UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA MECÂNICA
COMISSÃO DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA

ANÁLISE DE SENSIBILIDADE, ALGORITMOS DE OTIMIZAÇÃO E ORIENTAÇÃO POR OBJETOS EM HIPERELASTICIDADE NÃO-LINEAR

Autor: Cláudio Alessandro de Carvalho Silva
Orientador: Prof. Dr. Marco Lúcio Bittencourt

11/03

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA MECÂNICA
COMISSÃO DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA
DEPARTAMENTO DE PROJETO MECÂNICO

ANÁLISE DE SENSIBILIDADE, ALGORITMOS DE OTIMIZAÇÃO E ORIENTAÇÃO POR OBJETOS EM HIPERELASTICIDADE NÃO-LINEAR

Autor: Cláudio Alessandro de Carvalho Silva
Orientador: Prof. Dr. Marco Lúcio Bittencourt

Curso: Engenharia Mecânica
Área de Concentração: Mecânica dos Sólidos e Projeto Mecânico

Tese de doutorado apresentada à comissão de Pós-Graduação da Faculdade de Engenharia Mecânica como requisito para a obtenção de título de Doutor em Engenharia Mecânica.

Campinas, 2003
S.P. - Brasil

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA MECÂNICA
COMISSÃO DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA
DEPARTAMENTO DE PROJETO MECÂNICO

TESE DE DOUTORADO

ANÁLISE DE SENSIBILIDADE, ALGORITMOS DE OTIMIZAÇÃO E ORIENTAÇÃO POR OBJETOS EM HIPERELASTICIDADE NÃO-LINEAR

Autor: Cláudio Alessandro de Carvalho Silva
Orientador: Prof. Dr. Marco Lúcio Bittencourt

Prof. Dr. Marco Lúcio Bittencourt, Presidente
DPM/FEM/UNICAMP

Prof. Dr. José Herskovits
COPPE/UFRJ

Prof. Dr. Sérgio Persival Baroncini Proença
EESC/USP

Prof. Dr. Renato Pavanello DMC/FEM/UNICAMP

Prof. Dr. Janito Vaqueiro Ferreira
DMC/FEM/UNICAMP

Campinas, 13 de junho de 2003.

Dedico este trabalho a todos os professores que já tive. Vocês me mostraram que sempre haverá muito mais a aprender.

Agradecimentos

Ao Prof. Dr. Marco Lúcio Bittencourt pelas oportunidades que me proporcionou e pela orientação, confiança e incentivo.

Ao Prof. Dr. Kyung K. Choi e ao colega Dr. Nam Ho Kim por toda e orientação e ajuda enquanto estive em Iowa. Além da formação técnica, sempre guardarei como exemplo o altruísmo que demonstraram.

Ao Prof. Dr. Kamal Abel Radi Ismail pelo apoio e colaboração como coordenador da CPG/FEM e da Sub-CPG/FEM.

Aos amigos Alberto Costa Nogueira Júnior e Renato Fernandes Cantão pela confiança incondicional, conselhos e incentivo nos momentos de dificuldade. Considero uma grande sorte a oportunidade de trabalharmos juntos.

Aos companheiros de laboratório Luciano Driemeier, Wallace Gusmão, Carlos Eduardo Leite Pereira e Maurício Barbatto pelo companheirismo, amizade e ótimo clima de trabalho durante os anos em que trabalhamos juntos.

Aos colegas de pós-graduação e amigos Cristina Minioli Saracho e Rodrigo Nicoletti pelos exemplos de dedicação ao trabalho e profissionalismo.

A FAPESP, DPM, FEM e UNICAMP pelos suportes financeiro e institucional que possibilitaram a realização deste trabalho.

E, especialmente, à Ellen pela compreensão, carinho e dedicação em todos os momentos.

A melhor arma contra o medo é o conhecimento.

Jo-Ellan Dimitrius

Resumo

SILVA, Cláudio Alessandro de Carvalho, *Análise de Sensibilidade, Algoritmos de Otimização e Orientação por Objetos em Hiperelasticidade Não-Linear,* Campinas, Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas, 2003, 297p., Tese (Doutorado).

Neste trabalho, tem-se como objetivo principal desenvolver e implementar ferramenta de otimização de forma de componentes estruturais 2D e 3D apresentando material hiperelástico não-linear quasi-incompressível e valores finitos de deslocamentos e deformações. Para isso, formula-se a análise de resposta na abordagem Lagrangeana, introduzindo-se equação de restrição para tratamento da incompressibilidade, a qual é então resolvida por um método de projeção. Segue-se então a formulação de análise de sensibilidade utilizando o conceito de derivada material da Mecânica do Contínuo. Análises de resposta e sensibilidade são resolvidas numericamente pelo método de elementos finitos. A geometria dos componentes são definidas por curvas e superfícies NURBS, cujas características são utilizadas como variáveis de projeto do problema de otimização. As derivadas da geometria em relação às variáveis de projeto definem campos de velocidade sobre o contorno. São apresentadas técnicas para extrapolar tais campos para o interior do domínio respeitando os requisitos teóricos e práticos das formulações de análise de sensibilidade e otimização, bem como as condições e a maneira de utilizar tais campos na atualização da discretização na seqüência de alterações da geometria. Tais técnicas são utilizadas em conjunto com algoritmo de pontos interiores baseado na solução iterativa das condições de otimalidade de Karush- Kuhn-Tucker. Por fim, são discutidos aspectos de desenvolvimento e implementação de sistemas de software utilizando processo formal de desenvolvimento (Processo Unificado), linguagem de modelagem (UML) e orientação por objetos em C++.

Palavras-chave

Otimização Estrutural, Elementos Finitos, Elasticidade Linear e Não-Linear, Análise de Sensibilidade, Movimentação de Malhas, Programação Orientada a Objetos, C++, Programação Não-Linear.

Abstract

SILVA, Cláudio Alessandro de Carvalho, *Sensitivity Analysis, Otimization Algorithms and Object-Orientation in Nonlinear Hiperelasticity,* Campinas: Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas, 2003, 296p., Tese (Doutorado).

In this work, the main purpose is the development and implementtaion of a shape optimization tool for 2D and 3D structural components featuring nonlinear nearly-incompressible hyperelastic material and finite deformation and strain. For that task, the response analysis is formulated in the Lagrangian approach and a constraint equation is imposed for enforcing the incompressibility condition, which is solved by a projection method. This is followed by sensitivity analysis formulation that employs the material derivative concept from Continuum Mechanics. Response and sensitivity analysis are solved by the finite element method. The geometry of the components are defined by NURBS curves and surfaces whose characteristics are used as design variables. The derivatives of the geometry relative to the design variables define design velocity fields on the boundary. Techniques for extrapolating those fields to the interior of the domain which respect the theoretical and practical requirements of the sensitivity analysis and optimization techniques are presented, as well as the conditions and the way for applying those fields for updating the finite element discretization during the shape optimization sequence. Those techniques are associated to an interior point algorithm based on the iterative colution of the Karush-Kuhn-Tucker optimality conditions. At last, concepts of software system development and implementation using a formal development process (Unified Process), modeling language (UML) and object-orientation in C++ are discussed.

Keywords

Structural Optimization, Finite Elements, Linear and Nonlinear Elasticity, Sensitivity Analysis, Mesh Deforming, Object-oriented programming, C++, Nonlinear Programming.

Índice

Lista de Figuras

Lista de Tabelas

Nomenclatura

Conjuntos de Tensores

Símbolo	*Significado*
Lin	Conjunto de todos os tensores.
Lin^+	Conjunto de todos os tensores com determinante positivo.
Sym	Conjunto de todos os tensores simétricos.
Skw	Conjunto de todos os tensores antissimétricos.
Psym	Conjunto de todos os tensores simétricos positivo-definidos.
Orth	Conjunto de todos os tensores ortogonais.
Orth^+	Conjunto de todas as rotações.

Operadores

Símbolo	*Significado*
$(\dot{\ })$	Derivação de um campo material em relação a um parâmetro escalar. Se aplicado a campo espacial indica a derivada *total* do campo em relação ao parâmetro.
$(')$	Derivação de um campo espacial em relação a um parâmetro escalar mantendo-se fixo $\mathbf{x}$. Indica a derivada *parcial* do campo em relação ao parâmetro.
∇	Gradiente de campo material em relação ao ponto material $\mathbf{X}$.
grad	Gradiente de campo espacial em relação ao ponto espacial $\mathbf{x}$.
Div	Divergente de campo material.
div	Divergente de campo espacial.

