do século XX, quando os métodos integrais já eram utilizados para solucionar problemas do meio contínuo, sua utilização praticamente estagnou. Isto se deveu a dois motivos: sua formulação puramente matemática, distanciando-o do campo da engenharia, e o gigantesco desenvolvimento que ocorreu com o método dos elementos finitos (MEF) nas décadas de sessenta e setenta que, por sua abrangência, atraiu pesquisadores das mais diversas áreas, em todo o mundo.

Apenas na década de setenta, incorporando algumas características do MEF, o MEC tomou impulso e consolidou-se como uma alternativa viável para solução de problemas de engenharia. A principal característica do MEC reside na redução - em se tratando de problemas lineares - de uma dimensão do problema, o que diminui drasticamente a quantidade de dados necessários para solução computacional do mesmo. Em contrapartida,

resulta geralmente matrizes cheias e não-simétricas, impedindo ou dificultando o uso de estratégias de solução desenvolvidas para o MEF.

Inicialmente, o MEC apresentava duas grandes dificuldades: a primeira diz respeito à integrais contendo núcleos singulares, que obrigam a utilização de procedimentos de integração numérica especiais, e a segunda reside na necessidade do conhecimento de uma solução fundamental, o que nem sempre ocorre, dependendo da complexidade do problema. A primeira está atualmente superada graças ao amplo esforço dispendido nos últimos anos, resultando no surgimento de diversas quadraturas numéricas, para diferentes graus de singularidade. Quanto à segunda dificuldade, desenvolvimentos mais recentes da matemática têm levado a soluções fundamentais de problemas mais complexos, haja visto várias novas soluções fundamentais surgidas nos últimos anos, para algumas áreas da mecânica computacional.

No âmbito da análise estrutural, interesse do presente trabalho, vale dizer que o MEC tem um ótimo desempenho na solução de problemas lineares, levando inclusive a resultados melhores que o MEF, para discretizações equivalentes do domínio. Entretanto, para problemas não-lineares, o MEC ainda não experimentou sua aplicação para uma ampla variedade de equações governantes, algumas destas já resolvidas satisfatoriamente através de outros métodos. A aplicação e validação do método para esses problemas é objeto de intensa pesquisa em toda a comunidade científica mundial e disso depende, de certa forma, a aplicação geral do método como ocorreu com o MEF. Dada a marcante precisão do MEC na solução de problemas lineares, é natural que haja a expectativa de bom desempenho deste método também em problemas não-lineares, como aliás já foi demonstrado em diversas publicações da literatura. Como nestes casos o MEC normalmente também exige uma discretização do interior do domínio sob análise, poder-se-ia argumentar que o método perderia em competitividade com outros métodos consolidados. Mas a capacidade intrínseca do MEC de fornecer soluções coerentes com as equações da Elasticidade, em particular valores duais em torno de cantos ou singularidades, dentre outras vantagens, podem e devem ser aproveitadas.

Além disso, as tendências mais modernas de elementos finitos determinam claramente o uso de filosofias adaptativas, a fim de reduzir de forma automática o erro da análise. A utilização de procedimentos adaptativos no MEC ainda se encontra em desenvolvimen-

to, dependendo portanto de resultados para taxas de convergência e do comportamento assintótico de elementos de alta ordem (ainda muito escassos na literatura), para vários problemas práticos.

Um outro argumento muito explorado atualmente, e não apenas no âmbito da Mecânica dos Sólidos, está relacionado com o desenvolvimento de programas de computador para engenharia. Durante as décadas de sessenta, setenta e oitenta prevaleceu uma filosofia baseada na chamada programação estruturada. Com a crescente exigência de atualização destes programas, a fim de capacitá-los para solução de outros tipos de problemas, criou-se uma demanda por extensibilidade e reusabilidade dos códigos que as linguagens convencionais de programação não suportavam. Tal demanda foi percebida por companhias que desenvolviam programas comerciais e por centros de pesquisa acadêmicos simultaneamente, levando a uma rápida disseminação da programação orientada a objetos (POO). A POO forneceu uma solução eficiente e naturalmente adaptada para códigos de engenharia. Os programas baseados no MEF e desenvolvidos sob esta nova filosofia foram os primeiros a aproveitar as vantagens da POO. Como será visto no Capítulo 6, um grande número de propostas de arquiteturas para programas de elementos finitos tem surgido nos últimos anos, gerando uma infinidade de concepções novas que criam um paralelo sem precedentes entre a implementação numérica e a estrutura teórica do método. Infelizmente, desenvolvimentos similares voltados para elementos de contorno ainda são muito escassos e incompletos.

São os argumentos gerais expostos acima que justificam, em parte, o presente trabalho.

1.2 Revisão bibliográfica

Com vistas ao objetivo principal deste trabalho, não serão aqui revisadas as literaturas referentes à utilização do MEC para flexão linear estática de placas finas. Informações a este respeito podem ser encontradas nas excelentes revisões de Monken e Silva & de Barcellos (1985), Westphal Jr. (1990) e Beskos (1991). Aqui serão abordadas as literaturas referentes à aplicação do MEC para problemas de flexão não-linear de placas (e problemas correlatos), finas ou semi-espessas, bem como problemas de flambagem, pós-flambagem, e uso de subregiões em placas. Também será dada especial atenção à aplicação do MEC para

solução de problemas de flexão linear utilizando as teorias de R. D. Mindlin (Mindlin 1951) e E. Reissner, referências ainda relativamente escassas na literatura. As revisões bibliográficas sobre tópicos mais específicos deste trabalho (os modelos de placa de Mindlin e de Reissner; métodos de integração numérica para integrais contendo núcleos singulares; aplicação de programação orientada a objetos em elementos finitos e elementos de contorno) estão incluídas em seus respectivos capítulos.

Provavelmente, a primeira aplicação de métodos integrais para solução do problema de flexão de placas se deveu a Muskhelishvili em 1933, na primeira edição de seu livro clássico (Muskhelishvili 1963), utilizando o Teorema de Goursat para operadores biarmônicos. A literatura entretanto atribui a Jaswon e Maiti, em 1968 (Jaswon et al. 1967, Jaswon & Maiti 1968), a primeira formulação de elementos de contorno para solução do problema de flexão de placas finas.

Uma das primeiras aplicações do MEC a problemas de elasticidade envolvendo grandes deslocamentos foi publicada no final da década de setenta (Kompis 1978). Mas apenas em 1982 foi publicado o que parece ser o primeiro artigo que trata da aplicação do MEC para flexão geometricamente não-linear de placas (Tanaka 1982). Utilizando a versão direta do MEC e partindo da identidade de Rayleigh-Green, foi obtida uma formulação para as equações de von Kármán do modelo de placa de Kirchhoff. Infelizmente, nenhum resultado numérico foi apresentado.

Na mesma época, Kamiya e Sawaki publicaram dois artigos sobre este mesmo assunto. No primeiro (Kamiya & Sawaki 1982*b*) foram analisados casos de flexão não-linear que podem ser reduzidos à equação de Berger¹. No segundo (Kamiya & Sawaki 1982*c*) foi apresentada uma formulação baseada no método dos resíduos ponderados (MRP) correspondente às equações de von Kármán. Apesar de proporem uma solução iterativa, não foram mostrados resultados numéricos. Logo após, estes autores mostraram a aplicabilidade do MEC para solução de problemas de placas finas carregadas termicamente, envolvendo grandes deslocamentos (Kamiya & Sawaki 1982*a*). Foi utilizada uma solução aproximada baseada também na equação de Berger. Foram apresentados os resultados numéricos para um caso de placa circular engastada.

Nesse mesmo ano, Gospodinov e Ljutskanov publicaram um interessante artigo onde

¹A equação de Berger é uma forma simplificada das equações de von Kármán, onde o segundo invariante do tensor deformação é desprezado, o que permite o desacoplamento das equações.

foi analisada flexão linear de placas finas pelo MEC direto e vibrações e flambagem pelo MEC indireto (Gospodinov & Ljutskanov 1982). Este artigo é relevante por utilizar, pela primeira vez, soluções fundamentais particulares para vibrações ou flambagem. Foram analisadas apenas placas retangulares, sendo o contorno e o domínio discretizados por elementos e células constantes. Nos exemplos de flambagem foram consideradas placas carregadas igualmente nas direções x e y , com bons resultados.

Ainda em 1982 surgiram dois trabalhos relacionados à aplicação do MEC para flambagem. Tai et al. (1982) mostraram como calcular cargas de flambagem em placas utilizando métodos integrais a partir da medição experimental das deformações de membrana. Niwa et al. (1982) publicaram um excelente compêndio de formulações diretas e indiretas que levam a problemas de autovalores e autovetores, e apresentaram exemplos numéricos para equação de Helmholtz, elastodinâmica e vibrações harmônicas de placas finas.

Foi no mesmo ano de 1982 que F. van der Weeën publicou seus dois trabalhos pioneiros sobre aplicação do MEC para análise linear de placas usando o modelo de Reissner (van der Weeën 1982a, van der Weeën 1982b), condensando resultados de sua tese de doutorado (van der Weeën 1981). As equações integrais do problema foram obtidas via MRP, e a versão direta do MEC foi adotada na discretização das mesmas, utilizando elementos isoparamétricos quadráticos. A solução fundamental do problema foi obtida através do método de Hörmander (Hörmander 1964). A implementação numérica contou com vários aprimoramentos para a época, entre eles a utilização de imposição de movimentos de corpo rígido para cálculo das integrais singulares, transformação de integrais de carregamento para o contorno e utilização de subregiões, entre outros (van der Weeën 1982b, Monken e Silva & de Barcellos 1985). Para tratamento numérico dos vértices, as trações foram calculadas em um dos lados e então utilizadas para cálculos no lado adjacente. Os resultados para o esforço cortante sobre o contorno foram ruins em alguns casos analisados.

Em 1984 Tanaka desenvolveu uma formulação incremental para análise de grandes deslocamentos em placas finas (Tanaka 1984). Foi utilizado o MEC direto, com elementos de contorno constantes e células de domínio triangulares constantes. Ao contrário do seu trabalho anterior (Tanaka 1982), partiu-se diretamente das equações de von Kármán, originando uma formulação mais genérica, mas apenas um caso de placa circular fracamente não-linear foi resolvido.

A seguir, Ye & Liu (1985) apresentaram uma formulação similar à citada acima, mas utilizando células de domínio quadrangulares. Um fator de relaxação foi introduzido no cálculo numérico para acelerar o processo iterativo, o que lhes permitiu um levantamento mais completo da curva carga *versus* deslocamento. Nesse ano Bezzine et al. (1985) publicaram um interessante trabalho resolvendo flambagem de placas finas sobre fundações elásticas. A complexidade do problema é evidenciada pelo acoplamento dos dois fenômenos clássicos (flambagem e fundações elásticas), caracterizando na realidade um problema similar ao de contato de placas.

Durante 1984 também foi publicada o que parece ser uma das primeiras aplicações do MEC à flexão não-linear de placas sanduíche (Rao & Valsarajan 1986). Após uma adimensionalização das equações para grandes deslocamentos de placas sanduíche retangulares e rômbricas, foi possível utilizar uma função de Green associada às derivadas de quarta ordem do deslocamento transversal em relação às variáveis adimensionais mas, aparentemente, a metodologia só é aplicável a placas quadrangulares.

A seguir, Costa Jr. & Brebbia (1985*b*) mostraram a aplicabilidade do MEC em problemas de flambagem de placas finas retangulares ilustrando diversos casos de carregamento e condições de contorno. Foi utilizada a versão direta do método e deduzida uma solução fundamental para o operador diferencial de flambagem (Costa Jr. 1985). A discretização do contorno e do domínio foi realizada com elementos constantes e os resultados foram bons, em geral. Foi mostrada graficamente a convergência do método para placa engastada, mas foi comparada com um elemento finito de baixa taxa de convergência. Na mesma época estes autores desenvolveram uma formulação do MEC direto aplicado à flexão de placas finas sobre fundações elásticas (Costa Jr. & Brebbia 1985*a*). Foi obtida uma solução fundamental para o problema de modo similar ao apresentado no artigo anterior (Costa Jr. & Brebbia 1985*b*). Os núcleos singulares foram isolados e integrados analiticamente, e alguns resultados foram mostrados.

Nesse período Kamiya & Sawaki (1985) mostraram uma alternativa para transformar termos não-homogêneos de uma equação diferencial biharmônica (termos de carregamento, em placas finas) expandindo tais termos em séries de potências e aplicando o Teorema de Gauss. Um caso de flexão foi resolvido com essa formulação mostrando bons resultados. Este trabalho é citado aqui por inspirar a adoção de um procedimento previamente

empregado para integração de termos de acoplamento flexão-membrana em problemas de estabilidade elástica (Marczak 1996).

No ano seguinte Kamiya & Sawaki (1986) estenderam seus trabalhos anteriores para flexão não-linear de placas sanduíche. Foi utilizado o modelo de placa de Kirchhoff que, após simplificações, levou a uma forma da equação de Berger (Kamiya & Sawaki 1982*b*). A formulação permitia carregamentos mecânicos ou térmicos, e também abrangia cascas rasas. Diversos exemplos foram ilustrados. Na mesma época, Tosaka & Miyake (1986) obtiveram uma formulação integral para cascas esféricas rasas com grandes deslocamentos aplicando o MRP. Foi derivada uma solução fundamental do problema e os exemplos numéricos ilustravam ser possível detectar bifurcações sobre a trajetória de equilíbrio.

A seguir, Manolis e seus colaboradores (Manolis et al. 1986) aplicaram o MEC direto para problemas de flambagem em vigas longas e placas finas. Este trabalho é interessante porque seus autores compararam o uso de soluções fundamentais correspondentes ao operador de flexão simples e ao operador de flambagem. No segundo caso não é necessária a discretização do domínio como geralmente exige a integração do acoplamento flexão-membrana. Entretanto, poucas aplicações numéricas foram mostradas.

Tanaka (1986) estendeu uma formulação comum em elementos finitos para análise de flambagem de perfis de paredes finas planas utilizando o MEC. A metodologia consistiu em discretizar cada parede e escrever as matrizes correspondentes em relação a um sistema de coordenadas local. Então as matrizes são transformadas para o sistema de coordenadas global e sobrepostas de modo a garantir a continuidade dos deslocamentos nas interfaces. Apenas condições de contorno homogêneas são analisadas, mas o procedimento adotado neste caso interessa diretamente ao presente trabalho por delinear o tratamento de sub-regiões em problemas de placas resolvidos através do MEC. Este assunto será abordado em maiores detalhes no Capítulo 4. No mesmo ano Kamiya & Sawaki (1986) apresentaram uma formulação alternativa do MEC para placas sobre fundações elásticas. Nesse trabalho as integrais correspondentes à influência da fundação foram transformadas para o contorno.

Karam (Karam 1986, Karam & Telles 1986, Karam & Telles 1988) aprimorou o trabalho de van der Weeën (van der Weeën 1982*a*, van der Weeën 1982*b*) utilizando procedimentos de integração mais eficazes e elementos quadráticos descontínuos, obtendo

resultados muito bons. Foram também analisadas placas infinitas.

Em 1987, Costa Jr. desenvolveu um algoritmo para análise de flexão linear simples, flexão sobre apoio elástico, vibrações e flambagem de placas finas em um único programa de computador (Costa Jr. 1987). Foram utilizadas as suas formulações publicadas anteriormente (Costa Jr. & Brebbia 1985*b*, Costa Jr. & Brebbia 1985*a*, Costa Jr. 1986). No mesmo ano, Zhang e Atluri apresentaram uma formulação integral para análise não-linear estática e dinâmica de cascas e denominaram a metodologia de método dos elementos de contorno-domínio, por ser obrigatória a discretização do domínio para integração dos termos não-lineares (Zhang & Atluri 1987).

Ainda em 1987, Barcellos e Monken e Silva aplicaram o MEC direto para flexão linear de placas modeladas com a teoria de Mindlin (de Barcellos & Monken e Silva 1987, Monken e Silva 1988). A formulação integral do problema foi obtida a partir de uma relação de reciprocidade, e os tensores correspondentes ao estado fundamental auxiliar foram obtidos pelo método de Hörmander (Hörmander 1964). Foram resolvidos problemas de placas circulares e quadradas engastadas onde constatou-se que o fenômeno do *locking* não ocorria (Monken e Silva 1988, Westphal Jr. 1990).

Bezine (1988) modificou algumas formulações anteriores para flexão de placas finas sobre apoio elástico. As modificações na formulação parametrizavam as matrizes resultantes em relação à rigidez da fundação, eliminando novos cálculos quando esta fosse alterada. Permitia também a análise de fundações cujas rigidezes não eram constantes (mais tarde, Calderón & Venturini (1992) publicaram um trabalho similar onde a influência da fundação é transformada para o contorno da placa). Ohga & Shigematsu (1988) aplicaram a idéia de divisão da placa em subregiões para analisar placas com rigidez variável. As condições de compatibilidade foram impostas através de matrizes de transferência. Os resultados para casos de placas nos quais a espessura varia continuamente ficam comprometidos com a abordagem adotada, pois a espessura da placa é considerada constante em cada subregião.

Somente em 1990, nove anos após o trabalho inicial de van der Weeën (van der Weeën 1981), foi publicada a primeira formulação do MEC para flexão geometricamente não-linear de placas semi-espessas (Xiao-Yan et al. 1990). Os autores denominaram erroneamente o modelo de placa utilizado de modelo de Reissner, quando as expressões para

momentos mostram que se trata efetivamente do modelo de Mindlin. Foram analisados casos de placas circulares e retangulares sob carregamentos transversais, e os resultados foram comparados com soluções do MEF. A solução iterativa adota a mesma estratégia que Ye & Liu (1985).

Outro trabalho de grande interesse que foi publicado nesse ano foi a tese de Vilmann (1990), cujo conteúdo mantém um grande paralelo com o trabalho aqui desenvolvido. A solução fundamental do problema foi obtida através da aplicação de forças e momentos concentrados no centro de uma placa circular, e as expressões resultantes são comparadas com outras soluções fundamentais. Vale notar que além do trabalho de van der Weeën (1981), são feitas referências a outras formas de solução fundamental publicadas nos anos de 1966, 1975 e 1984 (Vilmann 1990). A formulação utiliza o MEC direto baseado no teorema de reciprocidade de Betti-Maxwell para obter as identidades de Somigliana. A discretização contou com elementos de contorno e células de domínio cúbicas. Embora problemas de flexão linear tenham sido bem explorados, foram apresentados resultados para apenas dois casos de grandes deslocamentos. O ponto fundamental deste trabalho está no fato dos termos referentes à grandes deslocamentos terem sido adicionados *a posteriori* nas relações de Betti-Maxwell, limitando o número de problemas que pode ser resolvido com a formulação proposta. Outro aspecto a ser destacado naquele trabalho se refere ao fato de se ter utilizado derivadas das funções de interpolação para o cálculo das derivadas dos deslocamentos, procedimento este que enfraquece a solução.

Westphal Jr. (1990) apresentou uma formulação integral para flexão linear estática de placas que unificava numericamente os modelos de Mindlin e Reissner. Foram também deduzidos todos os tensores fundamentais na sua forma mais geral, com a solução fundamental para os deslocamentos calculada através do método de Hörmander. Foram utilizados elementos constantes, lineares e quadráticos para resolver principalmente casos de placas circulares. No mesmo período Westphal Jr. & de Barcellos (1990) reestudaram a influência numérica das funções livres contidas nas soluções fundamentais utilizadas e demonstraram através de experimentos numéricos a arbitrariedade das mesmas.

Shi (1990) desenvolveu o que parece ser a primeira formulação integral para flambagem de placas ortotrópicas. A formulação abrangia também o problema de vibrações, mas o modelo de placa empregado foi o de Kirchhoff. Vários exemplos foram resolvidos com

resultados satisfatórios.

Em 1991 foi editado um compêndio com diversos trabalhos relacionados à solução de problemas de placas e cascas através do MEC (Beskos 1991). Dois artigos se destacam no contexto do presente trabalho: Atluri & Pipkins (1991), ilustrando problemas de grandes deslocamentos e pós-flambagem estáticos e dinâmicos. Foram utilizadas as equações para cascas finas, embora o texto afirme o contrário. As equações integrais estão escritas na forma incremental, o que não é muito comum em se tratando do MEC. O outro artigo (Antes 1991) aborda análise estática e dinâmica de placas de Mindlin/Reissner, onde foram resolvidos problemas estáticos utilizando a solução fundamental proposta por van der Weeën e problemas dinâmicos através de uma solução fundamental especialmente desenvolvida.

Ainda naquele ano, Wang et al. (1991), uma formulação integral para vibrações livres de placas finas com grandes deslocamentos. Utilizando um método de balanço harmônico, obtiveram um problema de autovalores e autovetores dependente da amplitude do movimento. Syngellakis e seu colaboradores publicaram neste ano um interessante artigo comparando resultados experimentais de pós-flambagem de placas com resultados numéricos obtidos via MEC (Syngellakis et al. 1991). A formulação numérica utilizada é caracterizada por retirar as curvaturas do sistema de equações resultante, de forma que apenas os deslocamentos transversais são tratados como variáveis primárias. Por outro lado, este procedimento é de aplicação direta apenas para placas finas.

Ribeiro & Venturini (1991*b*) estenderam a formulação integral apresentada em Karam (1986) e Karam & Telles (1988) para considerarem cargas distribuídas em linhas sobre o domínio da placa. No mesmo ano, esses autores adicionaram aos termos de carregamento a ocorrência de momentos iniciais no domínio, permitindo a consideração dos efeitos de temperatura e alguns problemas não-lineares correlatos (Ribeiro & Venturini 1991*a*). Estes trabalhos abordam tópicos muito interessantes usualmente ignorados na maioria das implementações do MEC para placas, mas indispensáveis para desenvolvimento de códigos de uso geral.

Em 1992, de Barcellos e Westphal Jr. voltaram a investigar os tensores fundamentais para placas de Mindlin e Reissner (de Barcellos & Westphal Jr. 1992), e mostraram como particularizar tais tensores para obter uma solução fundamental de placas finas. Esse

resultado é obtido mediante escolha apropriada das constantes livres. Foram resolvidos alguns exemplos com elementos constantes, lineares e quadráticos, cujos resultados são comparados com soluções analíticas.

Yuying et al. (1992) apresentaram uma formulação incremental para flambagem de placas finas sobre fundações elásticas. As trações são eliminadas das equações, simplificando a implementação desta abordagem. Sapountzakis & Katsikadelis (1992) publicaram um trabalho similar, onde a fundação não precisa ser necessariamente plana. Trabalhos nesta linha, mas utilizando modelos de placas semi-espessas foram publicados por Jianguo et al. (1993) e Qin (1993). Ambos deduziram uma solução fundamental para o problema, e estas se assemelham muito. Nos dois trabalhos as teorias de placa utilizadas são erroneamente atribuídas a Reissner, quando as expressões para momentos mostram que se trata de placa de Mindlin. Uma outra solução fundamental para este problema pode ser encontrada no trabalho de El-Zafrany et al. (1994).

Uma forma de lidar com termos não-lineares em placas finas foi desenvolvida por Katsikadelis & Nerantzaki (1993). O método, chamado de *método da equação análoga*, consiste em decompor as equações de von Karmán em duas equações de flexão linear de placas sujeitas a cargas fictícias. Utilizando então equações auxiliares para os esforços e para as curvaturas, pode-se resolver um sistema não-linear para determinação destas cargas fictícias. Este método foi depois utilizado para resolver problemas de vibrações e flambagem de placas com espessura variável (Nerantzaki & Katsikadelis 1996a, Nerantzaki & Katsikadelis 1996b). Posteriormente, Chaves et al. (1999) e Fernandes & Venturini (1999) aplicaram o MEC para flexão linear de placas finas com espessura variável usando uma abordagem diferente, onde os valores da rigidez à flexão em pontos internos são escritos em função de um valor de referência. Esta formulação permite assim tanto a análise de placas com variação contínua quanto discreta da espessura.

Uma formulação unificada para análise de estabilidade linear elástica de placas de Mindlin e de Reissner foi proposta pelo autor em 1993 (Marczak 1993, Marczak & de Barcellos 1993, Marczak 1994, Marczak 1995a). A formulação matricial é muito similar a outras adotadas para flambagem de placas finas (Costa Jr. & Brebbia 1985b, Costa Jr. & Brebbia 1985a, Tanaka 1986, Syngellakis & Elsein 1994). Toumi & Jezequel (1993) mostraram uma possibilidade de analisar vibrações em placas reforçadas através do uso

de subregiões. Uma técnica similar pode ser utilizada para problemas de flambagem de placas semi-espessas compostas por elementos de espessuras diversas (Marczak 1995*b*). Estes procedimentos serão também explorados e estendidos no presente trabalho.

Uma outra abordagem para a solução de problemas de placas sob grandes deslocamentos foi proposta por Ye (1994), onde as imperfeições na planicidade da placa são levadas em conta. O problema de membrana é tratado através do MEF e então acoplado com as equações do MEC através de um procedimento iterativo, com as tensões de membrana desempenhando um papel de tensões residuais.

Os desenvolvimentos do MEC aplicado a problemas não-lineares de placas finas ainda continuam intensos, como demonstrado por Tanaka et al. (1996), que obteve uma formulação regularizada para problemas de grandes deslocamentos em placas finas, eliminando assim a necessidade de procedimentos especiais de integração. Estes autores também obtiveram sucesso em resolver problemas de pós-flambagem com alto grau de não-linearidade via MEC empregando uma variação dos métodos de comprimento de arco tipicamente utilizados no MEF (Tanaka et al. 1999). Lin et al. (1999) modificaram formulações anteriores para problemas de flambagem (Costa Jr. & Brebbia 1985*b*, Costa Jr. & Brebbia 1985*a*, Tanaka 1986, Syngellakis & Elsein 1994) fim de eliminar a necessidade de discretização interna. Isto foi realizado através de procedimentos de reciprocidade dual (Partridge et al. 1992) aplicados aos termos não-lineares. Duarte & Palermo Jr. (1999) reescreveram alguns trabalhos de flambagem de placas finas via MEC (Costa Jr. 1985) utilizando aprimoramentos como integração analítica e elementos descontínuos.

O tratamento analítico e numérico de soluções fundamentais também constitui um atalho para solução de problemas específicos do MEC aplicado a placas semi-espessas. Xiao-Yan (1995) apresentou uma solução fundamental modificada para flexão de placas de Reissner considerando o tensor rotação infinitesimal. Disto resultou uma formulação do MEC que calcula todas as componentes das trações sobre o contorno da placa, sem a necessidade de derivadas tangenciais numéricas. Rashed e colaboradores também desenvolveram uma abordagem para cálculo direto das trações sobre o contorno, utilizando a teoria de Reissner (Rashed et al. 1997, Rashed et al. 1998*b*). O procedimento resulta uma formulação hipersingular, e as integrais com singularidades de Cauchy são tratadas através de expansão dos integrandos em séries de Taylor com posterior integração analíti-

ca, enquanto as integrais com singularidades mais fortes são calculadas com imposição de movimentos de corpo rígido. Estes trabalhos foram posteriormente estendidos para solução de placas sobre apoios elásticos (Rashed et al. 1998a, Rashed et al. 1999).

Um outro trabalho interessante nesta linha foi publicado por El-Zafrany et al. (1995), onde os efeitos das deformações cisalhantes advindos das equações do modelo de Reissner são acrescentados separadamente. Isto resulta uma solução fundamental dependente apenas de r , ao invés de λr , como é usual² (van der Weeën 1982a, van der Weeën 1982b, Westphal Jr. & de Barcellos 1990). Desta forma, um mesmo código numérico pode ser utilizado tanto para placas excepcionalmente finas quanto semi-espessas, sem os problemas numéricos normalmente originados por valores muito grandes de λ .

Westphal Jr. et al. (1998) voltaram a investigar as soluções fundamentais para os modelos de placa de Mindlin e de Reissner, bem como sua relação com a solução fundamental do modelo de Kirchhoff. Mais tarde, demonstraram ser rigorosamente possível separar a contribuição da influência do cisalhamento nas soluções fundamentais de Mindlin e de Reissner Westphal Jr. et al. (1999). Isto abre uma possibilidade interessante de aplicação do MEC onde, caso necessário, esta influência é incluída *a posteriori* em uma análise de placas inicialmente finas.

A aplicação do MEC para solução de cascas rasas também tem sido objeto de intensa pesquisa, muito embora as opiniões quanto a esta aplicação do método sejam bastante contraditórias. Por se tratar de um problema essencialmente tridimensional, este não pode ser escrito apenas por seu problema de valores no contorno associado sem maiores informações sobre a geometria da casca no interior do domínio. Isto claramente torna a discretização interna obrigatória. Adicionalmente, como o acoplamento flexão-membrana se torna mais severo à medida que o raio de curvatura da casca é reduzido, a grande maioria dos trabalhos nesta área se limitam a cascas rasas e de curvatura constante (cascas esféricas e cilíndricas, principalmente). Exemplos destes são os trabalhos de Tottenham (1979), Kamiya & Sawaki (1986), Tosaka & Miyake (1986), Zhang & Atluri (1986), Zhang & Atluri (1987), Simos & Sadegh (1989), Fu & Harb (1990), Ren & Fu (1991) e Ren & Fu (1998). No âmbito de placas/cascas semi-espessas ortotrópicas, Wang & Schweizerhof têm desenvolvido uma série de trabalhos sobre problemas estáticos e dinâmicos que parecem

²A variável λ é um parâmetro característico dos modelos de placa semi-espessas, enquanto r é a distância do ponto fonte ao ponto campo (ver Simbologia).

promissores Wang & Schweizerhof (1995, 1996a, 1996b, 1996c, 1996d, 1997). Alguns trabalhos, ainda, têm obtido sucesso na aplicação do MEC para problemas de cascas semi-espessas (Jinmu & Shuyao 1996, Dirgantara & Aliabadi 1999).

Outras referências complementares podem ser encontradas em Beskos (1991), para vários tipos de problemas envolvendo placas ou cascas. Adicionalmente, Aliabadi et al. (1991, pp.134-137), traz um levantamento parcial do estado da arte em problemas geometricamente não-lineares resolvidos pelo MEC, entre os anos de 1978 e 1991.

1.3 Objetivos do trabalho

Os modelos de placa propostos por R. D. Mindlin (Mindlin 1951) e E. Reissner (Reissner 1944, Reissner 1945) são adequados para análise numérica de flexão de placas semi-espessas porque representam de modo mais realista o fenômeno real, em relação ao modelo clássico de Kirchhoff. Além disso, tais modelos permitem também a solução de problemas de flexão de placas finas e não implicam aumento do custo computacional em comparação com o modelo clássico. Apesar de existirem modelos de placa de ordens ainda mais altas (a literatura ilustra modelos de 10^a até 22^a ordem), e sua aplicação ser necessária para a correta solução de certos fenômenos, a complexidade das equações destes modelos torna a utilização das equações da elasticidade tridimensional uma opção mais direta.

O uso dos modelos de placa de Mindlin e de Reissner no MEF já está consolidado. No entanto, muitos dos elementos finitos baseados em tais teorias possuem a inconveniente necessidade de manipulação de modos de deformação espúrios, decorrentes da eliminação do fenômeno do *locking* – travamento (Hughes 1987). Esta patologia tornou comum o emprego de campos de deformação ou tensão assumidos em elementos finitos modernos que, embora muito eficientes, ainda apresentam efeitos colaterais em alguns tipos de problemas específicos. Em contrapartida, torna-se viável a utilização de funções de interpolação C^0 devido ao desacoplamento dos graus de liberdade translacionais dos rotacionais.

Por outro lado, a revisão anterior mostra uma grande evolução no número de trabalhos que tratam da aplicação do MEC à solução de problemas de flexão linear de placas, considerando o efeito do cisalhamento. Nestes casos, a ausência do *locking* (Monken e Silva 1988, Westphal Jr. 1990, Marczak 1993) é uma característica particularmente

atrativa, pois permite tanto a análise de placas moderadamente espessas quanto placas finas sem maiores problemas, do ponto de vista numérico. E apesar dos modelos de Mindlin e Reissner terem sido inseridos no contexto dos problemas geometricamente não-lineares já no final da década de setenta, para elementos finitos (Pica & Wood 1980, Pica et al. 1980), não é grande o número de publicações correspondentes usando elementos de contorno.

Outro assunto ainda escasso na literatura é a aplicação de técnicas de programação orientada a objeto para o MEC, ao contrário do que ocorre com o MEF. A fim de evitar a perda de continuidade do texto, a revisão bibliográfica deste tópico será apresentada no Capítulo 6, e ela evidencia a carência de desenvolvimentos neste campo.

Assim, o presente trabalho se insere na mecânica computacional com dois objetivos básicos, que contribuem para as áreas de estudo destacadas acima:

- Estender e formalizar a aplicação do MEC para problemas de flexão de placas com grandes deslocamentos: Este tópico é abordado através do desenvolvimento das equações integrais correspondentes via inclusão dos termos não-lineares dos operadores diferenciais, tanto para o problema de flexão quanto para o problema de membrana. As equações integrais são resolvidas numericamente através da metodologia padrão do MEC direto, utilizando elementos de contorno descontínuos. A modelagem numérica de múltiplas subregiões, com propriedades geométricas e materiais distintos, será considerada pelo procedimento numérico. Devido aos diversos graus de singularidade dos tensores envolvidos, um novo procedimento de integração é testado como alternativa à tradicional imposição de movimentos de corpo rígido, buscando economia e eficiência no procedimento numérico. No cálculo dos termos não-lineares, correspondentes ao acoplamento flexão-membrana, não foram utilizadas derivadas das funções de interpolação. Em vez disso são desenvolvidas equações integrais auxiliares para cálculo das derivadas dos deslocamentos translacionais caracterizando, em problemas de grandes deslocamentos apenas, uma formulação hipersingular. Também são testados esquemas de diferenças finitas para cálculo destas derivadas. E a fim de evitar quadraturas para núcleos hipersingulares, a solução numérica conta com células de domínio descontínuas. O desempenho da formulação proposta será mostrado através da solução de alguns casos típicos de

flexão linear e não-linear de placas.

- Implementar um procedimento eficaz para cálculo de integrais contendo núcleos fortemente singulares. Este tópico objetiva fornecer uma alternativa à tradicional técnica de imposição de movimento de corpo rígido. Embora seja uma técnica eficaz, esta abordagem permite sua aplicação apenas a problemas governados por equações diferenciais para as quais se conhece uma determinada solução particular. Além disso, seu uso em problemas de placas não é tão direto quanto em problemas de elasticidade bi ou tridimensional. Não será feito uso de procedimentos de integração analítica dada a limitação que estas técnicas impõem quando da extensão do código para outras soluções fundamentais ou outros tipos de elementos. Assim, o procedimento adotado deverá contar com simplicidade e generalidade suficientes para permitir sua aplicação a códigos gerais baseados no MEC, independentemente da aplicação e dos tipos de elementos empregados.
- Conceber e implementar uma arquitetura orientada a objetos para aplicações baseadas no MEC: este objetivo visa desenvolver uma biblioteca de componentes modulares, extensíveis e reutilizáveis para simplificar a implementação de novas e atuais formulações do método. Dois aspectos básicos determinam sua concepção: (a) aplicações diferentes compartilham uma estrutura matemática e numérica comum e (b) classes de armazenamento não devem realizar etapas de análise propriamente dita. A arquitetura proposta deve fornecer um conjunto de classes de armazenamento auxiliares e um conjunto de classes de análise. Para criar um programa específico, o usuário *monta* o código básico agregando os objetos necessários. O trabalho de programação fica então limitado a implementar as classes de análise (caso não existam ainda) derivando-as das classes-mãe existentes.

1.4 Organização do texto

Este trabalho está estruturado em oito capítulos. O Capítulo 2 detalha os modelos de placa de sexta ordem de Mindlin e de Reissner, incluindo as equações de equilíbrio, condições de contorno e o campo de tensões considerando os termos não-lineares do tensor deformação. Os operadores diferenciais correspondentes são explicitamente apresentados. O

Capítulo 3 desenvolve as equações integrais correspondentes aos modelos empregados utilizando o método dos resíduos ponderados. Particular atenção é dada à obtenção das equações para curvaturas da placa (derivadas do deslocamento transversal) pelo seu caráter hipersingular. No Capítulo 4 a implementação numérica do presente trabalho é detalhada, incluindo o tratamento numérico de problemas com múltiplas regiões, utilizando o formalismo padrão do MEC direto. No Capítulo 5 está resumido o tratamento numérico empregado para cômputo de integrais singulares presentes nas equações integrais. O Capítulo 6 apresenta a proposta de arquitetura orientada a objetos utilizada na implementação numérica. Após uma análise de propostas similares para o MEF, é feita uma descrição sucinta dos conceitos empregados na proposta e detalhadas suas classes mais básicas. No Capítulo 7 são apresentados alguns resultados numéricos obtidos com a formulação proposta. Finalmente, o Capítulo 8 resume as principais conclusões e sugere alguns tópicos para continuidade do presente trabalho.

S

Capítulo 2

Os Modelos de Placa de Mindlin e de Reissner

Neste capítulo são desenvolvidas as equações básicas para flexão de placas utilizando-se os modelos de Mindlin (Mindlin 1951) e Reissner (Reissner 1944, Reissner 1945). No cálculo das componentes do tensor de deformação, foram mantidos alguns termos de alta ordem correspondentes ao gradiente do campo de deslocamentos de modo que, efetivamente, obtém-se uma forma das equações de von Kármán.

São também deduzidas as equações representativas do campo de tensões dos modelos empregados, bem como as equações gerais do problema para um estado de equilíbrio auxiliar.

2.1 Notas históricas

Os modelos mais conhecidos para descrição do fenômeno de flexão de placas são, sem dúvida, o modelo clássico de Kirchhoff-Love (Love 1888, Timoshenko & Woinowski-Krieger 1970) e os de R. D. Mindlin (Mindlin 1951) e E. Reissner (Reissner 1944, Reissner 1945). O primeiro destes, devido às hipóteses sobre as deformações que atuam na direção transversal da placa, leva a uma equação diferencial biharmônica cuja solução obriga a contração de duas das condições de contorno em uma única (Reissner 1944, Timoshenko & Woinowski-Krieger 1970). Os outros dois modelos são geralmente chamados de teorias de primeira ordem para flexão de placas por assumirem uma variação linear dos deslo-

camentos longitudinais da placa ao longo da espessura. Assim, tais modelos consideram deformações cisalhantes constantes ao longo da espessura o que resulta numa equação diferencial de sexta ordem possibilitando então a satisfação de três condições de contorno.

O modelo de placa de Reissner foi introduzido em 1944 (Reissner 1944) mas as equações governantes só foram publicadas na sua forma geral no ano seguinte (Reissner 1945). Nestes artigos, Reissner partiu de uma variação linear das tensões normais longitudinais ($\sigma_{\omega\omega}$) ao longo da espessura da placa e obteve, a partir das equações de equilíbrio, um comportamento quadrático para as componentes de tensão cisalhante transversal ($\sigma_{\alpha 3}$) e cúbico para a tensão normal transversal (σ_{33}). Uma outra forma de se deduzir as equações para este modelo de placa foi publicada pelo mesmo autor em 1947 (Reissner 1947). Apesar de Reissner ter obtido a lei de Hooke generalizada através do teorema de Castigliano, Green (1949) obteve as mesmas equações sem utilizar tal teorema.

O modelo de placa de Mindlin surgiu em 1951 (Mindlin 1951, Mindlin 1955) e, ao contrário do de Reissner, foram adotadas hipóteses sobre o campo de deslocamentos experimentados pela placa, o que permite uma dedução bem mais direta das equações governantes. Por incluir os termos de inércia nas equações de equilíbrio, o trabalho de Mindlin pode ser visto efetivamente como uma extensão dos trabalhos (pouco conhecidos, na época¹) de Hencky e Bollé, ambos publicados em 1947 (Bollé 1947a, Bollé 1947b, Reissner 1985).

Em 1957, Naghdi modificou o modelo de Reissner considerando carregamentos arbitrários, incluindo o efeito de deformação transversal (Naghdi 1957). Essas equações foram deduzidas para cascas, mediante o uso do Princípio Variacional de Reissner (Reissner 1950). No ano seguinte Goldenveizer apresentou uma generalização da teoria de Reissner para uma variação arbitrária das tensões $\sigma_{\alpha\beta}$ com a espessura (Reissner 1980). Em 1975, Reissner propôs uma modificação da sua teoria baseado em informações advindas de soluções tridimensionais de problemas de flexão de placas (Reissner 1975).

Uma característica inerente de algumas das chamadas teorias de primeira ordem é o fato de levarem a valores constantes para as deformações cisalhantes transversais determinando, nestes modelos de placa, campos de tensões cisalhantes transversais constantes.

¹Na realidade, já no final do século XIX, M. Lévy e A. B. Basset se preocupavam com a necessidade de satisfação de três condições de contorno, motivo pelo qual alguns pesquisadores não denominam as teorias de Mindlin e Reissner pelos nomes de seus autores (Reddy 1984b).

Por outro lado, a elasticidade tridimensional mostra que tais tensões variam ao longo da espessura segundo uma função quadrática ou de ordem mais alta. Esta constatação obriga o emprego de fatores corretivos para o cômputo global dos efeitos destas componentes de tensão. Tipicamente, utiliza-se o fator $\kappa^2 = \pi^2/12$ para o modelo de Mindlin² (o fator $\kappa^2 = 5/6$ aparece naturalmente na dedução das equações do modelo de Reissner). As teorias de ordem superior desenvolvidas nas últimas décadas eliminaram tal inconveniente, por utilizarem campos de deslocamento baseados em polinômios de ordens mais altas, o que leva naturalmente à distribuições mais complexas para as tensões cisalhantes ao longo da espessura da placa (Reddy 1984*b*). Em compensação, são modelos regidos por equações diferenciais de 10^a, 12^a, 14^a ou 22^a ordem (Reissner 1983, Reddy 1984*b*, Reissner 1985) e nem sempre podem ser facilmente reduzidas às equações correspondentes dos modelos de primeira ordem ou às do modelo clássico de Kirchhoff. Em geral, tais teorias confirmam o valor 5/6 para correção das tensões cisalhantes (ver, por exemplo, Voyiadjis & Baluch 1981). Recentemente, alguma ênfase tem sido dada à solução de problemas de flexão de placas através do uso de soluções analíticas da teoria da elasticidade tridimensional (Cheng 1979, Wittrick 1987).

No contexto geometricamente não-linear, as equações para flexão de placas tornam-se um pouco mais complexas, devido à necessidade de se manter alguns termos de ordem superior correspondentes ao gradiente do campo de deslocamentos nas expressões para as componentes de deformação. Esta necessidade provém do abandono da tradicional hipótese segundo a qual os deslocamentos e as rotações são suficientemente pequenos a ponto de se poder ignorar a magnitude dos produtos de derivadas na expressão geral do tensor deformação.

As equações para placas incorporando os termos de grandes deslocamentos são conhecidas como equações de von Kármán (von Kármán 1910), e são indispensáveis para a adequada análise de problemas como flambagem e pós-flambagem. A literatura é extensa para os assuntos relativos ao desenvolvimento e aplicação das equações de von Kármán para placas e cascas baseadas nas hipóteses de Kirchhoff³. No âmbito das equações de

²Se determinado como na formulação original o fator de correção das tensões cisalhantes do modelo de Mindlin é uma função dependente do coeficiente de Poisson do material (Mindlin 1951, Mindlin 1955).

³Teorias como a de W. T. Koiter (Koiter 1960, Koiter 1967) e W. Pietraszkiewicz (Pietraszkiewicz 1979) merecem ser destacadas, em particular a última, que formalmente não impõe qualquer limitação sobre a magnitude dos deslocamentos ou rotações da placa.

von Kármán para modelos de placa ou casca que consideram deformações cisalhantes, a literatura é bem mais escassa (Reddy 1984b), e soluções analíticas somente são possíveis para casos extremamente simples de geometria, condições de contorno e carregamento. Ainda assim, tais soluções são geralmente apresentadas na forma de expansões assintóticas (ver, por exemplo, Frakes & Simmonds 1985).

2.2 Relações cinemáticas

O domínio de análise que será utilizado daqui em diante é definido por um volume material $V = \Omega \times (\pm h/2) \subset \mathfrak{R}^3$, onde $\Omega(x_\alpha)$ é a superfície de referência da placa, aberta e limitada, e com contorno Γ lipschitziano sobre o qual é localmente definido um sistema de coordenadas $\mathbf{n}, \mathbf{t}, \mathbf{s}$, conforme ilustrado na figura 2-1. A espessura h da placa é considerada discretamente uniforme em Ω , e o material que ocupa V é isotrópico linear elástico.

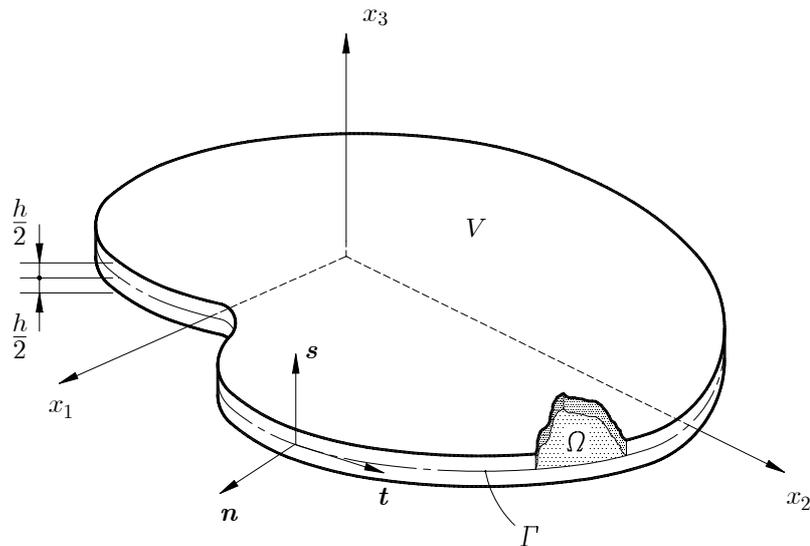


Figura 2-1: Definições geométricas do domínio de uma placa.

Nas equações apresentadas a seguir, os índices latinos i, j, k etc. variam de um até três, enquanto os índices gregos α, β, γ etc. variam de um até dois. A convenção de Einstein é empregada e os índices ω varia de um até dois e não implica em somatório, a menos que indicado.

2.2.1 Campo de deslocamentos

Conforme foi destacado em 2.2, o modelo de Mindlin permite um desenvolvimento mais natural das equações do problema por partir de um campo de deslocamentos assumido. Este será o modelo usado para obtenção das equações diferenciais governantes sendo que, quando pertinente, será indicada a modificação que leva às equações do modelo de Reissner. Uma análise mais geral é possível a partir do campo de deslocamentos postulado pela teoria generalizada de terceira ordem de Reddy (Reddy 1984b), da qual vários modelos de placas podem ser obtidos por particularização. Esse campo de deslocamentos é baseado nos seguintes polinômios:

$$\begin{aligned} U_\alpha(x_1, x_2, x_3) &= u_\alpha + a_1 x_3 \frac{\partial u_3}{\partial x_\alpha} + a_2 x_3 \psi_\alpha + a_3 x_3^2 \phi_\alpha + a_4 x_3^3 \theta_\alpha \\ U_3(x_1, x_2, x_3) &= u_3 + a_5 x_3 \phi_3 + a_6 x_3^3 \theta_3 \quad , \end{aligned} \quad (2.1)$$

com

$$\begin{aligned} u_i &= u_i(x_1, x_2) \\ \psi_\alpha &= \psi_\alpha(x_1, x_2) \end{aligned}$$

sendo os cinco deslocamentos generalizados da superfície de referência da placa (os termos ϕ_i e θ_i correspondem a deslocamentos de ordem superior). Os deslocamentos ψ_α representam a rotação de uma linha inicialmente normal à superfície de referência sobre os eixos x_α . A partir de valores apropriados para os coeficientes a_1, a_2, a_3, a_4, a_5 e a_6 obtém-se os campos de deslocamento de diversos modelos. São apresentados abaixo três exemplos:

- modelo clássico de Kirchhoff : $a_1 = -1, a_2 = a_3 = a_4 = a_5 = a_6 = 0$
- teorias de 1^a ordem: $a_1 = 0, a_2 = 1, a_3 = a_4 = a_5 = a_6 = 0$
- teorias de 2^a ordem: $a_1 = 0, a_2 = a_3 = 1, a_4 = 0$

Nas equações (2.1), ϕ_i e θ_i são funções a serem determinadas. A substituição, nas equações (2.1), das constantes referentes às teorias de 1ª ordem, leva às expressões:

$$\begin{aligned} U_\alpha(x_1, x_2, x_3) &= u_\alpha + x_3 \psi_\alpha \quad , \\ U_3(x_1, x_2, x_3) &= u_3 \quad , \end{aligned} \quad (2.2)$$

que é o campo de deslocamentos efetivamente utilizado neste trabalho.

2.2.2 Relações deformação-deslocamento

A inserção do campo de deslocamentos (2.2) nas componentes do tensor deformação finita (Malvern 1969):

$$\varepsilon_{ij} = \frac{1}{2} (U_{i,j} + U_{j,i} + U_{k,i} U_{k,j}) \quad , \quad (2.3)$$

fornece expressões que podem ser simplificadas da seguinte maneira: nas deformações $\varepsilon_{\alpha\beta}$, pode-se negligenciar os quadrados das derivadas dos deslocamentos longitudinais. A deformação transversal ε_{33} é desprezada, tendo em vista a hipótese de inextensibilidade transversal da placa. Nas expressões para deformações cisalhantes, desprezam-se os termos quadráticos em $U_{\alpha,\beta}$ pelo mesmo motivo. Este procedimento leva a:

$$\begin{aligned} 2\varepsilon_{\alpha\beta} &\doteq U_{\alpha,\beta} + U_{\beta,\alpha} + U_{3,\alpha} U_{3,\beta} \\ \varepsilon_{33} &\doteq 0 \\ 2\varepsilon_{\alpha 3} &= \gamma_{\alpha 3} = \gamma_{3\alpha} \doteq U_{\alpha,3} + U_{3,\alpha} \end{aligned} \quad (2.4)$$

Reescrevendo-se estas expressões para identificar as contribuições de membrana e flexão separadamente, obtém-se:

$$\varepsilon_{\alpha\beta} = E_{\alpha\beta} + x_3 K_{\alpha\beta} \quad (2.5a)$$

$$\varepsilon_{33} = 0 \quad (2.5b)$$

$$\gamma_{\alpha 3} = \psi_\alpha + u_{3,\alpha} \quad (2.5c)$$

onde:

$$E_{\alpha\beta} = \frac{1}{2} \left(u_{\alpha,\beta} + u_{\beta,\alpha} + u_{3,\alpha} u_{3,\beta} \right) \quad (2.6a)$$

$$K_{\alpha\beta} = \frac{1}{2} \left(\psi_{\alpha,\beta} + \psi_{\beta,\alpha} \right) \quad . \quad (2.6b)$$

As relações deformação-deslocamento são similares àquelas do problema de flexão linear de placas, apresentando no entanto, como fator de fundamental diferença, a presença dos termos dependentes das derivadas do deslocamento transversal no cálculo das deformações de membrana $E_{\alpha\beta}$. Estes termos são responsáveis, em parte, pelo acoplamento do problema de membrana com o problema de flexão.

2.2.3 Relações constitutivas

Fazendo-se uso da lei de Hooke generalizada (Malvern 1969),

$$\sigma_{ij} = 2G\varepsilon_{ij} + \lambda\varepsilon_{kk}\delta_{ij} \quad (2.7)$$

obtem-se as equações constitutivas para o problema:

$$\sigma_{\alpha\beta} = 2G \left[E_{\alpha\beta} + x_3 K_{\alpha\beta} + \frac{\nu}{1-\nu} (E_{\gamma\gamma} + x_3 K_{\gamma\gamma}) \delta_{\alpha\beta} \right] \quad (2.8a)$$

$$\sigma_{33} = \frac{\nu E}{(1+\nu)(1-2\nu)} [(E_{11} + E_{22}) + x_3 (K_{11} + K_{22})] \quad (2.8b)$$

$$\sigma_{\alpha 3} = G(\psi_{\alpha} + u_{3,\alpha}) \quad (2.8c)$$

Deve-se destacar que, estritamente falando, a relação dada pela eq.(2.7) não é correta, pois conjuga o tensor deformação de Green (ε_{ij}), que é Lagrangeano, com o tensor tensão de Cauchy (σ_{ij}), que é Euleriano. A definição correta deveria empregar o segundo tensor de Piola-Kirchhoff (S_{ij}), a fim de manter todas as quantidades em uma única descrição. No entanto, da relação $\mathbf{S} = \det \mathbf{F} (\mathbf{F}^{-1} \boldsymbol{\sigma} \mathbf{F}^{-T})$, verifica-se que se a compressibilidade do material é desprezável e as deformações são pequenas, o tensor gradiente de deformações \mathbf{F} se confunde com o tensor identidade \mathbf{I} (Fung 1965). Nestes casos, pode-se empregar σ_{ij} em lugar de S_{ij} sem que a relação constitutiva deixe de ser objetiva. É neste contexto que a eq.(2.7) foi aqui empregada.

Observando-se então as definições de tensões resultantes (tensões de placa), dadas por

$$N_{\alpha\beta} = \int_{-h/2}^{+h/2} \sigma_{\alpha\beta} dx_3 \quad (2.9a)$$

$$M_{\alpha\beta} = \int_{-h/2}^{+h/2} x_3 \sigma_{\alpha\beta} dx_3 \quad (2.9b)$$

$$Q_\alpha = \int_{-h/2}^{+h/2} \sigma_{\alpha 3} dx_3 \quad (2.9c)$$

e substituindo-se as expressões (2.8) nas mesmas resulta, após a integração na espessura:

$$N_{\alpha\beta} = C [(1 - \nu) E_{\alpha\beta} + \nu E_{\gamma\gamma} \delta_{\alpha\beta}] \quad (2.10a)$$

$$M_{\alpha\beta} = D [(1 - \nu) K_{\alpha\beta} + \nu K_{\gamma\gamma} \delta_{\alpha\beta}] \quad (2.10b)$$

onde

$$C = \frac{Eh}{(1 - \nu^2)}$$

$$D = \frac{Eh^3}{12(1 - \nu^2)} \quad .$$

Lembrando que o modelo de Mindlin leva a tensões cisalhantes transversais constantes ao longo da espessura, o que não condiz com o fenômeno real, deve-se ponderar o valor de $\sigma_{\alpha 3}$ através da seguinte expressão, antes da integração das equações (2.9c):

$$\hat{\sigma}_{\alpha 3} = \kappa^2 \sigma_{\alpha 3} \quad .$$

Deste modo tem-se, após a integração,

$$Q_\alpha = \kappa^2 Gh (\psi_\alpha + u_{3,\alpha}) \quad . \quad (2.11)$$

2.3 Equações de equilíbrio e condições de contorno

Conhecendo-se as relações deformação-deslocamento, bem como as relações constitutivas, e ainda as condições de contorno de deslocamento sobre uma parcela Γ_u do contorno da placa, é possível a aplicação imediata do Princípio dos Trabalhos Virtuais (PTV) para a obtenção das equações de equilíbrio e das condições de contorno essenciais do problema. A forma geral do PTV é escrita (Washizu 1982):

$$\int_V \sigma_{ij} \delta \varepsilon_{ij} dV = \int_V b_i \delta U_i dV + \int_{\Gamma} t_i \delta U_i d\Gamma \quad , \quad \forall \delta U_i, \quad (2.12)$$

onde b_i e t_i são forças de campo e forças de superfície atuantes em direções paralelas aos eixos x_i , respectivamente.

Variando-se as componentes de deslocamento e de deformação a partir de (2.2) e (2.5), respectivamente, resulta

$$\begin{aligned} \delta U_\alpha &= \delta u_\alpha + x_3 \delta \psi_\alpha \\ \delta U_3 &= \delta u_3 \\ \delta \varepsilon_{\alpha\beta} &= \delta E_{\alpha\beta} + x_3 \delta K_{\alpha\beta} \\ \delta \gamma_{\alpha 3} &= \delta \psi_\alpha + \delta u_{3,\alpha} \quad , \end{aligned}$$

e substituindo as mesmas na expressão (2.12) resulta:

$$\begin{aligned} \int_{\Omega} \int_{-h/2}^{+h/2} [\sigma_{\alpha\beta} (\delta E_{\alpha\beta} + x_3 \delta K_{\alpha\beta}) + \sigma_{\alpha 3} (\delta \psi_\alpha + \delta u_{3,\alpha})] dx_3 d\Omega = \\ = \int_{\Omega} \int_{-h/2}^{+h/2} [b_\alpha (\delta u_\alpha + x_3 \delta \psi_\alpha) + b_3 \delta u_3] dx_3 d\Omega + \\ + \int_{\Gamma} [t_\alpha (\delta u_\alpha + x_3 \delta \psi_\alpha) + t_3 \delta u_3] d\Gamma \quad . \end{aligned} \quad (2.13)$$

Utilizando as definições (2.9) e integrando por partes, obtém-se:

$$\begin{aligned}
& \int_{\Omega} \left(-N_{\alpha\beta,\beta} \delta u_{\alpha} - N_{\alpha\beta,\beta} u_{3,\alpha} \delta u_3 - N_{\alpha\beta} u_{3,\alpha\beta} \delta u_3 - M_{\alpha\beta,\beta} \delta \psi_{\alpha} + \right. \\
& \quad \left. - Q_{\alpha,\alpha} \delta u_3 + Q_{\alpha} \delta \psi_{\alpha} \right) d\Omega + \\
& \quad + \oint_{\Gamma} (N_{\alpha\beta} n_{\beta} \delta u_{\alpha} + N_{\alpha\beta} u_{3,\alpha} n_{\beta} \delta u_3 + M_{\alpha\beta} n_{\beta} \delta \psi_{\alpha} + Q_{\alpha} n_{\alpha} \delta u_3) d\Gamma = \\
& = \int_{\Omega} (q_i \delta u_i + m_{\alpha} \delta \psi_{\alpha}) d\Omega + \int_{\Gamma} (t_i \delta u_i + x_3 t_{\alpha} \delta \psi_{\alpha}) d\Gamma \quad , \tag{2.14}
\end{aligned}$$

sendo q_i os carregamentos distribuídos atuantes nas direções paralelas aos eixos x_i , m_{α} são momentos distribuídos correspondentes aos eixos x_{α} , e n_{α} são os cossenos diretores de \mathbf{n} sobre o contorno Γ da placa. Igualando-se os coeficientes das variações obtém-se prontamente as equações de equilíbrio, dadas a seguir. Cabe salientar que tais equações são independentes de quaisquer hipóteses cinemáticas adotadas.

$$N_{\alpha\beta,\beta} + q_{\alpha} = 0 \tag{2.15a}$$

$$(N_{\alpha\beta} u_{3,\alpha})_{,\beta} + Q_{\alpha,\alpha} + q_3 = 0 \tag{2.15b}$$

$$M_{\alpha\beta,\beta} - Q_{\alpha} + m_{\alpha} = 0 \tag{2.15c}$$

As condições de contorno são obtidas de modo idêntico. Antes de escrevê-las, porém, destaque-se que, em um problema bem posto, é necessária a prescrição ou do deslocamento ou da força de superfície correspondente a cada uma das cinco direções generalizadas. Sejam então

$$u_i = \bar{u}_i \quad \text{e} \quad \psi_{\alpha} = \bar{\psi}_{\alpha} \quad , \quad \text{conhecido sobre } \Gamma_u \quad , \quad \text{e} \tag{2.16a}$$

$$t_i = \bar{t}_i \quad , \quad \text{conhecido sobre } \Gamma_t \quad , \tag{2.16b}$$

sendo Γ_u e Γ_t regiões complementares de Γ para cada par dual de variáveis. Assim, retira-se dos coeficientes das variações nas integrais sobre o contorno de (2.14) as condições de

contorno:

$$t_\alpha = N_{\alpha\beta}n_\beta \quad \text{sobre } \Gamma_u, \quad \text{ou} \quad \bar{t}_\alpha = \bar{N}_{\alpha\beta}n_\beta \quad \text{sobre } \Gamma_t \quad (2.17a)$$

$$t_3 = N_{\alpha\beta}u_{3,\alpha}n_\beta + Q_\alpha n_\alpha \quad \text{sobre } \Gamma_u, \quad \text{ou} \\ \bar{t}_3 = \bar{N}_{\alpha\beta}u_{3,\alpha}n_\beta + \bar{Q}_\alpha n_\alpha \quad \text{sobre } \Gamma_t, \quad (2.17b)$$

$$l_\alpha = M_{\alpha\beta}n_\beta \quad \text{sobre } \Gamma_u, \quad \text{ou} \quad \bar{l}_\alpha = \bar{M}_{\alpha\beta}n_\beta \quad \text{sobre } \Gamma_t, \quad (2.17c)$$

onde as variáveis prescritas estão indicadas com uma barra sobreposta e $l_\alpha = x_3 t_\alpha$. A equação (2.17b) mostra claramente que as condições de contorno do problema variam com a geometria. Cabe destacar ainda que Γ_u e Γ_t estão associados a cada uma das variáveis (prescritas ou não) de (2.17), e podem não coincidir para duas variáveis distintas, isto é, podem corresponder a regiões diferentes para cada uma das eqs.(2.17).

A substituição das relações deformação-deslocamento (2.5) nas expressões para tensões resultantes (2.10) e (2.11), leva às equações para os esforços em termos dos deslocamentos:

$$N_{\alpha\beta} = C \frac{1-\nu}{2} \left[u_{\alpha,\beta} + u_{\beta,\alpha} + u_{3,\alpha}u_{3,\beta} + \frac{2\nu}{1-\nu} \left(u_{\gamma,\gamma} + \frac{1}{2}u_{3,\gamma}u_{3,\gamma} \right) \delta_{\alpha\beta} \right] \quad (2.18a)$$

$$M_{\alpha\beta} = D \frac{1-\nu}{2} \left[\psi_{\alpha,\beta} + \psi_{\beta,\alpha} + \frac{2\nu}{1-\nu} \psi_{\gamma,\gamma} \delta_{\alpha\beta} \right] \quad (2.18b)$$

$$Q_\alpha = D\lambda^2 \frac{1-\nu}{2} [\psi_\alpha + u_{3,\alpha}] \quad , \quad (2.18c)$$

onde

$$\lambda^2 = \frac{12\kappa^2}{h^2} \quad . \quad (2.19)$$

No modelo de Reissner, $M_{\alpha\beta}$ é dado por

$$M_{\alpha\beta} = D \frac{1-\nu}{2} \left[\psi_{\alpha,\beta} + \psi_{\beta,\alpha} + \frac{2\nu}{1-\nu} \psi_{\gamma,\gamma} \delta_{\alpha\beta} \right] + \frac{\nu}{(1-\nu)\lambda^2} q_3 \delta_{\alpha\beta} \quad . \quad (2.20)$$

Distinguem-se as parcelas lineares e não-lineares de forma mais conveniente da seguinte

maneira:

$$N_{\alpha\beta} = N_{\alpha\beta}^l + N_{\alpha\beta}^n \quad (2.21a)$$

$$Q_\alpha = Q_\alpha^l + Q_\alpha^n \quad (2.21b)$$

com

$$N_{\alpha\beta}^l = C \frac{1-\nu}{2} \left[u_{\alpha,\beta} + u_{\beta,\alpha} + \frac{2\nu}{1-\nu} u_{\gamma,\gamma} \delta_{\alpha\beta} \right] \quad (2.22a)$$

$$N_{\alpha\beta}^n = C \frac{1-\nu}{2} \left[u_{3,\alpha} u_{3,\beta} + \frac{\nu}{1-\nu} u_{3,\gamma} u_{3,\gamma} \delta_{\alpha\beta} \right] \quad (2.22b)$$

$$Q_\alpha^l = D\lambda^2 \frac{1-\nu}{2} (\psi_\alpha + u_{3,\alpha}) \quad (2.22c)$$

$$Q_\alpha^n = N_{\alpha\beta} u_{3,\beta} \quad , \quad (2.22d)$$

onde Q_α^n foi reconhecido comparando-se as eqs.(2.15b) com as eqs.(2.18) ⁴. Agora, torna-se direto reescrever as equações de equilíbrio (2.15) na conhecida forma:

$$N_{\alpha\beta,\beta} + q_\alpha = 0 \quad (2.23a)$$

$$Q_{\alpha,\alpha} + q_\beta = 0 \quad (2.23b)$$

$$M_{\alpha\beta,\beta} - Q_\alpha^l + m_\alpha = 0 \quad (2.23c)$$

onde valem as igualdades (2.21).

A diferenciação das expressões (2.22) e a inserção dos resultados em (2.23) fornece as equações de equilíbrio do problema em termos de deslocamentos. Nesta forma, são conhecidas como equações de von Kármán. A diferença fundamental em relação ao sistema de equações de equilíbrio para pequenos deslocamentos reside nos esforços de membrana, presentes na parcela não-linear das equações de equilíbrio transversal, eq.(2.23b), os quais também são responsáveis pelo acoplamento do problema de flexão com o problema de membrana. Na forma como estão apresentadas estas equações valem para quaisquer configurações da placa sobre a trajetória de equilíbrio, mesmo após a ocorrência de pontos

⁴Uma forma comum de se obter o termo não-linear Q_α^n diretamente em 2.18c é utilizar a definição $Q_\alpha = \int_{-h/2}^{+h/2} (\sigma_{\alpha 3} + \sigma_{\alpha\beta} u_{3,\beta}) dx_3$, que já inclui o acoplamento flexão-membrana (ver, por exemplo, Xiao-Yan et al. 1990).

de bifurcação.

Desta maneira, as equações de equilíbrio são:

$$\begin{aligned}
N_{\alpha\beta,\beta} + q_\alpha &= N_{\alpha\beta,\beta}^l + N_{\alpha\beta,\beta}^n + q_\alpha = \\
&= C \frac{1-\nu}{2} \left[\Delta u_\alpha + \frac{1+\nu}{1-\nu} u_{\beta,\alpha\beta} + (u_{3,\alpha} u_{3,\beta})_{,\beta} + \right. \\
&\quad \left. + \frac{\nu}{1-\nu} (u_{3,\beta} u_{3,\beta})_{,\alpha} \right] + q_\alpha = 0
\end{aligned} \tag{2.24a}$$

$$\begin{aligned}
Q_{\alpha,\alpha} + q_3 &= Q_{\alpha,\alpha}^l + Q_{\alpha,\alpha}^n + q_3 = \\
&= D \lambda^2 \frac{1-\nu}{2} (\psi_{\alpha,\alpha} + \Delta u_3) + (N_{\alpha\beta} u_{3,\beta})_{,\alpha} + q_3 = 0
\end{aligned} \tag{2.24b}$$

$$\begin{aligned}
M_{\alpha\beta,\beta} - Q_\alpha^l + m_\alpha &= M_{\alpha\beta,\beta} - Q_\alpha^l + \frac{\nu}{(1-\nu)\lambda^2} q_{3,\alpha} + m_\alpha = \\
&= D \frac{1-\nu}{2} \left[(\Delta - \lambda^2) \psi_\alpha + \frac{1+\nu}{1-\nu} \psi_{\beta,\alpha\beta} - \lambda^2 u_{3,\alpha} \right] + \\
&\quad + \frac{\nu}{(1-\nu)\lambda^2} q_{3,\alpha} + m_\alpha = 0 \quad ,
\end{aligned} \tag{2.24c}$$

que são as cinco equações de Navier do problema, onde $\Delta = \partial^2/(\partial x_\alpha \partial x_\alpha)$ é o operador de Laplace bidimensional. Os termos destacados nas equações acima, correspondentes à parcela não-linear, impossibilitam que se escreva as equações de Navier na forma como geralmente é feito para a teoria linear:

$$\mathcal{L}_{ij}(\partial_Q) u_j(Q) = -\mathcal{F}_{ij}(\partial_Q) q_j(Q) \quad , \quad Q \equiv (x_1, x_2) \in \Omega. \quad . \tag{2.25}$$

Entretanto, é possível utilizar a forma geral acima se os termos não-lineares das equações de equilíbrio forem incorporados aos termos de carregamento. Esse procedimento leva a um sistema do tipo

$$\begin{bmatrix} {}^m\mathbf{L} & \mathbf{0} \\ \mathbf{0} & {}^f\mathbf{L} \end{bmatrix} \begin{Bmatrix} m_{\mathbf{u}} \\ f_{\mathbf{u}} \end{Bmatrix} = \begin{Bmatrix} m_{\widehat{\mathbf{q}}} \\ f_{\widehat{\mathbf{q}}} \end{Bmatrix} \tag{2.26}$$

onde ${}^m\mathbf{L}$ é o operador diferencial da parte linear das equações de equilíbrio do problema de membrana, ${}^f\mathbf{L}$ é o operador equivalente do problema de flexão, $m_{\mathbf{u}} = \{u_1 \ u_2\}^T$ são os deslocamentos longitudinais e $f_{\mathbf{u}} = \{\psi_1 \ \psi_2 \ u_3\}^T$ são os deslocamentos de placa. O acopla-

mento flexão-membrana está assim implícito nos pseudo-carregamentos correspondentes, ${}^m\hat{\mathbf{q}}$ e ${}^f\hat{\mathbf{q}}$:

$${}^m\hat{q}_\alpha = - {}^m\mathcal{F}_{\alpha\beta}(\partial_Q) {}^mq_\beta^l(Q) + {}^mq_\alpha^n(Q) \quad (2.27a)$$

$${}^f\hat{q}_i = - {}^f\mathcal{F}_{ij}(\partial_Q) {}^fq_j^l(Q) + {}^fq_i^n(Q) \quad (2.27b)$$

As formas explícitas dos termos usados em (2.26) e (2.27) são dadas a seguir.

$${}^m\mathbf{L}(\partial_Q) = C \frac{1-\nu}{2} \begin{bmatrix} \Delta + \bar{\nu} \partial_{11}^2 & \bar{\nu} \partial_{12}^2 \\ \bar{\nu} \partial_{12}^2 & \Delta + \bar{\nu} \partial_{22}^2 \end{bmatrix} \quad (2.28a)$$

$${}^f\mathbf{L}(\partial_Q) = D \frac{1-\nu}{2} \begin{bmatrix} \Delta - \lambda^2 + \bar{\nu} \partial_{11}^2 & \bar{\nu} \partial_{12}^2 & -\lambda^2 \partial_1 \\ \bar{\nu} \partial_{12}^2 & \Delta - \lambda^2 + \bar{\nu} \partial_{22}^2 & -\lambda^2 \partial_2 \\ \lambda^2 \partial_1 & \lambda^2 \partial_2 & \lambda^2 \Delta \end{bmatrix} \quad (2.28b)$$

$${}^m\mathbf{F}(\partial_Q) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (2.29a)$$

$${}^f\mathbf{F}(\partial_Q) = \begin{bmatrix} 1 & 0 & m_f \partial_1 \\ 0 & 1 & m_f \partial_2 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.29b)$$

onde $\partial_\alpha \equiv \partial/\partial x_\alpha$, $\partial_{\alpha\beta}^2 \equiv \partial^2/(\partial x_\alpha \partial x_\beta)$, $\Delta \equiv \partial_{\alpha\alpha}^2$ e $\bar{\nu} = (1+\nu)/(1-\nu)$. Ainda,

$${}^m\mathbf{q}^l(Q) = \begin{Bmatrix} q_1 \\ q_2 \end{Bmatrix} \quad (2.30a)$$

$${}^m\mathbf{q}^n(Q) = C \frac{1-\nu}{2} \begin{Bmatrix} (u_{3,1} u_{3,\alpha})_{,\alpha} + \frac{\nu}{1-\nu} (u_{3,\gamma} u_{3,\gamma})_{,1} \\ (u_{3,2} u_{3,\beta})_{,\beta} + \frac{\nu}{1-\nu} (u_{3,\gamma} u_{3,\gamma})_{,2} \end{Bmatrix} \quad (2.30b)$$

$${}^f\mathbf{q}^l(Q) = \begin{Bmatrix} m_1 \\ m_2 \\ q_3 \end{Bmatrix} \quad (2.31a)$$

$${}^f\mathbf{q}^n(Q) = D \frac{1-\nu}{2} \begin{Bmatrix} 0 \\ 0 \\ (N_{\alpha\beta}u_{3,\beta})_{,\alpha} \end{Bmatrix} \quad (2.31b)$$

onde m_f é o fator de modelo, que permite a unificação dos modelos de placa de Mindlin e de Reissner, conforme proposto por Westphal Jr. & de Barcellos (1990). O fator de modelo é dado por:

$$m_f = \frac{\nu}{(1-\nu)\lambda^2} \quad , \text{ para o modelo de Reissner} \quad (2.32a)$$

$$m_f = 0 \quad , \text{ para o modelo de Mindlin.} \quad (2.32b)$$

Note-se que, embora as eqs.(2.26) estejam acopladas, os operadores ${}^m\mathbf{L}$ e ${}^f\mathbf{L}$ são os mesmos da teoria linear.

2.4 Campo de tensões

As equações até aqui desenvolvidas são suficientes para descrever completamente o problema de flexão geometricamente não-linear de placas semi-espessas. No entanto, do ponto de vista da análise estrutural, torna-se importante o conhecimento das tensões locais que atuam nas regiões de interesse da estrutura. Assim, serão aqui desenvolvidas expressões para as tensões locais em termos dos deslocamentos e em termos dos carregamentos e tensões resultantes.

A inserção das relações (2.5) nas relações constitutivas (2.7) leva às seguintes expressões para tensões em termos dos deslocamentos:

$$\sigma_{\alpha\beta} = {}^m\sigma_{\alpha\beta} + {}^f\sigma_{\alpha\beta} \quad (2.33)$$

onde

$${}^m\sigma_{\alpha\beta} = {}^m\sigma_{\alpha\beta}^l + {}^m\sigma_{\alpha\beta}^n \quad (2.34)$$

e

$${}^m\sigma_{\alpha\beta}^l = G \left[u_{\alpha,\beta} + u_{\beta,\alpha} + \frac{2\nu}{1-\nu} u_{\gamma,\gamma} \delta_{\alpha\beta} \right] \quad (2.35a)$$

$${}^m\sigma_{\alpha\beta}^n = G \left[u_{3,\alpha} u_{3,\beta} + \frac{\nu}{1-\nu} u_{3,\gamma} u_{3,\gamma} \delta_{\alpha\beta} \right] \quad (2.35b)$$

$${}^f\sigma_{\alpha\beta} = G \left[\psi_{\alpha,\beta} + \psi_{\beta,\alpha} + \frac{2\nu}{1-\nu} \psi_{\gamma,\gamma} \delta_{\alpha\beta} \right] x_3 \quad (2.35c)$$

com as parcelas lineares e não-lineares das tensões de membrana já identificadas. As componentes $\sigma_{\alpha 3}$ dadas por (2.8c) permanecem válidas, pois já estão escritas em termos de deslocamentos:

$$\sigma_{\alpha 3} = G(\psi_{\alpha} + u_{3,\alpha})$$

A tensão normal transversal é obtida substituindo-se (2.6) em (2.8b), o que fornece

$$\sigma_{33} = \frac{\nu E}{(1+\nu)(1-2\nu)} \left[u_{\alpha,\alpha} + \frac{1}{2}(u_{3,\alpha} u_{3,\alpha}) - x_3 \psi_{\alpha,\alpha} \right] \quad (2.36)$$

A comparação entre as expressões (2.35) com as relações tensão-deslocamento (2.18) ou (2.22) permite escrever

$${}^m\sigma_{\alpha\beta}^l = \frac{N_{\alpha\beta}^l}{h} \quad (2.37a)$$

$${}^m\sigma_{\alpha\beta}^n = \frac{N_{\alpha\beta}^n}{h} \quad (2.37b)$$

ou, genericamente:

$${}^m\sigma_{\alpha\beta} = \frac{N_{\alpha\beta}}{h} \quad (2.38)$$

que não variam ao longo da espessura. Ainda:

$${}^f\sigma_{\alpha\beta} = \frac{12}{h^3} M_{\alpha\beta} x_3 \quad (2.39)$$

e

$$\sigma_{\alpha 3}^l = \frac{Q_\alpha}{\kappa^2 h} \quad (2.40a)$$

$$\sigma_{\alpha 3}^n = \frac{Gh^3}{12} N_{\alpha\beta} u_{3,\beta} \quad (2.40b)$$

As expressões (2.33) a (2.38) dadas acima devem ser utilizadas com alguma cautela, dependendo do modelo de placa utilizado. Na realidade, os deslocamentos ψ_α e u_3 representam uma ponderação do que ocorre ao longo da espessura quando o problema de flexão de placas é analisado do ponto de vista tridimensional. Assim, se as componentes dos deslocamentos sofridos pela placa nas direções x_i são denominadas v_i , uma definição mais abrangente pode ser empregada para os deslocamentos (Reissner 1944, Reissner 1945):

$$\begin{aligned} \psi_\alpha &= \frac{12}{h^3} \int_{-h/2}^{h/2} v_\alpha(x_1, x_2, x_3) dx_3 \quad , \\ u_3 &= \frac{3}{2h} \int_{-h/2}^{h/2} v_3(x_1, x_2, x_3) \left[1 - \left(\frac{2x_3}{h} \right)^2 \right] dx_3 \quad . \end{aligned}$$

Portanto, os deslocamentos ψ_α e u_3 são, na verdade, médias ponderadas dos deslocamentos realmente sofridos pela placa e são definidos sobre a superfície média da mesma. Com efeito, se v_α é linear em x_3 e v_3 é constante, então:

$$\begin{aligned} v_\alpha(x_i) &= x_3 \psi_\alpha(x_\beta) \quad , \\ v_3(x_i) &= u_3(x_\beta) \quad , \end{aligned}$$

e os dois modelos de placa aqui abordados se identificam. Neste caso, e somente neste caso, poder-se-ia utilizar as expressões (2.33) a (2.40) independentemente do modelo adotado. Este fato também pode ser demonstrado variacionalmente utilizando médias ponderadas para as componentes de deformação. Por simplicidade, o desenvolvimento a seguir segue o modelo de Reissner, que utiliza tensões cisalhantes transversais variáveis com a espessura,

sem o conhecimento explícito das funções que regem o comportamento dos deslocamentos v_i (Reissner 1944, Reissner 1945).

Posto isto, a parcela linear das tensões cisalhantes transversais, eqs.(2.40a), deve ser interpretada como uma tensão média ao longo da espessura, e a princípio válida apenas para o modelo de Mindlin. Para obter-se o seu comportamento em função da espessura, parte-se das equações de equilíbrio de tensão, na ausência de forças de corpo (Malvern 1969):

$$\sigma_{ji,i} = 0 \quad (2.41)$$

bem como das equações de equilíbrio em termos das tensões resultantes, eqs.(2.23). Das duas primeira equações de equilíbrio (2.41), e usando (2.23c), obtém-se uma expressão para $\sigma_{\alpha 3,3}$ que pode ser integrada em x_3 . Lembrando-se que q_α , m_α e Q_α são funções de (x_1, x_2) , este procedimento leva a:

$$\sigma_{\alpha 3} = \frac{q_\alpha}{h} x_3 + \frac{6x_3^2}{h^3} (m_\alpha - Q_\alpha) + C_\alpha \quad , \quad (2.42)$$

onde as constantes de integração C_α são obtidas através das seguintes condições de contorno:

$$\sigma_{\alpha 3} (\pm h/2) = 0 \quad , \quad (2.43)$$

que fornece

$$\sigma_{\omega 3} = q_\omega \left[\frac{(-1)^{\omega-1}}{2} + \frac{x_3}{h} \right] + \frac{3}{2h} \left[1 - \left(\frac{2x_3}{h} \right)^2 \right] (Q_\omega - m_\omega) \quad . \quad (2.44)$$

Substituindo-se então (2.44) na terceira equação de equilíbrio (2.41) e usando (2.23b), obtém-se uma expressão para $\sigma_{33,3}$ que também pode ser integrada na espessura. A cons-

tante de integração correspondente é obtida impondo-se a seguinte condição de contorno⁵:

$$\sigma_{33}(-h/2) = 0 \quad . \quad (2.45)$$

Este procedimento resulta:

$$\begin{aligned} \sigma_{33} = & \frac{1}{2} \left\{ q_{\alpha,\alpha} \left[(-1)^\omega \left(x_3 + \frac{h}{2} \right) + \frac{x_3^2}{h} - \frac{h}{4} \right] + \right. \\ & \left. + 3(q_3 + m_{\alpha,\alpha}) \left[\frac{x_3}{h} - \frac{4}{3} \left(\frac{x_3}{h} \right)^3 + \frac{1}{3} \right] \right\} \quad , \quad \omega = \alpha \quad . \end{aligned} \quad (2.46)$$

Note-se que $\sigma_{\alpha 3}$ varia quadraticamente com x_3 enquanto σ_{33} varia cubicamente. No entanto, foi imposta apenas uma condição de contorno sobre σ_{33} . A fim de investigar o que ocorre com σ_{33} na superfície superior da placa, substitui-se $x_3 = +h/2$ em (2.46), resultando

$$\sigma_{33}(+h/2) = q_{\alpha,\alpha}(-1)^\omega \frac{h}{2} + q_3 + m_{\alpha,\alpha} \quad , \quad \omega = \alpha \quad . \quad (2.47)$$

O termo $m_{\alpha,\alpha}$ corresponde à carga distribuída equivalente, provocada por momentos distribuídos (a exemplo do que ocorre na teoria de vigas). O carregamento transversal é dado por q_3 , e o termo $q_{\alpha,\alpha}$ é não-nulo apenas se houver forças paralelas à superfície de referência da placa, atuando na parte superior ou inferior da mesma. É conveniente salientar que (2.47) se reduz aos resultados obtidos por Reissner (1945) e Mindlin (1951) se $m_\alpha = q_\alpha = 0$.

A forma como as eqs.(2.44) e (2.46) estão escritas não é normalmente encontrada na literatura, pois o procedimento mais comum na suas deduções é pressupor metade do carregamento transversal aplicado em cada superfície da placa.

2.5 Equações para um estado de equilíbrio auxiliar

O uso de relações de reciprocidade envolve, via de regra, a convolução de variáveis duais correspondentes a dois estados de equilíbrio distintos. Em geral, um deles se refere ao

⁵A condição de contorno (2.45) parece mais prática do ponto de vista da engenharia, pois pressupõe todo o carregamento transversal aplicado sobre a superfície superior da placa, estando a superfície inferior livre de qualquer carregamento.

problema que se está interessado em resolver (Ω) e o outro corresponde a um estado conhecido, denominado auxiliar (Ω^*). Nas formulações integrais, é comum serem utilizadas algumas relações para um estado auxiliar correspondente a uma solução fundamental do problema. A seguir são apresentadas algumas destas relações, sendo identificadas por um sobre-índice asterisco. Tais equações não incluem quaisquer efeitos não-lineares, e são obtidas a partir da suposição de que as forças de corpo f_i^* variam segundo uma distribuição de esforços concentrados F_i^* , G_α^* que possuem a mesma forma operacional que a distribuição de tensões do modelo de Reissner (Westphal Jr. 1990). Levando-se em conta as eqs.(2.38), (2.39) e (2.44) define-se (na ausência de esforços de membrana e momentos distribuídos):

$$f_\alpha^* = {}^m f_\alpha^* + {}^f f_\alpha^* = \frac{F_\alpha^*}{h} + \frac{12}{h^3} x_3 G_\alpha^* \quad (2.48a)$$

$$f_3^* = \frac{3}{2h} \left[1 - \left(\frac{2x_3}{h} \right)^2 \right] F_3^* , \quad (2.48b)$$

onde F_i^* representa uma carga concentrada unitária generalizada na direção x_i , enquanto G_α^* representa um momento concentrado generalizado na direção x_α .