Letras Latinas

Símbolo	*Significado*	*Definição (pag.)*
$a(\cdot,\cdot)$	Forma linear na segunda entrada, utilizada em enunciado variacional	50, 52, 59
A_{ijk}, A_{ij}	Constantes da expansão em série de W	47
$b_1(\cdot,\cdot)$	Forma linear na primeira entrada, utilizada em enunciado variacional misto	62
$b_2(\cdot,\cdot)$	Forma linear na segunda entrada, utilizada em enunciado variacional misto	62
$g(\cdot,\cdot)$	Forma linear na segunda entrada, utilizada em enunciado variacional misto	62
$\mathbf{b}$	Força por unidade de volume	238
$\mathbf{B}$	Tensor de deformação esquerdo de Cauchy-Green	233
$\mathbf{B}_F$	Matriz de implemetação de $\mathbf{F}$	266
$\mathbf{B}_G$	Matriz de derivadas parciais das funções de forma	264
$\mathbf{B}_L$	Matriz de deformação linear	265
$\mathbf{B}_N$	Matriz de deformação não-linear	266
$\mathcal{B}$	Configuração de referência de um corpo material	227
$\mathcal{B}_e$	Domínio do elemento e na configuração de referência	51
$\mathcal{B}_{ref}$	Domínio do elemento padrão	55
$\mathcal{B}_t$	Configuração no tempo t devido ao movimento x	235
$\mathbf{C}$	Tensor de deformação direito de Cauchy-Green	233
C	Tensor de elasticidade tangente	40, 242, 243
$\bar{\mathsf{C}}$	Parcela de C relacionada à distorção	63
$\tilde{\mathsf{C}}$	Parcela de C relacionada à dilatação	63
$\mathbf{d}$	Vetor de variáveis de projeto	28
$\mathbf{D}$	Parte simétrica de $\mathbf{L}$	237
$\tilde{E}$	Módulo de elasticidade	245
$\mathbf{E}$	Tensor lagrangeano de deformação finita	233
$\mathcal{E}$	Espaço Euclidiano tridimensional	227

Símbolo	*Significado*	*Definição (pag.)*
f	Função objetivo	28
$\mathbf{f}$	Alteração de configuração de um corpo	230
$\hat{\mathbf{f}}$	Vetor de carregamentos discretos	51
$\mathbf{F}$	Gradiente do movimento ou mudança de configuração	231, 236
F	Gradiente da modificação de projeto	85
g_i	Restrição de desigualdade	28
$\mathcal{G}$	Integrando de ψ	82
G	Função de compressibilidade	49
G^*	Funcional conjugado de G	56
$\mathbf{G}$	Matriz auxiliar $\mathbf{G} = \mathbf{B}_N^T [J_{,\mathbf{E}}]$	275
h	Função escalar de restrição	42
h_j	Restrição de igualdade	28
I	Intervalo de ocorrência do fenômeno em estudo	57
I_{n+1}	Subintervalo de I	57
I_1, I_2, I_3	Invariantes do tensor $\mathbf{C}$	41
$\bar{I}_1, \bar{I}_2$	Invariantes reduzidos	45
$\mathbf{I}$	Matriz identidade (tensor de segunda ordem)	233
I	Tensor identidade simétrico de quarta ordem	230
J	Determinante de $\mathbf{F}$	41
$\mathbf{K}$	Matriz de rigidez global	51
$l(\cdot)$	Forma linear associada ao carregamento em enunciados variacionais	59
$\mathbf{l}$	Elemento do espaço dual $\mathcal{V}'$	52
$\mathbf{L}$	Gradiente de $\mathbf{v}$	236
$\mathbf{m}_A$	Momento angular	238
$\mathbf{m}_L$	Momento linear	238
$\mathbf{m}$	Vetor normal unitário a um ponto de uma dada superfície	240
$\mathbf{n}$	Vetor normal unitário a um ponto de uma dada superfície	237
N	Dimensão do espaço discreto $\mathcal{V}_h$, número total de nós	51

Símbolo	*Significado*	*Definição (pag.)*
$\mathbf{N}$	Matriz de funções de forma	264
N^i	Função de forma associada ao nó i do elemento e	263
Nn	Número de nós do elemento e	263
$\mathbf{o}$	Origem adotada pelo observador do movimento	238
O	Observador	245
p	Pressão hidrostática, elemento do espaço $\mathcal{Q}$	259
$\hat{p}_e$	Pressão hidrostática condensada no elemento e	68
$\mathcal{P}$	Parte do corpo $\mathcal{B}$ (subdomínio regular de $\mathcal{B}$)	238
$\mathcal{P}_t$	Configuração da parte $\mathcal{P}$ em t devido ao movimento x	238
q	Multiplicador de Lagrange da equação constitutiva restrita	43, 259
$\mathcal{Q}$	Espaço de pressões admissíveis	259
$\mathbf{R}$	Tensor de rotação da decomposição polar	232
$\mathbf{s}$	Densidade de força de superfície na lei de Cauchy	237
s	Variável do problema de hiperelasticidade incompressível	62
$\mathbf{S}$	Segundo tensor de Piola-Kirchhoff	241
$\bar{\mathbf{S}}$	Parcela de $\mathbf{S}$ devida à distorção	63
$\tilde{\mathbf{S}}$	Parcela de $\mathbf{S}$ devida à dilatação	63
t	Tempo	227
$\bar{\mathbf{t}}$	Condição de contorno natural	253
$\mathbf{T}$	Primeiro tensor de Piola-Kirchhoff	240
T_τ	Modificação de projeto, transformação de domínio	82
$\mathcal{T}$	Trajetória	235
$\mathcal{T}_h$	Triangularização	51
$\mathbf{u}$	Deslocamento, função vetorial do espaço $\mathcal{V}$	231
$\mathbf{u}_h$	Aproximação de $\mathbf{u}$ no espaço de dimensão finita $\mathcal{V}_h$	50
$\mathbf{u}_e$	Deslocamento do ponto (ζ, η, ς) no elemento e	263
$\mathbf{u}_e^i$	Deslocamento do nó i do elemento e	263
$\hat{\mathbf{u}}$	Vetor de coeficientes de $\mathbf{u}_h$ de dimensão N	51
$\mathbf{U}$	Tensor de deformação da decomposição polar direita	232

Símbolo	*Significado*	*Definição (pag.)*
$\mathbf{v}$	Descrição espacial da velocidade	236
$\mathbf{V}$	Tensor de deformação da decomposição polar esquerda	232
$\mathbf{V}_s$	Campo de velocidades de projeto	83
$\mathcal{V}$	Espaço de deslocamentos admissíveis, espaço de Hilbert	257
$\mathcal{V}_h$	Subespaço de dimensão finita de $\mathcal{V}$	50
$\mathbf{x}$	Ponto espacial, vetor pertencente a $\Re^2$ ou $\Re^3$	230
$\mathbf{x}_e^i$	Coordenada espacial do nó i do elemento e	263
$\mathbf{x}_e$	Coordenada espacial do ponto (ζ, η, ς) no elemento e	263
$\mathbf{X}$	Ponto material, vetor pertencente a $\Re^2$ ou $\Re^3$	230
$\mathbf{X}_e^i$	Coordenada material do nó i do elemento e	263
$\mathbf{X}_e$	Coordenada material do ponto (ζ, η, ς) no elemento e	263
x	Movimento de um ponto material, $\mathbf{x} = \mathsf{x}(\mathbf{X}, t)$	235
$\bar{\mathsf{x}}$	Condição de contorno essencial	253
X	Transformação de referência do movimento, $\mathbf{X} = \mathsf{X}(\mathbf{x}, t)$	235
$\mathbf{w}$	Elemento do espaço $\mathcal{V}$	50
$\mathbf{w}_h$	Elemento do espaço $\mathcal{V}_h$	50
$\mathbf{w}_{ih}$	i-ésimo vetor da base de $\mathcal{V}_h$	51
W	Densidade de energia de deformação	38
$\bar{W}$	Densidade de energia de distorcão	47, 48
$\tilde{W}$	Densidade de energia de dilatação	48
W_{ext}	Trabalho das forças externas	257
W_{int}	Trabalho das forças internas	59, 63
$\mathbf{W}$	Parte antissimétrica de $\mathbf{L}$	237

Letras Gregas

Símbolo	*Significado*	*Definição (pag.)*
Γ_1	Região do contorno onde são aplicadas as condições essenciais	253
Γ_2	Região do contorno onde são aplicadas as condições naturais	253
δ_{ij}	Delta de Kronecker	233
$\boldsymbol{\epsilon}$	Tensor de deformação infinitesimal	234
$\iota_1, \iota_2, \iota_3$	Invariantes do tensor $\mathbf{U}$	41
$\tilde{\kappa}$	bulk modulus	49, 244
$\lambda_1, \lambda_2, \lambda_3$	Autovalores do tensor $\mathbf{U}$	41
$\tilde{\lambda}$	Módulo de Lamé	243
$\tilde{\mu}$	Módulo de Lamé	243
$\tilde{\nu}$	Coeficiente de Poisson	245
Π	Funcional de energia potencial	257
ρ	Densidade	238
$\boldsymbol{\Xi}$	Matriz auxiliar formada pelas componentes de $\mathbf{S}$	262, 267, 269
$\boldsymbol{\sigma}$	Tensor de tensões de Cauchy	238
τ	Parâmetro escalar de modificação do domímio $\mathcal{B}$	83
χ	Elemento de $\mathcal{V}$	257
ζ, η, ς	Coordenadas no elemento de referência $\mathcal{B}_{ref}$	263
ψ	Funcional de performance genérico	82
$\boldsymbol{\Upsilon}$	Equação constitutiva para o primeiro tensor de Piola-Kirchhoff	37
Ω	Região viável do problema de otimização	28

Capítulo 1

Introdução

1.1 Motivação

Tecnicamente, um '*projeto é um empreendimento temporário com o objetivo de criar um produto ou serviço único* [...]. *Único* significa que o produto ou serviço produzido é de alguma forma diferente de todos os outros produtos ou serviços semelhantes' (Project Management Institute, 2000). Pode-se inferir, portanto, que o projeto é motivado pela identificação de uma necessidade ou insatisfação não correspondida pelos produtos disponíveis, exigindo, em maior ou menor grau, *inovação* e *originalidade*.