Para tal estado auxiliar as equações de equilíbrio são:

$$\sigma_{ij,j}^* + f_i^* = 0$$

e, pressupondo-se que apenas forças de corpo são consideradas, pode-se utilizar as eqs.(2.38), (2.39), (2.42) e (2.46) para escrever,

$$\sigma_{\alpha\beta}^* = {}^m \sigma_{\alpha\beta}^* + {}^f \sigma_{\alpha\beta}^* = \frac{N_{\alpha\beta}^*}{h} + \frac{12}{h^3} x_3 M_{\alpha\beta}^* \quad (2.49a)$$

$$\sigma_{\alpha 3}^* = \frac{3}{2h} \left[1 - \left(\frac{2x_3}{h} \right)^2 \right] Q_\alpha^* \quad (2.49b)$$

$$\sigma_{33}^* = 0 . \quad (2.49c)$$

A substituição de (2.49a), (2.49b) e (2.48a) na primeira e terceira equações de equilíbrio 2.23 resulta:

$$\begin{aligned} N_{\alpha\beta,\beta}^* + F_\alpha^* &= 0 \\ M_{\alpha\beta,\beta}^* - Q_\alpha^* + G_\alpha^* &= 0 \ , \end{aligned}$$

enquanto que a inserção de (2.49b) e (2.49c) na equação de equilíbrio remanescente fornece:

$$Q_{\alpha,\alpha}^* + F_3^* = 0 \ .$$

As variáveis envolvidas nas eqs.(2.49) estão relacionadas linearmente com o campo de deslocamentos do estado auxiliar. Então emprega-se a mesma forma das eqs.(2.18b), (2.22a) e (2.22c) para escrever:

$$\begin{aligned} N_{\alpha\beta}^* &= C \frac{1-\nu}{2} \left[u_{\alpha,\beta}^* + u_{\beta,\alpha}^* + \frac{2\nu}{1-\nu} u_{\gamma,\gamma}^* \delta_{\alpha\beta} \right] \ , \\ M_{\alpha\beta}^* &= D \frac{1-\nu}{2} \left[\psi_{\alpha,\beta}^* + \psi_{\beta,\alpha}^* + \frac{2\nu}{1-\nu} \psi_{\gamma,\gamma}^* \delta_{\alpha\beta} \right] \ , \\ Q_\alpha^* &= \lambda^2 D \frac{1-\nu}{2} (\psi_\alpha^* + u_{3,\alpha}^*) \ . \end{aligned}$$

Resumindo-se as quantidades relativas aos dois estados, tem-se:

- Equações de equilíbrio:

$$N_{\alpha\beta,\beta} + q_\alpha = 0 \tag{2.50a}$$

$$Q_{\alpha,\alpha} + q_3 = 0 \tag{2.50b}$$

$$M_{\alpha\beta,\beta} - Q_\alpha^l + m_\alpha = 0 \ , \text{ para o problema original} \tag{2.50c}$$

e

$$N_{\alpha\beta,\beta}^* + F_\alpha^* = 0 \tag{2.51a}$$

$$Q_{\alpha,\alpha}^* + F_3^* = 0 \tag{2.51b}$$

$$M_{\alpha\beta,\beta}^* - Q_\alpha^* + G_\alpha^* = 0 \ , \text{ para o estado auxiliar.} \tag{2.51c}$$

- Deslocamentos:

$$u_i \quad , \quad \psi_\alpha \quad , \quad \text{para o problema original e} \quad (2.52a)$$

$$u_i^* \quad , \quad \psi_\alpha^* \quad , \quad \text{para o estado auxiliar.} \quad (2.52b)$$

- Forças de superfície:

$$t_\alpha = N_{\alpha\beta} n_\beta \quad (2.53a)$$

$$t_3 = N_{\alpha\beta} u_{3,\alpha} n_\beta + Q_\alpha n_\alpha \quad (2.53b)$$

$$l_\alpha = M_{\alpha\beta} n_\beta \quad , \quad \text{para o problema original} \quad (2.53c)$$

e

$$t_\alpha^* = N_{\alpha\beta}^* n_\beta \quad (2.54a)$$

$$t_3^* = Q_\alpha^* n_\alpha \quad (2.54b)$$

$$l_\alpha^* = M_{\alpha\beta}^* n_\beta \quad , \quad \text{para o estado auxiliar.} \quad (2.54c)$$

- Deformações:

$$\varepsilon_{\alpha\beta} = E_{\alpha\beta} + x_3 K_{\alpha\beta} \quad (2.55a)$$

$$\gamma_{\alpha 3} = \psi_\alpha + u_{3,\alpha} \quad (2.55b)$$

onde

$$E_{\alpha\beta} = \frac{1}{2}(u_{\alpha,\beta} + u_{\beta,\alpha} + u_{3,\alpha} u_{3,\beta}) \quad (2.56a)$$

$$K_{\alpha\beta} = \frac{1}{2}(\psi_{\alpha,\beta} + \psi_{\beta,\alpha}) \quad , \quad \text{para o problema original}$$

e

$$\varepsilon_{\alpha\beta}^* = E_{\alpha\beta}^* + x_3 K_{\alpha\beta}^* \quad (2.57a)$$

$$\gamma_{\alpha 3}^* = \psi_\alpha^* + u_{3,\alpha}^* \quad (2.57b)$$

onde

$$E_{\alpha\beta}^* = \frac{1}{2}(u_{\alpha,\beta}^* + u_{\beta,\alpha}^*) \quad (2.58a)$$

$$K_{\alpha\beta}^* = \frac{1}{2}(\psi_{\alpha,\beta}^* + \psi_{\beta,\alpha}^*) \quad , \quad \text{para o estado auxiliar.} \quad (2.58b)$$

Tensões resultantes:

$$N_{\alpha\beta} = C \frac{1-\nu}{2} \left[u_{\alpha,\beta} + u_{\beta,\alpha} + u_{3,\alpha} u_{3,\beta} + \frac{2\nu}{1-\nu} \left(u_{\gamma,\gamma} + \frac{1}{2} u_{3,\gamma} u_{3,\gamma} \right) \delta_{\alpha\beta} \right] \quad (2.59a)$$

$$M_{\alpha\beta} = D \frac{1-\nu}{2} \left[\psi_{\alpha,\beta} + \psi_{\beta,\alpha} + \frac{2\nu}{1-\nu} \psi_{\gamma,\gamma} \delta_{\alpha\beta} \right] + m_f q_3 \delta_{\alpha\beta} \quad (2.59b)$$

$$Q_\alpha = \lambda^2 D \frac{1-\nu}{2} (\psi_\alpha + u_{3,\alpha}) \quad , \quad (2.59c)$$

para o problema original, e

$$N_{\alpha\beta}^* = C \frac{1-\nu}{2} \left[u_{\alpha,\beta}^* + u_{\beta,\alpha}^* + \frac{2\nu}{1-\nu} u_{\gamma,\gamma}^* \delta_{\alpha\beta} \right] \quad (2.60a)$$

$$M_{\alpha\beta}^* = D \frac{1-\nu}{2} \left[\psi_{\alpha,\beta}^* + \psi_{\beta,\alpha}^* + \frac{2\nu}{1-\nu} \psi_{\gamma,\gamma}^* \delta_{\alpha\beta} \right] \quad (2.60b)$$

$$Q_\alpha^* = \lambda^2 D \frac{1-\nu}{2} (\psi_\alpha^* + u_{3,\alpha}^*) \quad (2.60c)$$

para o estado auxiliar.

2.6 Resumo do Capítulo 2

Neste capítulo foram apresentadas as equações necessárias para a completa descrição do fenômeno de flexão de placas semi-espessas, incluindo os efeitos geometricamente não-lineares, utilizando-se as teorias de Mindlin e de Reissner. Incluiu-se uma breve análise do comportamento das tensões transversais ao longo da espessura. Foram também desenvolvidas as equações para um estado de equilíbrio correspondente à uma solução fundamental do problema. As equações (2.50) a (2.60) formam a base do desenvolvimento da formulação integral apresentada no próximo capítulo.

Capítulo 3

Formulação Integral

Neste capítulo são desenvolvidas as equações integrais para os modelos de placa empregados, incluindo os termos geometricamente não-lineares. Inicialmente é apresentado o procedimento para um operador diferencial $\mathcal{L}(\partial_Q)$ genérico. Seguindo-se então tal procedimento, são obtidas as formulações integrais correspondentes aos operadores ${}^m\mathcal{L}(\partial_Q)$ e ${}^f\mathcal{L}(\partial_Q)$ dados pelas equações (2.28a) e (2.28b). Em ambos os casos o tratamento leva às identidades de Somigliana para os problemas de membrana e flexão, sobre as quais é aplicada a propriedade do traço, obtendo-se os problemas de valores sobre o contorno correspondentes. Tendo em vista o acoplamento dos termos de flexão com os de membrana, equações hipersingulares adicionais são derivadas, obtendo-se assim as equações integrais necessárias para a descrição de problemas de flexão não-linear de placas. Uma particularização dessas equações leva à formulação integral representativa do problema de estabilidade linear elástica de placas.

3.1 Formalismo geral

O formalismo algébrico de uma formulação integral data de meados do século XIX, quando o matemático inglês G. Green mostrou como transformar uma formulação diferencial de domínio para uma descrição integral definida sobre o contorno (Stein & Wendland 1988). Já no início do século XX, I. Fredholm aplicou a idéia a problemas do meio contínuo, e postulou os requisitos para existência e unicidade de soluções de equações integrais (Beskos 1991). As formas integrais de problemas de valores sobre o contorno podem ser

obtidas de diversas maneiras. Por exemplo, para problemas lineares, a aplicação imediata da terceira identidade de Green, ou dos teoremas de reciprocidade de Betti e Castigliano ou ainda uma forma geral da identidade de Somigliana fornecem resultados idênticos. Entretanto, a utilização de tais métodos não permite facilmente a extensão da formulação a problemas mais complexos, como os que envolvem algum tipo de não-linearidade (Stein & Wendland 1988). No caso geral, a utilização de uma forma do método dos resíduos ponderados (MRP) é mais direta e abrangente, permitindo inclusive uma unificação das relações funcionais que determinam a origem de vários métodos de solução, aproximados ou não (Banerjee & Butterfield 1981, Brebbia & Dominguez 1989).

Seja então um problema qualquer dado pela sua equação diferencial:

$$\mathcal{L}(\partial_Q)u(Q) = q(Q) \quad Q \in \mathfrak{R}^n \quad , \quad (3.1)$$

onde \mathcal{L} é um operador diferencial de ordem m . Seja também o produto interno

$$\langle u, v \rangle_{\mathcal{H}} = \int_{\mathcal{H}} uv \, dx \quad , \quad (3.2)$$

entre as variáveis u e v , definidas sobre os espaços de Hilbert \mathcal{H} e seu dual \mathcal{H}' , respectivamente. Admita-se que v seja k -vezes diferenciável¹ e que $k \leq m$. A aplicação de (3.1) na definição (3.2) gera:

$$\int_{\mathcal{H}} (\mathcal{L}u - q) v \, dx = 0 \quad . \quad (3.3)$$

A integração de (3.3) k -vezes, por partes, leva à fórmula de Green generalizada:

$$\langle \mathcal{L}u, v \rangle_{\mathcal{H}} - \langle \mathcal{L}v, u \rangle_{\mathcal{H}} = \langle \mathcal{N}v, \mathcal{D}u \rangle_{\partial\mathcal{H}} - \langle \mathcal{N}u, \mathcal{D}v \rangle_{\partial\mathcal{H}} \quad , \quad (3.4)$$

onde \mathcal{N} e \mathcal{D} são os operadores traço generalizados de Neumann e Dirichlet, respectivamente. A partir da prescrição de $\mathcal{N}u$ sobre $\partial\mathcal{H}_t$ e de $\mathcal{D}u$ sobre $\partial\mathcal{H}_u$, define-se as condições de contorno essenciais e naturais do problema. Uma nova integração de (3.4) por partes k -vezes, agora aplicando o operador diferencial novamente sobre u , leva ao chamado pro-

¹No sentido distributivo, v é uma função infinitamente diferenciável, o que é obtido com $k \rightarrow \infty$. Portanto, não há perda de generalidade.

blema direto:

$$\langle (\mathcal{L}u - q), v \rangle_{\partial\mathcal{H}} = \langle (\mathcal{D}u - \overline{\mathcal{D}u}), \mathcal{N}v \rangle_{\partial\mathcal{H}_u} + \langle (\overline{\mathcal{N}u} - \mathcal{N}u), \mathcal{D}v \rangle_{\partial\mathcal{H}_t} \quad , \quad (3.5)$$

onde o sobretraço indica uma quantidade prescrita. Se agora u é substituído por uma solução aproximada \tilde{u} , a eq.(3.5) passa a representar, na realidade, uma ponderação dos resíduos

$$\mathcal{R} = \mathcal{L}\tilde{u} - q \quad \text{em } \mathcal{H} \quad (3.6a)$$

$$\mathcal{R}_t = \mathcal{N}\tilde{u} - \overline{\mathcal{N}u} \quad \text{sobre } \partial\mathcal{H}_t \quad (3.6b)$$

$$\mathcal{R}_u = \mathcal{D}\tilde{u} - \overline{\mathcal{D}u} \quad \text{sobre } \partial\mathcal{H}_u \quad , \quad (3.6c)$$

que permitem reescrever (3.5) na forma como é conhecido o MRP:

$$\langle \mathcal{R}, v \rangle_{\mathcal{H}} = \langle \mathcal{R}_u, \mathcal{N}v \rangle_{\partial\mathcal{H}_u} - \langle \mathcal{R}_t, \mathcal{D}v \rangle_{\partial\mathcal{H}_t} \quad . \quad (3.7)$$

Se a eq.(3.7) é integrada por partes um número de vezes qualquer, menor que m , a fim de transferir uma parte da ordem de \mathcal{L} para v , obtém-se a chamada *forma fraca* do problema. A partir da forma fraca, ou de um funcional equivalente, são obtidos outros métodos de solução aproximada, como o MEF e o MDF, dependendo das propriedades da função peso v (Oden & Reddy 1976, Reddy 1984a). Por outro lado, pode-se integrar (3.7) por partes exatamente m -vezes, aplicando agora toda ordem de \mathcal{L} sobre a variável v , o que leva a:

$$\langle \mathcal{L}v, u \rangle_{\mathcal{H}} - \langle q, v \rangle_{\mathcal{H}} = \langle \mathcal{N}u, \mathcal{D}v \rangle_{\partial\mathcal{H}_t} - \langle \mathcal{N}v, \mathcal{D}u \rangle_{\partial\mathcal{H}_u} \quad , \quad (3.8)$$

que vem a ser o chamado *problema inverso*, o ponto de partida para métodos integrais como o MEC e o MFG. As funções u e v são geralmente associadas a dois estados distintos: o das incógnitas a serem determinadas e um outro auxiliar. No caso do MEC direto, o estado auxiliar geralmente corresponde a uma solução fundamental do problema, e é usado diretamente no problema inverso ou em relações de reciprocidade energéticas. A versão indireta do método é obtida através da superposição de produtos duais entre

soluções auxiliares singulares unitárias e funções densidade definidas sobre o contorno $\partial\mathcal{H}$, cujas amplitudes são determinadas a fim de garantir a unicidade da solução (Stein & Wendland 1988, Brebbia & Dominguez 1989). O MEC direto relaciona, portanto, as variáveis físicas definidas sobre o domínio e sobre o contorno, enquanto na versão indireta do método não há uma relação imediata entre as funções densidade e as variáveis no contorno (Brebbia & Dominguez 1989). Além disso, o MEC indireto não se comporta bem na solução numérica de problemas definidos por contornos que possuem vértices, ou seja, pontos onde existe descontinuidade da normal exterior (Stein & Wendland 1988). Claramente, tais aspectos têm favorecido sobremaneira o desenvolvimento e aplicação da versão direta do MEC para solução de problemas práticos de engenharia.

A seguir, partindo-se da expressão geral do MRP, equação (3.7), é obtido o problema inverso para os operadores diferenciais de membrana e de flexão utilizando como estado auxiliar o apresentado no final do Capítulo 2.

3.2 Formulação integral para o problema de membrana

Pela forma geral do MRP, equação (3.7), as equações integrais para ponderação dos resíduos do problema de membrana são escritas utilizando as equações de equilíbrio (2.15a) e as condições de contorno (2.15):

$$\int_{\Omega} {}^m\mathcal{R}^\alpha u^\alpha d\Omega = \int_{\Gamma_t} {}^m\mathcal{R}_t^\alpha u^\alpha d\Gamma - \int_{\Gamma_u} {}^m\mathcal{R}_u^\alpha q^\alpha d\Gamma \quad (3.9)$$

onde os resíduos produzidos por uma solução aproximada são dados por:

$${}^m R^\alpha = N_{\alpha\beta,\beta} + q_\alpha \neq 0 \quad \text{em } \Omega \quad , \quad (3.10a)$$

$${}^m R_u^\alpha = u_\alpha - \bar{u}_\alpha \neq 0 \quad \text{sobre } \Gamma_u \quad , \quad (3.10b)$$

$${}^m R_t^\alpha = t_\alpha - \bar{t}_\alpha \neq 0 \quad \text{sobre } \Gamma_t \quad . \quad (3.10c)$$

E as funções-peso adotadas, segundo as quais os erros são minimizados, são providas pelo estado de equilíbrio auxiliar:

$$u^\alpha = u_\alpha^* \quad (3.11a)$$

$$q^\alpha = t_\alpha^* \quad (3.11b)$$

$$u_\alpha = u_\alpha^* \quad (3.11c)$$

$$u_\alpha = u_\alpha^* \quad (3.11d)$$

A substituição das expressões (3.10) e (3.11) na eq.(3.9) e uma integração por partes da integral em Ω fornece:

$$\begin{aligned} \int_{\Gamma} N_{\alpha\beta} n_\beta u_\alpha^* d\Gamma - \int_{\Omega} N_{\alpha\beta} u_{\alpha,\beta}^* d\Omega &= \int_{\Gamma_t} (t_\alpha - \bar{t}_\alpha) u_\alpha^* d\Gamma + \\ &- \int_{\Gamma_u} (u_\alpha - \bar{u}_\alpha) t_\alpha^* d\Gamma - \int_{\Omega} q_\alpha u_\alpha^* d\Omega \end{aligned} \quad (3.12)$$

Utilizando (??) e (2.21a), e lembrando que Γ_u e Γ_t são regiões complementares de Γ , com $\Gamma_u \cap \Gamma_t = \mathbf{0}$, reescreve-se

$$\begin{aligned} \int_{\Omega} N_{\alpha\beta}^l u_{\alpha,\beta}^* d\Omega &= \int_{\Omega} q_\alpha u_\alpha^* d\Omega - \int_{\Omega} N_{\alpha\beta}^n u_{\alpha,\beta}^* d\Omega + \int_{\Gamma_t} \bar{t}_\alpha u_\alpha^* d\Gamma + \\ &+ \int_{\Gamma_u} t_\alpha u_\alpha^* d\Gamma + \int_{\Gamma_u} (u_\alpha - \bar{u}_\alpha) t_\alpha^* d\Gamma \quad . \end{aligned} \quad (3.13)$$

Dos teoremas de reciprocidade²,

$$N_{\alpha\beta}^l E_{\alpha\beta}^* = N_{\alpha\beta}^* u_{\alpha,\beta} \quad ,$$

e como $E_{\alpha\beta}^*$ tem a mesma forma operacional da parte linear de $E_{\alpha\beta}$, eqs.(2.56a) e (2.58a), vem

$$N_{\alpha\beta}^l u_{\alpha,\beta}^* = N_{\alpha\beta}^* u_{\alpha,\beta} \quad .$$

²Tal resultado é obtido com qualquer medida variacionalmente consistente de deformação, isto é, tanto com $E_{\alpha\beta}^l = u_{\alpha,\beta}$ quanto com a medida usual $E_{\alpha\beta}^l = \frac{1}{2}(u_{\alpha,\beta} + u_{\beta,\alpha})$.

Assim, é correto escrever, para a integral à esquerda de (3.13):

$$\int_{\Omega} N_{\alpha\beta}^l E_{\alpha\beta}^* d\Omega = \int_{\Omega} N_{\alpha\beta}^l u_{\alpha,\beta}^* d\Omega = \int_{\Omega} N_{\alpha\beta}^* u_{\alpha,\beta} d\Omega \quad ,$$

que, quando integrada por partes e substituída em (3.13) leva a:

$$\begin{aligned} - \int_{\Omega} N_{\alpha\beta,\beta}^* u_{\alpha} d\Omega &= - \int_{\Omega} N_{\alpha\beta}^n u_{\alpha,\beta}^* d\Omega + \int_{\Omega} q_{\alpha} u_{\alpha}^* d\Omega + \int_{\Gamma_t} \bar{t}_{\alpha} u_{\alpha}^* d\Gamma + \\ &+ \int_{\Gamma_u} t_{\alpha} u_{\alpha}^* d\Gamma + \int_{\Gamma_u} (u_{\alpha} - \bar{u}_{\alpha}) t_{\alpha}^* d\Gamma - \int_{\Gamma} N_{\alpha\beta}^* n_{\beta} u_{\alpha} d\Gamma \quad . \end{aligned}$$

Usando agora (2.51a) e (2.54a), vem

$$\int_{\Omega} F_{\alpha}^* u_{\alpha} d\Omega + \int_{\Gamma} u_{\alpha} t_{\alpha}^* d\Gamma = \int_{\Omega} q_{\alpha} u_{\alpha}^* d\Omega + \int_{\Gamma} t_{\alpha} u_{\alpha}^* d\Gamma - \int_{\Omega} N_{\alpha\beta}^n u_{\alpha,\beta}^* d\Omega \quad . \quad (3.14)$$

Da definição de solução fundamental de um problema, F_{α}^* corresponde à excitação gerada por uma carga generalizada unitária na direção e_{α} , isto é:

$$F_{\alpha}^* = \delta(Q, P) e_{\alpha}(P) \quad , \quad P, Q \in \Omega \quad , \quad (3.15)$$

sendo que a distribuição $\delta(Q, P)$, função generalizada delta de Dirac, possui a propriedade

$$\int_{\mathcal{H}} g(Q) \delta(Q, P) dx = g(P) \quad , \quad (3.16)$$

para um ponto P onde é aplicada a excitação (*ponto fonte*) e um ponto Q onde são medidos os efeitos da excitação (*ponto campo*). As excitações F_{α}^* representam, portanto, forças concentradas generalizadas unitárias nas direções e_{α} . Considerando cada uma dessas cargas atuando independentemente, resulta (Brebbia et al. 1984):

$$u_{\alpha}^* = {}^m U_{\beta\alpha}(Q, P) e_{\beta}(P) \quad (3.17a)$$

$$t_{\alpha}^* = {}^m T_{\beta\alpha}(Q, P) e_{\beta}(P) \quad , \quad (3.17b)$$

que são os deslocamentos e trações do estado auxiliar, dados pelos tensores correspondentes à uma solução fundamental do problema de membrana, ${}^m \mathbf{U}$ e ${}^m \mathbf{T}$. A substituição

de (3.17) em (3.14) resulta, considerando (3.15) e (3.16) para $g(Q) = u_\alpha(Q)$:

$$u_\alpha(P) + \int_{\Gamma} {}^m T_{\alpha\beta}(Q, P) u_\beta(Q) d\Gamma_Q = \int_{\Gamma} {}^m U_{\alpha\beta}(Q, P) t_\beta(Q) d\Gamma_Q + \int_{\Omega} {}^m U_{\alpha\beta}(Q, P) q_\beta(Q) d\Omega_Q - \int_{\Omega} {}^m U_{\alpha\beta,\gamma}(Q, P) N_{\beta\gamma}^n(Q) d\Omega_Q \quad , \quad (3.18)$$

que é a identidade de Somigliana para os deslocamentos u_α (Brebbia et al. 1984). O termo ${}^m U_{\alpha\beta,\gamma}$ representa uma derivada do tensor ${}^m \mathbf{U}$ em relação a $x_\gamma(Q)$. A eq.(3.18) é a contrapartida, para o problema de membrana, da relação geral (3.8).

3.3 Formulação integral para o problema de flexão

Para o problema de flexão, utiliza-se as equações de equilíbrio (2.50b) e (2.50c) para escrever o MRP na forma

$$\int_{\Omega} ({}^f \mathcal{R}^3 u^3 + {}^f \mathcal{R}^\alpha \psi^\alpha) d\Omega = \int_{\Gamma_t} ({}^f \mathcal{R}_t^3 u^3 + {}^f \mathcal{R}_t^\alpha \psi^\alpha) d\Gamma - \int_{\Gamma_u} ({}^f \mathcal{R}_u^3 q^3 + {}^f \mathcal{R}_u^\alpha m^\alpha) d\Gamma \quad , \quad (3.19)$$

onde os resíduos de uma solução aproximada são dados por:

$${}^f \mathcal{R}^3 = (N_{\alpha\beta} u_{3,\alpha})_{,\beta} + Q_{\alpha,\alpha}^l + q_3 \neq 0 \quad \text{em } \Omega \quad , \quad (3.20a)$$

$${}^f \mathcal{R}^\alpha = M_{\alpha\beta,\beta} - Q_\alpha^l + m_\alpha \neq 0 \quad \text{em } \Omega \quad , \quad (3.20b)$$

$${}^f \mathcal{R}_t^3 = t_3^l - \bar{t}_3 \neq 0 \quad \text{sobre } \Gamma_t \quad , \quad (3.20c)$$

$${}^f \mathcal{R}_t^\alpha = l_\alpha - \bar{l}_\alpha \neq 0 \quad \text{sobre } \Gamma_t \quad , \quad (3.20d)$$

$${}^f \mathcal{R}_u^3 = u_3 - \bar{u}_3 \neq 0 \quad \text{sobre } \Gamma_u \quad , \quad (3.20e)$$

$${}^f \mathcal{R}_u^\alpha = \psi_\alpha - \bar{\psi}_\alpha \neq 0 \quad \text{sobre } \Gamma_u \quad ; \quad (3.20f)$$

e as funções-peso são:

$$u^3 = u_3^* \quad (3.21a)$$

$$\psi^\alpha = \psi_\alpha^* \quad (3.21b)$$

$$q^3 = t_3^* \quad (3.21c)$$

$$m^\alpha = l_\alpha^* \quad , \quad (3.21d)$$

e então reescreve-se (3.19) na forma:

$$\begin{aligned} & \int_{\Omega} \left[(N_{\alpha\beta} u_{3,\alpha})_{,\beta} + Q_{\alpha,\alpha}^l + q_3 \right] u_3^* d\Omega + \int_{\Omega} (M_{\alpha\beta,\beta} - Q_\alpha^l + m_\alpha) \psi_\alpha^* d\Omega = \\ = & \int_{\Gamma_t} \left[(t_3^l - \bar{t}_3) u_3^* + (l_\alpha - \bar{l}_\alpha) \psi_\alpha^* \right] d\Gamma - \int_{\Gamma_u} \left[(u_3 - \bar{u}_3) t_3^* + (\psi_\alpha - \bar{\psi}_\alpha) l_\alpha^* \right] d\Gamma \end{aligned} \quad (3.22)$$

Os termos não-lineares da primeira integral em Ω podem ser incorporados ao carregamento transversal q_3 , definindo-se assim um *pseudo-carregamento* transversal³:

$$\hat{q} = q_3 + (N_{\alpha\beta} u_{3,\alpha})_{,\beta} \quad . \quad (3.23)$$

Utilizando (3.23) e integrando os termos em $Q_{\alpha,\alpha}^l$ e $M_{\alpha\beta,\beta}$ por partes, obtém-se:

$$\begin{aligned} & \int_{\Omega} \left(-Q_\alpha^l u_{3,\alpha}^* - M_{\alpha\beta} \psi_{\alpha,\beta}^* \right) d\Omega + \int_{\Omega} \left(\hat{q} u_3^* - Q_\alpha^l \psi_\alpha^* + m_\alpha \psi_\alpha^* \right) d\Omega = \\ & = \int_{\Gamma_t} \left[(t_3^l - \bar{t}_3) u_3^* + (l_\alpha - \bar{l}_\alpha) \psi_\alpha^* \right] d\Gamma + \\ & - \int_{\Gamma_u} \left[(u_3 - \bar{u}_3) t_3^* + (\psi_\alpha - \bar{\psi}_\alpha) l_\alpha^* \right] d\Gamma + \\ & - \int_{\Gamma} \left(Q_\alpha^l n_\alpha u_3^* + M_{\alpha\beta} n_\beta \psi_\alpha^* \right) d\Gamma \quad . \end{aligned} \quad (3.24)$$

Observando, das eqs.(2.17b) e (2.17c), que $t_3^l = Q_\alpha^l n_\alpha$ e $l_\alpha = M_{\alpha\beta} n_\beta$, e salientando

³Este pseudo-carregamento vem a ser a soma do carregamento transversal com a projeção do carregamento de membrana sobre a direção x_3 , após a deformação da placa.

que $\Gamma = \Gamma_u \cup \Gamma_t$, reescreve-se (3.24):

$$\begin{aligned} & \int_{\Omega} \left[M_{\alpha\beta} \psi_{\alpha,\beta}^* + Q_{\alpha} (\psi_{\alpha}^* + u_{3,\alpha}^*) \right] d\Omega = \int_{\Omega} (\hat{q}u_3^* + m_{\alpha} \psi_{\alpha}^*) d\Omega + \\ & + \int_{\Gamma} (t_3^l u_3^* + l_{\alpha} \psi_{\alpha}^*) d\Gamma + \int_{\Gamma_u} [(u_3 - \bar{u}_3) t_3^* + (\psi_{\alpha} - \bar{\psi}_{\alpha}) l_{\alpha}^*] d\Gamma \quad . \end{aligned} \quad (3.25)$$

A partir das expressões para o estado auxiliar Ω^* , verifica-se a seguinte identidade⁴:

$$(M_{\alpha\beta} - m_f q_3 \delta_{\alpha\beta}) K_{\alpha\beta}^* + Q_{\alpha} \gamma_{\alpha 3}^* = M_{\alpha\beta}^* K_{\alpha\beta} + Q_{\alpha}^* \gamma_{\alpha 3} \quad , \quad (3.26)$$

e como $M_{\alpha\beta}$ é simétrico, outra igualdade é verificada, no sentido variacional:

$$M_{\alpha\beta}^* K_{\alpha\beta} = M_{\alpha\beta}^* \psi_{\alpha,\beta} \quad . \quad (3.27)$$

Substituindo (3.26), (3.27) e (2.55b) em (3.25) resulta, após uma integração por partes dos termos envolvendo $\psi_{\alpha,\beta}^*$ e $u_{3,\alpha}^*$:

$$\begin{aligned} & - \int_{\Omega} (M_{\alpha\beta,\beta}^* - Q_{\alpha}^*) \psi_{\alpha} d\Omega - \int_{\Omega} Q_{\alpha,\alpha}^* u_3 d\Omega = \int_{\Omega} (\hat{q}u_3^* + m_{\alpha} \psi_{\alpha}^*) d\Omega + \\ & + \int_{\Gamma} (t_3^l u_3^* + l_{\alpha} \psi_{\alpha}^*) d\Gamma - \int_{\Gamma_u} [(u_3 - \bar{u}_3) t_3^* + (\psi_{\alpha} - \bar{\psi}_{\alpha}) l_{\alpha}^*] d\Gamma + \\ & - \int_{\Gamma_u} (M_{\alpha\beta}^* n_{\beta} \psi_{\alpha} + Q_{\alpha}^* n_{\alpha} u_3) d\Gamma - \int_{\Gamma_t} (M_{\alpha\beta}^* n_{\beta} \psi_{\alpha} + Q_{\alpha}^* n_{\alpha} u_3) d\Gamma + \\ & - m_f \int_{\Omega} q_3 \psi_{\alpha,\alpha}^* d\Omega \quad , \end{aligned} \quad (3.28)$$

que é reescrita, usando (2.54b) e (2.54c), na forma

$$\begin{aligned} & - \int_{\Omega} (M_{\alpha\beta,\beta}^* - Q_{\alpha}^*) \psi_{\alpha} d\Omega - \int_{\Omega} Q_{\alpha,\alpha}^* u_3 d\Omega = \int_{\Omega} (\hat{q}u_3^* + m_{\alpha} \psi_{\alpha}^*) d\Omega + \\ & - m_f \int_{\Omega} q_3 \psi_{\alpha,\alpha}^* d\Omega + \int_{\Gamma} (t_3^l u_3^* + l_{\alpha} \psi_{\alpha}^*) d\Gamma - \int_{\Gamma} (u_3 t_3^* + \psi_{\alpha} l_{\alpha}^*) d\Gamma \quad . \end{aligned} \quad (3.29)$$

A substituição das equações de equilíbrio do estado auxiliar, eqs.(2.51b) e (2.51c), leva

⁴A demonstração formal pode ser encontrada em Westphal Jr. (1990, Apêndice B).

finalmente ao problema inverso:

$$\begin{aligned} & \int_{\Omega} G_{\alpha}^* \psi_{\alpha} d\Omega + \int_{\Omega} F_3^* u_3 d\Omega = \int_{\Omega} (\hat{q} u_3^* + m_{\alpha} \psi_{\alpha}^*) d\Omega + \\ & -m_f \int_{\Omega} q_3 \psi_{\alpha,\alpha}^* d\Omega + \int_{\Gamma} (t_3^l u_3^* + l_{\alpha} \psi_{\alpha}^*) d\Gamma - \int_{\Gamma} (u_3 t_3^* + \psi_{\alpha} l_{\alpha}^*) d\Gamma \quad . \end{aligned} \quad (3.30)$$

De forma similar à definição de F_{α}^* emprega-se

$$F_3^* = \delta(Q, P) e_3(P) \quad , \quad P, Q \in \Omega \quad , \quad (3.31a)$$

$$G_{\alpha}^* = \delta(Q, P) e_{\alpha}(P) \quad , \quad P, Q \in \Omega \quad , \quad (3.31b)$$

onde agora F_3^* representa uma carga concentrada unitária na direção e_3 enquanto G_{α}^* são momentos unitários correspondentes às direções e_{α} . Considerando cada carga atuando independentemente, tem-se

$$u_3^* = {}^f U_{i3}(Q, P) e_i(P) \quad (3.32a)$$

$$\psi_{\alpha}^* = {}^f U_{i\alpha}(Q, P) e_i(P) \quad (3.32b)$$

e

$$t_3^* = {}^f T_{i3}(Q, P) e_i(P) \quad (3.33a)$$

$$l_{\alpha}^* = {}^f T_{i\alpha}(Q, P) e_i(P) \quad , \quad (3.33b)$$

que inseridas em (3.30) fornecem as identidades de Somigliana para os deslocamentos de flexão:

$$\begin{aligned} u_3(P) + \int_{\Gamma} [{}^f T_{33}(Q, P) u_3(Q) + {}^f T_{3\alpha}(Q, P) \psi_{\alpha}(Q)] d\Gamma_Q = \\ = \int_{\Gamma} [{}^f U_{33}(Q, P) t_3^l(Q) + {}^f U_{3\alpha}(Q, P) l_{\alpha}(Q)] d\Gamma_Q + \\ + \int_{\Omega} [{}^f U_{33}(Q, P) \hat{q}(Q) + {}^f U_{3\alpha}(Q, P) m_{\alpha}(Q)] d\Omega_Q + \\ - m_f \int_{\Omega} [{}^f U_{3\alpha,\alpha}(Q, P) q_3(Q)] d\Omega_Q \end{aligned} \quad (3.34)$$

e

$$\begin{aligned}
\psi_\alpha(P) + \int_\Gamma [{}^f T_{\alpha 3}(Q, P)u_3(Q) + {}^f T_{\alpha\beta}(Q, P)\psi_\beta(Q)] d\Gamma_Q = \\
= \int_\Gamma [{}^f U_{\alpha 3}(Q, P)t_3^l(Q) + {}^f U_{\alpha\beta}(Q, P)l_\beta(Q)] d\Gamma_Q + \\
+ \int_\Omega [{}^f U_{\alpha 3}(Q, P)\hat{q}(Q) + {}^f U_{\alpha\beta}(Q, P)m_\beta(Q)] d\Omega_Q + \\
- m_f \int_\Omega [{}^f U_{\alpha\beta,\beta}(Q, P)q_3(Q)] d\Omega_Q \quad , \quad (3.35)
\end{aligned}$$

onde as derivadas presentes nas últimas integrais do lado direito são tomadas em relação a $x_\beta(Q)$.

3.4 Equações integrais sobre o contorno

As equações integrais (3.18), (3.34) e (3.35) são definidas para pontos P exclusivamente pertencentes a Ω , isto é, pontos internos. Por outro lado, a solução das mesmas requer o conhecimento dos deslocamentos e trações sobre o contorno Γ . A fim de obter tais relações válidas para pontos p sobre o contorno e possibilitar assim a solução prévia do problema de valores sobre o contorno, torna-se necessária a aplicação de um mapeamento linear e contínuo de $P \in \Omega \rightarrow p \in \Gamma$, especificamente a aplicação da propriedade do traço. Isso é realizado mediante um procedimento de limite, modificando o contorno próximo a p por um arco circular de raio ε centrado em p (figura 3-1).

Realizando-se o limite $\varepsilon \rightarrow 0$, obtém-se as identidades desejadas. Antes de mais nada, deve-se lembrar que os tensores fundamentais \mathbf{U} e \mathbf{T} , bem como suas derivadas, são funções de dois pontos, o ponto fonte P sobre o qual tem-se a singularidade da função generalizada delta de Dirac, e o ponto campo Q onde estão definidas as variáveis envolvidas nas equações. Assim, tais equações são funções somente da distância r entre P e Q , sendo esta expressa por (Brebbia et al. 1984):

$$r = \|Q - P\| \quad . \quad (3.36)$$

As expressões analíticas de todos os tensores utilizados constituem o conteúdo do Apêndice A. Todas as derivadas dos tensores fundamentais utilizados neste trabalho são

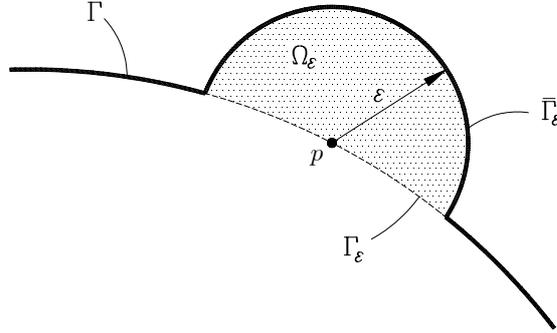


Figura 3-1: Modificação do contorno Γ para aplicação da propriedade do traço.

convencionadas em relação ao ponto campo Q , e portanto vale a notação:

$$U_{ij,\alpha} = \frac{\partial U_{ij}}{\partial x_\alpha(Q)} \quad .$$

Assim, quando houver alguma derivada em relação ao ponto fonte P é feito uso da relação:

$$\frac{\partial U_{ij}}{\partial x_\alpha(P)} = - \frac{\partial U_{ij}}{\partial x_\alpha(Q)} \quad ,$$

e então permanece válida a notação citada, sendo apenas um sinal negativo colocado à frente do termo que envolve tal derivada. Por esta razão, todas as derivadas de tensores fundamentais apresentados no Apêndice A são em relação ao ponto campo Q .

A inspeção das expressões analíticas dos tensores fundamentais leva a alguns comentários a respeito das singularidades envolvidas (quando $r \rightarrow 0$). Para o problema de membrana, eq.(3.18), verificam-se as singularidades $\ln r$ para o tensor ${}^m\mathbf{U}$, $1/r$ para o tensor ${}^m\mathbf{T}$ e $1/r$ para as derivadas cartesianas primeiras de ${}^m\mathbf{U}$. No problema de flexão, as singularidades são as mesmas, para os tensores ${}^f\mathbf{U}$ e ${}^f\mathbf{T}$, respectivamente. Devido a essas características, apenas as integrais no contorno que envolvem ${}^m\mathbf{T}$ ou ${}^f\mathbf{T}$ originam novos termos (termos de *salto* ou termos livres) quando se substitui Γ por $\Gamma - \Gamma_\varepsilon + \bar{\Gamma}_\varepsilon$ e procede-se o limite $\varepsilon \rightarrow 0$ (Brebbia et al. 1984). Assim, o traço será aplicado apenas aos

termos $\int_{\Gamma} \mathbf{T} \mathbf{u} d\Gamma$ nas equações (3.18), (3.34) e (3.35).

Então, para o problema de membrana, eq.(3.18), reescreve-se a primeira integral à esquerda como:

$$\begin{aligned}
\int_{\Gamma} {}^m T_{\alpha\beta}(q, p) u_{\beta}(q) d\Gamma_q &= \lim_{\varepsilon \rightarrow 0} \int_{\Gamma - \Gamma_{\varepsilon} + \bar{\Gamma}_{\varepsilon}} {}^m T_{\alpha\beta}(q, p) u_{\beta}(q) d\Gamma_q = \\
&= \lim_{\varepsilon \rightarrow 0} \int_{\Gamma - \Gamma_{\varepsilon}} {}^m T_{\alpha\beta}(q, p) u_{\beta}(q) d\Gamma_q + \\
&\quad + \lim_{\varepsilon \rightarrow 0} \int_{\bar{\Gamma}_{\varepsilon}} {}^m T_{\alpha\beta}(q, p) u_{\beta}(q) d\Gamma_q = \\
&= \lim_{\varepsilon \rightarrow 0} \int_{\Gamma - \Gamma_{\varepsilon}} {}^m T_{\alpha\beta}(q, p) u_{\beta}(q) d\Gamma_q + \\
&\quad + \lim_{\varepsilon \rightarrow 0} \int_{\bar{\Gamma}_{\varepsilon}} {}^m T_{\alpha\beta}(q, p) [u_{\beta}(q) - u_{\beta}(p)] d\Gamma_q + \\
&\quad + u_{\beta}(p) \left[\lim_{\varepsilon \rightarrow 0} \int_{\bar{\Gamma}_{\varepsilon}} {}^m T_{\alpha\beta}(q, p) d\Gamma_q \right]
\end{aligned}$$

Como os deslocamentos são contínuos, a segunda integral à direita da igualdade acima se anula. Então, reescreve-se o lado esquerdo de (3.18) na forma:

$$\begin{aligned}
u_{\alpha}(p) + \int_{\Gamma} {}^m T_{\alpha\beta}(q, p) u_{\beta}(q) d\Gamma_q &= u_{\alpha}(p) + u_{\beta}(p) \left[\lim_{\varepsilon \rightarrow 0} \int_{\bar{\Gamma}_{\varepsilon}} {}^m T_{\alpha\beta}(q, p) d\Gamma_q \right] + \\
&\quad + \int_{\Gamma} {}^m T_{\alpha\beta}(q, p) u_{\beta}(q) d\Gamma_q \quad , \quad (3.37)
\end{aligned}$$

sendo que o último termo à direita deve ser interpretado no sentido do valor principal de Cauchy, pois possui singularidade $(1/r)$ sobre um domínio de integração unidimensional. Substituindo (3.37) em (3.18) e definindo:

$${}^m C_{\alpha\beta} = \delta_{\alpha\beta} + \lim_{\varepsilon \rightarrow 0} \int_{\bar{\Gamma}_{\varepsilon}} {}^m T_{\alpha\beta}(q, p) d\Gamma_q \quad ,$$

reescreve-se a identidade de Somigliana para um ponto $P \in \Omega \cup \Gamma$ na forma:

$$\begin{aligned}
{}^m C_{\alpha\beta}(P)u_\beta(P) + \int_{\Gamma} {}^m T_{\alpha\beta}(q, P)u_\beta(q) d\Gamma_q &= \int_{\Gamma} {}^m U_{\alpha\beta}(q, P)t_\beta(q) d\Gamma_q + \\
&+ \int_{\Omega} {}^m U_{\alpha\beta}(Q, P)q_\beta(Q) d\Omega_Q - \int_{\Omega} {}^m U_{\alpha\beta,\gamma}(Q, P)N_{\beta\gamma}^n(Q) d\Omega_Q \quad .
\end{aligned} \tag{3.38}$$

onde

$${}^m C_{\alpha\beta} = \delta_{\alpha\beta} \quad \text{para } P \in \Omega \tag{3.39}$$

$${}^m C_{\alpha\beta} = \frac{1}{2}\delta_{\alpha\beta} \quad \text{para } P \in \Gamma \text{ suave.} \tag{3.40}$$

Para contornos que possuem vértices (descontinuidade da normal exterior), (3.40) assume valores dependentes do ângulo sólido (ver, por exemplo, Brebbia & Dominguez 1989). Chamando,

$${}^m \mathbf{u} = \{u_1 \quad u_2\}^T \tag{3.41a}$$

$${}^m \mathbf{t} = \{t_1 \quad t_2\}^T \tag{3.41b}$$

$${}^m \mathbf{q} = \{q_1 \quad q_2\}^T \quad , \tag{3.41c}$$

escreve-se a eq.(3.18) matricialmente na forma:

$$\begin{aligned}
{}^m \mathbf{u}(P) + \int_{\Gamma} {}^m \mathbf{T}(q, P){}^m \mathbf{u}(q) d\Gamma_q &= \int_{\Gamma} {}^m \mathbf{U}(q, P){}^m \mathbf{t}(q) d\Gamma_q + \\
&+ \int_{\Omega} {}^m \mathbf{V}(Q, P){}^m \mathbf{q}(Q) d\Omega_Q - \int_{\Omega} {}^m \hat{\mathbf{U}}(Q, P)\mathbf{N}^n(Q) d\Omega_Q \quad ,
\end{aligned} \tag{3.42}$$

onde

$${}^m \mathbf{V}(Q, P) = {}^m U_{\alpha\beta}(Q, P) \tag{3.43}$$

$${}^m \hat{\mathbf{U}}(Q, P) = {}^m U_{\alpha\beta,\gamma}(Q, P) = \frac{\partial {}^m U_{\alpha\beta}(Q, P)}{\partial x_\gamma(Q)} \tag{3.44}$$

e

$$\mathbf{N}^n(Q) = N_{\alpha\beta}^n(Q) = \begin{bmatrix} N_{11}^n(Q) & N_{12}^n(Q) \\ N_{21}^n(Q) & N_{22}^n(Q) \end{bmatrix} , \quad (3.45)$$

que são as parcelas não-lineares das tensões de membrana.

Caso existam n cargas concentradas aplicadas em pontos $Q = S^j$, $j = 1 \dots n$, deve-se utilizar a seguinte expressão para estes carregamentos (Kamiya & Sawaki 1985):

$${}^m\mathbf{q}(Q) = \sum_{j=1}^n {}^m\mathbf{f}(S^j)\delta(Q, S^j) \quad ,$$

onde

$${}^m\mathbf{f}(S^j) = \begin{Bmatrix} F_1(S^j) \\ F_2(S^j) \end{Bmatrix} . \quad (3.46)$$

Assim, além do termo de carregamento distribuído, segunda integral à direita de (3.42), deve-se adicionar o termo não-integral:

$${}^m\mathbf{v}(P) = \sum_{j=1}^n {}^m\mathbf{f}(S^j){}^m\mathbf{U}(S^j, P) \quad . \quad (3.47)$$

Portanto, a equação integral do problema de membrana fica completamente definida na forma⁵:

$$\begin{aligned} & {}^m\mathbf{C}(P){}^m\mathbf{u}(P) + \int_{\Gamma} {}^m\mathbf{T}(q, P){}^m\mathbf{u}(q) d\Gamma_q = \int_{\Gamma} {}^m\mathbf{U}(q, P){}^m\mathbf{t}(q) d\Gamma_q + \\ & + \int_{\Omega} {}^m\mathbf{V}(Q, P){}^m\mathbf{q}(Q) d\Omega_Q - \int_{\Omega} {}^m\hat{\mathbf{U}}(Q, P)\mathbf{N}^n(Q) d\Omega_Q + {}^m\mathbf{v}(P) \quad . \quad (3.48) \end{aligned}$$

Na expressão (3.48) a integral à esquerda deve ser interpretada no sentido do valor principal de Cauchy, enquanto a primeira e a última integrais à direita são fracamente singulares. Os demais termos são regulares.

Através de um procedimento algébrico análogo ao desenvolvido acima, obtém-se, para

⁵A segunda integral à direita de (3.48) pode ser transformada para o contorno em alguns casos simples de carregamento (Brebbia et al. 1984).

as eqs.(3.34) e (3.35), respectivamente:

$$\begin{aligned}
& {}^f C_{33}(P)u_3(P) + {}^f C_{3\alpha}(P)\psi_\alpha(P) + \int_{\Gamma} [{}^f T_{33}(q, P)u_3(q) + {}^f T_{3\alpha}(q, P)\psi_\alpha(q)] d\Gamma_q = \\
& = \int_{\Gamma} [{}^f U_{33}(q, P)t_3^l(q) + {}^f U_{3\alpha}(q, P)l_\alpha(q)] d\Gamma_q + \\
& + \int_{\Omega} [{}^f U_{33}(Q, P)\hat{q}(Q) + {}^f U_{3\alpha}(Q, P)m_\alpha(Q)] d\Omega_Q + \\
& - m_f \int_{\Omega} [{}^f U_{3\alpha,\alpha}(Q, P)q_3(Q)] d\Omega_Q \tag{3.49}
\end{aligned}$$

e

$$\begin{aligned}
& {}^f C_{\alpha 3}(P)u_3(P) + {}^f C_{\alpha\beta}(P)\psi_\beta(P) + \int_{\Gamma} [{}^f T_{\alpha 3}(q, P)u_3(q) + {}^f T_{\alpha\beta}(q, P)\psi_\beta(q)] d\Gamma_q = \\
& = \int_{\Gamma} [{}^f U_{\alpha 3}(q, P)t_3^l(q) + {}^f U_{\alpha\beta}(q, P)l_\beta(q)] d\Gamma_q + \\
& + \int_{\Omega} [{}^f U_{\alpha 3}(Q, P)\hat{q}(Q) + {}^f U_{\alpha\beta}(Q, P)m_\beta(Q)] d\Omega_Q + \\
& - m_f \int_{\Omega} [{}^f U_{\alpha\beta,\beta}(Q, P)q_3(Q)] d\Omega_Q \tag{3.50}
\end{aligned}$$

onde:

$${}^f C_{ij} = \delta_{ij} + \lim_{\varepsilon \rightarrow 0} \int_{\Gamma_\varepsilon} {}^f T_{ij}(q, p) d\Gamma_q \quad ,$$

sendo dado por:

$${}^f C_{ij} = \delta_{ij} \quad \text{para } P \in \Omega \tag{3.51}$$

$${}^f C_{ij} = \frac{1}{2}\delta_{ij} \quad \text{para } P \in \Gamma \text{ suave.} \tag{3.52}$$

Para pontos sobre vértices de Γ , as expressões de ${}^f C_{ij}$ podem ser encontradas na literatura (van der Weeën 1982a). Antes de escrever (3.49) e (3.50) em notação matricial, deve-se observar que os termos que envolvem as trações sobre o contorno consideram apenas a parcela linear das mesmas. Este fato pode ser explorado integrando-se por

partes o termo que envolve \hat{q} , o que resulta na identidade abaixo:

$$\begin{aligned} \int_{\Omega} {}^f U_{i3}(Q, P) \hat{q}(Q) d\Omega &= \int_{\Omega} {}^f U_{i3}(Q, P) q_3(Q) d\Omega + \\ &+ \int_{\Gamma} {}^f U_{i3}(q, P) t_3^n(q) d\Gamma - \int_{\Omega} {}^f U_{i3,\beta}(Q, P) N_{\alpha\beta} u_{3,\alpha}(Q) d\Omega \end{aligned} \quad (3.53)$$

onde foi utilizada a eq.(2.53b). Substituindo (3.53) em (3.49) e (3.50), reescreve-se estas duas últimas em uma única equação, usando a relação $t_3 = t_3^l + t_3^n$. Chamando

$${}^f \mathbf{u} = \{\psi_1 \quad \psi_2 \quad u_3\}^T \quad (3.54a)$$

$${}^f \mathbf{t} = \{l_1 \quad l_2 \quad t_3\}^T \quad (3.54b)$$

$${}^f \mathbf{q} = \{m_1 \quad m_2 \quad q_3\}^T \quad , \quad (3.54c)$$

este procedimento leva a:

$$\begin{aligned} {}^f \mathbf{C}(P) {}^f \mathbf{u}(P) + \int_{\Gamma} {}^f \mathbf{T}(q, P) {}^f \mathbf{u}(q) d\Gamma_q &= \int_{\Gamma} {}^f \mathbf{U}(q, P) {}^f \mathbf{t}(q) d\Gamma_q + \\ + \int_{\Omega} \left[{}^f \mathbf{U}(Q, P) - m_f \tilde{\mathbf{U}}(Q, P) \right] {}^f \mathbf{q}(Q) d\Omega_Q &- \int_{\Omega} {}^f \hat{\mathbf{U}}(Q, P) \mathbf{N}(Q) \bar{\mathbf{u}}(Q) d\Omega_Q \quad , \end{aligned}$$

ou

$$\begin{aligned} {}^f \mathbf{C}(P) {}^f \mathbf{u}(P) + \int_{\Gamma} {}^f \mathbf{T}(q, P) {}^f \mathbf{u}(q) d\Gamma_q &= \int_{\Gamma} {}^f \mathbf{U}(q, P) {}^f \mathbf{t}(q) d\Gamma_q + \\ + \int_{\Omega} {}^f \mathbf{V}(Q, P) {}^f \mathbf{q}(Q) d\Omega_Q &- \int_{\Omega} {}^f \hat{\mathbf{U}}(Q, P) \mathbf{N}(Q) \bar{\mathbf{u}}(Q) d\Omega_Q \quad , \end{aligned} \quad (3.55)$$

onde valem as relações abaixo:

$${}^f \hat{\mathbf{U}}(Q, P) = {}^f U_{i3,\alpha}(Q, P) = \frac{\partial {}^f U_{i3}(Q, P)}{\partial x_{\alpha}(Q)} \quad , \quad (3.56)$$

$$\mathbf{N}(Q) = N_{\alpha\beta}(Q) = \begin{bmatrix} N_{11}(Q) & N_{12}(Q) \\ N_{21}(Q) & N_{22}(Q) \end{bmatrix} \quad , \quad (3.57)$$

$$\tilde{\mathbf{U}}(Q, P) = {}^f U_{i\alpha, \alpha}(Q, P) = \frac{\partial {}^f U_{i\alpha}(Q, P)}{\partial x_\alpha(Q)} \quad , \quad (3.58)$$

$$\bar{\mathbf{u}}(Q) = \frac{\partial u_3(Q)}{\partial x_\alpha(Q)} = \left\{ u_{3,1}(Q) \quad u_{3,2}(Q) \right\}^T \quad , \quad (3.59)$$

$${}^f \mathbf{V}(Q, P) = {}^f \mathbf{U}(Q, P) - m_f \tilde{\mathbf{U}}(Q, P) \quad . \quad (3.60)$$

Note-se que os termos lineares da componente de tração t_3 da primeira integral à direita de (3.49) e (3.50) foram adicionados à parcela não-linear desta componente, presente no segundo termo à direita de (3.53).

A segunda integral à direita da equação (3.55), definidas sobre Ω , pode ser transformada para o contorno no caso de carregamentos distribuídos constantes. Tal transformação é encontradas na bibliografia (van der Weeën 1981, Westphal Jr. 1990), e leva à seguinte identidade:

$$\int_{\Omega} \left[{}^f \mathbf{U}(Q, P) - m_f \tilde{\mathbf{U}}(Q, P) \right] {}^f \mathbf{q}(Q) d\Omega_Q = q_3 \int_{\Gamma} {}^f \mathbf{Y}(q, p) \mathbf{n}(q) d\Gamma_q \quad , \quad (3.61)$$

para $\mathbf{q}(Q)$ constante. Onde

$$\mathbf{n}(q) = \left\{ n_1(q) \quad n_2(q) \right\}^T \quad , \quad (3.62)$$

e \mathbf{Y} é encontrado no Apêndice A.

Deve-se citar ainda o caso de carregamentos concentrados. Sejam m forças transversais ou momentos concentrados aplicadas em pontos $Q = S^j$, $i = 1 \dots m$, tem-se a seguinte expressão para o carregamento nestes casos (Kamiya & Sawaki 1985):

$${}^f \mathbf{q}(Q) = \sum_{j=1}^m {}^f \mathbf{f}(S^j) \delta(Q, S^j) \quad ,$$

sendo

$${}^f \mathbf{f}(s^j) = \left\{ \begin{array}{c} M_1(S^j) \\ M_2(S^j) \\ F_3(S^j) \end{array} \right\} \quad . \quad (3.63)$$

Logo, deve-se adicionar ao lado direito de (3.55) um termo não-integral correspondente às cargas concentradas, dado por:

$$\begin{aligned} {}^f\mathbf{v}(P) &= \sum_{j=1}^m {}^f\mathbf{f}(S^j) \left[{}^f\mathbf{U}(S^j, P) - m_f {}^f\hat{\mathbf{U}}(S^j, P) \right] \\ &= \sum_{j=1}^m {}^f\mathbf{f}(S^j) {}^f\mathbf{V}(S^j, P) \quad , \end{aligned} \quad (3.64)$$

resultando, para um ponto $P \in \Omega \cup \Gamma$:

$$\begin{aligned} {}^f\mathbf{C}(P) {}^f\mathbf{u}(P) + \int_{\Gamma} {}^f\mathbf{T}(q, P) {}^f\mathbf{u}(q) d\Gamma_q &= \int_{\Gamma} {}^f\mathbf{U}(q, P) {}^f\mathbf{t}(q) d\Gamma_q + \\ + \int_{\Omega} {}^f\mathbf{V}(Q, P) {}^f\mathbf{q}(Q) d\Omega_Q - \int_{\Omega} {}^f\hat{\mathbf{U}}(Q, P) \mathbf{N}(Q) \bar{\mathbf{u}}(Q) d\Omega_Q &+ {}^f\mathbf{v}(P) \quad . \end{aligned} \quad (3.65)$$

Na equação geral (3.65), a primeira integral à esquerda deve ser interpretada no sentido do valor principal de Cauchy. A primeira integral à direita é fracamente singular, enquanto a segunda é regular para o modelo de placa de Mindlin ($m_f = 0$) mas fracamente singular para o modelo de Reissner ($m_f = 1$). A última integral à direita possui singularidade $1/r$, mas como é definida sobre Ω , também é fracamente singular.

3.5 Equações integrais para flexão não-linear de placas

A análise do fenômeno de flexão não-linear de placas é particularmente importante no âmbito da engenharia devido às proporções geométricas díspares (comprimento \times largura \times espessura) que caracterizam estruturas compostas por placas. Tais proporções tendem a acentuar o efeito de imperfeições geométricas consequentes dos processos de fabricação e montagem, e tornam este tipo de estrutura particularmente suscetível à ocorrência de deslocamentos que impedem o uso das tradicionais hipóteses de pequenos deslocamentos. Problemas com estas características são tipicamente adjetivados de *geometricamente não-lineares* (Brush & Almroth 1975). Este é o contexto sob o qual o termo flexão não-linear deve ser entendido neste trabalho, isto é, não serão considerados quaisquer efeitos de

não-linearidade material.

As alternativas de solução numérica para este tipo de problema são dominadas pelo MEF para diversas classes estruturais (vigas, placas, cascas etc.). Em geral a solução de problemas não-lineares através do MEF utiliza descrições lagrangeanas atualizadas ou totais. Em se tratando do MEC, entretanto, não é muito comum a aplicação destas descrições mas é perfeitamente possível desenvolver as equações integrais de um problema na sua forma incremental (ver, por exemplo, Pietro et al. 1999).

Um caso particular decorrente da linearização das equações que governam o comportamento não-linear de uma estrutura é a análise de estabilidade. Estabilidade de estruturas é um assunto fartamente documentado na literatura, tanto para obtenção de soluções numéricas e analíticas como para a formalização dos conceitos físicos e matemáticos envolvidos no assunto (Brush & Almroth 1975).

O objetivo desta seção é apresentar na forma integral as equações que governam o fenômeno de flexão geometricamente não-linear de placas, utilizando as equações integrais derivadas nas seções anteriores. Como caso particular, serão também apresentadas as equações para análise de estabilidade linear de placas. As possibilidades de solução do sistema de equações decorrente serão discutidas no Capítulo 4. Para maior clareza, serão brevemente citados aqui alguns conceitos importantes sobre comportamento geometricamente não-linear de estruturas.

3.5.1 Generalidades

O comportamento não-linear de uma estrutura típica, carregada estaticamente, é ilustrado na figura 3-2. Esse comportamento é caracterizado por um parâmetro \mathbf{R} dependente do carregamento externo, e uma medida adequada de deslocamento, \mathbf{r} . A trajetória $\mathbf{R} \times \mathbf{r}$ é geralmente chamada *trajetória de equilíbrio* da estrutura. Os pontos B_1 e B_2 exemplificam os *pontos de bifurcação*, que levam a trajetórias de equilíbrio secundárias. Os pontos L_1 e L_2 ilustram os chamados *pontos limite*. Os pontos de bifurcação são localizações sobre a trajetória de equilíbrio que comportam mais de uma solução, enquanto um ponto limite indica uma configuração instável, e é meramente um caso patológico da trajetória de equilíbrio onde a derivada de \mathbf{R} em relação a \mathbf{r} se anula (Ramm & Stegmüller 1982, Kleiber 1989).

Os estados de equilíbrio são sempre enquadrados em uma das três possibilidades: estável, neutro ou instável. Em formulações variacionais obtidas a partir de um funcional como o MEF (ou em casos simples que possuem solução analítica) a estabilidade do equilíbrio pode ser numericamente analisada a partir da positividade da segunda variação do potencial total da estrutura (Π) em relação aos deslocamentos, $\delta^2\Pi/\delta\mathbf{r}^2$. Em formulações integrais já não é tão direta a avaliação numérica de quantidades energéticas.

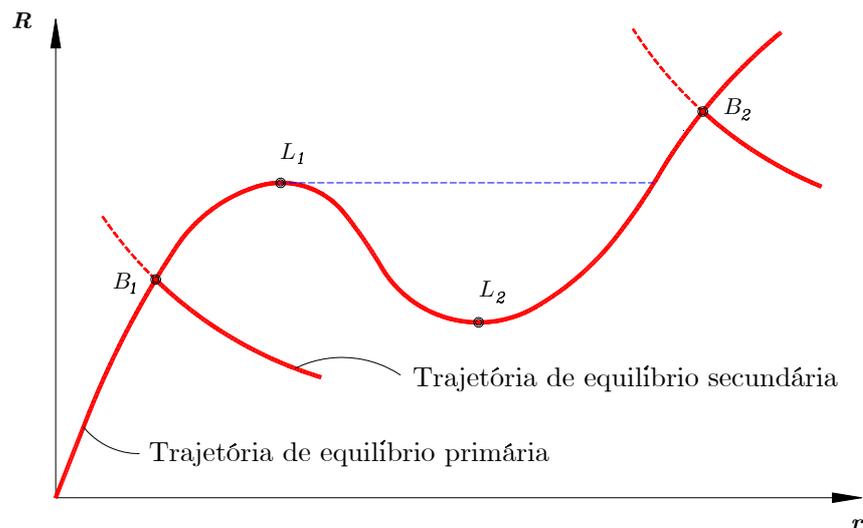


Figura 3-2: Comportamento não-linear típico de uma estrutura.

No contexto do presente trabalho, o interesse repousa sobre o levantamento de trajetórias de equilíbrio primárias, possivelmente até a ocorrência do primeiro ponto limite. A dificuldade inerente é justamente o maior ou menor grau de não-linearidade da curva $\mathbf{R} \times \mathbf{r}$, que obriga o uso de formulações incrementais e procedimentos iterativos para garantir que a solução obtida se encontra suficientemente próxima da trajetória verdadeira (Kleiber 1989).

Neste trabalho a análise de flexão não-linear considera placas perfeitamente planas antes dos carregamentos serem aplicados. O objetivo é obter a solução em termos de deslocamentos (e derivadas primeiras destes) e trações para vários níveis de carregamento aplicado até se alcançar o valor final do carregamento.

Entretanto, para uma grande parcela dos casos práticos requer-se apenas o nível de carregamento correspondente ao primeiro ponto de bifurcação. Tendo ainda em vista o

alto custo computacional envolvido na solução de problemas postos de forma incremental, é muitas vezes satisfatório apenas o conhecimento de informações parciais do problema. Tais informações são obtidas a partir da linearização das equações do problema, levando a uma simples análise de perturbação em torno do ponto de bifurcação. Esta abordagem é chamada, na Mecânica dos Sólidos, de análise clássica de estabilidade linear e leva a um problema de autovalores e autovetores que representa o problema de flambagem (Ramm & Stegmüller 1982).

Para a análise de estabilidade de placas, normalmente são consideradas placas perfeitamente planas, submetidas somente a um estado de tensões de membrana, antes da flambagem ocorrer. O objetivo é obter a menor carga \mathbf{N}^* (carga crítica) que provoca a flambagem quando a placa é ligeiramente perturbada (figura 3-3).

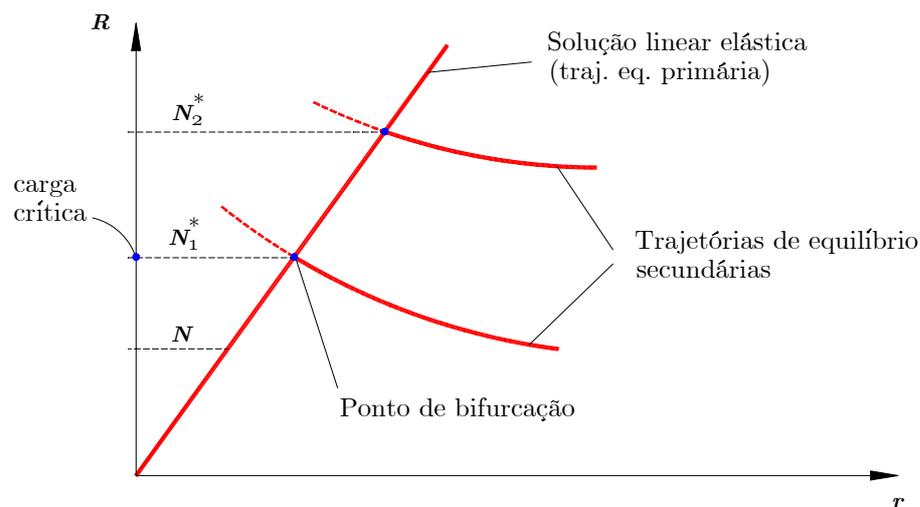


Figura 3-3: Análise linearizada de flambagem.

Fazendo uso do comportamento linear da estrutura (figura 3-3), é imediato escrever as cargas de flambagem em função de um estado de tensões qualquer \mathbf{N} sobre a trajetória de equilíbrio primária:

$$\mathbf{N}^* = \lambda \mathbf{N} \quad , \quad (3.66)$$

onde λ é um fator de carga a ser determinado.

3.5.2 Equações integrais para flexão geometricamente não-linear de placas

As equações (3.48) e (3.65) serão adotadas aqui como ponto de partida para uma formulação incremental escrita na forma integral. A partir dessas equações é possível descrever o comportamento de flambagem e pós-flambagem por métodos integrais. Reescrevendo estas equações em notação indicial e matricial, respectivamente, tem-se:

$$\begin{aligned} {}^m C_{\alpha\beta}(P)u_\beta(P) + \int_{\Gamma} {}^m T_{\alpha\beta}(q, P)u_\beta(q) d\Gamma_q &= \int_{\Gamma} {}^m U_{\alpha\beta}(q, P)t_\beta(q) d\Gamma_q + \\ &+ \int_{\Omega} {}^m V_{\alpha\beta}(Q, P)q_\beta(Q) d\Omega_Q - \int_{\Omega} {}^m U_{\alpha\beta,\gamma}(Q, P)N_{\beta\gamma}^n(Q) d\Omega_Q + {}^m v_\alpha(P) \quad , \end{aligned} \quad (3.67)$$

ou

$$\begin{aligned} {}^m \mathbf{C}(P){}^m \mathbf{u}(P) + \int_{\Gamma} {}^m \mathbf{T}(q, P){}^m \mathbf{u}(q) d\Gamma_q &= \int_{\Gamma} {}^m \mathbf{U}(q, P){}^m \mathbf{t}(q) d\Gamma_q + \\ &+ \int_{\Omega} {}^m \mathbf{V}(Q, P){}^m \mathbf{q}(Q) d\Omega_Q - \int_{\Omega} {}^m \hat{\mathbf{U}}(Q, P)\mathbf{N}^n(Q) d\Omega_Q + {}^m \mathbf{v}(P) \quad . \end{aligned} \quad (3.68)$$

para o problema de membrana, e

$$\begin{aligned} {}^f C_{ij}(P)u_j(P) + \int_{\Gamma} {}^f T_{ij}(q, P)u_j(q) d\Gamma_q &= \int_{\Gamma} {}^f U_{ij}(q, P)t_j(q) d\Gamma_q + \\ &+ \int_{\Omega} {}^f V_{ij}(Q, P)q_j(Q) d\Omega_Q - \int_{\Omega} {}^f U_{i3,\beta}(Q, P)N_{\alpha\beta}(Q)u_{3,\alpha}(Q) d\Omega_Q + {}^f v_i(P) \quad , \end{aligned} \quad (3.69)$$

ou

$$\begin{aligned} {}^f \mathbf{C}(P){}^f \mathbf{u}(P) + \int_{\Gamma} {}^f \mathbf{T}(q, P){}^f \mathbf{u}(q) d\Gamma_q &= \int_{\Gamma} {}^f \mathbf{U}(q, P){}^f \mathbf{t}(q) d\Gamma_q + \\ &+ \int_{\Omega} {}^f \mathbf{V}(Q, P){}^f \mathbf{q}(Q) d\Omega_Q - \int_{\Omega} {}^f \hat{\mathbf{U}}(Q, P)\mathbf{N}(Q)\bar{\mathbf{u}}(Q) d\Omega_Q + {}^f \mathbf{v}(P) \quad . \end{aligned} \quad (3.70)$$

para o problema de flexão.

Cabem agora alguns comentários a respeito destas equações.

1. **Quanto às singularidades envolvidas:** a integral à esquerda e a primeira integral à direita de ambas as equações são fortemente singular ($1/r$ sobre Γ) e fracamente singular ($\ln r$ sobre Γ), respectivamente. A última integral à direita de (3.67) é fracamente singular ($1/r$ sobre Ω). Notando a relação (3.60), resulta que a segunda integral à direita de (3.69) é fracamente singular ($1/r$ sobre Ω) somente no caso do modelo de placa de Reissner ser utilizado, caso contrário é regular ($\ln r$ sobre Ω). A última integral à direita de (3.69) também é fracamente singular ($1/r$ sobre Ω). Demais integrais são regulares. Finalmente, os termos de carregamentos concentrados ${}^m\mathbf{v}(P)$ e ${}^f\mathbf{v}(P)$ possuem singularidades logarítmicas (exceto para o modelo de placa de Reissner, quando ${}^f\mathbf{v}(P)$ manifesta singularidade do tipo $1/r$)⁶.

2. **Quanto à não-linearidade do problema:** a não linearidade do problema está explicitada nos termos de acoplamento flexão-membrana oriundos da utilização do tensor deformação de Green contendo os termos de segunda ordem - eqs.(2.6a). Deve-se notar que o problema de flexão está inserido no problema de membrana através da parcela não-linear de esforços de membrana $N_{\alpha\beta}^n$, que contém derivadas cartesianas do deslocamento transversal (curvaturas), como mostra a eq.(2.22b). Por sua vez, a eq.(3.69) contém os esforços de membrana totais $N_{\alpha\beta}$ também multiplicados por curvaturas na sua última integral à direita. Dado que $N_{\alpha\beta}$ contém também as derivadas cartesianas dos deslocamentos de membrana, torna-se necessário o cálculo de de todas as derivadas dos deslocamentos translacionais ($u_{i,\alpha}$) para se efetuar este acoplamento. É necessário destacar que quando uma placa sob flexão é deformada segundo uma superfície que não pode ser desenvolvida, ocorrerão tensões de membrana mesmo na ausência de carregamentos paralelos ao plano da placa. Este acoplamento é o aspecto que diferencia a presente abordagem da análise linear, quando as equações (3.67) e (3.69) são consideradas desacopladas.

3. **Quanto à recursividade da solução:** uma vez resolvido o problema de valores no contorno pela eq.(3.67), pode-se obter os deslocamentos e esforços em pontos

⁶Portanto, mesmo não necessitando de integração, estes termos devem ser manipulados com cuidado quando o ponto fonte P se encontra muito próximo dos pontos de aplicação das cargas concentradas (S^i).

internos. O cálculo das parcelas não-lineares dos esforços de membrana, entretanto, necessitam das derivadas cartesianas do deslocamento transversal $u_{3,\alpha}$. Mas u_3 só pode ser avaliado após a solução da eq.(3.69) no contorno e em pontos internos, e esta por sua vez depende dos esforços de membrana. Assim, torna-se clara a recursividade exigida para solução das equações (3.67) e (3.69).

Os dois últimos aspectos acima evidenciam a necessidade de se calcular as derivadas cartesianas dos deslocamentos lateral e transversal da placa. Em métodos como o MEF, é comum o uso direto das derivadas da forma interpolatória para uma variável u , isto é, $u_{,\alpha} = \phi_{i,\alpha} u_i$, onde ϕ_i são as função utilizadas para interpolar os valores nodais u_i . Embora simplifique sobremaneira o problema, incorre-se no empobrecimento da ordem de interpolação das derivadas da variável em relação à forma interpolatória originalmente utilizada. Ou seja, há um enfraquecimento da solução global para as derivadas da variável em questão.

No presente caso não há necessidade de se assumir de antemão uma forma interpolatória para os deslocamentos, uma vez que uma forma forte da solução para estas variáveis está disponível: trata-se das eqs.(3.67) e da terceira das equações (3.69). Portanto uma solução mais rigorosa pode ser obtida por derivação das equações integrais para u_i em (3.67) e (3.69) em relação às coordenadas $x_\alpha(P)$. Este procedimento leva a três equações integrais adicionais para cálculo preciso de $u_{i,\alpha}$.

Pressupondo-se que estas derivadas serão calculadas apenas em pontos internos, a diferenciação das integrais definidas sobre Γ é direta, pois neste caso os respectivos núcleos são regulares. No entanto, a diferenciação das duas última integrais à direita de (3.67) e (3.69) não o é, pois \mathbf{V} e $\hat{\mathbf{U}}$ podem possuir singularidades fracas⁷, como já detectado, quando $Q \equiv P$. Infelizmente, a diferenciação de integrais com núcleos singulares não obedece às regras clássicas de diferenciação, merecendo um tratamento especial mediante a aplicação da fórmula de Leibnitz (Bui 1978, Brebbia et al. 1984). Este fato foi ignorado na literatura até o final da década de setenta, quando Bui (1978) apresentou as equações integrais corrigidas para problemas de plasticidade destacando esta omissão. O desenvolvimento formal para a diferenciação das integrais em Ω de (3.67) e (3.69) é

⁷Conforme já salientado, a integral contendo ${}^f\mathbf{V}$ só manifesta singularidade fraca para o modelo de placa de Reissner. Caso o Modelo de Mindlin seja adotado, esta integral é regular. Além disso, ${}^m\mathbf{V}$ é sempre regular.

apresentado no Apêndice B, e origina o aparecimento de *termos convectivos* (Perez & Wrobel 1991, Brebbia et al. 1984), que são adicionado à expressão final para $u_{\beta,\alpha}(P)$ e $u_{3,\alpha}(P)$, escritas a seguir:

$$\begin{aligned}
u_{\beta,\alpha}(P) - \int_{\Gamma} {}^m T_{\beta\gamma,\alpha}(q, P) u_{\gamma}(q) d\Gamma_q &= - \int_{\Gamma} {}^m U_{\beta\gamma,\alpha}(q, P) t_{\gamma}(q) d\Gamma_q + \\
&- \int_{\Omega} {}^m V_{\beta\gamma,\alpha}(Q, P) q_{\gamma}(Q) d\Omega_Q + \int_{\Omega} {}^m U_{\beta\gamma,\delta\alpha}(Q, P) N_{\gamma\delta}^n(Q) d\Omega_Q + \\
&+ N_{\gamma\delta}^n(P) \int_{\Gamma'_1} {}^m U_{\beta\gamma,\delta}(Q, P) r_{,\alpha}(P) d\Gamma_{Q_1} + \\
&- q_{\gamma}(P) \int_{\Gamma'_1} {}^m V_{\beta\gamma}(Q, P) r_{,\alpha}(P) d\Gamma_{Q_1} - {}^m v_{\beta,\alpha}(P) \quad , \quad P \in \Omega, \quad (3.71)
\end{aligned}$$

$$\begin{aligned}
u_{3,\alpha}(P) - \int_{\Gamma} {}^f T_{3i,\alpha}(q, P) u_i(q) d\Gamma_q &= - \int_{\Gamma} {}^f U_{3i,\alpha}(q, P) t_i(q) d\Gamma_q + \\
&- \int_{\Omega} V_{3i,\alpha}(Q, P) q_i(Q) d\Omega_Q + \int_{\Omega} {}^f U_{33,\alpha\gamma}(Q, P) N_{\beta\gamma}(Q) u_{3,\beta}(Q) d\Omega_Q + \\
&+ N_{\beta\gamma}(P) u_{3,\beta}(P) \int_{\Gamma'_1} {}^f U_{33,\gamma}(Q, P) r_{,\alpha}(P) d\Gamma_{Q_1} + \\
&- m_f q_i(P) \int_{\Gamma'_1} {}^f V_{3i}(Q, P) r_{,\alpha}(P) d\Gamma_{Q_1} - {}^f v_{3,\alpha}(P) \quad , \quad P \in \Omega, \quad (3.72)
\end{aligned}$$

sendo o sinal negativo incorporado a todas as integrais, conforme a convenção adotada, pois as derivadas foram agora tomadas em relação a $x_{\alpha}(P)$.

Note-se que $P \in \Omega$ e portanto $C_{ij} = \delta_{ij}$. Os termos integrais sobre Γ'_1 de (3.71) e (3.72) são os referidos termos convectivos (Γ'_1 é uma circunferência de raio unitário centrada em P , como descrito no Apêndice B):

$$\begin{aligned}
{}^m c_{\alpha\beta}^N(P) &= N_{\gamma\delta}^n(P) \int_{\Gamma'_1} {}^m U_{\beta\gamma,\delta}(Q, P) r_{,\alpha}(P) d\Gamma_{Q_1} \\
{}^m c_{\alpha\beta}^q(P) &= q_{\gamma}(P) \int_{\Gamma'_1} {}^m V_{\beta\gamma}(Q, P) r_{,\alpha}(P) d\Gamma_{Q_1} \\
{}^f c_{\alpha\beta}^N(P) &= N_{\beta\gamma}(P) u_{3,\beta}(P) \int_{\Gamma'_1} {}^f U_{33,\gamma}(Q, P) r_{,\alpha}(P) d\Gamma_{Q_1} \\
{}^f c_{\alpha}^q(P) &= m_f q_i(P) \int_{\Gamma'_1} {}^f V_{3i}(Q, P) r_{,\alpha}(P) d\Gamma_{Q_1} \quad .
\end{aligned}$$

Estes termos são obtidos após um intenso trabalho analítico, e são respectivamente

dados por (ver Apêndice B):

$${}^m c_{\alpha\beta}^N(P) = \frac{-1}{8G(1-\nu)} \left[(3-4\nu) \delta_{\alpha\delta} \delta_{\beta\gamma} - \delta_{\alpha\beta} \delta_{\gamma\delta} - \delta_{\alpha\gamma} \delta_{\beta\delta} + \right. \\ \left. - \frac{1}{4} \delta_{\alpha\beta} \delta_{\gamma\delta} (1+2\delta_{\alpha\gamma}) \right] N_{\gamma\delta}^n(P) \quad , \quad (3.73a)$$

$${}^m c_{\alpha\beta}^q(P) = 0 \quad , \quad (3.73b)$$

$${}^f c_{\alpha\beta}^N(P) = -\frac{\delta_{\alpha\gamma}}{D(1-\nu)\lambda^2} N_{\gamma\beta}(P) \quad , \quad (3.73c)$$

$${}^f c_{\alpha}^q(P) = 0 \quad . \quad (3.73d)$$

Desta forma, as eqs.(3.71) e (3.72) serão doravante escritas:

$$u_{\beta,\alpha}(P) - \int_{\Gamma} {}^m T_{\beta\gamma,\alpha}(q,P) u_{\gamma}(q) d\Gamma_q = - \int_{\Gamma} {}^m U_{\beta\gamma,\alpha}(q,P) t_{\gamma}(q) d\Gamma_q + \\ - \int_{\Omega} {}^m V_{\beta\gamma,\alpha}(Q,P) q_{\gamma}(Q) d\Omega_Q + \int_{\Omega} {}^m U_{\beta\gamma,\delta\alpha}(Q,P) N_{\gamma\delta}^n(Q) d\Omega_Q + \\ + {}^m c_{\alpha\beta}^N(P) - {}^m v_{\beta,\alpha}(P) \quad , \quad P \in \Omega \quad (3.74)$$

$$u_{3,\alpha}(P) - \int_{\Gamma} {}^f T_{3i,\alpha}(q,P) u_i(q) d\Gamma_q = - \int_{\Gamma} {}^f U_{3i,\alpha}(q,P) t_i(q) d\Gamma_q + \\ - \int_{\Omega} V_{3i,\alpha}(Q,P) q_i(Q) d\Omega_Q + \int_{\Omega} {}^f U_{33,\alpha\gamma}(Q,P) N_{\gamma\beta}(Q) u_{3,\beta}(Q) d\Omega_Q + \\ + c_{\alpha\beta}^N(P) u_{3,\beta}(P) - {}^f v_{3,\alpha}(P) \quad , \quad P \in \Omega \quad (3.75)$$

ou, matricialmente:

$$\bar{\mathbf{u}}_{\alpha}(P) - \int_{\Gamma} {}^m \bar{\mathbf{T}}(q,P) {}^m \mathbf{u}(q) d\Gamma_q = - \int_{\Gamma} {}^m \bar{\mathbf{U}}(q,P) {}^m \mathbf{t}(q) d\Gamma_q + \\ - \int_{\Omega} {}^m \bar{\mathbf{V}}(Q,P) {}^m \mathbf{q}(Q) d\Omega_Q + \int_{\Omega} {}^m \bar{\mathbf{U}}(Q,P) \mathbf{N}^n(Q) d\Omega_Q + \\ + {}^m \mathbf{c}^N(P) - {}^m \bar{\mathbf{v}}(P) \quad , \quad (3.76)$$

$$\begin{aligned}
\bar{\mathbf{u}}_3(P) - \int_{\Gamma} {}^f \bar{\mathbf{T}}(q, P) {}^f \mathbf{u}(q) d\Gamma_q &= - \int_{\Gamma} {}^f \bar{\mathbf{U}}(q, P) {}^f \mathbf{t}(q) d\Gamma_q + \\
&- \int_{\Omega} {}^f \bar{\mathbf{V}}(Q, P) {}^f \mathbf{q}(Q) d\Omega_Q + \int_{\Omega} {}^f \bar{\bar{\mathbf{U}}}(Q, P) \mathbf{N}(Q) {}^f \bar{\mathbf{u}}(Q) d\Omega_Q \\
&+ {}^f \mathbf{c}^N(P) {}^f \bar{\mathbf{u}}(P) - {}^f \bar{\mathbf{v}}(P) \quad , \tag{3.77}
\end{aligned}$$

onde

$$\bar{\mathbf{u}}_{\alpha} = \{u_{\alpha,1} \quad u_{\alpha,2}\}^T \tag{3.78a}$$

$$\bar{\mathbf{u}}_3 = \{u_{3,1} \quad u_{3,2}\}^T \tag{3.78b}$$

e

$${}^m \bar{\mathbf{v}}(P) = {}^m v_{\beta,\alpha}(P) \tag{3.79a}$$

$${}^f \bar{\mathbf{v}}(P) = {}^f v_{3,\alpha}(P) \quad , \tag{3.79b}$$

e ainda

$${}^m \bar{\mathbf{U}}(q, P) = {}^m U_{\beta\gamma,\alpha}(q, P) = \frac{\partial^m U_{\beta\gamma}(q, P)}{\partial x_{\alpha}(q)} \quad , \tag{3.80a}$$

$${}^f \bar{\mathbf{U}}(q, P) = {}^f U_{3i,\alpha}(q, P) = \frac{\partial^f U_{3i}(q, P)}{\partial x_{\alpha}(q)} \quad , \tag{3.80b}$$

$${}^m \bar{\mathbf{T}}(q, P) = {}^m T_{\beta\gamma,\alpha}(q, P) = \frac{\partial^m T_{\beta\gamma}(q, P)}{\partial x_{\alpha}(q)} \quad , \tag{3.80c}$$

$${}^f \bar{\mathbf{T}}(q, P) = {}^f T_{3i,\alpha}(q, P) = \frac{\partial^f T_{3i}(q, P)}{\partial x_{\alpha}(q)} \quad , \tag{3.80d}$$

$${}^m \bar{\bar{\mathbf{U}}}(Q, P) = {}^m U_{\beta\gamma,\delta\alpha}(Q, P)(Q, P) = \frac{\partial^m U_{\beta\gamma}(Q, P)}{\partial x_{\delta}(Q) x_{\alpha}(Q)} \quad , \tag{3.80e}$$

$${}^f \bar{\bar{\mathbf{U}}}(Q, P) = {}^f U_{33,\alpha\beta}(Q, P) = \frac{\partial^f U_{33}(Q, P)}{\partial x_{\alpha}(Q) x_{\beta}(Q)} \quad , \tag{3.80f}$$

$${}^m \bar{\mathbf{V}}(Q, P) = {}^m V_{\beta\gamma,\alpha}(Q, P) = \frac{\partial^m V_{\beta\gamma}(Q, P)}{\partial x_{\alpha}(Q)} \quad , \tag{3.80g}$$

$${}^f \bar{\mathbf{V}}(Q, P) = {}^f V_{3i,\alpha}(Q, P) = \frac{\partial^f V_{3i}(Q, P)}{\partial x_{\alpha}(Q)} \quad . \tag{3.80h}$$

Observam-se as singularidades $1/r^2$ para $\bar{\mathbf{T}}$, $1/r$ para $\bar{\mathbf{U}}$ e $1/r^2$ para $\bar{\bar{\mathbf{U}}}$. Devido à dimensão dos domínios sobre os quais estão definidas as integrais destes núcleos, as integrais à esquerda de (3.71) e (3.72) devem ser interpretada no sentido do valor principal

de Haddamard, o que demonstra o caráter *hipersingular* destas equações. As demais integrais devem ser todas interpretadas no sentido do valor principal de Cauchy. Observe-se novamente que as eqs.(3.71) e (3.72) são válidas para pontos exclusivamente interiores, caso contrário, o traço deve ser aplicado sobre as mesmas, como para as equações definidas sobre o contorno, ou ainda utilizar-se uma outra alternativa para cálculo das derivadas de u_i para pontos sobre Γ .

3.5.3 Equações integrais para análise de estabilidade elástica de placas

As equações (3.67), (3.69) (3.71) e (3.72) podem agora ser particularizadas para a análise de estabilidade linear elástica de placas⁸. Tendo em vista a eq.(3.66), a eq.(3.67) não é utilizada devido a linearidade assumida na trajetória de equilíbrio pré-flambagem. Como não existem carregamentos transversais ou momentos, reescreve-se (3.69) e (3.72) na forma:

$$\begin{aligned} {}^f C_{ij}(P)u_j(P) + \int_{\Gamma} {}^f T_{ij}(q, P)u_j(q) d\Gamma_q = \int_{\Gamma} {}^f U_{ij}(q, P)t_j(q) d\Gamma_q + \\ -\lambda \int_{\Omega} {}^f U_{i3,\beta}(Q, P)N_{\alpha\beta}(Q)u_{3,\alpha}(Q) d\Omega_Q \quad , \end{aligned} \quad (3.81)$$

e

$$\begin{aligned} u_{3,\alpha}(P) - \int_{\Gamma} {}^f T_{3i,\alpha}(q, P)u_i(q) d\Gamma_q = - \int_{\Gamma} {}^f U_{3i,\alpha}(q, P)t_i(q) d\Gamma_q + \\ +\lambda \int_{\Omega} {}^f U_{33,\alpha\gamma}(Q, P)N_{\beta\gamma}(Q)u_{3,\beta}(Q) d\Omega_Q + \\ +\lambda c_{\alpha\beta}^N(P)u_{3,\beta}(P) \quad , \quad P \in \Omega. \end{aligned} \quad (3.82)$$

Matricialmente, as eqs.(3.81) e (3.82) são escritas (como as equações do problema de

⁸É oportuno salientar que isso pode ser realizado de duas formas: ou se lineariza as equações de von Kármán e se repete a dedução das equações integrais, ou simplesmente se particulariza as equações integrais não-lineares. Este último será o procedimento aqui adotado.

membrana não são utilizadas, o prefixo f foi omitido das equações a seguir):

$$\begin{aligned} \mathbf{C}(P)\mathbf{u}(P) + \int_{\Gamma} \mathbf{T}(q, P)\mathbf{u}(q) d\Gamma_q = \int_{\Gamma} \mathbf{U}(q, P)\mathbf{t}(q) d\Gamma_q + \\ -\lambda \int_{\Omega} \mathbf{U}(Q, P)\mathbf{N}(Q)\bar{\mathbf{u}}(Q) d\Omega_Q \quad . \end{aligned} \quad (3.83)$$

$$\begin{aligned} \bar{\mathbf{u}}_3(P) - \int_{\Gamma} \bar{\mathbf{T}}(q, P)\mathbf{u}(q) d\Gamma_q = - \int_{\Gamma} \bar{\mathbf{U}}(q, P)\mathbf{t}(q) d\Gamma_q + \\ +\lambda \int_{\Omega} \bar{\mathbf{U}}(Q, P)\mathbf{N}(Q)\bar{\mathbf{u}}(Q) d\Omega_Q + \lambda \mathbf{c}^N(P)\bar{\mathbf{u}}(P) \quad , \end{aligned} \quad (3.84)$$

3.6 Resumo do Capítulo 3

Neste capítulo foi apresentado inicialmente o formalismo geral para obtenção do problema inverso governado por uma dada equação diferencial. Tal formalismo permite a clara identificação do método dos resíduos ponderados, que foi aplicado aos operadores diferenciais da elasticidade bidimensional e dos modelos de placa descritos no Capítulo 2. Foram obtidas assim as identidades de Somigliana para os deslocamentos do problema de membrana e do problema de flexão. A aplicação da propriedade do traço sobre essas equações levou aos problemas de valores sobre o contorno correspondentes. A necessidade de equações para cálculo das derivadas cartesianas dos deslocamentos translacionais da placa gerou três equações integrais hipersingulares adicionais. Estas foram derivadas levando-se em conta os termos livres provenientes da aplicação da fórmula de Leibnitz, e são adicionada ao sistema de equações de membrana e de flexão unificado, dispensando o uso de derivadas numéricas. Assim foi obtido o sistema de equações integrais unificado para análise de flexão geometricamente não-linear dos modelos de placa de Mindlin e de Reissner. Finalmente, as equações integrais foram particularizadas para a análise linearizada de estabilidade elástica de placas.

As equações (3.68), (3.70), (3.76), (3.77), (3.83), e (3.84) serão discretizadas no capítulo seguinte, através da metodologia padrão do MEC direto. Após alguns algebrismos, estas equações levam a dois sistemas de equações distintos. Nos problemas de estabilidade, obtém-se um sistema de autovalores e autovetores que representa o problema de flambagem. Após a solução do mesmo obtém-se, a partir dos autovalores λ , o valor das

cargas de flambagem via equação (3.66). Na análise de flexão não-linear, obtém-se um sistema de equações acopladas nas variáveis $m_{\mathbf{u}}$, $m_{\mathbf{t}}$, $f_{\mathbf{u}}$, $f_{\mathbf{t}}$, e $\bar{\mathbf{u}}$. Um procedimento para solução iterativa deste sistema será especialmente desenvolvido, para cada nível de carregamento.

Capítulo 4

Implementação Numérica

Este capítulo apresenta o procedimento de solução numérica das equações integrais desenvolvidas no Capítulo Três, através do uso da metodologia padrão do método direto dos elementos de contorno (Banerjee & Butterfield 1981, Brebbia et al. 1984). Inicialmente as equações são escritas na forma discretizada utilizando o conceito de interpolação paramétrica para a geometria e as variáveis do problema. Um algoritmo de solução iterativa das equações integrais apresentadas no capítulo anterior é desenvolvido, e alguns aspectos de sua implementação são abordados. O tratamento de subregiões é detalhado, buscando um procedimento aplicável a um número arbitrário de subregiões. Finalmente, são enfatizados alguns aspectos sobre os procedimentos de integração adotados para os vários graus de singularidade presentes nos tensores envolvidos.

Com o objetivo de facilitar a dedução das equações em sua forma discretizada, serão aqui apresentadas algumas convenções adotadas no restante deste capítulo. Seja um domínio de solução de dimensão n_d que se deseja discretizar para aplicação do MEC, conforme ilustrado na figura 4-1 para o caso bidimensional ($n_d = 2$):

- O contorno Γ é dividido em N elementos $(n_d - 1)$ -dimensionais Γ_n , tal que $\cup_{n=1}^N \Gamma_n = \Gamma$.
- O domínio Ω é dividido em R células n_d -dimensionais Ω_r , tal que $\cup_{r=1}^R \Omega_r \leq \Omega$.

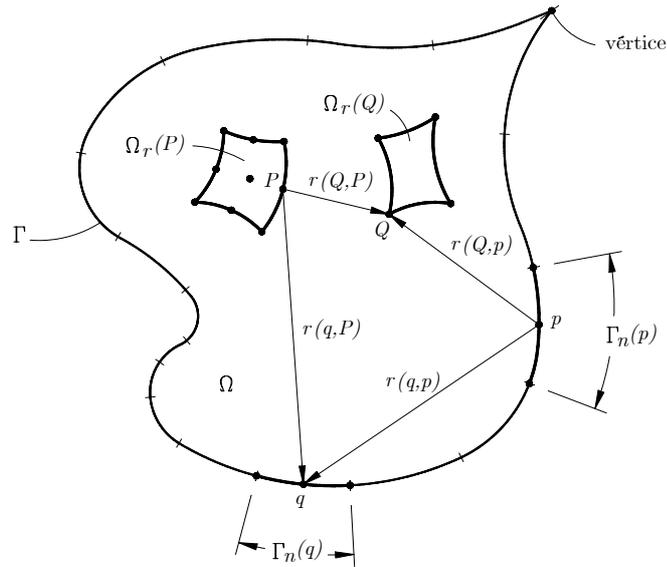


Figura 4-1: Convenções para discretização de Γ e Ω de um domínio bidimensional.

4.1 Interpolação de geometria, variáveis e carregamentos

A construção de elementos de contorno e células de domínio pode ser bastante flexibilizada através da utilização não apenas de interpolações isoparamétricas, mas também de suas variações sub e superparamétricas. O caso mais simples e natural é o do elemento de contorno constante, onde a geometria é definida por dois pontos, mas as variáveis são calculadas em um único ponto central. Portanto a geometria possui aproximação linear enquanto as variáveis possuem aproximação constante, caracterizando um elemento superparamétrico. Este conceito pode ser estendido dependendo da razão entre a precisão desejada na aproximação da geometria e a precisão desejada na aproximação das variáveis físicas do problema.

Neste trabalho serão empregadas aproximações diferentes para a geometria e para as variáveis. A interpolação da geometria é governada por uma *partição geométrica*, enquanto a interpolação das variáveis físicas é governada por uma *partição física*. A partição geométrica é definida através de n_p pontos geométricos e a partição física é definida por n_n nós (figura 4-2)¹. Então, os elementos de contorno têm sua geometria

¹A distinção entre pontos e nós fica evidente no contexto em que são usados. Ambos são coordenadas

interpolada por um polinômio de grau $(n_p - 1)^{n_d - 1}$ enquanto as variáveis o são por um polinômio de grau $(n_n - 1)^{n_d - 1}$. Como não necessariamente $n_n = n_p$, um elemento de contorno genérico Γ_n pode ser construído com um grande número de possibilidades, como exemplificado na figura 4-2 para o caso bidimensional.

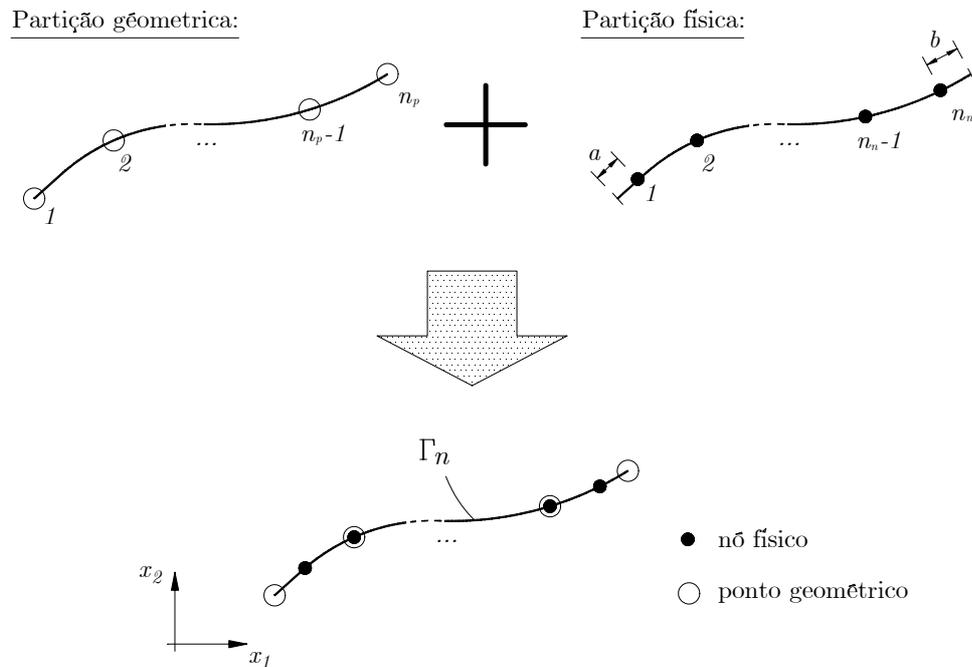


Figura 4-2: Construção de um elemento de contorno bidimensional genérico através da superposição de suas partições física e geométrica.

Deve ser observado que esta forma de construção de elementos discretos não impõe muitas restrições sobre a topologia da partição física. Pode-se tirar proveito disso em diversas situações. Elementos descontínuos, por exemplo, são facilmente obtidos fazendo-se $a \neq 0$ e/ou $b \neq 0$ nas extremidades do elemento (figura 4-2).

No desenvolvimento que segue, a superposição das partições geométrica e física geram *partições de domínio* genéricas. Assim, é possível aplicar o mesmo tratamento para a discretização de elementos de contorno e células de domínio independentemente da dimensão de Ω . Uma partição de domínio é usualmente mapeada para um domínio normalizado de dimensão n_d . Desta forma, a localização de um ponto qualquer sobre a partição de

espaciais, mas apenas os nós são dotados de graus de liberdade.

domínio é determinada univocamente por suas coordenadas normalizadas $\boldsymbol{\xi} = (\xi_1, \xi_2, \xi_3)^2$.

4.1.1 Interpolação da geometria

As coordenadas \mathbf{x} de um ponto p pertencente a uma partição de domínio são interpoladas a partir das coordenadas dos pontos de sua partição geométrica. Sendo então $\bar{\phi}_q(\boldsymbol{\xi})$ a função interpoladora da partição geométrica correspondente ao j -ésimo ponto, escreve-se:

$$x_a(p) = \bar{\phi}_q(\boldsymbol{\xi}(p))x_a(\boldsymbol{\xi}^q) = \bar{\phi}_q x_a^q \quad . \quad (4.1)$$

Formalmente não é imposta nenhuma condição especial sobre as funções $\bar{\phi}$ além das condições obrigatórias para formas interpolatórias (Bathe 1996). A título de ilustração, sejam as funções $\bar{\phi}_q$ polinômios de Lagrange associados ao q -ésimo ponto de uma partição de domínio unidimensional formada por n_p pontos. Sua definição é dada por (Dhatt & Touzot 1984):

$$\bar{\phi}_q(\xi_1) = \prod_{\substack{i=1 \\ i \neq q}}^{n_p} \frac{(2i - n_p - 1) - (n_p - 1)\xi_1}{2(i - q)} \quad , \quad q > 1 \quad (4.2a)$$

$$\bar{\phi}_q(\xi_1) = 1 \quad , \quad q = 1 \quad (4.2b)$$

onde a relação entre as coordenadas normalizadas $\boldsymbol{\xi}$ e a coordenada real (curvilínea) \mathbf{y} medida sobre a partição de domínio é simplesmente:

$$\xi_g = \frac{2}{l_g} y_g \quad ,$$

e l_g é o comprimento total do elemento na direção de y_g . Exemplos destas partições estão ilustradas na figura 4-3.

Para partições bi ou tridimensionais o procedimento é idêntico, sendo os polinômios interpoladores obtidos por produto tensorial das funções usadas em (4.1). Por exemplo,

²Note-se que são necessárias apenas $n_d - 1$ variáveis normalizadas para localização do ponto em um elemento de contorno, mas são necessárias n_d variáveis normalizadas em células de domínio. Por exemplo, um elemento de contorno bidimensional ($n_d = 2$) utiliza $\boldsymbol{\xi} = (\xi_1)$ enquanto uma célula de domínio bidimensional utiliza $\boldsymbol{\xi} = (\xi_1, \xi_2)$. Já um elemento de contorno tridimensional ($n_d = 3$) utiliza $\boldsymbol{\xi} = (\xi_1, \xi_2)$ e assim por diante. O número de variáveis normalizadas necessárias fica claro no contexto em que são utilizadas (elementos de contorno ou células de domínio).

assumindo-se o mesmo número m de nós em cada direção do plano normalizado (figura 4-4), os mesmos graus de interpolação usados nas partições unidimensionais valem para as bidimensionais, sendo os polinômios obtidos por:

$$\zeta_{kl}(\xi_1, \xi_2) = \bar{\phi}_k(\xi_1) \times \bar{\phi}_l(\xi_2) \quad ,$$

e como $n_n = m^2$, é direto associar a cada função ζ_{kl} um ponto específico, a fim de manter a notação usada em (4.1):

$$\bar{\phi}_q(\boldsymbol{\xi}) = \zeta_{kl}(\xi_1, \xi_2) \quad , \quad (4.3)$$

onde os índices k e l determinam qual é o j -ésimo ponto da partição. Fica evidente que as células de domínio têm sua geometria interpolada por um polinômio de grau $(n_p - 1)^{n_d}$ enquanto as variáveis o são por um polinômio de grau $(n_n - 1)^{n_d}$.

Portanto, a interpolação da geometria de uma partição qualquer de domínio assume a seguinte forma matricial:

$$\mathbf{x} = \bar{\Phi} \mathbf{x}^e \quad (4.4)$$

onde \mathbf{x} têm dimensão n_d e

$$\mathbf{x}^e = \left\{ x_1^1 \quad \dots \quad x_{n_d}^1 \quad \vdots \quad x_1^2 \quad \dots \quad x_{n_d}^2 \quad \vdots \quad \dots \quad \vdots \quad x_1^{n_p} \quad \dots \quad x_{n_d}^{n_p} \right\}^T .$$

A fim de expressar as integrais em Γ_n e Ω_r em termos das coordenadas adimensionais $\boldsymbol{\xi}$ é necessário ainda o cálculo dos Jacobianos da transformação $\mathbf{x} \rightarrow \boldsymbol{\xi}$. Para $n_d = 2$ tem-se (Bathe 1996):

$$\begin{aligned} |J_n| &= \sqrt{\frac{dx_\alpha}{d\xi} \frac{dx_\alpha}{d\xi}} \\ |J_r| &= \frac{dx_1}{d\xi_1} \frac{dx_2}{d\xi_2} - \frac{dx_1}{d\xi_2} \frac{dx_2}{d\xi_1} \quad , \end{aligned}$$

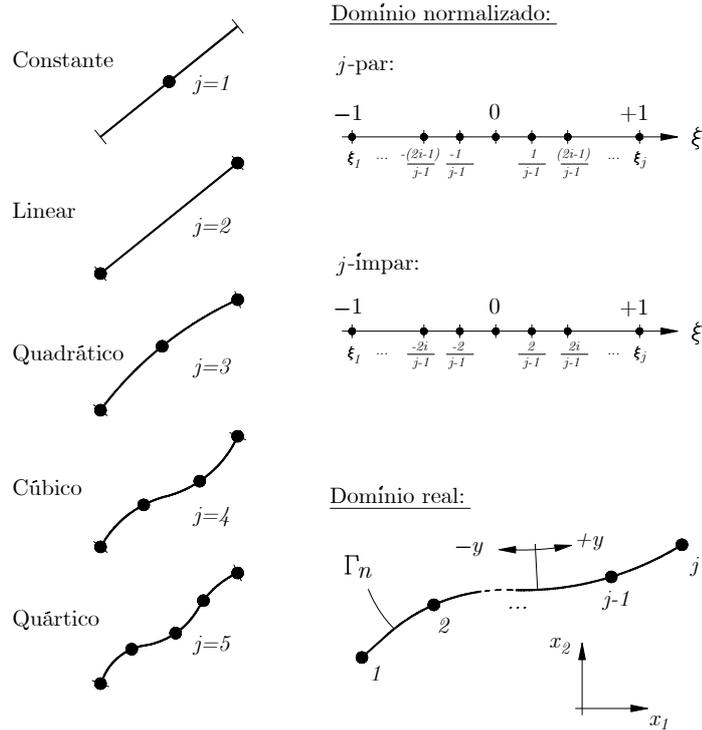


Figura 4-3: Configurações possíveis de partições geométricas lagrangeanas unidimensionais.

de forma que:

$$d\Gamma_n = |J_n| d\xi_1 \tag{4.6a}$$

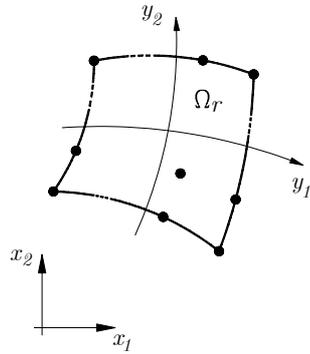
$$d\Omega_r = |J_r| d\xi_1 d\xi_2 \tag{4.6b}$$

4.1.2 Interpolação de variáveis físicas e carregamentos

Os polinômios gerados por (4.2) podem ser utilizados também pela partição física dos elementos, ou ainda adotar-se outra classe de funções para esta tarefa. Neste trabalho, os polinômios dados por (4.2) e seus descendentes obtidos por produto tensorial serão modificados para permitir o recuo dos nós extremos (elementos descontínuos) utilizados pela partição física.

Seja então um problema genérico associado a n_v graus de liberdade. Chamando os

Domínio real:



Domínio normalizado:

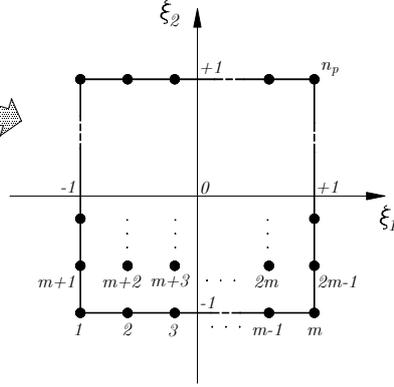


Figura 4-4: Partição geométrica lagrangeana bidimensional genérica.

graus de liberdade de v_i , estes serão interpolados na forma:

$$v_i(p) = \phi_r(\boldsymbol{\xi}(p))v_i(\boldsymbol{\xi}^r) = \phi_r v_i^r \quad . \quad (4.7)$$

Usualmente, as incógnitas e condições de contorno utilizadas no MEC direto são subconjuntos das variáveis duais (potenciais) e primais (fluxos) sobre Γ ou em Ω . Adicionalmente tem-se os carregamentos em Ω . Em algumas outras situações pode ser necessário manipular derivadas de uma ou mais destas variáveis. Neste trabalho, estas variáveis são agrupadas em conjuntos referentes ao problema de membrana, ao problema de flexão e ao cálculo de derivadas, conforme as equações (3.41), (3.54) e (3.78), respectivamente. De acordo com o descrito acima, estas variáveis são interpoladas na forma:

$$\mathbf{u} = \Phi \mathbf{u}^e \quad (4.8a)$$

$$\mathbf{t} = \Phi \mathbf{t}^e \quad (4.8b)$$

$$\mathbf{q} = \Phi \mathbf{q}^e \quad (4.8c)$$

$$\bar{\mathbf{u}} = \Phi \bar{\mathbf{u}}^e \quad (4.8d)$$

onde os índices m e f são usados em concordância com o problema. Nas eqs.(4.8), o índice

e indica qual o tipo de partição, isto é, utiliza-se $e = n$ para $\xi \in \Gamma_n$ e $e = r$ para $\xi \in \Omega_r$ ³.

Deve-se dar especial atenção às dimensões dos vetores das expressões (4.8). Com vistas a uma implementação numérica geral, será adotado um sistema de índices maiúsculos e minúsculos que simplifica a notação indicial (ver Simbologia). Para tanto, note-se que enquanto os vetores \mathbf{x} e $\bar{\mathbf{u}}$ têm dimensões múltiplas de n_d , os vetores \mathbf{u} , \mathbf{t} e \mathbf{q} possuem dimensão $n_v n_n$. Por outro lado, n_d e n_n são parâmetros geométricos e n_v é simplesmente a ordem dos tensores básicos (\mathbf{U} e \mathbf{T}) da solução fundamental do problema. Ou ainda, n_v é o número de incógnitas do problema de valores no contorno associado.

Conseqüentemente, no âmbito deste trabalho tem-se $n_d = 2$, $n_v = 2$ para o problema de membrana e $n_v = 3$ para o problema de flexão, como pode ser lembrado das variáveis de membrana:

$${}^m \mathbf{u} = \{u_1 \ u_2\}^T \quad (4.9a)$$

$${}^m \mathbf{t} = \{t_1 \ t_2\}^T \quad (4.9b)$$

$${}^m \mathbf{q} = \{q_1 \ q_2\}^T \quad , \quad (4.9c)$$

de flexão

$${}^f \mathbf{u} = \{\psi_1 \ \psi_2 \ u_3\}^T \quad (4.10a)$$

$${}^f \mathbf{t} = \{l_1 \ l_2 \ t_3\}^T \quad (4.10b)$$

$${}^f \mathbf{q} = \{m_1 \ m_2 \ q_3\}^T \quad , \quad (4.10c)$$

e das derivadas

$${}^m \bar{\mathbf{u}} = \{u_{1,1} \ u_{1,2} \ u_{2,1} \ u_{2,2}\}^T \quad (4.11a)$$

$${}^f \bar{\mathbf{u}} = \{u_{3,1} \ u_{3,2}\}^T \quad . \quad (4.11b)$$

As formas dos vetores de variáveis nodais de (4.8) é tal que agrega os graus de liberdade

³Nas equações discretizadas apresentadas neste Capítulo o sufixo e será omitido a fim de evitar a proliferação de símbolos, dado que seu significado torna-se óbvio no contexto em que os símbolos das eqs.(4.8) são utilizados.

de todos os nós da partição e . Explicitamente:

$${}^m \mathbf{u}^e = \left\{ u_1^1 \quad u_2^1 \quad \vdots \quad u_1^2 \quad u_2^2 \quad \vdots \quad \dots \quad \vdots \quad u_1^{n_n} \quad u_2^{n_n} \right\}^T \quad (4.12a)$$

$${}^f \mathbf{u}^e = \left\{ \psi_1^1 \quad \psi_2^1 \quad u_3^1 \quad \vdots \quad \psi_1^2 \quad \psi_2^2 \quad u_3^2 \quad \vdots \quad \dots \quad \vdots \quad \psi_1^{n_n} \quad \psi_2^{n_n} \quad u_3^{n_n} \right\}^T \quad (4.12b)$$

$${}^m \mathbf{t}^e = \left\{ t_1^1 \quad t_2^1 \quad \vdots \quad t_1^2 \quad t_2^2 \quad \vdots \quad \dots \quad \vdots \quad t_1^{n_n} \quad t_2^{n_n} \right\}^T \quad (4.12c)$$

$${}^f \mathbf{t}^e = \left\{ l_1^1 \quad l_2^1 \quad t_3^1 \quad \vdots \quad l_1^2 \quad l_2^2 \quad t_3^2 \quad \vdots \quad \dots \quad \vdots \quad l_1^{n_n} \quad l_2^{n_n} \quad t_3^{n_n} \right\}^T \quad (4.12d)$$

$${}^m \mathbf{q}^e = \left\{ q_1^1 \quad q_2^1 \quad \vdots \quad q_1^2 \quad q_2^2 \quad \vdots \quad \dots \quad \vdots \quad q_1^{n_n} \quad q_2^{n_n} \right\}^T \quad (4.12e)$$

$${}^f \mathbf{q}^e = \left\{ m_1^1 \quad m_2^1 \quad q_3^1 \quad \vdots \quad m_1^2 \quad m_2^2 \quad q_3^2 \quad \vdots \quad \dots \quad \vdots \quad m_1^{n_n} \quad m_2^{n_n} \quad q_3^{n_n} \right\}^T \quad (4.12f)$$

$$\bar{\mathbf{u}}_\beta^e = \left\{ u_{1,1}^1 \quad u_{1,2}^1 \quad u_{2,1}^1 \quad u_{2,2}^1 \quad \vdots \quad \dots \quad \vdots \quad u_{1,1}^{n_n} \quad u_{1,2}^{n_n} \quad u_{2,1}^{n_n} \quad u_{2,2}^{n_n} \right\}^T \quad (4.12g)$$

$$\bar{\mathbf{u}}_3^e = \left\{ u_{3,1}^1 \quad u_{3,2}^1 \quad \vdots \quad u_{3,1}^2 \quad u_{3,2}^2 \quad \vdots \quad \dots \quad \vdots \quad u_{3,1}^{n_n} \quad u_{3,2}^{n_n} \right\}^T \quad (4.12h)$$

Enquanto as matrizes $\bar{\Phi}$ e Φ contêm as funções de interpolação das partições geométrica e física do elemento (ou célula), respectivamente:

$$\bar{\Phi} = \begin{bmatrix} \bar{\varphi}^1 & \bar{\varphi}^2 & \dots & \bar{\varphi}^{n_p} \end{bmatrix} \quad (4.13a)$$

$$\Phi = \begin{bmatrix} \varphi^1 & \varphi^2 & \dots & \varphi^{n_n} \end{bmatrix} \quad , \quad (4.13b)$$

com (ver Simbologia para a convenção de índices)

$$\bar{\varphi}_{ab}^p = \bar{\phi}_p(\boldsymbol{\xi}) \delta_{ab} \quad , \quad \boldsymbol{\xi} \in \Gamma_n \quad (4.14a)$$

$$\varphi_{ij}^r = \phi_r(\boldsymbol{\xi}) \delta_{ij} \quad , \quad \boldsymbol{\xi} \in \Gamma_n \quad (4.14b)$$

4.2 Discretização das equações de membrana

A inserção de (4.12) e (4.13b) em (3.67) resulta na seguinte equação algébrica:

$$\begin{aligned}
{}^m C_{\alpha\beta}(P) \Phi_{\beta I}(P) {}^m u_I(P) + \sum_{n=1}^N \left\{ \int_{\Gamma_n} {}^m T_{\alpha\beta}(q, P) \Phi_{\beta I}(q) d\Gamma_n \right\} {}^m u_I(P) = \\
= \sum_{n=1}^N \left\{ \int_{\Gamma_n} {}^m U_{\alpha\beta}(q, P) \Phi_{\beta I}(q) d\Gamma_n \right\} {}^m t_I(P) + \\
+ \sum_{r=1}^R \left\{ \int_{\Omega_r} {}^m U_{\alpha\beta}(Q, P) \Phi_{\beta I}(Q) d\Omega_r \right\} {}^m q_I(Q) + \\
- \sum_{r=1}^R \left\{ \int_{\Omega_r} {}^m U_{\alpha\beta,\gamma}(Q, P) N_{\beta\gamma}^n(Q) d\Omega_r \right\} + {}^m v_\alpha(P) \quad . \quad (4.15a)
\end{aligned}$$

Ou, em notação matricial (o índice m está omitido, por clareza):

$$\begin{aligned}
\mathbf{C}(P) \Phi(P) \mathbf{u}(P) + \sum_{n=1}^N \left\{ \int_{\Gamma_n} \mathbf{T}(q, P) \Phi(q) d\Gamma_n \right\} \mathbf{u}(P) = \\
= \sum_{n=1}^N \left\{ \int_{\Gamma_n} \mathbf{U}(q, P) \Phi(q) d\Gamma_n \right\} \mathbf{t}(P) + \\
+ \sum_{r=1}^R \left\{ \int_{\Omega_r} \mathbf{U}(Q, P) \Phi(Q) d\Omega_r \right\} \mathbf{q}(Q) + \\
- \sum_{r=1}^R \left\{ \int_{\Omega_r} \widehat{\mathbf{U}}(Q, P) \mathbf{N}^n(Q) d\Omega_r \right\} + \mathbf{v}(P) \quad . \quad (4.16)
\end{aligned}$$

Neste caso, $n_v = 2$. Então, as equações (4.16) relacionam quatro variáveis (dois deslocamentos e dois esforços) para cada ponto nodal sobre Γ . Através do processo de colocação destas equações sobre cada um dos N_n nós do contorno, obtém-se um sistema linear a $2N_n$ equações e $2N_n$ incógnitas (já que duas das quatro variáveis são prescritas como condições de contorno, em cada nó). Isto porque a implementação numérica não fará uso da técnica do nó duplo (Brebbia & Dominguez 1989), pois serão utilizados elementos descontínuos. A possibilidade de uso da referida técnica não está excluída, entretanto, e nestes casos o sistema seria aumentado em $2n_c$ incógnitas e $2n_c$ equações, onde n_c é o número de vértices.

A imposição de condições de contorno para geometrias irregulares é facilitada se as variáveis nodais \mathbf{u}^n e \mathbf{t}^n são referenciadas em relação ao sistema local de coordenadas $\mathbf{n}, \mathbf{t}, \mathbf{s}$ do elemento Γ_n ao qual o ponto pertence. Quando este é o caso, os vetores \mathbf{u}^n e \mathbf{t}^n

seriam redefinidos como segue:

$${}^m \mathbf{u} = \left\{ u^1 \quad v^1 \quad \vdots \quad u^2 \quad v^2 \quad \vdots \quad \dots \quad \vdots \quad u^{n_n} \quad v^{n_n} \right\}^T$$

$${}^m \mathbf{t} = \left\{ N_{nn}^1 \quad N_{nt}^1 \quad \vdots \quad N_{nn}^2 \quad N_{nt}^2 \quad \vdots \quad \dots \quad \vdots \quad N_{nn}^{n_n} \quad N_{nt}^{n_n} \right\}^T$$
(4.17)

Seja agora π_p o ângulo entre o eixo \mathbf{n} e o eixo \mathbf{x}_1 , sobre o nó p , como ilustrado na figura 4-5. A matriz de muncaça de base do sistema $\mathbf{x}_1 \times \mathbf{x}_2$ para $\mathbf{n} \times \mathbf{t}$ é:

$$P_{ij}(p) = \begin{bmatrix} \cos \pi_p & \sin \pi_p \\ -\sin \pi_p & \cos \pi_p \end{bmatrix},$$
(4.18)

e portanto, as matrizes de transformação de todos os nós de Γ_n podem ser agrupadas na forma:

$$R_{IJ}^n = \begin{bmatrix} \mathbf{P}(p_1) & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{P}(p_2) & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{P}(p_{n_n}) \end{bmatrix}.$$
(4.19)

Agora as eqs.(4.15a) podem ser reescritas considerando um processo de colocação sobre

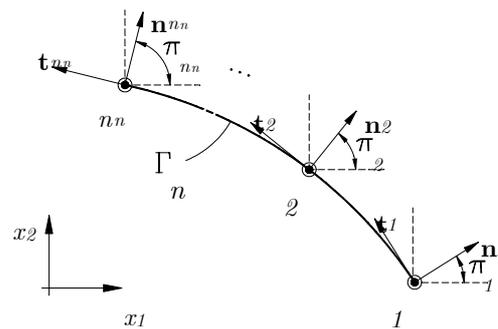


Figura 4-5: Definições geométricas dos sistemas de coordenadas nodais de um elemento de contorno bidimensional.

pontos $p \in \Gamma$:

$$\begin{aligned}
& C_{\alpha\beta}(p) \Phi_{\beta J}(p) R_{JI}^n(p) u_I(p) + \\
& + \sum_{n=1}^N \left\{ \int_{\Gamma_n} T_{\alpha\beta}(q, p) \Phi_{\beta J}(q) d\Gamma_n \right\} R_{JI}^n(q) u_I(q) = \\
& = \sum_{n=1}^N \left\{ \int_{\Gamma_n} U_{\alpha\beta}(q, p) \Phi_{\beta J}(q) d\Gamma_n \right\} R_{JI}^n(q) t_I(q) + \\
& + \sum_{r=1}^R \left\{ \int_{\Omega_r} U_{\alpha\beta}(Q, p) \Phi_{\beta I}(Q) q_I(Q) d\Omega_r \right\} + \\
& - \sum_{r=1}^R \left\{ \int_{\Omega_r} U_{\alpha\beta, \gamma}(Q, p) N_{\beta\gamma}^n(Q) d\Omega_r \right\} + {}^m v_\alpha(p) \quad , \quad (4.20)
\end{aligned}$$

ou, matricialmente:

$$\begin{aligned}
& \mathbf{C}(p) \Phi(p) \mathbf{R}^n(p) \mathbf{u}(p) + \sum_{n=1}^N \left\{ \int_{\Gamma_n} \mathbf{T}(q, p) \Phi(q) d\Gamma_n \right\} \mathbf{R}^n(q) \mathbf{u}(q) = \\
& = \sum_{n=1}^N \left\{ \int_{\Gamma_n} \mathbf{U}(q, p) \Phi(q) d\Gamma_n \right\} \mathbf{R}^n(q) \mathbf{t}(q) + \\
& + \sum_{r=1}^R \left\{ \int_{\Omega_r} \mathbf{U}(Q, p) \Phi(Q) \mathbf{q}(Q) d\Omega_r \right\} + \\
& - \sum_{r=1}^R \left\{ \int_{\Omega_r} \hat{\mathbf{U}}(Q, p) \mathbf{N}^n(Q) d\Omega_r \right\} + \mathbf{v}(p) \quad , \quad (4.21)
\end{aligned}$$

onde \mathbf{q}^r , \mathbf{N}^n e \mathbf{v} são definidos no sistema global de coordenadas, como em (4.12e). A eq.(4.20) é válida para \mathbf{u}^n e \mathbf{t}^n definidos no sistema $\mathbf{x}_1 \times \mathbf{x}_2$ ou $\mathbf{n} \times \mathbf{t}$. No primeiro caso, basta alterar (4.18) para $\mathbf{P} = \mathbf{I}$ (isto é, $\pi_p = 0$). Note-se ainda que nesta forma as variáveis são calculadas segundo o sistema de coordenadas $\mathbf{x}_1 \times \mathbf{x}_2$. Após a solução do problema, os resultados sobre o contorno podem ser convertidos para o sistema local $\mathbf{n} \times \mathbf{t}$ utilizando-se (4.18).

Para facilitar a implementação das quadraturas numéricas, as integrais em Γ_n e Ω_r são mapeadas para os domínios normalizados através da substituição dos diferenciais dados pelas equações (4.6). Utilizando convenções similares às de Brebbia et al. (1984) define-se

as submatrizes:

$$\bar{\mathbf{H}}_n = \left\{ \int_{-1}^{+1} \mathbf{T}(q, p) \Phi(q) |J_n| d\xi_1 \right\} \mathbf{R}^n(q) \quad , \quad (4.22a)$$

$$\mathbf{G}_n = \left\{ \int_{-1}^{+1} \mathbf{U}(q, p) \Phi(q) |J_n| d\xi_1 \right\} \mathbf{R}^n(q) \quad , \quad (4.22b)$$

$$\mathbf{F}_r = \left\{ \int_{-1}^{+1} \int_{-1}^{+1} \mathbf{U}(Q, p) \Phi(Q) |J_r| \mathbf{q}(Q) d\xi_1 d\xi_2 \right\} \quad , \quad (4.22c)$$

$$\mathbf{b}_r = - \left\{ \int_{-1}^{+1} \int_{-1}^{+1} \hat{\mathbf{U}}(Q, p) \mathbf{N}^n(Q) |J_r| d\xi_1 d\xi_2 \right\} \quad . \quad (4.22d)$$

Considerando que o processo de colocação é nodal verifica-se que, para um ponto de colocação p coincidente com o j -ésimo nó do elemento que está sendo integrado, todas as submatrizes φ de (4.13b) se anulam, com exceção da submatriz φ^j , que se reduz à identidade. Portanto, o primeiro termo à esquerda de (4.21) sofre na realidade uma contração que resulta nas submatrizes que são adicionadas à diagonal principal de $\bar{\mathbf{H}}_n$ (Brebbia et al. 1984). Assim, (4.21) é reescrita:

$$\mathbf{CIR}^n \mathbf{u}(p) + \sum_{n=1}^N \bar{\mathbf{H}}_n \mathbf{u}(q) = \sum_{n=1}^N \mathbf{G}_n \mathbf{t}(q) + \sum_{r=1}^R \mathbf{F}_r + \sum_{r=1}^R \mathbf{b}_r + \mathbf{v}(p) \quad . \quad (4.23)$$

Convencionando ainda (Brebbia et al. 1984):

$$\mathbf{H}_n = \left\{ \begin{array}{ll} \bar{\mathbf{H}}_n & \text{para } p \neq q \\ \bar{\mathbf{H}}_n + \mathbf{C}\mathbf{R}^n & \text{para } p \equiv q \end{array} \right\} \quad (4.24)$$

e sobrepondo as matrizes obtidas após o processo de colocação, resulta o seguinte sistema de equações:

$$\mathbf{H}\mathbf{u} = \mathbf{G}\mathbf{t} + \mathbf{f} \quad , \quad (4.25)$$

onde \mathbf{f} é obtido pela soma das contribuições de \mathbf{F}_r , \mathbf{b}_r e \mathbf{v} :

$$\mathbf{f} = \sum_{r=1}^R \mathbf{F}_r + \sum_{r=1}^R \mathbf{b}_r + \mathbf{v}(p) \quad . \quad (4.26)$$

Uma vez obtidas as incógnitas remanescentes sobre o contorno, através da solução de

(4.25), obtém-se os deslocamentos em qualquer ponto interno $P \in \Omega$ particularizando (4.21) com $\mathbf{C}(P) = \mathbf{I}$, o que resulta:

$$\begin{aligned} \mathbf{u}(P) = & - \sum_{n=1}^N \left\{ \int_{\Gamma_n} \mathbf{T}(q, P) \Phi(q) \mathbf{R}^n(q) \mathbf{u}(q) d\Gamma_n \right\} + \\ & + \sum_{n=1}^N \left\{ \int_{\Gamma_n} \mathbf{U}(q, P) \Phi(q) \mathbf{R}^n(q) \mathbf{t}(q) d\Gamma_n \right\} + \\ & + \sum_{r=1}^R \left\{ \int_{\Omega_r} \mathbf{U}(Q, P) \Phi(q) \mathbf{q}(Q) d\Omega_r \right\} + \\ & - \sum_{r=1}^R \left\{ \int_{\Omega_r} \widehat{\mathbf{U}}(Q, P) \mathbf{N}^n(Q) d\Omega_r \right\} + \mathbf{v}(P) \quad . \quad (4.27) \end{aligned}$$

O cálculo de esforços em pontos internos é fartamente documentado (Banerjee & Butterfield 1981, Brebbia et al. 1984, Brebbia & Dominguez 1989) e não será repetido aqui.

4.3 Discretização das equações de flexão

Aqui serão repetidos os passos da seção anterior para as equações (3.70), utilizando as mesmas formas interpolatórias dadas por (4.12) e (4.13b), o que resulta a equação algébrica:

$$\begin{aligned} {}^f C_{ij}(P) \Phi_{jI}(P) {}^f u_I(P) + \sum_{n=1}^N \left\{ \int_{\Gamma_n} {}^f T_{ij}(q, P) \Phi_{jI}(q) d\Gamma_n \right\} {}^f u_I(q) = \\ = \sum_{n=1}^N \left\{ \int_{\Gamma_n} {}^f U_{ij}(q, P) \Phi_{jI}(q) d\Gamma_n \right\} {}^f t_I(q) + \\ + \sum_{r=1}^R \left\{ \int_{\Omega_r} {}^f V_{ij}(Q, P) \Phi_{jI}(Q) {}^f q_I(Q) d\Omega_r \right\} + \\ - \sum_{r=1}^R \left\{ \int_{\Omega_r} {}^f U_{i3,\beta}(Q, P) N_{\alpha\beta}(Q) \Phi_{\alpha M}(Q) d\Omega_r \right\} \bar{u}_{3M}(Q) + {}^f v_i(P) \quad . \quad (4.28) \end{aligned}$$

Ou, em notação matricial (o índice f está omitido):

$$\begin{aligned}
\mathbf{C}(P) \Phi(P) \mathbf{u}(P) + \sum_{n=1}^N \left\{ \int_{\Gamma_n} \mathbf{T}(q, P) \Phi(q) d\Gamma_n \right\} \mathbf{u}(q) = \\
= \sum_{n=1}^N \left\{ \int_{\Gamma_n} \mathbf{U}(q, P) \Phi(q) d\Gamma_n \right\} \mathbf{t}(q) + \\
+ \sum_{r=1}^R \left\{ \int_{\Omega_r} \mathbf{V}(Q, P) \Phi(Q) \mathbf{q}(Q) d\Omega_r \right\} + \\
- \sum_{r=1}^R \left\{ \int_{\Omega_r} \widehat{\mathbf{U}}(Q, P) \mathbf{N}(Q) \Phi(Q) d\Omega_r \right\} \bar{\mathbf{u}}_3(Q) + \mathbf{v}(P). \quad (4.29)
\end{aligned}$$

Para este caso, $n_v = 3$, de forma que as equações (4.29) relacionam seis variáveis (três deslocamentos e três esforços) para cada ponto nodal sobre Γ . O processo de colocação destas equações sobre cada um dos N_n nós do contorno gera um sistema linear a $3N_n$ equações e $3N_n$ incógnitas. Os comentários feitos na seção anterior para pontos com descontinuidade da normal permanecem válidos.

Para imposição de condições de contorno, as variáveis nodais \mathbf{u}^n e \mathbf{t}^n também poderiam ser referenciadas em relação ao sistema local de coordenadas $\mathbf{n}, \mathbf{t}, \mathbf{s}$ do elemento Γ_n ao qual o ponto pertence. Ou seja:

$$\begin{aligned}
f \mathbf{u} &= \left\{ \psi_{nn}^1 \quad \psi_{tt}^1 \quad w^1 \quad \vdots \quad \psi_{nn}^2 \quad \psi_{tt}^2 \quad w^2 \quad \vdots \quad \dots \quad \vdots \quad \psi_{nn}^{n_n} \quad \psi_{tt}^{n_n} \quad w^{n_n} \right\}^T \\
f \mathbf{t} &= \left\{ M_{nn}^1 \quad M_{nt}^1 \quad Q_n^1 \quad \vdots \quad M_{nn}^2 \quad M_{nt}^2 \quad Q_n^2 \quad \vdots \quad \dots \quad \vdots \quad M_{nn}^{n_n} \quad M_{nt}^{n_n} \quad Q_n^{n_n} \right\}^T
\end{aligned} \quad (4.30)$$

De forma similar à definição dada por (4.18), a matriz de mudança de base do sistema $\mathbf{x}_1 \times \mathbf{x}_2$ para $\mathbf{n} \times \mathbf{t}$ neste caso é:

$$P_{ij}(p) = \begin{bmatrix} \cos \pi_p & \sin \pi_p & 0 \\ -\sin \pi_p & \cos \pi_p & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (4.31)$$

com a forma geral de (4.19) permanecendo válida. Agora as eqs.(4.28) podem ser reescritas

para um processo de colocação sobre pontos $p \in \Gamma$:

$$\begin{aligned}
& C_{ik}(p) \Phi_{kJ}(p) R_{JI}^n(p) u_I(p) + \\
& + \sum_{n=1}^N \left\{ \int_{\Gamma_n} T_{ik}(q, p) \Phi_{kJ}(q) d\Gamma_n \right\} R_{JI}^n(q) u_I(q) = \\
& = \sum_{n=1}^N \left\{ \int_{\Gamma_n} U_{ik}(q, p) \Phi_{kJ}(q) d\Gamma_n \right\} R_{JI}^n(q) t_I(q) + \\
& + \sum_{r=1}^R \left\{ \int_{\Omega_r} V_{ik}(Q, p) \Phi_{kI}(Q) q_I(Q) d\Omega_r \right\} + \\
& - \sum_{r=1}^R \left\{ \int_{\Omega_r} U_{i3,\alpha}(Q, p) N_{\alpha\beta}(Q) \Phi_{\beta M}(Q) d\Omega_r \right\} \bar{u}_{3M}(Q) + v_i(p) \quad , \quad (4.32)
\end{aligned}$$

ou, matricialmente:

$$\begin{aligned}
& \mathbf{C}(p) \Phi(p) \mathbf{R}^n(p) \mathbf{u}^n(p) + \sum_{n=1}^N \left\{ \int_{\Gamma_n} \mathbf{T}(q, p) \Phi(q) d\Gamma_n \right\} \mathbf{R}^n(q) \mathbf{u}(q) = \\
& = \sum_{n=1}^N \left\{ \int_{\Gamma_n} \mathbf{U}(q, p) \Phi(q) d\Gamma_n \right\} \mathbf{R}^n(q) \mathbf{t}(q) + \\
& + \sum_{r=1}^R \left\{ \int_{\Omega_r} \mathbf{V}(Q, p) \Phi(Q) \mathbf{q}(Q) d\Omega_r \right\} + \\
& - \sum_{r=1}^R \left\{ \int_{\Omega_r} \hat{\mathbf{U}}(Q, p) \mathbf{N}(Q) \Phi(Q) d\Omega_r \right\} \bar{\mathbf{u}}_3(Q) + \mathbf{v}(p) \quad , \quad (4.33)
\end{aligned}$$

onde \mathbf{q}^r , \mathbf{N} e \mathbf{v} são definidos no sistema global de coordenadas, como em (4.12f). De forma idêntica ao problema de membrana, define-se:

$$\bar{\mathbf{H}}_n = \left\{ \int_{-1}^{+1} \mathbf{T}(q, p) \Phi(q) |J_n| d\xi_1 \right\} \mathbf{R}^n(q) \quad , \quad (4.34a)$$

$$\mathbf{G}_n = \left\{ \int_{-1}^{+1} \mathbf{U}(q, p) \Phi(q) |J_n| d\xi_1 \right\} \mathbf{R}^n(q) \quad , \quad (4.34b)$$

$$\mathbf{F}_r = \left\{ \int_{-1}^{+1} \int_{-1}^{+1} \mathbf{V}(Q, p) \Phi(Q) |J_r| \mathbf{q}(Q) d\xi_1 d\xi_2 \right\} \quad , \quad (4.34c)$$

$$\mathbf{B}_r = - \left\{ \int_{-1}^{+1} \int_{-1}^{+1} \hat{\mathbf{U}}(Q, p) \mathbf{N}(Q) |J_r| d\xi_1 d\xi_2 \right\} \quad , \quad (4.34d)$$

e (4.33) é reescrita:

$$\mathbf{C}\mathbf{I}\mathbf{R}^n\mathbf{u}(p) + \sum_{n=1}^N \bar{\mathbf{H}}_n\mathbf{u}(q) = \sum_{n=1}^N \mathbf{G}_n\mathbf{t}(q) + \sum_{r=1}^R \mathbf{F}_r + \sum_{r=1}^R \mathbf{B}_r\bar{\mathbf{u}}_3(Q) + \mathbf{v}(p) \quad . \quad (4.35)$$

Convencionando que (Brebbia et al. 1984):

$$\mathbf{H}_n = \begin{cases} \bar{\mathbf{H}}_n & \text{para } p \neq q \\ \bar{\mathbf{H}}_n + \mathbf{C}\mathbf{R}^n & \text{para } p \equiv q \end{cases} \quad , \quad (4.36)$$

e sobrepondo as matrizes obtidas após o processo de colocação, resulta o seguinte sistema de equações:

$$\mathbf{H}\mathbf{u} = \mathbf{G}\mathbf{t} + \mathbf{B}\bar{\mathbf{u}} + \mathbf{f} \quad . \quad (4.37)$$

onde \mathbf{f} é obtido pela soma das contribuições de \mathbf{F}_r e \mathbf{w} , isto é:

$$\mathbf{f} = \sum_{r=1}^R \mathbf{F}_r + \mathbf{v}(p) \quad (4.38)$$

Após a solução do problema de valores no contorno representado por (4.37), obtém-se os deslocamentos em pontos internos $P \in \Omega$ particularizando (4.33) com $\mathbf{C}(P) = \mathbf{I}$:

$$\begin{aligned} \mathbf{u}(P) = & - \sum_{n=1}^N \left\{ \int_{\Gamma_n} \mathbf{T}(q, P) \Phi(q) \mathbf{R}^n(q) \mathbf{u}(q) d\Gamma_n \right\} + \\ & + \sum_{n=1}^N \left\{ \int_{\Gamma_n} \mathbf{U}(q, P) \Phi(q) \mathbf{R}^n(q) \mathbf{t}(q) d\Gamma_n \right\} + \\ & + \sum_{r=1}^R \left\{ \int_{\Omega_r} \mathbf{V}(Q, P) \Phi(Q) \mathbf{q}(Q) d\Omega_r \right\} + \\ & - \sum_{r=1}^R \left\{ \int_{\Omega_r} \hat{\mathbf{U}}(Q, P) \mathbf{N}(Q) d\Omega_r \right\} \bar{\mathbf{u}}_3(Q) + \mathbf{v}(P) \quad . \quad (4.39) \end{aligned}$$

O cálculo de esforços em pontos internos pode ser encontrado na literatura (van der Weeën 1982a, Westphal Jr. & de Barcellos 1990).

4.4 Discretização das equações para cálculo dos gradientes dos deslocamentos translacionais

Novamente será empregado o procedimento básico utilizado nas equações de membrana e flexão, tomando-se como ponto de partida as eqs.(3.76) e (3.77) que, conforme discutido no Capítulo 3, são equações integrais hipersingulares. Com o objetivo de evitar a integração de núcleos singulares, será assumido de antemão o cálculo das derivadas $u_{i,\alpha}$ em pontos exclusivamente internos. Esta abordagem justifica a utilização de células de domínio completamente descontínuas, isto é, nenhum nó da partição física de quaisquer células está posicionado sobre o contorno. Por outro lado, torna evidente a necessidade de cuidados com integrais quase-singulares.

A forma discretizada da eq.(3.76) é escrita:

$$\begin{aligned}
 \bar{u}_\beta(P) - \sum_{n=1}^N \left\{ \int_{\Gamma_n} {}^m T_{\beta\gamma,\alpha}(q, P) \Phi_{\gamma I}(q) d\Gamma_n \right\} {}^m u_I(q) = \\
 = - \sum_{n=1}^N \left\{ \int_{\Gamma_n} {}^m U_{\beta\gamma,\alpha}(q, P) \Phi_{\gamma I}(q) d\Gamma_n \right\} {}^m t_I(q) + \\
 + \sum_{r=1}^R \left\{ \int_{\Omega_r} {}^m V_{\beta\gamma,\alpha}(Q, P) \Phi_{\gamma I}(Q) {}^m q_I(Q) d\Omega_r \right\} + \\
 + \sum_{r=1}^R \left\{ \int_{\Omega_r} {}^m U_{\beta\gamma,\delta\alpha}(Q, p) N_{\gamma\delta}^n(Q) d\Omega_r \right\} + {}^m c_\beta^N(P) + {}^m v_{3,\alpha}(P) \quad . \quad (4.40)
 \end{aligned}$$

Ou, em notação matricial:

$$\begin{aligned}
 \bar{\mathbf{u}}_\beta(P) - \sum_{n=1}^N \left\{ \int_{\Gamma_n} \bar{\mathbf{T}}(q, P) \Phi(q) d\Gamma_n \right\} \mathbf{u}(q) = \\
 = - \sum_{n=1}^N \left\{ \int_{\Gamma_n} \bar{\mathbf{U}}(q, P) \Phi(q) d\Gamma_n \right\} \mathbf{t}(q) + \\
 - \sum_{r=1}^R \left\{ \int_{\Omega_r} \bar{\mathbf{V}}(Q, P) \Phi(Q) \mathbf{q}(Q) d\Omega_r \right\} + \\
 + \sum_{r=1}^R \left\{ \int_{\Omega_r} \bar{\bar{\mathbf{U}}}(Q, P) \mathbf{N}^n(Q) d\Omega_r \right\} + \mathbf{c}_\beta^N(P) + \bar{\mathbf{v}}(P) \quad . \quad (4.41)
 \end{aligned}$$

Para este caso as equações (4.41) relacionam duas variáveis, quais sejam, as derivadas cartesianas do deslocamento longitudinal u_α da placa. Como estas variáveis são definidas em pontos internos, adota-se o sistema de coordenadas global como referência. Mas como as variáveis sobre o contorno podem estar definidas em relação ao sistema de coordenadas local de cada nó, torna-se necessário o emprego da matriz (4.19):

$$\begin{aligned}
\bar{u}_\beta(P) - \sum_{n=1}^N \left\{ \int_{\Gamma_n} {}^m T_{\beta\gamma,\alpha}(q, P) \Phi_{\gamma J}(q) d\Gamma_n \right\} R_{JI}^n(q) {}^m u_I(q) = \\
= - \sum_{n=1}^N \left\{ \int_{\Gamma_n} {}^m U_{\beta\gamma,\alpha}(q, P) \Phi_{\gamma J}(q) d\Gamma_n \right\} R_{JI}^n(q) {}^m t_I(q) + \\
+ \sum_{r=1}^R \left\{ \int_{\Omega_r} {}^m V_{\beta\gamma,\alpha}(Q, P) \Phi_{\gamma I}(Q) {}^m q_I(Q) d\Omega_r \right\} + \\
+ \sum_{r=1}^R \left\{ \int_{\Omega_r} {}^m U_{\beta\gamma,\delta\alpha}(Q, p) N_{\gamma\delta}^n(Q) d\Omega_r \right\} + {}^m c_\beta^N(P) + {}^m v_{3,\alpha}(P) \quad , \quad (4.42)
\end{aligned}$$

ou

$$\begin{aligned}
\bar{\mathbf{u}}_\beta(P) - \sum_{n=1}^N \left\{ \int_{\Gamma_n} \bar{\mathbf{T}}(q, P) \Phi(q) d\Gamma_n \right\} \mathbf{R}^n(q) \mathbf{u}(q) = \\
= - \sum_{n=1}^N \left\{ \int_{\Gamma_n} \bar{\mathbf{U}}(q, P) \Phi(q) d\Gamma_n \right\} \mathbf{R}^n(q) \mathbf{t}(q) + \\
- \sum_{r=1}^R \left\{ \int_{\Omega_r} \bar{\mathbf{V}}(Q, P) \Phi(Q) \mathbf{q}(Q) d\Omega_r \right\} + \\
+ \sum_{r=1}^R \left\{ \int_{\Omega_r} \bar{\mathbf{U}}(Q, P) \mathbf{N}^n(Q) d\Omega_r \right\} + \mathbf{c}_\beta^N + \bar{\mathbf{v}}(P) \quad . \quad (4.43)
\end{aligned}$$

De forma similar às seções anteriores, parametriza-se (4.43) com as definições

$$\mathbf{H}_n = - \left\{ \int_{-1}^{+1} \bar{\mathbf{T}}(q, P) \Phi(q) |J_n| d\xi_1 \right\} \mathbf{R}^n(q) \quad , \quad (4.44a)$$

$$\mathbf{G}_n = - \left\{ \int_{-1}^{+1} \bar{\mathbf{U}}(q, P) \Phi(q) |J_n| d\xi_1 \right\} \mathbf{R}^n(q) \quad , \quad (4.44b)$$

$$\mathbf{F}_r = - \left\{ \int_{-1}^{+1} \int_{-1}^{+1} \bar{\mathbf{V}}(Q, P) \Phi(Q) |J_r| \mathbf{q}(Q) d\xi_1 d\xi_2 \right\} \quad , \quad (4.44c)$$

$$\bar{\mathbf{B}}_r = \left\{ \int_{-1}^{+1} \int_{-1}^{+1} \bar{\mathbf{U}}(Q, P) \mathbf{N}^n(Q) |J_r| d\xi_1 d\xi_2 \right\} \quad . \quad (4.44d)$$

Então (4.43) é reescrita:

$$\bar{\mathbf{u}}_\beta(P) + \sum_{n=1}^N \mathbf{H}_n \mathbf{u}(q) = \sum_{n=1}^N \mathbf{G}_n \mathbf{t}(q) + \sum_{r=1}^R \mathbf{F}_r + \sum_{r=1}^R \bar{\mathbf{B}}_r(Q) + \mathbf{c}_\beta^N(P) + \bar{\mathbf{v}}(P) \quad , \quad (4.45)$$

lembrando que os termos convectivos \mathbf{c}^N só são considerados quando $P \equiv Q$. Adicionalmente, seja

$$\mathbf{B}_r = \left\{ \begin{array}{ll} \bar{\mathbf{B}}_r & \text{para } p \neq q \\ \bar{\mathbf{B}}_r + \mathbf{c}_\beta^N & \text{para } p \equiv q \quad , \end{array} \right\} \quad (4.46)$$

e sobrepondo as matrizes obtidas após o processo de colocação, resulta o seguinte sistema de equações:

$$\bar{\mathbf{u}}_\beta + \mathbf{H}\mathbf{u} = \mathbf{G}\mathbf{t} + \mathbf{B} + \mathbf{f} \quad . \quad (4.47)$$

onde \mathbf{f} é obtido pela soma das contribuições de \mathbf{F}_r e $\bar{\mathbf{v}}$, ou seja:

$$\mathbf{f} = \sum_{r=1}^R \mathbf{F}_r + \bar{\mathbf{v}}(P) \quad (4.48)$$

Seguindo um procedimento idêntico para a eq.(3.77) tem-se:

$$\begin{aligned}
\bar{u}_3(P) - \sum_{n=1}^N \left\{ \int_{\Gamma_n} {}^f T_{3i,\alpha}(q, P) \Phi_{iJ}(q) d\Gamma_n \right\} {}^f u_J(q) = \\
= - \sum_{n=1}^N \left\{ \int_{\Gamma_n} {}^f U_{3i,\alpha}(q, P) \Phi_{iJ}(q) d\Gamma_n \right\} {}^f t_J(q) + \\
+ \sum_{r=1}^R \left\{ \int_{\Omega_r} {}^f V_{3i,\alpha}(Q, P) \Phi_{iJ}(Q) {}^f q_J(Q) d\Omega_r \right\} + \\
+ \sum_{r=1}^R \left\{ \int_{\Omega_r} {}^f U_{33,\alpha\beta}(Q, P) N_{\beta\gamma}(Q) \Phi_{\gamma M}(Q) d\Omega_r \right\} \bar{u}_{3M}(Q) + \\
+ c_{\alpha\beta}^N(P) \Phi_{\beta M}(P) \bar{u}_{3M}(P) + {}^f v_{3,\alpha}(P) \quad . \quad (4.49)
\end{aligned}$$

Ou, em notação matricial:

$$\begin{aligned}
\bar{\mathbf{u}}_3(P) - \sum_{n=1}^N \left\{ \int_{\Gamma_n} \bar{\mathbf{T}}(q, P) \Phi(q) d\Gamma_n \right\} \mathbf{u}(q) = \\
= - \sum_{n=1}^N \left\{ \int_{\Gamma_n} \bar{\mathbf{U}}(q, P) \Phi(q) d\Gamma_n \right\} \mathbf{t}(q) + \\
- \sum_{r=1}^R \left\{ \int_{\Omega_r} \bar{\mathbf{V}}(Q, P) \Phi(Q) d\Omega_r \right\} \mathbf{q}(Q) + \\
+ \sum_{r=1}^R \left\{ \int_{\Omega_r} \bar{\bar{\mathbf{U}}}(Q, P) \mathbf{N}(Q) \Phi(Q) d\Omega_r \right\} \bar{\mathbf{u}}_3(Q) + \\
+ \mathbf{c}^N(P) \bar{\mathbf{u}}_3(P) + \bar{\mathbf{v}}(P) \quad . \quad (4.50)
\end{aligned}$$

Novamente, levando-se em conta que as variáveis sobre o contorno podem estar definidas

em relação ao sistema de coordenadas local de cada nó, emprega-se a matriz (4.19):

$$\begin{aligned}
\bar{u}_3(P) - \sum_{n=1}^N \left\{ \int_{\Gamma_n} {}^f T_{3i,\alpha}(q, P) \Phi_{iJ}(q) d\Gamma_n \right\} R_{JI}^n(q) {}^f u_I(q) = \\
= - \sum_{n=1}^N \left\{ \int_{\Gamma_n} {}^f U_{3i,\alpha}(q, P) \Phi_{iJ}(q) d\Gamma_n \right\} R_{JI}^n(q) {}^f t_I(q) + \\
+ \sum_{r=1}^R \left\{ \int_{\Omega_r} {}^f V_{3i,\alpha}(Q, P) \Phi_{iJ}(Q) {}^f q_J(Q) d\Omega_r \right\} + \\
+ \sum_{r=1}^R \left\{ \int_{\Omega_r} {}^f U_{33,\alpha\beta}(Q, P) N_{\beta\gamma}(Q) \Phi_{\gamma M}(Q) d\Omega_r \right\} {}^f \bar{u}_{3M}(Q) + \\
+ c_{\alpha\beta}^N(P) \Phi_{\beta M}(P) \bar{u}_{3M}(P) + {}^f v_{3,\alpha}(P) \quad . \quad (4.51)
\end{aligned}$$

ou, matricialmente:

$$\begin{aligned}
\bar{\mathbf{u}}_3(P) - \sum_{n=1}^N \left\{ \int_{\Gamma_n} \bar{\mathbf{T}}(q, P) \Phi(q) d\Gamma_n \right\} \mathbf{R}^n(q) \mathbf{u}(q) = \\
= - \sum_{n=1}^N \left\{ \int_{\Gamma_n} \bar{\mathbf{U}}(q, P) \Phi(q) d\Gamma_n \right\} \mathbf{R}^n(q) \mathbf{t}(q) + \\
- \sum_{r=1}^R \left\{ \int_{\Omega_r} \bar{\mathbf{V}}(Q, P) \Phi(Q) \mathbf{q}(Q) d\Omega_r \right\} + \\
+ \sum_{r=1}^R \left\{ \int_{\Omega_r} \bar{\bar{\mathbf{U}}}(Q, P) \mathbf{N}(Q) \Phi(Q) d\Omega_r \right\} \bar{\mathbf{u}}_3(Q) + \\
+ \mathbf{c}^N(P) \bar{\mathbf{u}}_3(P) + \bar{\mathbf{v}}(P) \quad . \quad (4.52)
\end{aligned}$$

Parametrizando (4.52) com as definições

$$\mathbf{H}_n = - \left\{ \int_{-1}^{+1} \bar{\mathbf{T}}(q, P) \Phi(q) |J_n| d\xi_1 \right\} \mathbf{R}^n(q) \quad , \quad (4.53a)$$

$$\mathbf{G}_n = - \left\{ \int_{-1}^{+1} \bar{\mathbf{U}}(q, P) \Phi(q) |J_n| d\xi_1 \right\} \mathbf{R}^n(q) \quad , \quad (4.53b)$$

$$\mathbf{F}_r = - \left\{ \int_{-1}^{+1} \int_{-1}^{+1} \bar{\mathbf{V}}(Q, P) \Phi(Q) |J_r| \mathbf{q}(Q) d\xi_1 d\xi_2 \right\} \quad , \quad (4.53c)$$

$$\bar{\mathbf{B}}_r = \left\{ \int_{-1}^{+1} \int_{-1}^{+1} \bar{\bar{\mathbf{U}}}(Q, P) \mathbf{N}(Q) |J_r| d\xi_1 d\xi_2 \right\} \quad , \quad (4.53d)$$

a eq.(4.52) é reescrita:

$$\bar{\mathbf{u}}_3(P) + \sum_{n=1}^N \mathbf{H}_n \mathbf{u}(q) = \sum_{n=1}^N \mathbf{G}_n \mathbf{t}(q) + \sum_{r=1}^R \mathbf{F}_r + \sum_{r=1}^R \bar{\mathbf{B}}_r \bar{\mathbf{u}}(Q) + \mathbf{c}^N(P) \bar{\mathbf{u}}_3(P) + \bar{\mathbf{v}}(P) \quad , \quad (4.54)$$

onde os termos convectivos \mathbf{c}^N só são considerados quando $P \equiv Q$. Definindo

$$\mathbf{B}_r = \left\{ \begin{array}{ll} \bar{\mathbf{B}}_r & \text{para } p \neq q \\ \bar{\mathbf{B}}_r + \mathbf{c}^N & \text{para } p \equiv q \quad , \end{array} \right\} \quad (4.55)$$

e sobrepondo as matrizes obtidas após o processo de colocação, resulta o seguinte sistema de equações:

$$\bar{\mathbf{u}}_3 + \mathbf{H}\mathbf{u} = \mathbf{G}\mathbf{t} + \mathbf{B}\bar{\mathbf{u}}_3 + \mathbf{f} \quad . \quad (4.56)$$

onde \mathbf{f} efetua a soma das contribuições de \mathbf{F}_r e $\bar{\mathbf{v}}$:

$$\mathbf{f} = \sum_{r=1}^R \mathbf{F}_r + \bar{\mathbf{v}}(P) \quad (4.57)$$

4.5 Forma geral dos sistemas de equações resultantes

Neste ponto torna-se necessário avaliar a estrutura dos sistemas de equações algébricas que resultam das classes de problemas que podem ser resolvidas com a formulação apresentada.

A seguir são resumidos os sistemas resultantes para o problema de membrana, de flexão e de cálculo dos gradientes dos deslocamentos:

- **Problema de membrana:**

$${}^m \mathbf{H} {}^m \mathbf{u} = {}^m \mathbf{G} {}^m \mathbf{t} + {}^m \mathbf{B} + {}^m \mathbf{f} \quad , \quad (4.58)$$

onde ${}^m \mathbf{f}$ é dado por (4.26).

- **Problema de flexão:**

$${}^f\mathbf{H} {}^f\mathbf{u} = {}^f\mathbf{G} {}^f\mathbf{t} + {}^f\mathbf{B} \bar{\mathbf{u}}_3 + {}^f\mathbf{f} \quad , \quad (4.59)$$

onde ${}^f\mathbf{f}$ é dado por (4.38).

- **Derivadas dos deslocamentos longitudinais:**

$$\bar{\mathbf{u}}_\beta + {}^\beta\mathbf{H} {}^m\mathbf{u} = {}^\beta\mathbf{G} {}^m\mathbf{t} + {}^\beta\mathbf{B} + {}^\beta\mathbf{f} \quad , \quad (4.60)$$

onde ${}^\beta\mathbf{f}$ é obtido dado por (4.48).

- **Derivadas dos deslocamentos transversais:**

$$\bar{\mathbf{u}}_3 + {}^3\mathbf{H} {}^f\mathbf{u} = {}^3\mathbf{G} {}^f\mathbf{t} + {}^3\mathbf{B} \bar{\mathbf{u}}_3 + {}^3\mathbf{f} \quad . \quad (4.61)$$

onde ${}^3\mathbf{f}$ é obtido dado por (4.57).

4.5.1 Elasticidade bidimensional:

Os problemas de elasticidade bidimensional estão desacoplados do problema de flexão de placas, de modo que a eq.(4.58) se degenera no sistema (Brebbia et al. 1984):

$${}^m\mathbf{H} {}^m\mathbf{u} = {}^m\mathbf{G} {}^m\mathbf{t} + {}^m\mathbf{f} \quad , \quad (4.62)$$

onde ${}^m\mathbf{f}$ é composto apenas pelos primeiro e último vetores de (4.26). Após o agrupamento das incógnitas em um vetor ${}^m\mathbf{x}$ resulta o sistema linear:

$$\boxed{{}^m\mathbf{A} {}^m\mathbf{x} = {}^m\mathbf{f}} \quad .$$

O vetor ${}^m\mathbf{x}$ contém os deslocamentos e trações de membrana.

4.5.2 Flexão linear elástica de placas:

Aqui novamente há um desacoplamento entre o problema de flexão e o problema de membrana, e a eq.(4.59) se degenera na forma (van der Weeën 1982a, van der Weeën

1982b):

$${}^f\mathbf{H} {}^f\mathbf{u} = {}^f\mathbf{G} {}^f\mathbf{t} + {}^f\mathbf{f} \quad . \quad (4.63)$$

Após o agrupamento das incógnitas em um vetor ${}^f\mathbf{x}$ resulta o sistema linear:

$$\boxed{{}^f\mathbf{A} {}^f\mathbf{x} = {}^f\mathbf{f}} \quad .$$

O vetor ${}^f\mathbf{x}$ contém os deslocamentos e esforços de flexão.

4.5.3 Estabilidade linear elástica de placas:

Neste caso, embora haja o acoplamento entre o problema de flexão e membrana, o último é parametrizado por (3.66), conforme descrito em 3.6.1. Assim, a única incógnita remanescente do problema de membrana é o fator de carga λ . Por outro lado, a placa é considerada perfeitamente plana e livre de quaisquer carregamentos transversais. Então reescreve-se (4.59) e (4.61) na forma (Marczak & de Barcellos 1993, Marczak 1998):

$${}^f\mathbf{H} {}^f\mathbf{u} = {}^f\mathbf{G} {}^f\mathbf{t} + \lambda {}^f\mathbf{B} \bar{\mathbf{u}}_3 \quad , \quad (4.64)$$

e

$$\bar{\mathbf{u}}_3 + {}^3\mathbf{H} {}^f\mathbf{u} = {}^3\mathbf{G} {}^f\mathbf{t} + \lambda {}^3\mathbf{B} \bar{\mathbf{u}}_3 \quad . \quad (4.65)$$

A manipulação destas equações leva a:

$${}^f\mathbf{A} {}^f\mathbf{x} = \lambda {}^f\mathbf{B} \bar{\mathbf{u}}_3 \quad , \quad (4.66)$$

$$\bar{\mathbf{u}} + {}^3\mathbf{A} {}^f\mathbf{x} = \lambda {}^3\mathbf{B} \bar{\mathbf{u}}_3 \quad , \quad (4.67)$$

que quando combinadas originam o seguinte problema clássico de autovalores e autovetores:

$$\boxed{[{}^3\mathbf{A} {}^f\mathbf{A}^{-1} {}^f\mathbf{B} - {}^3\mathbf{B}] \bar{\mathbf{u}}_3 = \frac{1}{\lambda} \bar{\mathbf{u}}_3} \quad . \quad (4.68)$$

O menor autovalor $1/\lambda$ é usado em (3.66) para fornecer a carga crítica, enquanto os autovetores $\bar{\mathbf{u}}_3$ podem ser substituídos em (4.66), fornecendo as extensões dos autovetores (${}^f\mathbf{x}$) que representam o padrão de deslocamentos e esforços sobre o contorno, para cada modo de flambagem. Para cálculo dos deslocamentos correspondentes a cada modo, em pontos internos, substitui-se $\bar{\mathbf{u}}_3$ e ${}^f\mathbf{x}$ (devidamente desmembrado em ${}^f\mathbf{u}$ e ${}^f\mathbf{t}$) na eq.(4.39), particularizada com $\mathbf{q} = \mathbf{v} = \mathbf{0}$.

4.5.4 Flexão geometricamente não-linear de placas:

Neste caso as eqs.(4.58), (4.59), (4.60) e (4.61) devem ser utilizadas sem qualquer simplificação, e um procedimento iterativo torna-se necessário para a solução das mesmas. Este procedimento está esquematizado na figura 4-6, que ilustra os passos de uma iteração genérica k .

Este algoritmo é geral, e pode ser generalizado para outras aplicações não-lineares. Além disso, note-se que a convergência foi verificada utilizando-se uma norma sobre as curvaturas (\bar{u}_3). Evidentemente esta não é a única possibilidade. A rigor, quaisquer variáveis de membrana ou flexão poderiam ser utilizadas. Em experimentos numéricos preliminares, entretanto, os valores das curvaturas mostraram-se dominantes no problema.

Os passos 2 e 4 do algoritmo ilustrado na figura 4-6 pressupõem a utilização das eqs.(4.60) e (4.61) para obtenção das derivadas dos deslocamentos em pontos internos. Este é o procedimento mais formal para cômputo destas variáveis dado que estas equações advêm das formas integrais integrais (3.76-3.77), que não impõem qualquer aproximação diretamente sobre o gradiente do campo de deslocamentos.

Por outro lado, deve-se ressaltar que a dedução das eqs.(4.60) e (4.61) possuem a inconveniente necessidade de diferenciar núcleos singulares. Em cada nova aplicação que exige a utilização de novos tensores estes devem ser novamente diferenciados, e possivelmente novos termos convectivos e/ou termos livres aparecerão. Esta dificuldade torna-se ainda maior em aplicações não-lineares, onde o número de termos integrais na identidade de Somigliana aumenta⁴. Existem algumas alternativas para obtenção de aproximações numéricas para $u_{i,\alpha}$ que podem ser aproveitadas nestes casos. Dado que os ter-

⁴Além do acoplamento flexão-membrana do presente caso, poder-se-ia citar os termos de deformações plásticas, termos dinâmicos, efeitos térmicos etc.

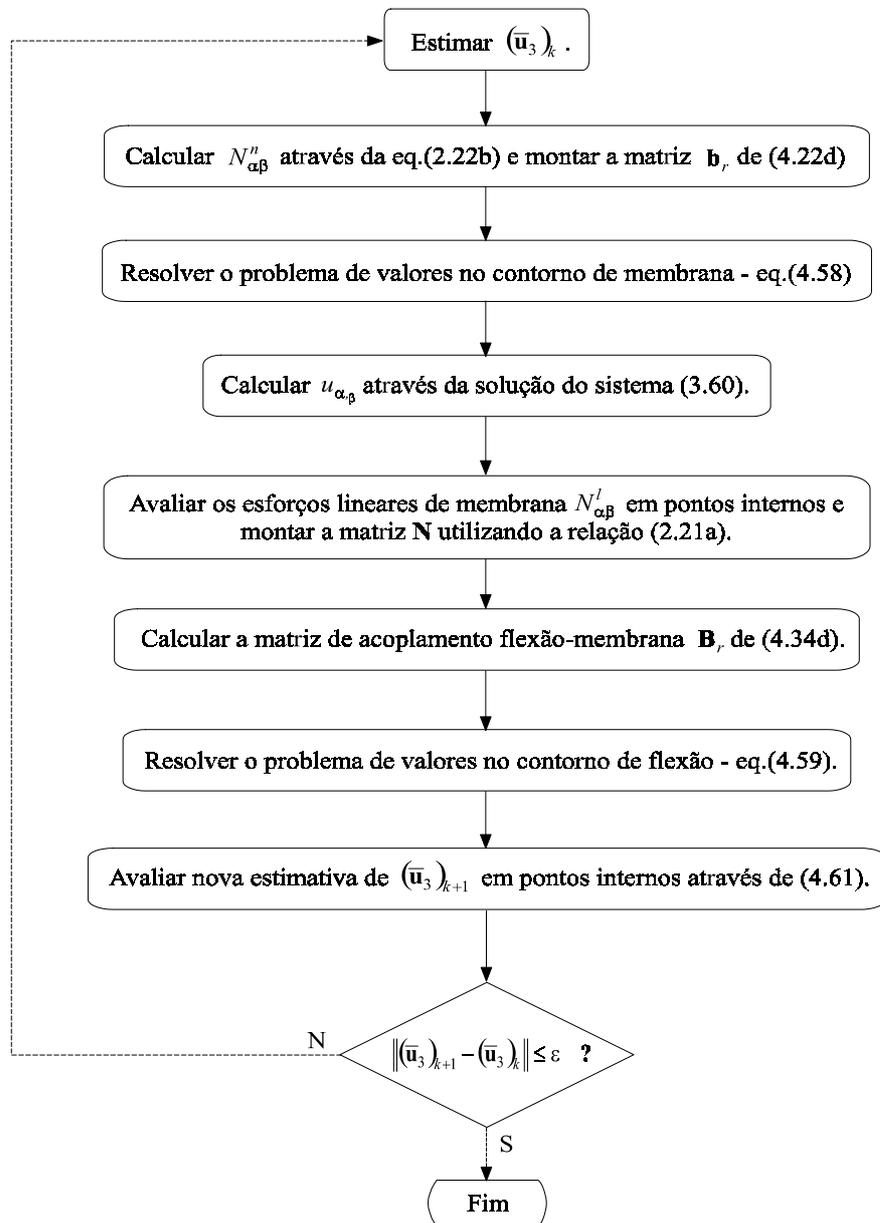


Figura 4-6: Algoritmo para solução iterativa das equações de flexão não-linear de placas.

mos de acoplamento flexão-membrana devem ser integrados sobre o domínio, as referidas derivadas devem ser calculadas nos nós físicos das células de domínio, aqui consideradas descontínuas.

Dentre as aproximações possíveis a mais popular é sem dúvida o uso de derivadas das funções de interpolação das células de domínio. Embora de fácil implementação, esta abordagem implica em um enfraquecimento significativo da função de interpolação global para estas variáveis, obrigando o uso de malhas mais refinadas para o domínio ou de células de alta ordem (Bathe 1996).

Uma técnica pouco utilizada para este propósito é o emprego de diferenças finitas, onde a derivada é aproximada da forma:

$$u_{i,\alpha}(p) = \frac{u_i(p + \Delta x_\alpha) - u_i(p - \Delta x_\alpha)}{2\Delta x_\alpha} \quad (4.69)$$

No âmbito do MEF, o uso de diferenças finitas para cálculo de gradientes de deslocamentos tem sido relegado a segundo plano. Isto é compreensível em se tratando de elementos finitos porque neste, como um método de domínio, a precisão das variáveis sobre o contorno ou no interior do domínio mantém a mesma dependência das funções de interpolação empregadas. No contexto do MEC, entretanto, vale lembrar que a precisão no cálculo das variáveis em pontos internos depende fundamentalmente da precisão das variáveis sobre o contorno. Desta forma, o uso de diferenças finitas torna-se particularmente atrativo porque a técnica torna-se insensível à distância Δx_α utilizada na eq.(4.69).

Embora o presente trabalho tenha contado com a implementação das eqs.(4.60) e (4.61) para problemas lineares, estas não foram extendidas para o caso não-linear. Em vez disto, utilizou-se um esquema de diferenças finitas em torno de cada nó físico das células de domínio. Como será mostrado no Capítulo 7, a precisão das derivadas assim obtidas é da mesma ordem que a fornecida pelas eqs.(4.60) e (4.61). Por outro lado, implica na solução das equações para deslocamentos em pontos internos, eqs.(3.48) e (3.65), para um número maior de pontos, gerando um aumento no tempo de pós-processamento do problema.

Outro aspecto importante do algoritmo da figura 4-6 é a atualização das estimativas das derivadas $u_{i,\alpha}$. Ao fim de uma iteração k , os novos valores calculados para estas

variáveis (que devem ser utilizados na iteração seguinte) não substituem completamente os antigos da forma:

$$\{u_{i,\alpha}\}_{k+1} = \{u_{i,\alpha}\}_k \quad . \quad (4.70)$$

Experimentos numéricos mostram que a atualização destas variáveis pela eq.(4.70) levam à convergência apenas para não-linearidades muito pequenas. Assim, tornou-se necessário utilizar um fator de relaxação para estabilizar o processo iterativo:

$$\{u_{i,\alpha}\}_{k+1} = \rho \{u_{i,\alpha}\}_k + (1 - \rho) \{u_{i,\alpha}\}_{k-1} \quad . \quad (4.71)$$

A razão disto está diretamente relacionada ao fato de se estar resolvendo um mapeamento contrativo das equações de membrana e de flexão. No Capítulo 7 serão discutidos alguns detalhes relacionados à eq.(4.71), bem como a faixa de valores para o fator de relaxação ρ utilizado neste trabalho.

4.6 Tratamento numérico de subregiões

A divisão do domínio sob análise em duas ou mais subregiões é uma técnica muito utilizada na mecânica computacional. Dependendo do método empregado para solução da equação diferencial governante, o uso de subregiões pode ser a única forma de resolver o problema discretizado. Domínios envolvendo áreas com materiais ou propriedades geométricas diferentes são exemplos clássicos. Outros exemplos interessantes são aqueles onde uma parte do domínio é governada por uma equação diferencial diferente da que governa a outra parte. Acoplamento fluido-estrutura e o uso de teorias estruturais diferentes em um mesmo problema são casos que se enquadram nesta categoria. Mesmo nos problemas onde poderia ser dispensado, o uso de subregiões pode ser benéfico por melhorar o condicionamento numérico dos sistemas de equações resultantes. Mais recentemente, a análise de problemas subdivididos em subregiões empregando computadores dotados de processamento paralelo tem mostrado grandes vantagens em termos de custo computacional, principalmente para soluções baseadas no MEF (McKenna 1997).

No âmbito do MEC, é sabido que as implementações convencionais do método encontram dificuldades numéricas em problemas cujas dimensões possuem razões de aspecto muito díspares, normalmente associadas ao mal condicionamento das matrizes envolvidas. Nestes casos, a divisão do domínio em subregiões com razão de aspecto mais favorável leva a uma melhora bastante significativa no condicionamento numérico (Rudolphi 1983). Adicionalmente, os algoritmos de solução podem tirar grande vantagem das matrizes resultantes, normalmente dominadas por blocos, com significativa economia no armazenamento de dados (Santiago & Telles 1997, Santiago et al. 1999). O acoplamento do MEC com outros métodos como o MEF também faz uso de subregiões, em cujas interfaces pode-se compatibilizar as variáveis de variadas maneiras, dada a natureza diferente dos sistemas matriciais originados por ambos (Kita & Kamiya 1994).

Em outros casos ainda, não se dispõe de soluções fundamentais adequadas para o problema e o uso de subregiões pode ajudar a resolver o problema. Exemplos destes são os problemas espaciais (sem solução fundamental) que podem ser descritos por superposição de problemas planos (com solução fundamental mais simples e conhecida), como o ilustrado na figura 4-7.

Estruturas deste tipo são facilmente modeladas como uma composição de placas planas (*plate assembled structures*). Basicamente, cada parede da estrutura é representada por uma placa com um sistema de coordenadas $(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3)$, onde o eixo \mathbf{x}_3 é perpendicular à sua superfície de referência (figura 4-7). É evidente a necessidade de se acoplar o problema de membrana ao de flexão quando duas paredes não são coplanares. Um caso particular deste são os problemas modelados por múltiplas placas de espessuras e/ou materiais diversos (figura 4-8).

Um dos objetivos deste trabalho é desenvolver um procedimento numérico que permita a manipulação de um número arbitrário de subregiões. Algoritmos para manipulação de um pequeno número de subregiões são facilmente encontrados na literatura (Brebbia et al. 1984, Brebbia & Dominguez 1989). No entanto, procedimentos gerais são bem mais escassos.

Primeiramente, são necessários alguns comentários sobre as equações que serão resolvidas e suas modificações para o emprego das mesmas em subregiões:

- As equações necessárias para solução do caso mais completo contemplado neste tra-

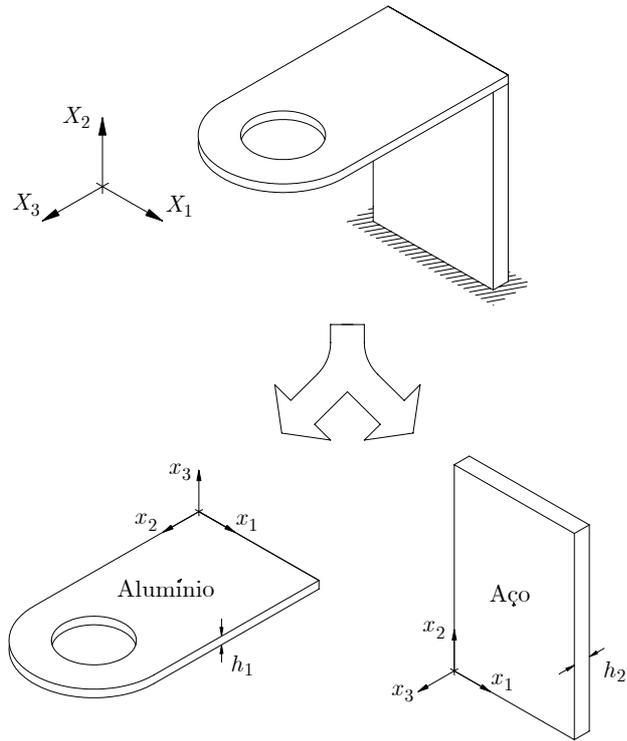


Figura 4-7: Um exemplo típico de estrutura modelada por placas planas.

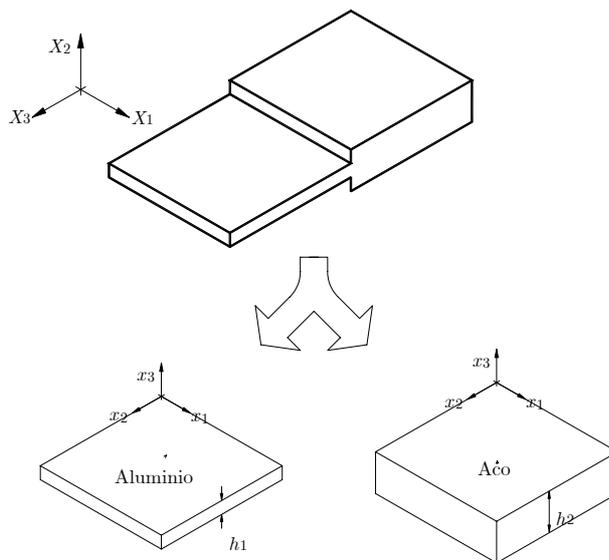


Figura 4-8: Placa plana com espessura variável.

balho são as eqs.(4.58), (4.59), (4.60) e (4.61). Dado que estas podem ser reduzidas ao casos linear e de flambagem, o procedimento a seguir não perde em generalidade.