Além da definição puramente técnica, inovação e diferenciação são atualmente pré-requisitos básicos num contexto de mercados competitivos, consistindo em fatores de sobrevivência de empresas em quase todos os setores da economia. Simultaneamente, têm-se exigências cada vez mais severas (em termos legais e em percepção dos consumidores) quanto a menor consumo de energia e impacto ambiental, o que se traduz em busca por eficiência, menores peso, arrasto, ruído, emissões, resíduos e preço. Em resumo, apesar dos critérios de julgamento das soluções adotadas dependerem do contexto sócio-econômico e de nem sempre serem claramente definidos ou compreendidos, é uma constante na atividade de projeto o questionamento sobre a qualidade das alternativas possíveis e a busca daquelas que tenham *a melhor* performance em algum sentido.

Quando a demanda por inovações é baixa, é razoável supor que novos projetos possam ser obtidos de pequenas alterações de trabalhos anteriores. Nesses casos, pode-se basear quase exclusivamente na experiência em projetos semelhantes. Porém, num ambiente de competição tecnológia, as demandas por qualidade, baixo custo, pouco impacto ambiental e tempo de desenvolvimento cada vez mais restrito exigem a criação de produtos para os quais inexiste experiência anterior.

Em decorrência desse ambiente, os diversos campos da engenharia têm sido impulsionados na busca de metodologias e técnicas mais eficazes, precisas e seguras de desenvolvimento de produtos. Certamente, pode-se enquadrar aí o que se observou nos últimos 40 anos nas áreas de modelos matemáticos e métodos numéricos para sua solução computacional, denominadas em conjunto, no contexto de problemas de Mecânica, como Mecânica Computacional.

Com tais técnicas numéricas, a obtenção da reposta dos mais diversos tipos de sistemas tem se tornado cada vez mais viável. Os recursos computacionais disponíveis permitem executar análises numéricas bastante complexas envolvendo, por exemplo, diversos tipos de não-linearidades e modelos de grandes dimensões. A facilidade e o custo decrescente de solução computacional têm deslocado as dificuldades da *possibilidade de executar* análises de resposta para a *capacidade de interpretar* volumes cada vez maiores de dados e propor novas soluções. Em outras palavras, também na busca de novas soluções de engenharia, mostram-se necessárias abordagens sistemáticas e o desenvolvimento de ferramentas que as suportem. A conversão das saídas dos programas de análise de resposta em medidas escalares de performance e a parametrização dos modelos matemáticos dos sistemas são claramente recursos úteis na simplificação da atividade de projeto e proposição de soluções. As etapas seguintes são a inclusão de técnicas de análise de sensibilidade de projeto e acoplamento com algoritmos de programação matemática, fornecendo uma ferramenta automatizada de otimização de projetos, ou seja, uma ferramenta de *síntese* de componentes ou sistemas.

As técnicas de análise de sensibilidade e otimização fornecem critérios objetivos para comparar opções de projeto, permitindo direcionar os esforços de solução. Além disso, a complexidade do problema fica sintetizada através da definição de critérios de performance, representados por funcionais escalares, e parâmetros discretos de controle, também chamados de variáveis de projeto. Matematicamente, a análise de sensibilidade se traduz na determinação de gradientes dos funcionais de performance em relação às variáveis de projeto, e a otimização na determinação de máximos ou mínimos de um ou mais funcionais, tomando outros como condições limites a serem respeitadas. Se, uma vez assumidos valores para as variáveis de projeto dentro de um intervalo, a resposta do sistema pode ser obtida e os funcionais de performance podem ser avaliados, o problema de projeto pode ser formulado como um problema de otimização. Se é possível também obter os gradientes desses funcionais, é possível aplicar técnicas de otimização altamente eficientes.

Atualmente, as técnicas numéricas de análise de resposta estão bastante difundidas na atividade de projeto mecânico. Em contrapartida, a incorporação de técnicas de análise de sensibilidade e otimização (especialmente de forma) no dia-a-dia do projeto tem ocorrido de forma lenta, sendo sua utilização ainda restrita. Uma das causas, é a inexistência de ferramentas comercialmente disponíveis que executem de forma conveniente as diversas competências envolvidas num problema de otimização de forma de estruturas (mesmo aquelas lineares) combinando eficiência, confiabilidade e precisão. Em alguns casos, os recursos de parametrização e atualização de malhas são incipientes. Em outros, a análise de sensibilidade e otimização estão implementados de maneiras reconhecidamente ineficientes, sendo essa justamente a situação dos problemas envolvendo não-linearidades: esse recurso somente é disponível hoje em programas comerciais através de diferenças finitas (Ansys Inc., 1995a; Ansys Inc., 1995b).

De fato, é interessante observar que após décadas de pesquisa em análise de sensibilidade e otimização, as ferramentas de análise numérica de estruturas dominantes do mercado não tenham incorporado as melhores técnicas. Talvez porque tais programas tenham sido concebidos muito antes da maturação das técnicas de análise de sensibilidade e otimização (que ocorreram a partir da metade dos anos 80 para problemas lineares e apenas na década seguinte para problemas não-lineares). Dessa forma, a arquitetura desses sistemas não poderia, através apenas de evoluções incrementais, cumprir os requisitos associados a uma ferramenta genérica de projeto baseada em otimização. Isso se deve à inerente complexidade desse tipo de sistema, cujas características exigem a combinação de recursos de análise de resposta e sensibilidade, parametrização geométrica, definição de funcionais de performance, geração e atualização automática de malhas. Além disso, existem aspectos ainda não estabelecidos na formulação.

Verifica-se, portanto, um descompasso entre as necessidades da atividade de projeto e as ferramentas disponíveis. Considere, por exemplo, o caso de veículos automotores, nos quais os diversos requisitos de performance se traduzem, muitas vezes, em severas exigências de projeto estrutural. Com isso, uma situação cada vez mais comum tem sido a consideração de características e propriedades não-lineares em condição de trabalho[1]: limitações de peso levam ao projeto de estruturas que operam

[1]Além dessas situações em condição de trabalho, existe todo o campo de planejamento de processos de produção (estampagem, forjamento, fundição, etc.), os quais são fundamentalmente não-lineares.

em regime plástico infinitesimal, modelos de contato unilateral permitem a avaliação mais correta das regiões de fronteira entre componentes, normas de segurança exigem definições precisas das zonas de deformação e absorção de energia durante impactos, parâmetros de conforto envolvem o projeto de componentes de elastômeros para a suspensão de conjuntos motrizes com rigidez e posição controlada, estruturas delgadas e molas sob carregamento de trabalho podem apresentar configuração bastante diversa da geometria inicial, elementos de vedação envolvem elastômeros e contato com atrito.

Tome-se, por exemplo, o projeto dos elementos de suspensão de conjuntos motrizes. O objetivo é a obtenção de coxins[2] com parâmetros de rigidez corretos e, portanto, com a performance dinâmica desejada, o que permite isolar a vibração do conjunto motriz de outros subsistemas do veículo, resultando em maior conforto para os ocupantes do veículo. As características também têm impacto nos níveis de tensão alcançados nos suportes dos conjunto motriz, ou seja, em sua durabilidade. Os valores ideais de rigidez e o posicionamento dos coxins são obtidos a partir de modelos de dinâmica de corpos rígidos. O passo seguinte está em como obter componentes apresentando tais características, as quais dependem da posição de equilíbrio do sistema, o que envolve não-linearidades geométrica, de material e restrição de incompressibilidade. A própria durabilidade da borracha é função do componente volumétrico da tensão, estando relacionado às propriedades de incompressibilidade. Os parâmetros de controle disponíveis de projeto são a composição do elastômero, para a qual há menor margem de alterações, e a *forma* do componente.

Com relação à análise de resposta, a busca de modelos mais precisos de comportamento de material e formulações cinemáticas de incompressibilidade faz com que a simulação numérica de componentes de elastômeros seja uma preocupação atual, apesar de existirem opções de pacotes comerciais com funcionalidades satisfatórias. Recursos computacionais de síntese de componentes, porém, são inéditos neste contexto e, em conseqüência, a dificuldade em se projetar elementos para novas aplicações tem, em vários casos da indústria nacional, levado à adoção de soluções subótimas em termos de performance em durabilidade e conforto (adoção de componentes já existentes), ou custo (importação). A competitividade do produto final pode ser comprometida em quaisquer dos casos.

Obviamente, as dificuldades de competitividade da indústria brasileira estão relacionadas a muito mais fatores que apenas as técnicas atualmente aplicadas no projeto de componentes e sistemas. Con-

[2]Coxins são elementos de ligação de subsistemas e isolamento de vibrações compostos de elastômeros (borracha) e metal (geralmente aço) com função de mola.

tudo, no contexto da engenharia, a solução está no desenvolvimento de soluções inovadoras de projeto e produção. A adoção de recursos de síntese de componentes e sistemas, muito mais que recursos de análise de resposta somente, é fundamental nesse processo. Nesse sentido, uma seqüência de trabalhos foi desenvolvida, iniciando-se com métodos de solução de sistemas de equações lineares de elementos finitos e estimação de erros (Bittencourt, 1996), dirigindo-se em seguida para análise de sensibilidade e otimização de forma de estruturas lineares bidimensionais (Silva, 1997; Silva e Bittencourt, 1997; Silva e Bittencourt, 1998; Silva e Bittencourt, 1999b), tridimensionais (Silva e Bittencourt, 1999a; Silva e Bittencourt, 2000) e otimização topológica (Driemeier, 2002). Uma característica comum desses trabalhos tem sido a ênfase na qualidade dos códigos produzidos através da aplicação do paradigma de orientação por objetos e conceitos de engenharia de software.