- As eqs.(4.58) e (4.59) lidam com as variáveis definidas sobre o contorno. Como a conexão de duas ou mais subregiões se dá sobre a parcela comum do contorno, estas equações serão diretamente afetadas. Por outro lado, as eqs.(4.60) e (4.61) são utilizadas para o cálculo de derivadas dos deslocamentos apenas em pontos internos de cada subregião, a partir do conhecimento das variáveis sobre seu contorno. Portanto, as eqs.(4.60) e (4.61) são tratadas como um *pós-processamento* realizado após a solução dos problemas de valores no contorno de membrana e de flexão, para cada subregião.
- Seguindo o algoritmo descrito na fig.(4-6), a cada iteração da solução uma estimativa do vetor de derivadas nos pontos internos está disponível. Estas estimativas são utilizadas no termo ${}^m\mathbf{f}$ de (4.58), e também no termo ${}^f\mathbf{B}\bar{\mathbf{u}}_3$ de (4.59), originando um vetor que pode ser adicionado ao termo ${}^f\mathbf{f}$ daquela equação. Consequentemente, cada subregião origina um sistema de equações do tipo:

$${}^m\mathbf{H} {}^m\mathbf{u} = {}^m\mathbf{G} {}^m\mathbf{t} + {}^m\mathbf{f} \quad (4.72a)$$

$${}^f\mathbf{H} {}^f\mathbf{u} = {}^f\mathbf{G} {}^f\mathbf{t} + {}^f\mathbf{f} \quad . \quad (4.72b)$$

Ou seja, a forma básica dos sistemas resultantes é a mesma. É possível ainda agrupar-se as equações (4.72) em um único sistema cujas variáveis sobre o contorno concatenam os vetores (4.12a) e (4.12b), (4.12c) e (4.12d) e também (4.12e) e (4.12f). Qualquer que seja o caso, o procedimento detalhado a seguir pode ser empregado diretamente.

- Pressupõe-se que as eqs.(4.72) já estão referenciadas a um sistema de coordenadas comum ($\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3$). Portanto, as variáveis sobre o contorno consideram três translações sobre os eixos globais e duas rotações, que permanecem referenciadas ao sistema de coordenadas local de cada subregião.

Seja o domínio Ω dividido em M subregiões Ω_i com contornos Γ_i . Cada uma destas subregiões Ω_i possui até N interfaces Γ^{ij} unindo-a a outras subregiões Ω_j ,

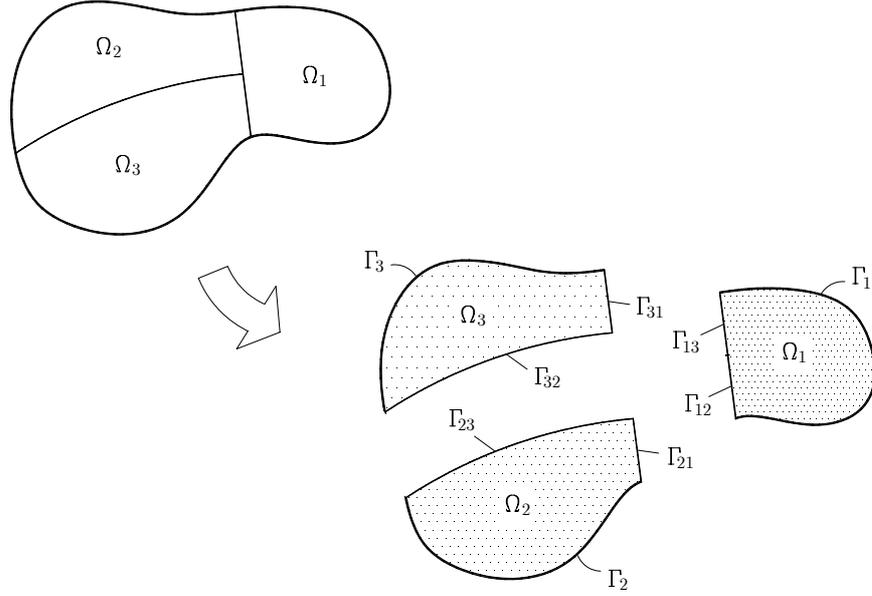


Figura 4-9: Exemplo de divisão de domínio em subregiões ($M = 3$, neste caso).

tal que $\Gamma^{i1} \cup \Gamma^{i2} \cup \dots \cup \Gamma^{in_s} \cup \Gamma^i = \Gamma_i$ e $\bigcup_{j=1}^{n_s} \Omega_j = \Omega$ (figura 4-9). Chamando de u^{ij} e t^{ij} os deslocamentos e as trações que ocorrem na interface Γ^{ij} e u^i e t^i os deslocamentos e as trações que ocorrem na parcela complementar de Γ_i , pode-se escrever as eqs.(4.72), para cada subregião Ω_i , na forma:

$$\begin{aligned}
 & \left[\mathbf{H}^i \quad \mathbf{H}^{i2} \quad \mathbf{H}^{i3} \quad \dots \quad \mathbf{H}^{iN} \right] \left\{ \begin{array}{c} \mathbf{u}^i \\ \mathbf{u}^{i2} \\ \mathbf{u}^{i3} \\ \vdots \\ \mathbf{u}^{iN} \end{array} \right\} = \\
 & = \left[\mathbf{G}^i \quad \mathbf{G}^{i2} \quad \mathbf{G}^{i3} \quad \dots \quad \mathbf{G}^{iN} \right] \left\{ \begin{array}{c} \mathbf{t}^i \\ \mathbf{t}^{i2} \\ \mathbf{t}^{i3} \\ \vdots \\ \mathbf{t}^{iN} \end{array} \right\} + \left\{ \begin{array}{c} \mathbf{b}^i \\ \mathbf{b}^{i2} \\ \mathbf{b}^{i3} \\ \vdots \\ \mathbf{b}^{iN} \end{array} \right\} \quad i = 1 \dots M \quad (4.73)
 \end{aligned}$$

Utilizando-se as condições de compatibilidade nas interfaces:

$$\mathbf{u}^{ij} = \mathbf{u}^{ji} \quad (4.74a)$$

$$\mathbf{t}^{ij} = -\mathbf{t}^{ji} \quad , \quad (4.74b)$$

pode-se isolar as variáveis remanescentes sobre todas as interfaces e seus contornos complementares Γ^i em um único vetor:

$$\mathbf{x} = \left\{ \begin{array}{cccccccc} \mathbf{u}^1 & \mathbf{t}^1 & \mathbf{u}^{12} & \mathbf{t}^{12} & \mathbf{u}^{13} & \mathbf{t}^{13} & \dots & \mathbf{u}^{1,N-1} & \mathbf{t}^{1,N-1} & \mathbf{u}^{1,N} & \mathbf{t}^{1,N} \\ \mathbf{u}^2 & \mathbf{t}^2 & \mathbf{u}^{23} & \mathbf{t}^{23} & \mathbf{u}^{24} & \mathbf{t}^{24} & \dots & \mathbf{u}^{2,N-1} & \mathbf{t}^{2,N-1} & \mathbf{u}^{2,N} & \mathbf{t}^{2,N} \\ \mathbf{u}^3 & \mathbf{t}^3 & \mathbf{u}^{34} & \mathbf{t}^{34} & \mathbf{u}^{35} & \mathbf{t}^{35} & \dots & \mathbf{u}^{3,N-1} & \mathbf{t}^{3,N-1} & \mathbf{u}^{3,N} & \mathbf{t}^{3,N} \\ \vdots & & & & & & & & & & \\ \mathbf{u}^{M-1} & \mathbf{t}^{M-1} & \mathbf{u}^{M-1,N} & \mathbf{t}^{M-1,N} & & & & & & & \\ \mathbf{u}^M & \mathbf{t}^M & & & & & & & & & \end{array} \right\}^T . \quad (4.75)$$

A fim de simplificar a notação, é possível reescrever a eq.(4.75) como uma concatenação de vetores:

$$\mathbf{x} = \left\{ \mathbf{x}^1 \quad \mathbf{x}^2 \quad \dots \quad \mathbf{x}^M \right\}^T$$

onde

$$\mathbf{x}^i = \left\{ \mathbf{u}^i \quad \mathbf{t}^i \quad \mathbf{u}^{i,i+1} \quad \mathbf{t}^{i,i+1} \quad \mathbf{u}^{i,i+2} \quad \mathbf{t}^{i,i+2} \quad \dots \quad \mathbf{u}^{i,N} \quad \mathbf{t}^{i,N} \right\}^T \quad i = 1 \dots M ; i < N . \quad (4.76)$$

Note-se que cada vetor \mathbf{x}^i contém os deslocamentos e trações de Γ^i e de todos Γ^{ij} para $i < j$. Isto é, cada vetor \mathbf{x}^i considera os graus de liberdade da parcela do contorno de Ω_i que não se conecta com nenhuma outra subregião bem como os graus de liberdade de todas as interfaces das subregiões seguintes a Ω_i (que se conectam com este) seguindo uma ordem de numeração. Portanto, um dado \mathbf{x}^j não precisa levar em conta os deslocamento \mathbf{u}^{ji} (onde $j > i$) porque estes já estão incluídos em \mathbf{x}^i através de \mathbf{u}^{ij} , antecipando o uso da relação (4.74a). Deste modo o número de vetores \mathbf{u} e \mathbf{t} participantes de cada \mathbf{x}^i vai

decrecendo à medida que se incrementa i . Mais explicitamente:

$$\begin{aligned}
\mathbf{x}^i &= \left\{ \mathbf{u}^i \quad \mathbf{t}^i \quad \mathbf{u}^{i,i+1} \quad \mathbf{t}^{i,i+1} \quad \mathbf{u}^{i,i+2} \quad \mathbf{t}^{i,i+2} \quad \dots \quad \mathbf{u}^{i,M-1} \quad \mathbf{t}^{i,M-1} \right\}^T, \quad i = 1 \dots M-2 \\
\mathbf{x}^{M-1} &= \left\{ \mathbf{u}^{M-1} \quad \mathbf{t}^{M-1} \quad \mathbf{u}^{M-1,M} \quad \mathbf{t}^{M-1,M} \right\}^T \\
\mathbf{x}^M &= \left\{ \mathbf{u}^M \quad \mathbf{t}^M \right\}^T.
\end{aligned} \tag{4.77}$$

Aplicando-se as eqs.(4.74) à eq.(4.73) para cada subregião e usando o vetor (4.76) como vetor de incógnitas resulta o seguinte sistema para todo o domínio Ω :

$$\mathbf{Ax} = \begin{bmatrix} \mathbf{A}^1 & \mathbf{A}^2 & \dots & \mathbf{A}^{M-1} & \mathbf{A}^M \end{bmatrix} \mathbf{x} = \mathbf{b} \tag{4.78}$$

onde a matrix \mathbf{A} é concatenada a partir da seguinte forma geral para as matrizes \mathbf{A}^i :

$$\mathbf{A}^1 = \begin{bmatrix} \mathbf{H}^1 & -\mathbf{G}^1 & \mathbf{H}^{12} & -\mathbf{G}^{12} & \mathbf{H}^{13} & -\mathbf{G}^{13} & \dots & \mathbf{H}^{1,N-1} & -\mathbf{G}^{1,N-1} & \mathbf{H}^{1,N} & -\mathbf{G}^{1,N} \\ 0 & 0 & \mathbf{H}^{21} & \mathbf{G}^{21} & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathbf{H}^{31} & \mathbf{G}^{31} & \dots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & \mathbf{H}^{N-1,1} & \mathbf{G}^{N-1,1} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & \mathbf{H}^{N,1} & \mathbf{G}^{N,1} \end{bmatrix}$$

$$\mathbf{A}^2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ \mathbf{H}^2 & \mathbf{G}^2 & \mathbf{H}^{23} & -\mathbf{G}^{23} & \mathbf{H}^{24} & -\mathbf{G}^{24} & \dots & \mathbf{H}^{2,N-1} & -\mathbf{G}^{2,N-1} & \mathbf{H}^{2,N} & -\mathbf{G}^{2,N} \\ 0 & 0 & \mathbf{H}^{32} & \mathbf{G}^{32} & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathbf{H}^{42} & \mathbf{G}^{42} & \dots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & \mathbf{H}^{N-1,2} & \mathbf{G}^{N-1,2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & \mathbf{H}^{N,2} & \mathbf{G}^{N,2} \end{bmatrix}$$

$$\begin{aligned}
\mathbf{A}^3 &= \begin{bmatrix}
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{H}^3 & \mathbf{G}^3 & \mathbf{H}^{34} & -\mathbf{G}^{34} & \mathbf{H}^{35} & -\mathbf{G}^{35} & \dots & \mathbf{H}^{3,N-1} & -\mathbf{G}^{3,N-1} & \mathbf{H}^{3,N} & -\mathbf{G}^{3,N} \\
\mathbf{0} & \mathbf{0} & \mathbf{H}^{43} & \mathbf{G}^{43} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{H}^{N-1,3} & \mathbf{G}^{N-1,3} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} & \mathbf{H}^{N,3} & \mathbf{G}^{N,3}
\end{bmatrix} \\
&\vdots \\
\mathbf{A}^{M-1} &= \begin{bmatrix}
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\vdots & \vdots & \vdots & \vdots \\
\mathbf{H}^{N-1,N-1} & -\mathbf{G}^{N-1,N-1} & \mathbf{H}^{N-1,N} & -\mathbf{G}^{N-1,N} \\
\mathbf{0} & \mathbf{0} & \mathbf{H}^{N,N-1} & \mathbf{G}^{N,N-1}
\end{bmatrix} \\
\mathbf{A}^M &= \begin{bmatrix}
\mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} \\
\vdots & \vdots \\
\mathbf{0} & \mathbf{0} \\
\mathbf{H}^N & -\mathbf{G}^N
\end{bmatrix} . \tag{4.79}
\end{aligned}$$

Agora o sistema (4.78) pode ser resolvido uma vez que se prescreva as condições de contorno sobre as parcelas Γ^i de cada subregião. O número de graus de liberdade resultante após a aplicação das condições de contorno é igual ao número total de graus de liberdade sobre os contornos Γ^i mais duas vezes o número de graus de liberdade sobre os contornos Γ^{ij} . Adicionalmente, a matriz \mathbf{A} se torna esparsa, com uma banda dependente da numeração adotada para as subregiões e seus elementos de contorno. Este aspecto

também pode ser explorado para otimizar o armazenamento de dados (Santiago et al. 1999).

4.7 Resumo do Capítulo 4

As quatro equações integrais (3.68), (3.70), (3.76) e (3.77) obtidas no Capítulo 2 foram desenvolvidas neste capítulo em sua forma discretizada, utilizando a metodologia padrão do MEC direto, resultando nas eqs.(4.21), (4.33), (4.43), e (4.52), respectivamente. Um processo de colocação destas equações gera os sistemas dados pelas eqs.(4.58), (4.59), (4.60) e (4.61), que representam uma forma geral do problema de flexão geometricamente não-linear de placas. Mesmo a inclusão de outros tipos de excitação fica contemplada a partir de sua adição aos termos de carregamento. Desses sistemas pode-se extrair ainda, como casos particulares, os problemas de elasticidade bidimensional, flexão linear e flambagem de placas. Um procedimento iterativo foi idealizado para a solução de problemas de flexão geometricamente não-linear, a fim de permitir a solução acoplada dos sistemas de equações. Também foi desenvolvida uma estratégia para manipulação de domínios particionados em um número arbitrário de subregiões, permitindo a solução de problemas mais complexos.

Tendo em vista a necessidade de manipulação de integrais contendo núcleos singulares em todas as equações abordadas, torna-se mandatória a adoção de procedimentos especiais para uma integração numérica precisa destes termos. Os procedimentos adotados com esta finalidade no presente trabalho serão tratados no próximo capítulo.

Capítulo 5

Tratamento das Integrais Singulares

Como citado no Capítulo 4, a montagem dos sistemas de equações (4.58), (4.59), (4.60) e (4.61) se utiliza das submatrizes elementares calculadas ao longo do processo de colocação, dadas respectivamente pelas eqs.(4.22), (4.34), (4.44) e (4.53). Todas estas últimas são obtidas através de integração numérica de núcleos que podem conter algum tipo de singularidade. Como as regras baseadas em quadratura Gaussiana padrão não são convergentes (ou apresentam convergência muito lenta) para estas situações, este capítulo detalha as abordagens utilizadas para tratamento numérico das integrais singulares no contexto do presente trabalho.

5.1 Integração numérica de núcleos singulares

Durante o processo de colocação, as submatrizes \mathbf{H}_n , \mathbf{G}_n e \mathbf{F}_r , \mathbf{B}_r são calculadas através de integração numérica unidimensional (sobre Γ_n) e bidimensional (sobre Ω_r), respectivamente. Quando o ponto fonte p ou P não pertence ao elemento Γ_n ou à célula Ω_r que está sendo integrada todas as integrais referentes à estas submatrizes são regulares e, por estarem definidas em domínios normalizados, podem ser integradas numericamente através de quadratura de Gauss (Stroud & Secrest 1966):

$$I_1 = \int_{-1}^{+1} f(\xi_1) d\xi_1 = \sum_{i=1}^K f(\xi_{1_i}) \omega_i \quad , \quad (5.1a)$$

$$I_2 = \int_{-1}^{+1} \int_{-1}^{+1} f(\xi_1, \xi_2) d\xi_1 d\xi_2 = \sum_{i=1}^K \sum_{j=1}^K f(\xi_{1_i}, \xi_{2_j}) \omega_i \omega_j \quad , \quad (5.1b)$$

onde K é o número de pontos de integração ξ_i necessários para integrar de maneira exata um função polinomial $f(\xi)$ de grau $p = 2K - 1$, e ω_i são os respectivos pesos. No entanto, as integrais aqui em questão não envolvem apenas polinômios, de modo que o número de pontos de integração necessários para o cálculo das referidas submatrizes, com precisão suficiente, deve ser determinado experimentalmente. Na implementação do código numérico dispôs-se, com este objetivo, de 1(1)10, 12(2)20, 24(6)48 e 64 pontos de integração de Gauss-Legendre, sendo utilizados quando $p, P \notin \Gamma_n, \Omega_r$.

Quando $p, P \in \Gamma_n, \Omega_r$, as integrais das submatrizes citadas acima tornam-se impróprias devido à presença de núcleos singulares nos tensores envolvidos (Brebbia et al. 1984). Nestes casos, as singularidades presentes são sumariadas na tabela 5.1.

A fim de distinguir a severidade das diversas singularidades, foi adotada a seguinte terminologia na tabela 5.1, bem como ao longo deste trabalho: seja uma integral em um domínio d -dimensional contendo um núcleo $1/r^\alpha$. Se $\alpha < d$, denominamos a integral de fracamente singular. Esta categoria também engloba as integrais contendo núcleos logarítmicos. Se $\alpha = d$, denominamos a integral de fortemente singular (normalmente interpretadas no sentido do valor principal de Cauchy), enquanto que se $\alpha > d$ a integral é denominada hipersingular, ou integrais de Hadamard.

Adicionalmente, denomina-se de integrais quase-singulares aquelas cujo pólo singular não se encontra dentro do domínio de integração, mas suficientemente próximo deste a ponto de dificultar seu cômputo por quadratura gaussiana padrão, devido aos altos gradientes do integrando em relação às variáveis de integração.

Submatriz	Equação(ões)	Núcleo	Domínio	Singularidade	Ocorrência
$\bar{\mathbf{H}}_n$	4.22a e 4.34a	$1/r$	1-D	Forte	$p \equiv q$
\mathbf{G}_n	4.22b e 4.34b	$\ln(r)$	1-D	Fraca	$p \equiv q$
\mathbf{F}_r	4.22c e 4.34c	$1/r$	2-D	Fraca ¹	$p, P \equiv Q$
$\mathbf{b}_r, \mathbf{B}_r$	4.22d e 4.34d	$1/r$	2-D	Fraca	$p, P \equiv Q$
\mathbf{H}_n	4.44a e 4.53a	$1/r^2$	1-D	Hipersingular	$p \equiv q$
\mathbf{G}_n	4.44b e 4.53b	$1/r$	1-D	Forte	$p \equiv q$
\mathbf{F}_r	4.44c	$1/r$	2-D	Fraca	$P \equiv Q$
\mathbf{F}_r	4.53c	$1/r^2$	2-D	Forte ²	$P \equiv Q$
$\bar{\mathbf{B}}_r$	4.44d e 4.53d	$1/r^2$	2-D	Forte	$P \equiv Q$

¹Para o modelo de Reissner, apenas. Caso contrário, esta submatriz é regular.

²Para o modelo de Reissner, apenas. Caso contrário, esta submatriz é fracamente singular.

Tabela 5.1: Tipos de singularidade presentes nas submatrizes

O cálculo numérico de integrais contendo núcleos singulares é objeto de intensa pesquisa a pelo menos trinta anos. O número de métodos e quadraturas numéricas propostas na literatura é bastante significativo, mas são na sua maioria aplicáveis a integrais contendo núcleos com um tipo específico de singularidade. Apesar dos esforços no sentido de desenvolver uma quadratura genérica que permita a integração tanto de núcleos regulares como singulares (Dumont 1992, Rosen & Cormack 1994, Rosen & Cormack 1995), tal objetivo parece ainda não ter sido satisfatoriamente alcançado. Um método com tal generalidade e de implementação fácil é provalvemente impossível de ser desenvolvido dadas as naturezas diversas dos funcionais lineares representados por integrais singulares. Com efeito, integrais por parte-finita são funcionais contínuos e lineares que incluem as integrais regulares (Kutt 1975*c*), de forma que métodos aplicáveis às integrais regulares não podem ser diretamente estendidos para tratar integrais impróprias.

Assim, torna-se conveniente (pelo menos para integrais fortemente singulares ou hiper-singulares) o isolamento dos termos singulares dos integrandos e a aplicação de abordagens especiais à cada termo. O uso deste expediente é precisamente o aspecto que tem dificultado a implementação de códigos gerais de elementos de contorno. Cada nova solução fundamental ou tipo de elemento implementado necessita de um tratamento analítico para isolar os termos singulares que serão integrados com um método particular. Outras vezes ainda estes termos não se enquadram exatamente em uma classe específica de singularidade, isto é, são compostos por combinações (não-separáveis) de núcleos que podem dificultar uma integração precisa com um método desenvolvido para um tipo particular de singularidade.

Uma análise dos métodos de integração disponíveis não será realizada aqui. Tal análise revela, entretando, que é possível enquadrar a grande maioria destes em uma (ou uma combinação) das técnicas gerais apresentadas na tabela 5.2.

5.1.1 Integrais fracamente singulares

Integrais fracamente singulares são reconhecidamente o tipo mais tratável de integrais singulares. Efetivamente, estas integrais não se enquadram na categoria de integrais divergentes e podem ser integradas com quadratura de Gauss-Legendre padrão, mas requerem um número proibitivamente alto de pontos de integração para convergirem. Diversas téc-

Grupo	Técnica utilizada
I	Integração analítica ou semi-analítica.
II	Integração sobre o contorno do domínio, usando variações do teorema da divergência.
III	Quadraturas desenvolvidas para tipos específicos de singularidade.
IV	Transformação de coordenadas. Troca de variáveis.
V	Expansão dos núcleos em séries com posterior tratamento analítico. Adição/subtração dos termos singulares.
VI	Embutimento invariante. Integração escalada. Métodos de continuação.
VII	Regularização analítica ou semi-analítica dos núcleos.
VIII	Subdivisão do intervalo de integração. Aumento adaptativo do número de pontos de integração.

Tabela 5.2: Classificação genérica das principais técnicas de tratamento de integrais singulares.

nicas eficientes já foram desenvolvidas para cálculo deste tipo de integral, sendo que a maioria se enquadra nos grupos I, III, IV e VIII da tabela 5.2.

Como este tipo de singularidade é convergente, o método mais eficiente para sua integração é a utilização de quadraturas cujos pontos de integração são determinados por funções-peso logarítmicas (Stroud & Secrest 1966). Entretanto, a quadratura é válida para o intervalo normalizado $[0, 1]$, com o pólo da singularidade localizado na origem do sistema de coordenadas o que obriga a alguns algebrismos para integrar funções sobre intervalos $[-1, +1]$. Além disso, o isolamento da parcela logarítmica do integrando é mandatório. Existem algumas variações destas quadraturas para integração sobre $[-1, +1]$. Nahlik & Bialecki (1983) desenvolveram uma quadratura deste tipo onde a o pólo está sobre a origem do sistema de coordenadas e, portanto, pode ser diretamente utilizada para integração de elementos constantes no MEC. Recentemente, Smith (2000) propôs uma quadratura similar diretamente aplicável a elementos quadráticos ou cúbicos, onde o ponto singular pode estar posicionado sobre quaisquer nós do elemento.

Técnicas de subdivisão progressiva do intervalo de integração e utilização de quadratura gaussiana a cada subintervalo ou aplicação de troca de variáveis não exigem isolamento dos termos logarítmicos (Lachat & Watson 1976, Jun et al. 1985). No entanto, implicam um esforço computacional intenso, pois o número efetivo de pontos de integração cresce abruptamente. Uma alternativa enquadrada nesta mesma categoria é aplicar um aumento

iterativo do número de pontos de integração até atingir a convergência (Patterson 1968). A vantagem desta abordagem está na reutilização dos pontos de integração e pesos da última iteração, que não mudam de valor. Deste modo os somatórios de (5.1) não são totalmente recalculados, apenas são adicionados novos termos a cada iteração. Por outro lado, os pontos de Gauss assim determinados não estão em suas coordenadas ótimas.

As técnicas de transformação de coordenadas são boas alternativas para integrações deste tipo e têm se tornado muito populares, pois não exigem o isolamento da singularidade (Telles 1987, Telles & Oliveira 1994, Johnston & Elliott 2001). Com isto permitem a utilização do mesmo algoritmo de cálculo das integrais regulares. A severidade da transformação é auto-adaptativa, perdendo o efeito à medida que o ponto fonte se distancia do domínio de integração, o que torna o método diretamente aplicável à casos quase-singulares (Telles & Oliveira 1994). Em essência, esses procedimentos são transformações não-lineares de coordenadas cujo Jacobiano se anula no ponto singular. Assim obtém-se uma concentração dos pontos de Gauss em torno do pólo singular, acelerando significativamente a convergência em relação à quadratura padrão. A extensão para domínios bi ou tridimensionais é imediata.

As figuras 5-1 e 5-2 ilustram uma comparação entre as quadraturas de Gauss-Legendre padrão e com função-peso logarítmica, a quadratura de Nahlik & Bialecki (1983) e a transformação de Telles (1987). Desta comparação fica claro que a integração de núcleos logarítmicos através de quadraturas com função-peso logarítmicas é imbatível quando o núcleo pode ser isolado.

Apesar do excelente desempenho das quadraturas com função-peso logarítmica, deve-se lembrar que os tensores utilizados neste trabalho possuem, além dos termos puramente logarítmicos, funções de Bessel modificadas de primeira e segunda espécies. As funções de Bessel foram implementadas numericamente através de expansões polinomiais e utilizando bibliotecas numéricas (*HARWELL Subroutine Library Specification: EB06A* 1978). Tais expansões dificultam sobremaneira o isolamento das singularidades dos tensores, motivo pelo qual adotou-se o procedimento de Telles (1987) para cálculo de integrais fracamente singulares ($\ln(r)$ sobre Γ_n e $1/r$ sobre Ω_r), com 2(2)20, 24(6)48 e 64 pontos de integração.

Nos tensores fundamentais dos modelos de placa empregados, as funções de Bessel $K_0(z)$ e $K_1(z)$ não aparecem somente isoladas, mas também combinadas nos termos (ver

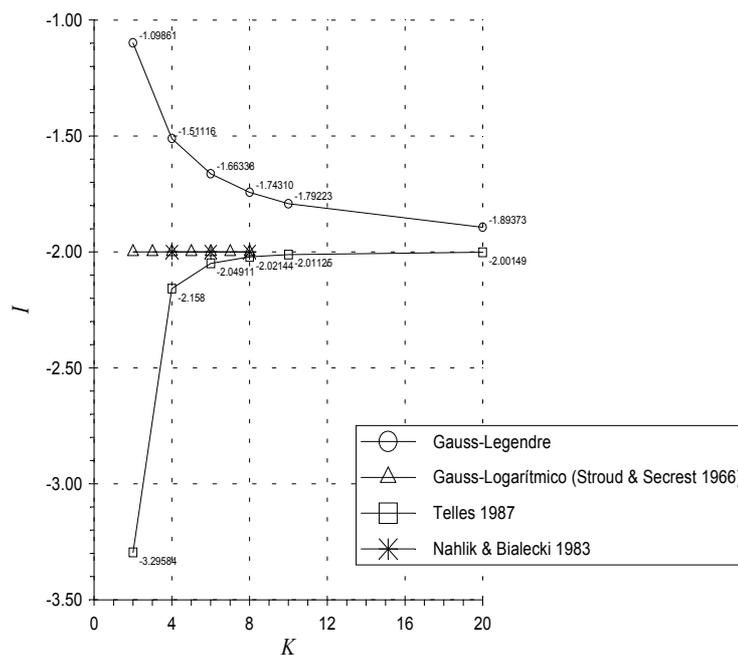


Figura 5-1: Desempenho de algumas quadraturas para integrais contendo núcleos logarítmicos. $I = \int_{-1}^{-1} \ln(|x|) dx = -2.0$.

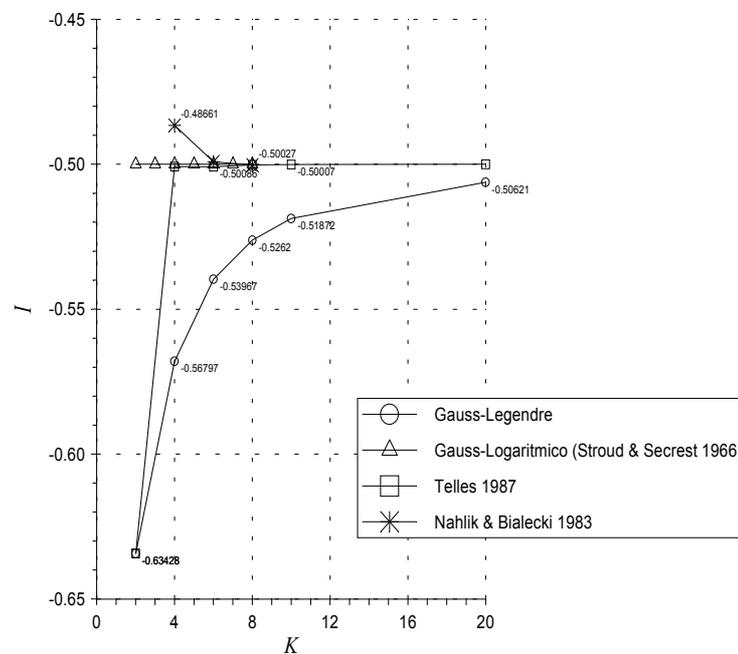


Figura 5-2: Desempenho de algumas quadraturas para integrais contendo núcleos logarítmicos. $I = \int_{-1}^{-1} x \ln(|x|) dx = -0.5$.

K	$\int_{-1}^{+1} A(z) dz$		$\int_{-1}^{+1} B(z) dz$	
	Gauss-Legendre	Telles	Gauss-Legendre	Telles
2	-0.7559794575	-0.7722537595	0.2520970798	0.2911975590
4	-0.7534558720	-0.7536327129	0.3285977752	0.3594333398
6	-0.7534122728	-0.7534075558	0.3451181697	0.3600912946
8	-0.7534084187	-0.7534074965	0.3513717854	0.3601258403
10	-0.7534077527	-0.7534074979	0.3543974382	0.3601318258
12	-0.7534075866	-0.7534074979	0.3560884257	0.3601333201
14	-0.7534075341	-0.7534074979	0.3571284551	0.3601337924
16	-0.7534075145	-0.7534074979	0.3578134154	0.3601339687
18	-0.7534075062	-	0.3582883260	0.3601340431
20	-0.7534075024	-	0.3586310639	0.3601340777
24	-0.7534074994	-	0.3590819314	0.3601341043
32	-0.7534074982	-	0.3595362564	0.3601341156
40	-0.7534074980	-	0.3597491448	0.3601341175
48	-0.7534074979	-	0.3598656810	0.3601341179
64	-0.7534074979	-	0.3599823460	0.3601341181

Tabela 5.3: Resultados da integração numérica de $A(z)$ e $B(z)$.

Apêndice A):

$$A(z) = K_0(z) + \frac{2}{z} \left(K_1(z) - \frac{1}{z} \right) \quad (5.2a)$$

$$B(z) = K_0(z) + \frac{1}{z} \left(K_1(z) - \frac{1}{z} \right) \quad (5.2b)$$

A presença dos fatores $1/z$ leva à conclusão prematura de que as expressões 5.2 são fortemente singulares. Uma análise cuidadosa destas expressões revela que na verdade $A(z)$ é regular e $B(z)$ tem comportamento logarítmico quando $z \rightarrow 0$. Ainda assim, sua integração é mais difícil que outros termos puramente logarítmicos, particularmente para $B(z)$. O conhecimento do número de pontos de integração necessários para a integração precisa de 5.2 revelou-se fator determinante na qualidade dos resultados para flexão de placas. Com esse objetivo, foi feito um levantamento experimental da convergência dessas expressões, integradas isoladamente sobre um intervalo $[-1, +1]$, empregando quadratura gaussiana padrão e com transformação cúbica. Os resultados estão apresentados na tabelas 5.3. Note-se que embora $A(z)$ possa ser integrada apenas com pontos de Gauss-Legendre, o uso da transformação cúbica permite o uso de apenas 6 pontos para uma precisão de 6 dígitos. Já no caso de $B(z)$ o uso da transformação torna-se indispensável, e ainda assim são requeridos 18 pontos de Gauss para a mesma precisão.

5.1.2 Integrais fortemente singulares

Para integrais fortemente singulares vários procedimentos de integração estão disponíveis na literatura, sendo que a grande maioria deles exige isolamento da singularidade. Os grupos I, III, V e VII da tabela 5.2 enquadram a maioria dos casos. Quando tratadas isoladamente, estas integrais são comumente calculadas como integrais por parte-finita ou como valor principal de Cauchy (Krommer & Ueberhuber 1998).

No âmbito do MEC, uma estratégia muito comum é a imposição de movimentos arbitrários de corpo rígido, que é uma forma de se utilizar uma solução particular do problema para se calcular indiretamente os termos singulares (Brebbia et al. 1984). A imposição de movimentos de corpo rígido permite o cálculo das submatrizes que multiplicam os deslocamentos nodais nos pontos fonte, já acrescidas dos fatores geométricos \mathbf{C} , a segunda das equações (4.24) e (4.36). Esta técnica, não enquadrada na tabela 5.2, garante o equilíbrio do corpo, mas depende diretamente da discretização e da qualidade da integração da parcela complementar à região que está sendo integrada (integrações regulares e quase-singulares). Do ponto de vista numérico, isto pode aumentar também os erros de arredondamento e truncamento (Guiggiani e Gigante [1990]) e o esforço computacional é bem superior ao gasto pelo uso de quadraturas especiais. Adicionalmente, nem todos os problemas possuem uma solução particular que possa ser diretamente aplicada. Ainda assim há uma perda de generalidade um vez que tais soluções são distintas para cada classe de problemas, obrigando a implementações diferentes. Por outro lado, esta técnica não exige isolamento das singularidades, independe da dimensão do problema e é aplicável a qualquer grau de função de interpolação.

Diversos trabalhos sugerem a aplicação de quadraturas especiais para esta classe de integrais. Note-se, no entanto, que integrais fortemente singulares não podem ser descritas como somas de Reimann (Krommer & Ueberhuber 1998), e portanto tais quadraturas fornecem o valor da integral como parte finita. Uma quadratura deste tipo foi proposta por Kutt em 1975 (Kutt 1975a, Kutt 1975c, Kutt 1975b) para integrais unidimensionais de várias ordens de singularidade $1/r^\alpha$, onde $\alpha \geq 1$. Para singularidades isoladas, constitui a melhor quadratura já desenvolvida (Dumont 1992). Mesmo implicando no isolamento da singularidade, a quadratura de Kutt (válida para o intervalo $[0, +1]$) não envolve outros

cálculos e exige poucos pontos de integração¹, resultando em um tempo de computação bastante modesto em comparação com outros métodos. A inconveniência desta técnica reside no fato de uma integral por parte finita não permitir escalamento do intervalo de integração (Kutt 1975*b*). Isto significa que a aplicação da quadratura a intervalos de integração mapeados para domínios normalizados é direta apenas se o Jacobiano da transformação for constante. Caso contrário, a ordem do Jacobiano pode mascarar o real grau de singularidade do núcleo. Para domínios bi ou tridimensionais torna-se necessário escrever as integrais segundo um sistema de coordenadas polar ou esférico e aplicar a quadratura de Kutt na direção radial, motivo pelo qual é geralmente utilizada apenas para elementos de contorno bidimensionais.

Em função da demanda que o MEC causou para cálculo de integrais singulares, um número idéias errôneas também foram propostas. Um exemplo é a transformação desenvolvida por Cerrolaza & Alarcón (1989), que consiste em aplicar a transformação cúbica de Telles (1987) em cada lado do ponto singular, provocando um cancelamento dos termos ilimitados do integrando.

A título de ilustração, as figuras 5-3 e 5-4 mostram uma comparação entre alguns destes procedimentos para cálculo de integrais contendo núcleo de Cauchy.

Da figura 5-3 conclui-se que a quadratura de Kutt é realmente muito efetiva. A quadratura gaussiana padrão e a transformação de Telles obviamente falham, pois não são apropriadas para núcleos de Cauchy, e estão incluídas na figura apenas para comparação. Na figura 5-4 utilizou-se um núcleo de Cauchy e uma função densidade que tende a zero no ponto singular, o que permite uma comparação entre todas as quadraturas citadas. Entretanto, mesmo neste caso a transformação de Cerrolaza & Alarcón não forneceu o resultado esperado. Isto pode ser explicado pelo fato de uma transformação de coordenadas não ser capaz de regularizar uma integral imprópria, como ocorre com integrais fracamente singulares, ao menos para domínios de integração finitos. Huang & Cruse (1993) demonstram isto e afirmam que *"...Se existe uma transformação que permite mapear um intervalo finito em um outro intervalo finito e remover qualquer ordem de singularidade, então é possível obter um resultado numérico finito para uma integral que é divergente. Aparentemente, isto vai contra a lógica matemática..."*.

¹O número de pontos de integração necessários para o cálculo exato é dado pela mesma regra da quadratura gaussiana, $p = 2K - 1$, sendo p a ordem de uma função densidade polinomial.

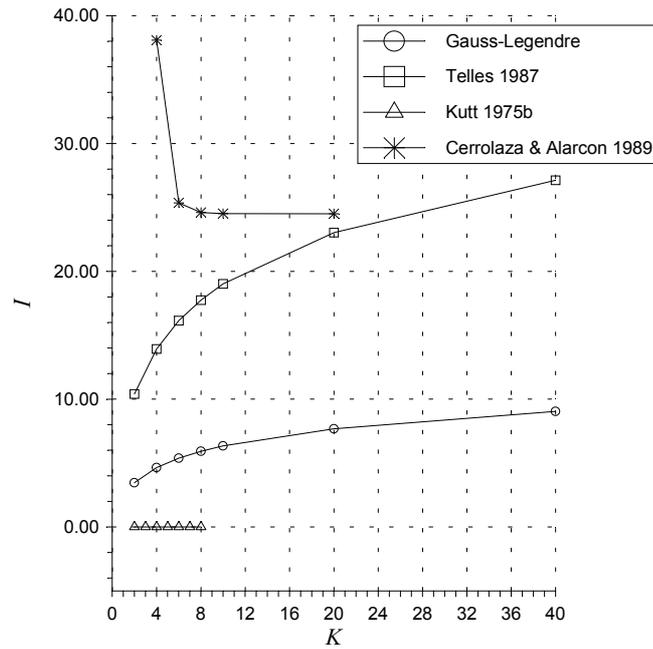


Figura 5-3: Desempenho de algumas quadraturas para integrais contendo núcleos fortemente singulares. $I = \int_{-1}^{-1} \frac{1}{|x|} dx = 0.0$.

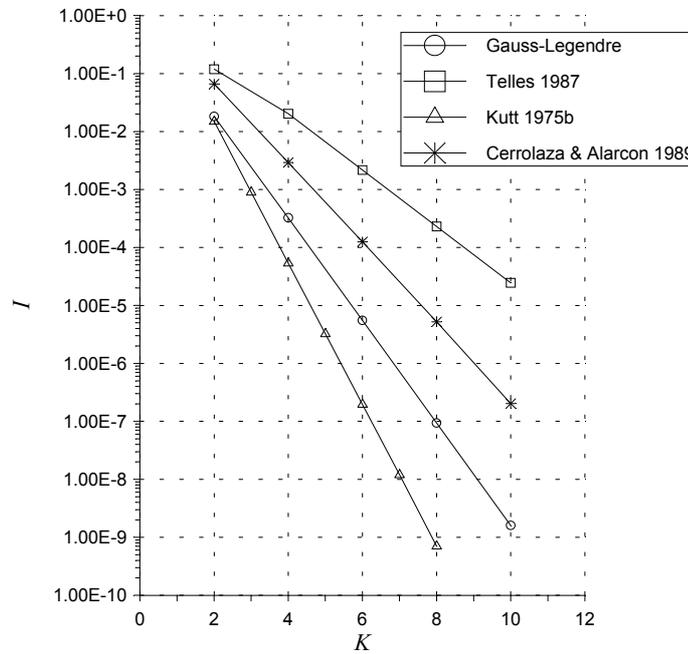


Figura 5-4: Desempenho de algumas quadraturas para integrais contendo núcleos fortemente singulares. $I = \int_{-1}^{-1} \frac{\tan x}{|x|} dx = 2.29830246$.

O fato da quadratura de Kutt não ser diretamente aplicável a domínios de integração curvos impõe uma séria restrição aos tipos de elementos/células que podem ser implementados. As técnicas enquadradas nos grupos V e VI da tabela 5.2 reportadamente fornecem as abordagens mais genéricas do ponto de vista da implementação numérica. Outras abordagens podem ser encontradas na excelente revisão de Tanaka et al. (1994). Dentre estas, o método direto parece ser um dos mais atraentes, já que formalmente não impõe nenhuma restrição ao tipo de singularidade e faz uso de pontos de Gauss-Legendre padrão (Guiggiani & Casalini 1987, Guiggiani & Gigante 1990). Por outro lado, o método requer o conhecimento de expansões assintóticas analíticas dos integrandos na vizinhança do ponto singular.

Apesar da crescente popularidade desta abordagem, poucas publicações efetivamente fornecem as expressões para as expansões assintóticas, mesmo as mais comuns como as referentes a problemas de elasticidade. Além disso, normalmente ignora-se o fato de que a dedução destas expansões analíticas fornecem uma visão sistemática e didática do comportamento do integrando singular em torno do ponto fonte. Este aspecto pode ser utilizado no estudo da real ordem de singularidade dos tensores uma vez que, conforme já salientado, a simples presença do fator $1/r$ não garante o caráter singular de um tensor.

O procedimento básico do método direto é bem documentado na literatura e não será explorado aqui em detalhes (Guiggiani & Casalini 1987, Guiggiani & Gigante 1990, Guiggiani et al. 1992, Guiggiani 1998). Apenas alguns resultados básicos que serão necessários ao longo deste capítulo serão revisados. Maiores detalhes podem ser encontrados no Apêndice C.

Seja um elemento de contorno singular Γ_n mapeado para o domínio normalizado $\xi = [-1, +1]$ e seja $\eta = \xi(p)$ a imagen do ponto fonte p sobre ξ . Admitindo-se exclusivamente o uso de elementos descontínuos tem-se $\eta \in (-1, +1)$. Com vistas à implementação no MEC, o interesse está na integração de um tensor típico sobre um elemento de contorno em suas coordenadas normalizadas :

$$\begin{aligned} I_{ij} &= \int_{\Gamma_n} K_{ij}(\xi, \eta) d\Gamma \\ &= \int_{-1}^{+1} T_{ij}[q(\xi), p(\eta)] \phi_a(\xi) J_n(\xi) d\xi \end{aligned} \quad (5.3)$$

onde ϕ_a é a função de interpolação da partição física associada ao nó a e J_n é o determinante do Jacobiano do mapeamento geométrico. Portanto, o integrando de (5.3) já leva em conta a forma interpolatória adotada para as variáveis físicas sobre o elemento Γ_n .

Um dos passos fundamentais do método direto é expandir assintoticamente o núcleo em série de Laurent em torno do ponto-fonte (Guiggiani 1998):

$$\mathbf{K}(\xi, \eta) = \frac{\mathbf{F}_{-1}(\eta)}{\rho} + \frac{\mathbf{F}_{-2}(\eta)}{\rho^2} + \dots + O(1) \quad (5.4)$$

onde $\rho = \xi - \eta$ é a imagem de r no domínio normalizado. Esta expansão separa as parcelas fortemente singular ($1/\rho$) e hipersingular ($1/\rho^2$) através das contribuições \mathbf{F}_{-1} e \mathbf{F}_{-2} respectivamente. Assim, integrandos regulares e fracamente singulares resultam em $\mathbf{F}_{-1} = \mathbf{F}_{-2} = 0$. Integrandos fortemente singulares $\mathbf{F}_{-2} = 0$ e assim por diante. Singularidades mais fortes podem ser contempladas aumentando-se o número de termos da série.

A expansão (5.4) pode ser agora subtraída e somada à integral original (5.3), sem alterar seu conteúdo. Desta maneira, a subtração provoca-se uma regularização das contribuições singulares na integral original enquanto os termos adicionados são tratados separadamente de forma analítica, mediante um procedimento de limite (Guiggiani 1998). Como a série (5.4) isola os fatores $1/\rho^\alpha$, pode-se dar a estes termos um tratamento genérico que depende apenas de α , isto é, os resultados são válidos para quaisquer \mathbf{F}_{-1} e \mathbf{F}_{-2} . Esta é a fundamental diferença em relação a outros métodos de expansão do núcleo.

A expressão final resultante para o caso de apenas um elemento de contorno conter o ponto singular é dada por (Guiggiani 1998):

$$\begin{aligned} \mathbf{I} = & \int_{-1}^{+1} \left[\mathbf{K}(\xi, \eta) - \left(\frac{\mathbf{F}_{-1}(\eta)}{\rho} + \frac{\mathbf{F}_{-2}(\eta)}{\rho^2} \right) \right] d\xi + \\ & + \mathbf{F}_{-1}(\eta) \ln \left| \frac{1-\eta}{-1-\eta} \right| + \mathbf{F}_{-2}(\eta) \left(-\frac{1}{1-\eta} + \frac{1}{-1-\eta} \right) \end{aligned} \quad (5.5)$$

Note-se que as expansões \mathbf{F}_{-1} e \mathbf{F}_{-2} utilizadas em (5.4) e (5.5) têm validade apenas em torno do ponto singular η , o que invalida, ao menos em princípio, sua utilização para integrais quase-singulares. Por outro lado, a integral (5.5) contém todas as informações relativas às funções de interpolação utilizadas pelas variáveis e pelo mapeamento geométri-

co, permitindo assim uma aplicação bastante geral. A eq.(5.5) possui uma contrapartida para o caso de elementos contínuos que não será empregada aqui (Guiggiani 1998).

A dedução da eq.(5.5) envolve um extenso tratamento analítico e um criterioso processo de limite (Guiggiani 1998), mas não possui qualquer aproximação em relação à sua forma original dada pela eq.(5.3). Como o termo integral de (5.5) contém um cancelamento de termos divergentes, ele é regular e pode ser calculado com quadratura gaussiana padrão. A solução da integral singular original (5.3) passa a ser reescrita como uma integral regular adicionada a dois termos calculados sobre o pólo η . Assim, a aplicação genérica do método direto implica tão somente no conhecimento das expansões \mathbf{F}_{-1} e \mathbf{F}_{-2} do integrando da eq.(5.3).

Dada a falta de generalidade da técnica de imposição de movimento de corpo rígido e as limitações da quadratura de Kutt, o método direto foi escolhido para integração das submatrizes de elementos de contorno singulares no presente trabalho. Com vistas à implementação numérica, estas expansões foram deduzidas para os tensores tração de elasticidade bidimensional e de flexão de placas, o que é apresentado em detalhes no Apêndice C.

O método direto pode ser igualmente empregado para integrais singulares bidimensionais. No presente trabalho isto ocorre apenas nas eqs.(4.60) e (4.61), para os casos não-lineares. Como foi empregado um esquema de diferenças finitas para cálculo do gradiente do campo de deslocamentos via eqs.(4.27) e (4.39), não se fez necessário o cômputo destas integrais.

5.1.3 Integrais hipersingulares

No âmbito das integrais com singularidades mais fortes, ou integrais no sentido do valor principal de Hadamard (Krommer & Ueberhuber 1998), geralmente se faz uso do fato da mesma se originar da diferenciação de uma integral no sentido do valor principal de Cauchy (Paget 1981, Bonnet 1997). A escolha natural seria o emprego do método direto também para este casos (Guiggiani et al. 1992, Guiggiani 1998, Frangi & Guiggiani 2000). Apesar dos bons resultados obtidos pelos procedimentos propostos na literatura, estes não serão utilizadas neste trabalho por pressupor-se apenas a utilização de elementos e células descontínuas na solução de problemas não-lineares e cálculo de gradientes do campo de

deslocamentos, onde tais integrais ocorrem. Assim, as submatrizes ${}^i\mathbf{H}$ e ${}^i\mathbf{G}$ serão, no máximo, quase-singulares. Isso é justificado pelo fato dos pontos de colocação do domínio serem os nós físicos (nós onde são calculadas as variáveis) da malha do domínio que, para células descontínuas, não coincidem com pontos de colocação sobre o contorno. De qualquer forma, deve-se evitar células de domínio cujos nós físicos estejam muito próximos do contorno, o que pode prejudicar a qualidade da integração.

5.1.4 Integrais quase-singulares

Esta situação, que rigorosamente não caracteriza uma integral singular, ocorre quando o ponto de colocação está suficientemente próximo de um elemento/célula a ponto de provocar grandes oscilações no integrando, dificultando sua integração com poucos pontos de Gauss. Atualmente há um consenso entre pesquisadores do MEC que as integrais quase-singulares merecem tanta atenção quanto as integrais singulares. Diversas estratégias têm sido propostas para solucionar esta questão, em sua maioria de natureza algorítmica. A transformação de Telles e suas variações (Telles 1987, Telles & Oliveira 1994, Johnston & Elliott 2001) é naturalmente uma boa opção se o código computacional já a utiliza para integrais fracamente singulares. Outro atrativo desta abordagem está no fato da transformação perder efeito à medida que o ponto-fonte se afasta do elemento/célula que está sendo integrado, fazendo com que a severidade da transformação se adapte à severidade do comportamento do integrando. No presente trabalho, não foi implementada nenhuma estratégia neste sentido, de modo que as integrais quase-singulares foram tratadas como integrais regulares.

5.2 Aplicação do método direto aos tensores ${}^m\mathbf{T}$ e ${}^f\mathbf{T}$

O procedimento descrito na seção anterior será aqui aplicado aos tensores tração das soluções fundamentais de elasticidade bidimensional e dos modelos de placa empregados, dados por ${}^m\mathbf{T}$ e ${}^f\mathbf{T}$, respectivamente. A dedução das expansões assintóticas \mathbf{F}_{-1} e \mathbf{F}_{-2} para estes casos envolve um certo trabalho analítico, e esse é o conteúdo do Apêndice C. Aqui serão resumidos os principais resultados e também será feita uma análise do efeito da regularização sobre os tensores singulares, de acordo com a eq.(5.5).

5.2.1 Elasticidade bidimensional

Neste caso o tensor ${}^m\mathbf{T}$ correspondente a um estado plano de deformações é dado por (Brebbia et al. 1984):

$$T_{\alpha\beta} = \frac{-1}{4\pi(1-\nu)r} \{r_{,n} [(1-2\nu)\delta_{\alpha\beta} + 2r_{,\alpha}r_{,\beta}] + (1-2\nu)(n_{\alpha}r_{,\beta} - n_{\beta}r_{,\alpha})\} \quad , \quad (5.6)$$

e as expansões assintóticas correspondentes são dados por (ver Apêndice C):

$$\mathbf{F}_{-1} = -\frac{1-2\nu}{4\pi(1-\nu)} [n_{\alpha}(p)t_{\beta}(p) - n_{\beta}(p)t_{\alpha}(p)] \phi_a(\eta) \quad (5.7a)$$

$$\mathbf{F}_{-2} = 0 \quad . \quad (5.7b)$$

Não se tem conhecimento da publicação explícita das eqs.(5.7), embora se encontrem referências à sua utilização (Guiggiani & Casalini 1987). É interessante notar que em domínios de integração retos apenas os termos fora da diagonal de \mathbf{F}_{-1} revelam singularidade forte, ao contrário do que a expressão (5.6) possa sugerir. Isto é, a simples presença do denominador r em uma solução fundamental não deve ser tomada como condição suficiente para caracterizar um comportamento singular, conforme já destacado. É necessário levar em conta o comportamento assintótico de todo o integrando, incluindo a função de interpolação e o Jacobiano do mapeamento.

5.2.2 Placas de Mindlin/Reissner

Os tensores ${}^f\mathbf{T}$ para ambos os modelos de placa abordados no presente trabalho são idênticos, sendo dados por (van der Weeën 1982a, Westphal Jr. 1990):

$$T_{ij} = \frac{-1}{4\pi r} [(4A(z) + 2zK_1(z) + 1 - \nu)(r_{,j}n_i - r_{,n}\delta_{ij}) + (4A(z) + 1 + \nu)r_{,i}n_j + \\ -2(8A(z) + 2zK_1(z) + 1 - \nu)r_{,i}r_{,j}r_{,n}] \quad . \quad (5.8)$$

As expansões assintóticas resultantes para este caso são (ver Apêndice C):

$$\mathbf{F}_{-1} = -\frac{1-\nu}{4\pi} [n_i(p)t_j(p) - n_j(p)t_i(p)] \phi_a(\eta) \quad (5.9a)$$

$$\mathbf{F}_{-2} = 0 \quad . \quad (5.9b)$$

As eqs.(5.9) parecem ser originais. Note-se a sua semelhança com as expressões (5.7). Com efeito, Guiggiani (1992) destaca que as expansões para a maioria dos problemas de elasticidade são muito parecidas. Assim como ocorreu com as expansões (5.7), as eqs.(5.9) apresentam regularidade para os termos da diagonal no caso de domínios de integração retos.

5.2.3 Alguns experimentos numéricos

As expressões (5.7) e (5.9) foram implementadas no código computacional deste trabalho juntamente com a eq.(5.5) para integração dos elementos de contorno singulares. A fim de prover uma verificação de sua eficiência, estas foram testadas para três configurações de elementos de contorno descontínuos, ilustrados na figura 5-5. Um recuo de 15% do comprimento do elemento foi utilizado para os nós das extremidades em todos os casos. Por generalidade, o pólo singular não foi posicionado sobre nenhum dos nós. Em vez disso, posicionou-se a singularidade sobre as coordenadas correspondentes ao ponto $\xi_1 = 0,3$ do domínio normalizado.

As análises se limitaram ao cálculo do termo

$$I = \int_{-1}^{+1} T_{12}\phi_1 J d\xi_1 \quad (5.10)$$

tanto para os problemas de membrana quanto para os problemas de flexão de placas. O comportamento do integrando de (5.10) pode ser verificado graficamente, permitindo uma visualização do efeito da regularização de acordo com as expressões (5.5) e (5.7) ou (5.9), para cada problema. Para efeito de comparação, serão sobrepostos o comportamento do tensor original (sem regularização) e das expansões assintóticas utilizadas em cada caso.

Para o caso de membrana (${}^mT_{12}$), a regularização pode ser visualizada nas figuras (5-6 – 5-8). Dos gráficos para os casos A e B - figuras (5-6) e (5-7) conclui-se que quando os elementos são retos o integrando regularizado de (5.5) torna-se uma função linear, permitindo o uso de um número muito pequeno de pontos de integração. A fim examinar ainda mais este aspecto, a tabela (5.4) apresenta os resultados obtidos com a integração de (5.10) utilizando (5.6) e (5.7), para diversas ordens de quadratura. Os dígitos não convergidos estão sublinhados.

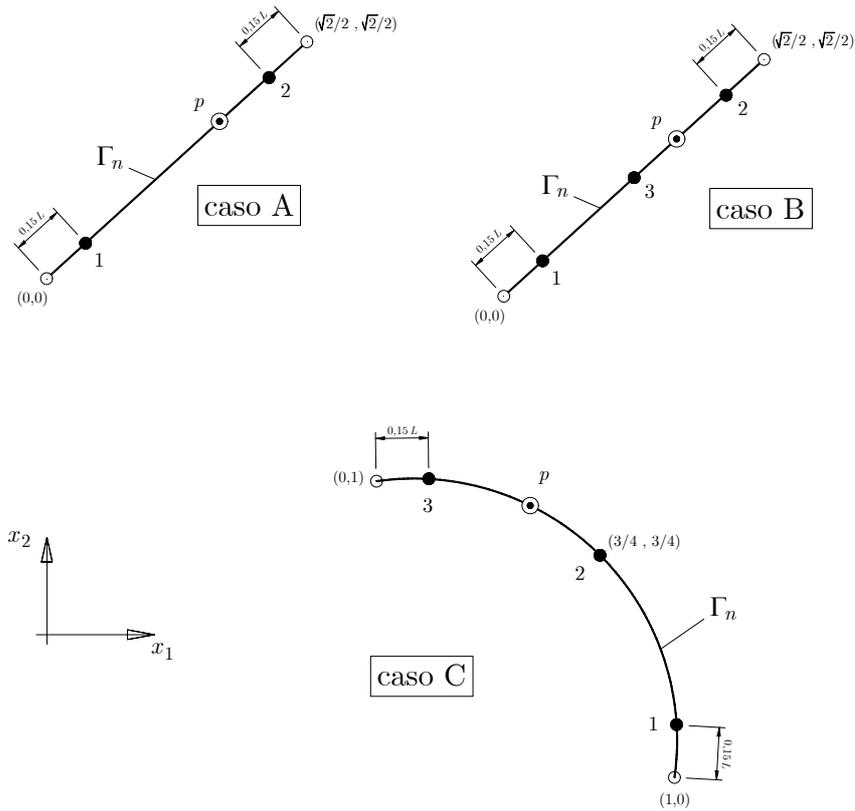


Figura 5-5: Casos analisados: caso A - elemento linear; caso B - elemento quadrático reto; caso C - elemento quadrático curvo.

A análise dos tensores de flexão de placas procedeu de forma similar, e a regularização do tensor ${}^fT_{12}$ está ilustrada nas figuras (5-9 – 5-11) para os três casos analisados. Dada a maior complexidade desta solução fundamental, os integrandos regularizados não são curvas tão simples como as do problema de membrana, mas ainda assim resultaram em funções bastante suaves.

A convergência da integração deste núcleo está ilustrada na tabela 5.5, onde os termos não-convergidos estão sublinhados. Estes resultados revelam um aspecto muito importante: a aplicação do método direto regulariza apenas as contribuições fortemente singulares. Isto torna-se evidente pela inspeção da eq.(5.4) que expande o integrando em inversos de potências inteiras de ρ , de modo que tal expansão não pode contemplar integrais fracamente singulares. Ocorre que (5.8) é composta também por diversos termos de ordem logarítmica, notadamente as funções de Bessel modificadas de segunda espécie e combinações destas, e estes termos não são afetados pela regularização. Como consequência,

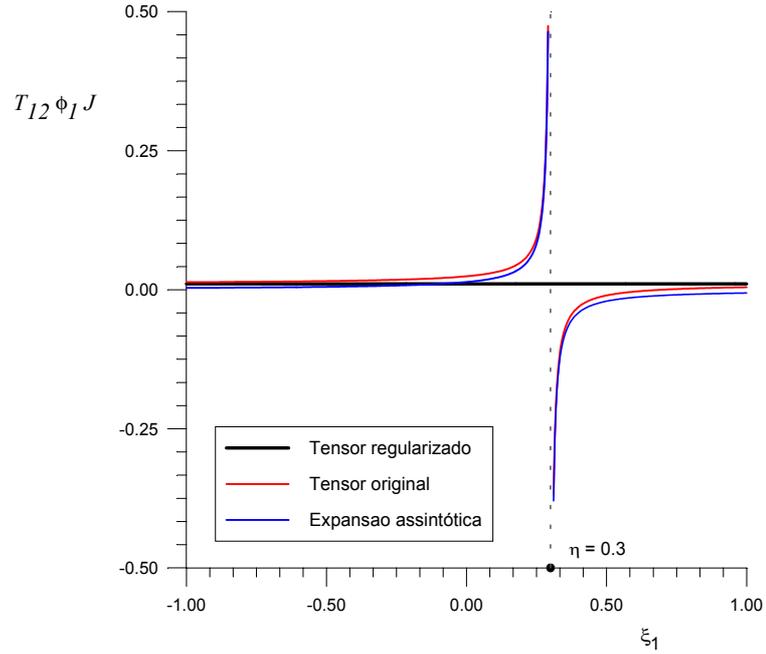


Figura 5-6: Comportamento de ${}^m T_{12} \phi_1 J$ para o caso A.

os integrandos mostrados nas figuras (5-9 – 5-11) são todos limitados, mas de difícil integração pelas regras de Gauss-Legendre padrão. Esta convergência foi facilmente acelerada aplicando-se a transformação cúbica de Telles (Telles 1987) diretamente à eq.(5.5). Os resultados assim obtidos estão ilustrados na tabela 5.6, revelando uma marcante melhora na convergência.

K	Caso A	Caso B	Caso C
2	0.023228516	0.010714395	0.060076142
4	0.023228516	0.010714395	0.052963928
6	0.023228516	0.010714395	0.052923048
8	-	-	0.052922962
10	-	-	0.052922962
12	-	-	0.052922962

Tabela 5.4: Resultados de integração numérica de (5.10) para estado plano de deformações (quadratura de Gauss-Legendre padrão).

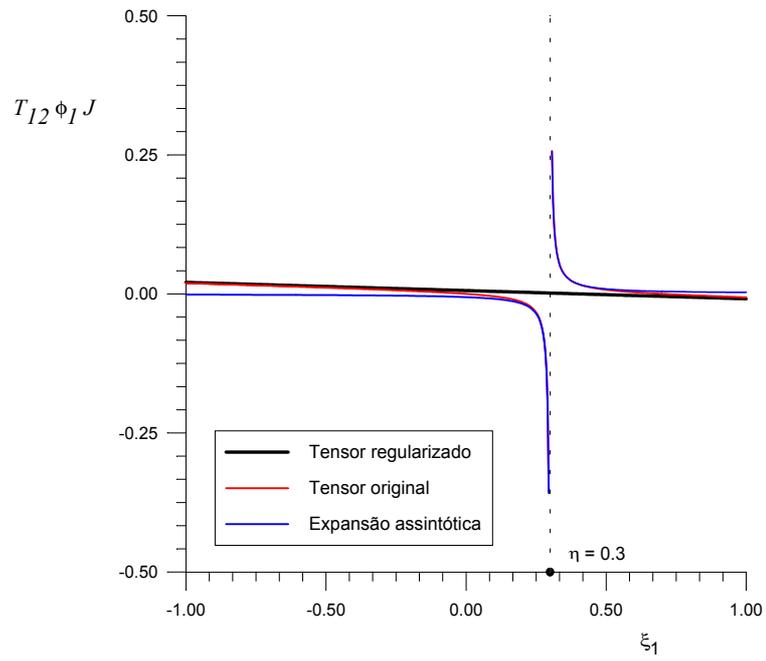


Figura 5-7: Comportamento de ${}^m T_{12}\phi_1 J$ para o caso B.

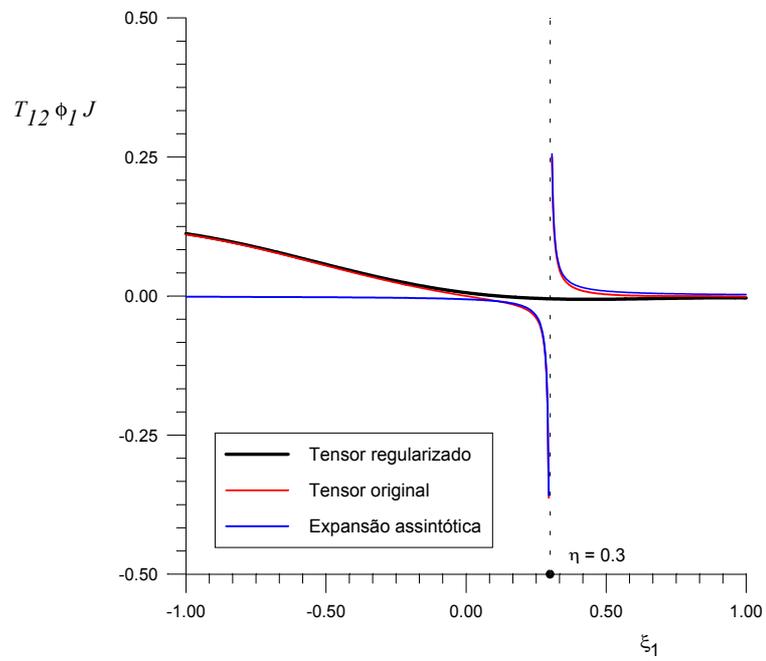


Figura 5-8: Comportamento de ${}^m T_{12}\phi_1 J$ para o caso C.

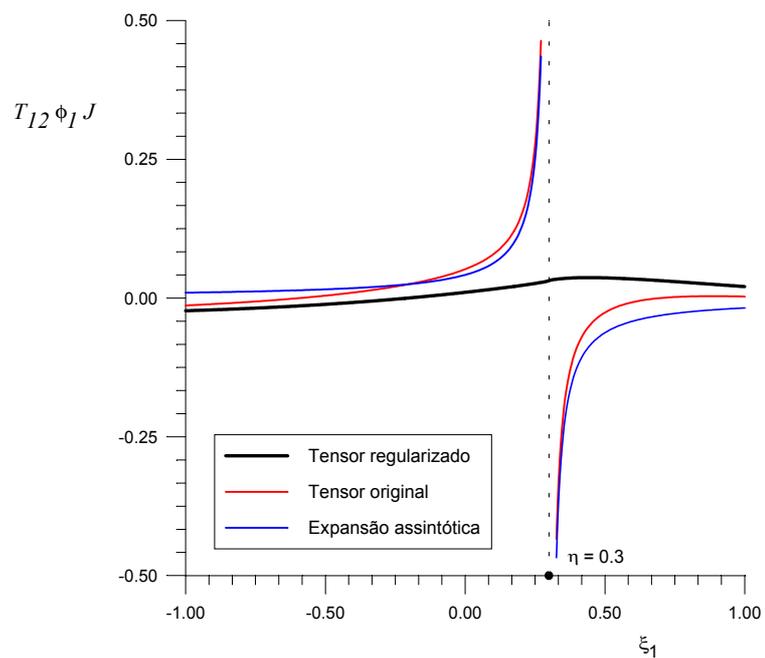


Figura 5-9: Comportamento de ${}^f T_{12} \phi_1 J$ para o caso A.

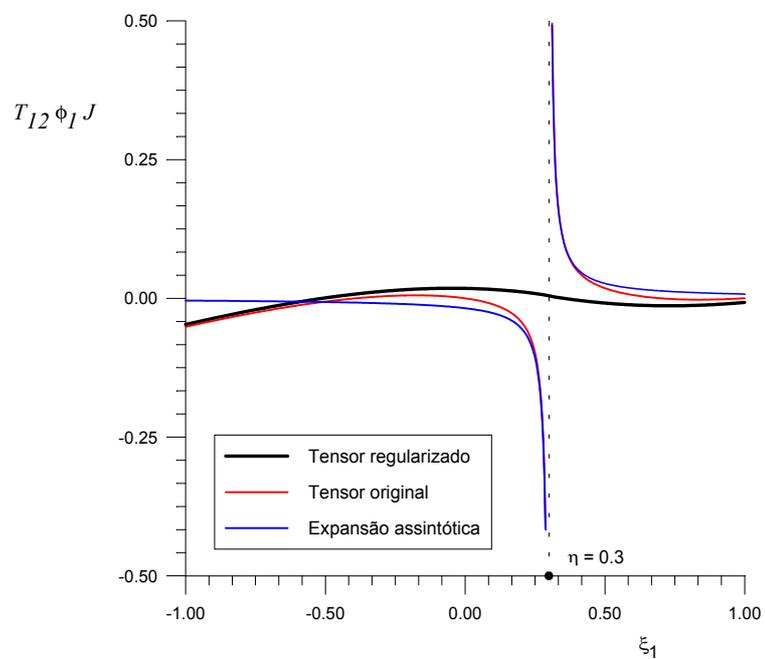


Figura 5-10: Comportamento de ${}^f T_{12} \phi_1 J$ para o caso B.

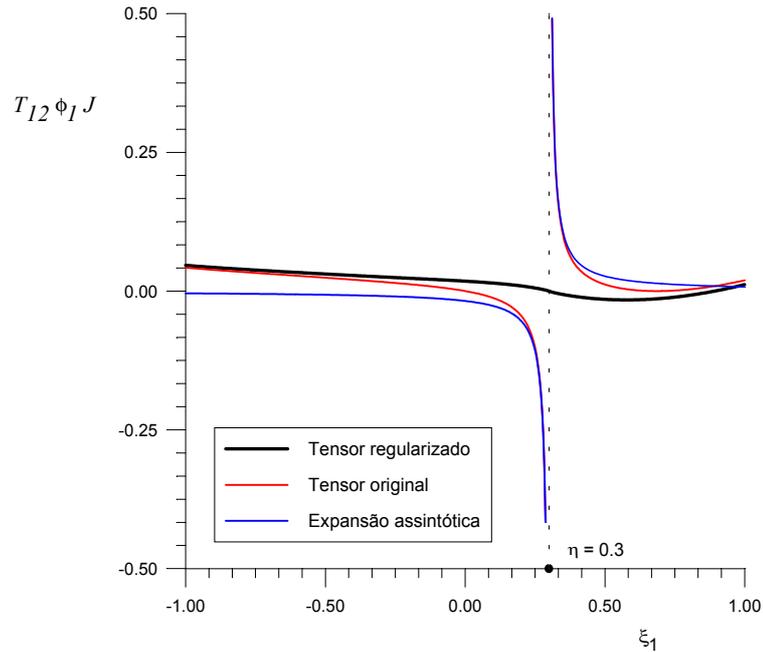


Figura 5-11: Comportamento de ${}^f T_{12} \phi_1 J$ para o caso C.

5.3 Resumo do Capítulo 5

Este capítulo abordou diversos aspectos relativos à integração numérica de integrais contendo núcleos singulares. Foram comentadas as abordagens adotadas no presente trabalho, tanto para as integrais fracamente quanto as fortemente singulares. A transformação cúbica de Telles foi escolhida para integração fracamente singular por sua generalidade e facilidade de implementação. Para as integrais fortemente singulares, o método direto foi adotado. Para tal, as expressões analíticas das expansões assintóticas dos tensores aqui empregados foram deduzidas e verificadas. Estas expressões ilustram que o real grau de singularidade mantém uma dependência das funções de interpolação adotadas para as variáveis. De modo particular, foi demonstrado que a simples presença do fator $1/r$ no integrando não é condição suficiente para garantir seu caráter singular.

A regularização produzida pelo uso das expansões (5.7) e (5.9) em conjunto com (5.5) merece destaque, pois exige apenas pontos de Gauss-Legendre para uma integração precisa. As expansões (5.7) e (5.9) podem ser utilizadas em códigos gerais do MEC independentemente da função de interpolação utilizada, e são o principal resultado deste capítulo.

K	Caso A	Caso B	Caso C
2	0.028879841	-0.019775277	0.013603463
4	0.028400492	-0.011111221	0.023886107
6	0.025048353	-0.009616529	0.026522157
8	0.025218977	-0.009914611	0.025815326
10	0.025631058	-0.010133399	0.025350626
12	0.025880005	-0.010215774	0.025184356
14	0.025892861	-0.010191578	0.025243153
16	0.025575045	-0.010054894	0.025538264
18	0.025590601	-0.010074522	0.025491747
20	0.025673541	-0.010114558	0.025404066
24	0.025760604	-0.010144374	0.025341528
32	0.025711883	-0.010126553	0.025379346
40	0.025681169	-0.010112925	0.025409089
48	0.025673388	-0.010108823	0.025418240
64	0.025699043	-0.010119971	0.025393922

Tabela 5.5: Resultados da integração numérica de (5.10) para flexão de placas (quadratura de Gauss-Legendre padrão).

K	Caso A	Caso B	Caso C
2	0.045054311	0.018484718	0.014087607
4	0.025175066	-0.010942197	0.022917282
6	0.025759213	-0.010075284	0.025307587
8	0.025699404	-0.010122893	0.025400520
10	0.025692084	-0.010117071	0.025400500
12	0.025690580	-0.010116337	0.025401766
14	0.025690194	-0.010116156	0.025402245
16	0.025690103	-0.010116111	0.025402338
18	0.025690092	-0.010116105	0.025402352
20	0.025690103	-0.010116109	0.025402343
24	0.025690122	-0.010116102	0.025402324
32	0.025690135	-0.010116128	0.025402313
40	0.025690135	-0.010116122	0.025402313
48	0.025690134	-0.010116122	0.025402314
64	0.025690135	-0.010116123	0.025402313

Tabela 5.6: Resultados da integração numérica de (5.10) para flexão de placas (quadratura de Gauss-Legendre com transformação cúbica).

Capítulo 6

Proposta de uma Arquitetura Orientada a Objetos para o MEC

O objetivo deste capítulo é discutir alguns tópicos básicos sobre a aplicação de técnicas de programação orientada a objetos (POO)¹ em programas de elementos finitos ou elementos de contorno, a fim de fornecer subsídios para o desenvolvimento da biblioteca de classes utilizada na implementação computacional deste trabalho. Inicialmente, são feitos alguns comentários sobre as vantagens do uso de OO para métodos numéricos em engenharia. Identifica-se então algumas características que são encontradas em muitos dos programas OO existentes para o MEF e o MEC. Estas características são utilizadas como base na análise de alguns trabalhos relevantes extraídos da literatura, no âmbito da Mecânica dos Sólidos. Desta análise resulta uma proposta de biblioteca de classes especialmente desenvolvida para uso em programas de elementos de contorno, denominada McBEM. Na ilustração de trechos de programas apresentados ao longo deste capítulo será utilizada a linguagem C++ padrão.

6.1 Conceitos de programação orientada a objetos

As primeiras linguagens de programação orientadas a objeto disponíveis comercialmente surgiram em meados dos anos 80 (Stroustrup 1986) e seu uso na Engenharia ganhou

¹Aqui serão utilizadas as siglas POO para programação orientada a objeto e OO para orientação a objeto ou orientado(a) a objeto.

impulso no início da década de 90. Embora o uso de encapsulamento de dados já estivesse bastante difundido por linguagens como o C e o Pascal, apenas a partir da normalização de linguagens orientadas a objeto em 1990 e 1991 os fabricantes de compiladores passaram a oferecer pacotes com a confiabilidade necessária para uso difundido de programação com encapsulamento de dados e métodos. A partir de então, é notória a popularização mundial destas linguagens nas mais diversas áreas de aplicação.

Existem atualmente várias implementações de linguagens OO, interpretadas e compiladas. Dentre as mais famosas, pode-se encontrar APL, Ada, Clu, C++, CLOS, Object Pascal, Java, SmallTalk e certas versões do Fortran. Na realidade, apenas algumas destas podem ser consideradas efetivamente linguagens OO dado que diversas delas não suportam construções hoje normalizadas (Stroustrup 1991).

A difusão do uso de linguagens OO se deve a uma série de fatores reconhecidos. Os mais claros estão diretamente relacionados ao ciclo de desenvolvimento de programas e seu custo. A constante atualização de códigos utilizados em pesquisa ou comercialmente - a fim de torná-los mais abrangentes e eficazes - atinge atualmente limites práticos muitas vezes intransponíveis. Em alguns casos, trata-se de programas iniciados a mais de trinta anos, desenvolvidos em linguagens já consideradas obsoletas e contendo tipicamente dezenas ou mesmo centenas de milhares de linhas de programação estruturada. Isto criou uma demanda de *extensibilidade* e *reusabilidade* dos programas (ou parte deles) sem incorrer nos custos de desenvolver programas totalmente novos ou alterar trechos de programas já suficientemente testados e homologados. Esta é uma necessidade relativamente antiga, mas apenas o surgimento da POO propiciou uma solução adequada. Com efeito, esta demanda está intimamente relacionada com as próprias origens de muitas linguagens OO.

Uma descrição detalhada do paradigma de OO foge ao objetivo deste texto. Serão abordados aqui apenas aquelas propriedades que caracterizam as linguagens OO de forma comum, a fim de permitir uma compreensão mais clara do restante deste capítulo. Pressupõe-se o conhecimento da sintaxe padrão da linguagem C/C++;

6.1.1 Classes e objetos

De uma maneira resumida, a POO se distingue das formas usuais de programação estruturada por lidar com *objetos*. Objetos são entidades auto-contidas compostas por *dados*

(ou *membros* ou *atributos*) e *métodos* (ou *funções-membro*). Um objeto é criado por uma *classe* (ou: a ocorrência de uma classe é chamada objeto). Por exemplo, seja uma classe `NoGeraL`, responsável pela criação e manipulação de nós de elementos finitos. Um objeto nó chamado `meuNo` (criado pela classe `NoGeraL`) pode conter as suas coordenadas `coordX` e `coordY` como dados e ter como métodos funções para ler as coordenadas de um arquivo e transladar estas coordenadas sobre os eixos x e y , chamadas `LeiaNo`, `TranslEmX` e `TranslEmY`. Assim, as variáveis (dados) de qualquer nó e as rotinas (métodos) que as manipulam estão agrupadas logicamente. O objeto se encarrega de manipular os seus dados através destes métodos. Isto é realizado através de *mensagens*. Então o usuário realiza a leitura das coordenadas de um nó invocando `meuNo.LeiaNo(cx,cy)` e translada suas coordenadas com a instrução `meuNo.TranslEmX(dX)`. A diferença em relação às linguagens tradicionais é que o usuário não tem acesso aos dados do objeto (a menos que se deseje), apenas o próprio objeto o faz. Quando se translada o nó na direção x , apenas o objeto `meuNo` sabe como fazê-lo. Uma instrução do programa como `coordX = 5.0` não é permitida. Pode-se fazer o mesmo para aqueles métodos aos quais não se deseja que o usuário tenha acesso. Assim define-se o que se chama de *interface* da classe - aquelas partes (dados ou métodos) visíveis para o usuário. As demais partes ficam *escondidas*. Isto permite que o armazenamento dos dados, detalhes da implementação dos métodos etc. não sejam acessíveis para o programa que utiliza este objeto. Os dados/métodos visíveis são chamados de *públicos*, enquanto os demais são chamados *privados*².

Em outras palavras, um objeto é um novo tipo. Um paralelo didático pode ser observado com um número inteiro `j` em um programa qualquer. Poder-se-ia dizer que o objeto `j` é uma ocorrência da classe `int` (tipo inteiro), e que dispõe dos métodos `+`, `-`, `*` e `/` (adição, subtração, multiplicação e divisão). Mas não se pode manipular a forma como este número é armazenado na memória, por exemplo, pois esta seria uma parte privada da classe `int`.

²Algumas linguagens como o C++ oferecem ainda mais um especificador de acesso, denominado *protegido*, cujo uso está relacionado ao conceito de herança.

6.1.2 Encapsulamento de dados e métodos

Este processo de relacionar um ente a um objeto, selecionar quais são seus dados relevantes, quais métodos são necessários, o que deve ser público e o que deve ser privado, é chamado de *encapsulamento*. Ou seja, os objetos encapsulam seus dados e métodos, e estes podem ser manipulados apenas pelo próprio objeto. Em resumo, o encapsulamento de dados e métodos é definido por uma classe, cuja ocorrência define um objeto dentro do programa. Isto leva ao conceito de tipos de dados abstratos. Neste contexto, as declarações

```
NoGeral meuNo;  
int j;
```

são similares. A diferença é que o objeto `meuNo` é um novo tipo, definido pela classe `NoGeral`, enquanto `int` é um tipo nativo do compilador. Torna-se óbvia a possibilidade que se abre de implementar tipos abstratos não suportados pelo compilador.

A eficiência e robustez de uma classe está diretamente relacionada à forma como seus dados e métodos estão encapsulados. Se a classe `NoGeral` do exemplo anterior é bem projetada, ela pode ser reaproveitada em diversos programas. Se sua interface for eficiente, ela proverá todos os métodos relevantes para se manipular nós de malhas de elementos finitos. Deve-se notar, entretanto, que até este ponto não há grandes vantagens em relação à programação estruturada usual. O que efetivamente caracteriza uma linguagem OO são as possibilidades de *herança* e *polimorfismo*.

6.1.3 Herança e polimorfismo

Em linguagens OO, a herança é caracterizada pela possibilidade de toda classe ser utilizada para construir classes descendentes que têm acesso aos dados e métodos da(s) classe(s) que a originou(aram). Ou seja, uma *classe derivada* (ou *subclasse*) herda dados e métodos de sua *classe-mãe* (ou *superclasse*). Durante a execução, quando uma classe derivada recebe uma mensagem invocando um método, verifica-se se este método existe na própria definição da classe derivada. Caso não exista, o método é procurado na classe antecessora

(classe-mãe). Se por sua vez esta também for uma classe derivada, a procura continua pelos ancestrais até o método ser encontrado.

A herança permite a construção de uma *hierarquia* entre os objetos de um programa. Mais do que isso, facilita o agrupamento de dados e/ou métodos comuns a famílias de classes nas classes-mãe, enquanto os dados e métodos particulares das classes derivadas são implementados em níveis hierárquicos mais baixos (classes derivadas)

Já o polimorfismo se refere à possibilidade de uma mesma mensagem ativar diferentes métodos. Um método é sempre identificado por três características: seu nome, seus parâmetros e o objeto ao qual pertence. Portanto, é possível definir dois ou mais métodos com o mesmo nome desde que pertençam a objetos diferentes ou tenham uma lista de parâmetros diferentes. Seja por exemplo uma classe-mãe `ElementoFinito`, da qual derivam duas classes: `DKT` e `QUAD4`. Ambas possuem um método para cálculo das matrizes de rigidez correspondentes, denominado `calcMatRig` em ambas. Então, se uma malha possui objetos `DKT` e `QUAD4`, como no trecho a seguir:

```
DKT elem1;
QUAD4 elem2;
...
K1 = elem1.calcMatRig();
K2 = elem2.calcMatRig();
```

a mensagem `calcMatRig` ativa o método correto, isto é, calcula a matriz de rigidez do elemento `DKT` ou do elemento `QUAD4`, dependendo do objeto, embora o nome do método seja o mesmo em ambas. Note-se que o uso do polimorfismo adiciona clareza ao programa, uma vez que métodos que executam tarefas similares são chamados da mesma maneira. Por exemplo, seja uma classe `No` que permite a aplicação de dois tipos de condições de contorno: temperatura ou fluxo prescritos, como no segmento de programa a seguir:

```
class No {
```

```
    aplicarCondContorno(Temp x);
    aplicarCondContorno(Fluxo x);
    ...
}
```

Agora a aplicação das condições de contorno se torna muito mais simples:

```
No no1;
No no2;
Temp t_dado;
Fluxo q_dado;
...
no1.aplicarCondContorno(t_dado);
no2.aplicarCondContorno(q_dado);
```

Existe ainda um outro tipo de polimorfismo, relacionado à sobrecarga de operadores lógicos, mas não será tratado aqui devido à extensão do assunto. Maiores detalhes podem ser encontrados na literatura (Stroustrup 1986, Stroustrup 1991, Brokken & Kubat 1998).

Estes exemplos são simples e não permite uma compreensão completa das potencialidades do polimorfismo, mas são suficientes para os objetivos deste texto. Deve-se destacar ainda que a escolha do método a ser utilizado não ocorre em tempo de compilação, mas em tempo de execução. O mecanismo que permite isto é denominado *dynamic binding* (Hekmatpour 1990, *Borland C++ Programmer's Guide - Version 4.5* 1994, Brokken & Kubat 1998), e este mecanismo é uma das características que diferencia as linguagens OO das linguagens que apenas permitem encapsulamento de dados/métodos.

São justamente as propriedade de herança e polimorfismo que facilitam enormemente a extensão dos códigos. Dado um conjunto de classes, pode-se derivar outras classes destas sem alterar as partes já desenvolvidas e testadas. Exemplos interessantes são a inclusão de novos tipos de elementos, de materiais ou novos tipos de análise em em programa de elementos finitos. Já a reusabilidade está relacionada com o caráter auto-suficiente dos

objetos. A reutilização de uma classe não exige conhecimentos da sua implementação - apenas se extrai as informações necessárias disponibilizadas através da sua interface. Em um caso extremo, um programa OO completo pode ser convertido em uma classe de um outro programa maior.

Neste trabalho, a hierarquia e o relacionamento entre classes seguirá a notação de Rumbaugh et al. (1991), conforme ilustrado na figura 6-1. O relacionamento entre classes se dá através de três formas básicas:

- A herança caracteriza uma relação do tipo *é-um* (*is-a*). Pode ser uma herança simples, onde uma superclasse origina outras subclasses (figuras 6-1.a e 6-1.b) ou herança múltipla, onde diversas superclasses originam uma subclasse (figura 6-1.c). Ou seja, um objeto derivado é também um objeto da classe-mãe, pois herda suas propriedades.
- Uma relação do tipo *conhece-um* (*knows-a*) é utilizada quando um objeto tem acesso a outro (figura 6-1.d). Se esse acesso ocorrer para diversos objetos similares, denomina-se relação *conhece-um* múltipla (figura 6-1.e).
- Quando um objeto é formado por diversos outros objetos, ocorre o que se denomina agregação, ou relação *tem-um* (*has-a*), e a notação é a indicada na figura 6-1.f .

6.2 Características comuns de arquiteturas OO para elementos finitos e elementos de contorno

Não há dúvida que existem muitas possibilidades de se projetar uma arquitetura OO para programas de elementos finitos ou elementos de contorno. A breve revisão bibliográfica a seguir ilustra muitas das abordagens possíveis. A escolha de uma ou outra abordagem passa necessariamente pela aplicação que se deseja para o programa, seja pesquisa, uso prático em engenharia ou educação. Qualquer que seja o caso, a parte essencial de um programa de elementos finitos (ou elementos de contorno) são os algoritmos de solução do problema discretizado. É este aspecto que determina a eficiência, a robustez e a estabilidade da solução obtida.

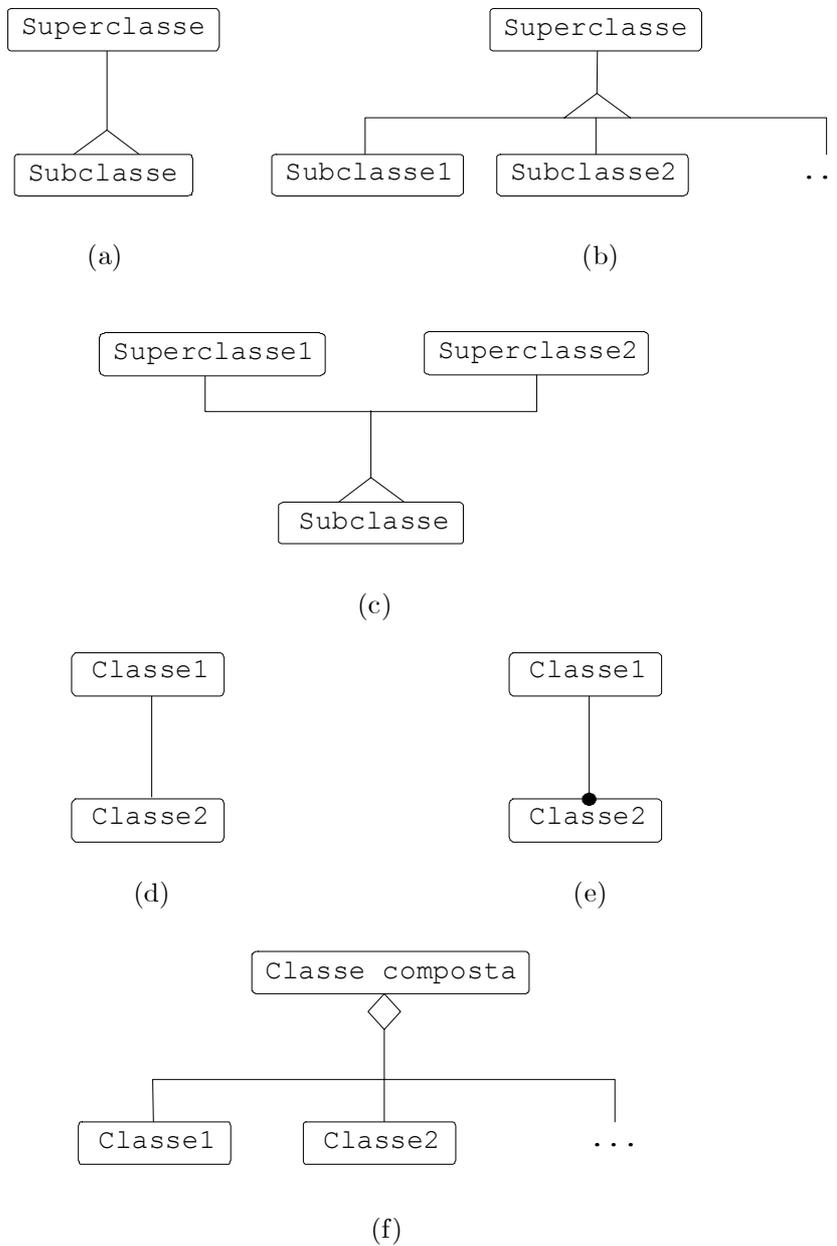


Figura 6-1: Notação de Rumbaugh: (a) herança simples (relação *é-um*) originando uma classe derivada; (b) herança simples originando diversas classes derivadas; (c) herança múltipla; (d) associação *conhece-um*; (e) associação *conhece-um* múltipla; (f) agregação de classes (relação *tem-um*).

Com a finalidade de discernir as diversas abordagens OO encontradas na literatura, bem como limitar significativamente esta revisão, identifica-se três grandes níveis de abstração encontrados em programas de elementos finitos e elementos de contorno. São eles:

- **Baixo nível:** Este nível de abstração lida principalmente com *métodos*³. Geralmente são identificados como algoritmos que manipulam entes básicos da álgebra linear, como tensores, matrizes ou vetores. Exemplos típicos são a solução de um sistema linear ou de um problema de autovalores/autovetores. A representação interna destes entes também é controlada aqui, ou seja, diferentes métodos são empregados para matrizes simétricas e não-simétricas, em banda ou esparsas, por exemplo. Embora matrizes sejam o caso mais comum, outras entidades também são tratadas neste nível como: palavras (*strings*), números complexos, funções de interpolação, integração numérica, gerenciamento de memória, armazenamento de variáveis etc.
- **Médio nível:** Aqui se manipula aqueles entes intermediários, ou seja, que geram e/ou armazenam informações imprescindíveis para solução do problema. São estas informações que relacionam diretamente o problema matemático com o problema físico real. No contexto do MEF, por exemplo, é neste nível que se encontram os nós (pontos no espaço), elementos (geometria), carregamentos, materiais, condições de contorno, superelementos etc.
- **Alto nível:** Referem-se basicamente à estratégia de solução do problema. É neste nível que o problema é solucionado do ponto de vista da equação governante. Aqui se utiliza e se controla as informações geradas nos dois níveis de abstração anteriores para se obter a solução global de um problema descrito por uma malha e um dado material, por exemplo. Esta mesma malha e material pode ser utilizada para se resolver um problema estático não-linear ou um problema dinâmico transiente linear. Em essência, portanto, os dois problemas diferem da forma como as rotinas de alto nível manipularão as informações geradas pelas rotinas de médio nível, utilizando as ferramentas providas pelas rotinas de baixo nível. Em outras palavras, a abstração de alto nível é responsável pelo fluxo das informações durante a solução do problema.

³Neste contexto, métodos se referem a operações matemáticas, gerenciamento de memória etc. e não deve ser confundido com funções-membro de uma classe.

Nível de abstração	Denominação
Baixo	Classes auxiliares
Médio	Classes do modelo
Alto	Classes de análise

Tabela 6.1: Associação genérica entre classes e níveis de abstração.

A POO é naturalmente atrativa para organizar a solução de um problema em níveis hierárquicos através de derivação de classes. Dado que as diversas arquiteturas encontradas para programas de engenharia se utilizam de alguma forma de pelo menos um dos três níveis discutidos acima, estes serão aqui utilizados para enquadrar qualitativamente em que nível de abstração se encontra uma determinada classe. Com o objetivo de facilitar este enquadramento a revisão bibliográfica apresentada a seguir faz a associação entre classes e um dos três níveis de abstração de acordo com a denominação dada na tabela 6.1.

Esta nomenclatura foi escolhida meramente para facilitar a identificação da tarefa básica de uma classe neste trabalho. Classes auxiliares então se referem a classes para matrizes, solução de sistemas, listas de armazenamento etc. As classes do modelo lidam com o modelo computacional de uma forma geral (malha, materiais, condições de contorno, detalhes de geometria etc.), enquanto as classes de análise constituem o algoritmo de solução do problema, gerenciamento dos passos de carregamento em processos incrementais, integração no tempo, dentre outros. A classificação da tabela 6.1 não significa, por exemplo, que uma classe de matrizes efetivamente se reveste de um nível de abstração baixo. No presente contexto, chamá-la de classe de baixo nível significa apenas que não existem muito mais tarefas que uma classe deste tipo esteja encarregada de realizar, além de operações matriciais. Um classe de análise, por sua vez, pode abranger deste uma simples análise linear até um problema dinâmico de iteração fluido-estrutura.

6.3 Algumas referências relevantes

O trabalho de Forde et al. (1990) é citado como a primeira abordagem do uso de POO em programas de elementos finitos. Basicamente, são implementadas apenas classes de modelo como nó, elemento, material, funções de interpolação, condições de contorno de deslocamento e pontos de integração. Estas classes básicas são integradas através de

listas. Entretanto, o programa mescla partes em linguagem C e partes em linguagem Object Pascal, o que não permite o uso apropriado de herança e polimorfismo. Este trabalho foi implementado em linguagem C++ por Scholz (1992). Os objetos constituintes do modelo são armazenadas através de uma classe genérica denominada `Tree`. Esta classe manipula objetos tipo `void`, de modo que listas de nós, elementos, materiais etc. podem ser manipuladas pela mesma classe. Entretanto, nenhuma classe de análise é implementada, deixando para o usuário o desenvolvimento dos algoritmos de solução.

Diversos artigos discutem de forma superficial as vantagens do uso de POO em programas de elementos finitos. Mackie (1992) apresenta uma tentativa de transformar um programa baseado em subrotinas para POO, utilizando linguagem Object Pascal. Miller (Miller 1991, Miller 1993) apresenta uma discussão similar, mas salientando aspectos como herança, reusabilidade e extensibilidade de programas. Novamente, não são implementadas classes de análise. O desenvolvimento é baseado em geometrias independentes do sistema de coordenadas e voltado para algoritmos de solução do tipo elemento-por-elemento.

Feijóo et al. (1991) implementaram classes básicas para o modelo como: coordenadas, nós, elementos, material, renumeração nodal, numeração dos graus de liberdade etc. Não foram desenvolvidas classes específicas para análise.

Pidaparti & Hudli (1993) apresentam uma proposta simples para análise dinâmica modal e transiente. O modelo é implementado através das classes de médio nível `Node`, `DisplacementBoundaryConditions`, `IsotropicMaterial`, `CompositeMaterial`, `ShapeFunction` e `Element`. Para a análise, uma classe-mãe `DynamicAnalysis` é usada como base para derivação de duas classes abstratas: `EigenSolution` e `DirectIntegration`, das quais derivam classes que fornecem diversos métodos de solução (figura 6-2).

Zimmermann e seus colaboradores (Zimmermann et al. 1992, Dubois-Pèlerin et al. 1992, Dubois-Pèlerin & Zimmermann 1993) desenvolveram e implementaram uma arquitetura OO bastante completa e eficiente para análise por elemento finitos. Embora não contenha um nível de abstração muito elevado, a estrutura de classes deste trabalho reflete visivelmente a experiência prática dos autores com o MEF. A arquitetura apresenta dois níveis básicos de programação: a classe `Domain` e a classe `FemComponent`. A classe `Domain` é responsável pelo gerenciamento do modelo de elementos finitos e da solução das

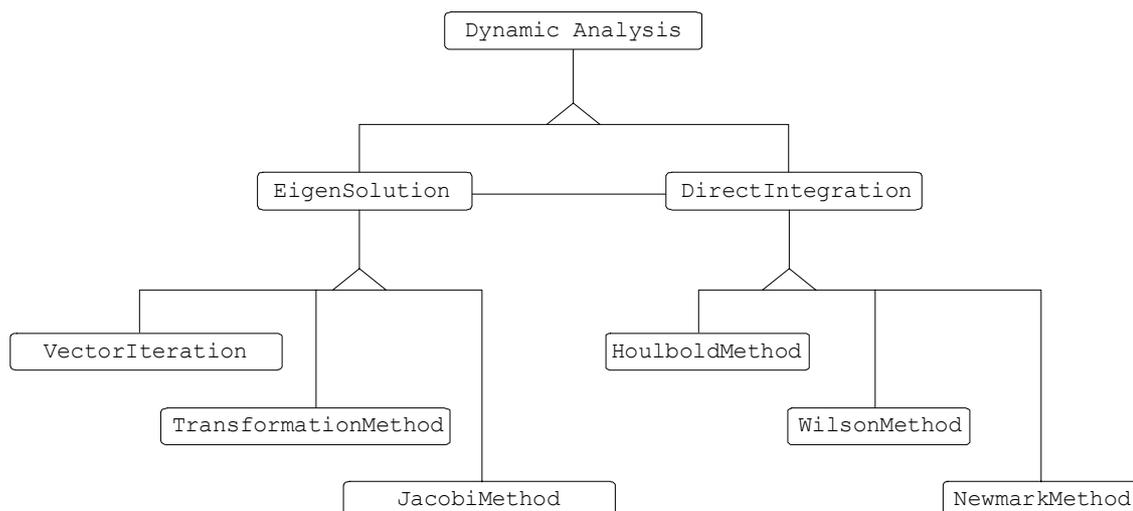


Figura 6-2: Hierarquia de classes de análise dinâmica de Pidaparti & Hudli (1993).

equações (análise), enquanto da classe `FemComponent` são derivadas diversas sub-classes (modelo) : `Element`, `Node`, `Material`, `Load` etc. Desta estrutura básica derivam diversas outras classes (Commend & Zimmermann 2001). A figura 6-3 ilustra a hierarquia básica.

Uma característica interessante da abordagem de Zimmermann et al. é o uso do *princípio da não-antecipação*, segundo o qual não se assume *a priori* a existência de qualquer entidade, sejam nós, elementos, carregamentos ou condições de contorno. As próprias classes se encarregam de procurar os dados necessários para construir seus objetos e, se existirem, o que fazer com estes objetos após sua criação. Muito embora avançada, a idéia cria situações um tanto quanto absurdas como o seguinte programa de elementos finitos:

```

main {
    Domain Meu_Problema;
    Meu_Problema.solveYourself();
}
  
```

Desta forma não resta muito para o usuário fazer caso se deseje utilizar esta arquitetura na solução de um problema não contemplado pelas classes já implementadas. Embora já

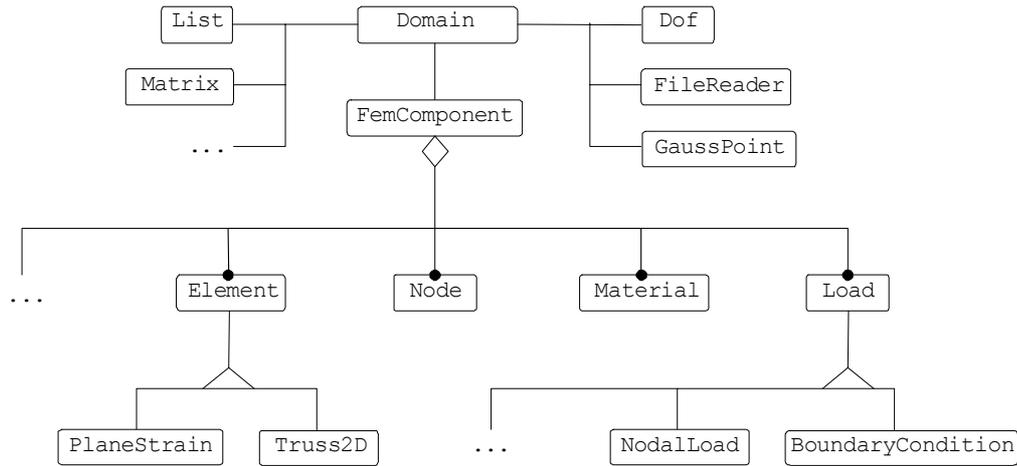


Figura 6-3: Hierarquia de classes de Dubois-Pèlerin and Zimmermann (1993).

esteja mostrando sinais de obsolescência, esta arquitetura ainda é considerada uma das mais avançadas abordagens de POO para elementos finitos.

Hedegal (1994) propôs uma estruturação excepcionalmente simples e versátil para implementação de elementos finitos para análise linear e não-linear de problemas de potencial e elasticidade em duas ou três dimensões, chamada **ObjectFEM**. Este trabalho é atrativo pela sua simplicidade e faz uso intensivo de métodos virtuais a fim de permitir a extensão das classes apresentadas. Basicamente, as classes de modelo são implementadas através de nós, elementos, material e propriedades (**Node**, **Element**, **Material** e **Property** - ver figura 6-4). Os nós e elementos se comunicam, mas apenas os elementos têm acesso aos objetos material e propriedade. As classes auxiliares contam com classes de armazenamento (classe **List**) baseadas em listas ligadas e também com classes para manipulação de entidades algébricas (classes **Matrix** e **Vector**). O uso de herança e polimorfismo permite que o usuário regule a profundidade da hierarquia das classes principais (figura 6-5). Esta arquitetura conta ainda com classes especiais como a classe **GaussPoint**, que além das funções usuais relacionadas à integração numérica pode armazenar tensores constitutivos, deformações plásticas, tensões e outras variáveis de interesse nos pontos de integração, auxiliando na solução de problemas não-lineares. A classe **Element** realiza as tarefas de montagem das matrizes relevantes, incluindo as necessárias para implementação de formulações Lagrangeanas totais ou atualizadas. Exemplos de não-linearidade geométrica e material são mostrados para elementos de barra e de elasticidade bidimen-

sional (Hedegal 1994). Não existem classes de análise. Estas são sugeridas em trechos de código como parte do programa desenvolvido pelo usuário e incluem geração de malha, solução de sistemas e pós-processamento (figura 6-4). No entanto, são ótimas candidatas a classes de análise.

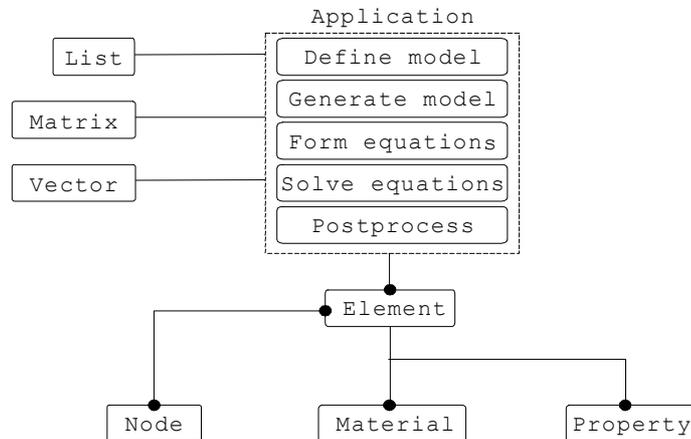


Figura 6-4: Hierarquia das classes de maior nível de Hedegal (1994). O bloco **Application** se refere à parte implementada pelo usuário.

Zeglinski et al. (1994) implementaram classes auxiliares para manipulação de matrizes que ocorrem tipicamente em elementos finitos. Este trabalho sugere que a inclusão de novos elementos finitos em um programa OO deva fazer uso de derivação simples de classes básicas, a fim de permitir maior flexibilidade do programa na expansão da biblioteca de elementos. A hierarquia das classes é bem rasa, e não são implementadas classes de análise. Um exemplo de implementação em C++ é apresentado e este é injustamente comparado com o equivalente em Fortran. Os trabalhos de Kong e colaboradores (Kong & Chen 1995, Kong 1996) mostram uma implementação mais geral, voltada para a manipulação dos diferentes tipos de entidades encontradas em programas de elementos finitos. Um dos trabalhos ilustra como proceder para prover o compartilhamento de variáveis em um programa em C++ (Kong 1996), similar ao comando `common` do Fortran, o que vai contra as filosofias mais modernas de POO. Por outro lado, o trabalho constitui uma das primeiras tentativas de se utilizar classes de análise para manipulação de eventos genéricos (ler arquivo, resolver sistema linear etc.), além das classes usuais para o modelo. É oportuno notar que este trabalho implementa uma hierarquia interessante para os elementos finitos, através de uma classe-mãe denominada `BlocElem`. Duas classes derivam de `BlockElem`:

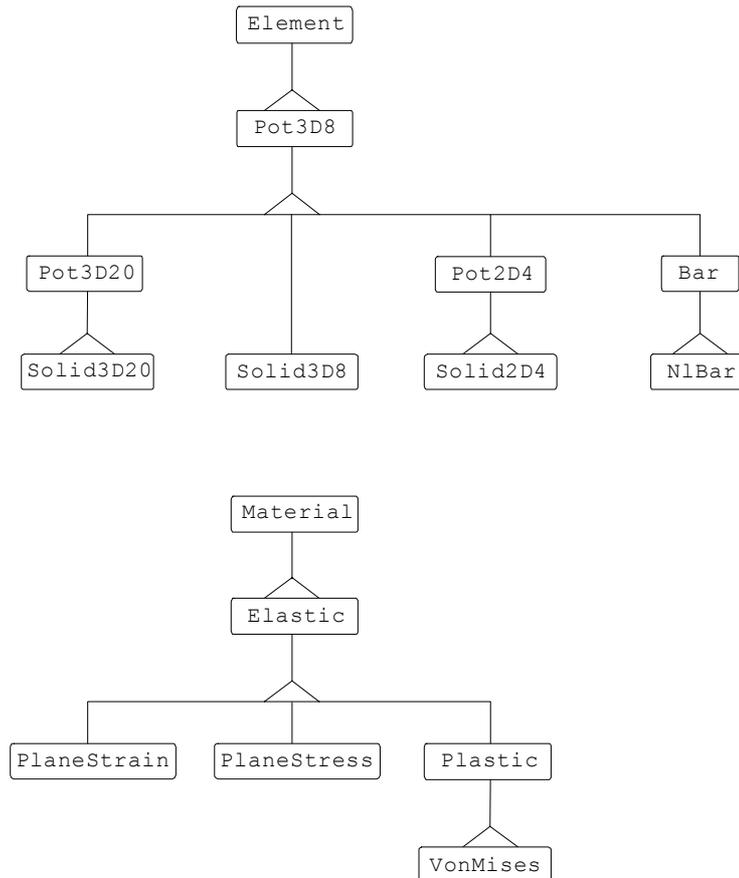


Figura 6-5: Exemplos da hierarquia de classes proposta por Hedegal (1994).

uma relativa à topologia do elemento (classe **Geometry**) e outra relativa ao tipo de equação governante do problema (classe **Physical**). Assim, as diversas topologias podem ser combinadas com o problema desejado a fim de prover um elemento específico .

Uma área que ganhou um significativo impulso com o advento da POO é possibilidade de implementação robusta e extensão de programas de elementos finitos adaptativos. Devloo (Devloo et al. 1995, Devloo 1997) propõe um ambiente bastante completo para análise de problemas através do MEF. Além das classes de análise, as classes de modelo foram divididas em modelo geométrico e modelo computacional. O modelo geométrico é governado pela classe **TGeoGrid**, e manipula três entidades geoméricas: nós (classe **TGeoNod**), elementos (classe **TGeoEl**) e nós do contorno da malha (classe **TGeoNodBc**), cujos objetos são armazenados em listas binárias. O modelo computacional é gerenciado pela classe **TCompGrid**, que armazena listas de graus de liberdade (classe **TDofNod**), material (classe **TMaterial**), condições de contorno (classe **TBndCond**) etc. Esta classe deriva

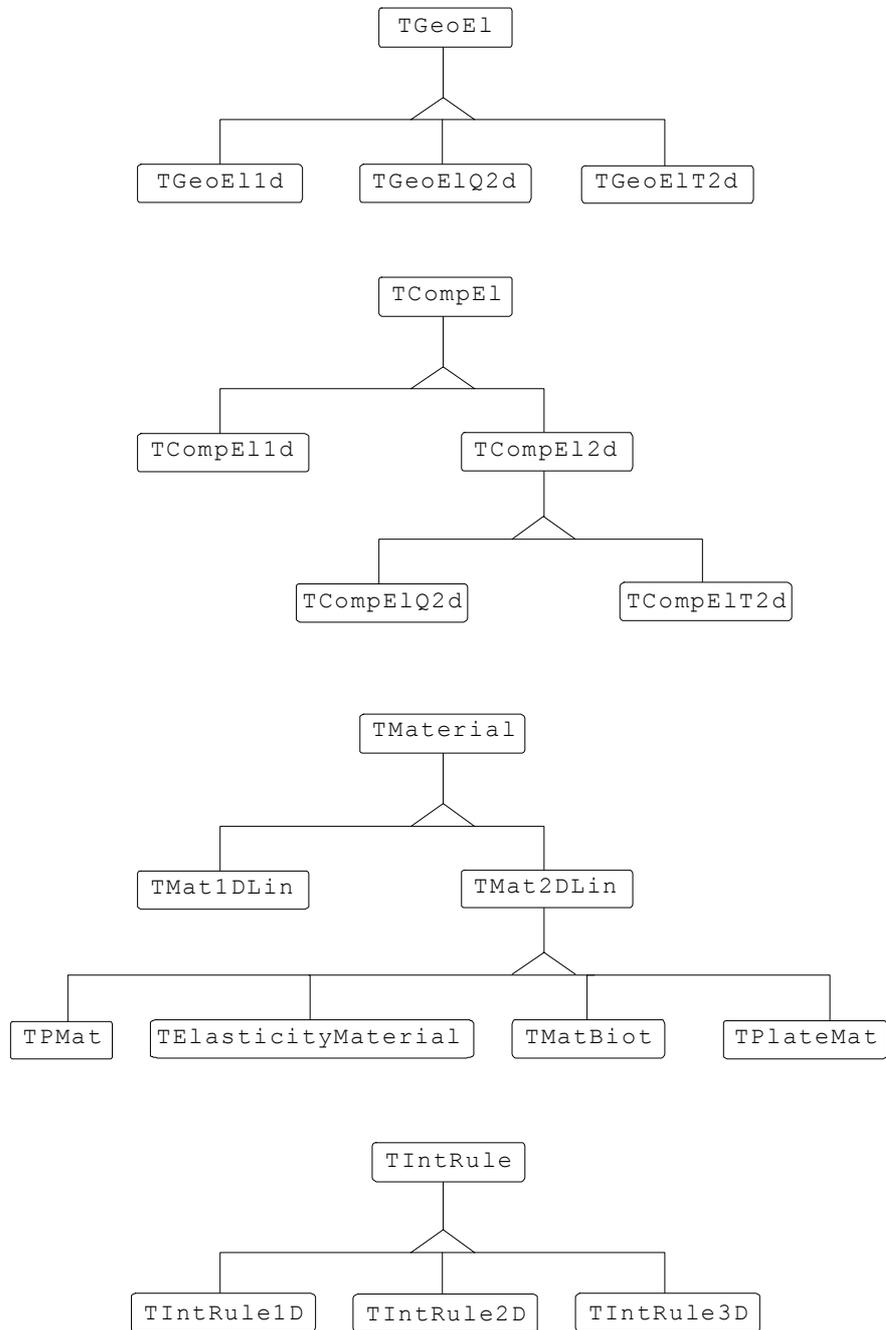


Figura 6-7: Hierarquia de algumas das classes de Devloo (1997).

Zimmermann 1993) para problemas de plasticidade. No entanto, esta extensão torna obrigatória a redefinição de muitas das classes da arquitetura original. O trabalho de Olsson (1998) utiliza uma implementação em C++ para problemas geometricamente não-lineares de vigas. Além das classes usuais para o modelo, são descritos os métodos de duas classes para análise, responsáveis pelo gerenciamento do processo incremental ao longo de uma trajetória de equilíbrio não-linear. Um programa de elementos finitos para problemas de plasticidade extremamente interessante para foi desenvolvido por Commend (1998), e este trabalho ilustra o grau de aperfeiçoamento que uma abordagem OO para problemas não-lineares específicos pode atingir. Também são encontrados trabalhos em outras áreas importantes da Mecânica dos Sólidos como problemas de contato (Feng 1995) e propagação de ondas (Budge & Peery 1993), além de outras áreas tal como transferência de calor (Tiller 1993, Cross et al. 1997, Masters 1997) e eletromagnetismo (Silva et al. 1994). Além disto, são bastante numerosos os artigos relativos a pré e pós-processamento gráfico de resultados numéricos, provavelmente devido à evidente proximidade evolutiva destes programas com os sistemas gráficos de modelamento geométrico (ver, por exemplo, Bettig & Han 1996).

As publicações acima demonstram algumas das diversas possibilidades de implementação de programas de elementos finitos OO. Entretanto, todas possuem algum nível de dependência entre suas classes, possivelmente devido à cultura de programação em linguagens convencionais como Fortran, Pascal e C. Mais recentemente, alguns trabalhos têm destacado a necessidade de se isolar o máximo possível as diversas classes umas das outras, em especial as classes responsáveis pelo modelo das classes responsáveis pela análise. Isto é consequência direta da necessidade de se realizar alterações e extensões nos programas, a fim de capacitá-los para outros tipos de análise. Caso exista uma interdependência muito grande entre as classes, mesmo aquelas modificações mais simples podem ficar comprometidas. O trabalho de Menétrey & Zimmermann (1993) é um exemplo concreto disto, onde a extensão de um programa para problemas de plasticidade envolveu modificações nas classes originais. Embora isto seja tolerável para firmas que desenvolvem softwares, impõe uma grande limitação para usuários de bibliotecas de classes, caso não se disponha de classes abstratas suficientemente genéricas.

Alguns trabalhos mais recentes apresentam arquiteturas com aparente sucesso em se

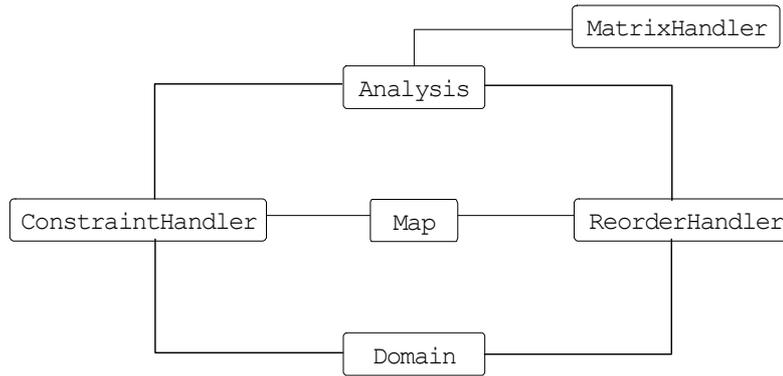


Figura 6-8: Hierarquia das classes de maior nível de Archer (1996).

isolar as classes de hierarquia mais alta umas das outras. O trabalho de Hedegal (1994), embora não atinja este objetivo, fornece uma visão clara de como isto pode ser realizado, especialmente se uma determinada arquitetura já está suficientemente implementada e testada. As funções implementadas pelo usuário (retângulo tracejado na figura 6-4) podem ser facilmente transformadas em classes de análise, desde que já tenham comprovado sua funcionalidade para os diversos problemas, lineares ou não-lineares.

Os trabalhos de Archer et al. (Archer 1996, Archer et al. 1996a, Archer et al. 1996b) apresentam seis classes básicas para análise linear e não-linear de problemas estáticos ou dinâmicos: **Analysis** (análise), **Model** (modelo), **ConstraintHandler**, **ReorderHandler**, **Map** e **MatrixHandler** (figura 6-8). Esta proposta apresenta uma hierarquia de classes rasa, dificultando a reutilização do código. Por outro lado, os autores obtiveram sucesso em isolar o modelo de elementos finitos da análise propriamente dita. Isto é realizado através de uma classe denominada **Map**, que permite a comunicação entre as classes **Model** e **Analysis** auxiliada por classes de manipulação de graus de liberdade, de imposição de restrições e de matrizes. Demais classes derivam destas classes básicas. Também foram desenvolvidas classes para subestruturas.

Besson et al. (1997) propõem uma arquitetura bastante completa, compreendendo classes de modelo e de análise integradas em uma classe-mãe **Problem**, além de diversas classes auxiliares. Esta abordagem foi criada especificamente para facilitar o desenvolvimento a longo prazo de softwares de elementos finitos. A hierarquia das classes é relativamente ampla e foi projetada para reduzir a interdependência entre as classes, facilitando a ex-

pansão do código. A proposta é dotada de classes abstratas para solução de casos bastante específicos como problemas de contato e otimização, além de algoritmos para controle de convergência de processos incrementais. Infelizmente, os detalhes de sua implementação são omitidos.

O trabalho de McKenna (1997) constitui um exemplo representativo do alto grau de evolução que as arquiteturas OO para elementos finitos atingem atualmente. Trata-se de um projeto bastante avançado, incluindo classes de modelo e de análise muito bem desenvolvidas e independentes, além de classes para decomposição de domínio (subestruturas) e classes voltadas especificamente para processamento paralelo. A profundidade da hierarquia de classes é a mais alta dos trabalhos aqui analisados, garantindo um ótimo nível de extensibilidade e reusabilidade do código (figuras 6-9 e 6-10). As classes do modelo são gerenciadas por uma classe chamada `AnalysisModel`, que se encarrega de separá-las das classes de análise e fazer a comunicação entre elas. Esta arquitetura é dotada de classes de análise especiais divididas em dois grupos: tipo de análise e tipo de algoritmo a ser usado. Isto é realizado por duas superclasses: `Analysis` e `SolutionAlgorithm`. Da classe `Analysis` derivam diversos tipos de análise possíveis: análise estática, modal, transiente etc., enquanto da classe `SolutionAlgorithm` derivam subclasses que executam os diversos algoritmos requeridos: `Linear`, `Newton-Raphson`, `BFGS` etc. Isto permite que um dado tipo de análise possa ser realizada por diferentes algoritmos, se desejado. Adicionalmente, as classes `SystemOfEqn` e `Solver` permitem fazer uso da esparsidade típica das matrizes de elementos finitos e adotar diversos métodos de solução do sistema de equações. A figura 6-10 ilustra uma parte desta arquitetura, detalhando um pouco mais algumas das classes ilustradas na figura 6-9.

Também existe um grande número de referências abordando necessidades específicas de programas de engenharia. Mackie (1998) desenvolve uma abordagem relacionando dinamicamente entidades geométricas (pontos, linhas e superfícies) e entidades típicas de elementos finitos (nós e elementos), com vistas ao uso em subestruturação de malhas. Com o advento dos computadores dotados de múltiplos processadores, algumas arquiteturas têm sido capacitadas para processamento paralelo (Duarte 1995, Adeli & Yu 1995, Devloo 1997, Sotelino et al. 1998, McKenna 1997, Masters 1997). Isto permite a solução eficiente de grandes problemas de engenharia, com classes especiais gerenciando a distribuição de

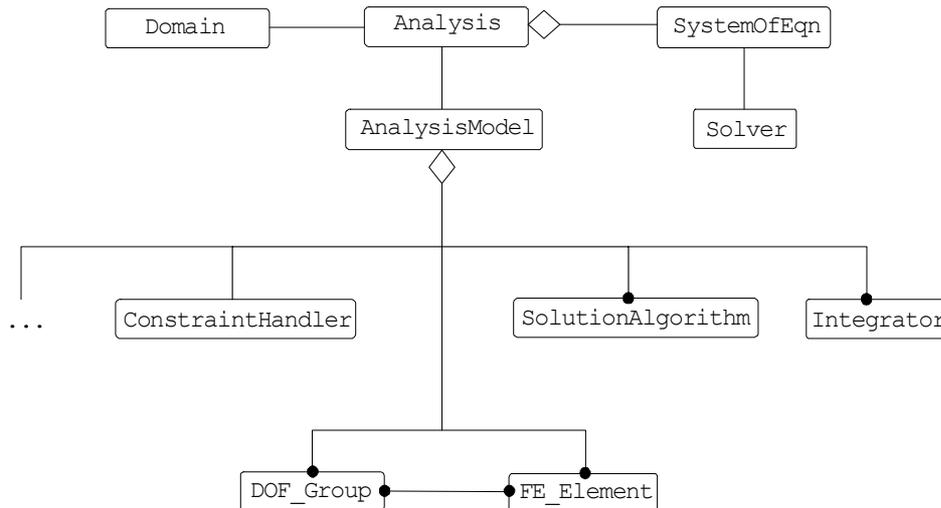


Figura 6-9: Hierarquia de classes de maior nível de McKeena (1997).

tarefas para os processadores.

No âmbito do MEC as publicações relativas a POO são bem mais escassas. Uma revisão de trabalhos publicados sobre o assunto ilustra que de 1996 a 1999 foram divulgados 137 trabalhos de técnicas OO com elementos finitos e apenas 15 com elementos de contorno (Mackerle 2000). Usualmente, procura-se adaptar classes já desenvolvidas para uso no MEF, por simplicidade. No entanto, esta solução seguramente não é a mais adequada em todos os casos, dada a natureza diversa de ambos os métodos. Classes específicas para uso no MEC devem ser desenvolvidas para maior eficiência e reusabilidade do código, embora diversas idéias presentes em programas de elementos finitos possam ser rapidamente adaptadas. Em alguns casos pode-se chegar a arquiteturas bastante especializadas para o MEC, como demonstrado no trabalho de Lage (1998). Noronha et al. (1996) desenvolveram uma arquitetura interessante, permitindo tanto a implementação de programas baseados no MEC convencional quanto nas versões simétricas do método. Este trabalho mostra uma das primeiras tentativas de se utilizar classes específicas de elementos de contorno como uma classe `fundSol`, responsável pela solução fundamental do problema, e uma classe `integral`, que gerencia procedimentos de integração especiais para integrais singulares. Outro trabalho recente lidando com a aplicação de POO em elementos de contorno foi publicado por Wang et al. (1999). Este trabalho se caracteriza por limitar sobremaneira a abrangência da arquitetura proposta devido à inexistência de

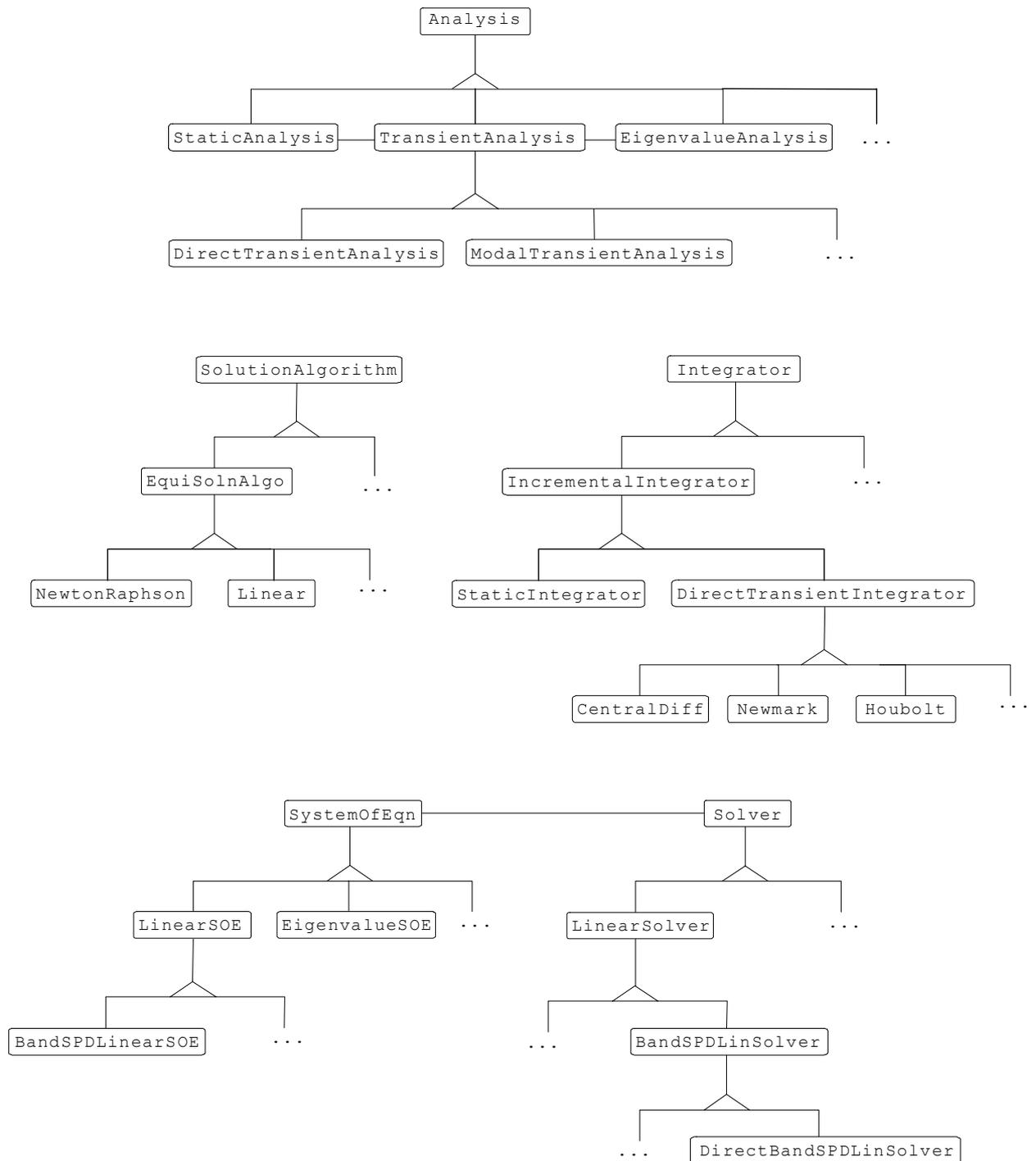


Figura 6-10: Hierarquia de algumas subclasses da arquitetura de McKeena (1997) (ver figura 6-9).

classes de análise e por abranger apenas problemas cujas equações possam ser reduzidas a um sistema linear. É proposta uma classe-mãe `BasicMessage`, que não contém quaisquer métodos, mas apenas dados (número de nós, de elementos, matriz do sistema linear etc.). Desta classe derivam demais classes contendo métodos virtuais para cálculo das matrizes dos elementos e sobreposição das mesmas.

Um bom exemplo do grau de encapsulamento que se pode obter com linguagens OO, no contexto do desenvolvimento de programas de engenharia, é o trabalho de Jeremić & Sture (1998). Este trabalho mostra a implementação de uma biblioteca para manipulação de tensores gerais multidimensionais. Seguindo-se esta filosofia, abre-se uma alternativa excelente para manipulação simbólica de relações constitutivas e de soluções fundamentais, por exemplo. Idéias semelhantes já haviam sido discutidas por Miller (1993) para elementos finitos isoparamétricos independentes de sistemas de coordenadas, mas sem maiores detalhes da implementação. Outro exemplo é o trabalho de Foerch et al. (1995), que ilustra o uso de polimorfismo para implementação de relações constitutivas arbitrárias em elementos finitos, tornando bastante eficiente a implementação de materiais que requerem algoritmos complexos, como materiais viscoelastoplásticos.

Quanto às classes auxiliares, no contexto do presente trabalho, estas são geralmente constituídas por classes para manipulação de matrizes (incluindo solução de sistemas), vetores (incluindo operações algébricas básicas) e listas (incluindo o gerenciamento das mesmas: inserção, remoção e modificação de termos), além de outras ferramentas. Excelentes bibliotecas para manipulação de matrizes podem ser encontradas na literatura. Obviamente, outros tipos também são encontrados, dependendo da aplicação, mas estas três categorias englobam a grande maioria das classes auxiliares. As bibliotecas de classes de Lu et al. (1995), Scholz (1992) e Zeglinski et al. (1994) são bons exemplos da farta literatura sobre classes para matrizes. Davies (1997) desenvolveu uma excelente biblioteca de domínio público, com métodos de solução para diversos tipos de matrizes reais. Pozo (1997) apresenta uma biblioteca similar, mas implementada com o uso de gabaritos (*templates*). A biblioteca de Barker (1996) implementa, além de classes para matrizes e vetores, também uma classe especial para manipulação de cadeias de caracteres (*strings*), o que é muito útil para manipulação de nós, elementos, materiais etc. identificados por palavras, permitindo variações sobre a tradicional identificação por números inteiros.

Outros exemplos de arquiteturas gerais podem ser encontradas em pacotes comerciais e de domínio público, que em geral oferecem uma interface bastante ampla para o usuário (Beck et al. 1995, Langtangen 1996). Infelizmente, tais pacotes normalmente não oferecem classes de análise muito especializadas, dado que são desenvolvidos para uma faixa de aplicações razoavelmente abrangente, deixando para o usuário a programação da estratégia de solução. Por outro lado, possuem funções de pós-processamento gráfico bastante avançadas, facilitando a tarefa de gerenciamento e interpretação dos resultados.

Finalmente, é oportuno destacar que a utilização do paradigma de orientação a objetos tem propiciado também o desenvolvimento de programas com capacidade de manipulação simbólica das equações de um problema. Esta filosofia permite o desenvolvimento das equações do problema até estágios bastante avançados antes de se proceder a solução numérica. A automatização destas etapas permite a derivação da forma fraca da equação diferencial, discretização e obtenção das matrizes correspondentes sem necessidade de cálculos manuais (Dubois-Pèlerin & Pegon 1998, Eyheramendy & Zimmermann 1996a). Ou seja, permite a automatização do desenvolvimento dos programas⁴. Alguns trabalhos nesta linha foram publicados por Zimmermann e seus colaboradores (Zimmermann & Eyheramendy 1996, Eyheramendy & Zimmermann 1996b, Eyheramendy & Zimmermann 1998, Zimmermann et al. 1998).

6.4 A biblioteca mcBEM

A implementação computacional do presente trabalho é baseada em uma biblioteca de classes denominada mcBEM, desenvolvida em linguagem C++. O objetivo desta seção é apresentar as classes mais importantes desta biblioteca e alguns detalhes da sua concepção.

A filosofia adotada no seu desenvolvimento se concentra na aplicação de técnicas de programação OO para desenvolver componentes de programas flexíveis, modulares e re-utilizáveis para solução de equações diferenciais através do MEC direto, usando de um conjunto básico de classes de modelo, de análise e auxiliares. A idéia por trás de seu projeto se baseia no fato de diferentes aplicações numéricas compartilharem uma mesma estrutura matemática e numérica e, mais importante, classes de auxiliares ou de modelo

⁴Estes programas são conhecidas na literatura como ferramentas CASE (*Computer Aided Software Engineering*).

não devem realizar qualquer tarefa de análise. Assim, o usuário que deseje criar um código deve simplesmente *montar* a agregação que contém os objetos necessários. O trabalho de programação fica então limitado a implementar novas classes de análise (caso não existam) derivando-as de uma das classes existentes.

Uma característica importante desta biblioteca é que esta foi estruturada, desde sua concepção inicial, para permitir seu uso por diversos métodos de solução de equações diferenciais baseados em discretização do domínio de solução. Evidentemente algumas particularizações se fazem necessárias para utilizá-la com outros métodos, mas capacita esta arquitetura a abranger - além do MEC - não apenas o MEF como também diferenças finitas, volumes finitos, volumes de controle e métodos de função de Green.

Deve-se destacar ainda que foi dada particular atenção à desvinculação das classes de análise das classes do modelo. Isto foi necessário não apenas para permitir a inclusão futura de outros tipos de análise, mas também para generalizar o uso da biblioteca por outros métodos de solução como os citados acima. Desta forma, a estrutura da biblioteca preserva as características e particularidades inerentes de cada método através de derivação de classes-mãe suficientemente genéricas. Estas últimas, por sua vez, estando em um nível hierarquicamente superior, permitem manusear aquelas entidades comuns a todos os métodos, aliviando as subclasses derivadas de funções-membro repetitivas.

Para que este objetivo pudesse ser alcançado, as classes de análise constituem um conjunto de unidades lógicas que realizam tarefas importantes nas diversas etapas da solução. Assim, o usuário pode montar um programa de acordo com o tipo de problema que será resolvido: linear ou não-linear, estático ou dinâmico, tipo de sistema de equações resultante, estratégia de solução que será utilizada e assim por diante.

A seguir descreve-se superficialmente as classes mais importantes assim como seus dados e métodos mais relevantes, utilizando os níveis de abstração definidos na secção 6.2.

6.4.1 Classes auxiliares

Conforme já discutido, o objetivo maior das classes auxiliares é gerar objetos com funções de suporte (mas não menos importantes) ou mesmo funções que facilitem o uso da biblioteca por parte do usuário. A implementação deste trabalho contou com um grande número

de classes auxiliares que encapsulam muitas das características/operações normalmente requisitadas na mecânica computacional. Alguns exemplos relevantes implementados são dicionários, listas, identificadores, iteradores, manipuladores de erro, manipuladores de arquivos, graus de liberdade, soluções fundamentais e integradores numéricos.

O modelo de armazenamento e manipulação dos sistemas matriciais gerados são de particular importância na eficiência da solução numérica. Com o objetivo de prover classes para estas funções foram implementadas duas superclasses que operam juntas: `mcSystemOfEquations` e `mcSolver`. Uma delas é responsável apenas pelo armazenamento e manipulação do sistema matricial, enquanto a outra é a que efetivamente resolve o sistema. Esta abordagem dá mais flexibilidade ao sistema que os tradicionais *solvers* porque permite a troca de método de solução durante a análise. A figura 6-11 ilustra a hierarquia implementada.

Classe `mcSystemOfEquations`

É responsável pelo armazenamento e gerenciamento do sistema de equações. Esta é uma classe virtual genérica de onde podem ser derivadas classes para as diversas aplicações. No presente trabalho, derivou-se uma outra classe virtual denominada `mcLinearSOE`, para atuar como classe-mãe de sistemas de equações lineares. Desta, por sua vez, derivam outras duas sub-classes para sistemas lineares simétricos e não-simétricos. Estas classes são providas de métodos para montagem das sub-matrizes dos elementos, para rearranjo de linhas e colunas, para eliminação ou adição de variáveis etc. Alguns esquemas de armazenamento são previstos para acomodar os diferentes tipos de sistemas gerados pelos vários métodos (cheia, banda, simétrica, não-simétrica etc.). Como já salientado, objetos `mcSystemOfEquations` não realizam a solução do sistema propriamente dita. Os principais métodos desta classe operam sobre a forma geral

$$\mathbf{Ax} = \mathbf{b} \tag{6.1}$$

e estão analisados a seguir:

► Método `setSize`: Permite a alteração das dimensões do sistema. A definição deste

membro é

```
void setSize(int newsize);
```

► Método `addA`: Realiza a superposição ou substituição de um bloco da matriz **A** de (6.1). Foi especialmente projetado para superposição das matrizes elementares de elementos de contorno ou elementos finitos. Duas versões básicas são mais utilizadas, sendo definidas como

```
int addA(const Matrix& m, mcArray& pos, double fact);  
int addA(const Matrix& m, mcArray& row, mcArray& col, double fact);
```

onde `m` é a submatriz que se deseja adicionar à **A**, `row` e `col` indicam sobre quais as linhas e colunas de **A** a submatriz será adicionada. Caso `row` e `col` sejam idênticas, utiliza-se a primeira definição acima. O parâmetro `fact` é utilizado como multiplicador da submatriz.

► Método `addB`: Similar à `addA`, mas operando sobre o vetor **b**. É particularmente útil para superposição de vetores de carregamento elementares. Sua definição básica é

```
int addB(const Vector& v, mcArray& pos, double fact);
```

► Métodos `getX`, `getA` e `getB`: São utilizados para retornar as matrizes e vetores do sistema. O método `getX` retorna a última solução obtida. Suas definições são:

```
Matrix& getA();  
Vector& getB();  
Vector& getX();
```

► Método `solve`: Este método não resolve efetivamente o sistema. Ao invés disso, este método simplesmente requisita ao objeto `mcSolver` associado para realizar esta operação:

```
int solve();
```

Este método retorna um número inteiro associado à mensagens de erro.

Classe `mcSolver`

É esta superclasse que efetivamente resolve o sistema de equações. Por ser independente de `mcSystemOfEquations`, um objeto `mcSolver` pode ser conectado com bibliotecas numéricas conhecidas (mesmo as programadas em outras linguagens) como a Linpack, Eispack, Lapack, TNT etc. (Zeglinski et al. 1994, McKenna 1997). A hierarquia lógica é muito similar à de `mcSystemOfEquations`. A diferença entre as classes deste grupo é essencialmente algorítmica, isto é, os métodos de solução do sistema é que diferem entre si. Também o acesso às matrizes é diferente para cada tipo de sistema linear, devido às diversas formas de armazenamento. Portanto, o único método que se faz necessário é o que inicia o processo de solução, descrito a seguir. Note-se que como todos os componentes do sistema linear são passados como parâmetros, diversos sistemas lineares podem utilizar o mesmo objeto `mcSolver`.

► Método `solve`: Dispara o início da solução do sistema. Sua interface é definida como:

```
int solve(Matrix& A, Vector& b, Vector& x);
```

onde `A`, `b`, `x` são os componentes de um sistema geral como em (6.1). O método retorna um número inteiro associado à mensagens de erro.

Além destas, um grande número de outras classes auxiliares foi desenvolvido para compor um núcleo básico sobre o qual outras classes pudessem ser desenvolvidas. Embora muitas destas classes sejam concretas e possam ser utilizadas de forma isolada, sua real importância está na possibilidade de agregação delas para construir objetos mais sofisticados, normalmente associados às classes de modelo. Dentre as classes auxiliares desenvolvidas para o presente trabalho, destacam-se:

Classe `mcCoor`

Implementa um ponto coordenado no espaço. É utilizada para aliviar objetos como nós e pontos do trabalho de gerenciamento das coordenadas.

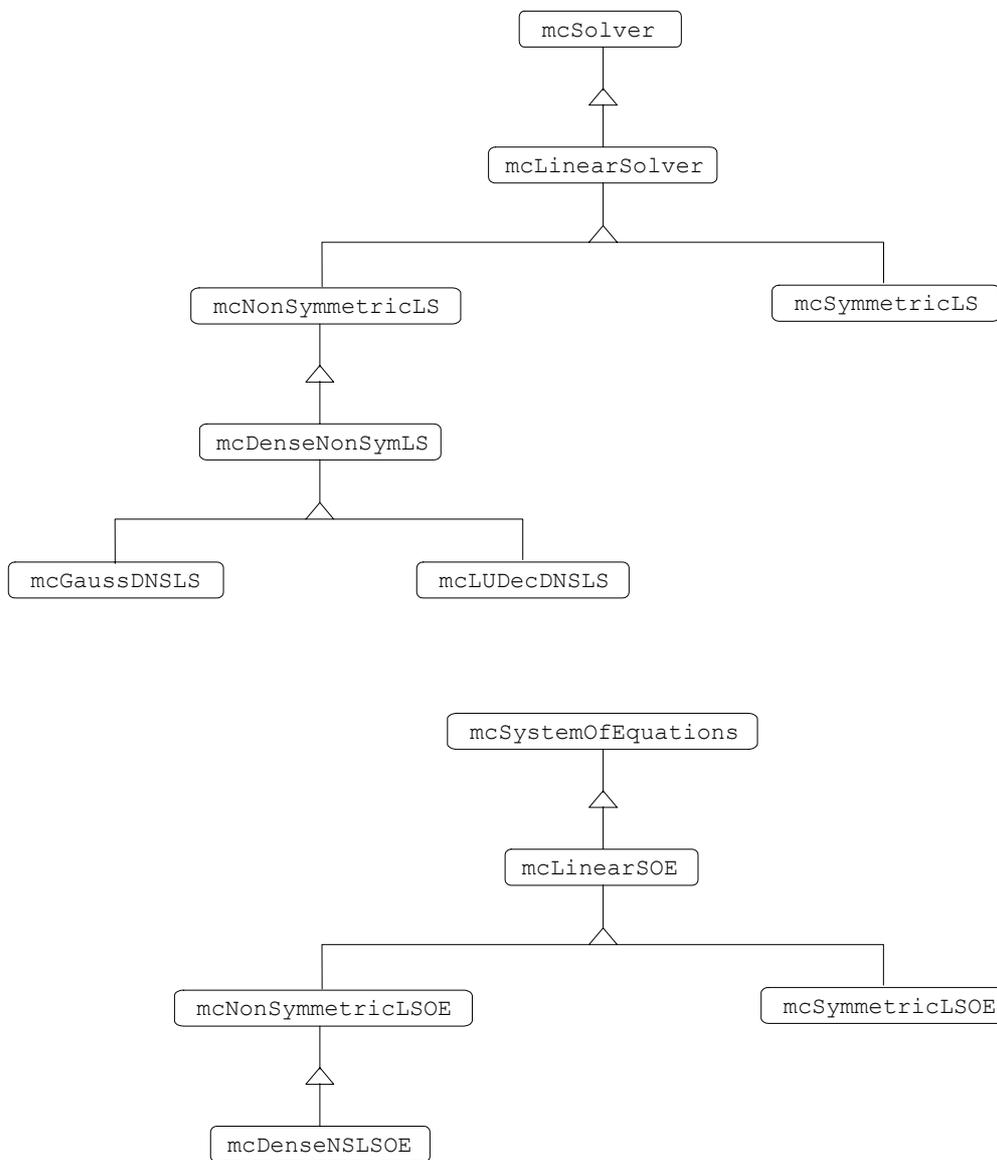


Figura 6-11: Hierarquia básica derivada das classes `mcSystemOfEquations` e `mcSolver`.

Classe mcID

É uma classe de identificação. Um objeto mcID possui dois dados básicos: um número e um nome. Isto permite que o usuário dê nomes aos objetos mais importantes, como alternativa à comum identificação apenas por números.

Classe mcList

Constitui o objeto de armazenamento utilizado para alocar listas de entidades. Quaisquer objetos mcEntity podem ser armazenados aqui. Estas listas são utilizadas por iteradores para automatizar varredura de entidades em *loopings*, embora elas incorporem alguns métodos de iteração primitivos próprios. As listas contam com sobrecarga de alguns operadores binários, o que torna muito simples tarefas de união e intersecção. Por exemplo, se o usuário deseja construir uma lista que contém todos os elementos pertencentes a uma subregião chamada **Aluminio** e que compartilham o nó número 25, o seguinte fragmento de código realiza a operação:

```
lista1 = theModel->getBElementsSharing(
        theDomain->getBESubregion('Aluminio') );
lista2 = theModel->getBElementsSharing(theDomain->getNode(25));
lista3 = lista1 && lista2;
```

Classe mcDictionary

Um objeto dicionário é utilizado para construir listas contendo uma associação código-número-significado. Os dicionários são utilizados para verificar a validade de entidades nativas da biblioteca mcBEM (tipos de materiais, de elementos, de carregamentos etc.). Também permitem uma associação entre números e nomes de graus de liberdade, tensores de soluções fundamentais etc.

Classe `mcArray`

Implementa um arranjo (matriz ou vetor) de números inteiros. É utilizada para criar listas de ordenação, conectividade, graus de liberdade etc.

Classe `mcFunction`

É utilizada para criar funções matemáticas genéricas. Polinômios são o caso mais comum, mas pode-se utilizar variações derivadas desta como `mcDiscreteFunction`, que permite o uso de uma função dada por um conjunto de pontos discretos. Diversos descendentes foram implementados, com uma até três variáveis independentes.

Classe `mcGeometricPartition`

Esta classe implementa partições geométricas utilizadas pelas partições de domínio, conforme destacado no Capítulo 4. É basicamente constituída por uma lista de pontos e suas funções de interpolação associadas. Também dispõe de funções para calcular Jacobianos, número de coordenadas necessárias, funções de mapeamento para o domínio normalizado e vice-versa etc. A conectividade é recuperada através de diversas versões do membro `getConnect`. As versões básicas de outros métodos importantes desta classe estão descritas a seguir.

► Método `getNormCoor`: Retorna as coordenadas normalizadas de um dado ponto.

```
Vector getNormCoor(int tag) const;
```

► Método `getShape`: Retorna o valor das funções de interpolação geométricas em uma dada coordenada normalizada.

```
Vector getShape(double c1, double c2, double c3) const;
```

► Método `getDShape`: Retorna uma matriz contendo as derivadas das funções de interpolação geométricas com respeito às coordenadas normalizadas.

```
Matrix getDShape(double c1, double c2, double c3) const;
```

► Método `detJ`: Calcula o Jacobiano do mapeamento geométrico.

```
double detJ(double c1, double c2, double c3) const;
```

► Método `interpolate`: Este método realiza a interpolação da geometria em uma dada coordenada normalizada.

```
double interpolate(Vector& values,  
                  double c1, double c2, double c3) const;
```

Desta classe-mãe descendem as classes particulares para partições uni, bi ou tridimensionais, dadas por `mc1DGeometricPartition`, `mc2DGeometricPartition` etc. Destas, a hierarquia continua para definir linhas, áreas e volumes.

Classe `mcPhysicalPartition`

Implementa as partições físicas utilizadas pelas partições de domínio. Ao contrário de `mcGeometricPartition`, é constituída por uma lista de nós e suas funções de interpolação associadas. A conectividade é recuperada através de diversas versões do membro `getConnect`. Os métodos mais importantes desta classe são basicamente os mesmos de `mcGeometricPartition`, mas utilizando as funções de interpolação das variáveis físicas.

Desta classe-mãe descendem as classes particulares para partições uni, bi ou tridimensionais, dadas por `mc1DPhysicalPartition`, `mc2DPhysicalPartition` etc, e destas descendem as definições de interpolação sobre linhas, áreas e volumes.

Classe `mcDOF`

Representa um único grau de liberdade. Armazena o valor do grau de liberdade, seu número correspondente ao sistema de equações e indicadores de estado (precrito/não-prescrito e ativo/inativo).

Classe `mcDOFSet`

É composta por uma lista de objetos `mcDOF`. Assim, o usuário pode montar um conjunto de graus de liberdade nodais de acordo com a equação diferencial que está sendo resolvida. Provê acesso aos mesmos métodos de `mcDOF` para cada objeto da lista. Para uso com o

MEC, desta classe normalmente descendem classes contendo os graus de liberdade primais, duais e suas derivadas primeiras. Por exemplo, os problemas de elasticidade bidimensional utilizam as seguintes subclasses:

- `mcPlaneStressDisp`: Contém os deslocamentos u_1 e u_2 .
- `mcPlaneStressTrac`: Contém as trações t_1 e t_2 .
- `mcPlaneStressGradDisp1`: Contém os termos do gradiente de u_1 : $u_{1,1}$ e $u_{1,2}$.
- `mcPlaneStressGradDisp2`: Contém os termos do gradiente de u_2 : $u_{2,1}$ e $u_{2,2}$.

Subclasses similares são utilizadas para outros problemas.

Classe `mcFundSolution`

Esta classe foi desenvolvida para encapsular os dados e métodos relevantes para descrever o tipo de equação diferencial que governa uma subregião, bem como os tensores fundamentais correspondentes. Desta forma, o usuário precisa somente informar qual a equação diferencial que se pretende resolver, e o objeto `mcFundSolution` associado sabe quantos e quais são os graus de liberdade primais e duais do problema, qual o número e a ordem do sistema de equações diferenciais além de outros dados. Os principais membros estão listados a seguir.

► Método `isScalar`: Indicador de problema escalar ou vetorial.

```
bool isScalar() const;
```

► Método `Tensor`: Calcula um dado tensor em função do ponto-fonte e ponto-campo, utilizando um dado vetor normal.

```
Matrix Tensor(char* id, mcCoord& q, Vector& n);
```

onde `id` identifica qual o tensor desejado, `q` é o objeto `mcCoord` do ponto campo e `n` contém o vetor normal.

► Método `index`: Retorna um arranjo contendo o ordem de um tensor.

```
mcArray index(char* id);
```

► Método `Limit`: Calcula as expansões assintóticas de um dado tensor, para fins de cálculo das integrais singulares.

```
Matrix Limit(int ns, char* id, Vector& N, Vector& T,  
             Matrix& Shape, Matrix& dShape);
```

onde `ns` identifica o nó singular, `id` identifica o tensor desejado, `N` e `T` contêm os vetores normal e tangencial, enquanto `Shape` e `dShape` contêm as funções de interpolação e suas derivadas.

► Método `Jump`: Calcula os termos livres resultantes da aplicação do traço a um dado tensor. O caso mais comum são os fatores geométricos C_{ij} , mas quaisquer termos de salto podem ser incluídos.

```
Matrix Jump(Vector& ang, char* id);
```

► Método `getSingType`: Retorna um arranjo de códigos indicando o tipo de singularidade que cada componente de um dado tensor possui. Isto é de vital importância para que as integrais sejam calculadas corretamente.

```
mcArray getSingType(char* id);
```

► Método `setLoadPoint`: Especifica a coordenada atual do ponto-fonte.

```
void setLoadPoint(mcCoord& P);
```

Classe `mcIterator`

Estas são classes que implementam iteradores sobre listas que são recebidas como argumentos no seu construtor. Na presente implementação, os iteradores são baseados em gabaritos, de forma que listas de quaisquer objetos podem gerar estes iteradores. Como os objetos `mcList` aceitam operações binárias básicas, este recurso facilita sobremaneira a programação de laços. Por exemplo, suponha-se que o usuário deseje varrer em um laço

todos os elementos contidos no objeto `lista3` usado como exemplo na descrição da classe `mcList`. Isto seria facilmente obtido de diversas formas. O seguinte fragmento de código ilustra duas possibilidades (note-se que não há necessidade de se utilizar contadores):

```
mcIterator<mcBElement> itr(lista3);

// Exemplo 1 (incremento manual):
while (itr) {
    // fazer algo com itr.current() ...
    itr++; // ou itr.next();
}

itr.reset();
// Exemplo 2 (incremento automatico):
for (itr.first(); itr; itr++) {
    // fazer algo com itr.current() ...
}
```

Classe Vector e Matrix

Não foram implementadas classes para matrizes e vetores. Em vez disso, este trabalho utilizou a biblioteca Newmat (Davies 1997). Trata-se de uma implementação bastante geral e voltada apenas para matrizes e vetores reais. Tendo ainda em vista que no MEC as matrizes resultantes são normalmente cheias e não-simétricas, nenhum esquema especial de armazenamento foi necessário.

Classe mcQuadrature

Esta é uma das classes principais proposta por esta arquitetura. O objetivo desta classe é realizar integrações uni, bi ou tridimensionais através de quadraturas numéricas. A integração é realizada no domínio $[-1, +1]$ para cada direção. Foi feito uso intensivo de ponteiros para funções, de forma que o usuário pode realizar a integração fornecendo

o nome da função que calcula o integrando, sem necessidade de recompilação. Desta forma, um único objeto `mcQuadrature` pode ser reutilizado quantas vezes for necessário. A interface básica está apresentada na figura 6-12.

```
template <class T> class mcQuadrature

    mcQuadrature(dim, nip, typ)
    virtual ~mcQuadrature()

    void    setOrder(n1, n2, n3)
    int     getOrder(dir)

    void    setType(t1, t2, t3)
    int     getType(dir)

    Vector  getStations(dir)
    Vector  getWeights(dir)

    void    setIntegrand(mc1DIntegrand)
    void    setIntegrand(mc2DIntegrand)
    void    setIntegrand(mc3DIntegrand)

    T       NIntegrate(acc)
    T       NIntegrate(mc1DIntegrand, acc)
    T       NIntegrate(mc2DIntegrand, acc)
    T       NIntegrate(mc3DIntegrand, acc)
```

Figura 6-12: Interface simplificada da classe `mcQuadrature`.

Alguns métodos desta classe merecem destaque e estão comentados a seguir:

► Método `setOrder`: Especifica a ordem da quadratura (número de pontos) em cada direção.

```
void setOrder(int n1, int n2, int n3);
```

► Método `setType`: Especifica o tipo de quadratura a ser empregado em cada direção. Os tipos mais utilizados são a quadratura de Gauss-Legendre padrão e com transformação cúbica.

```
void setType(int t1, int t2, int t3);
```

► Métodos `getStations` e `getWeights`: Retornam vetores contendo as coordenadas dos

pontos de integração e os respectivos pesos para cada direção. Este método foi incluído para permitir a construção de rotinas de integração customizadas:

```
Vector getStations(int dir);  
Vector getWeights(int dir);
```

► Métodos `setIntegrand`: Informa tacitamente qual o integrando. Este membro dispensa que o usuário informe qual é o integrando desejado cada vez que a integração é realizada:

```
void setIntegrand(mcIntegrand& obj, double acc);
```

► Métodos `NIntegrate`: Executa a integração numérica e retorna o resultado com o tipo correto. Foram implementadas diversas variações baseadas na seguinte definição:

```
T NIntegrate(mcIntegrand& obj, double acc);
```

Além da quadratura de Gauss-Legendre padrão, outros tipos de quadraturas foram implementados para cálculo de integrais fracamente singulares e integrais por parte finita. Mas a grande vantagem no uso desta classe reside no fato de ser baseada em gabaritos. Este mecanismo permite a integração de quaisquer objetos que sobrecarregem as operações algébricas básicas. A figura 6-13 mostra um exemplo de aplicação. Nas linhas 1 a 3 foram criados os objetos `Func`, `Vect` e `Mat` para integração específica de funções reais, vetores e matrizes, respectivamente. Na linha 4, a instrução retorna uma matriz de rigidez `Ke` integrada com quatro pontos de integração em cada direção. Na linha 5 o vetor `Fe` retorna o resultado de uma integração com dois pontos em cada direção, enquanto a linha 6 exemplifica a integração de uma função real com três pontos de integração. Note-se que não há necessidade de realizar a varredura dos elementos das matrizes/vetores. Nestes exemplos, os argumentos passados para o membro `NIntegrate` são os nomes reais das rotinas que calculam os respectivos integrandos. Portanto, os objetos `Func`, `Vect` e `Mat` podem ser reutilizados para outros integrandos simplesmente alterando-se os parâmetros correspondentes.

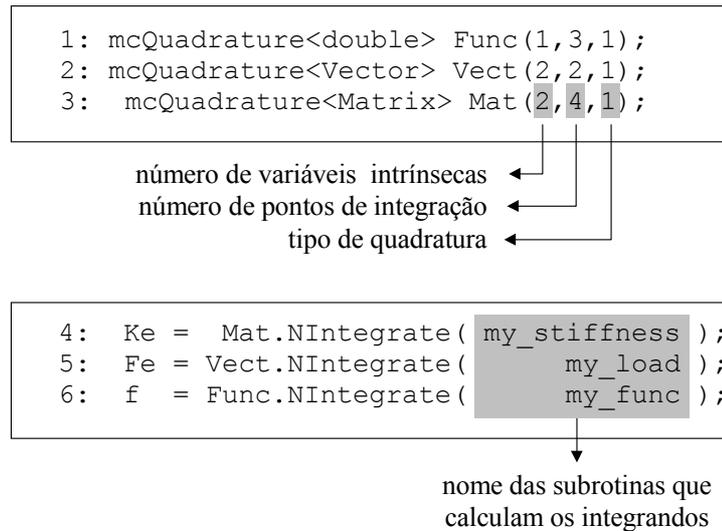


Figura 6-13: Exemplos de aplicação da classe mcQuadrature.

6.4.2 Classes de modelo

As classes de modelo utilizadas pelo McBEM são na maioria derivadas de uma classe-mãe chamada mcEntity. Esta abordagem, por vezes, se revela inadequada por forçar uma associação hierárquica entre entidades que não compartilham uma estrutura lógica de dados ou métodos. Por outro lado, permite que todas entidades do modelo sejam manipuladas e armazenadas pela mesma família de classes de armazenamento, eliminando um *overhead* muito significativo em termos de performance do programa quando da utilização de classes similares de domínio público (por exemplo, *Standard Template Library Programmer's Guide* 1999).

As principais classes de modelo implementadas neste trabalho estão ilustradas na figura 6-14. Uma descrição sucinta das mesmas é apresentada a seguir.

Classe mcPoint

Implementa um ponto coordenado no espaço \mathfrak{R}^3 . Pode estar associado a um sistema de coordenadas específico, além do sistema de coordenadas global. Sua principal função é armazenar e manipular as coordenadas de um ponto.

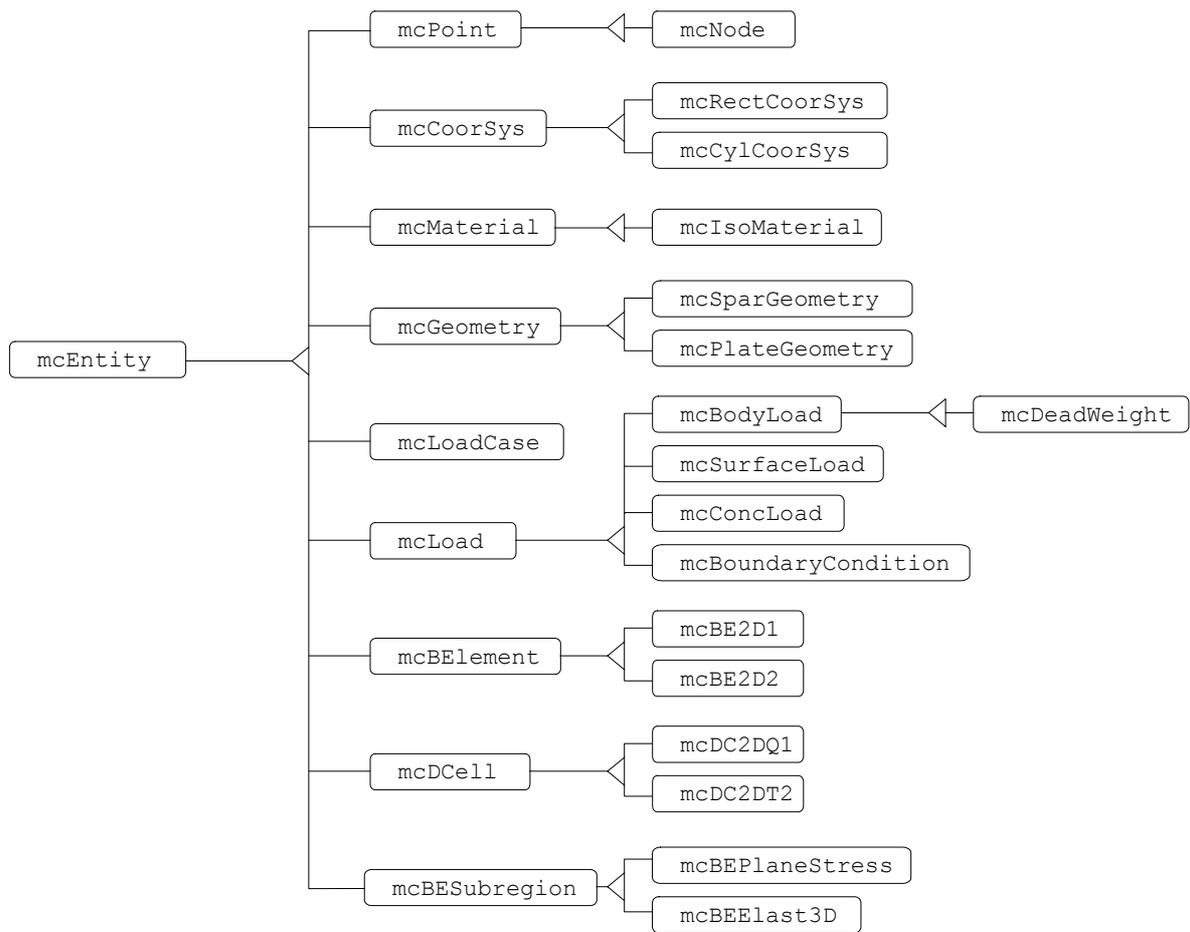


Figura 6-14: Hierarquia básica das classes de modelo.

Classe mcCoorSys

Classe-mãe para sistemas de coordenadas genéricos. Exemplos de classes derivadas desta são as classes `mcRectCoorSys` e `mcCylCoorSys`, que implementam sistemas de coordenadas retangulares e cilíndricos, respectivamente. Sua interface possui métodos para mudança de base de um sistema de coordenadas para outro e cálculo de matrizes de cossenos diretores.

Classe mcGeometry

Classe-mãe para propriedades geométricas. As subclasses `mcSparGeometry`, `mcBeamGeometry` e `mcPlateGeometry` são exemplos de classes derivadas desta. Estas classes armazenam e manipulam as propriedades geométricas como área, espessura, momentos de inércia etc.

Classe mcMaterial

Classe-mãe para propriedades dos materiais. Esta é uma classe genérica para ser utilizada para definir um material genérico, mas possui derivações óbvias como as classes `mcIsoMaterial`, para materiais isotrópicos, e `mcVonMisesPlastMaterial`, para materiais elastoplásticos. Além de armazenar e manipular as propriedades do material, esta classe provê métodos para cálculo das matrizes constitutivas.

Classe mcNode

Implementa um ponto que possui graus de liberdade associados ao problema, isto é, contém as variáveis físicas nodais da solução do problema. É obtida por derivação múltipla das classes `mcCoor` e `mcPoint`. Dentre os métodos que compõem sua interface, destacam-se os responsáveis pela criação ou associação de graus de liberdade, prescrição de valores, ativação/desativação de graus de liberdade e gerenciamento do número da equação associado ao grau de liberdade.

Classe mcBElement

Classe-mãe para elementos de contorno. No presente trabalho foram implementados apenas elementos de contorno bidimensionais, exemplificados pelas subclasses `mcBE2D0`,

`mcBE2D1`, `mcBE2D2`, relativos a elementos de contorno constante, linear e quadrático, respectivamente. A classe `mcBEElement` é particularmente importante no presente trabalho, porque fornece abordagem automatizada para implementação de elementos de contorno genéricos.

Um objeto `mcBEElement` é composto basicamente por uma agregação de duas super-classes: `mcGeometricPartition` e `mcPhysicalPartition`, conforme descrito na seção 4.1. Um objeto `mcBEElement` também tem acesso ao objeto `mcBESubregion` que o contém, a fim de acessar dados como propriedades do material e geometria da subregião. As partições físicas e geométricas são acessadas com os seguintes métodos:

► Método `getGeoPart`: Retorna a partição geométrica do elemento.

```
mcGeometricPartition& getGeoPart() const;
```

► Método `getPhyPart`: Retorna a partição física do elemento.

```
mcPhysicalPartition& getPhyPart() const;
```

Estas classes foram projetadas para realizar o cálculo das matrizes elementares para uma solução fundamental genérica. Na implementação do MEC direto, o cálculo de certas submatrizes envolvendo os tensores das soluções fundamentais é mandatório. Por exemplo, na grande maioria dos códigos é necessário o cômputo das submatrizes \mathbf{H} e \mathbf{G} genericamente dadas por:

$$\mathbf{H}_n = \int_{\Gamma_n} \mathbf{T}(q, p) \Phi(q) d\Gamma_n \quad (6.2a)$$

$$\mathbf{G}_n = \int_{\Gamma_n} \mathbf{U}(q, p) \Phi(q) d\Gamma_n \quad . \quad (6.2b)$$

Outras vezes, ainda, a necessidade está no cálculo de vetores, como na etapa de cálculo das variáveis em pontos internos, onde faz-se necessário o cálculo dos vetores

$$\mathbf{h}_n = \int_{\Gamma_n} \mathbf{T}(q, p) \Phi(q) \mathbf{u}(q) d\Gamma_n \quad (6.3a)$$

$$\mathbf{g}_n = \int_{\Gamma_n} \mathbf{U}(q, p) \Phi(q) \mathbf{t}(q) d\Gamma_n \quad . \quad (6.3b)$$

Estes cálculos são realizados através de uma única linha de programação utilizando os métodos `formMatrix` e `formVector`, descritos a seguir.

► Método `formMatrix`: Responsável pelo cálculo de submatrizes do tipo (6.2). A definição básica deste membro é

```
int formMatrix(Matrix& A, char id, int sing, int nip);
```

onde `A` retorna a submatriz desejada, `id` identifica qual o tensor a ser empregado, `sing` sinaliza se o ponto de singularidade se encontra sobre algum dos nós do elemento e `nip` permite que o usuário especifique um número de pontos de integração diferente do *default*.

► Método `formVector`: Responsável pelo cálculo de subvetores do tipo (6.3). A definição básica deste membro é

```
int formVector(Vector& V, char id, Vector& var, int sing, int nip);
```

onde `V` retorna o vetor desejado e `var` contém o vetor a ser utilizado como último termo do integrando de (6.3). Demais parâmetros são idênticos a `formMatrix`.

O uso generalizada destes membros é possível porque os objetos `mcBElement` (e `mcFElement`) são dotados de um membro `mcQuadrature`. Assim, seja por exemplo um elemento de contorno identificado por um apontador `BE`, o cálculo de (6.2a) é realizado com a seguinte instrução:

```
BE->formMatrix(H, 'T', sing_node);
```

Este comando pede ao elemento apontado por `BE` para calcular uma matriz utilizando o tensor `T` multiplicado pela matriz de funções de interpolação de sua partição física e devolver o resultado em `H`. O nó singular é indicado por `sing_node`, de forma que o elemento sabe se existe singularidade e como proceder. De forma similar, o cálculo de (6.3b) é realizado com a instrução:

```
BE->formVector(g, 'U', trac, sing_node);
```

onde `trac` é o vetor correspondente às variáveis `t`. O resultado retorna no vetor `g`.

Embora não seja explícito, deve-se destacar a elegância desta abordagem. Como o objeto mantém conhecimento sobre as características do elemento, não é necessário que o usuário especifique as dimensões das matrizes, ordem dos tensores ou qual função de interpolação utilizar. O próprio método se encarrega de buscar estas informações. Além disso, um objeto `mcBElement` pode determinar se este é bi ou tridimensional através da dimensão do seu espaço normalizado, e portanto aplica a quadratura com a dimensão correta. Existem variações destes métodos obtidas com sobrecarga de parâmetros, mas não serão discutidas aqui.

Classe `mcDCell`

Classe-mãe para células de domínio. No presente trabalho foram implementados apenas células de domínio bidimensionais. Células triangulares são exemplificados pelas subclasses `mcDC2DT0`, `mcDC2DT1`, `mcDC2DT2`, com funções de interpolação constante, linear e quadrática, respectivamente. Células quadrilaterias seguem a mesma regra, onde os nomes são alterados para `mcDC2DQ0`, `mcDC2DQ1` e `mcDC2DQ2`, respectivamente. Esta classe é muito similar à `mcBElement`, apenas que as dimensões são uma ordem maior.

Objetos `mcDCell` também são compostos por agregação de superclasses de partição física e geométrica, conforme descrito na seção 4.1. Sua implementação é muito parecida com `mcBElement`, e foram projetadas para realizar cálculos similares, apenas que sobre as células de domínio. Um exemplo típico é o cálculo dos subvetores \mathbf{F}_r dos carregamentos de domínio, genericamente dados por:

$$\mathbf{f}_r = \int_{\Omega_r} \mathbf{U}(Q, p) \Phi(Q) \mathbf{q}(Q) d\Omega_r \quad . \quad (6.4)$$

Algumas vezes a convolução no integrando não utiliza as funções de interpolação, mas um outro vetor fornecido pelo usuário, como na forma geral:

$$\mathbf{b}_r = \int_{\Omega_r} \widehat{\mathbf{U}}(Q, p) \mathbf{u}(Q) d\Omega_r \quad . \quad (6.5)$$

Em outras situações, como em problemas não-lineares, é muito comum a necessidade de cálculo de matrizes contendo os termos não-lineares, como no caso da eq.(4.34d):

$$\mathbf{B}_r = \int_{\Omega_r} \hat{\mathbf{U}}(Q, p) \mathbf{N}(Q) d\Omega_r \quad . \quad (6.6)$$

Estes cálculos também são realizados através de métodos chamados `formMatrix` e `formVector`, descritos a seguir.

► Método `formMatrix`: Responsável pelo cálculo de submatrizes do tipo (6.6). A definição mais utilizada deste membro é

```
int formMatrix(Matrix& A, char id, Matrix& mat, int sing, int nip);
```

onde `A` retorna a submatriz desejada, `id` identifica qual o tensor a ser empregado, `mat` contém a matriz que multiplica o tensor, `sing` sinaliza se o ponto de singularidade se encontra sobre algum dos nós do elemento e `nip` permite que o usuário especifique um número de pontos de integração diferente do *default*.

► Método `formVector`: Responsável pelo cálculo de subvetores do tipo (6.4) e (6.5). As duas definições mais utilizadas destes membro são

```
int formVector(Vector& V, char id, Vector& vet, int sing, int nip);
int formVector2(Vector& V, char id, Vector& vet, int sing, int nip);
```

onde `V` retorna o vetor desejado. Na primeira versão, `vet` contém um vetor de variáveis a ser interpolado, como `q` na eq.(6.4). Na segunda versão, o usuário fornece o vetor a ser multiplicado diretamente pelo tensor, como `u` na eq.(6.5). Demais parâmetros são idênticos a `formMatrix`.

Estes membros também fazem uso de um membro `mcQuadrature`. Então, seja por exemplo uma célula identificada por um apontador `DC`, o cálculo de (6.6) é realizado com a seguinte instrução:

```
DC->formMatrix(B, 'dU', N, sing_node);
```

Este comando pede à célula apontada por `DC` para calcular uma matriz utilizando o tensor $\hat{\mathbf{U}}$ multiplicado por uma matriz \mathbf{N} e devolver o resultado em \mathbf{B} . O nó singular é indicado por `sing_node`, de forma que a célula sabe a respeito de quaisquer singularidades e como proceder. De forma similar, o cálculo de (6.4) e (6.5) é realizado respectivamente pelas instruções:

```
DC->formVector(f, 'U', load, sing_node);
DC->formVector2(b, 'dU', disp, sing_node);
```

onde `load` é o vetor correspondente às variáveis \mathbf{q} e `disp` é o vetor correspondente às variáveis \mathbf{u} . Os resultados retornam nos vetores `f` e `b`, respectivamente. A diferença básica entre estas duas instruções está no fato da primeira interpolar os valores nodais contidos em `load` utilizando Φ , o que não ocorre na segunda.

Aqui também os objetos mantêm conhecimento sobre as características da célula, não sendo necessário que o usuário especifique as dimensões das matrizes, ordem dos tensores ou qual função de interpolação utilizar. Novamente, um objeto `mcDCell` pode determinar se este é bi ou tridimensional através da dimensão do seu espaço normalizado e aplicar a quadratura corretamente. Foram implementadas outras variações destes métodos obtidas com sobrecarga de parâmetros.

Classe `mcFElement`

Classe-mãe para elementos finitos. Não foi utilizada no presente trabalho.

Classe `mcBESubregion`

Classe-mãe para subregiões de elementos de contorno. Suas subclasses são obtidas através do encapsulamento de propriedades da equação diferencial associada à subregião. Exemplos destas são as classes `mcPot2D`, `mcEl2DPlaneStress` e `mcElThickPlate`, que implementam subregiões para condução de calor bidimensional, estado plano de tensões e placas de Mindlin/Reissner, respectivamente. Os objetos `mcBESubregion` estão sempre associados a três outros objetos: `mcGeometry`, `mcMaterial` e `mcFundSolution`. Se desejado, o

usuário pode atribuir à uma subregião um sistema de coordenadas particular, a fim de facilitar a orientação de carregamentos e condições de contorno.

Classe `mcFESubregion`

Classe-mãe para subregiões de elementos finitos. Não foi utilizada no presente trabalho.

Classe `mcLoadCase`

Implementa casos de carregamento para análise de múltiplas soluções.

Classe `mcLoad`

Implementa os carregamentos gerais utilizados na análise. Três exemplos de classes derivadas desta são as classes `mcBodyLoad`, `mcSurfaceLoad` e `mcBoundaryCondition` que implementam forças de corpo, forças de superfície e condições de contorno, respectivamente. Em problemas vetoriais, os carregamentos são definidos separadamente para cada direção. Quase todos os objetos descendentes desta classe contam com um método particularmente útil, que permite o cálculo do valor da excitação em uma dada coordenada. Por exemplo, a subclasse `mcSurfaceLoad` permite essa operação com a seguinte definição:

► Método `at`: Retorna o valor da carga em uma dada coordenada espacial. Definições básicas:

```
double at(double x1, double x2, double x3);
double at(Vector& x);
```

A implementação atual não conta com problemas transientes, mas isto foi previsto com a associação de um objeto `mcFunction` relacionado à variável tempo. Este procedimento faz uso da possibilidade de carregamentos dinâmicos serem escritos como uma convolução:

$$f(x, y, t) = g(x, y) h(t) \quad ,$$

onde $g(x, y)$ é o resultado do membro `at(x, y)` de um objeto `mcLoad` e $h(t)$ é um objeto `mcFunction`.