Nessa tese, busca-se a generalização de conceitos de análise de sensibilidade e otimização de trabalhos anteriores para aplicações de componentes estruturais apresentando não-linearidades geométrica e de material hiperelástico, com contribuições tanto na formulação quanto nas técnicas de implementação e desenvolvimento de programas. O escopo do trabalho desenvolvido será detalhado no decorrer deste capítulo. As questões colocadas acima serão inicialmente avaliadas em termos de sua perspectiva histórica. Em seguida, as mesmas questões serão formuladas matematicamente, detalhando-se os objetivos a serem atingidos para respondê-las.

1.2 Revisão de Literatura

1.2.1 Técnicas de Otimização de Forma e Análise de Sensibilidade de Estruturas Lineares

A teoria matemática clássica de otimização tem seu início no século XVII juntamente com o desenvolvimento do cálculo diferencial. As condições necessárias de pontos estacionários de uma função diferenciável foram estabelecidos por Euler e a extensão de tais condições para problemas sujeitos a restrições de igualdade foi realizada por Lagrange. Já no século XX, Karush, Kuhn e Tucker estabeleceram as condições de otimalidade para problemas também sujeitos a restrições de desigualdade, obtendo assim o enunciado geral do problema de otimização de funções diferenciáveis (Bazaraa *et al.*, 1993).

Com relação ao projeto de estruturas, Galileu Galilei, ainda no século XVII, discutiu a forma ótima da viga e Lagrange formulou o problema da coluna elástica de peso mínimo. Também podem ser citados os trabalhos de estudiosos como Jacob Bernoulli (1655-1705), Willian Rowan Hamilton (1808-1865) e James Clerk Maxwell (1831-1879) dedicados a formulações iniciais do problema de otimização estrutural (Timoshenko, 1953). No início do século XX, Michell desenvolveu uma teoria de configuração ótima de treliças submetidas a uma única carga e sujeitas apenas a restrições de tensão (Michell, 1904). O ponto comum desses trabalhos é a abordagem analítica de componentes altamente idealizados, dificilmente aplicáveis em situações práticas. Dessa forma, a evidente inexistência de recursos de solução de problemas estruturais genéricos, associada à falta de compreensão clara de como formular o problema de projeto em termos de um problema de otimização, fez com que questões de otimização estrutural viessem a ser consideradas sistematicamente apenas após a metade do século XX. De fato, não é mera coincidência que isso tenha ocorrido simultaneamente à disponibilidade dos primeiros computadores digitais e do Método de Elementos Finitos (MEF). A teoria matemática de otimização, contudo, constitui a base teórica sobre a qual se desenvolveu a fase seguinte da otimização numérica.

(Klein, 1955) reconheceu que os problemas de otimização estrutural podem ser vistos como problemas de programação matemática não-linear. Apesar de tratar casos relativamente simples, a grande inovação desse trabalho está na inclusão de restrições de desigualdade, permitindo formular adequadamente o problema de otimização estrutural. Entretanto, o impacto desse trabalho foi reduzido, pois foi aplicada a técnica de converter as restrições de desigualdade em restrições de igualdade, aumentando a dimensão do problema e dificultando ainda mais a sua solução.

(Schmit, 1960) foi o primeiro a compreender a viabilidade da aplicação sistemática de técnicas de programação matemática – já conhecidas pela comunidade de pesquisa operacional (Lavi e Vogl, 1966) – na solução de problemas gerais de otimização de projetos de estruturas. Dessa forma, esse pode ser reconhecido como o primeiro trabalho em otimização estrutural moderna (Schmit, 1981; Vanderplaats, 1982; Vanderplaats, 1993). A importância desse artigo reside em dois aspectos: o problema de projeto foi reconhecido e formulado como um problema de otimização sujeito a restrições e resolvido numericamente acoplando códigos de elementos finitos e programação matemática e; demonstrou-se com um experimento numérico simples (treliça estaticamente indeterminada de três barras sujeita a vários casos de carregamento) como as características não-lineares de um problema estrutural não permitem sua correta solução com base nos critérios de projeto até então utilizados[3]. Esse resultado não-intuitivo –

[3]Num problema estaticamente indeterminado de uma treliça de três barras, o ótimo não corresponde a nenhum vértice da região viável (nesses vértices cada uma das barras está sob tensão máxima em pelo menos um dos critérios – conceito

e que contrariava a prática corrente – demonstrou a necessidade de se aplicarem métodos sistemáticos de otimização na solução de problemas estruturais.

(Zienkiewicz e Campbell, 1973) apresentaram a primeira formulação numérica do problema de otimização de forma. Nesse trabalho, o problema de projeto foi formulado diretamente em elementos finitos, sendo as coordenadas dos nós do contorno as variáveis de projeto. Aplicou-se o método de diferenciação direta das equações discretas para análise de sensibilidade e o método de programação linear seqüencial para solução do problema de programação matemática.

A formulação contínua da análise de sensibilidade é introduzida no início da década seguinte, através da aplicação do conceito de derivada material da Mecânica do Contínuo para formular expressões gerais de sensibilidade de funcionais de performance em relação à variação do domínio do problema estrutural na forma de integrais sobre seu contorno (Cea, 1981; Zolésio, 1981; Choi e Haug, 1983; Pironneau, 1984). Nesses trabalhos, é estabelecido o conceito de *campo de velocidades de projeto*, que consiste na derivada da forma do domínio do problema estrutural em relação às variáveis de projeto e que desempenha função central na integração do modelo matemático do problema de análise de resposta com os algoritmos de análise de sensibilidade e otimização de forma.

A comparação entre os diversos métodos de análise de sensibilidade[4] foi o tema de vários trabalhos, com maior atenção aos problemas envolvendo variáveis de forma (Yang e Botkin, 1987; Choi e Twu, 1989; Haftka e Adelman, 1989; Zhang e Beckers, 1989). Os resultados mostraram que sob condições de comparação equivalentes (parametrização e campos de velocidades) o método contínuo de domínio é equivalente ao método de diferenciação direta, tanto teórica quanto numericamente.

A principal atração da abordagem contínua da análise de sensibilidade é permitir calcular analiticamente as derivadas de performances estruturais sem envolver nenhuma discretização e não ser necessário especificar um tamanho de passo (Haug *et al.*, 1986), como é o caso do método de diferenças finitas. A generalidade desses resultados é uma vantagem adicional, uma vez que as expressões de sensibilidade obtidas dessa forma podem ser igualmente aplicadas a quaisquer métodos de solução analítica ou numérica tais como elementos finitos (Haug *et al.*, 1986), elementos de contorno (Zhao,

de projeto totalmente tensionado), soluções normalmente adotadas pelos critérios de projeto ótimo utilizados na época.

[4]Uma outra abordagem para diferenciação da resposta de estruturas consiste em diferenciar diretamente o *programa computacional* usado na análise de resposta (Masmoudi *et al.*, 1993). Entretanto, essa abordagem de diferenciação automática requer disponibilidade substancial de memória, podendo-se ser proibitiva em problemas de grandes dimensões.

1991), formulação p de elementos finitos (Hwang *et al.*, 1997) ou *meshfree*[5] (Kim, 1999). No caso de problemas reversíveis, essa independência se estende até a etapa de implementação, pois as expressões podem ser avaliadas a partir apenas de informações de pós-processamento. Nesse caso, ferramentas comerciais podem ser aplicadas na análise de resposta (o que não pode ser feito em problemas irreversíveis, como será discutido na seção seguinte). Por outro lado, a abordagem contínua requer certo grau de esforço analítico para desenvolver as expressões, e portanto maior conhecimento teórico da formulação da análise de resposta. Apesar de originar expressões aplicáveis apenas ao método de análise de resposta empregado, o método discreto tem a vantagem de ser mais simples, sendo defendido por alguns autores (Lund, 1994; Yatheendhar e Belegundu, 1993).

Durante a primeira metade da década de 1980, tem-se uma caracterização dos aspectos multidisciplinares da otimização de forma, observando-se que a construção de ferramentas úteis envolveriam a combinação eficiente e coerente de diversas capacidades tais como descrição geométrica, parametrização e geração automática de malhas, além de algoritmos de programação matemática adequados às necessidades da otimização estrutural e a maturação das técnicas de análise de sensibilidade de projetos.

Nessa época, já se reconhece que a otimização de forma apresenta dificuldades que não estão presentes ou são mais simples na otimização de propriedades discretas, tais como a atualização da malha devido à modificação da geometria e o maior custo e complexidade para se obter bons valores de sensibilidade (Haftka e Grandhi, 1986). A otimização de forma exige um processo automático de geração de malhas uma vez que a geometria do domínio é a própria incógnita a ser determinada. A cada alteração do domínio, duas abordagens de atualização podem ser aplicadas: na primeira, trabalha-se com um modelo geométrico e a cada modificação deste, uma nova malha é gerada (possivelmente com análise adaptável); na segunda, são utilizadas regras para deformar a malha inicial de acordo como a modificação do contorno. A deformação da malha pode ser feita com equações diferenciais de equilíbrio (Cea, 1981) ou relações paramétricas[6] (Wang *et al.*, 1985). Em qualquer dos casos, a correspodência

[5] *Métodos sem malha.* Tais métodos foram inicialmente denominados *meshless*, como mostra a literatura. Porém a denominação correta seria *meshfree* (adotada posteriormente), pois sua característica principal não seria a inexistência da malha (que continua sendo definida para execução de integração numérica), mas sim a dissociação entre domínio de integração e suporte de funções de forma, coincidentes no conceito de elementos finitos. Em resumo, em métodos *meshfree*, as funções de forma são independentes da malha.

[6] Uma técnica de parametrização da geometria alternativa à parametrização do contorno é baseada na utilização de *elementos de projeto*, que consiste na definição de uma malha auxiliar menos refinada adjacente ao contorno composta de quadriláteros em domínios bidimensionais e hexaedros nos tridimensionais. Definindo uma relação paramétrica entre a malha auxiliar e a malha de análise, é possível modificar a geometria alterando-se pontos de controle da malha auxiliar.

um-a-um entre modelo de elementos finitos e variáveis de projeto é abandonada em favor da utilização de splines para representação geométrica, as quais são polinômios de baixa ordem (evitam oscilação) combinados para maximizar a suavidade do contorno. Além disso, com B-splines (Rogers e Adams, 1990), a regularidade do domínio já é automaticamente considerada. (Yang e Choi, 1985) demonstraram a superioridade dessa representação para cálculo de sensibilidade.

(Bennett e Botkin, 1985) apresentaram uma abordagem integrada de técnicas com o objetivo de se obter uma ferramenta automatizada de projeto baseada em otimização, apontando inclusive a necessidade da adoção de recursos de geração automática de malhas e análise adaptável. Aplicaram um estilo de parametrização bastante adequado, baseado em descrição geométrica do contorno em termos de funções de forma e pontos de controle. Essa técnica, que veio a se tornar dominante, é independente do gerador de malhas, simplifica a integração de ferramentas de CAD e a parametrização, independentemente da dimensão do domínio. A preocupação com uma abordagem integrada também transparece na avaliação do custo global da utilização de análise adaptável na otimização. Propõe-se ainda a extrapolação linear dos valores de tensão máxima para evitar grande número de análises. O grande número de iterações exigido para convergência nos problemas de otimização apresentados e o comportamento oscilatório do funcional de tensão são conseqüências da não utilização de informações de campos de velocidade para atualização da malha, segundo os próprios autores. A aplicação da análise adaptável nos ciclo de otimização também foi investigada por (Botkin e Bennett, 1986; Kikuchi *et al.*, 1986).

(Ding, 1986) retomou as questões envolvidas na obtenção de uma ferramenta automatizada de projeto de componentes estruturais baseada em otimização de forma, analisando de forma integrada desde a formulação do problema, representação do contorno, geração automática de malhas e análise adaptável, até análise de sensibilidade e algoritmos de programação matemática. Algumas características importantes foram bem identificadas: a superioridade da representação em B-splines, a problemática da atualização de malhas, o custo da análise adaptável num ciclo iterativo e a importância da análise de sensibilidade no contexto da otimização de forma. Entretanto, um dos maiores empecilhos para incorporar a geração automática de malhas em algoritmos de otimização ainda está no fato de que em grande parte dos casos a geração de malhas é basicamente manual. Um motivo seria a dependência na experiência do analista para julgar a qualidade da malha. Uma resposta para isso seria a combinação

O maior problema dessa técnica, porém, é a dificuldade em gerar a malha auxiliar em domínios complexos, especialmente em volumes.

de análise adaptável e geração automática. Essa técnica, porém, não pode ser utilizada em modelos que não são componentes individuais, tem custo proibitivo em problemas não-lineares e a seqüência de iterações de otimização pode se tornar instável quando a malha é regenerada para todo novo projeto devido a não-lineridade envolvida na regeneração de malha, a qual não é considerada nas equações de sensibilidade, as quais se baseiam somente na informação contida nos campos de velocidade.

No campo da programação matemática, (Belegundu e Arora, 1985a; Belegundu e Arora, 1985b) realizaram um estudo extenso dos algoritmos com o foco em otimização estrutural. Uma das principais conclusões é que não é possível correlacionar a performance de um algoritmo num problema analítico de programação matemática com seu comportamento em problemas de projeto ótimo, pois problemas de programação matemática tem baixíssimo custo de avaliação de funcionais e gradientes. Assim, em problemas de otimização estrutural, é necessário convergir rápido para um ótimo aproximado realizando o menor número possível de avaliações de funcionais de performance e seus gradientes (Vanderplaats, 1982). Também enfatizou-se a convergência global[7] como característica necessária para robustez dos algoritmos. O desenvolvimento de algoritmos com convergência global e taxa superlinear foi o tema nos trabalhos de (Lim e Arora, 1986), nos quais se aplicou programação quadrática recursiva e estratégia de conjunto ativo, e (Herskovits, 1986), com método de pontos interiores e busca linear imprecisa. Para aplicações em otimização estrutural (Evsukoff, 1992; Herskovits e Coelho, 1989; Herskovits e Santos, 1997). Também pode-se citar como um desenvolvimento importante nesse campo, a disponibilidade pública de sistemas de propósito geral combinando opções de algoritmos de programação matemática implementados para problemas de alto custo de avaliação (Vanderplaats e Sugimoto, 1986; Fleury, 1989), o que simplificou o desenvolvimento de ferramentas de otimização estrutural (Liefooghe e Fleury, 1989).

A disponibilidade de algoritmos eficientes de programação matemática e pacotes comerciais de análise de resposta fez com que análise de sensibilidade representasse a tendência dominante do período (Levy e Lev, 1987). A formulação contínua da análise de sensibilidade, tal como inicialmente estabelecida, exigia a avaliação de valores precisos de tensão no contorno, que são em geral difíceis de se obter no MEF. (Choi e Seong, 1986) propuseram então procedimento de reescrever as integrais de contorno como integrais de domínio, tirando-se vantagem da característica de maior precisão do MEF no interior do domínio. As desvantagens do método contínuo de domínio são a necessidade de definir campos de

[7]Convergência global corresponde à capacidade de convergir para *algum* ótimo local iniciando-se de qualquer ponto do domínio viável. Não deve ser confundido com convergência para ótimo global.

velocidades no interior do domínio (cuja definição não é unívoca) e o maior custo. Por outro lado, tem a vantagem de fornecer um procedimento geral e simples para a análise de sensibilidade de estruturas formadas pela união de tipos diferentes de elementos finitos (*built up structures*). Quando se utiliza um método contínuo de contorno nesses casos, é necessário considerar condições de interface entre elementos de tipos diferentes. Com o método de domínio, a sensibilidade da estrutura é obtida através da soma das contribuições individuais das integrais de cada subdomínio elementar, sem a necessidade de contabilizar condições de interface. Essa formulação foi extendida por (Twu e Choi, 1992) para variáveis de configuração. A necessidade de definir um campo de velocidades no domínio indica que este procedimento é relacionado com o método discreto, o qual também requer o conhecimento sobre a variação no domínio para a diferenciação da matriz de rigidez.

Para amenizar a demanda computacional do método de domínio, (Seong e Choi, 1987) introduziram o *método de camada de contorno* no qual o campo de velocidades é não nulo apenas numa camada do domínio adjacente ao contorno e nulo no restante do domínio. Com isso, as expressões de sensibilidade somente precisam ser avaliadas na camada. É interessante observar que o método de camada de contorno tal como proposto tem grande semelhança com a técnica de elementos de projeto[8], apresentando inclusive as mesmas deficiências: a distorção dos elementos é praticamente inevitável ao se utilizar o campo de velocidades para atualizar a malha; e a dificuldade prática em se definir a camada de contorno em geometrias complexas e tridimensionais utilizando elementos quadrangulares e hexaédricos respectivamente. Um procedimento automatizado de determinação de camada de contorno juntamente com geração automática de malhas de triângulos e tetraedros foi aplicada posteriormente por (Botkin, 1992; Fancello, 1993; Silva, 1997; Silva e Bittencourt, 1997; Silva e Bittencourt, 1998; Silva e Bittencourt, 1999b; Silva e Bittencourt, 1999a; Silva e Bittencourt, 2000).

(Yao e Choi, 1989) retomaram a discussão acerca de campos de velocidades para o método de domínio da análise de sensibilidade, apresentando aspectos computacionais dos problemas auxiliares decorrentes do algoritmo baseado em equação elíptica (Cea, 1981). Esse método é denominado de *deslocamentos fictícios do contorno.* Esse campo, devido à sua continuidade, fornece malhas de alta qualidade quando aplicado na atualização da discretização. (Rajan e Belegundu, 1989) apresentaram o conceito de parametrização associada a carregamentos fictícios, ou seja, o que modifica o projeto é a deformação fictícia causada por esse carregamento. É interessante observar que o papel do campo

[8]Os autores usam o termo 'elemento de velocidade' ao invés de 'elemento de projeto'.

de velocidades determinado desta forma é bastante análogo ao exercido pelo método de deslocamentos fictícios do contorno. Por outro lado, representa um retrocesso em termos da parametrização geométrica, pois utiliza propriedades do modelo de elementos finitos como variáveis de forma.