Classe mcPatch

A classe `mcPatch` é utilizada para criar grupos de entidades que são manipulados em conjunto. Três subclasses descendentes desta foram implementadas: `mcPointPatch`, `mcNodePatch` e `mcBEPatch`. Exemplos de aplicação destas classes são *patches* de elementos finitos para recuperação e suavização de tensões, agrupamento de pontos para definição de *splines* e definição de interfaces para aplicação de condições de contorno, carregamentos ou intersecção de subregiões. Os principais métodos de sua interface estão apresentados a seguir.

► Método `getNumberOfEntities`: Retorna número de entidades que compõem a o grupo:

```
int    getNumberOfEntities() const;
```

► Método `getListOfEntities`: Retorna uma lista contendo as entidades que compõem a o grupo:

```
mcList& getListOfEntities() const;
```

► Método `add`: Adiciona uma nova entidade ao grupo:

```
int    add(mcEntity& anObject);
```

Classe mcIntersection

A classe `mcIntersection` foi desenvolvida especialmente para aplicar as condições de compatibilidade sobre a parcela de contorno comum a duas ou mais subregiões. Estas classes definem grupos de *patches* que constituem a união das subregiões no modelo computacional. Durante a fase de análise estes objetos são utilizados para associar corretamente os números dos graus de liberdade para os nós envolvidos.

Classe `mcDomain`

A maior parte das classes descritas acima são utilizadas pela classe `mcDomain`, que essencialmente implementa todo o armazenamento dos dados do problema (o domínio de solução). Suscintamente, `mcDomain` é uma classe *container* dotada de métodos que permitem ao usuário acessar quaisquer objetos criados após a leitura de um arquivo de dados ou durante a execução do programa. No entanto, nenhuma função de análise do problema é realizada por objetos `mcDomain`. Embora existam alguns métodos especiais como os de gerenciamento de casos de carregamento e tempo (para problemas transientes), estes apenas informam ao objeto `mcDomain` o estado das variáveis, mas não operam sobre estas. Ou seja, um objeto `mcDomain` é um banco de dados que contém o modelo computacional do problema.

6.4.3 Classes de análise

Muitas das idéias utilizadas na concepção das classes de análise aqui propostas foram baseadas no trabalho de McKenna (1997), e adaptadas para o MEC. As classes de análise são compostas basicamente por classes derivadas de cinco superclasses:

Classe `mcSolutionAlgorithm`

Esta superclasse é a responsável por orquestrar as etapas da solução do problema. Sua tarefa principal é reger a montagem dos lados direito e esquerdo do sistema de equações e solicitar a solução de sistemas lineares ou de autovalores/autovetores. Isto é realizado invocando os métodos necessários para cada classe de análise. Em problemas lineares estas tarefas são executadas uma única vez, mas em problemas não-lineares ou dinâmicos a sequência é repetida até que a convergência seja atingida. Neste trabalho, duas principais subclasses são derivadas desta: `mcBEMEigenvalueSolAlgo` e `mcBEMEquilibriumSolAlgo`. Ambas podem ser particularizadas para casos especiais, como ilustrado na figura 6-15.

Neste trabalho os problemas estáticos foram implementadas com duas subclasses principais: `mcBEMLinear` e `mcBEMLargeDisp`, que controlam a montagem e a solução do sistema de equações para problemas lineares e de grande deslocamentos, respectivamente. Três métodos compõem a interface básica de objetos `mcSolutionAlgorithm`:

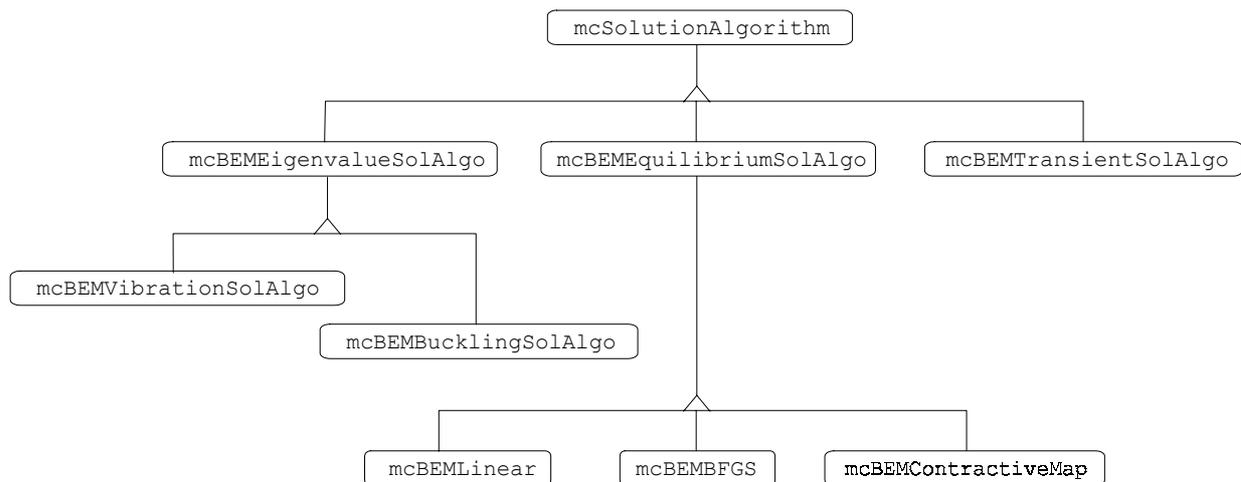


Figura 6-15: Hierarquia básica derivada da classe `mcSolutionAlgorithm`.

`solveCurrentStep`, `postProcessCurrentStep` e `analysisModelChanged`. Uma descrição de cada um deles é dada a seguir.

► Método `solveCurrentStep`: Monta os lados direito e esquerdo do sistema de equações para o passo de carregamento/tempo atual:

```
int solveCurrentStep();
```

► Método `postProcessCurrentStep`: Realiza o pós-processamento do passo atual. No MEC, isto normalmente engloba o cálculo das variáveis em pontos internos:

```
int postProcessCurrentStep();
```

► Método `analysisModelChanged`: Utilizado para verificar se o modelo sofreu alterações e necessita de reanálise. Problemas envolvendo adaptação de malha ou atualização da geometria podem utilizar este recurso para invocar automaticamente uma nova análise:

```
bool analysisModelChanged();
```

As figuras (6-16) e (6-17) ilustram possibilidades de implementação da função-membro `solveCurrentStep` para problemas lineares e não-lineares, respectivamente.

```

int mcBEMLinear::solveCurrentStep() {

    theAssembler->formBVPtangent();
    theAssembler->formBVPUnbalance();
    theConstraint->applyConstraints();
    theSysOfEqs->solve();
    theAssembler->updateBVPDOF(theSysOfEqs->getX());

    return 0;

}

```

Figura 6-16: Código simplificado de `solveCurrentStep` para problemas lineares.

```

int mcBEMLargeDisp::solveCurrentStep() {

    do {

        theAssembler->formBVPtangent();
        theAssembler->formBVPUnbalance();
        theConstraint->applyConstraints();
        theSysOfEqs->solve();
        theAssembler->updateBVPDOF(theSysOfEqs->getX());
        theAssembler->evaluateInternalPoints();
        theAssembler->evaluateGradients();

        // Calcular erro ...

    } while (erro > tol);

    return 0;

}

```

Figura 6-17: Código simplificado de `solveCurrentStep` para problemas não-lineares.

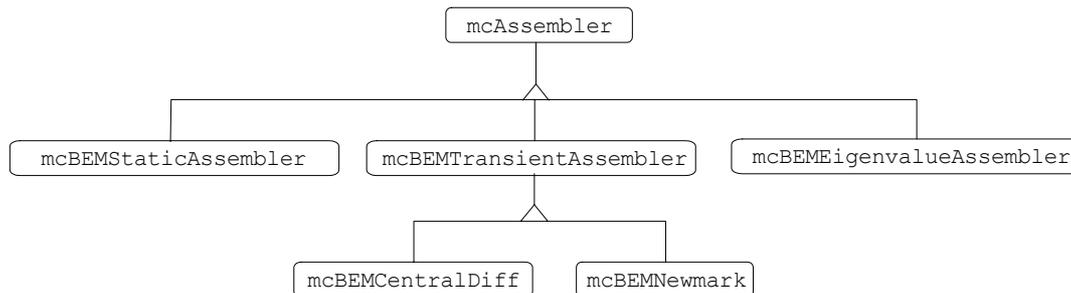


Figura 6-18: Hierarquia básica derivada da classe `mcAssembler`.

Classe `mcAssembler`

Objetos `mcAssembler` fornecem os métodos necessários para formar o sistema de equações. São responsáveis pelo acesso a cada elemento de contorno, célula de domínio ou elemento finito e adicionar suas contribuições ao sistema de equações global. A maior subclasse derivada desta correntemente implementada é a classe `mcBEMIncrementalAssembler`, da qual descendem a classes como `mcBEMStaticAssembler` (para problemas estáticos), `mcBEMTransientAssembler` (para problemas transientes) e `mcBEMEigenvalueAssembler` (para problemas de autovalores/autovetores). A figura 6-18 mostra a hierarquia básica.

Classes similares podem ser implementadas para elementos finitos. No caso do MEC, a solução numérica é normalmente dividida em um problema de valores no contorno e eventualmente um problema de domínio associado. Como mesmo em problemas de grandes deslocamentos é possível incluir os termos não-lineares nos carregamentos, os métodos mais importantes das classes `mcBEMStaticAssembler` são naturalmente os que disparam a montagem da matriz \mathbf{A} e do vetor \mathbf{b} de (6.1). Os métodos responsáveis por esta tarefa, bem como outros métodos comuns à estas classes estão apresentados a seguir.

► Método `formBVPTangent`: Responsável pela montagem do lado esquerdo do sistema de equações do problema de valores no contorno. No MEC, isto normalmente é realizado aplicado-se as condições de contorno e separando-se as contribuições das matrizes \mathbf{H} e \mathbf{G} associadas às variáveis prescritas e não-prescritas. Note-se que, ao contrário do que normalmente ocorre com o MEF, uma parte do lado direito do sistema já é obtida durante esta operação. Sua definição é:

```
int formBVPTangent();
```

► Método `formBVPUnbalance`: Aqui o lado direito do sistema é completado com as contribuições de domínio, como os carregamentos internos e termos não lineares. Em problemas de valores no contorno puros, este método não precisa ser utilizado.

```
int formBVPUnbalance();
```

► Métodos `formDVPTangent` e `formDVPUnbalance`: Não foram utilizados neste trabalho, mas foram incluídos na implementação para permitir a montagem de sistemas de equações associados aos problemas de valores no domínio.

```
int formDVPTangent(mcBESubregion& sr);  
int formDVPUnbalance(mcBESubregion& sr);
```

► Métodos `updateBVPDOF` e `updateBVPDOFIncr`: Utilizados para atualizar os valores dos graus de liberdade dos nós do contorno. Duas definições são possíveis:

```
int updateBVPDOF(Vector& X, double fact);  
int updateBVPDOFIncr(Vector& X, double fact);
```

onde `X` é um vetor contendo os novos valores e `fact` é um fator que permite relaxação na atualização destes valores. Na segunda definição, `X` contém somente os incrementos das variáveis, de modo que neste caso os valores antigos não são substituídos, mas adicionados aos incrementos.

► Métodos `updateDVPDOF` e `updateDVPDOFIncr`: Similares à `updateBVPDOF` e `updateBVPDOFIncr` mas operam sobre os graus de liberdade dos nós do(s) domínio(s). Duas definições são possíveis:

```
int updateDVPDOF(mcBESubregion& sr, char* id, Vector& X, double fact);  
int updateDVPIncr(mcBESubregion& sr, char* id, Vector& X, double fact);
```

onde `sr` é a subregião a ser atualizada e `id` indica qual conjunto de graus de liberdade serão modificados.

► Método `evaluateInternalPoints`: Realiza o cálculo das variáveis em pontos internos. Este método é invocado como uma etapa de pós-processamento do problema. No MEC, isto não implica em montagem de um sistema.

```
int evaluateInternalPoints();
```

► Método `evaluateGradients`: Realiza o cálculo dos gradientes das variáveis em pontos internos. Este método também é invocado como uma etapa de pós-processamento do problema e não implica em montagem de um sistema.

```
int evaluateGradients();
```

O código básico dos métodos `formBVPTangent` e `formBVPUnbalance` estão ilustrados nas figuras (6-19) e (6-20), respectivamente, conforme implementados para a classe `mcBEMStaticAssembler`.

```
int mcBEMStaticAssembler::formBVPTangent() {  
  
    theSOE->reset(theConstraint->getNumberOfBVPDOFs());  
    while (SR) { // Loop over the subregions:  
        mcBElementITR BE (theModel->getBEElementsSharing(*SR));  
        mcNodeITR ND (theModel->getBVPCollocationPointsFor(*SR));  
        while (BE) { // Loop over the elements:  
            while (ND) { // Loop over the collocation nodes:  
                SR->getFundSolution().setLoadPoint(coor);  
                sing_node = theModel->detectSingularNode(*SR,*BE,*ND);  
                BE->formMatrix(H,"T",sing_node);  
                BE->formMatrix(G,"U",sing_node);  
                theConstraint->applyBC(*BE,*ND,H,G);  
                ND++;  
            }  
            BE++;  
        }  
        SR++;  
    }  
  
    return 0;  
}
```

Figura 6-19: Código simplificado de `formBVPTangent` para problemas lineares.

Deve-se destacar que o código de `formBVPTangent` é essencialmente o mesmo para problemas lineares e não-lineares, uma vez que os termos não-lineares são manipulados pelo método `formBVPUnbalance`.

```

int mcBEMStaticAssembler::formBVPUnbalance() {

while (SR) { // Loop over the subregions:
mcDCellITR DC (theModel->getDCCellsSharing(*SR));
mcNodeITR nod_itr(theModel->getBVPCollocationPointsFor(*SR));
while (DC) { // Loop over the cells:
while (ND) { // Loop over the collocation nodes:
SR->getFundSolution().setLoadPoint(coor);
sing_node = theModel->detectSingularNode(*SR,*DC,*ND);
load = theModel->getNodalLoadsFor(*DC,mcLoad::SURFACELOAD);
DC->formVector(B,"V",load,sing_node);
map = theConstraint->mapCollocationDOF(*DC,*ND);
theSOE->addB(B, map);
ND++;
}
DC++;
}
SR ++;
}

return 0;

}

```

Figura 6-20: Código simplificado de `formBVPUnbalance` para problemas lineares.

Classe `mcConstraintHandler`

A superclasse `mcConstraintHandler` implementa os métodos usuais para aplicação de restrições ao sistema de equações. Deslocamentos e trações prescritas são manuseadas por estes objetos. Em particular, a imposição das condições de compatibilidade sobre uma interface compartilhada por duas subregiões também é realizada através de objetos `mcConstraintHandler`. Outros exemplos de tarefas comuns realizadas aqui são a numeração dos graus de liberdade e o uso de multiplicadores de Lagrange. No caso do MEC, um objeto `mcConstraintHandler` também é responsável pela geração automática dos nós de elementos de contorno e células de domínio. Dada a natureza específica das equações algébricas resultantes dos diversos métodos, sugere-se uma derivação de `mcConstraintHandler` para cada método empregado. No presente trabalho, foi derivada uma classe específica para o MEC, como ilustrado na figura 6-21.

A classe-mãe `mcConstraintHandler` conta com apenas um método virtual denominado `applyConstraints`, que pode ser adaptado para uma série de tarefas. No caso da subclasse `mcBEMConstraintHandler`, incorporou-se algumas funções de uso específico

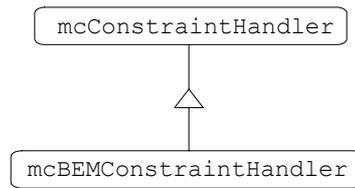


Figura 6-21: Hierarquia básica derivada da classe `mcConstraintHandler` para o MEC.

para o MEC, sintetizadas a seguir.

► Método `numberDOF`: Este método deve ser invocado antes de cada reanálise. É responsável pela numeração dos graus de liberdade.

```
int numberDOF();
```

► Métodos `createBVPDOFs` e `createDVPDOFs`: São métodos virtuais protegidos invocados por `numberDOF`. Quando um elemento de contorno é criado `mcDomain`, apenas sua geometria definida. Os nós correspondentes são criados antes de cada reanálise, quando necessário. Esta abordagem foi projetada para comportar alteração do tipo de elemento entre análises, com vistas a procedimentos adaptativos.

```
void createBVPDOFs();
void createDVPDOFs();
```

► Método `applyBC`: Realiza a separação das matrizes **H** e **G** de acordo com as condições de contorno, levando em conta a possível existência de mais de uma subregião. As matrizes e vetores resultantes são automaticamente superpostas às matrizes globais, dispensando a necessidade do usuário realizar esta operação. Sua definição mais simples é:

```
void applyBC(mcBElement& be, mcNode& node, Matrix& H, Matrix& G,
             double factor, double time);
```

onde `be` e `node` são o elemento que está sendo integrado e o ponto de colocação, respectivamente. As submatrizes elementares são os argumentos `H` e `G`, enquanto `factor` é um fator de multiplicação e `time` é usado para aplicar condições de contorno dinâmicas.

► Método `createDefaultBC`: Este método foi idealizado para simplificar a imposição de condições de contorno. Na maioria dos problemas práticos, apenas uma parcela do contorno está submetida a valores prescritos, estando o restante *livre* (isto é, as variáveis duais são homogêneas e as primais são incógnitas do problema). Quando um elemento de contorno não possui nenhuma condição de contorno tacitamente especificada, assume-se que este esteja livre e uma condição de contorno *default* é utilizada. Esta condição de contorno é sempre compatível com a solução fundamental do problema. A interface é dada por:

```
mcBoundaryCondition createDefaultBC(mcBElement& be, int local);
```

► Métodos `mapCollocationDOF` e `mapDOF`: Esta família de métodos é utilizada para mapear a numeração dos graus de liberdade de um nó, elemento ou célula nas posições correspondentes do sistema linear. As definições de alguns exemplos são:

```
mcArray mapCollocationDOF(mcBElement& be, mcNode& node, char* tag);  
mcArray mapDOF(mcNode& node, char* tag);
```

onde `be` e `node` são o elemento que está sendo integrado e o ponto de colocação, respectivamente. A segunda definição é utilizada para se mapear o número das equações de um único nó. O identificador `tag` é usado para indicar se os graus de liberdade mapeados são primais ou duais.

Classe `mcModelHandler`

Os objetos derivados desta classe são responsáveis por prover acesso aos objetos armazenados em `mcDomain` durante a fase de solução, de forma que nenhum objeto da agregação `mcAnalysis` precise fazê-lo diretamente. Portanto, os objetos armazenados nas listas de `mcDomain` não são visíveis aos demais objetos de análise. Os objetos podem ser acessados individualmente ou em listas, através de iteradores. A figura 6-22 ilustra a hierarquia básica utilizado no presente trabalho.

A subclasse `mcBEMModelHandler` dispõe de um grande número métodos especialmente voltados para o MEC, projetados para simplificar a implementação do método. A seguir estão apresentadas as definições de alguns deles.

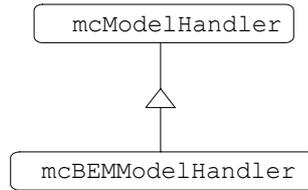


Figura 6-22: Hierarquia básica derivada da classe `mcModelHandler` para o MEC.

► Método `applyLoadToDomain`: Este método provoca a aplicação dos carregamentos e condições de contorno do caso de carregamento corrente sobre o modelo computacional. No caso dinâmico, as cargas já são aplicadas de acordo com a variável temporal.

```
void applyLoadToDomain(double time, double fact)
```

► Método `getNumberOfBVPDOFsFor`: Calcula o número de graus de liberdade do problema de valores no contorno para uma dada subregião:

```
int getNumberOfBVPDOFsFor(mcBESubregion& sr);
```

► Método `getBElementsSharing`: Retorna uma lista contendo os elementos de contorno que compartilham uma entidade específica. Nas versões exemplificadas a seguir, o método retorna os elementos que compartilham um nó `nd` e uma subregião `sr`:

```
mcList getBElementsSharing(mcNode& nd);
mcList getBElementsSharing(mcBESubregion& sr);
```

Existem métodos similares para células de domínio.

► Método `getBVPCollocationPointsFor`: Retorna uma lista de pontos de colocação para o problema de valores no contorno da subregião `sr`:

```
mcList getBVPCollocationPointsFor(mcBESubregion& sr);
```

Este método pode ser utilizado para realizar a colocação em quaisquer conjuntos de pontos.

► Método `getBVPCollocationNodesFor`: Similar ao anterior, mas onde os pontos de colocação são necessariamente os nós do contorno da subregião `sr`:

```
mcList getBVPCollocationNodesFor(mcBESubregion& sr);
```

► Método `detectSingularNode`: Realiza a verificação da singularidade sobre um elemento de contorno ou célula de domínio, com respeito a um ponto de colocação `pt`:

```
int detectSingularNode(mcBESubregion& sr, mcBElement& be, mcPoint& pt);  
int detectSingularNode(mcBESubregion& sr, mcDCell& dc, mcPoint& pt);
```

► Método `getSolidAngle`: Retorna um vetor contendo as componentes do ângulo sólido de um nó. Estes valores são utilizados para cálculo dos fatores geométricos C_{ij} :

```
Vector getSolidAngle(mcNode& nd);
```

A versatilidade provida por estes métodos é muito superior aos métodos de programação convencional. Isto pode ser observado na figura (6-19): para cada subregião calculada, a seleção de quais elementos de contorno seriam integrados é realizada em uma única instrução. Naquele código, um objeto `mcBEMModelHandler` fornece os pontos de colocação e verifica a singularidade. No caso de mais de uma subregião estas operações normalmente envolvem várias linhas de programação estruturada. O uso combinado destes métodos com iteradores automatiza significativamente o trabalho de programação. Por exemplo, seja uma situação em que o analista deseja imprimir os graus de liberdade primais de todos os pontos de colocação das subregiões denominadas `Aluminio` ou `Aco`. O seguinte fragmento de código realizaria a tarefa:

```
lista1 = theModel->getBVPCollocationPointsFor('Aluminio');  
lista2 = theModel->getBVPCollocationPointsFor('Aco');  
mcNodeITR ND (lista1 || lista2);  
for (ND.first(); ND; ND++) {  
    print << ND.current().getDOFs('P');  
}
```

Note-se que nenhum objeto `mcDomain` precisa se invocado. Ou seja, não há necessidade do usuário manipular diretamente as classes de armazenamento. Este é um dos objetivos propostos por esta arquitetura.

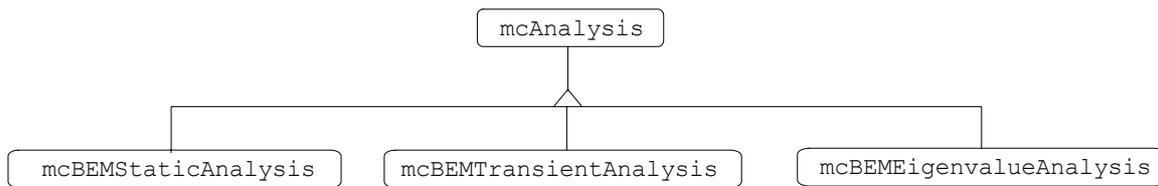


Figura 6-23: Os dois principais níveis de hierarquia derivados da classe `mcAnalysis`.

Classe `mcAnalysis`

Esta é a classe-mãe principal de uma análise. Um objeto `mcAnalysis` é uma agregação de classes responsáveis por diversas etapas da solução. Embora não realize explicitamente qualquer cálculo, a classe `mcAnalysis` é que define propriamente qual problema será resolvido e de que maneira. Também é responsável por verificar a validade das classes na agregação para que esta tenha sentido físico. Um único método virtual chamado `analyze()` dispara a execução da análise. Os objetos `mcAnalysis` também sabem se um domínio foi alterado e é necessária uma nova análise (como ocorre em problemas adaptativos ou não-lineares). Exemplos de classes derivadas desta são as classes `mcBEMStaticAnalysis` e `mcFEMEigenAnalysis`, que implementam uma solução estática pelo MEC e uma solução modal pelo MEF, respectivamente (figura 6-23).

Para ilustrar o uso da arquitetura proposta, deve-se primeiramente esclarecer o relacionamento entre as superclasses mais importantes. Embora seja possível utilizar muitas das classes implementadas em qualquer outro programa como ferramentas adicionais independentes, as vantagens da arquitetura proposta ficam evidentes quando todas as superclasses são conectadas para gerar um aplicativo específico. A figura 6-24 ilustra de forma simplificada como os principais componentes são montados para gerar um programa completo. Embora outras superclasses possam ser criadas para tipos especiais de análise, este conjunto se mostra bastante versátil e suficiente na maioria dos casos.

6.4.4 Exemplos de programação

A arquitetura proposta pode ser utilizada como um ambiente de programação dotado de funções de alto nível. Uma grande parte dos construtores das classes desenvolvidas foi bastante sobrecarregada, permitindo que o usuário programe sua análise de forma muito

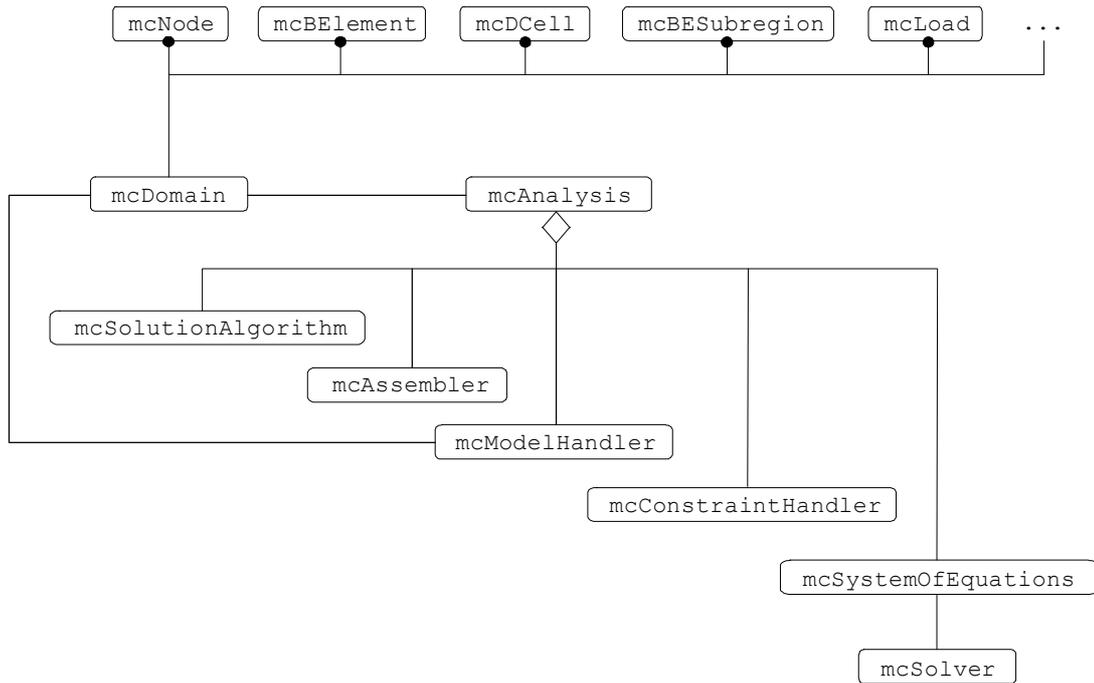


Figura 6-24: Agregação básica necessária para um programa.

simples e intuitiva. Exemplos de tarefas que normalmente demandam uma programação relativamente mais trabalhosa são a definição de grupos (*patches*) e a definição de interfaces em subregiões. A título de exemplo, suponha-se que o analista esteja interessado em resolver um problema de estado plano de tensões como o ilustrado na figura 6-25. O problema é dividido em duas subregiões Ω_1 e Ω_2 , cada uma composta por materiais de propriedades diferentes. A parcela Γ_{12} de Ω_1 é composta pelos elementos e_1 e e_2 , enquanto a parcela Γ_{21} de Ω_2 é formada pelos elementos e_3 e e_4 .

A geração do modelo computacional deste problema dentro da estrutura proposta se torna simples em muitos aspectos. A definição dos materiais e das subregiões é realizada por duas instruções simples:

```

mcIsoMaterial  mat_1(E1,nu1), mat_2(E2,nu2);
mcElast2D      omega_1(mat_1), omega_2(mat_2);
  
```

Conforme já mencionado, os objetos `omega_1` e `omega_2` conhecem suas soluções fundamentais e seus graus de liberdade (u_1 e u_2). E a definição da interface compartilhada

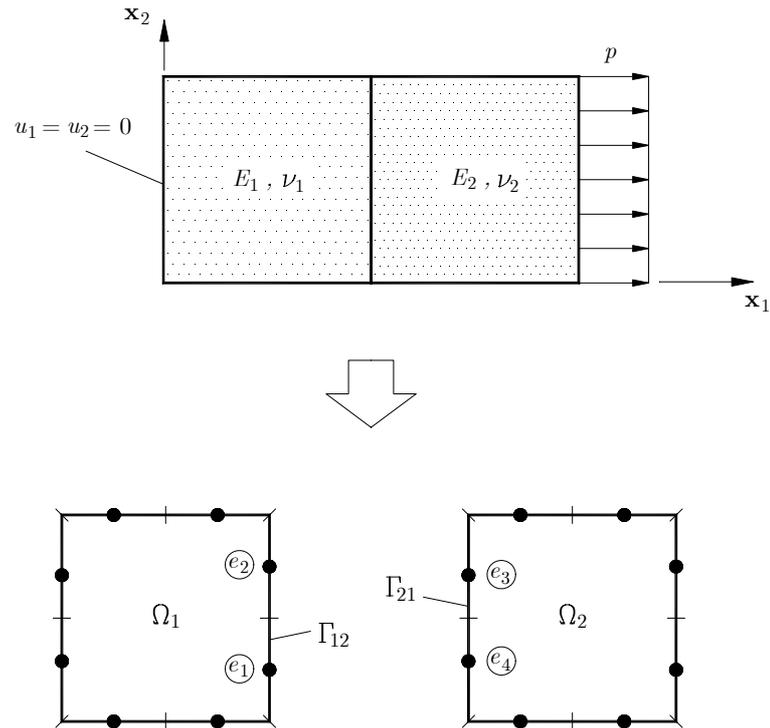


Figura 6-25: Um exemplo simples de uso de subregiões.

pelas duas subregiões pode ser definida com o seguinte código:

```

mcPatch      gama_12(e1,e2), gama_21(e3,e4);
mcInterface  interface(gama_12, gama21);

```

Este fragmento de código é suficiente para definir os materiais, as subregiões e impor as condições de compatibilidade sobre Γ_{12} e Γ_{21} , de forma que os deslocamentos sejam os mesmos e as trações sejam opostas.

Um outro ponto de interesse que deve ser destacado é a forma como as condições de compatibilidade são manipuladas. Neste trabalho, todos os graus de liberdade (`mcDOF`) são identificados por um código e um nome (`mcID`). Se dois nós pertencentes a subregiões diferentes são interfaceados, o programa compara ambos os nomes e códigos. Se estes dados coincidem, são impostos os mesmos valores para as variáveis primais e valores opostos para as variáveis duais. Este procedimento é interessante porque permite ao usuário unir duas subregiões governadas por equações diferenciais diferentes, desde que

estas possuam ao menos um grau de liberdade em comum (embora isso não seja, por si só, suficiente para garantir a unicidade da solução).

Finalmente, vale ressaltar a possibilidade de uso das estruturas aqui apresentadas para o MEF. Devido à similaridade entre diversos componentes usados para o MEF e para o MEC, a adaptação da estrutura aqui proposta é muito simples. A figura 6-26 ilustra um fragmento de código com nove linhas de programação onde as matrizes de rigidez e massa de uma malha de elementos finitos são montadas. Neste exemplo, um único objeto `mcQuadrature` foi utilizado para integrar as matrizes elementares `Ke` e `Me` usando uma regra 2×2 , e um iterador `fe` se encarrega de percorrer os elementos desejados. Note-se que a superposição das matrizes é automatizada com o uso de membros que mapeiam os números dos graus de liberdade do elemento finito.

```
// Matrix integrator - 2x2 Gauss rule:
mcQuadrature<Matrix> Mat(2,2,1);

// Get the finite elements iterator:
fe = ModelHandler->getFElementsUsingMaterial("Steel");

// Loop over the elements:
while (fe) {

    // Evaluate the stiffness and mass matrices:
    Ke = Mat.NIntegrate(fe.current->Stiffness);
    Me = Mat.NIntegrate(fe.current->Mass);

    // Assemble:
    SysOfEq->AssembleK(Ke, ModelHandler->mapDOF(fe.current) );
    SysOfEq->AssembleM(Me, ModelHandler->mapDOF(fe.current) );

    fe++;
}
```

Figura 6-26: Exemplo de código para montagem das matrizes de rigidez e massa em um programa de elementos finitos.

6.4.5 Extensibilidade e reusabilidade

A característica que deve ser preservada para o sucesso da arquitetura aqui proposta está baseada na premissa do usuário ter que programar poucas linhas para adaptar o código para uma dada aplicação. Isto garante a reusabilidade dos componentes e a extensibilidade da estrutura básica de um programa. Por exemplo, se o problema ilustrado na figura 6-25 deve ser investigado para uma análise linear elástica, então o fragmento de código

ilustrado na figura 6-27 poderia ser empregado.

```
1. mcDomain domain;
2. domain.readInputFile("problema2D.dat");
3. domain.setCurrentLoadCase(domain.getDefaultLoadCase());
4. mcBEMLinear solAlgo;
5. mcBEMStaticAssembler assembler;
6. mcBEMModelHandler model;
7. mcBEMConstraintHandler constraint;
8. mcDenseNonSymLS solver;
9. mcDenseNonSymLSOE SOE(solver);
10. mcBEMStaticAnalysis analysis(domain, solAlgo, assembler,
                                model, constraint, SOE);
11. analysis.analyse();
```

Figura 6-27: Exemplo de código para análise de um problema linear estático através do MEC.

Note-se que a solução de um problema de transferência de calor, por exemplo, utilizaria a mesma estrutura (as diferenças inerentes estão implícitas nos objetos subregião).

Agora, se este mesmo exemplo deve ser resolvido para uma análise dinâmica transiente, seria necessário apenas alterar as linhas 4, 5 e 10 do código acima de acordo com a figura 6-28.

```
1. mcDomain domain;
2. domain.readInputFile("problema2D.dat");
3. domain.setCurrentLoadCase(domain.getDefaultLoadCase());
4. mcBEMTransientSolAlgo solAlgo;
5. mcBEMNewmark assembler;
6. mcBEMModelHandler model;
7. mcBEMConstraintHandler constraint;
8. mcDenseNonSymLS solver;
9. mcDenseNonSymLSOE SOE(solver);
10. mcBEMTransientAnalysis analysis(domain, solAlgo, assembler,
                                    model, constraint, SOE);
11. analysis.analyse();
```

Figura 6-28: Exemplo de código para análise de um problema dinâmico transiente através do MEC.

6.5 Resumo do Capítulo 6

Este capítulo fez uma descrição resumida dos conceitos envolvidos em POO bem como sua aplicação geral em programas de elementos finitos e, de forma mais incomum, de elementos de contorno. Algumas das vantagens do uso de POO são superficialmente discutidas. São identificadas três categorias básicas que englobam boa parte das classes tipicamente utilizadas em programas de mecânica computacional.

Uma revisão bibliográfica ilustra o grande universo de possibilidades para implementação de filosofias OO utilizadas no MEF e no MEC. Alguns exemplos interessantes de arquiteturas são explorados um pouco mais, fornecendo idéias para o desenvolvimento de um conjunto de classes específico para programas de elementos de contorno.

Finalmente, apresenta-se a arquitetura básica da biblioteca `mcBEM`, desenvolvida para manipular objetos tradicionalmente encontrados em programas de elementos de contorno, e que foi efetivamente utilizada na implementação computacional do presente trabalho. Um conjunto razoavelmente completo de classes auxiliares, de modelo e de análise é apresentado. Tarefas de programação mais elaboradas foram encapsuladas em métodos que simplificam sobremaneira o desenvolvimento e extensão de aplicações para o MEC e o MEF. A funcionalidade atingida com as idéias aqui propostas é marcante, como foi demonstrado através de exemplos.

“...Se você escolher as classes erradas, elas lhe dirão... Acerte nas classes e você saberá disso com a mesma rapidez... As nuvens irão embora e o arco-íris brilhará. Sua vida sexual vai melhorar e sua calvície prematura vai reverter, como por um passe de mágica.”⁵

⁵Cf. S. R. Davis: *C++ Para Leigos*, Berkeley Brasil Editora, 1994.

Capítulo 7

Resultados Numéricos

Este capítulo apresenta alguns resultados numéricos obtidos a partir da implementação da formulação apresentada no Capítulo 5. Todos os resultados aqui ilustrados foram gerados através de um programa de computador baseado na estrutura orientada a objetos descrita no Capítulo 6.

Os resultados numéricos apresentados a seguir têm como objetivos básicos analisar o desempenho da formulação apresentada para duas classes de problemas:

a.) Problemas lineares: são obtidas as curvas de convergência para os elementos constante, linear e quadrático. Vale lembrar a escassez deste tipo de informação na literatura do MEC. A maior parte dos casos analisados utilizaram como parâmetro de comparação o deslocamento máximo da placa, já que nos métodos integrais as taxas de convergência para esforços normalmente são idênticas às dos deslocamentos.

b.) Problemas não-lineares: foram levantadas as trajetórias de equilíbrio primárias para diversos casos de problemas geometricamente não-lineares. Foi dada maior ênfase a placas moderadamente espessas, pois estes casos não possuem solução analítica.

Em todos os casos, foram utilizados 10 pontos de integração para as integrais regulares, fracamente singulares e quase-singulares, enquanto empregou-se 24 pontos para as integrais fortemente singulares.

7.1 Problemas lineares

Conforme citado no Capítulo 5, o procedimento de integração numérica empregado para cômputo das integrais singulares é bem documentado na literatura. No entanto, o efeito desse procedimento sobre a precisão de soluções de problemas específicos em comparação com outros métodos de integração é escassamente reportado. Assim, esta seção ilustra os resultados obtidos para alguns problemas lineares a fim de comparar o desempenho da formulação proposta com outras abordagens bem como validar os procedimentos empregados para uso nas aplicações não-lineares.

7.1.1 Elasticidade 2D

Embora a solução de problemas lineares de elasticidade bidimensional não seja o principal objetivo do presente trabalho, a qualidade desta terá uma forte influência na solução dos problemas não-lineares tendo em vista a existência do acoplamento flexão-membrana. Um único teste clássico será abordado aqui para ilustrar o desempenho do método direto na solução destes problemas. Trata-se de uma placa com furo sob tração, em regime de estado plano de tensões, conforme ilustrado na figura 7-1. Dada a simetria do problema, apenas um quarto da geometria foi discretizado, utilizando elementos constantes, lineares e quadráticos. Para as malhas de elementos lineares e quadráticos, dois valores de recuo dos nós da extremidade foram testados: 15% e 10% do comprimento do elemento. A figura 7-1 também ilustra a malha de elementos lineares com recuo de 15%. O valor do fator de concentração de tensões (κ_t) foi obtido com base nas tensões σ_{xx} calculadas sobre os elementos ao longo da linha $x = 0$, para um carregamento σ_0 unitário. Devido à descontinuidade dos elementos empregados, o valor de σ_{xx} na posição $(x, y) = (0, 10)$ foi extrapolado utilizando as próprias funções de interpolação.

As figuras 7-3 e 7-2 mostram as distribuições de tensões σ_{xx} para elementos com recuo de 15% e 10% respectivamente. Os valores obtidos com elementos constantes também estão ilustrados para fins de comparação. Este problema também foi analisado por Guigiani & Casalini (1987) utilizando uma malha de 20 elementos constantes integrados pelo método direto. A tabela 7.1 compara os resultados obtidos com essas soluções.

Note-se que a performance do elemento quadrático mostra certa dependência do valor

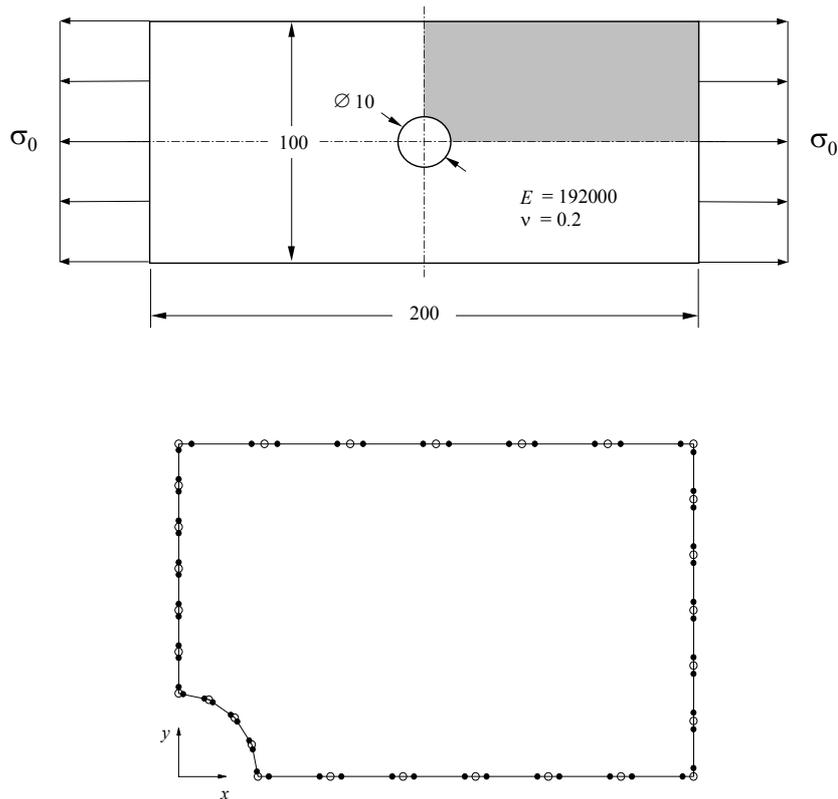


Figura 7-1: Placa com furo sob tração uniaxial. Malha de elementos lineares com 15% de recuo nos nós.

do recuo, enquanto o elemento linear mostrou-se pouco sensível aos valores empregados.

7.1.2 Flexão de placas

Existe um bom número de abordagens do MEC para problemas de flexão linear de placas de Mindlin e de Reissner, conforme ilustra a revisão bibliográfica do Capítulo 1. No entanto, não são muitas as publicações que exploram placas com espessura

	κ_t	erro %
Elemento linear - recuo de 15%	3.11	-0.9
Elemento linear - recuo de 10%	3.11	-0.9
Elemento quadrático - recuo de 15%	3.23	2.9
Elemento quadrático - recuo de 10%	3.12	-0.6
Guiggiani & Casalini (1987) - método direto	3.09	-1.6
Guiggiani & Casalini (1987) - mov. corpo rígido	2.89	-8.0
Solução analítica (Guiggiani & Casalini 1987)	3.14	

Tabela 7.1: Resultados de κ_t para placa com furo.

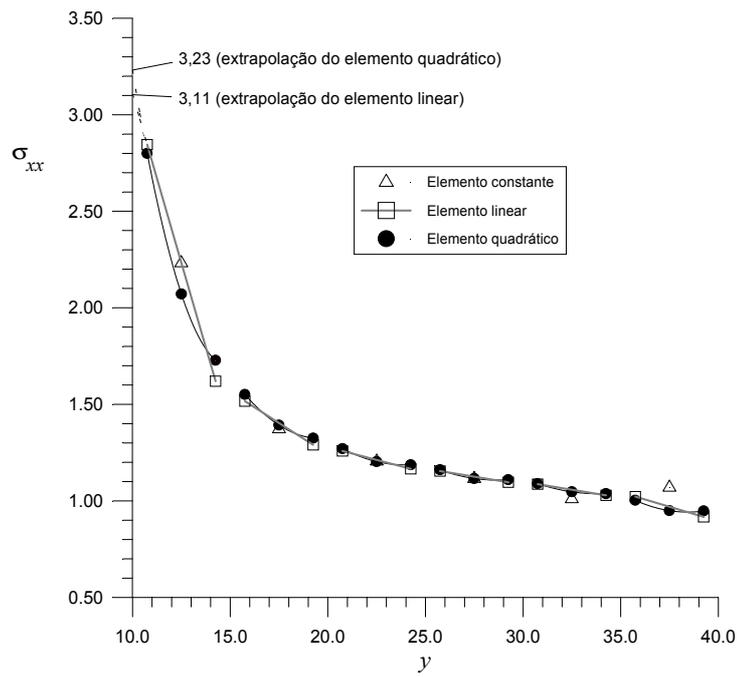


Figura 7-2: Tensões σ_{xx} ao longo da linha $x = 0$ para elementos lineares e quadráticos com recuo de 15%.

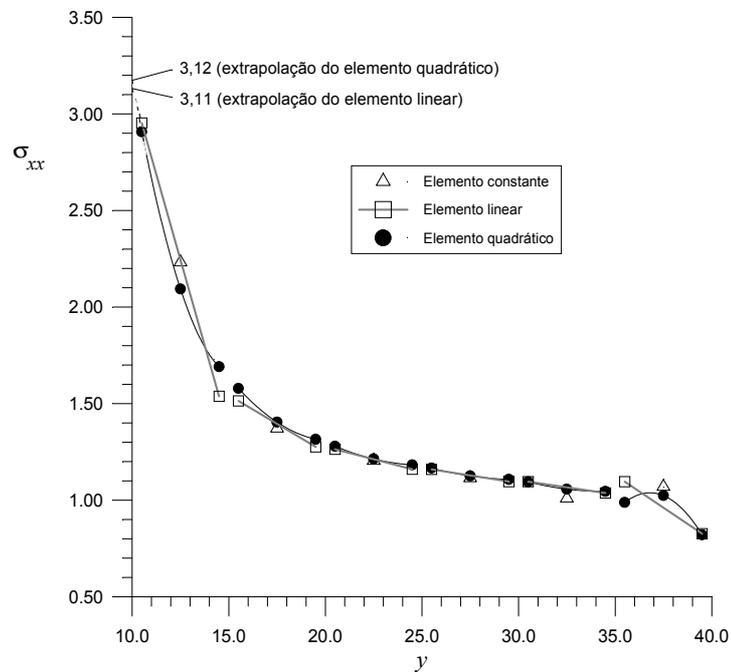


Figura 7-3: Tensões σ_{xx} ao longo da linha $x = 0$ para elementos lineares e quadráticos com recuo de 10%.

moderadamente grande. Deve-se destacar a inexistência de soluções analíticas fechadas para placas apoiadas. Embora essas soluções existam para certas geometrias engastadas (Reissner 1945), nestes casos os termos fortemente singulares são anulados e portanto não podem ser utilizados para validar os procedimentos de integração empregados neste trabalho. Adicionalmente, como os modelos de placa de Mindlin e de Reissner permitem a imposição de condições de contorno sobre três variáveis, dois tipos de condições de contorno de apoio tornam-se possíveis: apoio *hard* (onde são prescritos o deslocamento transversal, a rotação normal e o momento tangencial) e apoio *soft* (onde são prescritos o deslocamento e os momentos normal e tangencial). Com efeito, estas duas condições de contorno levam a resultados diversos e essa diferença se torna mais significativa à medida que a espessura da placa aumenta.

Esta seção explora basicamente os resultados obtidos com a presente formulação aplicada a flexão de placas quadradas de dimensão lateral a sob carregamento distribuído uniforme. Como se dispõe de resultados similares obtidos com imposição de movimento de corpo rígido e com a quadratura de Kutt (Marczak 1993), estes serão utilizados para comparação. A convergência dos resultados é verificada para malhas regulares com $N \times N$ elementos. As condições de contorno *hard* e *soft* serão aqui denominadas SS1 e SS2, respectivamente. A menos que especificado em contrário, os problemas de placas analisados são baseados no modelo de placa de Mindlin e utilizaram os seguintes dados:

$$\begin{aligned}
 E &= 3.0 \times 10^6 \\
 \nu &= 0.30 \\
 \kappa^2 &= \pi^2/12 \\
 q_3 &= 1000 \\
 m_f &= 0 \\
 a &= 1.0
 \end{aligned}$$

As figuras 7-4, 7-5 e 7-6 ilustram a convergência h para placas finas utilizando elementos constantes, lineares e quadráticos, respectivamente. Foi empregado apoio SS2 e uma razão $h/a = 0.015$. O deslocamento central da placa foi normalizado com a solução de

Navier para placas finas (Timoshenko & Woinowski-Krieger 1970):

$$w_n = \frac{u_{3\max}}{w_{\text{Navier}}}$$

Deve-se destacar que os resultados obtidos com imposição de movimento de corpo rígido e com a quadratura de Kutt utilizaram elementos contínuos. Adicionalmente, o número de pontos de integração utilizado para o método direto foi o mesmo em todos os elementos testados, de forma que não foi extraída a melhor performance possível do elemento quadrático.

A figura 7-4 mostra claramente que não há diferença significativa de desempenho dos elementos constantes integrados com quaisquer dos métodos. Por outro lado, a convergência do método direto nos elementos lineares mostrou-se melhor que a quadratura de Kutt (figura 7-5). Esta comparação não deve ser tomada em termos definitivos dado que os elementos utilizados neste trabalho são descontínuos, e portanto as pequenas diferenças percentuais de desempenho podem ser manipuladas com diversos valores de recuo para nós.

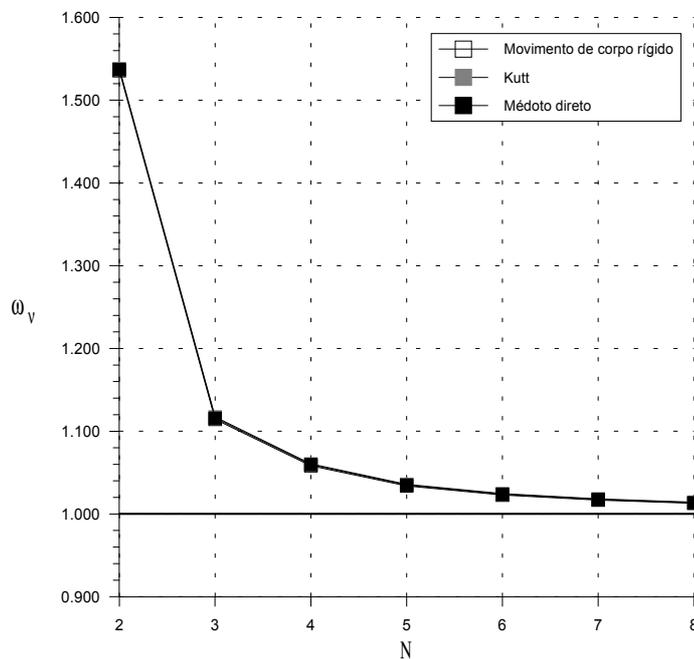


Figura 7-4: Convergência h para placa quadrada apoiada (SS2) sob carregamento uniforme - elemento constante.

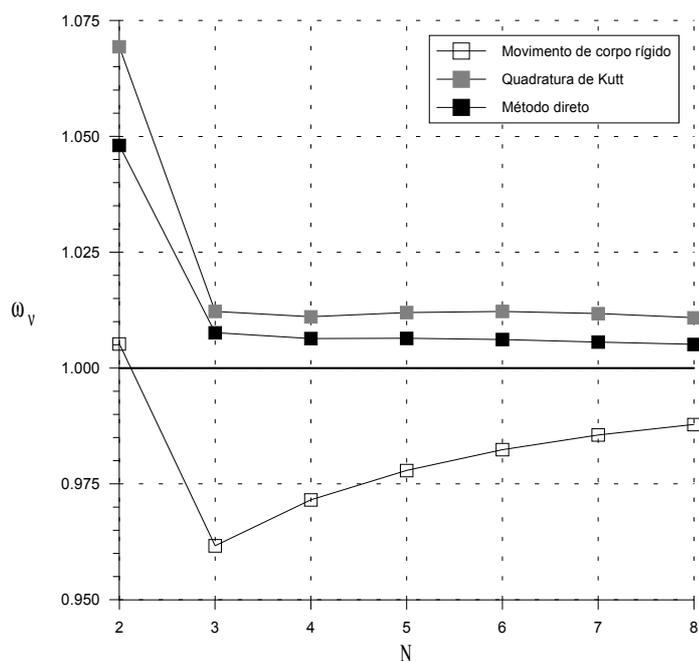


Figura 7-5: Convergência h para placa quadrada apoiada (SS2) sob carregamento uniforme - elemento linear.

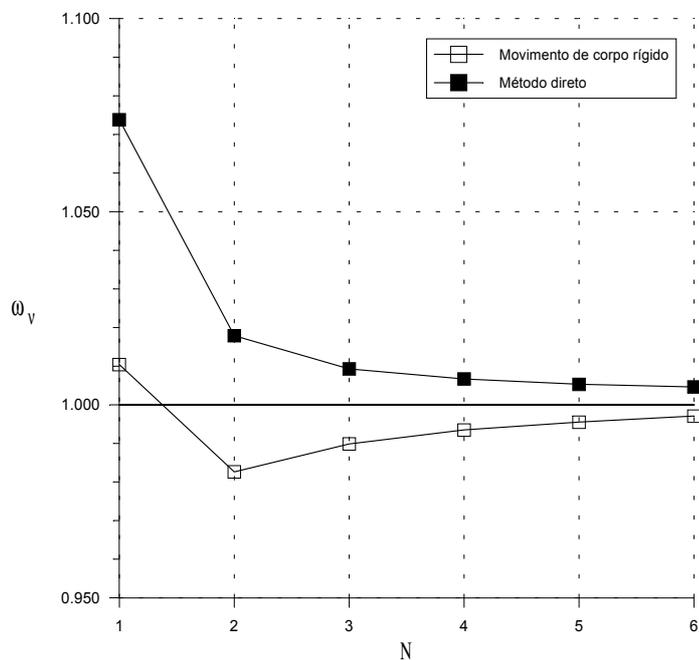


Figura 7-6: Convergência h para placa quadrada apoiada (SS2) sob carregamento uniforme - elemento quadrático.

O desempenho da presente formulação na análise de placas semi-espessas também foi verificado. Foram levantadas as curvas de convergência para placas quadradas apoiadas com razão h/a variando de 0.05 até 0.25. O deslocamento central da placa foi utilizado como parâmetro para comparação. Embora existam dezenas de elementos finitos baseados nos modelos de Mindlin e de Reissner, são poucas as soluções alternativas para as equações diferenciais destes modelos, tornando qualquer comparação dependente de aspectos numéricos como densidade de malha, padrão de convergência etc. Além disto, o número de soluções analíticas ou semi-analíticas para o modelo de placa de Mindlin é surpreendentemente pequeno em comparação com o modelo Reissner. Ainda persiste uma certa confusão entre as diferenças inerentes destes dois modelos de placa, e muitas vezes os nomes dessas teorias são empregados de forma equivocada (Wang et al. 2001).

Os resultados para placas com apoio SS2 estão ilustrados graficamente nas figuras 7-7 a 7-9, onde as soluções de Yuan & Miller (1988) foram utilizados como referência. A tabela 7.2 apresenta os valores de deslocamento adimensionalizados na forma:

$$w_n = u_{3\max} \frac{Eh^3}{q_0 a^4} \quad , \quad (7.1)$$

onde $u_{3\max}$ é o deslocamento transversal no centro da placa. Para fins de comparação estes resultados estão apresentados juntamente com os obtidos nos seguintes trabalhos:

- Srinivas & Rao (1973): Solução numérica aproximada das equações da elasticidade tridimensional.
- Yuan & Miller (1988): Elemento finito retangular baseado no modelo de Mindlin. Os resultados apresentados se referem a uma malha 6×6 . Este elemento utiliza uma interpolação bi-quadrática e é bem referenciado na literatura, razão pela qual esta solução escolhida para normalização nas figuras 7-7 a 7-9.
- Katsikadelis & Yotis (1993): Solução híbrida utilizando elementos de contorno e diferenças finitas para o modelo de Reissner. Os resultados ilustrados se referem a uma malha de 10×10 elementos constantes.
- Marczak (1993): Solução de elemento de contorno para o modelo de Mindlin. É similar ao presente trabalho mas adota elementos contínuos integrados pela técnica

	h/a				
	0.05	0.10	0.15	0.20	0.25
Yuan & Miller (1988)	0.04650	0.05019	0.05480	0.06018	0.06683
Srinivas & Rao (1973)	0.04677	0.05010	–	0.05912	–
Katsikadelis & Yotis (1993)	0.04538	0.04870	–	0.06099	0.06590
Marczak (1993) (linear)	0.04535	0.04882	0.05367	0.05954	0.06631
Marczak (1993) (quadrático)	0.04578	0.04942	0.05432	0.06014	0.06684
Elemento linear (6×6)	0.04688	0.05031	0.05420	0.06047	0.06704
Elemento quadrático (6×6)	0.04692	0.05047	0.05504	0.06057	0.06704

Tabela 7.2: Deslocamento central máximo para placa semi-espessa apoiada (SS2).

de imposição de movimento de corpo rígido. Os resultados para elementos lineares se referem a uma malha de 6×6 , enquanto os elementos quadráticos empregaram uma malha 3×3 .

Deve-se que todos estes trabalhos, mesmo os baseados no modelo de Mindlin, utilizaram um fator de correção das tensões cisalhantes $\kappa^2 = 5/6$.

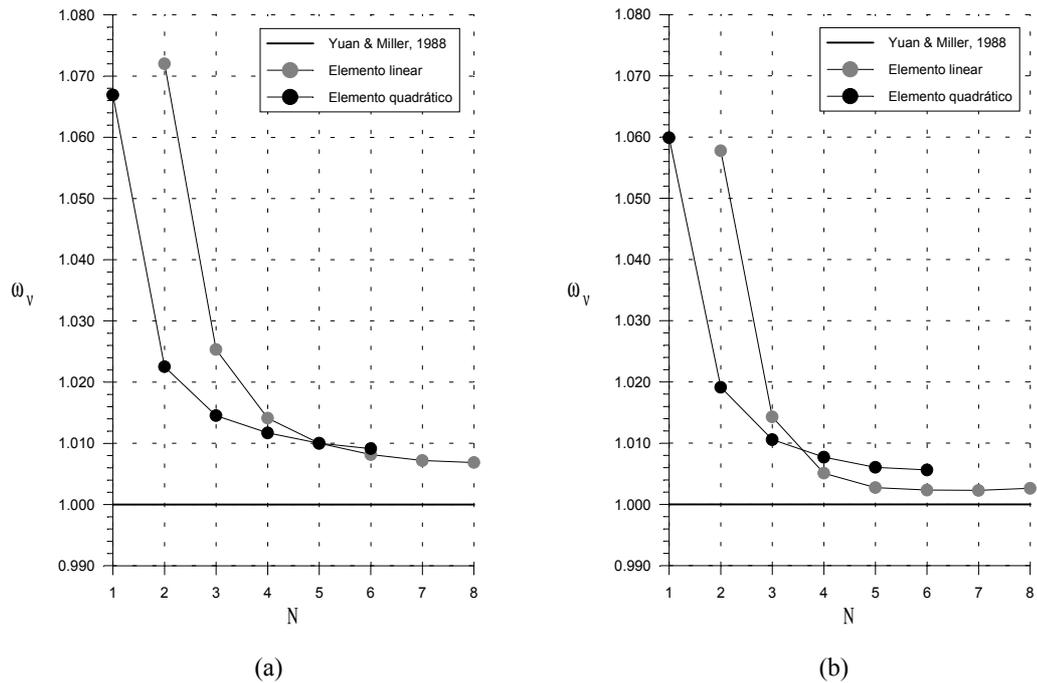


Figura 7-7: Convergência h para placa quadrada apoiada (SS2). (a) $h/a = 0.05$. (b) $h/a = 0.10$.

As curvas ilustradas nas figuras 7-7 a 7-9 apresentam um certo grau de não-monotonicidade, especialmente para o elemento linear. Foi verificado que as curvas referentes aos elementos

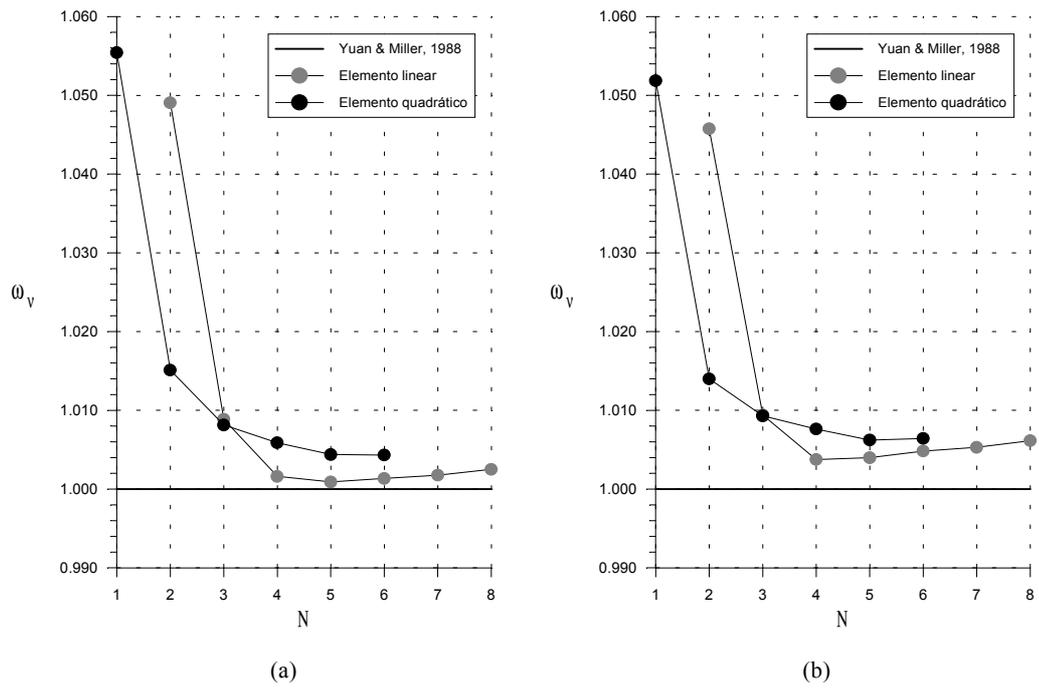


Figura 7-8: Convergência h para placa quadrada apoiada (SS2). (a) $h/a = 0.15$. (b) $h/a = 0.20$.

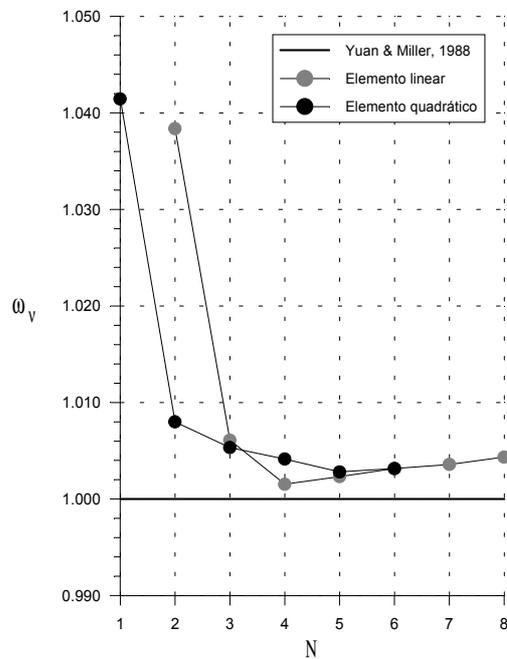


Figura 7-9: Convergência h para placa quadrada apoiada (SS2). $h/a = 0.25$.

lineares se equiparam às curvas dos elementos quadráticos para malhas 8×8 em diante. De qualquer maneira, nota-se claramente uma estabilização dos resultados com uma diferença pouco acima de 1% em relação à solução de referência adotada. Isto se deve aos diferentes fatores de correção das tensões cisalhantes, que repondem por boa parte das diferenças encontradas. A título de ilustração, empregando $\kappa^2 = 5/6$ para razão $h/a = 0,25$, o presente trabalho forneceu $w_n = 0.06671$ para o elemento linear e $w_n = 0.06682$ para o elemento quadrático, corroborando o argumento exposto.

As figuras 7-10 a 7-12 ilustram as curvas de convergência para placas com apoio SS1. Os valores foram normalizados com os resultados recentes de Lee et al. (2001), com exceção da figura 7-12, que foi normalizada com os resultados de Rao et al. (1974). Estes são baseados no modelo de Reissner, o que justifica as diferenças significativas ilustradas na figura 7-12. Este expediente foi adotado por não se dispor de uma solução mais moderna baseada no modelo de Mindlin para $h/a = 0,25$. A tabela 7.3 compara estes resultados adimensionalizados pela eq.(7.1) com os obtidos nos seguintes trabalhos:

- Salerno & Goldberg (1960), Lee et al. (2001): Soluções por séries para o modelo de Mindlin. A solução de Lee et al. (2001) é particularmente interessante por ser um dos raros trabalhos recentes a explorar o Modelo de Mindlin analiticamente.
- Craig (1987): Solução por diferenças finitas para o modelo de Reissner.
- Wang et al. (2001): Solução por séries para o modelo de Reissner. Este trabalho aplica a mesma abordagem empregada por Lee et al. (2001) para o modelo de Mindlin.
- Kant & Hinton (1980): Elemento finito para um modelo de placa de ordem superior.
- Rao et al. (1974): Solução de elementos finitos para o modelo de Reissner.
- Katsikadelis & Yotis (1993): Solução híbrida utilizando elementos de contorno e diferenças finitas para o modelo de Reissner. Os resultados ilustrados se referem a uma malha de 10×10 elementos constantes.
- Long et al. (1988): Solução de elementos de contorno, muito similar ao trabalho de van der Weeën (1982a).

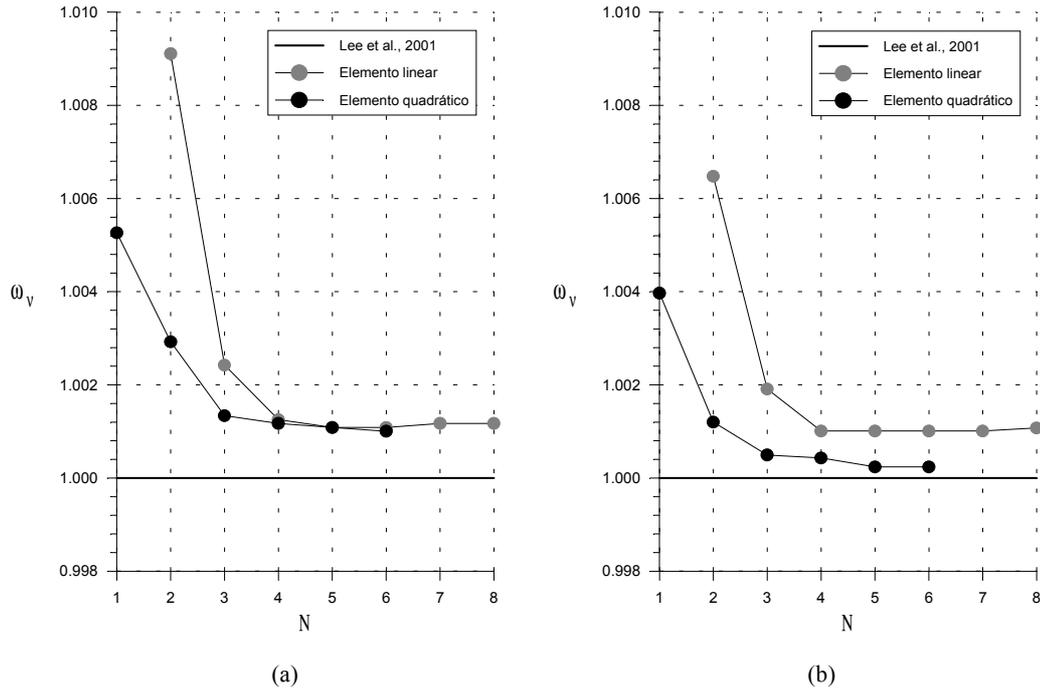


Figura 7-10: Convergência h para placa quadrada apoiada (SS1). (a) $h/a = 0.05$. (b) $h/a = 0.10$.

Aqui, também, todos os trabalhos utilizaram um fator de correção das tensões cisalhantes $\kappa^2 = 5/6$.

Note-se o bom desempenho da formulação proposta nestes casos. A curva da figura da 7-12 apresenta uma significativa discrepância com a solução de referência adotada, o que já era esperado dado que esta se refere ao modelo de Reissner.

Para verificar o desempenho da presente formulação para problemas compostos por subregiões, foi analisada uma placa retangular de razão $a/b = 2$ simplesmente apoiada sob carregamento uniforme unitário. Este caso foi analisado para duas situações: espessura constante e espessura variável, como ilustrado na figura 7-13. Em ambos os casos foram utilizadas duas subregiões, sendo que para espessura constante utilizou-se $h = 0,02$ enquanto que para espessura variável empregou-se $h = 0,02$ e $h = 0,015$ para cada subregião. A figura 7-13 também ilustra a malha analisada (foi utilizada apenas uma célula para cada subregião), onde se verifica que a interface das subregiões é formada por apenas quatro elementos de contorno.

Os deslocamentos transversais e as rotações foram calculados sobre pontos internos ao

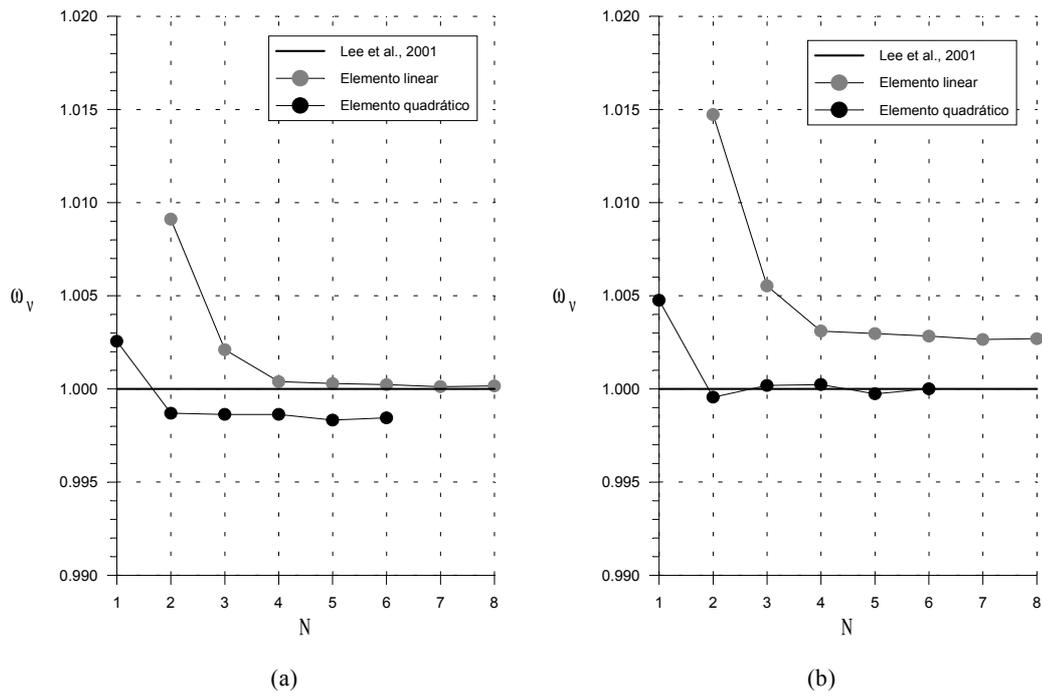


Figura 7-11: Convergência h para placa quadrada apoiada (SS1). (a) $h/a = 0.15$. (b) $h/a = 0.20$.

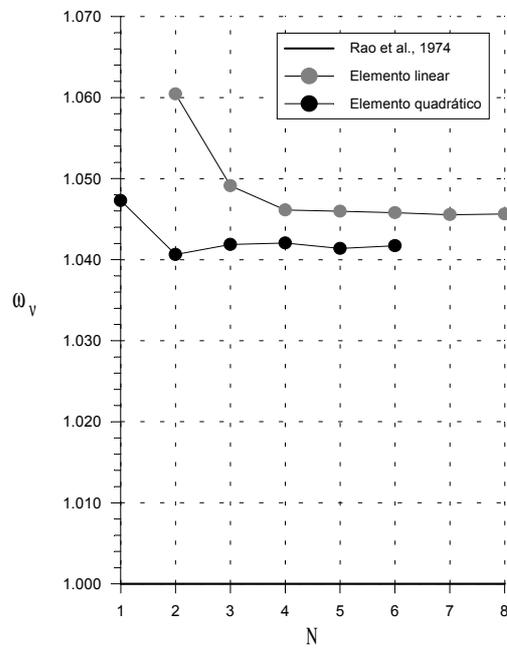


Figura 7-12: Convergência h para placa quadrada apoiada (SS1). $h/a = 0.25$.

	h/a				
	0.05	0.10	0.15	0.20	0.25
Salerno & Goldberg (1960)	0.04486	0.04632	0.04676	0.05360	0.05656
Rao et al. (1974)	0.04483	0.04627	0.04866	0.05201	0.05631
Wang et al. (2001)	0.04488	0.04630	0.04870	0.05220	–
Lee et al. (2001)	0.04488	0.04663	0.04958	0.05355	–
Kant & Hinton (1980)	–	0.04663	–	0.05351	–
Craig (1987)	0.04492	0.04683	0.05183	0.05353	0.06344
Katsikadelis & Yotis (1993)	0.04487	0.04634	–	0.05221	0.05660
Long et al. (1988)	0.04458	0.04612	–	0.05220	0.05666
Elemento linear (6×6)	0.04493	0.04668	0.04959	0.05366	0.05889
Elemento quadrático (6×6)	0.04492	0.04664	0.04950	0.05351	0.05866

Tabela 7.3: Deslocamento central máximo para placa semi-espessa apoiada (SS1).

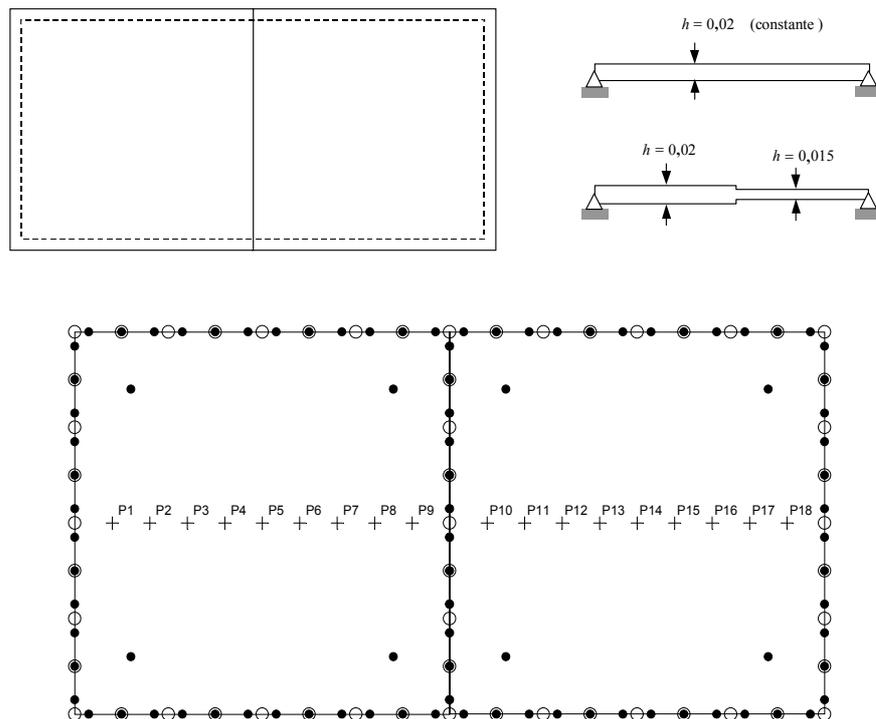


Figura 7-13: Placa retangular modelada com duas subregiões. Malha de elementos de contorno, células de domínio e pontos internos para análise linear.

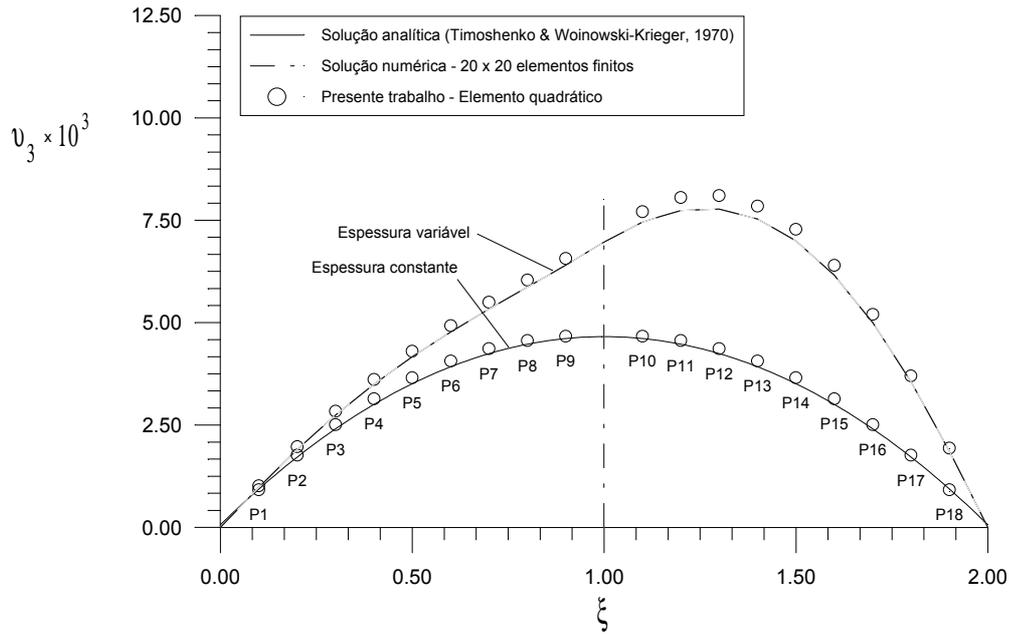


Figura 7-14: Deslocamento u_3 para placa retangular apoiada.

longo da linha de simetria longitudinal (pontos P1 a P18 na figura 7-13), e estão apresentados nas figuras 7-14 e 7-15. Para o caso de espessura constante os resultados são comparados com uma solução analítica baseada no modelo de Kirchhoff (Timoshenko & Woinowski-Krieger 1970). Para espessura variável utilizou-se como referência os resultados obtidos com um elemento finito de placa fina (Ansys 5.6 - User's Guide, 2001). Trata-se de um elemento (STIFF63) quadrangular formado pela condensação de quatro elementos triangulares cúbicos, o que lhe confere um desempenho muito bom na análise de placas finas. Adicionalmente, as figuras 7-16 e 7-17 comparam os mapas de deslocamento obtidos no casos de espessura variável em ambas as soluções. Os resultados de elementos finitos foram obtidos com uma malha de 20×20 elementos, enquanto os da presente formulação foram calculados com a malha da fig.7-13 e 81 pontos internos regularmente espaçados em cada subregião.

7.2 Problemas não-lineares

Nesta seção serão apresentados alguns resultados referentes à flexão geometricamente não-linear de placas obtidos com a formulação proposta. Inicialmente deve-se destacar

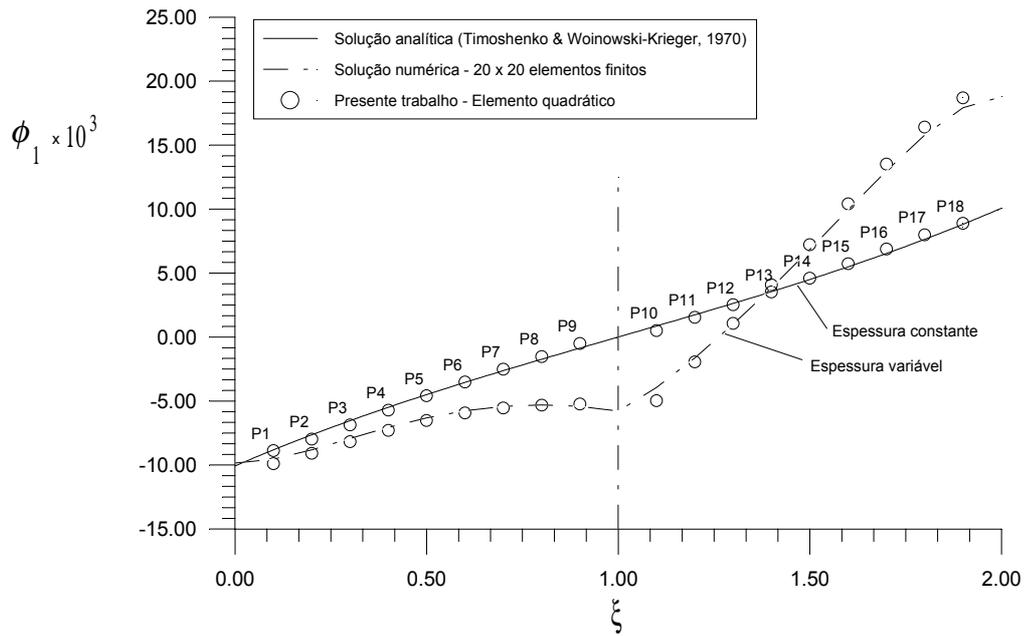
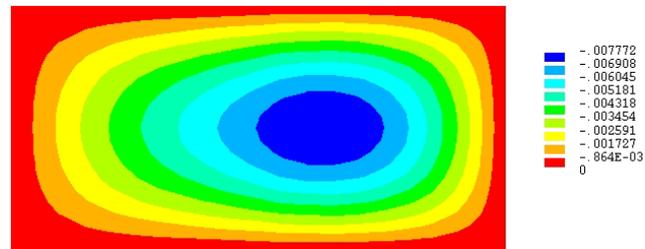
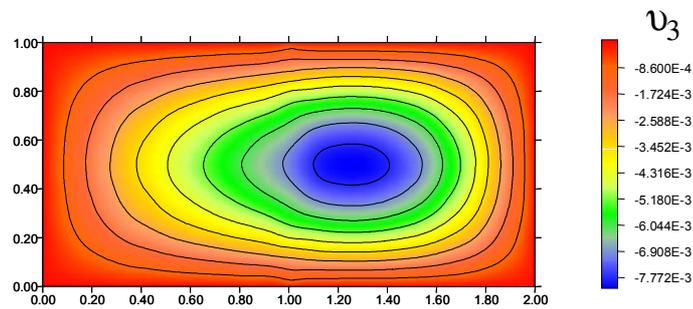


Figura 7-15: Rotação ψ_1 para placa retangular apoiada.



(a)



(b)

Figura 7-16: Padrão de deslocamentos u_3 . (a) Solução pelo MEF; (b) Presente trabalho.

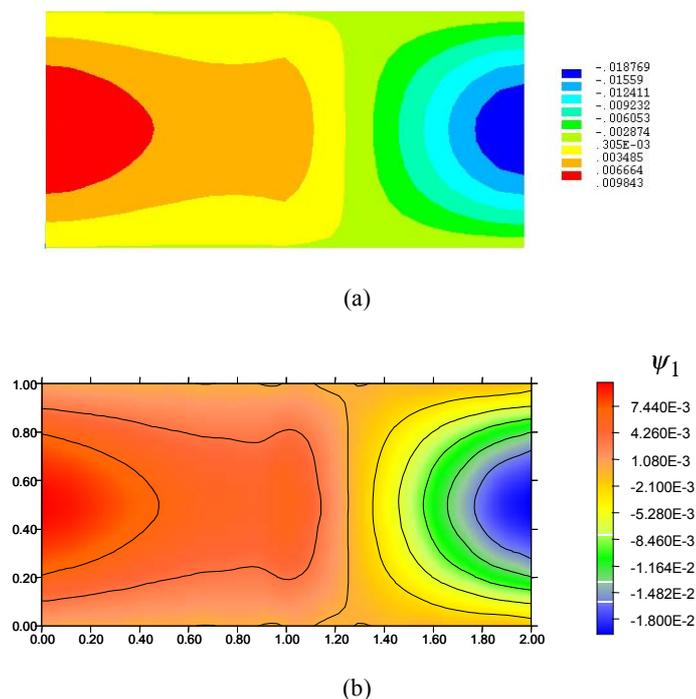


Figura 7-17: Padrão de deslocamentos ψ_1 . (a) Solução pelo MEF; (b) Presente trabalho.

que se por um lado soluções analíticas para flexão linear de placas finas já são poucas, para flexão não-linear estas soluções se tornam ainda mais escassas. No âmbito dos problemas de placas semi-espessas estas soluções se tornam então efetivamente raras. Isto é corroborado pelo fato de que mesmo trabalhos baseados no MEF poucas vezes fornecem resultados para espessuras significativamente grandes.

Para placas finas vários trabalhos têm persistido como principais referências, sendo alguns já bastante antigos. Dentre estes, selecionou-se as seguintes soluções analíticas ou semi-analíticas para serem utilizadas como comparação:

- Lévy (1942a), Lévy (1942b) e Wang (1948): Representam uma série de trabalhos sobre o assunto desenvolvidos pela NACA durante a década de 1940. São soluções baseadas em séries de Fourier duplas e bastante referenciadas. Embora tais soluções considerem poucos termos das séries, seus erros para placas engastadas são reportadamente menores que 2% (Pica et al. 1980).
- Berger (1955): Propõe soluções analíticas fechadas e por séries para placas circulares e quadradas, usando as equações de Berger. No entanto, sabe-se que esta abordagem

leva a resultados incorretos em certos problemas (Banerjee & Datta 1981).

- Rushton (1970): Fornece soluções obtidas com um método de relaxação dinâmica para placas quadradas. Seus erros são reportados como menores que 0.5% (Pica et al. 1980)
- Banerjee & Datta (1981): Extende o trabalho de Berger (1955) e obtém soluções para placas circulares e quadradas a partir do funcional energia de deformação total.