No trabalho de (Zhang e Beckers, 1989), demonstrou-se que os métodos discreto e contínuo de análise de sensibilidade são analiticamente equivalentes em elementos finitos apesar das diferenças do ponto de vista de implementação. Portanto, segundo os autores, diferenças observadas nos resultados seriam provenientes dos diferentes algoritmos de extensão de campos de velocidade para o interior do domínio. Com esse objetivo, comparou os métodos de elementos de projeto, camada de contorno e deslocamentos fictícios do contorno do ponto de vista da taxa de convergência de problemas de otimização de forma e de implementação. Os autores concluíram que o método de deslocamentos fictícios do contorno é o mais flexível e prático, pois não requer a definição de elementos de projeto, podendo-se usar qualquer método de geração de malhas. Essa conclusão, porém, está mais associada aos recursos disponíveis do que às características dos campos pesquisados, pois flexibilidade e praticidade também podem ser obtidas com implementações corretas de outros campos de velocidades.

Com a relação a problemas estruturais lineares, pode-se afirmar que, ao final da década de 1980, as principais questões relacionadas a formulações de análise de sensibilidade à forma se encontravam razoavelmente discutidas. A disponibilidade e o aumento de capacidade gráfica e de processamento nos computadores da década seguinte permitiu então que fossem retomados os trabalhos de desenvolvimento de ferramentas automatizadas integrando-se sistemas de CAD e geradores de malha comerciais com recursos de análise adaptável, cujo custo em termos de tempo para problemas lineares passa a ser aceitável, principalmente se comparado com meados da década de 1980. Essa integração envolveu a proposição de novos esquemas de determinação de campos de velocidades no domínio, algumas vezes violando requisitos teóricos decorrentes das expressões de análise de sensibilidade.

Nesse sentido, (Choi e Chang, 1994) apresentaram uma revisão das técnicas de geração de campos de velocidades e atualização de malhas desenvolvidas até então usando como critérios de julgamento requisitos teóricos e práticos decorrentes das expressões de sensibilidade e de técnicas de otimização estrutural. Um aspecto importante desse trabalho é mostrar que apesar do refinamento da malha diminuir a sensibilidade do resultado em relação ao tipo de campo de velocidades usado, os cálculos da análise de sensibilidade não podem ser dissociados da malha em que foram efetuados. Ou seja, a sensibilidade é calculada para um determinado movimento dos nós, descrito pelo campo de velocidades utilizado, em torno da posição discreta corrente. Se a nova malha não for obtida utilizando o campo de velocidades usado na análise de sensibilidade, o valor da sensibilidade são será rigorosamente 'correto', pois não haverá relação entre os dois estados. Além disso, observou-se que a utilização de campos de velocidades que não mantenham linearidade com as variáveis de projeto introduz perturbações locais nos funcionais de performance não previstas pelas expressões de sensibilidade. Logo, à dependência não-linear da resposta estrutural em relação às variáveis de projeto, soma-se outra contribuição devido ao movimento dos nós. Mesmo quando o tipo de avaliação de gradientes considera tal não-linearidade, como é o caso de técnicas que utilizam a sensibilidade total baseada em diferenças finitas (*e.g.* método semi-analítico), a combinação de não-linearidades pode ser muito grande e a escolha dos tamanhos de passo passa a ser problemática. Como, do ponto de vista do algoritmo de minimização, a precisão da sensibilidade está associada à qualidade da previsão fornecida por essa informação de primeira ordem, a taxa de convergência da otimização geralmente é comprometida em quaisquer dessas situações. A recomendação dos autores foi combinar informações de CAD sobre o contorno geométrico (para determinação de campos de velocidades sobre o contorno), variáveis que forneçam relação linear com o campo de velocidades sobre o contorno e extensão das velocidades para o interior do domínio pelo método de deslocamentos fictícios do contorno. Além disso, as características das expressões de sensibilidade indicam que o campo de velocidades deve ser utilizado na atualização da malha de elementos finitos.

Propôs-se a inclusão de análise adaptável no ciclo de otimização nos trabalhos de (Bugeda e Oliver, 1993; Canales *et al.*, 1993; Dufeu *et al.*, 1997; Fancello, 1993; Özakça *et al.*, 1993; Younsi *et al.*, 1996). As técnicas de campos de velocidades apresentadas são variadas: suavização de Laplace (Bugeda e Oliver, 1993; Dufeu *et al.*, 1997), que resulta numa relação não-linear entre contorno e posição dos nós; elementos de projeto (Canales *et al.*, 1993) e camada de contorno de espessura unitária (Fancello, 1993; Özakça *et al.*, 1993). Em todos os casos, a malha é reconstruída através de análise adaptável a cada novo projeto, independentemente do campo de velocidades, o que pode reduzir a taxa de convergência do algoritmo de minimização. Por outro lado, esses trabalhos manifestam a preocupação legítima com a possível degradação da qualidade da discretização devido a alterações sucessivas da geometria. Numa abordagem mais estrita segundo (Choi e Chang, 1994), recomenda-se interromper a otimização assim que esse problema for identificado, reconstruir a malha e iniciar um novo ciclo.

Ao incluir refinamento automático no ciclo de otimização, a maioria dos trabalhos considera apenas estimadores de erro do campo de deslocamentos. Entretanto, observa-se menor taxa de convergência da informação de sensibilidade se comparada com a convergência dos campos de resposta primários do problema de elementos finitos, tanto com refinamento uniforme da malha de elementos (Haftka e Barthelemy, 1989) ou com análise adaptável dirigida por estimador baseado apenas no campo primário (Fuenmayor *et al.*, 1997). No caso da análise adaptável, são duas as causas: os carregamentos das equações de equilíbrio e de sensibilidade são diferentes (de modo que uma malha adaptada para um caso pode não ser a mais conveniente para outro), e as expressões de sensibilidade de funcionais de performance são escritas em termos de derivações dos campos primários de resposta (que são apenas indiretamente controladas pelos estimadores usuais). Por isso, com o objetivo de controlar com precisão a convergência da informação de sensibilidade, um estimador de erro *a posteriori* da sensibilidade devido a discretização em elementos finitos foi proposto por (Buscaglia *et al.*, 1995) para problemas lineares de Poisson e elasticidade linear, sendo posteriormente extendido pelos mesmos autores para o problema de contato sem atrito de corpos elásticos (Buscaglia *et al.*, 1998). As formulações consideram indicador de erro da sensibilidade do campo primário de resposta (resultado da equação de sensibilidade) e dos funcionais de performance.

Outros desenvolvimentos têm ocorrido no sentido de disponibilizar de modo efetivo os recursos de análise de sensibilidade e otimização na atividade de projeto, extendendo características de ambientes comerciais de CAD na sua integração com programas de elementos finitos. São levantadas principalmente questões de parametrização geométrica e geração de campos de velocidades devido à complexidade de estrutura de dados dos programas de modelagem geométrica atuais, nos quais utiliza-se combinação de representações implícita (fórmulas analíticas definindo exatamente a geometria tais como cônicas e sólidos) ou paramétrica (*e.g.* NURBS e B-splines) com geometria variacional (relações de tangência, paralelismo, simetria, etc. definidas por sistema de equações não-lineares).

No trabalho de (Lindby e Santos, 1997), o objetivo da integração de programas de CAD e CAE[9] é utilizar uma parametrização próxima das características geralmente utilizadas no ambiente de projeto. Para permitir a parametrização de qualquer característica de uma representação geométrica genérica, propuseram determinar o campo de velocidades no contorno perturbando a geometria dentro do ambiente de CAD e calculando as variações das posições dos nós sobre o contorno. Esse campo de velocidades no contorno é então transferido para o interior do domínio usando técnicas de suavização de Laplace. Os autores reconhecem a qualidade superior do método de deslocamentos fictícios do contorno, o qual

[9] I/EMS CAD e I/FEM da Intergraph.

é utilizado como medida de comparação, porém não são discutidos os efeitos não-lineares da geometria variacional e da suavização de Laplace na relação entre variáveis de projeto e campo de velocidades, que deve ser linear. Tem-se ainda o problema de seleção do tamanho da perturbação a ser imposta à geometria. De modo similar, (Chen e Tortorelli, 1997) discutiram a geração de campos de velocidades na integração de recursos de parametrização e análise de sensibilidade em sistemas de CAD e FEM já estabelecidos[10]. Neste trabalho, porém, são desenvolvidas formulações analíticas dos campos de velocidades no contorno para os vários tipos de representação geométrica. Logo, as características parametrizáveis são restritas, mas os campos de velocidade no contorno são obtidos precisamente. Os campos no interior do domínio são gerados por suavização de Laplace, técnica também usada para atualizar a geometria para pequenas alterações do projeto (o tamanho de alterações "pequenas" é definido *a priori* em até 5% da dimensão do vetor de variáveis de projeto), sendo a malha regenerada completamente para grandes alterações.

A partir das referências, pode-se afirmar que a análise de sensibilidade à forma de problemas com equação de equilíbrio linear teve suas principais questões resolvidas antes do final da década de 1990. Vantagens e desvantagens relativas, bem como situações de equivalência, já eram conhecidas. Apesar da grande maioria dos trabalhos estar associada ao MEF, resultados satisfatórios também foram obtidos com outros métodos numéricos. Quanto à programação matemática, nota-se certo predomínio no uso de métodos de programação seqüêncial, linear ou quadrática, o que se deve à maior facilidade de incorporar técnicas de restrições ativas e aproximação nesses casos. Porém existe o inconveniente de permitirem a violação de restrições no decorrer do processo de otimização. Métodos de pontos interiores eficientes são atualmente disponíveis e têm se mostrado mais adequados em problemas práticos de otimização em engenharia. Por fim, tem-se as questões associadas a parametrização, campos de velocidades e atualização de malhas. É consenso que a parametrização da forma deve ser feita no modelo geométrico. Por outro lado, isso induz uma dependência em relação a sistemas de modelagem geométrica, em geral fechados e com estruturas de representação complexas, o que dificulta o cumprimento da exigência teórica de linearidade entre variáveis e campos de velocidades. Em muitos casos, os campos de velocidades são estendidos para o interior do domínio sem também respeitar a exigência de lineridade. Sobre a atualização, observa-se que, com a disponibilidade de melhores recursos computacionais, há certa tendência em se redefinir a discretização a cada novo projeto, eventualmente com

[10] Aqui usa-se Pro-Engineer, que possui API's de programação para a inclusão de funcionalidades pelo usuário, e Nastran, que incorpora recursos de análise de sensibilidade pelo método semi-analítico.

a aplicação de análise adaptável, apesar das implicações numéricas em termos de oscilação e queda na taxa de convergência da otimização.

1.2.2 Análise de Sensibilidade e Otimização de Estruturas Não-Lineares

De acordo com as referências (Arora e Haug, 1979; Haftka e Grandhi, 1986; Haftka e Adelman, 1989), observa-se que o desenvolvimento da base da formulação de análise de sensibilidade de projeto em problemas lineares ocorreu desde a segunda metade da década de 70 até o início da segunda metade da década de 80. Nos períodos seguintes, a pesquisa em análise de sensibilidade de problemas lineares esteve voltada para a solução de questões acessórias à formulação básica, aspectos computacionais e de implementação.

Na primeira metade da década de 80 houve rápido desenvolvimento de teoria e ferramentas de solução de problemas não-lineares, vide referências citadas por (Bathe, 1996; Simo e Hughes, 1998). Paralelamente, tem-se também uma atenção crescente na utilização de materiais não-lineares como polímeros e borrachas, além do projeto de estruturas para resistência em condições extremas de utilização tais como segurança veicular, estruturas metálicas delgadas e projeto para vida finita. Buscou-se, portanto, aplicar conceitos de projeto ótimo em problemas estruturais não-lineares.

Em sua revisão dos desenvolvimentos em análise de sensibilidade de estruturas, (Haftka e Adelman, 1989) observaram que o foco do interesse se concentrava cada vez mais no regime não-linear, buscando-se tratar problemas de resposta transiente ou de resposta estática envolvendo não-linearidades geométricas, de material e de condições de contorno, com ênfase na aplicação de formulações variacionais (formulação contínua da análise de sensibilidade). Devem-se citar os trabalhos pioneiros de (Ryu *et al.*, 1985; Choi e Santos, 1987; Santos e Choi, 1988) na generalização dos métodos direto e adjunto de análise de sensibilidade para o contexto desses novos problemas.

A partir de uma formulação genérica de problemas estruturais não-lineares, (Ryu *et al.*, 1985) chegaram a conclusões importantes acerca da análise de sensibilidade: a equação de sensibilidade tanto no método direto quanto no adjunto é sempre linear, mesmo que a análise de resposta seja não-linear; a matriz da equação de sensibilidade coincide com a matriz de rigidez tangente do último estado de equilíbrio; por outro lado, o carregamento da equação de sensibilidade não pode ser deduzido numa forma genérica uma vez que depende de cada modelo ou tipo de não-linearidade, sendo necessário desenvolver expressões para cada não-linearidade; os métodos Lagrangeano total e Lagrangeano atualizado podem ser indistintamente aplicados na análise de resposta sem afetar os resultados de análise de sensibilidade. Contudo, os autores se precipitaram ao afirmar ser possível executar a análise de sensibilidade através de pós-processamento da saída de pacotes comerciais fechados de análise de resposta em todos os casos de não-linearidades estruturais (os exemplos apresentados não incluíram irreversibilidades.). De fato, estudos subseqüentes mostraram que os resultados obtidos através dessa abordagem para não-linearidades irreversíveis não são satisfatórios.

(Choi e Santos, 1987; Santos e Choi, 1988), por sua vez, apresentaram uma formulação mais detalhada e menos geral, tratando apenas problemas não- lineares sem restrições internas e reversíveis não relacionados à forma. (Santos e Choi, 1992) consideraram a extensão dessa formulação para incluir a variação da forma do domínio. Nos três trabalhos, os exemplos discutidos envolveram apenas não-linearidade geométrica.

Utilizando o princípio da Enegia Mútua de Hu-Washizu, (Phelan *et al.*, 1989) desenvolveram expressões de sensibilidade para hiperelasticidade. Apesar da formulação ser genérica o suficiente para tratar formulações mistas, os autores analisaram somente problemas de único campo, utilizando elementos quadrangulares de oito nós e integração reduzida 2×2 para descrever domínios de material de Mooney-Rivlin incompressível.

Problemas dependentes do histórico do carregamento (problemas irreversíveis) foram o tema de (Tsay e Arora, 1989; Tsay e Arora, 1990; Tsay *et al.*, 1990). A conclusão mais importante desses trabalhos foi a de que a análise de sensibilidade de problemas dependentes do caminho também é dependente do caminho. Ou seja, diferentemente dos problemas reversíveis, nos quais a análise de sensibilidade é feita apenas no último estado de equilíbrio da análise de resposta, a presença de irreversibilidades faz com que a equação de sensibilidade precise ser resolvida em cada estado intermediário de equilíbrio, atingido ao final de cada passo de carregamento, tanto no método direto quanto no adjunto. Apesar disso, o processo continua sendo linear: a equação de equilíbrio é sempre linear e a sensibilidade total consiste na soma das sensibilidades intermediárias. Dessa característica da formulação de dependência do caminho do carregamento decorrem duas conseqüências práticas: a análise de sensibilidade não pode ser tratada como pós-processamento da análise de resposta, devendo ser resolvida simultaneamente, o que impede a aplicação de pacotes comerciais de análise de resposta; e a escolha entre método direto ou adjunto passa a não depender apenas das características do problema de projeto (número de variáveis e funcionais de performance estrutural), uma vez que o método direto passa a ser claramente superior nessa classe de problemas. Os autores também reportam que o custo computacional para a solução de problemas de otimização envolvendo plasticidade é significativamente alto. Portanto, a solução de problemas práticos está no escopo de computação de alta performance.

(Tortorelli, 1992) apresentou formulação de análise de sensibilidade para problemas elastostáticos com restrições aplicando o princípio de variacional de sensibilidade, vide (Arora e Cardoso, 1992). Essa abordagem, porém, está restrita a problemas com operador de rigidez simétrico. É discutido exemplo tridimensional de material hiperelástico com restrição de incompressibilidade utilizando formulação mista de elementos finitos, com hexaedros lineares e integração reduzida. É importante observar que esse é um dos primeiros trabalhos nos quais não se utiliza pacote comercial para análise de resposta, usado apenas para verificação de resultados.

(Cardoso e Arora, 1992) extenderam a formulação variacional de sensibilidade introduzida por (Arora e Cardoso, 1992) para problemas dinâmicos transientes, obtendo-se formulação adjunta. Por se tratar de um problema não-linear, apresenta as mesmas características já apontadas anteriormente, inclusive de que a sensibilidade é dependente do caminho e que a formulação adjunta não é a mais conveniente para esse contexto, uma vez que é um problema de valor terminal que somente pode ser definido ao final da análise de resposta, mas também necessita das informações de todos os estados de equilíbrio intermediários.

(Fancello, 1993) discutiu a sensibilidade em problemas elastostáticos de contato sem atrito. A questão da não-diferenciabilidade da equação de estado é abordada de forma rigorosa, sendo a desigualdade variacional resolvida por penalização e método quasi-Newton. Dessa forma, as matrizes de rigidez tangente não são exatas, ao contrário do exigido pela equação de sensibilidade. Além disso, a transformação entre domínio e campo de deslocamentos é contínua mas não-diferenciável. Em conseqüência, funcionais cuja sensibilidade envolve a derivada do campo de deslocamentos também não são, a rigor, diferenciáveis. Essas duas questões foram tratadas pelo autor através da escolha de funcionais tais que sua sensibilidade não envolvesse a derivada do campo de deslocamentos. Portanto, optou-se por uma formulação rigorosa mas menos geral. Essa formulação é revisitada e extendida em (Fancello e Feijóo, 1994; Fancello *et al.*, 1995).

(Vidal e Haber, 1993) foram os primeiros a desenvolver uma formulação de sensibilidade de método implícito de integração de modelos constitutivos algorítmicos, obtendo uma formulação de sensibilidade para problemas de elastoplasticidade com encruamento e mostrando, através de resultados numéricos, que a precisão dos resultados de sensibilidade depende da utilização de operador elastoplástico tangente consistente na análise de resposta. Se o operador é inconsistente, os resultados são imprecisos e pioram com o aumento do tamanho do passo de carregamento. (Kleiber, 1993) tratou do mesmo problema apenas do ponto de vista da formulação, sem apresentar exemplos numéricos, mas também concluiu ser necessário aplicar operador elastoplástico tangente consistente. Segundo o autor, somente assim é mantida a linearidade da equação de sensibilidade.