Para placas semi-espessas, os seguintes trabalhos serão utilizados como referências:

- Wilson & Borei (1964): Um dos primeiros trabalhos a propor soluções analíticas para flexão não-linear de placas que incluem o efeito do cisalhamento. O método é baseado na solução simultânea de cinco equações acopladas, nas quais o efeito do cisalhamento é inserido *a posteriori*. Apenas placas circulares engastadas são analisadas.
- Lim et al. (1988): Um dos raros trabalhos recentes a propor soluções não-lineares para placas semi-espessas. São deduzidas soluções por séries para uma versão simplificada da teoria de ordem superior de Reddy (1984a).

Dentre os resultados obtidos através de diversos métodos numéricos, as referências selecionadas para comparação foram:

- Bencharif & Ng (1994a), Bencharif & Ng (1994b): Fornece resultados para placas semi-espessas a partir da solução das equações da elasticidade tridimensional. As equações são simplificadas para pequenas rotações e resolvidas por diferenças finitas.
- Turvey & Osman (1990): Resolve as equações para flexão com grandes deslocamentos de placas de Mindlin através de um algoritmo de relaxação dinâmica.
- Azizian & Dawe (1985): Apresenta soluções numéricas para problemas de grandes deslocamentos de placas retangulares através do MTF - método das tiras finitas (*finite strip method*). Foi empregado o modelo de Mindlin. Este trabalho também apresenta uma solução para placas finas usando o método de Rayleigh-Ritz.

- Xiao-Yan et al. (1990): Um dos raros trabalhos de flexão de placas semi-espessas com o MEC. Embora os autores aleguem o uso do modelo de Reissner trata-se, efetivamente, de um modelo de Mindlin incompatível.
- Pica et al. (1980): Referência clássica do MEF no assunto, embora os autores explorem apenas casos de placas finas. Aqui são empregados apenas seus resultados obtidos com malhas de 8×8 elementos quadráticos lagrangeanos.

7.2.1 Placas quadradas

Os resultados apresentados a seguir se referem a placas quadradas sob carregamento uniforme. Variando-se o valor da carga foram obtidas as curvas $R \times r$ para os casos analisados. A medida de carregamento é dada por:

$$R = \frac{q_0 a^4}{Eh^4} \quad ,$$

enquanto a medida de deslocamento foi normalizado com a espessura da placa:

$$r = \frac{u_{3\max}}{h} \quad ,$$

onde w_{\max} é o deslocamento no centro da placa. Nas placas apoiadas, apenas o apoio SS2 foi utilizado.

As malhas empregadas são todas regulares, com as células de domínio sendo compatíveis com a malha de elementos de contorno. Tendo em vista que apenas células constantes e lineares foram implementadas será dada ênfase aos resultados para elementos de contorno constantes e lineares. As figuras 7-18 ilustram a disposição dos nós para alguns exemplos de malhas utilizadas. Dadas as diversas possibilidades combinações elemento de contorno/célula de domínio, nos resultados apresentados a seguir os elementos de contorno constantes, lineares e quadráticos serão identificados por BE2D0, BE2D1 e BE2D2, respectivamente. As células (quadrilaterais) constantes e lineares serão denominadas DC2DQ0 e DC2DQ1, respectivamente. A menos que especificado em contrário, apenas elementos de contorno lineares (BE2D1) foram utilizados.

Inicialmente foram analisados problemas de placas finas, a fim de comparar os resul-

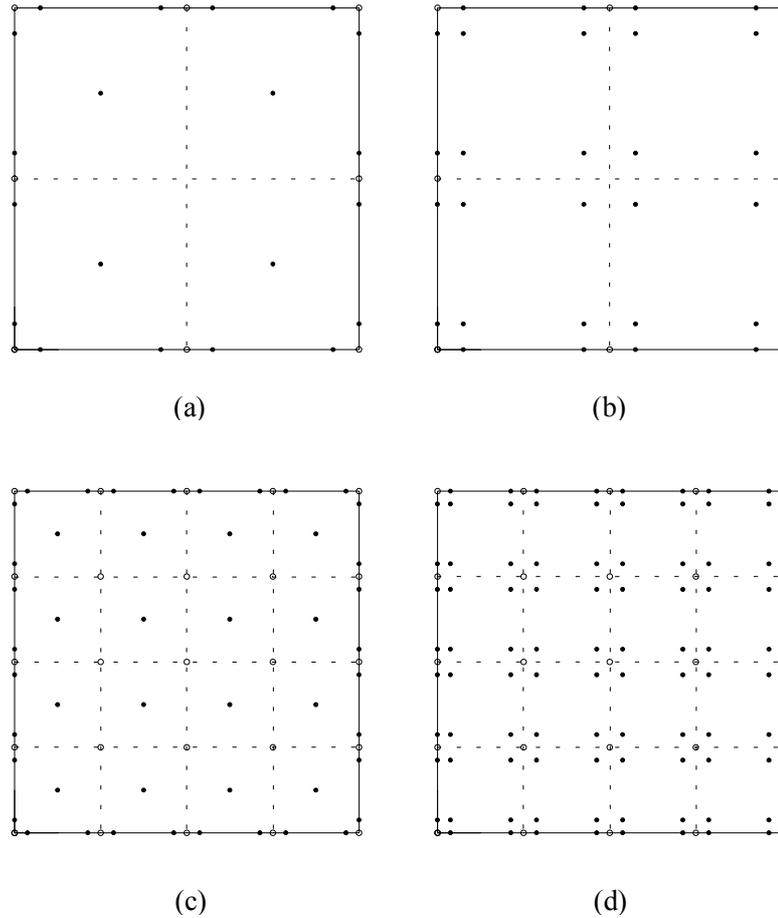


Figura 7-18: Exemplos de malhas empregadas em problemas não-lineares: (a) malha 2×2 - BE2D1/DC2D0; (b) malha 2×2 - BE2D1/DC2D1; (c) malha 4×4 - BE2D1/DC2D0; (d) malha 4×4 - BE2D1/DC2D1.

tados obtidos com as soluções analíticas disponíveis. Para tal foram analisadas placas com razão $h/a = 0.015$. Tendo em vista que o cálculo das derivadas dos deslocamentos translacionais é realizado através de diferenças finitas, o esforço computacional para cálculo dessas variáveis aumenta abruptamente com o refinamento da malha do domínio, pois o esquema de diferenças finitas é aplicado em torno de cada nó físico das células. Sendo assim, torna-se interessante manter a malha de domínio o mais pobre possível tanto em número de células quanto em ordem de interpolação. A figura 7-19 ilustra a trajetória de equilíbrio para uma placa quadrada apoiada utilizando as malhas das figuras 7-18a e 7-18b. Note-se que os resultados são satisfatórios para ambos os tipos de células empregadas, mesmo para malhas tão grosseiras. O refinamento da malha produz resultados similares até um certo tamanho de célula, quando passa a provocar um enrijecimento

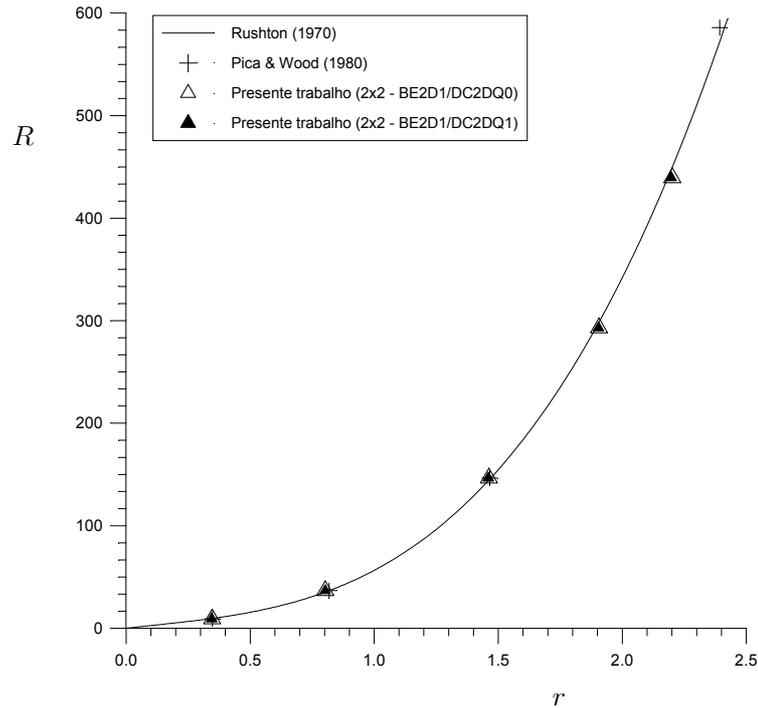


Figura 7-19: Curvas carga \times deslocamento para placa fina quadrada apoiada.

numérico da placa. Este efeito está ilustrado na figura 7-20, onde a malha de domínio foi refinada de 2×2 até 5×5 células. Aparentemente não há uma estabilização plenamente satisfatória dos resultados com o aumento do refino de malha. Dois fatores contribuem para este comportamento. Em primeiro lugar, o apoio SS2 não é o mais adequado para comparação com soluções baseadas no modelo de placa de Kirchhoff, embora a influência do tipo de apoio não seja predominante em problemas com espessura reduzida. E em segundo lugar, deve-se destacar que não foram tomados cuidados especiais com as integrais quase-singulares. Assim, à medida que a malha de domínio é refinada, o cálculo dos deslocamentos em pontos internos - eqs.(4.27) e (4.39) - é realizado em posições cada vez mais próximas do contorno da placa. Embora o uso de células constantes alivie o problema em relação às lineares (pela existência de um único nó), a partir de um certo refino o problema volta a se manifestar. Isto pode ser comprovado através da análise de placas engastadas, onde os termos fortemente singulares não têm efeito. A figura 7-21 ilustra claramente isto, onde as mesmas malhas da figura 7-20 convergem de forma satisfatória.

A seguir foi investigado como o efeito do aumento da espessura se manifestava nos resultados obtidos com a presente formulação. As figuras 7-22 a 7-24 ilustram estes

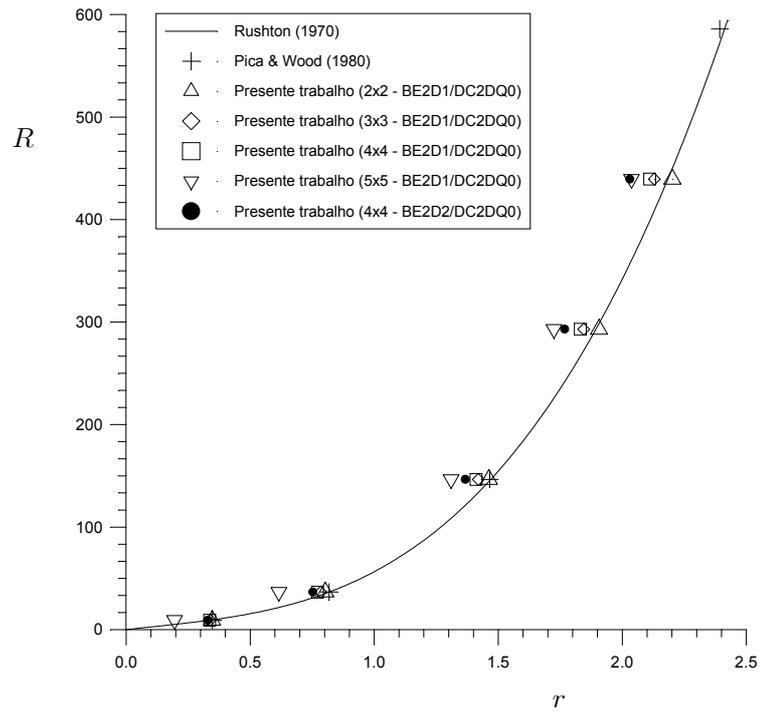


Figura 7-20: Curvas carga \times deslocamento para placa fina quadrada apoiada.

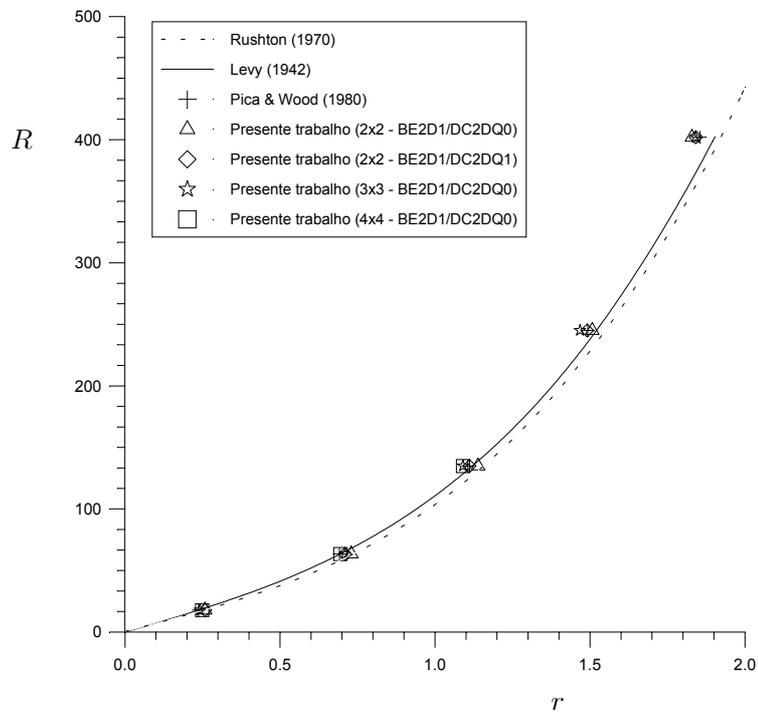


Figura 7-21: Curvas carga \times deslocamento para placa fina quadrada engastada.

resultados, para diversas combinações de elementos de contorno e células de domínio. Na figura 7-22 foram empregadas malhas do tipo BE2D1/DC2DQ0, na 7-23 foram usadas malhas BE2D1/DC2DQ1 e na figura 7-24 foram utilizadas malhas BE2D2/DC2DQ0. A figura 7-22 indica que os resultados obtidos com das malhas BE2D1/DC2DQ0 são bastante satisfatórios, mesmo para malhas grosseiras. Já as figuras 7-23 e 7-24 apontam para uma piora dos resultados em relação à figura 7-22. Embora os resultados da figura 7-23 também sejam bons, eles mostram sinais de contaminação de integrais quase-singulares mal integradas. Na figura 7-24 este efeito volta a se manifestar de forma ainda mais severa, mesmo se comparado com as soluções de referência para placas finas. Destaque-se que, em experimentos numéricos preliminares, o uso de interpolação quadrática sobre o contorno com interpolações constantes ou lineares no domínio não melhorou os resultados. Ao contrário, sugere uma incompatibilidade desta combinação para problemas não-lineares.

O fato dos modelos de placa de Mindlin e de Reissner incluírem o efeito do cisalhamento transversal obviamente implica em um aumento da flexibilidade em relação ao modelo de Kirchhoff. No regime não linear, entretanto, esta influência não parece ocorrer ao longo de uma grande faixa de não-linearidade. Isto é, a partir de uma certa magnitude de deslocamento máximo, outros efeitos passam a ser mais importantes que esta flexibilidade adicional. Isto fica mais evidente na análise de placas quadradas engastadas, onde o padrão de deformação é mais severo que em placas apoiadas e como consequência o efeito do cisalhamento se manifesta de forma mais visível no trecho inicial da curva carga \times deslocamento. Na figura 7-25 estão ilustrados os resultados obtidos para placa quadrada engastada juntamente com algumas soluções analíticas para placas finas. Note-se que à medida que a espessura é aumentada, a inclinação da trajetória de equilíbrio é reduzida, notadamente em função das deformações cisalhantes. Esta diferença, no entanto, persiste até deslocamentos máximos da ordem de 100% do valor da espessura. A partir deste ponto, as tensões de membrana passam a desempenhar um papel mais importante no suporte ao carregamento transversal. Alguns dos resultados para placas semi-espessas disponíveis na literatura parecem ignorar este fato, acompanhando a solução de placa fina por toda a trajetória de equilíbrio.

A base argumentativa do que se está expondo aqui é, basicamente, a inexistência de soluções analíticas ou semi-analíticas definitivas para placas semi-espessas em regime de

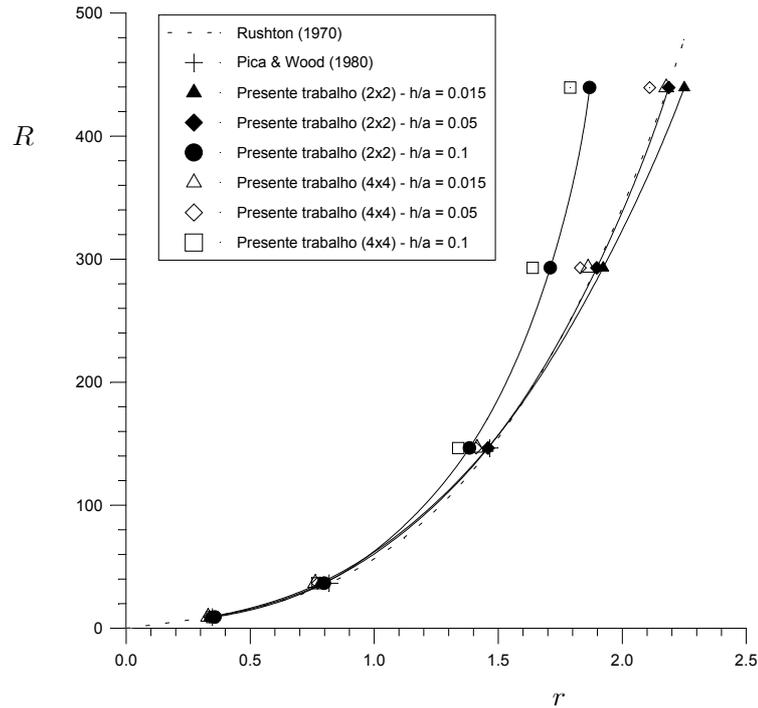


Figura 7-22: Curvas carga \times deslocamento para placa semi-espessa quadrada apoiada. Malha BE2D1/DC2DQ0.

grandes deslocamentos, ao menos no âmbito das geometrias retangulares. Este fato tem contribuído - ou talvez forçado - diversos autores a limitar sua apresentação de resultados a faixas moderadas de não-linearidade ou a espessuras reduzidas. Isto também alimenta diversas dúvidas quanto ao processo cinemático das teorias de placa aqui abordadas quando em regimes fortemente não lineares (tipicamente com razões $r > 1$). Para ilustrar ainda mais este aspecto, a figura 7-26 apresenta uma amostra da grande diversidade de soluções disponíveis na literatura para flexão de placas quadradas apoiadas. Infelizmente não se pode simplesmente recorrer à solução das equações da elasticidade tridimensional, pois este seria um problema diferente dos modelos aqui analisados e tornaria inconsistente uma comparação direta dos resultados.

Para fins de comparação numérica, a tabela 7.4 apresenta os resultados aqui obtidos com algumas outras que utilizam o modelo de Mindlin. Estes valores se referem a placa quadrada apoiada. Estão ilustrados os resultados de Azizian & Dawe (1985) e uma das raras soluções de elementos de contorno para este tipo de problema (Xiao-Yan et al. 1990), ambas baseadas em modelos de Mindlin incompatíveis. A título de comparação, também

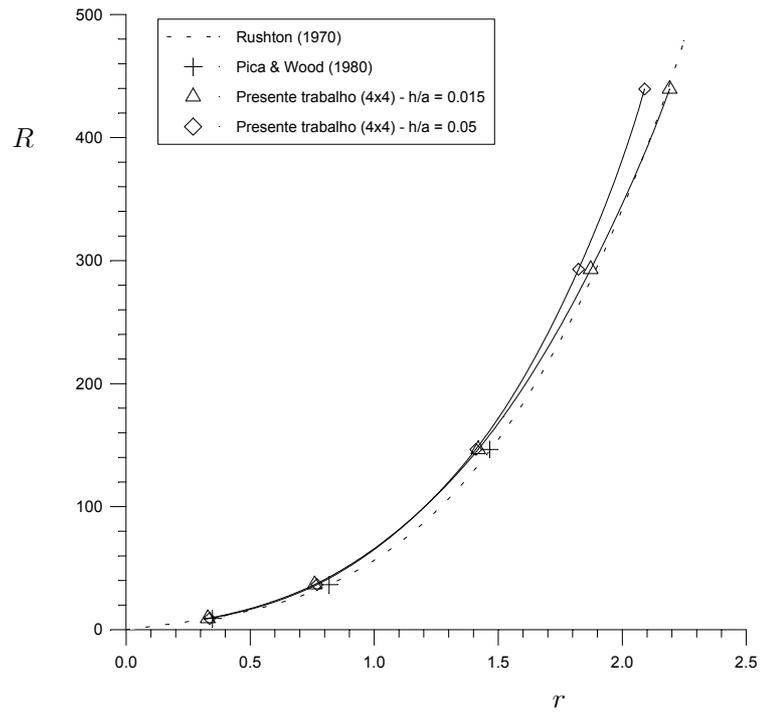


Figura 7-23: Curvas carga \times deslocamento para placa semi-espessa quadrada apoiada. Malha BE2D1/DC2DQ1.

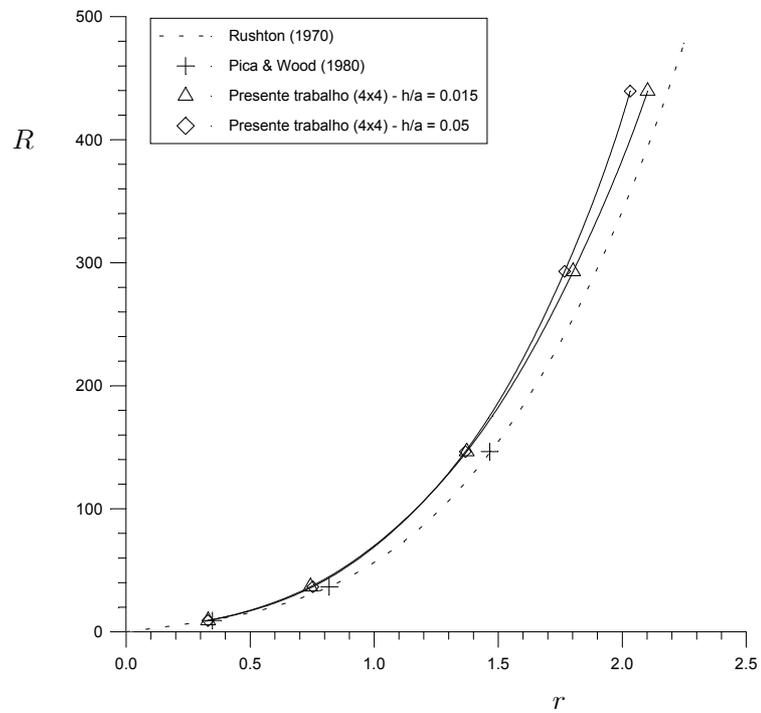


Figura 7-24: Curvas carga \times deslocamento para placa semi-espessa quadrada apoiada. Malha BE2D2/DC2DQ0.

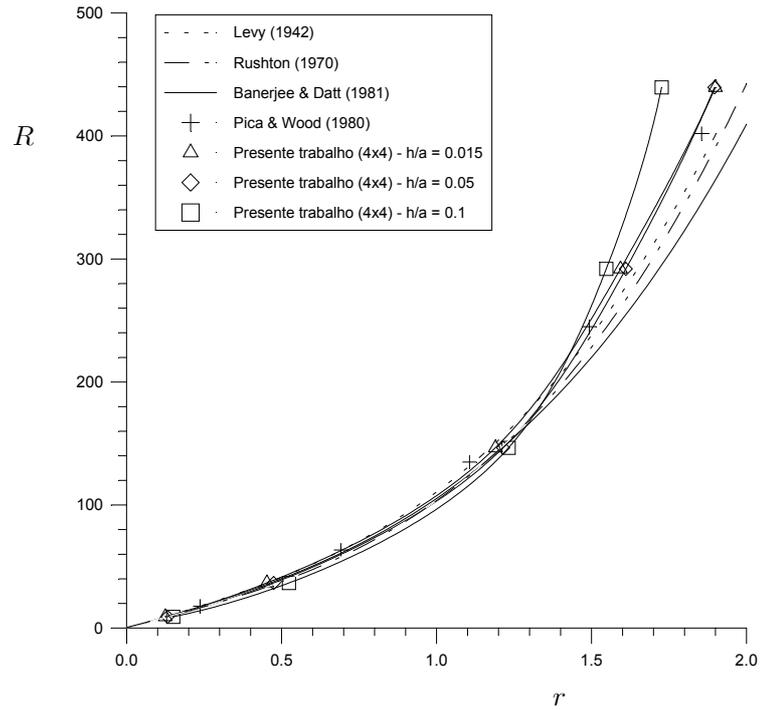


Figura 7-25: Curvas carga \times deslocamento para placa semi-espessa quadrada engastada. Malha BE2D1/DC2DQ0.

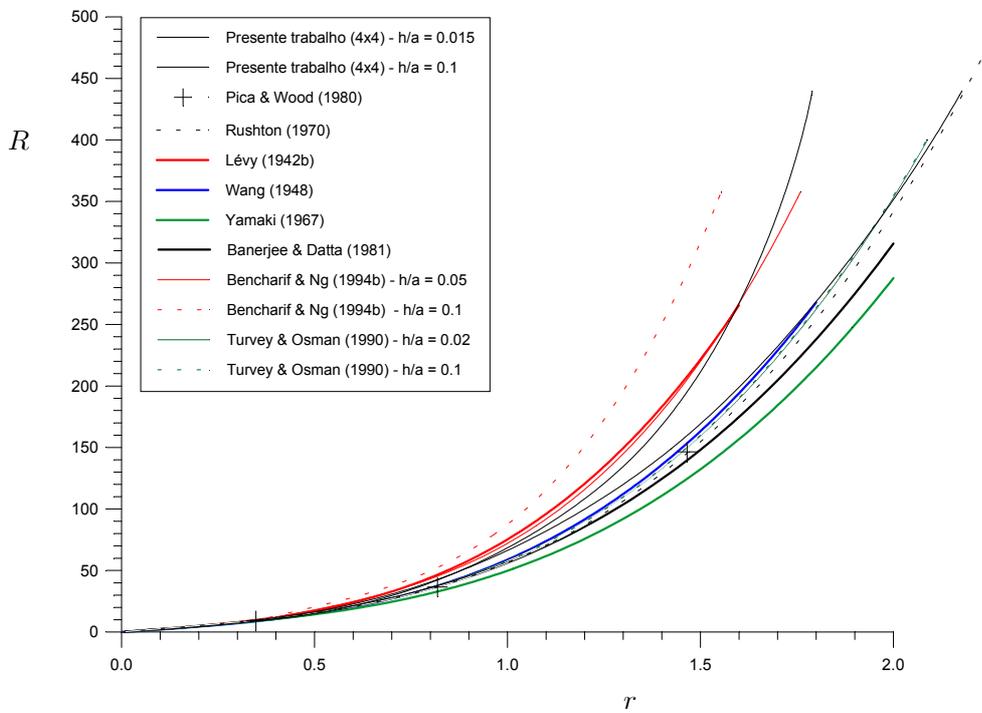


Figura 7-26: Exemplos de soluções para flexão não-linear de placas finas e semi-espessas.

	<i>R</i>			
	0.9158	4.579	6.868	9.158
Modelo de Kirchhoff (Azizian & Dawe 1985) ¹	0.04053	0.1929	0.2750	0.3467
MTF (Azizian & Dawe 1985)	0.04205	0.1950	0.2776	0.3494
MEC (Xiao-Yan et al. 1990)	0.04090	0.1942	0.2767	0.3489
Presente trabalho (SS2) BE2D1/DC2DQ0	0.04200	0.1969	0.2775	0.3464
Presente trabalho (SS2) BE2D1/DC2DQ1	0.04199	0.1958	0.2753	0.3425
Presente trabalho (SS1) BE2D1/DC2DQ0	0.04428	0.2056	0.2878	0.3573
Presente trabalho (SS1) BE2D1/DC2DQ1	0.04426	0.2041	0.2849	0.3534

¹Método de Rayleigh-Ritz (Azizian & Dawe 1985).

Tabela 7.4: Deslocamento central normalizado r para placa semi-espessa apoiada sob carga uniforme ($h/a = 0,05$). Malha 2×2 .

	<i>R</i>			
	0.9158	4.579	6.868	9.158
Modelo de Kirchhoff (Azizian & Dawe 1985)	0.01915	0.09513	0.1416	0.1867
MTF (Azizian & Dawe 1985)	0.01991	0.09883	0.1469	0.1936
MEC (Xiao-Yan et al. 1990)	0.01991	0.09840	0.1455	0.1904
Presente trabalho BE2D1/DC2DQ0	0.02020	0.1002	0.1488	0.1957
Presente trabalho BE2D1/DC2DQ1	0.02020	0.1001	0.1485	0.1952

Tabela 7.5: Deslocamento central normalizado r para placa semi-espessa apoiada/engastada sob carga uniforme ($h/a = 0,05$). Malha 2×2 .

estão incluídos os resultados de uma solução baseada no modelo clássico e as obtidas com o presente trabalho usando apoio SS1. A tabela 7.5 apresenta a mesma comparação para uma placa quadrada apoiada (SS2) em dois lados opostos e engastada nos outros.

7.2.2 Placas circulares

A título de ilustração, um único caso de placa circular foi analisado, sendo esta engastada sob carregamento uniforme. Este caso foi escolhido por possuir soluções analíticas tanto para flexão linear quanto não-linear. A malha empregada está ilustrada na figura 7-27, e os resultados correspondentes estão mostrados na figura 7-28 juntamente com outras soluções conhecidas.

Este exemplo é interessante por ratificar alguns dos argumentos utilizados na discussão dos resultados de placas quadradas. Note-se que a solução exata do problema linear (Reissner 1945) está sobreposta às soluções não-lineares na figura 7-28, onde se verifica que os resultados obtidos respeitam a solução linear no início da trajetória de equilíbrio, e este aspecto depõe a favor da formulação proposta. Também é interessante

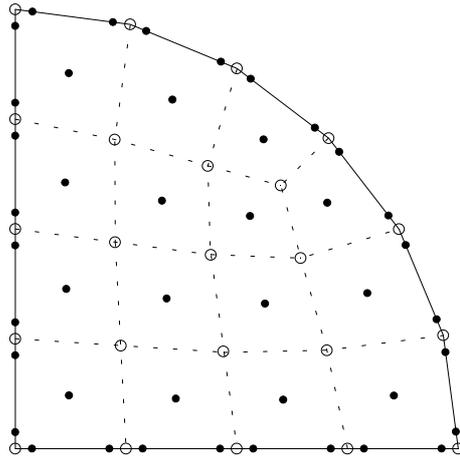


Figura 7-27: Malha empregada na análise de placas circulares.

notar que o aumento da espessura implica em rigidez menor para quase toda a faixa de deslocamentos ilustrada, como é de se esperar. No entanto, observa-se que à medida que a carga é incrementada ocorre uma inversão deste comportamento. Com efeito, isto foi verificado em experimentos numéricos, indicando que para parâmetros r maiores as curvas correspondentes a cada espessura invertem suas posições relativas. Não se pode advogar este comportamento cinemático apenas à luz dos resultados aqui obtidos, afinal não é possível afirmar que este enrijecimento proporcional à espessura não seja puramente um efeito numérico espúrio, mas sugere uma interessante área de investigação. Na realidade, o fato da inclinação das curvas $R \times r$ serem inversamente proporcionais à espessura foi sugerido por Wilson & Boresi (1964), mas seus resultados estavam limitados à razões $h/2R = 0,05$. E além disto, esta regra não parece prevalecer para placas retangulares, como indicam outras soluções (a figura 7-26 ilustra algumas).

7.2.3 Placas com espessura variável

O caso de placa retangular com espessura variável estudado no final da seção 7.1.2 foi reanalisado considerando grandes deslocamentos, a fim de validar o procedimento de manipulação de subregiões também para o caso não-linear. Os resultados foram comparados com a solução obtida através da mesma malha de elementos finitos do caso linear. A malha de elementos de contorno empregou apenas interpolação linear enquanto a malha do domínio consistiu de quatro células constantes por subregião, a fim de permitir uma

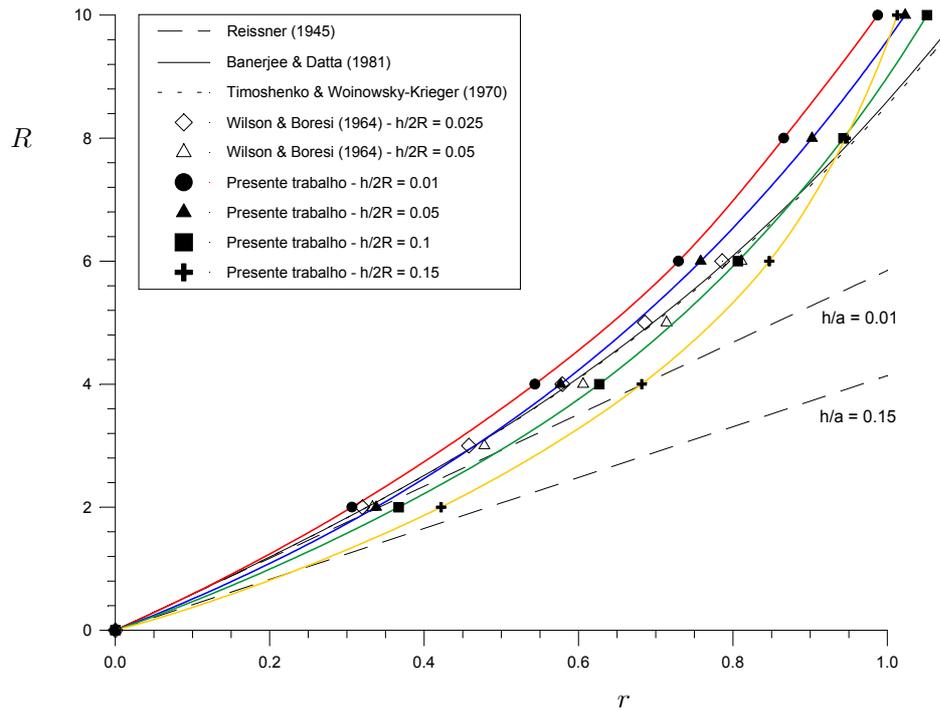


Figura 7-28: Curvas carga \times deslocamento para placa semi-espessa circular engastada. Malha BE2D1/DC2DQ0.

melhor integração dos termos não-lineares. A figura 7-29 ilustra esta malha. Os deslocamentos transversais e as rotações foram calculados sobre pontos internos ao longo da linha de simetria longitudinal (pontos P1 a P18 na figura 7-29), e estão apresentados nas figuras 7-30 e 7-31 para cinco passos de carregamento.

7.3 Aspectos numéricos

Os resultados apresentados nas seções anteriores mostram erros perfeitamente aceitáveis para aplicações em engenharia. No entanto, diversos parâmetros contribuem de forma significativa para alterar o valor destes erros. Além da própria malha empregada, pode-se citar o número de pontos de integração, o valor do recuo dos nós das extremidades dos elementos, as diversas combinações de malha do contorno e do domínio, fatores de correção das tensões cisalhantes, qualidade da integração numérica etc. Alguns destes já foram direta ou indiretamente abordados neste capítulo ou nos anteriores.

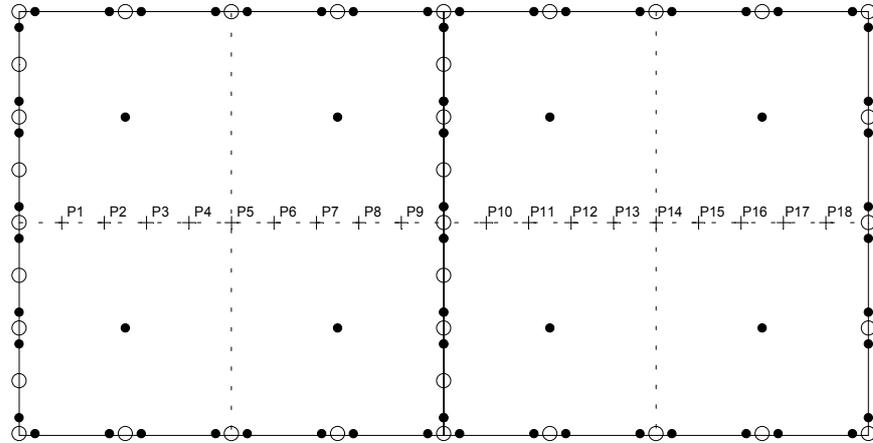


Figura 7-29: Placa retangular modelada com duas subregiões. Malha de elementos de contorno, células de domínio e pontos internos para análise não-linear.

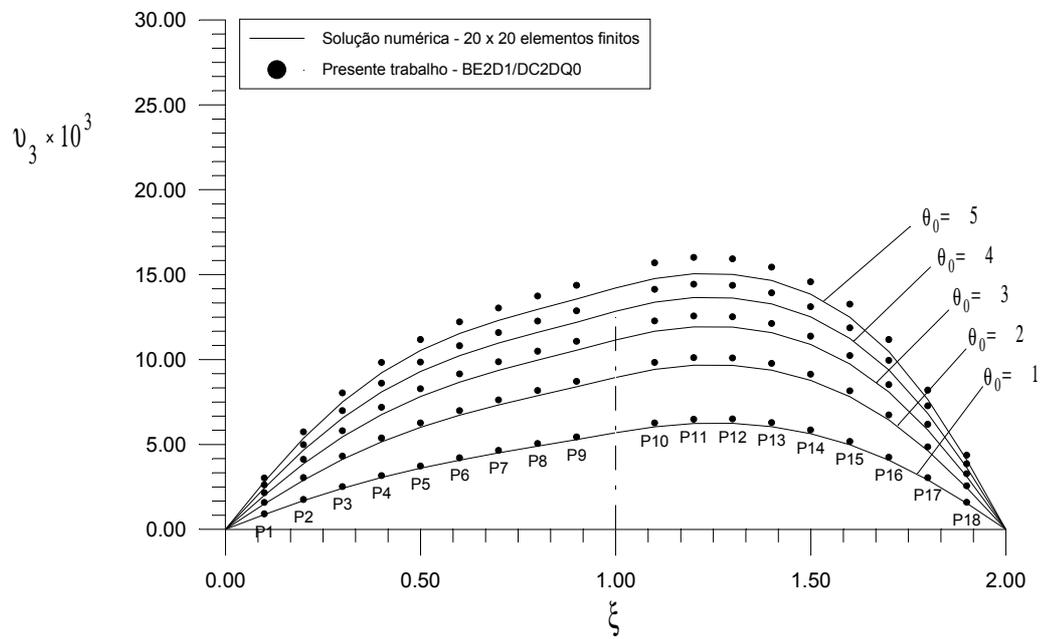


Figura 7-30: Deslocamento u_3 para placa retangular apoiada.

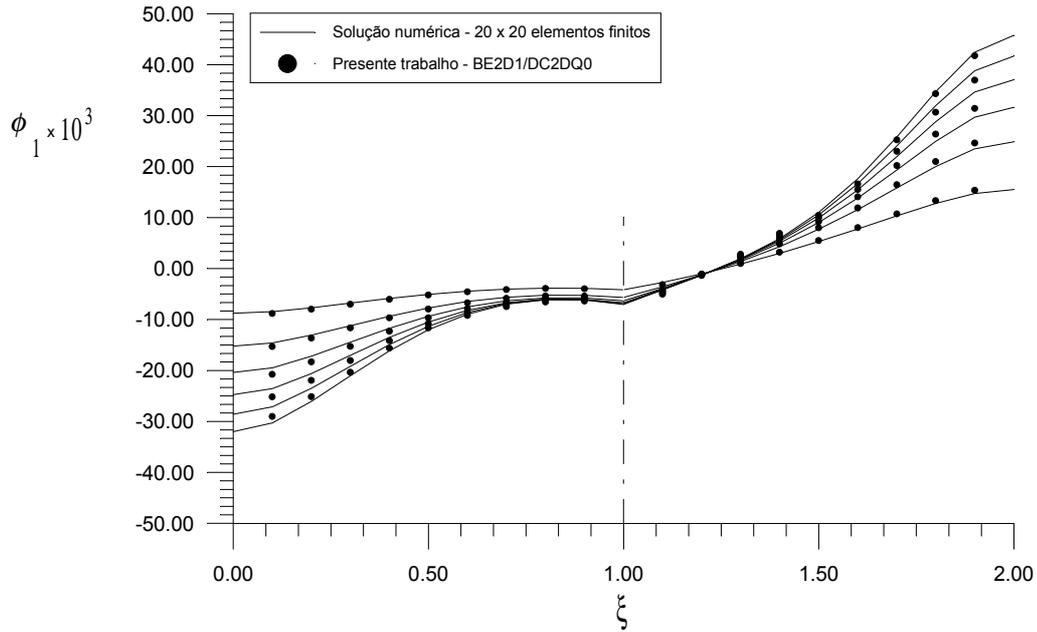


Figura 7-31: Deslocamento ψ_1 para placa retangular apoiada.

7.3.1 Cálculo de $u_{i,\alpha}$ através de diferenças finitas

Conforme destacado no Capítulo 4, evitou-se o caráter hipersingular das equações para derivadas dos deslocamentos através do uso de diferenças finitas em torno dos pontos internos. No entanto, as equações para cálculo destas variáveis foram implementadas na análise linear com o objetivo de verificar o desempenho comparativo destas duas técnicas. Para este fim foi analisada uma placa quadrada apoiada sob carga uniforme $q_0 = 1000$ e as derivadas $\partial u_3 / \partial x_2$ foram calculadas ao longo da linha de simetria $x_1 = 0,5$ pela eq.(4.52) e pela eq.(4.69). A tabela 7.6 apresenta os resultados obtidos, onde também estão incluídos os valores analíticos obtidos para o modelo de placa de Kirchhoff (Timoshenko & Woinowski-Krieger 1970). A concordância entre os valores obtidos com a equação integral e com diferenças finitas é marcante, demonstrando a aplicabilidade deste procedimento para cálculo das derivadas dos deslocamentos.

7.3.2 Fator de relaxação

O objetivo desta seção é ilustrar a influência deste parâmetro empregado na análise não-linear. Trata-se do fator de relaxação ρ , que foi introduzido na eq.(4.71) como multipli-

h/a		x_2			
		0.1	0.2	0.3	0.4
0, 015	Placas finas	13.524	11.036	7.7084	3.9449
	Equação integral	13.506	11.023	7.6920	3.9562
	Diferenças finitas	13.545	11.048	7.7069	3.9632
0, 02	Placas finas	5.7054	4.6558	3.2520	1.6643
	Equação integral	5.7054	4.6552	3.2478	1.6707
	Diferenças finitas	5.7210	4.6656	3.2542	1.6737
0, 025	Placas finas	2.9212	2.3838	1.6650	0.85211
	Equação integral	2.9255	2.3863	1.6646	0.85644
	Diferenças finitas	2.9329	2.3913	1.6677	0.85791
0, 05	Placas finas	0.36514	0.29797	0.20813	0.10651
	Equação integral	0.36975	0.30101	0.20968	0.10806
	Diferenças finitas	0.37041	0.30148	0.20998	0.10821
0, 1	Placas finas	0.045643	0.037247	0.026016	0.013314
	Equação integral	0.048188	0.038954	0.026995	0.014003
	Diferenças finitas	0.048220	0.038983	0.027016	0.014013
0, 15	Placas finas	0.013524	0.011036	0.0077084	0.0039449
	Equação integral	0.015244	0.012193	0.0083837	0.0043921
	Diferenças finitas	0.015238	0.012194	0.0083855	0.0043934
0, 2	Placas finas	0.0057054	0.0046558	0.0032520	0.0016643
	Equação integral	0.0069997	0.0055277	0.0037635	0.0019961
	Diferenças finitas	0.0069905	0.0055248	0.0037627	0.0019960
0, 25	Placas finas	0.0029212	0.0023838	0.0016650	0.00085211
	Equação integral	0.0039574	0.0030822	0.0020758	0.00011161
	Diferenças finitas	0.0039490	0.0030791	0.0020746	0.00011157

Tabela 7.6: Comparação de formas de cálculo das rotações $\partial u_3/\partial x_2$ em uma placa quadrada apoiada.

cador empregado na atualização das variáveis após cada iteração da figura 4-6. A escolha deste parâmetro numérico se revelou particularmente importante na análise dos problemas não-lineares. Uma escolha inapropriada de ρ na eq.(4.71) pode impedir completamente a convergência do processo iterativo.

A fim de ilustrar este comportamento, a convergência do deslocamento máximo uma placa quadrada apoiada (SS2) foi acompanhada ao longo de certos passos de carregamento, para três valores de fator de relaxação. As figuras 7-32, 7-33 e 7-34 apresentam este comportamento para os três primeiros passo de carga ilustrados na figura 7-19. Os deslocamentos estão normalizados com a solução de Rushton (1970):

$$w_n = \frac{u_{3\max}}{w_{\text{Rushton}}} .$$

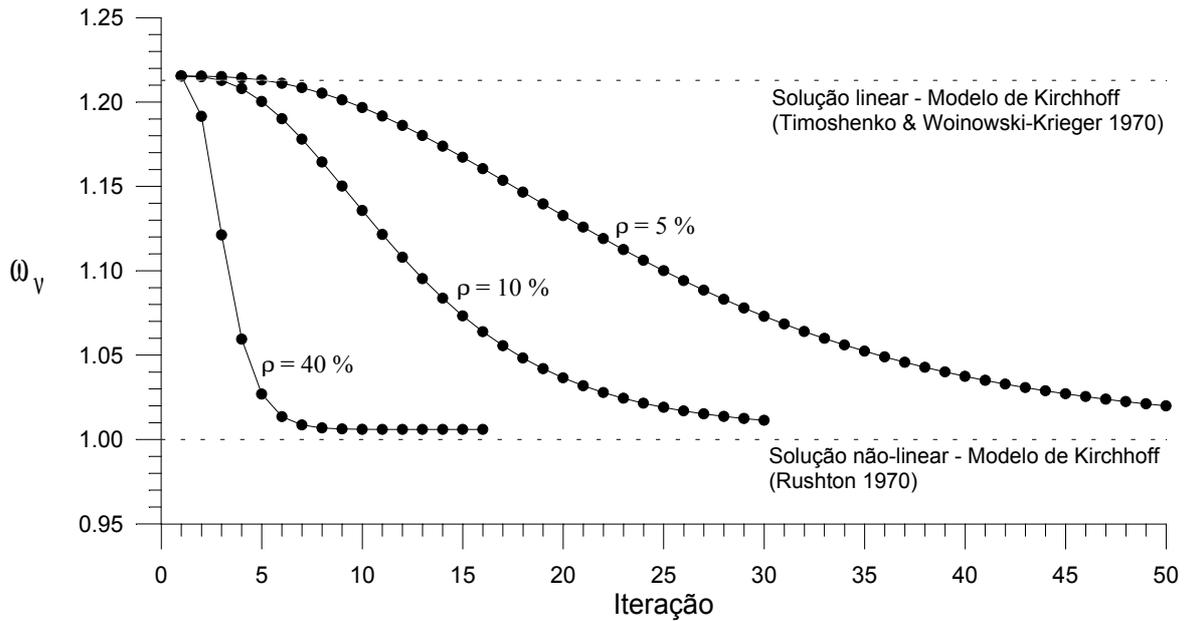


Figura 7-32: Convergência do deslocamento central para diversos valores de ρ ($q_0 = 172$).

A figura 7-32 mostra claramente que a convergência é monotônica para todos os valores de ρ utilizados. Na figura 7-33 verifica-se que para um fator de relaxação de 40% a convergência perde sua monotonicidade. Já figura 7-34 ilustra a necessidade de valores abaixo de 10% para garantir uma convergência monotônica. No caso desta figura, não se obteve convergência para $\rho = 40\%$. Estes gráficos comprovam que à medida que o grau de não-linearidade aumenta, fatores de relaxação cada vez menores devem ser empregados. Mesmo para o caso da figura 7-32, onde a não-linearidade ainda é bastante pequena, valores maiores que 60% já dificultaram a convergência do processo.

Além disso, é interessante destacar que a relaxação não teve efeito quando aplicada às derivadas $u_{\beta,\alpha}$, isto é, não se verificou nenhuma diferença significativa nos resultados obtidos pelas equações (4.70) e (4.71) para estas variáveis. A aplicação da relaxação mostrou-se mandatória apenas para as derivadas do deslocamento transversal $u_{3,\alpha}$.

Evidentemente o comportamento da convergência do processo iterativo é consequência da forma como o algoritmo não-linear é implementado, e outras variações do algoritmo ilustrado na figura 4-6 podem estabilizar mais o processo em relação ao fator de relaxação.

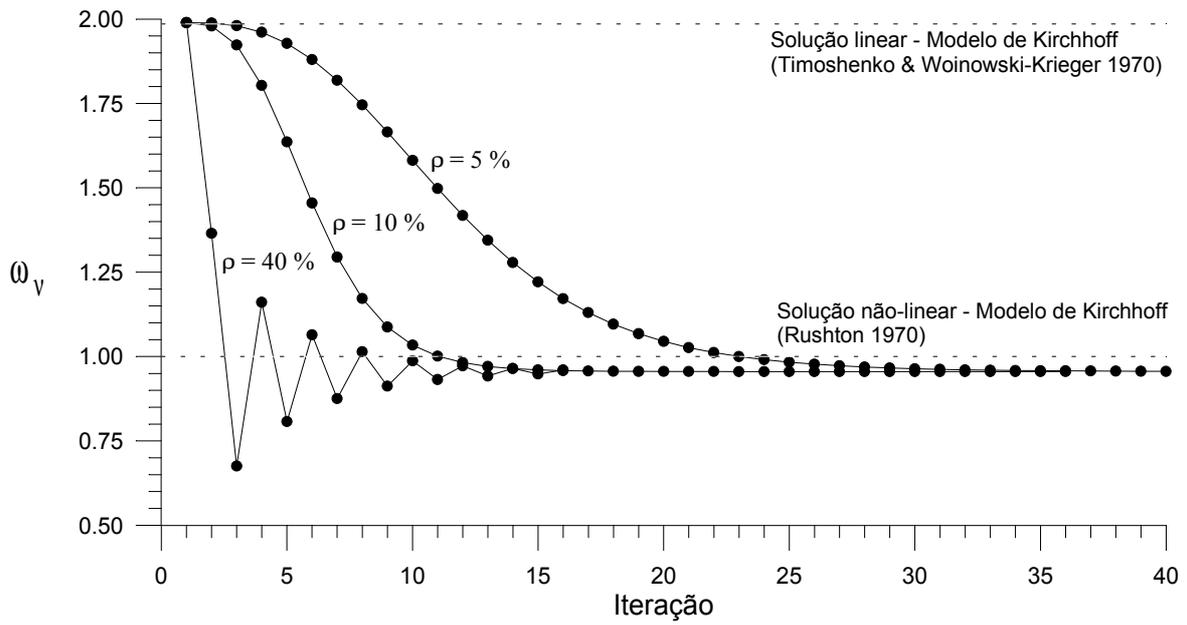


Figura 7-33: Convergência do deslocamento central para diversos valores de ρ ($q_0 = 686$).

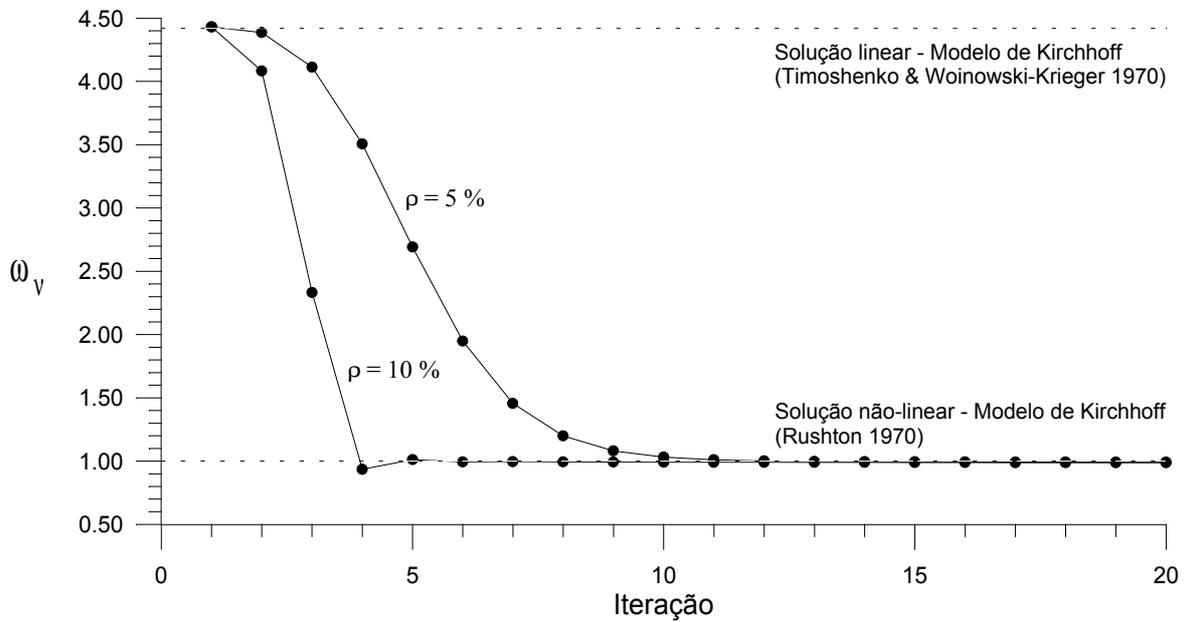


Figura 7-34: Convergência do deslocamento central para diversos valores de ρ ($q_0 = 2747$).

7.4 Resumo do Capítulo 7

Este capítulo apresentou os resultados numéricos obtidos com a implementação numérica da formulação proposta. Foram analisados problemas lineares de elasticidade bidimensional e flexão de placas, utilizando os novos procedimentos de integração desenvolvidos. Estes resultados comprovam a eficiência das expansões assintóticas desenvolvidas em problemas práticos.

Também foram apresentados diversos resultados para flexão não-linear elástica, para diversas geometrias e condições de contorno. As curvas carga \times deslocamento se mostraram bastante satisfatórias, com erros normalmente inferiores a 4%. Os resultados foram comparados com outras soluções disponíveis na literatura e alguns comentários a respeito do desempenho da formulação foram destacados. Dada a escassez de soluções similares para placas de Mindlin, os resultados apresentados neste capítulo se revestem de grande interesse.

Capítulo 8

Conclusões

Este presente trabalho apresentou basicamente o seguinte:

- Uma formulação integral unificada para análise de flexão linear e geometricamente não-linear de placas modeladas pelas teorias de Mindlin e de Reissner.
- Uma metodologia de solução das equações integrais através do MEC direto, contando com diversos elementos de contorno e células de domínio. Desta metodologia são extraídos algoritmos para solução de problemas de elasticidade bidimensional, estabilidade de placas, e flexão linear e não-linear elástica de placas.
- Um procedimento de regularização para as integrais fortemente singulares encontradas na análise de problemas de elasticidade bidimensional e flexão de placas semi-espessas.
- Uma proposta geral de arquitetura orientada a objetos para implementação do MEC e do MEF, baseada em um conjunto classes de análise, classes de modelo computacional e classes auxiliares.

Os resultados apresentados ao longo dos capítulos anteriores permitem agora alguns comentários a respeito dos pontos destacados acima.

A formulação integral é particularmente genérica, e comporta modificações que permitam a consideração de outros efeitos. Os modelos de placa empregados estão considerados de forma unificada, de modo que as respectivas soluções fundamentais se encarregam de

manter suas diferenças inerentes. Isto permite que outros modelos de placa similares possam ser estudados sob uma mesma abordagem. Portanto a estrutura matemática aqui desenvolvida constitui um bom ponto de partida para construção de uma plataforma para estudo de outros modelos de placa.

A metodologia de solução empregada no caso linear não apresenta diferença em relação à outras formulações baseadas no MEC. Para flexão não-linear, entretanto, foi sugerida uma possibilidade dentre algumas possíveis. O procedimento iterativo, por exemplo, foi baseado na idéia de que as estimativas dos termos não-lineares são conhecidas no início de cada iteração, pois foram calculados na iteração anterior. Esta metodologia parece conceitualmente vantajosa porque desta forma o problema de valores no contorno associado desempenha o papel principal na solução. Variações desta podem ser projetadas de modo que não apenas o problema de valores no contorno mas também o problema de domínio sejam resolvidos simultaneamente. Embora o número de variáveis simultâneas aumente, este procedimento levaria a uma convergência mais rápida. A metodologia de solução e as matrizes associadas descritas no Capítulo 4 não eliminam esta possibilidade. O tratamento dado à problemas modelados por subregiões é um outro aspecto importante do presente trabalho porque formaliza a montagem das submatrizes para um número arbitrário de subregiões.

Um dos aspectos originais do presente trabalho é o tratamento dado às integrais fortemente singulares. O uso do método direto nestes casos mostrou-se bastante atrativo, dispensando o uso de procedimentos mais elaborados para cômputo deste tipo de integral. No entanto, o maior valor desta abordagem está na sua robustez, uma vez que não impõe qualquer restrição quanto à forma interpolatória empregada para as variáveis ou geometria. A necessidade de se conhecer as expansões assintóticas para os tensores envolvidos não impõe dificuldade alguma à aplicação do método, já que sua dedução é sistemática para a maioria das soluções fundamentais. As expansões assintóticas para os tensores fundamentais de placa desenvolvidas no Apêndice C constituem uma contribuição neste sentido. Quanto à eficiência do método direto, os resultados para os casos lineares apresentados no Capítulo 7 servem para confirmar seu bom desempenho. Em problemas de flexão de placas finas, as curvas de convergência obtidas foram todas satisfatórias e, no caso do elemento linear, mostrou-se mais precisa que a quadratura de Kutt. A comparação

com resultados similares obtidos através de imposição de movimento de corpo rígido sugere que o método direto é um substituto tão preciso quanto o primeiro. Adicionalmente, destaque-se a grande diferença no tempo consumido no cálculo computacional, em favor do método direto. Quanto às integrais fracamente singulares, é suficiente mencionar que o procedimento de Telles é uma técnica muito versátil não apenas para integração numérica destes termos, mas também de funções que apresentam gradientes moderados dentro do intervalo de integração. No entanto, os resultados do Capítulo 5 demonstram que a combinação de funções de Bessel $B(z)$ necessita de em torno de 30 pontos de Gauss para ser integrada corretamente quando combinada com funções de interpolação quadráticas, mesmo sob efeito da transformação.

A arquitetura orientada a objetos proposta no Capítulo 6 é um dos aspectos do presente trabalho que se reveste de maior originalidade. Sua aplicação ao longo deste trabalho permitiu a implementação de diversos algoritmos com grande simplicidade. O objetivo inicial de desvincular tarefas de análise das classes de modelo foi plenamente atingido. Este é um dos principais fatores que permitiram que esta arquitetura chegasse ao grau de generalidade pretendido na sua concepção. A estrutura básica implementada pode realmente ser estendida e reutilizada para um grande número de aplicações, e neste sentido pode ser interpretada como um bom projeto de plataforma de desenvolvimento de programas de pesquisa baseados no MEC e também no MEF.

No que diz respeito aos resultados numéricos apresentados, vale destacar que a formulação proposta forneceu valores satisfatórios tanto para problemas de flexão linear quanto não-linear. Foi dada ênfase ao modelo de Mindlin, pois estes resultados são pouco explorados no âmbito das soluções analíticas. Os resultados para flexão linear de placas semi-espessas mostraram boa concordância com outras soluções, e os valores de deslocamento foram sistematicamente maiores que as soluções baseadas no modelo de Reissner. Este comportamento está correto, especialmente em placas com espessuras maiores, conforme demonstrado por publicações recentes (por exemplo, Wang et al. 2001). Vale lembrar que as diferenças observadas nas curvas de convergência ilustradas no Capítulo 7 dependem significativamente da solução utilizada na normalização dos resultados, e estas não são numerosas, especialmente para fatores de correção das tensões cisalhantes diferentes de $5/6$. Adicionalmente, há alguma dificuldade em se encontrar soluções confiáveis

para razões $h/a > 0, 2$.

Nos problemas não-lineares deve-se lembrar que foram utilizados elementos e células constantes e lineares, e ainda assim foram obtidos resultados bastante satisfatórios mesmo para malhas 2×2 , algo dificilmente atingível com elementos finitos convencionais de baixa ordem. Foi possível atingir faixas de deslocamento máximo bem maiores que a espessura da placa. Notadamente, não se tem conhecimento de trabalhos aplicando o MEC para placas semi-espessas com deslocamentos desta ordem. O recurso da aplicação de diferenças finitas no cálculo de $u_{i,\alpha}$ mostrou-se satisfatório para efeito de validação da formulação, mas implica no cômputo dos deslocamentos em um número muito grande de pontos internos, o que torna a abordagem menos competitiva em relação a outros métodos. O uso das equações integrais para cálculo destas variáveis é a melhor opção neste caso mas então a formulação torna-se, efetivamente, hipersingular.

Retornando ao objetivo inicial deste trabalho, conclui-se o MEC é perfeitamente aplicável para solução de problemas de flexão geometricamente não-linear de placas, fornecendo resultados plenamente competitivos com outras técnicas.

8.1 Recomendações para continuidade da pesquisa

Um número tópicos complementam ou dão continuidade ao presente trabalho. Na metodologia de solução numérica, poder-se-ia incorporar procedimentos de mudança de base adequados para a solução de casos como o ilustrado na figura 4-7, extendendo bastante o número de problemas que poderiam ser analisados com a abordagem proposta. Esta forma de resolver tais casos mantém um paralelo muito próximo com diversos elementos finitos de casca. Existem também algumas abordagens para consideração de espessura continuamente variável, e este é um melhoramento muito interessante do ponto de vista dos problemas práticos.

Uma expansão da família de elementos implementada também pode ser considerada, como a implementação de elementos de ordens mais altas e elementos de arco de círculo, o que favoreceria a análise de geometrias circulares, furos etc. Nesta mesma linha, o uso de elementos de contorno baseados em splines forneceria uma opção muito interessante para aplicação da tecnologia atualmente em uso nos geradores de malha. No entanto,

a determinação criteriosa da ordem das quadraturas para o melhor desempenho dos elementos implementados e um procedimento de seleção automática do número de pontos de integração são aspectos que devem ser considerados em trabalhos futuros. O método direto pode ser estendido para células de domínio, o que eliminaria o uso de diferenças finitas para cálculo das derivadas dos deslocamentos nos casos não-lineares.

A implementação de algoritmos efetivos para cálculo das integrais quase-singulares é imperativa, principalmente se for utilizada a versão hipersingular da formulação. Aplicados em conjunto com o método direto, estes algoritmos permitem o cálculo das variáveis em pontos muito próximos – e até mesmo sobre – o contorno.

Um outro aspecto que pode ser explorado é a otimização do recuo dos elementos de contorno e células de domínio. Se por um lado as partições descontínuas eliminaram uma série de inconvenientes relacionados ao uso de nó duplo e à necessidade de continuidade das funções de interpolação, por outro sabe-se que este parâmetro altera sensivelmente a precisão dos resultados.

Um estudo comparativo do comportamento não-linear de placas com modelo de Reissner e de Mindlin pode ser realizado com a formulação apresentada. Os comentários feitos no Capítulo 7 a respeito de condições de contorno e modelo de placa são importantes, pois apontam para algumas áreas de investigação interessantes. Os resultados do presente trabalho ilustram como as curvas carga \times deslocamento para placas semi-espessas no regime não-linear diferem das curvas conhecidas para placas finas. E além disso, sabe-se que outras variáveis influem muito na forma destas curvas, principalmente o coeficiente de Poisson e o fator κ^2 , o que não foi explorado aqui.

Finalmente, convém enfatizar que quaisquer das possibilidades sugeridas acima são passíveis de implementação imediata na biblioteca orientada a objetos desenvolvida neste trabalho, necessitando apenas de um projeto adequado de extensão ou reutilização das classes já implementadas.

Referências Bibliográficas

- Abramowitz, M. & Stegun, I. A. (1972), *Handbook of Mathematical Functions*, Dover.
- Adeli, H. & Yu, G. (1995), ‘An integrated computing environment for solution of complex engineering problems using the object-oriented programming paradigm and a blackboard architecture’, *Computers & Structures* **54**(2), 255–265.
- Aliabadi, M. H., Brebbia, C. A. & Mackerle, J. (1991), *Boundary Element Abstracts* **2**, 134–137.
- Antes, H. (1991), Static and dynamic analysis of reissner-mindlin plates, *in* D. E. Beskos, ed., ‘Boundary Element Analysis of Plates and Shells’, Springer-Verlag, pp. 312–340.
- Archer, G. C. (1996), Object-Oriented Finite Element Analysis, PhD thesis, University of California at Berkeley.
- Archer, G. C., Thewalt, C. & Fenves, G. L. (1996a), Facilitating analytical algorithm development for finite elements, *in* ‘Third Canadian Conference on Computing in Civil and Building Engineering’, Montreal, Quebec, pp. 558–564.
- Archer, G. C., Thewalt, C. & Fenves, G. L. (1996b), A new software architecture for finite element analysis, *in* ‘Third Congress on Computing in Civil Engineering’, Anaheim, California, pp. 683–689.
- Atluri, S. N. & Pipkins, D. S. (1991), Large deformation analysis of plates and shells, *in* D. E. Beskos, ed., ‘Boundary Element Analysis of Plates and Shells’, Springer-Verlag, pp. 141–166.
- Azizian, Z. G. & Dawe, D. J. (1985), ‘Geometrically nonlinear analysis of rectangular mindlin plates using the finite strip method’, *Computers & Structures* **21**, 423–436.

- Banerjee, B. & Datta, S. (1981), ‘A new approach to an analysis of large deflections of thin elastic plates’, *Int. J. Num. Meth. Engng.* **16**, 47–52.
- Banerjee, P. K. & Butterfield, R. (1981), *Boundary Element Methods in Engineering Science*, McGraw-Hill.
- Barker, A. L. (1996), *String, Array and Matrix Class in C++. Version 1.0*, www.cs.virginia.edu/~alb.
- Bathe, K. J. (1996), *Finite Element Procedures*, Prentice-Hall.
- Beck, R., Erdmann, B. & Roitzsch, R. (1995), Kaskade 3.0 - an object-oriented adaptive finite element code, Technical Report 95-4, Konrad-Zuse-Zentrum für Informationstechnik Berlin.
- Bencharif, N. & Ng, S. F. (1994a), ‘Linear and non-linear deflection analysis of thick rectangular plates - i. theoretical derivation’, *Computers & Structures* **50**, 757–761.
- Bencharif, N. & Ng, S. G. (1994b), ‘Linear and non-linear deflection analysis of thick rectangular plates - II. numerical applications’, *Computers & Structures* **50**, 763–776.
- Berger, H. M. (1955), ‘A new approach to the analysis of large deflection of plates’, *Journal of Applied Mechanics* pp. 465–472.
- Beskos, D. E., ed. (1991), *Boundary Element Analysis of Plates and Shells*, Springer-Verlag.
- Besson, J., Foerch, R., Cailletaut, G., Aazizou, K. & Hourlier, F. (1997), Large scale object oriented finite element code design.
- Bettig, B. P. & Han, R. P. S. (1996), ‘An object-oriented framework for interactive numerical analysis in a graphical user interface environment’, *Int. J. Num. Meth. Engng.* **39**, 2945–2971.
- Bezine, G. (1988), ‘A new boundary element method for bending of plates on elastic foundation’, *Int. J. Solids & Structures* **24**, 557–565.

- Bezine, G., Cimetiere, A. & Gelbert, J. P. (1985), ‘Unilateral buckling of thin elastic plates by the boundary integral equation method’, *Int. J. Num. Meth. Engng.* **21**, 2189–2199.
- Bollé, L. (1947a), ‘Contribution au problème linéaire de flexion d’une plaque élastique’, *Bulletin Technique de la Suisse Romande* (21), 281–285.
- Bollé, L. (1947b), ‘Contribution au problème linéaire de flexion d’une plaque élastique’, *Bulletin Technique de la Suisse Romande* (22), 293–298.
- Bonnet, M. (1997), ‘Differentiability of strongly singular and hypersingular boundary integral formulations with respect to boundary perturbations’, *Computational Mechanics* **19**, 240–246.
- Borland C++ Programmer’s Guide - Version 4.5* (1994), Borland International Inc.
- Brebbia, C. A. & Dominguez, J. (1989), *Boundary Elements - An Introductory Course*, Computational Mechanics Publications.
- Brebbia, C. A., Telles, J. C. F. & Wrobel, L. C. (1984), *Boundary Element Techniques - Theory and Applications in Engineering*, Springer-Verlag.
- Brokken, F. B. & Kubat, K. (1998), *C++ Annotations Version 4.3.1*, ICCE, University of Groningen, Netherlands.
- Brush, D. O. & Almroth, B. O. (1975), *Buckling of Bars, Plates and Shells*, McGraw-Hill.
- Budge, K. G. & Peery, J. S. (1993), ‘RHALE - a MMALE shock physics code written in C++’, *Int. J. Impact Engng.* **108**, 165–183.
- Bui, H. D. (1978), ‘Some remarks about the formulation of three-dimensional thermoelastoplastic problems by integral equations’, *Int. J. Solids & Structures* **14**, 935–939.
- Calderón, E. T. & Venturini, W. S. (1992), Análise de placas sobre fundação de comportamento não-linear através do método dos elementos de contorno, *in* ‘Anais Do XIII Cong. Ibero-Latino Americano Sobre Métodos Computacionais Em Engenharia’, Vol. 2, Porto Alegre, pp. 137–146.

- Cerrolaza, M. & Alarcón, E. (1989), ‘A bi-cubic transformation for the numerical evaluation of the Cauchy principal value integrals in boundary methods’, *Int. J. Num. Meth. Engng.* **28**, 987–999.
- Chaves, E. W. V., Fernandes, G. R. & Venturini, W. S. (1999), ‘Plate bending boundary element formulation considering variable thickness’, *Engineering Analysis with Boundary Elements* **25**, 405–418.
- Cheng, S. (1979), ‘Elasticity theory of plates and a refined theory’, *Journal of Applied Mechanics* **46**, 644–650.
- Commend, S. (1998), An object-oriented approach to nonlinear finite element programming, Technical report, LSC - Laboratoire de mécanique des structures et milieux continus, Département de Génie Civil, École Polytechnique Fédérale de Lausanne. Internal Report 98/07.
- Commend, S. & Zimmermann, T. (2001), ‘Object-oriented nonlinear finite element programming: A primer’, *Advances in Engineering Software* **32**, 611–628.
- Costa Jr., J. A. (1985), The Boundary Element Method Applied to Plate Problems, PhD thesis, Department of Civil Engineering, The University of Southampton.
- Costa Jr., J. A. (1986), O método dos elementos de contorno aplicado ao problema de placas, in ‘Anais Do VII Cong. Ibero-Latino Americano Sobre Métodos Computacionais Em Engenharia’, São Carlos, pp. 1–13.
- Costa Jr., J. A. (1987), ‘A boundary element algorithm for plate problems’, *J. of the Braz. Soc. Mechanical Sciences* **IX**, 157–171.
- Costa Jr., J. A. & Brebbia, C. A. (1985a), ‘The boundary element method applied to plates on elastic foundations’, *Engineering Analysis* **2**, 174–183.
- Costa Jr., J. A. & Brebbia, C. A. (1985b), Elastic buckling of plates using the boundary element method, in C. A. Brebbia & G. Maier, eds, ‘Boundary Elements VII - Proc. 7th Int. Conf.’, Springer-Verlag, Villa Olmo, pp. 4.29–4.42.
- Craig, R. J. (1987), ‘Finite difference solutions of Reissner’s plate equations’, *ASCE Journal of Engineering Mechanics* **113**, 31–48.

- Cross, J. T., Masters, I., Sukirman, Y. & Lewis, R. W. (1997), Object-oriented programming techniques for finite element methods in heat transfer, *in* R. W. Lewis & J. T. Cross, eds, 'Proc. 10th Conf. For Num. Meth. Thermal Problems', Swansea, pp. 757–766.
- Davies, R. B. (1997), *Documentation for Newmat09, a Matrix Library in C++*, robertd@netlink.co.nz.
- de Barcellos, C. S. & Monken e Silva, L. H. (1987), A boundary element formulation for the mindlin's plate model, *in* C. A. Brebbia & W. S. Venturini, eds, 'Proc. III Int. Conf. On Boundary Element Technology', Computational Mechanics Publ., Southampton, pp. 123–130.
- de Barcellos, C. S. & Westphal Jr., T. (1992), Reissner/Mindlin's plate models and the boundary element method, *in* C. A. Brebbia & M. S. Ingber, eds, 'Proc. 7th Conf. On Boundary Element Technology', Computational Mechanics Publ., Albuquerque, pp. 589–604.
- Devloo, P. R. B. (1997), 'Pz: An object oriented environment for scientific programming', *Comput. Methods Appl. Mech. Engrg.* **150**, 133–153.
- Devloo, P. R. B., Duarte, C. A. M. & Noel, A. T. (1995), On the implementation of the p-adaptive finite element method using the object oriented programming philosophy, *in* R. M. Daledone, ed., 'CILAMCE XVI - Proc. Of the 16th Iberian Latin American Congress on Comp. Meth. In Engineering', Universidade Federal do Paraná, Curitiba.
- Dhatt, G. & Touzot, G. (1984), *The Finite Element Method Displayed*, John Wiley Sons.
- Dirgantara, T. & Aliabadi, M. H. (1999), 'A new boundary element formulation for shear deformable shells analysis', *Int. J. Num. Meth. Engng.* **45**, 1257–1275.
- Duarte, C. A. M. (1995), Object-oriented parallel implementation of the element by element conjugate gradient method. TICAM - Texas Inst. for Comput. and Appl. Mathematics, Austin, Texas.

- Duarte, J. B. P. & Palermo Jr., L. (1999), Aplicação do método dos elementos de contorno às placas finas incluindo o efeito da não-linearidade geométrica, *in* ‘Anais Do XV Cong. Bras. Engenharia Mecânica’, Águas de Lindóia.
- Dubois-Pèlerin, Y. & Pegon, P. (1998), ‘Linear constraints in object-oriented finite element programming’, *Comput. Methods Appl. Mech. Engrg.* **154**, 31–39.
- Dubois-Pèlerin, Y. & Zimmermann, T. (1993), ‘Object-oriented finite element programming: III. an efficient implementation in C++’, *Comput. Methods Appl. Mech. Engrg.* **108**, 165–183.
- Dubois-Pèlerin, Y., Zimmermann, T. & Bomme, P. (1992), ‘Object-oriented finite element programming: II. a prototype program in smalltalk’, *Comput. Methods Appl. Mech. Engrg.* **98**, 361–397.
- Dumont, N. A. (1992), A unified technique for the evaluation of general singular integrals, *in* ‘CILAMCE XIII - Proc. of the 13th Iberian Latin American Congress on Comp. Meth. in Engineering, Vol. 2’, pp. 194–203.
- El-Zafrany, A., Fadhil, S. & Al-Hosani, K. (1994), ‘A new fundamental solution for boundary element analysis of thin plates on winkler foundation’, *Int. J. Num. Meth. Engrg.* **38**, 887–903.
- El-Zafrany, A., Fadhil, S. & Debbih, M. (1995), ‘An efficient approach for boundary element bending of thin and thick plates’, *Computers & Structures* **56**, 565–576.
- Eyheramendy, D. & Zimmermann, T. (1996a), ‘Object-oriented finite element programming: An interactive environment for symbolic derivations, application to an initial boundary value problem’, *Advances in Engineering Software* **27**, 3–10.
- Eyheramendy, D. & Zimmermann, T. (1996b), ‘Object-oriented finite elements ii. a symbolic environment for automatic programming’, *Comput. Methods Appl. Mech. Engrg.* **132**, 277–304.
- Eyheramendy, D. & Zimmermann, T. (1998), ‘Object-oriented finite elements iii. theory and application of automatic programming’, *Comput. Methods Appl. Mech. Engrg.* **154**, 41–68.

- Feijóo, R. A., Guimarães, A. C. S. & Fancello, E. A. (1991), Algumas experiencias en la programación orientada por objetos y su aplicación en el método de los elementos finitos, Technical Report 015/91, LNCC - Laboratório Nacional de Computação Científica.
- Feng, Z. Q. (1995), ‘2D of 3D frictional contact algorithms and applications in a large deformation context’, *Comm. Numer. Meth. Engng.* **11**, 409–416.
- Fernandes, G. R. & Venturini, W. S. (1999), Análise de pavimentos de eficiência através do método dos elementos de contorno, *in* P. M. Pimenta, R. M. L. R. F. Brasil & E. S. Almeida, eds, ‘Computational Methods in Engineering ’99’.
- Foerch, R., Besson, J., Cailletaut, G. & Pilvin, P. (1995), Polymorphic constitutive equations in finite element codes, Technical Report 335, Math.-Nat. Fakultät, Universität Augsburg.
- Forde, B. W. R., Foschi, R. O. & Steimer, S. F. (1990), ‘Object-oriented finite element analysis’, *Computers & Structures* **34**(3), 355–374.
- Frakes, J. P. & Simmonds, J. G. (1985), ‘Asymptotic solutions of the von Kármán equations for a circular plate under a concentrated load’, *Journal of Applied Mechanics* **52**, 326–330.
- Frangi, A. & Guiggiani, M. (2000), ‘A direct approach for boundary integral equations with high-order singularities’, *Int. J. Num. Meth. Engng.* **49**, 871–898.
- Fu, K. C. & Harb, A. I. (1990), ‘Integral equation method for spherical shell under axisymmetric loads’, *ASCE Journal of Engineering Mechanics* **116**, 309–342.
- Fung, Y. C. (1965), *Foundations of Solid Mechanics*, Prentice-Hall.
- Gospodinov, G. & Ljutskanov, D. (1982), ‘The boundary element method applied to plates’, *Appl. Math. Modelling* **6**, 237–244.
- Green, A. E. (1949), ‘On Reissner’s theory of bending of elastic plates’, *Quarterly of Applied Mathematics* **VII**, 223–228.

- Guiggiani, M. (1992), ‘Computing principal-value integrands in 3-d BEM for time-harmonic elastodynamics - a direct approach’, *Communications in Applied Numerical Methods* **8**, 141–149.
- Guiggiani, M. (1995), ‘Hypersingular boundary integral equations have an additional free term’, *Computational Mechanics* **16**, 245–248.
- Guiggiani, M. (1998), Formulation and numerical treatment of boundary integral equations with hypersingular kernels, *in* B. Sladek & J. Sladek, eds, ‘Singular Integrals in Boundary Element Methods’, Computational Mech. Publ., chapter 3, pp. 85–124.
- Guiggiani, M. & Casalini, P. (1987), ‘Direct computation of Cauchy principal value integrals in advanced boundary elements’, *Int. J. Num. Meth. Engng.* **24**, 1711–1720.
- Guiggiani, M. & Gigante, A. (1990), ‘A general algorithm for multidimensional Cauchy principal value integrals in the boundary element method’, *Journal of Applied Mechanics* **57**, 906–915.
- Guiggiani, M., Krishnasamy, G., Rudolphi, T. J. & Rizzo, F. J. (1992), ‘A general algorithm for the numerical solution of hypersingular boundary integral equations’, *Journal of Applied Mechanics* **59**, 604–614.
- HARWELL Subroutine Library Specification: EB06A* (1978). Boeing Aircraft Company.
- Hedegal, O. (1994), Object-Oriented Structuring of Finite Elements, PhD thesis, Aalborg University.
- Hekmatpour, S. (1990), *C++ A Guide for C Programmers*, Prentice-Hall.
- Hörmander, L. (1964), *Linear Partial Differential Operators*, Springer-Verlag.
- Huang, Q. & Cruse, T. A. (1993), ‘Some notes on singular integral techniques in boundary element analysis’, *Int. J. Num. Meth. Engng.* **36**, 2643–2659.
- Hughes, T. J. R. (1987), *Finite Element Method - Linear Static and Dynamic Finite Element Analysis*, Prentice-Hall.
- Jaswon, M. A. & Maiti, M. (1968), ‘An integral equation formulation of plate bending problems’, *J. Eng. Math.* **2**, 83–93.

- Jaswon, M. A., Maiti, M. & Symm, G. T. (1967), ‘Numerical biharmonic analysis and some applications’, *Int. J. Solids & Structures* **3**, 309–332.
- Jeremić, B. & Sture, S. (1998), ‘Tensor objects in finite element programming’, *Int. J. Num. Meth. Engng.* **41**, 113–126.
- Jianguo, W., Xiuxi, W. & Maokuang, H. (1993), ‘A boundary integral equation formulation for thick plates on a winkler foundation’, *Computers & Structures* **49**, 179–185.
- Jinmu, L. & Shuyao, L. (1996), ‘Geometrically nonlinear analysis of the shallow shell by the displacement based boundary element formulation’, *Engineering Analysis with Boundary Elements* **18**, 63–70.
- Johnston, P. R. & Elliott, D. (2001), ‘A generalisation of Telles’ method for evaluation weakly singular boundary element integrals’, *Journal of Computational and Applied Mathematics* **131**, 223–241.
- Jun, L., Beer, G. & Meek, J. L. (1985), The application of double exponential formulas in the boundary element method, *in* C. A. Brebbia & G. Maier, eds, ‘Boundary Elements VII - Proc. of the 7th Int. Conference, Vol. II’, Springer-Verlag, pp. 13.3–13.16.
- Kamiya, N. & Sawaki, Y. (1982a), ‘An approximate finite deflection analysis of heated elastic plate by the boundary element method’, *Appl. Math. Modelling* **6**, 23–27.
- Kamiya, N. & Sawaki, Y. (1982b), ‘An integral equation approach to finite deflection of elastic plates’, *Int. J. Non-Lin. Mech.* **17**, 187–194.
- Kamiya, N. & Sawaki, Y. (1982c), ‘Integral equations formulation for non-linear bending of plates - formulation by weighted residual method’, *ZAMM - Z. Angew. Math. Mech.* **62**, 651–655.
- Kamiya, N. & Sawaki, Y. (1985), An efficient BEM for some inhomogeneous and nonlinear problems, *in* C. A. Brebbia & G. Maier, eds, ‘Proc. 7th Int. Seminar BEM Eng.’, Computational Mechanics Publ., Villa Olmo, pp. 13.59–13.68.

- Kamiya, N. & Sawaki, Y. (1986), An alternative boundary element analysis of plates resting on elastic foundation, *in* C. A. Brebbia & M. Tanaka, eds, 'Boundary Elements VIII - Proc. 8th Int. Conf.', Vol. II, Springer-Verlag, Tokyo, pp. 561–570.
- Kant, T. & Hinton, E. (1980), Numerical analysis of rectangular mindlin plates by the segmentation method, Technical Report C/R/365/80, Civil Engineering Department, University of Wales.
- Karam, V. J. (1986), Aplicação do método dos elementos de contorno à teoria de reissner para flexão de placas, Master's thesis, COPPE, Rio de Janeiro.
- Karam, V. J. & Telles, J. C. F. (1986), O método dos elementos de contorno aplicado à teoria de reissner para flexão de placas, *in* 'Anais Do VIII Cong. Ibero-Latino Americano Sobre Métodos Computacionais Em Engenharia', São Carlos, pp. 29–42.
- Karam, V. J. & Telles, J. C. F. (1988), 'On boundary elements for reissner's plate theory', *Engineering Analysis* **5**, 21–27.
- Katsikadelis, J. T. & Nerantzaki, M. S. (1993), Non-linear analysis of plates by the analog equation method, *in* C. A. Brebbia & J. J. Rencis, eds, 'Boundary Elements XV - Proc. 15th Int. Conf.', Vol. 2, Computational Mechanics Publ. / Elsevier Applied Science, Worcester, pp. 165–178.
- Katsikadelis, J. T. & Yotis, A. J. (1993), 'A new boundary element solution of thick plates modelled by Reissner's theory', *Engineering Analysis with Boundary Elements* **12**, 65–74.
- Kita, E. & Kamiya, N. (1994), 'Subregion boundary element method', *J. of the Japan Society of Mechanical Engineering - Series A* **37**, 366–372.
- Kleiber, M. (1989), *Incremental Finite Element Modelling in Non-Linear Solid Mechanics*, Ellis Horwood.
- Koiter, W. T. (1960), A consistent first approximation in the general theory of thin elastic shells, *in* W. T. Koiter, ed., 'Theory of Thin Shells', North-Holland, pp. 19–23.
- Koiter, W. T. (1967), General equations of elastic stability for thin shells, *in* W. T. Koiter, ed., 'Proc. Symp. Theory of Shells', pp. 187–228.

- Kompis, V. (1978), Boundary integral equation method for three dimensional elastostatic problems and a formulation for the problem for large displacements, *in* ‘Proc 14th Yugoslavian Congr. Rac. Mech.’, pp. 113–120.
- Kong, X. A. (1996), ‘A data design approach for object-oriented fem programs’, *Computers & Structures* **61**(3), 503–513.
- Kong, X. A. & Chen, D. P. (1995), ‘An object-oriented design of fem programs’, *Computers & Structures* **57**(1), 157–166.
- Krommer, A. R. & Ueberhuber, C. H. (1998), *Computational Integration*, SIAM.
- Kutt, H. R. (1975a), ‘The numerical evaluation of principal value integrals by finite-part integration’, *Numer. Math.* **24**, 205–210.
- Kutt, H. R. (1975b), On the numerical evaluation of finite-part integrals involving an algebraic singularity, Technical Report WISK 179, National Research Institute for Mathematical Sciences, Pretoria.
- Kutt, H. R. (1975c), Quadrature formulae for finite-part integrals, Technical Report WISK 178, National Research Institute for Mathematical Sciences, Pretoria.
- Lachat, J. C. & Watson, J. O. (1976), ‘Effective numerical treatment of boundary integral equations: Formulation of three-dimensional elastostatics’, *Int. J. Num. Meth. Engng.* **10**, 991–1005.
- Lage, C. (1998), ‘The application of object-oriented methods to boundary elements’, *Comput. Methods Appl. Mech. Engrg.* **157**, 205–213.
- Langtangen, H. P. (1996), Details of finite element programming in diffpack, Technical report, Department of Mathematics, University of Oslo.
- Lee, K. H., Lim, G. T. & Wang, C. M. (2001), ‘Thick Lévy plates re-visited’, *Int. J. Solids & Structures* .
- Lévy, S. (1942a), Bending of rectangular plates with large deflections, Report no. 737, NACA.

- Lévy, S. (1942*b*), Square plate with clamped edges under normal pressure producing large deflections, Tech. note 847, NACA.
- Lim, S. P., Lee, K. H., Chow, S. T. & Senthilnathan, N. R. (1988), ‘Linear and nonlinear bending of shear-deformable plates’, *Computers & Structures* **30**, 945–952.
- Lin, J., Duffield, R. C. & Shih, H. R. (1999), ‘Buckling analysis of elastic plates by boundary element method’, *Engineering Analysis with Boundary Elements* **23**, 131–137.
- Long, S. Y., Brebbia, C. A. & Telles, J. C. F. (1988), ‘Boundary element bending analysis of moderately thick plates’, *Engineering Analysis* **5**, 64–74.
- Love, A. E. H. (1888), ‘The small vibrations and deformation of a thin elastic shell’, *Philosophical Trans. of the Royal Society of London* **179**, 491–546.
- Lu, J., White, D. W., Chen, W. F. & Dunsmore, H. E. (1995), ‘A matrix class library in C++ for structural engineering computing’, *Computers & Structures* **55**(1), 95–111.
- Mackerle, J. (2000), ‘Object-oriented techniques in FEM and BEM. a bibliography (1996-1999)’, *Finite Elements in Analysis and Design* **36**, 189–196.
- Mackie, R. I. (1992), ‘Object oriented programming of the finite element method’, *Int. J. Num. Meth. Engng.* **35**, 425–436.
- Mackie, R. I. (1998), ‘An object-oriented approach to fully interactive finite element software’, *Advances in Engineering Software* **29**(2), 139–149.
- Malvern, L. E. (1969), *Introduction to the Mechanics of a Continuous Medium*, Prentice-Hall.
- Manolis, G. D., Beskos, D. E. & Pineros, M. F. (1986), ‘Beam and plate stability by boundary elements’, *Computers & Structures* **22**, 917–923.
- Marczak, R. J. (1993), Alguns desenvolvimentos do método dos elementos de contorno aplicado a placas semi-espessas, Master’s thesis, Universidade Federal de Santa Catarina, Florianópolis.

- Marczak, R. J. (1994), Boundary element method linear elastic stability of thin and moderately thick plates, *in* ‘Anais Do XV Cong. Ibero-Latino Americano Sobre Mét. Computacionais Em Engenharia’, Vol. I, Belo Horizonte, pp. 651–660.
- Marczak, R. J. (1995a), ‘Análise de estabilidade linear elástica de placas de mindlin-reissner através do método dos elementos de contorno’, *Revista Internacional de Metodos Numericos para Calculo y Diseño en Ingeniaria* **11**, 625–635.
- Marczak, R. J. (1995b), Buckling analysis of variable thickness plates using a BEM formulation with subregions, *in* J. D. Riera & M. M. Rocha, eds, ‘Proc. 13th Conf. Structural Mech. Reactor Technology’, Vol. II, Porto Alegre, pp. 759–764.
- Marczak, R. J. (1996), ‘Analytical integration of membrane-bending coupling terms in non-linear boundary element analysis of mindlin-reissner plates’, *J. of the Braz. Soc. Mechanical Sciences* **XVIII**, 218–226.
- Marczak, R. J. (1998), A boundary element formulation for linear and non-linear bending of plates, *in* S. Idhelson, ed., ‘Computational Mechanics - New Trends and Application’, Int. Assoc. Comp. Mech.
- Marczak, R. J. & de Barcellos, C. S. (1993), Análise de estabilidade linear elástica de placas de mindlin e reissner através do método dos elementos de contorno, *in* ‘Anais Do XII Cong. Bras. Engenharia Mecânica’, Brasília.
- Masters, I. (1997), Parallel Heat Transfer, PhD thesis, University of Wales, Swansea.
- McKenna, F. T. (1997), Object-Oriented Finite Element Programming: Frameworks for Analysis, Algorithms and Parallel Computing, PhD thesis, University of California at Berkeley.
- Menétrey, P. & Zimmermann, T. (1993), ‘Object-oriented non-linear finite element analysis: Application to j2 plasticity’, *Computers & Structures* **49**(5), 767–777.
- Mikhlin, S. G. (1962), ‘Singular integral equations’, *Amer. Math. Soc. Transl. Series 1*, **10**, 84–198.
- Miller, G. R. (1991), ‘An object-oriented approach to structural analysis and design’, *Computers & Structures* **40**(1), 75–82.

- Miller, G. R. (1993), ‘Coordinate-free isoparametric elements’, *Computers & Structures* **49**(6), 1027–1035.
- Mindlin, R. D. (1951), ‘Influence of rotatory inertia and shear on flexural motions of isotropic, elastic plates’, *Journal of Applied Mechanics* **18**, 31–38.
- Mindlin, R. D. (1955), An introduction to the mathematical theory of vibrations of elastic plates, Technical report, U.S. Army Signal Corps Engineering Laboratories, Fort Monmouth, N.J.
- Monken e Silva, L. H. (1988), Novas Formulações Integrais Para Problemas Da Mecânica, PhD thesis, Universidade Federal de Santa Catarina, Florianópolis.
- Monken e Silva, L. H. & de Barcellos, C. S. (1985), Review sobre o método dos elementos de contorno aplicado para solução de problemas de placas elásticas, *in* ‘Anais Do VIII Cong. Bras. Engenharia Mecânica’, São José dos Campos, pp. 457–460.
- Muskhelishvili, N. I. (1963), *Some Basic Problems of the Mathematical Theory of Elasticity*, Noordhoff.
- Naghdi, P. M. (1957), ‘On the theory of thin elastic shells’, *Quarterly of Applied Mathematics* **XIV**, 369–380.
- Nahlik, R. & Bialecki, R. (1983), ‘A new method of numerical evaluation of singular integrals occurring in two-dimensional BIEM’, *Appl. Math. Modelling* **7**, 169–172.
- Nerantzaki, M. S. & Katsikadelis, J. T. (1996a), ‘An analog equation solution to dynamic analysis of plates with variable thickness’, *Engineering Analysis with Boundary Elements* **17**, 145–152.
- Nerantzaki, M. S. & Katsikadelis, J. T. (1996b), ‘Buckling of plates with variable thickness - an analog equation solution’, *Engineering Analysis with Boundary Elements* **18**, 149–154.
- Niwa, Y., Kobayashi, M. & Kitahara, M. (1982), Determination of eigenvalues by boundary element methods, *in* P. K. Banerjee & R. P. Shaw, eds, ‘Developments in Boundary Element Methods’, Vol. 2, Elsevier Applied Science, pp. 143–176.

- Noronha, M., Wagner, M., Wenzel, W., J. Wirnitzer, N. A. D. & Gaul, L. (1996), On a robust, object-oriented code for the implementation of conventional and hybrid boundary element methods, *in* ‘Anais Do IX SIBRAT - Simp. Bras. Tubulações E Vasos de Pressão’, Rio de Janeiro.
- Oden, J. T. & Reddy, J. N. (1976), *An Introduction to the Mathematical Theory of Finite Elements*, John Wiley Sons.
- Ohga, M. & Shigematsu, T. (1988), ‘Bending analysis of plates with variable thickness by boundary element-transfer matrix method’, *Computers & Structures* **28**, 635–640.
- Olsson, A. (1998), ‘An object-oriented implementation of structural path-following’, *Comput. Methods Appl. Mech. Engrg.* **161**, 19–47.
- Paget, D. F. (1981), ‘The numerical evaluation of Hadamard finite-part integrals’, *Numer. Math.* **36**, 447–453.
- Partridge, P. W., Brebbia, C. A. & Wrobel, L. C. (1992), *The Dual Reciprocity Boundary Element Method*, Computational Mechanics Publ.
- Patterson, T. N. L. (1968), ‘The optimum addition of points to quadrature formulae’, *Math. Comp.* **22**, 847–856.
- Perez, M. M. & Wrobel, L. C. (1991), Aplicação de soluções fundamentais isotrópicas na análise de problemas potenciais ortotrópicos, *in* ‘Anais Do XI Congresso Brasileiro de Engenharia Mecânica’, Assoc. Bras. Ciências Mecânicas, São Paulo, pp. 145–148.
- Pica, A. & Wood, R. D. (1980), ‘Postbuckling behaviour of plates and shells using mindlin shallow shell formulation’, *Computers & Structures* **12**, 759–768.
- Pica, A., Wood, R. D. & Hinton, E. (1980), ‘Finite element analysis of geometrically nonlinear plate behaviour using a mindlin formulation’, *Computers & Structures* **11**, 203–215.
- Pidaparti, R. M. V. & Hudli, A. V. (1993), ‘Dynamic analysis of structures using object-oriented techniques’, *Computers & Structures* **49**(1), 149–156.

- Pietraszkiewicz, W. (1979), *Finite Rotations and Lagrangian Description in the Non-Linear Theory of Shells*, Polish Scientific.
- Pietro, I., Albán, A. L. & Garrido, J. A. (1999), ‘A 2d analysis for geometrically non-linear elastic problems using the BEM’, *Engineering Analysis with Boundary Elements* **23**, 247–256.
- Pozo, R. (1997), *MV++ v. 1.5a: Matrix / Vector Class - Reference Guide*, Appl. and Comput. Math. Div., National Institute of Standards and Technology.
- Qin, Q. H. (1993), ‘Nonlinear analysis of reissner plates on an elastic foundation by the BEM’, *Int. J. Solids & Structures* **30**, 3101–3111.
- Ramm, E. & Stegmüller, H. (1982), The displacement finite element method in nonlinear buckling analysis of shells, *in* E. Ramm, ed., ‘Buckling of Shells - Proceedings of a State-of-the-Art Colloquium’, pp. 201–235.
- Rao, G. V., Venkataramana, J. & Raju, I. S. (1974), ‘A high precision triangular plate bending element for the analysis of thick plates’, *Nucl. Engng. Des.* **30**, 408–412.
- Rao, N. R. & Valsarajan, K. V. (1986), ‘An integral-equation solution for the finite deflection of clamped skew sandwich plates’, *Computers & Structures* **22**, 665–668.
- Rashed, Y. F., Aliabadi, M. H. & Brebbia, C. A. (1997), ‘On the evaluation of the stresses in the BEM for reissner plate-bending problems’, *Appl. Math. Modelling* **21**, 155–163.
- Rashed, Y. F., Aliabadi, M. H. & Brebbia, C. A. (1998a), ‘The boundary element method for thick plates on a winkler foundation’, *Int. J. Num. Meth. Engng.* **41**, 1435–1462.
- Rashed, Y. F., Aliabadi, M. H. & Brebbia, C. A. (1998b), ‘Hypersingular boundary element formulation for reissner plates’, *Int. J. Solids & Structures* **35**, 2229–2249.
- Rashed, Y. F., Aliabadi, M. H. & Brebbia, C. A. (1999), ‘A boundary element formulation for a reissner plate on a paternak foundation’, *Computers & Structures* **70**, 515–532.
- Reddy, J. N. (1984a), *Energy and Variational Methods in Applied Mechanics*, John Wiley & Sons.

- Reddy, J. N. (1984*b*), ‘A refined nonlinear theory of plates with transverse shear deformation’, *Int. J. Solids & Structures* **20**, 881–896.
- Reissner, E. (1944), ‘On the theory of bending of elastic plates’, *Journal of Mathematical Physics* **23**, 184–191.
- Reissner, E. (1945), ‘The effect of transverse shear deformation on the bending of elastic plates’, *Journal of Applied Mechanics* **12**, A69–A77.
- Reissner, E. (1947), ‘On the bending of elastic plates’, *Quarterly of Applied Mathematics* **V**, 55–68.
- Reissner, E. (1950), ‘On a variational theorem in elasticity’, *Journal of Mathematical Physics* **29**, 90–95.
- Reissner, E. (1975), ‘On transverse bending of plates including the effect of transverse shear deformation’, *Int. J. Solids & Structures* **11**, 569–573.
- Reissner, E. (1980), ‘On the analysis of first and second-order shear deformation effects for isotropic elastic plates’, *Journal of Applied Mechanics* **47**, 939–961.
- Reissner, E. (1983), ‘A twelfth order theory of transverse bending of transversely isotropic plates’, *ZAMM - Z. Angew. Math. Mech.* **63**, 2850289.
- Reissner, E. (1985), ‘Reflections of the theory of elastic plates’, *Applied Mechanics Review* **38**, 1453–1464.
- Ren, D. & Fu, K. C. (1991), A boundary element technique on the solution of cylindrical shell bending problems, *in* C. A. Brebbia & G. S. Gipson, eds, ‘Boundary Elements XIII - Proc. 13th Int. Conf.’, Computational Mechanics Publ. / Elsevier Applied Science, Tulsa, pp. 499–510.
- Ren, D. & Fu, K. C. (1998), ‘Analysis of shallow cylindrical shell by boundary element method’, *Int. J. Solids & Structures* **35**, 1–17.
- Ribeiro, G. O. & Venturini, W. S. (1991*a*), Aplicação do método dos elementos de contorno para análise de placas submetidas a variação de temperatura, *in* E. B. L. C.

- e F. A. Paulo, ed., ‘Anais Do Simpósio Mineiro de Mecânica Computacional’, Universidade Federal de Minas Gerais, Belo Horizonte, pp. 134–144.
- Ribeiro, G. O. & Venturini, W. S. (1991*b*), Consideração de cargas distribuídas em linha e concentradas no cálculo de placas pelo método dos elementos de contorno, *in* E. B. L. C. e F. A. Paulo, ed., ‘Anais Do Simpósio Mineiro de Mecânica Computacional’, Universidade Federal de Minas Gerais, Belo Horizonte, pp. 123–133.
- Rosen, D. & Cormack, D. E. (1994), ‘The continuation approach for singular and near-singular integration’, *Engineering Analysis with Boundary Elements* **13**, 99–113.
- Rosen, D. & Cormack, D. E. (1995), ‘The continuation approach: A general framework for the analysis and evaluation of singular and near-singular integrals’, *SIAM J. Appl. Math.* **55**, 723–762.
- Rudolphi, T. J. (1983), ‘An implementation of the boundary element method for zoned media with stress discontinuities’, *Int. J. Num. Meth. Engng.* **19**, 1–15.
- Rumbaugh, J., Blaha, M., Premerhani, W., Eddy, F. & Lorensen, W. (1991), *Object-Oriented Modeling and Design*, Prentice-Hall.
- Rushton, K. R. (1970), ‘Large deflexion of plates with initial curvature’, *Int. J. Mech. Sci.* **12**, 1037–1051.
- Salerno, V. L. & Goldberg, M. A. (1960), ‘Effect of shear deformation on the bending of rectangular plates’, *Journal of Applied Mechanics* **27**, 54–58.
- Santiago, J. A. F. & Telles, J. C. F. (1997), A boundary element optimized block manipulation for subregion problems, *in* ‘Anais Do XVIII Cong. Ibero-Latino Americano Sobre Mét. Computacionais Em Engenharia’, Vol. 1, Brasília, pp. 245–252.
- Santiago, J. A. F., Telles, J. C. F. & Valentin, V. A. (1999), ‘An optimized block matrix manipulation for boundary elements with subregions’, *Advances in Engineering Software* **30**, 701–713.
- Sapountzakis, E. J. & Katsikadelis, J. T. (1992), ‘Unilaterally supported plates on elastic foundations by the boundary element method’, **59**, 580–586.

- Scholz, S. P. (1992), ‘Elements of an object-oriented fem++ program in C++’, *Computers & Structures* **43**(3), 517–529.
- Shi, G. (1990), ‘Flexural vibration and buckling analysis of orthotropic plates by the boundary element method’, *Int. J. Solids & Structures* **26**, 1351–1370.
- Silva, E. J., Mesquita, P., Saldanha, R. R. & Palmeira, P. F. M. (1994), ‘An object-oriented finite element program for electromagnetic field computation’, *IEEE Transactions on Magnetics* **30**, 3618–3621.
- Simos, N. & Sadegh, A. M. (1989), ‘An indirect boundary integral equation applied to nonshallow spherical shell problems with arbitrary boundary constraints’, *Journal of Applied Mechanics* **56**, 918–925.
- Smith, R. N. L. (2000), ‘Direct Gauss quadrature formulae for logarithmic singularities on isoparametric elements’, *Engineering Analysis with Boundary Elements* **24**, 161–167.
- Sokolnikoff, I. S. & Redheffer, R. M. (1958), *Mathematical Physics & Modern Engineering*, McGraw-Hill.
- Sotelino, E. D., Chen, W. F. & White, D. W. (1998), Future challenges for simulation in structural engineering, in S. Idelsohn, E. O. Nate & E. Dvorkin, eds, ‘Computational Mechanics - New Trends and Applications’, CIMNE, Barcelona, Spain.
- Srinivas, S. & Rao, A. K. (1973), ‘Flexure of thick rectangular plates’, *Journal of Applied Mechanics* **39**, 289–299.
- Standard Template Library Programmer’s Guide* (1999), Technical report, Silicon Graphics Computer Systems, Inc.
- Stein, E. & Wendland, W., eds (1988), *Finite Element and Boundary Element Techniques From Mathematical and Engineering Point of View*, Springer-Verlag.
- Stroud, A. H. & Secrest, D. (1966), *Gaussian Quadrature Formulas*, Prentice-Hall.
- Stroustrup, B. (1986), *The C++ Programming Language*, Addison-Wesley.
- Stroustrup, B. (1991), What is object-oriented programming? (1991 revised version), in ‘Proc. 1st European Software Festival’, Murray Hill, New Jersey.

- Syngellakis, S. & Elsein, A. (1994), ‘Plate buckling loads by the boundary element method’, *Int. J. Num. Meth. Engng.* **37**, 1763–1778.
- Syngellakis, S., Elsein, A. & Walker, A. C. (1991), ‘Comparision between experimental and boundary element predictions of plate buckling behavior’, *Engineering Analysis with Boundary Elements* **8**, 103–108. Short Communication.
- Tai, H., Katayama, T. & Sekyia, T. (1982), ‘Buckling analysis by influence function’, *Mech. Res. Commun.* **9**, 139–144.
- Tanaka, M. (1982), Integral equation approach to small and large displacement of thin elastic plates, *in* C. A. Brebbia, ed., ‘Boundary Element Method in Eng. - Proc. 4th Int. Seminar’, Springer-Verlag, pp. 526–539.
- Tanaka, M. (1984), Large deflection analysis of thin elastic plates, *in* P. K. Banerjee & S. Mukherjee, eds, ‘Developments in Boundary Element Methods’, Vol. 3, Elsevier Applied Science, pp. 115–136.
- Tanaka, M. (1986), Elastic buckling analysis of assembled plate structures by boundary element method, *in* C. A. Brebbia & M. Tanaka, eds, ‘Boundary Elements VIII - Proc. 8th Int. Conf.’, Vol. II, Springer-Verlag, Japan, pp. 547–559.
- Tanaka, M., Matsumoto, T. & Zheng, Z. D. (1996), ‘Incremental analysis of finite deflection of elastic plates via boundary-domain-element method’, *Engineering Analysis with Boundary Elements* **17**, 123–131.
- Tanaka, M., Matsumoto, T. & Zheng, Z. D. (1999), ‘Application of the boundary-domain-element method to the Pre/Post-buckling problem of von kármán plates’, *Engineering Analysis with Boundary Elements* **23**, 399–404.
- Tanaka, M., Sladek, V. & Sladek, J. (1994), ‘Regularization techniques applied to boundary element methods’, *Appl. Mech. Rev.* **47**, 457–499.
- Telles, J. C. F. (1987), ‘A self-adaptive co-ordinate transformation for efficient numerical evaluation of general boundary element integrals’, *Int. J. Num. Meth. Engng.* **24**, 959–973.

- Telles, J. C. F. & Oliveira, R. F. (1994), ‘Third degree polynomial transformation for boundary element integrals: Further improvements’, *Engineering Analysis with Boundary Elements* **13**, 135–141.
- Tiller, M. M. (1993), Implementation of an object-oriented finite element code, Technical report, Department of Mechanical and Industrial Engineering, University of Illinois.
- Timoshenko, S. P. & Woinowski-Krieger, S. (1970), *Theory of Plates and Shells*, 2nd edn, McGraw-Hill.
- Tosaka, N. & Miyake, S. (1986), Geometrically nonlinear analysis of shallow spherical shell using an integral equation method, *in* C. A. Brebbia & M. Tanaka, eds, ‘Boundary Elements VIII - Proc. 8th Int. Conf.’, Vol. II, Springer-Verlag, Tokyo, pp. 537–546.
- Tottenham, H. (1979), The boundary element method for plates and shells, *in* P. K. Banerjee & R. Butterfield, eds, ‘Developments in Boundary Element Methods’, Vol. 1, Elsevier Applied Science, pp. 173–205.
- Toumi, K. & Jezequel, L. (1993), A boundary element method approach for the vibration of stiffened plates, *in* C. A. Brebbia & J. J. Rencis, eds, ‘Boundary Elements XV - Proc. 15th Int. Conf.’, Vol. 2, Computational Mechanics Publ. / Elsevier Applied Science, Worcester, pp. 391–406.
- Turvey, G. J. & Osman, M. Y. (1990), ‘Elastic large deflection analysis of isotropic rectangular mindlin plates’, *Int. J. Mech. Sci.* **32**, 315–328.
- van der Weeën, F. (1981), Randintegraalvergelijkingen Voor Het Plaatmodel Van Reissner, PhD thesis, State University of Ghent, Belgium.
- van der Weeën, F. (1982a), ‘Application of the boundary integral equation method to reissner’s plate model’, *Int. J. Num. Meth. Engng.* **18**, 1–10.
- van der Weeën, F. (1982b), Application of the direct boundary element method to reissner’s plate model, *in* C. A. Brebbia, ed., ‘Proc. 4th Int. Sem. BEM Eng.’, Southampton, pp. 487–499.

- Vilman, O. (1990), The Boundary Element Method Applied in Mindlin Plate Bending Analysis, PhD thesis, Department of Structural Engineering, Technical University of Denmark, Denmark.
- von Kármán, T. (1910), Festigkeitsprobleme im maschinenbau, *in* ‘Encyklopädie der Mathematischen Wissenschaften’, Vol. IV, 4, chapter 27, p. 349. Vol. IV, 4.
- Voyiadjis, G. Z. & Baluch, M. H. (1981), ‘Refined theory for flexural motions of isotropic elastic plates’, *Journal of sound & Vibration* **76**, 57–64.
- Wang, C. M., Lim, G. T., Reddy, J. N. & Lee, K. H. (2001), ‘Relationships between bending solutions of Reissner and Mindlin plate theories’, *Engineering Structures* **23**, 838–849.
- Wang, C. T. (1948), Bending of rectangular plates with large deflections, Report no. 1462, NACA.
- Wang, J. & Schweizerhof, K. (1995), ‘The fundamental solutions of moderately thick laminated anisotropic shallow shells’, *Int. J. Engng. Sci.* **33**, 994–1004.
- Wang, W., Ji, X. & Wang, Y. (1999), ‘Object-oriented programming in boundary element methods using c++’, *Advances in Engineering Software* **30**, 127–132.
- Wang, X., Qian, J. & Huang, M. (1991), ‘Boundary integral equation formulation for large amplitude nonlinear vibration of thin elastic plates’, *Comput. Methods Appl. Mech. Engrg.* **86**, 73–86.
- Washizu, K. (1982), *Variational Methods in Elasticity and Plasticity*, Pergamon.
- Westphal Jr., T. (1990), Abordagem direta dos modelos de placa de reissner e mindlin pelo método dos elementos de contorno, Master’s thesis, Universidade Federal de Santa Catarina, Florianópolis.
- Westphal Jr., T., Andrä, H. & Schnack, E. (1999), Green’s functions and fundamental solution for the kirchhoff’s plate bending model, Technical report, Karlsruhe University.

- Westphal Jr., T. & de Barcellos, C. S. (1990), Applications of the boundary element method to reissner's and mindlin's plate models, *in* M. Tanaka, C. A. Brebbia & T. Honma, eds, 'Proc. 12th Int. Conf. BEM', Vol. 1, Sapporo, Japão, pp. 467–477.
- Westphal Jr., T., Schnack, E. & de Barcellos, C. S. (1998), 'The general fundamental solution of the sixth-order reissner and mindlin plate bending models revisited', *Comput. Methods Appl. Mech. Engrg.* **166**, 363–378.
- Wilson, P. E. & Boresi, A. P. (1964), 'Large deflection of a clamped circular plate including effects of transverse shear', *Journal of Applied Mechanics* pp. 540–541.
- Wittrick, W. H. (1987), 'Analytical three-dimensional elasticity solutions to some plate problems, and some observations on mindlin's plate theory', *Int. J. Solids & Structures* **23**, 441–464.
- Xiao-Yan, L. (1995), 'A new BEM approach for reissner's plate bending', *Computers & Structures* **54**, 1085–1090.
- Xiao-Yan, L., Mao-Kwang, H. & Xiuxi, W. (1990), 'Geometrically nonlinear analysis of a reissner type plate by the boundary element method', *Computers & Structures* **37**, 911–916.
- Ye, J. (1994), 'Large deflection of imperfect plates by iterative BE-FE method', *ASCE Journal of Engineering Mechanics* **120**, 431–444.
- Ye, T. Q. & Liu, Y. (1985), 'Deflection analysis of elastic plate by the boundary element method', *Appl. Math. Modelling* **9**, 183–188.
- Yuan, F. G. & Miller, R. E. (1988), 'A rectangular finite element for moderately thick flat plates', *Computers & Structures* **30**, 1375–1387.
- Yuying, H., Weifang, Z. & Qinghua, Q. (1992), 'Postbuckling analysis of plates on an elastic foundation by the boundary element method', *Comput. Methods Appl. Mech. Engrg.* **100**, 315–323.
- Zeglinski, G. W., Han, R. P. S. & Aitchison, P. (1994), 'Object-oriented matrix classes for use in a finite element code using C++', *Int. J. Num. Meth. Engng.* **37**, 3921–3937.

- Zhang, J. D. & Atluri, S. N. (1986), ‘A Boundary/Interior element method for quasi-static and transient response analysis of shallow shells’, *Computers & Structures* **24**, 213–223.
- Zhang, J. D. & Atluri, S. N. (1987), Nonlinear static and dynamic analysis of shells by a field-boundary element method, *in* ‘Proc. 28th Str. Dynam. Conf.’, Monterey, pp. 488–495.
- Zimmermann, T., Dubois-Pèlerin, Y. & Bomme, P. (1992), ‘Object-oriented finite element programming: I. governing principles’, *Comput. Methods Appl. Mech. Engrg.* **98**, 291–303.
- Zimmermann, T. & Eyheramendy, D. (1996), ‘Object-oriented finite elements i. principles of symbolic derivations and automatic programming’, *Comput. Methods Appl. Mech. Engrg.* **132**, 259–276.
- Zimmermann, T., Eyheramendy, D. & Bomme, P. (1998), Governing principles of object-oriented finite element programming - application of automatic coding to a one-dimensional nonlinear advection problem, *in* S. Idelsohn, E. O. nate & E. Dvorkin, eds, ‘Computational Mechanics - New Trends and Applications’, CIMNE, Barcelona, Spain.

Apêndice A

Soluções Fundamentais

Como foi descrito no Capítulo 3, as equações integrais utilizadas no presente trabalho fazem uso de soluções fundamentais, cujas expressões analíticas devem ser conhecidas. Estes tensores descrevem basicamente deslocamentos e trações generalizadas que ocorrem em um problema fundamental, que em geral corresponde ao estado auxiliar utilizado nas relações de reciprocidade. Cabe salientar que podem existir mais de uma solução fundamental para um operador diferencial, mas comumente utiliza-se aquela correspondente à resposta de um domínio infinito submetido a excitações unitárias concentradas generalizadas, paralelas às direções coordenadas. Não será aqui desenvolvido o processo de obtenção da solução fundamental. Serão apenas destacados alguns aspectos e apresentadas as expressões analíticas dos tensores fundamentais implementados no código computacional utilizado neste trabalho.

A solução fundamental do problema de elasticidade bidimensional na forma governada pelo operador (2.28a) é bem conhecida da literatura (Brebbia et al. 1984). A determinação da solução fundamental correspondente ao operador diferencial das equações de placa de Mindlin e de Reissner, o operador ${}^f\mathcal{L}$ da equação (2.28b), é detalhadamente apresentada nos trabalhos de Monken e Silva (1988) e Westphal Jr. (1990). Nestas referências, foi utilizado o método de Hörmander (Hörmander 1964), que reduz o problema de encontrar uma solução fundamental para um operador \mathbf{L} no de encontrar uma solução escalar para o operador $\|\mathbf{L}\|$ (Monken e Silva 1988). Assim, é obtido o tensor deslocamento fundamental \mathbf{U} , composto por uma combinação linear de seis funções, apropriadamente multiplicadas por seis constantes. Algumas destas constantes são obtidas através da aplicação da teoria

das distribuições e de condições de regularidade (Brebba et al. 1984). Duas destas constantes não possuem quaisquer condições impostas, caracterizando a solução fundamental do problema como composta por funções essenciais e funções livres. Então é possível estipular para estas constantes valores tais que simplifiquem as expressões dos tensores. Uma destas formas simplificadas da solução fundamental foi adotada no presente trabalho. O tensor fundamental \mathbf{T} é obtido por relações tração-deslocamento, envolvendo uma combinação de derivadas do tensor \mathbf{U} . Note-se que embora existam algumas variações das expressões aqui empregadas, as formas operacionais dos tensores \mathbf{U} ou \mathbf{T} são as mesmas para ambos os modelos de placas analisados.

A seguir são apresentadas as expressões de todos os tensores das soluções fundamentais utilizados neste trabalho. Nestas expressões, vale a seguinte notação:

$$z = \lambda r \quad (\text{A.1})$$

$$r_\alpha = x_\alpha(Q) - x_\alpha(P) \quad (\text{A.2})$$

$$r = \|Q - P\| = \sqrt{r_\alpha r_\alpha} \quad (\text{A.3})$$

$$r_{,\alpha} = \frac{\partial r}{\partial x_\alpha(Q)} = \frac{1}{r} r_\alpha \quad (\text{A.4})$$

$$A(z) = K_0(z) + \frac{2}{z} \left(K_1(z) - \frac{1}{z} \right) \quad (\text{A.5})$$

$$B(z) = K_0(z) + \frac{1}{z} \left(K_1(z) - \frac{1}{z} \right) \quad , \quad (\text{A.6})$$

onde K_0 e K_1 são funções modificadas de Bessel de segunda espécie, de ordem zero e um, respectivamente (Abramowitz & Stegun 1972). Os termos F_3 e F_6 são as funções livres citadas (de Barcellos & Westphal Jr. 1992). Adicionalmente, os tensores de elasticidade bidimensional são válidos para estado plano de deformações. Para modificá-los para estado plano de tensões deve-se substituir ν por $\nu/(1 - \nu)$.

A.1 Tensores fundamentais de deslocamentos

Elasticidade bidimensional:

$${}^m U_{\alpha\beta} = \frac{-1}{8\pi G(1 - \nu)} \left[(3 - 4\nu) \ln r \delta_{\alpha\beta} - r_{,\alpha} r_{,\beta} \right] \quad (\text{A.7})$$

Flexão de placas:

$${}^f U_{\alpha\beta} = \frac{1}{8\pi D(1-\nu)} \left\{ [8B - (1-\nu)(2\ln z + 1 + 8F_3)] \delta_{\alpha\beta} + \right. \\ \left. - [8A + 2(1-\nu)] r_{,\alpha} r_{,\beta} \right\} \quad (\text{A.8a})$$

$${}^f U_{\alpha 3} = \frac{1}{8\pi D} (2\ln z + 1 + 8F_3) r r_{,\alpha} \quad (\text{A.8b})$$

$${}^f U_{3\alpha} = -U_{\alpha 3} \quad (\text{A.8c})$$

$${}^f U_{33} = \frac{1}{8\pi D(1-\nu)\lambda^2} \left\{ z^2 (1-\nu) (\ln z + 4F_3) - 8\ln z + \right. \\ \left. - 4[(3-\nu)(4F_3 + 1) - (1-\nu)F_6] \right\} \quad (\text{A.8d})$$

A.2 Tensores fundamentais de tração

Elasticidade bidimensional:

$${}^m T_{\alpha\beta} = \frac{-1}{4\pi(1-\nu)r} \left\{ [(1-2\nu)\delta_{\alpha\beta} + 2r_{,\alpha} r_{,\beta}] r_{,n} - (1-2\nu)(n_{\alpha} r_{,\beta} - n_{\beta} r_{,\alpha}) \right\} \quad (\text{A.9})$$

Flexão de placas:

$${}^f T_{\alpha\beta} = \frac{-1}{4\pi r} \left[(4A + 2zK_1 + 1 - \nu)(r_{,\beta} n_{\alpha} + r_{,n} \delta_{\alpha\beta}) + \right. \\ \left. + (4A + 1 + \nu)r_{,\alpha} n_{\beta} - 2(8A + 2zK_1 + 1 - \nu)r_{,\alpha} r_{,\beta} r_{,n} \right] \quad (\text{A.10a})$$

$${}^f T_{\alpha 3} = \frac{\lambda^2}{2\pi} (B n_{\alpha} - A r_{,\alpha} r_{,n}) \quad (\text{A.10b})$$

$${}^f T_{3\alpha} = \frac{-1}{8\pi} \left\{ [2(1+\nu)\ln z + (1+8F_3) + (3+8F_3)\nu] n_{\alpha} + 2(1-\nu)r_{,\alpha} r_{,n} \right\} \quad (\text{A.10c})$$

$${}^f T_{33} = \frac{-1}{2\pi r} r_{,n} \quad (\text{A.10d})$$

A.3 Tensores derivados

As expressões a seguir apresentam todas as derivadas dos tensores \mathbf{U} e \mathbf{T} empregadas na formulação apresentada no Capítulo 3. Entretanto, os tensores $\overline{\mathbf{U}}$, $\overline{\mathbf{T}}$, $\overline{\mathbf{V}}$, $\tilde{\mathbf{U}}$ e $\overline{\overline{\mathbf{U}}}$ são tensores de ordens diversas e então apenas algumas das componentes abaixo são utilizadas. O contexto em que são empregados esclarece quais componentes devem ser consideradas em cada caso. Todas as derivadas estão calculadas em relação ao ponto campo Q .

A.3.1 Derivadas primeiras de U

Elasticidade bidimensional:

$${}^m U_{\alpha\beta,\gamma} = \frac{-1}{8\pi G(1-\nu)r} [(3-4\nu)r_{,\gamma}\delta_{\alpha\beta} - r_{,\alpha}\delta_{\beta\gamma} - r_{,\beta}\delta_{\alpha\gamma} + 2r_{,\alpha}r_{,\beta}r_{,\gamma}] \quad (\text{A.11})$$

Flexão de placas:

$${}^f U_{\alpha 3,\beta} = \frac{1}{8\pi D} [(2\ln z + 1 + 8F_3)\delta_{\alpha\beta} + 2r_{,\alpha}r_{,\beta}] \quad (\text{A.12a})$$

$${}^f U_{3\alpha,\beta} = -{}^f U_{\alpha 3,\beta} \quad (\text{A.12b})$$

$${}^f U_{33,\beta} = \frac{1}{8\pi D(1-\nu)z^2} [(2\ln z + 1 + 8F_3)(1-\nu)z^2 - 8] r r_{,\beta} \quad (\text{A.12c})$$

A.3.2 Derivadas primeiras de T

Elasticidade bidimensional:

$$\begin{aligned} {}^m T_{\alpha\beta,\gamma} = \frac{-1}{4\pi(1-\nu)r^2} \left\{ [(1-2\nu)r_{,\gamma}\delta_{\alpha\beta} - r_{,\alpha}\delta_{\beta\gamma} - r_{,\beta}\delta_{\alpha\gamma} + 4r_{,\alpha}r_{,\beta}r_{,\gamma}] 2r_{,n} + \right. \\ \left. + (1-2\nu)[2n_{\alpha}r_{,\beta}r_{,\gamma} - 2n_{\beta}r_{,\alpha}r_{,\gamma} - n_{\alpha}\delta_{\beta\gamma} + n_{\beta}\delta_{\alpha\gamma} - n_{\gamma}\delta_{\alpha\beta}] + \right. \\ \left. - 2n_{\gamma}r_{,\alpha}r_{,\beta} + \right\} \quad (\text{A.13}) \end{aligned}$$

Flexão de placas:

$${}^f T_{3\alpha,\beta} = \frac{-(1-\nu)}{4\pi r} \left[r_{,\alpha}n_{\beta} + r_{,n}\delta_{\alpha\beta} + \frac{(1+\nu)}{(1-\nu)}r_{,\beta}n_{\alpha} - 2r_{,\alpha}r_{,\beta}r_{,n} \right] \quad (\text{A.14a})$$

$${}^f T_{33,\beta} = \frac{-1}{2\pi r^2} (n_{\alpha} - 2r_{,\alpha}r_{,n}) \quad (\text{A.14b})$$

A.3.3 Derivadas segundas de U

Flexão de placas:

$${}^f U_{33,\alpha\beta} = \frac{1}{2\pi D} \left\{ \frac{-2(\delta_{\alpha\beta} - 2r_{,\alpha}r_{,\beta})}{(1-\nu)z^2} + 2F_3\delta_{\alpha\beta} + \frac{1}{4} [(2\ln z + 1)\delta_{\alpha\beta} + 2r_{,\alpha}r_{,\beta}] \right\} \quad (\text{A.15})$$

A.4 Tensores envolvidos nos termos de carregamento

Os tensores utilizados nos termos de carregamento são o tensor \mathbf{U} e eventualmente combinações de suas derivadas, conforme indicado pelas eqs.(3.43) e (3.60). No cálculo das eqs.(3.76) e (3.77) também são necessárias as derivadas de \mathbf{V} , e estas são obtidas diretamente das eqs.(A.11), (A.12) e (A.15), esta última sendo necessária apenas para o modelo de placa de Reissner.

Para placas sob cargas transversais uniformemente distribuídos, as integrais dos termos de carregamento podem ser convertidas para o contorno, de acordo com o que foi descrito no Capítulo 3. Neste caso, é feito uso do tensor \mathbf{V} que utiliza componentes de \mathbf{U} conforme a eq.(3.61), e também do tensor \mathbf{Y} , dado a seguir.

$$\begin{aligned}
 {}^f Y_{\alpha\beta} &= \frac{r^2}{32\pi D} \left\{ \left(4F_3 + \frac{1}{4} \right) (2r_{,\alpha} r_{,\beta} + \delta_{\alpha\beta}) + [r_{,\alpha} r_{,\beta} (2 \ln z + 1) + \ln z \delta_{\alpha\beta}] \right\} \quad \text{A.16} \\
 {}^f Y_{3\beta} &= \frac{r r_{,\alpha}}{8\pi D \lambda^2} \left[\frac{-2}{(1-\nu)} (2 \ln z - 1) + F_3 z^2 + \frac{z^2}{16} (4 \ln z - 1) + \right. \\
 &\quad \left. - \frac{2(3-\nu)}{(1-\nu)} (4F_3 + 1) + 2F_6 \right] \quad \text{A.17}
 \end{aligned}$$

Uma expressão similar para ${}^m \mathbf{Y}$ pode ser encontrada na literatura (Brebbia et al. 1984).

Apêndice B

Diferenciação de Integrais com Núcleos Singulares

No Capítulo 3 a obtenção das equações integrais para as derivadas do deslocamento transversal são obtidas mediante diferenciação das identidades de Somigliana referentes aos deslocamentos translacionais em relação às coordenadas x_α . Especificamente, é necessário diferenciar as equações (3.18) e (3.34), originando as relações (3.76) e (3.77). Sob a condição de que as derivadas são tomadas com referência ao ponto campo $Q \in \Omega$, é possível diferenciar quase todos os termos de (3.18) e (3.34), porque estes serão no máximo quase singulares. Entretanto, em pontos internos algumas integrais de domínio são singulares quando $P \equiv Q$. Este fato impede que esta diferenciação seja realizada da maneira usual. Quando se diferencia uma integral singular, deve-se fazer uso do conceito originalmente proposto por Mikhlin (Mikhlin 1962, Bui 1978), através da aplicação da fórmula de Leibnitz (Sokolnikoff & Redheffer 1958), pois os limites de integração dependem da variável segundo a qual se está integrando.

Este apêndice mostra o desenvolvimento formal para obtenção das equações (3.76) e (3.77). Seguindo o procedimento descrito em Brebbia et al. (1984), obtém-se as equações integrais para as derivadas dos deslocamentos translacionais em relação a dois eixos coordenados. O desenvolvimento aqui apresentado leva ao aparecimento de termos adicionais, chamados termos convectivos ou termos livres (Bui 1978), dados pelas eqs.(3.73).

Uma inspeção sobre as equações (3.48) e (3.65), particularizadas para pontos internos, revela que todas as integrais sobre Ω são candidatas à originar termos convectivos. Estes

termos são:

$$I_i^N = \int_{\Omega} {}^f U_{i3,\beta}(Q, P) N_{\alpha\beta}(Q) u_{3,\alpha}(Q) d\Omega_Q \quad (\text{B.1a})$$

$$I_i^q = \int_{\Omega} {}^f V_{ij}(Q, P) q_j(Q) d\Omega_Q \quad (\text{B.1b})$$

$$J_{\alpha}^N = \int_{\Omega} {}^m U_{\alpha\beta,\delta}(Q, P) N_{\beta\delta}^n(Q) d\Omega_Q \quad (\text{B.1c})$$

$$J_{\alpha}^q = \int_{\Omega} {}^m V_{\alpha\beta}(Q, P) q_{\beta}(Q) d\Omega_Q \quad (\text{B.1d})$$

A rigor, o mesmo procedimento aqui desenvolvido deveria ser aplicado para as integrais definidas sobre Γ , mas o cálculo das derivadas dos deslocamentos faz-se necessário apenas em pontos internos, quando da utilização de células descontínuas. Outros termos convectivos (não abordados aqui) surgem no cálculo destas derivadas em pontos sobre o contorno da placa, o que ocorreria se fossem utilizados células contínuas (Guiggiani 1995). Entretanto, fatalmente seriam necessários procedimentos de integração hipersingulares. Procede-se agora o cálculo das derivadas correspondentes de (B.1):

$$\frac{\partial I_i^N}{\partial x_{\gamma}(P)} \quad (\text{B.2})$$

$$\frac{\partial I_i^q}{\partial x_{\gamma}(P)} \quad (\text{B.3})$$

$$\frac{\partial J_{\alpha}^N}{\partial x_{\gamma}(P)} \quad (\text{B.4})$$

$$\frac{\partial J_{\alpha}^q}{\partial x_{\gamma}(P)} \quad (\text{B.5})$$

B.1 Cálculo de $\frac{\partial I_i^N}{\partial x_{\gamma}(P)}$

Reescreva-se a eq.(B.1a) na forma (Brebbia et al. 1984):

$$I_i^N = \lim_{\epsilon \rightarrow 0} \int_{\Omega - \Omega_{\epsilon}} U_{i3,\alpha}(Q, P) M_{\alpha}(Q) d\Omega_Q \quad , \quad (\text{B.6})$$

onde

$$M_{\alpha}(Q) = N_{\alpha\beta}(Q) u_{3,\beta}(Q) \quad ,$$

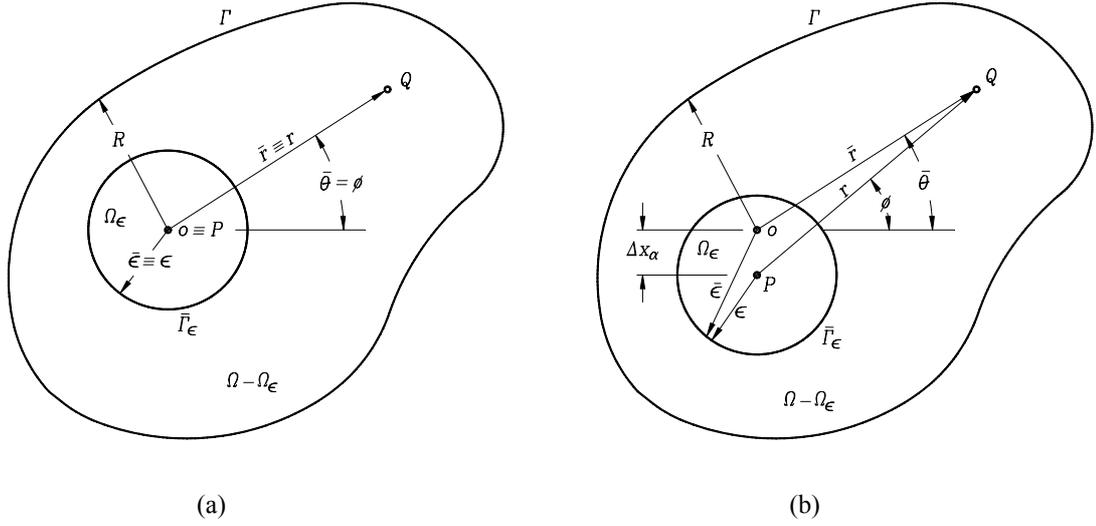


Figura B-1: Definição do contorno $\bar{\Gamma}_\epsilon$ em torno do ponto-fonte. (a) Configuração inicial. (b) Efeito de um incremento Δx_α aplicado às coordenadas do ponto-fonte.

que corresponde ao acoplamento flexão-membrana. Na eq.(B.6), Ω_ϵ é um círculo de raio ϵ centrado em P , com contorno $\bar{\Gamma}_\epsilon$. Para diferenciar eq.(B.6) em relação às coordenadas x_α deve-se então proceder:

$$\frac{\partial I_i^N}{\partial x_\gamma(P)} = \lim_{\epsilon \rightarrow 0} \left(\frac{\partial}{\partial x_\gamma} \int_{\Omega - \Omega_\epsilon} U_{i3,\alpha}(Q, P) M_\alpha(Q) d\Omega_Q \right) . \quad (\text{B.7})$$

Definindo-se um sistema de coordenadas polar $(\bar{r}, \bar{\theta})$ centrado na origem $P \equiv o$, como ilustrado na figura B-1, reescreve-se $U_{i3,\alpha}$ na forma (apenas as parcelas singulares do tensor $U_{i3,\alpha}$ devem ser consideradas) :

$$U_{i3,\alpha} = \frac{1}{r(\bar{r}, \bar{\theta})} \Lambda_{i3,\alpha}(\phi) . \quad (\text{B.8})$$

A figura B-1a mostra que $r(\bar{r}, \bar{\theta}) = \bar{r}$ e $\phi(\bar{r}, \bar{\theta}) = \bar{\theta}$. Entretanto, se o ponto carga P se move um incremento Δx_α , em coordenadas cartesianas, r e ϕ tornam-se diferentes de \bar{r} e $\bar{\theta}$ e o contorno $\bar{\Gamma}_\epsilon$ também é alterado. Isto indica a dependência de $\bar{\Gamma}_\epsilon$ da posição do

ponto carga. No sistema de coordenadas polar, a eq.(B.7) é reescrita:

$$\frac{\partial I_i^N}{\partial x_\gamma} = \int_0^{2\pi} \lim_{\epsilon \rightarrow 0} \left(\frac{\partial}{\partial x_\gamma} \int_{\bar{\epsilon}}^{R(\bar{\theta})} \frac{\Lambda_{i3,\alpha}(\phi)}{r} M_\alpha(Q) \bar{r} d\bar{r} \right) d\bar{\theta} \quad . \quad (\text{B.9})$$

Note-se que na eq.(B.9) os limites de integração dependem das variáveis segundo a qual se está integrando. Quando isto acontece, deve-se recorrer à fórmula de Leibnitz (Sokolnikoff & Redheffer 1958):

$$\frac{d}{d\alpha} \int_{\phi_1(\alpha)}^{\phi_2(\alpha)} f(x, \alpha) dx = \int_{\phi_1(\alpha)}^{\phi_2(\alpha)} \frac{\partial f(x, \alpha)}{\partial \alpha} dx - f(\phi_1, \alpha) \frac{d\phi_1}{d\alpha} + f(\phi_2, \alpha) \frac{d\phi_2}{d\alpha} \quad . \quad (\text{B.10})$$

Na forma como (B.9) está escrita, é possível aplicar diretamente (B.10) ao termo entre parênteses, o que leva a:

$$\begin{aligned} \frac{\partial}{\partial x_\gamma} \int_{\bar{\epsilon}}^{R(\bar{\theta})} \frac{\Lambda_{i3,\alpha}(\phi)}{r} M_\alpha(Q) \bar{r} d\bar{r} &= \int_{\bar{\epsilon}}^{R(\bar{\theta})} \frac{\partial}{\partial x_\gamma} \left(\frac{\Lambda_{i3,\alpha}(\phi)}{r} \right) M_\alpha(Q) \bar{r} d\bar{r} + \\ &- \frac{\Lambda_{i3,\alpha}(\phi)}{r(\bar{\epsilon}, \bar{\theta})} M_\alpha(P) \bar{\epsilon} \frac{d\bar{\epsilon}}{dx_\gamma} + \frac{\Lambda_{i3,\alpha}(\phi)}{r(R, \bar{\theta})} M_\alpha(P) R \frac{dR}{dx_\gamma} \quad . \end{aligned} \quad (\text{B.11})$$

Como a origem do sistema de coordenadas está sobre o ponto fonte P antes da aplicação do incremento Δx_α , e ali permanece após este incremento, apenas $\bar{\epsilon}$ muda com x_α , enquanto R não. Por isto o último termo à direita de (B.11) se anula. Assim, (B.9) é reescrita:

$$\begin{aligned} \frac{\partial I_i^N}{\partial x_\gamma} &= \int_0^{2\pi} \lim_{\epsilon \rightarrow 0} \left[\int_{\epsilon}^{R(\phi)} \frac{\partial}{\partial x_\gamma} \left(\frac{\Lambda_{i3,\alpha}(\phi)}{r} \right) M_\alpha(Q) r dr \right] d\phi + \\ &- M_\alpha(P) \int_0^{2\pi} \Lambda_{i3,\alpha}(\phi) \cos(r, x_\gamma) d\phi \quad , \end{aligned} \quad (\text{B.12})$$

onde foi levado em consideração que se $P \equiv o$, $r(\bar{\epsilon}, \bar{\theta}) = \epsilon = \bar{\epsilon}$.

Resta agora verificar a existência da primeira integral do lado direito de (B.12). Para isto, reescreve-se:

$$\begin{aligned} &\int_0^{2\pi} \lim_{\epsilon \rightarrow 0} \left[\int_{\epsilon}^R \frac{\partial}{\partial x_\gamma} \left(\frac{\Lambda_{i3,\alpha}(\phi)}{r} \right) M_\alpha(Q) r dr \right] d\phi = \\ &= \int_0^{2\pi} \lim_{\epsilon \rightarrow 0} \left[\int_{\epsilon}^R r^2 \frac{\partial}{\partial x_\gamma} \left(\frac{\Lambda_{i3,\alpha}(\phi)}{r} \right) M_\alpha(Q) \frac{1}{r} dr \right] d\phi \quad . \end{aligned} \quad (\text{B.13})$$

Chamando:

$$\bar{\Lambda}_{i3,\alpha\gamma}(\phi) = r^2 \frac{\partial}{\partial x_\gamma} \left(\frac{\Lambda_{i3,\alpha}(\phi)}{r} \right) \quad , \quad (\text{B.14})$$

e somando e subtraindo o termo

$$\int_0^{2\pi} \lim_{\epsilon \rightarrow 0} \left[\int_\epsilon^R \frac{\partial}{\partial x_\gamma} \left(\frac{\Lambda_{i3,\alpha}(\phi)}{r} \right) M_\alpha(P) r dr \right] d\phi \quad ,$$

na eq.(B.13), vem:

$$\begin{aligned} & \int_0^{2\pi} \lim_{\epsilon \rightarrow 0} \left[\int_\epsilon^R \frac{\partial}{\partial x_\gamma} \left(\frac{\Lambda_{i3,\alpha}(\phi)}{r} \right) M_\alpha(Q) r dr \right] d\phi = \\ & = \int_0^{2\pi} \lim_{\epsilon \rightarrow 0} \left\{ \bar{\Lambda}_{i3,\alpha\gamma}(\phi) \int_\epsilon^R [M_\alpha(Q) - M_\alpha(P)] \frac{1}{r} dr \right\} d\phi + \\ & + M_\alpha(P) \int_0^{2\pi} \bar{\Lambda}_{i3,\alpha\gamma}(\phi) \ln(R) d\phi - \lim_{\epsilon \rightarrow 0} \left[M_\alpha(P) \ln \epsilon \int_0^{2\pi} \bar{\Lambda}_{i3,\alpha\gamma}(\phi) d\phi \right] \end{aligned} \quad (\text{B.15})$$

As integrais da eq.(B.15) são todas limitadas, desde que o acoplamento flexão membrana satisfaça a condição de Hölder em P , isto é (Brebbia et al. 1984):

$$\|M_\alpha(Q) - M_\alpha(P)\| \leq Ar^\alpha \quad , \quad (\text{B.16})$$

onde A e α são constantes positivas. O tensor $\bar{\Lambda}_{i3,\alpha m}$ satisfaz a propriedade

$$\int_0^{2\pi} \bar{\Lambda}_{i3,\alpha\gamma}(\phi) d\phi = 0 \quad , \quad (\text{B.17})$$

e por isso o último termo à direita de (B.15) é nulo. As duas primeiras integrais do lado direito de (B.15) são convergentes pois:

$$\lim_{\epsilon \rightarrow 0} \left[\bar{\Lambda}_{i3,\alpha m}(\phi) \int_\epsilon^R \frac{Ar^\alpha}{r} dr \right] = \lim_{\epsilon \rightarrow 0} \left[\frac{Ar^{\alpha+2}}{\alpha-1} \ln(r) \frac{\partial}{\partial x_\gamma} \left(\frac{\Lambda_{i3,\alpha}}{r} \right) \right]_\epsilon^R < \infty \quad ,$$

e a segunda se anula devido à propriedade (B.17), o que completa a demonstração.

Agora $\partial I_i^N / \partial x_\gamma$ pode ser reescrita novamente em coordenadas retangulares:

$$\frac{\partial I_i^N}{\partial x_\gamma} = - \int_{\Omega} \frac{\partial U_{i3,\alpha}(Q,P)}{\partial x_\gamma} N_{\alpha\beta}(Q) u_{3,\beta}(Q) d\Omega_Q - N_{\alpha\beta}(P) u_{3,\beta}(P) \int_{\Gamma'_1} U_{i3,\alpha} r_{,\gamma} d\Gamma' \quad (\text{B.18})$$

onde a primeira integral deve ser interpretada no sentido do valor principal de Cauchy. Note-se que o sinal negativo foi adicionado à primeira integral de (B.18), pois a derivada está sendo tomada em relação ao ponto carga. Γ'_1 é definido por uma circunferência de raio unitário centrada em P e $r_{,\gamma}$ é a derivada de r em relação às coordenadas do ponto campo. A segunda integral do lado direito de (B.18) é a chamada parcela convectiva, porque aparece de uma alteração da posição do ponto P .

No presente trabalho, é necessário desenvolver o segundo termo à direita de (B.18) para $i = 3$, pois não serão utilizadas apenas as derivadas cartesianas das rotações.

Como a normal exterior de Γ'_1 aponta para o centro do círculo, tem-se:

$$r_{,\alpha} = -n_\alpha \quad .$$

Assim, pode-se escrever o termo convectivo como:

$${}^f \mathbf{c}^N(P) = N_{\alpha\beta}(P) \int_{\Gamma'_1} U_{33,\alpha}^s r_{,\gamma} d\Gamma' = -N_{\alpha\beta}(P) \int_{\Gamma'_1} U_{33,\alpha}^s n_\gamma d\Gamma' \quad , \quad (\text{B.19})$$

onde $U_{33,\alpha}^s$ contém apenas a parcela singular do tensor $U_{33,\alpha}$ (ver equação (A.8d)), tendo em vista que as parcelas regulares e fracamente singulares não originam qualquer contribuição convectiva.

Como

$$U_{33,\alpha}^s = \frac{-1}{\pi D(1-\nu)z^2} r r_{,\alpha} = \frac{-1}{\pi D(1-\nu)\lambda^2} \frac{r_{,\alpha}}{r} \quad ,$$

(o que valida a representação (B.8)) e $d\Gamma = r d\phi$, então a equação (B.19) é analiticamente definida por:

$${}^f \mathbf{c}^N(P) = \frac{-1}{\pi D(1-\nu)\lambda^2} \left[\int_{2\pi}^0 n_\gamma n_\alpha d\phi \right] N_{\gamma\beta}(P) \quad .$$

Lembrando, da figura B-1, que $n_1 = -\cos \phi$, $n_2 = -\sin \phi$ e usando as relações:

$$\int_{2\pi}^0 \cos^2 \phi \, d\phi = \int_{2\pi}^0 \sin^2 \phi \, d\phi$$

$$\int_{2\pi}^0 \cos \phi \sin \phi \, d\phi = 0 \quad ,$$

obtém-se, levando em conta o sinal do último termo à direita de de (B.18):

$$\boxed{{}^f c_{\alpha\beta}^N(P) = \frac{-N_{\alpha\beta}(P)}{D(1-\nu)\lambda^2}} \quad . \quad (\text{B.20})$$

Assim, a matriz de acoplamento flexão membrana é obtida através da adição de (B.20), como termo não integral, às matrizes $\bar{\mathbf{B}}_r$ conforme indicado pelas eqs.(4.55). Esta operação deve ser realizada apenas no caso singular $P \equiv Q$. A expressão (B.20) está de acordo com o termo convectivo obtido por Xiao-Yan et al. (1990).

B.2 Cálculo de $\frac{\partial I_i^q}{\partial x_\gamma}(P)$

Relembre-se da eq.(3.60) que

$${}^f \mathbf{V}(Q, P) = {}^f \mathbf{U}(Q, P) - m_f \tilde{\mathbf{U}}(Q, P) \quad , \quad (\text{B.21})$$

onde $\tilde{\mathbf{U}}$ é matricialmente dado por

$$\tilde{\mathbf{U}} = \begin{bmatrix} 0 & 0 & U_{11,1} + U_{12,2} \\ 0 & 0 & U_{21,1} + U_{22,2} \\ 0 & 0 & U_{31,1} + U_{32,2} \end{bmatrix}$$

porque este é conjugado com o vetor carregamento ${}^f \mathbf{q} = \left\{ m_1 \quad m_2 \quad q_3 \right\}^T$. Então, seguindo o procedimento da seção anterior, quando se reescreve a eq.(B.1b) :

$$I_i^q = \lim_{\epsilon \rightarrow 0} \int_{\Omega - \Omega_\epsilon} V_{ij}(Q, P) q_j(Q) \, d\Omega_Q \quad , \quad (\text{B.22})$$

as derivadas resultam na forma:

$$\frac{\partial I_i^N}{\partial x_\gamma(P)} = \lim_{\epsilon \rightarrow 0} \left(\frac{\partial}{\partial x_\gamma} \int_{\Omega - \Omega_\epsilon} U_{ij}(Q, P) q_j(Q) d\Omega_Q + m_f \frac{\partial}{\partial x_\gamma} \int_{\Omega - \Omega_\epsilon} \tilde{U}_{ij}(Q, P) q_j(Q) d\Omega_Q \right) . \quad (\text{B.23})$$

Portanto, o tratamento deve ser dado apenas para o modelo de placa de Reissner ($m_f = 1$). Caso contrário, ${}^f \mathbf{V} = {}^f \mathbf{U}$ e como $\mathbf{U} = O(\ln r)$, a primeira integral sobre Ω não manifesta singularidade e não origina termos convectivos. Já a segunda integral sobre Ω necessita de uma inspeção mais cuidadosa. Como o interesse está apenas sobre a derivada do deslocamento transversal, particulariza-se (B.23) considerando apenas os termos de interesse:

$$\frac{\partial I_3^N}{\partial x_\gamma(P)} = \lim_{\epsilon \rightarrow 0} \left(\frac{\partial}{\partial x_\gamma} \int_{\Omega - \Omega_\epsilon} U_{3\alpha,\alpha}(Q, P) q_3(Q) d\Omega_Q \right) . \quad (\text{B.24})$$

Mas como $U_{3\alpha,\alpha}$ são todos regulares sobre Ω (ver eqs.(A.12)), não se aplica a representação

$$U_{3\alpha,\alpha} = \frac{1}{r(\bar{r}, \bar{\theta})} \Lambda_{3\alpha,\alpha}(\phi) , \quad (\text{B.25})$$

e portanto não há qualquer contribuição convectiva. Logo:

$$\boxed{{}^f c_{\alpha\beta}^a(P) = 0} . \quad (\text{B.26})$$

B.3 Cálculo de $\frac{\partial J_\alpha^N}{\partial x_\gamma(P)}$

Neste caso, a eq.(B.1c) é reescrita:

$$J_\alpha^N = \lim_{\epsilon \rightarrow 0} \int_{\Omega - \Omega_\epsilon} U_{\alpha\beta,\delta}(Q, P) N_{\beta\delta}^n(Q) d\Omega_Q . \quad (\text{B.27})$$

Repetindo o procedimento das seções anteriores, tem-se:

$$\frac{\partial J_\alpha^N}{\partial x_\gamma(P)} = \lim_{\epsilon \rightarrow 0} \left(\frac{\partial}{\partial x_\gamma} \int_{\Omega - \Omega_\epsilon} U_{\alpha\beta,\delta}(Q, P) N_{\beta\delta}^n(Q) d\Omega_Q \right) . \quad (\text{B.28})$$

Chamando:

$$U_{\alpha\beta,\delta} = \frac{1}{r(\bar{r}, \bar{\theta})} \Lambda_{\alpha\beta,\delta}(\phi) \quad , \quad (\text{B.29})$$

resulta, após a aplicação da fórmula de Leibnitz:

$$\frac{\partial J_\alpha^N}{\partial x_\gamma} = - \int_\Omega \frac{U_{\alpha\beta,\delta}}{\partial x_\gamma} N_{\beta\delta}^n(Q) d\Omega_Q - N_{\beta\delta}^n(P) \int_{\Gamma'_1} U_{\alpha\beta,\delta} r_{,\gamma} d\Gamma' \quad (\text{B.30})$$

onde a primeira integral deve ser interpretada no sentido do valor principal de Cauchy.

Assim, pode-se escrever o termo convectivo como:

$${}^m \mathbf{c}^N(P) = N_{\beta\delta}^n(P) \int_{\Gamma'_1} U_{\alpha\beta,\delta} r_{,\gamma} d\Gamma' = -N_{\beta\delta}^n(P) \int_{\Gamma'_1} U_{\alpha\beta,\delta} n_\gamma d\Gamma' \quad , \quad (\text{B.31})$$

onde $U_{\alpha\beta,\delta}$ é $O(r^{-1})$, como mostra a eq.(A.11). A equação (B.31) fica analiticamente definida por:

$${}^m \mathbf{c}^N(P) = \frac{1}{8\pi G(1-\nu)} \left\{ \int_{2\pi}^0 [(3-4\nu) r_{,\gamma} \delta_{\alpha\beta} - r_{,\alpha} \delta_{\beta\gamma} - r_{,\beta} \delta_{\alpha\gamma} + 2r_{,\alpha} r_{,\beta} r_{,\gamma}] d\phi \right\} N_{\beta\delta}^n(P)$$

Como $n_1 = -\cos \phi$, $n_2 = -\sin \phi$ e lembrando as seguintes relações trigonométricas:

$$\begin{aligned} \int_{2\pi}^0 \cos^4 \phi d\phi &= \int_{2\pi}^0 \sin^4 \phi d\phi = -\frac{3\pi}{4} \\ \int_{2\pi}^0 \sin^3 \phi \cos \phi d\phi &= 0 \\ \int_{2\pi}^0 \sin^2 \phi \cos^2 \phi d\phi &= -\frac{\pi}{4} \\ \int_{2\pi}^0 \sin \phi \cos^3 \phi d\phi &= 0 \quad , \end{aligned}$$

resulta:

$$\boxed{{}^m c_{\alpha\beta}^N(P) = \frac{-1}{8G(1-\nu)} \left[(3-4\nu) \delta_{\alpha\delta} \delta_{\beta\gamma} - \delta_{\alpha\beta} \delta_{\gamma\delta} - \delta_{\alpha\gamma} \delta_{\beta\delta} + \frac{1}{4} \delta_{\alpha\beta} \delta_{\gamma\delta} (1+2\delta_{\alpha\gamma}) \right] N_{\gamma\delta}^n(P)} \quad (\text{B.32})$$

B.4 Cálculo de $\frac{\partial J_\alpha^q}{\partial x_\gamma}(P)$

Aqui também ${}^m\mathbf{V} = {}^m\mathbf{U}$, e como $\mathbf{U} = O(\ln r)$, não se aplica a representação

$$U_{\alpha\beta} = \frac{1}{r(\bar{r}, \bar{\theta})} \Lambda_{\alpha\beta}(\phi) \quad , \quad (\text{B.33})$$

e portanto não há contribuição convectiva. Logo:

$$\boxed{{}^m c_{\alpha\beta}^q(P) = 0} \quad . \quad (\text{B.34})$$

Apêndice C

Expansões Assintóticas para os Tensores Fortemente Singulares

Este apêndice ilustra a metodologia geral para obtenção de expansões assintóticas de núcleos singulares, com vistas à aplicação do método direto conforme descrito no Capítulo 5. Embora o procedimento seja bastante geral, estas expansões são válidas apenas em torno do ponto singular e não devem, em princípio, ser utilizadas para elementos quase-singulares. O procedimento é desenvolvido para elementos de contorno bidimensionais. Células de domínio não serão tratadas, mas o procedimento seria análogo ao descrito aqui (Guiggiani & Gigante 1990, Guiggiani 1998). Deve-se ressaltar que a rigor o interesse não está na integração do núcleo isolado:

$$I_{ij} = \int_{\Gamma} T_{ij} d\Gamma \quad ,$$

mas na integração deste já considerando a forma interpolatória adotada para as variáveis, isto é:

$$I_{ij} = \int_{\Gamma} K_{ij} d\Gamma = \int_{-1}^{+1} T_{ij} \phi_{\alpha} J d\Gamma \quad . \quad (\text{C.1})$$

Aqui serão abordados os núcleos fortemente singulares encontrados em algumas aplicações típicas do MEC como problemas potenciais, elasticidade bidimensional e placas semi-espessas. Os dois últimos casos são interesse direto do presente trabalho. Os núcleos de problemas potenciais foram aqui incluídos porque suas expansões fornecerem uma

prova formal da regularidade destes núcleos.

C.1 Algumas relações úteis

Seja um elemento de contorno Γ_n mapeado para o domínio normalizado $\xi = [-1, +1]$. Seja também $\eta = \xi(p)$ a imagem do ponto fonte sobre ξ . A fim de se obter as expansões assintóticas \mathbf{F}_{-1} e \mathbf{F}_{-2} para os tensores utilizados, serão aqui apresentadas algumas relações úteis no desenvolvimento analítico destas. Relembre-se do Capítulo 5 que os núcleos devem ser expandidos assintoticamente em séries de Laurent em torno do ponto singular (Guiggiani 1998):

$$\mathbf{K}(\xi, \eta) = \frac{\mathbf{F}_{-2}}{\rho^2} + \frac{\mathbf{F}_{-1}}{\rho} + O(1) \quad (\text{C.2})$$

onde $\rho = \xi - \eta$ é a imagem de r no domínio normalizado.

Expandindo-se a projeção de r sobre o eixo i em série de Taylor:

$$x_i^q - x_i^p = \rho A_i + \rho^2 B_i + O(\rho^3)$$

com

$$\begin{aligned} A_i &= \left. \frac{dx_i}{d\xi} \right|_{\xi=\eta} \\ B_i &= \frac{1}{2} \left. \frac{d^2 x_i}{d^2 \xi} \right|_{\xi=\eta} \end{aligned} ,$$

onde verifica-se as relações:

$$\begin{aligned} A^2 &= A_i A_i = |J_n(\eta)|^2 = J(\eta)^2 \\ B^2 &= B_i B_i \\ C &= A_i B_i \quad . \end{aligned}$$

Outras relações envolvendo o Jacobiano podem ser obtidas. Em especial, as seguintes

identidades são verificadas para elementos bidimensionais:

$$J = |J_n| = \sqrt{\frac{dx_\alpha}{d\xi} \frac{dx_\alpha}{d\xi}} = \sqrt{J_1(\xi)^2 + J_2(\xi)^2}$$

onde,

$$\begin{aligned} J_1 &= A_2 + 2B_2\rho + O(\rho^2) \\ J_2 &= -A_1 - 2B_1\rho + O(\rho^2) \end{aligned}$$

ou:

$$\boxed{J_i = J_{i_0} + \rho J_{i_1}} + O(\rho^2) \quad .$$

Adicionalmente,

$$\boxed{J_i = n_i J} \tag{C.3a}$$

$$\boxed{A_i J_{i_0} = 0} \tag{C.3b}$$

Expandindo-se as potências de r em série de Taylor:

$$r^n = A^n |\rho|^n \left(1 + n\rho \frac{C}{A^2} \right) + O(\rho^{n+2}) \quad ,$$

logo,

$$\boxed{r^{-1} = \frac{1}{A|\rho|} - \text{sign}(\rho) \frac{C}{A^3}} + O(\rho) \tag{C.4a}$$

$$\boxed{r^{-2} = \frac{1}{A^2\rho^2} - \frac{2C}{A^4\rho}} + O(1) \tag{C.4b}$$

Agora, da definição da projeção de r sobre a direção coordenada i :

$$\begin{aligned}
r_{,i} &= \frac{x_i^q - x_i^p}{r} \\
&= \frac{\rho A_i + \rho^2 B_i + O(\rho^3)}{A |\rho| \left(1 + \frac{C}{A^2 \rho}\right) + O(\rho^3)} \\
&= \text{sign}(\rho) \frac{A_i}{A} + |\rho| \left(\frac{B_i}{A} - \frac{A_i C}{A^3}\right) + O(\rho^2)
\end{aligned}$$

Então:

$$\boxed{r_{,i} = d_{i_0} + \rho d_{i_1}} + O(\rho^2) \tag{C.5}$$

onde:

$$\begin{aligned}
d_{i_0} &= \text{sign}(\rho) \frac{A_i}{A} \\
d_{i_1} &= \text{sign}(\rho) \left(\frac{B_i}{A} - \frac{A_i C}{A^3}\right)
\end{aligned}$$

Agora, considerando-se que os núcleos a serem integrados são normalmente multiplicados pelo Jacobiano do elemento, e fazendo-se uso das relações (C.3a) e (C.5), pode-se escrever algumas relações úteis:

$$\boxed{r_{,n} = r_{,i} n_i} \tag{C.6a}$$

$$\boxed{r_{,n} J = r_{,i} J_i} \tag{C.6b}$$

$$\boxed{r_{,i} J_i = \rho f_1} + O(\rho^2) \tag{C.6c}$$

$$\boxed{r_{,k} (r_{,i} J_i) = \rho g_{k_1}} + O(\rho^2) \tag{C.6d}$$

$$\boxed{r_{,n} r_{,i} r_{,j} J = \rho h_{ij_1}} + O(\rho^2) \tag{C.6e}$$

$$\boxed{n_i r_{,j} J = l_{ij_0} + \rho l_{ij_1}} + O(\rho^2) \tag{C.6f}$$

onde:

$$\begin{aligned}
 f_1 &= \frac{A_i J_{i_1} + \text{sign}(\rho) B_i J_{i_0}}{A} \\
 g_{k_1} &= \frac{\text{sign}(\rho) A_k}{A} f_1 \\
 h_{ij_1} &= \frac{\text{sign}(\rho) A_i}{A} g_{j_1} \\
 l_{ij_0} &= J_{i_0} d_{j_0} \\
 &= J_{i_0} \frac{\text{sign}(\rho) A_j}{A} \\
 l_{ij_1} &= \text{sign}(\rho) J_{i_0} d_{j_1} + J_{i_1} d_{j_0} \\
 &= J_{i_0} \left(\frac{B_j}{A} - \frac{A_j C}{A^3} \right) + J_{i_1} \frac{\text{sign}(\rho) A_j}{A}
 \end{aligned}$$

Note-se ainda que as componentes do sistema de coordenadas local na vizinhança de η são identificadas com as seguintes relações:

$$\boxed{n_i = \frac{J_i}{J} = \frac{J_i}{A}}$$

$$\boxed{t_i = \frac{A_i}{J}}$$

Finalmente, expande-se as funções de interpolação em torno do ponto singular:

$$\phi_a(\xi) = \phi_a(\eta) + \left. \frac{d\phi_a}{d\xi} \right|_{\xi=\eta} (\xi - \eta) + \dots,$$

ou

$$\boxed{\phi_a(\xi) = \phi_{a_0} + \phi_{a_1} \rho} + O(\rho^2)$$

C.2 Expansões assintóticas para $\frac{r, n}{r}$

O termo $\frac{r, n}{r}$ é comum em muitas soluções fundamentais. Embora se saiba que este é um termo regular, ainda é objeto de muita confusão, em vista da presença do denominador r . Existem discussões empíricas sobre a regularidade deste termo, mas não se tem conhecimento de uma demonstração formal. A ocorrência mais simples deste termo se dá na solução fundamental para fluxo em problemas potenciais. No caso bidimensional, esta

solução fundamental é dada por (Brebbia et al. 1984):

$$T = \frac{-1}{2\pi r} r_{,n} \quad .$$

Então reescreve-se o integrando da integral (C.1) como o escalar:

$$K = \frac{-1}{2\pi r} r_{,n} \phi_a J \quad .$$

Substituindo-se (C.4a) e (C.6) resulta na seguinte expansão de K em torno do ponto singular:

$$K = \frac{-1}{2\pi r} \left[\text{sign}(\rho) \frac{f_1}{A} - |\rho| \frac{f_1 C}{A^3} \right] \phi_{a_0} + O(\rho^2) \quad ,$$

e, comparando-se este resultado com a eq.(C.2) torna-se claro que

$$\mathbf{F}_{-2} = \mathbf{F}_{-1} = 0 \quad .$$

Em outras palavras, foi demonstrado o seguinte limite inferior:

$$\boxed{\frac{r_{,n}}{r} = O(1)} \quad (C.7)$$

C.3 Expansões assintóticas para ${}^m T_{\alpha\beta}$

A solução fundamental para deslocamentos de membrana possui no máximo singularidades logarítmicas, e portanto não será considerada aqui. Por outro lado, é sabido que a solução fundamental para trações possui singularidades fortes, sendo sua expressão geral dada no Apêndice A (o sobre-índice m será omitido ao longo desta seção):

$$T_{\alpha\beta} = \frac{-1}{4\pi(1-\nu)r} \left\{ [(1-2\nu)\delta_{\alpha\beta} + 2r_{,\alpha}r_{,\beta}] r_{,n} - (1-2\nu)(n_{\alpha}r_{,\beta} - n_{\beta}r_{,\alpha}) \right\}$$

A forma geral de um núcleo envolvendo $T_{\alpha\beta}$ pode ser expressa como:

$$K_{\alpha\beta} = T_{\alpha\beta} \phi_a J$$

ou:

$$K_{\alpha\beta} = \frac{-1}{4\pi(1-\nu)} [(1-2\nu)A_{\alpha\beta} + 2B_{\alpha\beta} + (1-2\nu)C_{\alpha\beta}]$$

onde:

$$\begin{aligned} A_{\alpha\beta} &= \frac{1}{r} (r_{,n} \delta_{\alpha\beta}) \phi_a J \\ B_{\alpha\beta} &= \frac{1}{r} (r_{,n} r_{,\alpha} r_{,\beta}) \phi_a J \\ C_{\alpha\beta} &= \frac{1}{r} (n_{\alpha} r_{,\beta} - n_{\beta} r_{,\alpha}) \phi_a J \end{aligned}$$

Desenvolvendo cada termo separadamente, resulta:

$$\begin{aligned} A_{\alpha\beta} &= \frac{r_{,\gamma} J_{\gamma} \delta_{\alpha\beta}}{r} \phi_a \\ &= \frac{\rho}{r} f_1 \delta_{\alpha\beta} \phi_a \\ &= \rho \left(\frac{1}{A|\rho|} - \text{sign}(\rho) \frac{C}{A^3} \right) f_1 \delta_{\alpha\beta} (\phi_{a_0} + \phi_{a_1} \rho) \\ &= \frac{f_1}{A} \left[\text{sign}(\rho) \phi_{a_0} + \left(\phi_{a_1} - \frac{C}{A^2} \phi_{a_0} \right) \rho - \left(\frac{\text{sign}(\rho) C}{A^2} \phi_{a_1} \right) \rho^2 \right] \delta_{\alpha\beta} \\ &= \frac{f_1}{A} [O(1) + O(\rho) + O(\rho^2)] \delta_{\alpha\beta} \quad . \end{aligned}$$

$$\begin{aligned} B_{\alpha\beta} &= \frac{r_{,\gamma} n_{\gamma} r_{,\alpha} r_{,\beta} J}{r} \phi_a \\ &= \frac{r_{,\gamma} J_{\gamma} r_{,\alpha} r_{,\beta}}{r} \phi_a \\ &= \frac{\rho}{r} g_{\alpha_1} r_{,\beta} \phi_a \\ &= \frac{\rho}{r} g_{\alpha_1} (d_{\beta_0} + \rho d_{\beta_1}) \phi_a \\ &= \frac{\rho}{r} g_{\alpha_1} d_{\beta_0} \phi_a + O(\rho^2) \\ &= \rho \left(\frac{1}{A|\rho|} - \text{sign}(\rho) \frac{C}{A^3} \right) g_{\alpha_1} d_{\beta_0} \phi_a \\ &= \rho \left(\frac{1}{A|\rho|} - \text{sign}(\rho) \frac{C}{A^3} \right) g_{\alpha_1} d_{\beta_0} (\phi_{a_0} + \phi_{a_1} \rho) \\ &= \frac{g_{\alpha_1} d_{\beta_0}}{A} \left[\text{sign}(\rho) \phi_{a_0} + \left(\phi_{a_1} - \frac{\text{sign}(\rho) C}{A^2} \phi_{a_0} \right) \rho - \left(\frac{\text{sign}(\rho) C}{A^2} \phi_{a_1} \right) \rho^2 \right] \\ &= \frac{g_{\alpha_1} d_{\beta_0}}{A} [O(1) + O(\rho) + O(\rho^2)] \quad . \end{aligned}$$

$$\begin{aligned}
C_{\alpha\beta} &= \frac{l_{\alpha\beta_0} - l_{\beta\alpha_0} + (l_{\alpha\beta_1} - l_{\beta\alpha_1})\rho}{r} \phi_a \\
&= \left(\frac{1}{A|\rho|} - \text{sign}(\rho) \frac{C}{A^3} \right) [m_{\alpha\beta_0} + m_{\alpha\beta_1}\rho] \phi_a \\
&= \left\{ \frac{\text{sign}(\rho)}{A} \left(m_{\alpha\beta_1} - m_{\alpha\beta_0} \frac{C}{A^2} \right) - \left(\text{sign}(\rho) m_{\alpha\beta_1} \frac{C}{A^3} \right) \rho + \left(\text{sign}(\rho) \frac{m_{\alpha\beta_0}}{A} \right) \frac{1}{\rho} \right\} \phi_a \\
&= \left[O(1) + O(\rho) + \left(\text{sign}(\rho) \frac{m_{\alpha\beta_0}}{A} \right) \frac{1}{\rho} \right] (\phi_{a_0} + \phi_{a_1}\rho) \\
&= \left(\text{sign}(\rho) \frac{m_{\alpha\beta_0}\phi_{a_0}}{A} \right) \frac{1}{\rho} + O(1) + O(\rho) + O(\rho^2) \quad .
\end{aligned}$$

onde

$$\begin{aligned}
m_{\alpha\beta_0} &= l_{\alpha\beta_0} - l_{\beta\alpha_0} \\
m_{\alpha\beta_1} &= l_{\alpha\beta_1} - l_{\beta\alpha_1}
\end{aligned}$$

E comparando-se estes termos com a série de Laurent (C.2) conclui-se que:

$$\begin{aligned}
\mathbf{F}_{-2}^A &= \mathbf{F}_{-1}^A = 0 \\
\mathbf{F}_{-2}^B &= \mathbf{F}_{-1}^B = 0 \\
\mathbf{F}_{-2}^C &= 0 \quad ,
\end{aligned}$$

mas

$$\mathbf{F}_{-1}^C = \text{sign}(\rho) \frac{m_{\alpha\beta_0}\phi_{a_0}}{A} \quad .$$

Como $\mathbf{F}_{-n} = \mathbf{F}_{-n}^A + \mathbf{F}_{-n}^B + \mathbf{F}_{-n}^C$, resulta que:

$$\begin{aligned}
\mathbf{F}_{-1} &= \text{sign}(\rho) \frac{m_{\alpha\beta_0}\phi_{a_0}}{A} = \text{sign}(\rho) \frac{(l_{\alpha\beta_0} - l_{\alpha i_0})\phi_{a_0}}{A} \\
&= \frac{(J_{\alpha_0}A_\beta - J_{\beta_0}A_\alpha)\phi_{a_0}}{A^2} \\
&= \frac{(n_\alpha A_\beta - n_\beta A_\alpha)\phi_{a_0}}{A} = \frac{(n_\alpha A_\beta - n_\beta A_\alpha)\phi_{a_0}}{J} \quad .
\end{aligned}$$

Portanto:

$$\boxed{{}^m\mathbf{F}_{-1} = -\frac{(1-2\nu)}{4\pi(1-\nu)} [n_\alpha(p)t_\beta(p) - n_\beta(p)t_\alpha(p)] \phi_a(\eta)} \quad (\text{C.8a})$$

$$\boxed{{}^m\mathbf{F}_{-2} = 0} \quad . \quad (\text{C.8b})$$

C.4 Expansões assintóticas para ${}^fT_{ij}$

Aqui também apenas a solução fundamental para trações possui singularidades fortes. De acordo com as expressões apresentadas no Apêndice A, verifica-se que as componentes ${}^fT_{\alpha 3}$, ${}^fT_{3\alpha}$ e ${}^fT_{33}$ são, no máximo, fracamente singulares. Assim, as únicas componentes que merecem tratamento especial são correspondentes aos momentos generalizados (o sobre-índice f será omitido):

$$\begin{aligned} {}^fT_{\alpha\beta} = & \frac{-1}{4\pi r} \left[(4A + 2zK_1 + 1 - \nu)(r_{,\beta}n_\alpha + r_{,n}\delta_{\alpha\beta}) + \right. \\ & \left. + (4A + 1 + \nu)r_{,\alpha}n_\beta - 2(8A + 2zK_1 + 1 - \nu)r_{,\alpha}r_{,\beta}r_{,n} \right] \quad (\text{C.9}) \end{aligned}$$

Seguindo o mesmo procedimento da seção anterior, a forma geral de $T_{\alpha\beta}$ é escrita:

$$K_{\alpha\beta} = \frac{-1}{4\pi} [A_{\alpha\beta} + B_{\alpha\beta} + C_{\alpha\beta}] \quad (\text{C.10})$$

onde:

$$\begin{aligned} A_{\alpha\beta} &= \frac{1}{r} [(4A + 2zK_1 + 1 - \nu)(r_{,\beta}n_\alpha + r_{,n}\delta_{\alpha\beta})] \phi_a J \\ B_{\alpha\beta} &= \frac{1}{r} [(4A + 1 + \nu)r_{,\alpha}n_\beta] \phi_a J \\ C_{\alpha\beta} &= \frac{-1}{r} [2(8A + 2zK_1 + 1 - \nu)r_{,\alpha}r_{,\beta}r_{,n}] \phi_a J \quad . \end{aligned}$$

Antes de proceder o desenvolvimento destes termos, as combinações $A(z)$ e $B(z)$ são expandidas em séries:

$$A(z) = -\frac{1}{2} + \left(\frac{3}{4} - \gamma - \ln\left(\frac{z}{2}\right) \right) \frac{z^2}{8} \quad (\text{C.11a})$$

$$B(z) = -\frac{1}{4} - \frac{1}{2}\gamma + \frac{3}{16} \left(\frac{11}{12} - \gamma \right) z^2 - \frac{1}{2} \ln\left(\frac{z}{2}\right) - \frac{1}{2} \ln\left(\frac{z}{2}\right) z^2 \quad (\text{C.11b})$$

de onde fica claro que estas combinações se comportam de forma regular e fracamente singular, respectivamente, em torno de $z = 0$. Manipulando-se as expressões (C.11) resultam os seguintes comportamentos assintóticos:

$$4A(z) = -2 + O(z^2) \quad (\text{C.12a})$$

$$4A(z) + 2zK_1(z) = O(z^2) \quad (\text{C.12b})$$

$$8A(z) + 2zK_1(z) = -2 + O(z^2) \quad (\text{C.12c})$$

Substituindo-se as relações (C.12) em $A_{\alpha\beta}$, $B_{\alpha\beta}$ e $C_{\alpha\beta}$, e levando-se em conta o comportamento assintótico de $r_{,n}/r$ dado por (C.7), resultam os termos com contribuições fortemente singulares de (C.10):

$$\begin{aligned} A_{\alpha\beta}^s &= \frac{-1}{r} [(1-\nu) r_{,\beta} n_\alpha] \phi_a J \\ B_{\alpha\beta}^s &= \frac{1}{r} [(1-\nu) r_{,\beta} n_\alpha] \phi_a J \\ C_{\alpha\beta}^s &= 0 \quad . \end{aligned}$$

E portanto, a parcela singular de (C.10) é reescrita:

$$K_{\alpha\beta}^s = -\frac{(1-\nu)}{4\pi r} [r_{,\beta} n_\alpha - r_{,\alpha} n_\beta] \phi_a J$$

Manipulando-se os termos de $K_{\alpha\beta}^s$ de forma análoga ao tensor de elasticidade bidimensional resulta

$$\boxed{f\mathbf{F}_{-1} = \frac{(1-\nu)}{4\pi} [n_\alpha(p) t_\beta(p) - n_\beta(p) t_\alpha(p)] \phi_a(\eta)} \quad (\text{C.13a})$$

$$\boxed{f\mathbf{F}_{-2} = 0} \quad , \quad (\text{C.13b})$$

que possuem a mesma forma operacional das expansões para elasticidade bidimensional. Segundo Guiggiani (1998), esta é uma característica comum nas soluções fundamentais da elasticidade. Não se tem conhecimento da publicação da expansão (C.13) na literatura.