A análise de sensibilidade em plasticidade foi considerada nos seguintes trabalhos. (Lee e Arora, 1995) investigaram a ocorrência de descontinuidade de coeficientes de expressões de sensibilidade na transição entre estado elástico e plástico. (Kleiber e Kowalczyk, 1996) desenvolveram expressões para problemas de tensão plana. (Park e Choi, 1996b; Park e Choi, 1996a) consideraram elementos estruturais de placas e vigas sujeitos a grandes deslocamentos e deformações infinitesimais, com variáveis de dimensões discretas e otimização de forma. Os mesmos elementos estruturais e tipos de variáveis foram posteriormente aplicados à dinâmica transiente por (Cho e Choi, 2000a; Cho e Choi, 2000b).

Observa-se também que o problema de hiperlasticidade não-linear tem sido seguidamente revisitado. Isso ocorre porque a formulação da análise de resposta ainda apresenta desafios relacionados à definição de equações constitutivas de material mais precisas (principalmente de elastômeros), aplicação da restrição de incompressibilidade e estabilidade numérica, fazendo com que sejam desenvolvidas técnicas alternativas em elementos finitos (Chen e Pan, 1996; Chen *et al.*, 1996; Chen *et al.*, 1997a) à formulação mista clássica (Sussman e Bathe, 1987), ou novos métodos de solução tal como o método *meshfree* desenvolvido por (Chen *et al.*, 1997a) e revisto por (Chen *et al.*, 2000). Além disso, a hiperelasticidade incompressível ou quasi-incompressível (incluindo não-linearidades geométricas) constitui a base da formulação da parcela elástica de problemas de elastoplasticidade finita (Simo e Hughes, 1998) e, portanto, da análise de problemas práticos de impacto e processos de fabricação baseados em conformação plástica tais como forjamento, laminação e estampagem.

(Grindeanu *et al.*, 1998) aplicaram código próprio baseado em método *meshfree* na análise de sensibilidade e otimização de componentes de elastômeros, modelados com hiperelasticidade não-linear quasi-incompressível. Posteriormente, (Kim *et al.*, 2001b) extenderam esse trabalho, incluindo contato com atrito. A vantagem do método de análise de resposta aplicado nesses trabalhos é a obtenção da mesma precisão com um menor número de graus de liberdade se comparado ao método de elementos finitos h com funções de baixa ordem de interpolação quando aplicados a situações de severa distorção do domínio. Isso se deve à utilização de funções de alta ordem e ao desacoplamento entre domínio de integração e domínio de suporte das funções de forma. Essa última característica torna tais métodos menos sensíveis à ocorrência de distorção geométrica da discretização, uma vez que as funções de interpolação são definidas sobre o domínio real do problema e não sobre um subdomínio de referência, o que elimina uma transformação de domínio. Por outro lado, essa característica não permite conhecer *a priori* a conectividade dos graus de liberdade do problema, impossibilitando a utilização de algoritmos eficientes de construção e armazenamento de matrizes, tão pouco de solução de sistemas, ao contrário do que ocorre no MEF. Em aplicações de otimização de forma, a vantagem dos métodos *meshfree* está no procedimento trivial necessário para atualizar a discretização pois não ocorre distorção da interpolação[11]: o processo consiste apenas em reposicionar os suportes funcionais sem alterar sua geometria. A desvantagem está no procedimento mais complicado de discretização das equações de sensibilidade. Deve-se ainda ressaltar que, dentro do mesmo grupo de pesquisa, o MEF continua sendo aplicado a problemas reversíveis (ver (Choi e Duan, 2000)).

Também no campo da hiperelasticidade, um tipo diferenciado de aplicação de otimização de forma e do MEF foi apresentado por (Dias *et al.*, 1998) e (Herskovits *et al.*, 1998): ao invés de tratar a solução da equação de equilíbrio como um subproblema do ciclo de otimização, essa equação é modelada como uma restrição de igualdade do problema de otimização, possibilidade já reconhecida por (Vanderplaats, 1982). Com isso, os problemas não-lineares de equilíbrio e otimização são resolvidos simultaneamente, o que representa uma redução significativa no custo total necessário para atingir o projeto ótimo. É importante mencionar, porém, que essa estratégia impede a atualização da discretização sem interromper o processo, mesmo que ocorra distorção durante a seqüência de alterações da geometria. Além disso, há pouca flexibilidade na aplicação de ações externas (*e.g.* carregamentos aplicados em passos de carregamento distintos) e não se pode ter problema de equilíbrio dependente do histórico de carregamentos (irreversíveis).

Utilizando método *meshfree*, (Kim, 1999) a partir de formulação de análise de resposta e sensibilidade de não-linearidades geométrica e de hiperelasticidade, incorporou os desenvolvimentos em contato sem atrito, plasticidade infinitesimal e dinâmica transiente já aplicados em elementos finitos, extendendo-os para plasticidade finita e contato com atrito. Uma das aplicações de seus resultados é a

[11] A menos da possibilidade de que algum domínio de integração perca sua consistência geométrica.

otimização de processos de estampagem (Kim *et al.*, 2001a). De fato, uma das motivações para busca de formulações de análise de sensibilidade e otimização de problemas estruturais não-lineares tem sido justamente a aplicação de métodos sistemáticos de projeto a processos de fabricação, vide (Fourment e Chenot, 1996a; Fourment e Chenot, 1996b; Maniatty e Chen, 1996; Zhao *et al.*, 1997; Balagangadhar e Tortorelli, 2000a; Balagangadhar e Tortorelli, 2000b) para exemplos em conformação plástica e (Smith *et al.*, 1998b; Smith *et al.*, 1998c; Smith *et al.*, 1998a) para injeção de plástico.

Podem ser citados também trabalhos em análise de sensibilidade de segunda ordem (Taroco *et al.*, 1998) e Mecânica da Fratura (Taroco, 2000).

Por fim, deve-se mencionar que o alto custo de solução de problemas não-lineares impede a aplicação de análise adaptável com estimadores de erro *a posteriori* tal como o apresentado por (Gallimard *et al.*, 1996) dentro de ciclos de otimização. Não se tem observado trabalhos nessa direção.

1.2.3 Orientação por Objetos em Aplicações do Método de Elementos Finitos

Apesar da técnica de orientação por objetos datar do final da década de 1970, sua aplicação em métodos numéricos para engenharia, especialmente em técnicas de elementos finitos, ocorreu com ênfase apenas no início da década de 1990, vide (Bittencourt, 1990; Fancello *et al.*, 1991; Feijóo *et al.*, 1991; Fenves, 1990; Filho e Devloo, 1991; Forde *et al.*, 1990; Lee e Arora, 1991; Mackie, 1992; Miller, 1991; Scholz, 1992; Zimmermann *et al.*, 1992). Nesse primeiro momento, a preocupação principal foi a investigação inicial das possibilidades oferecidas pela programação orientada por objetos (POO) no contexto da computação numérica. Houve trabalhos no campo de geração automática de malhas (Fancello *et al.*, 1991) e algoritmos de reordenamento de nós e elementos (Fenves, 1990), mas a quase totalidade dos trabalhos teve como tema central a definição do núcleo de solução do MEF, ou seja, os algoritmos de construção das matrizes e vetores locais dos elementos e a montagem do sistema global. Foram identificadas as primeiras abstrações necessárias ao contexto de problemas lineares e as técnicas de construção de algoritmos básicos utilizando características da POO tais como encapsulamento, herança, polimorfismo (Cox, 1986; Martin e Odell, 1998). Observou-se, por exemplo, a possibilidade de generalizar o núcleo de solução através de uma hierarquia de tipos de elementos finitos (Mackie, 1992).

Nessa fase, não havia consenso quanto a linguagens de programação, tendo sido usadas Smalltalk (Zimmermann *et al.*, 1992), Objective C (Bittencourt, 1990), Turbo Pascal (Mackie, 1992) e C++

(Fancello *et al.*, 1991). A falta de prática nesse estilo de programação e o uso de linguagens de baixa eficiência numérica, tal como o Smalltalk, foi a causa da insatisfação com a performance numérica reportada na maioria dos trabalhos, contrastando com o entusiasmo em relação ao potencial da técnica em termos da produtividade de desenvolvimento dos programas devido à facilidade de manutenção e reutilização de códigos.

É interessante observar que o MEF é intrinsicamente modular, facilitando a obtenção de códigos razoavelmente genéricos mesmo dentro do paradigma tradicional orientado por procedimentos. Porém, já se identificavam limitações de flexibilidade e dificuldades de manutenção que dificultariam a extensão da base de software existente para se lidar com as novas demandas em surgimento (Miller, 1991): computação distribuída, paralelismo, não-linearidades complexas, novos tipos de elementos e interpolações, novos recursos de hardware. As caracteríticas da POO permitem o desenvolvimento de códigos mais estáveis e tolerantes a alterações e portanto, resilientes o suficiente para acomodar essas novas funcionalidades e aplicações (Zimmermann *et al.*, 1992).

Uma vez investigados os fundamentos básicos da POO, a atenção se voltou para a busca de eficiência numérica na implementação de algoritmos (Dubois-Pèlerin e Zimmermann, 1993; Yu e Adeli, 1993; Zeglinski *et al.*, 1994). As possibilidades de se obter códigos genéricos e modulares mais claros e extensíveis foram demonstradas, mas a comparação com a performance dos códigos tradicionais em Fortran ainda era bastante desfavorável.

(Dubois-Pèlerin e Zimmermann, 1993) listaram as características do C++ que o levariam a se tornar a linguagem orientada por objetos dominante nas aplicações de computação científica: essa linguagem permite um amplo espectro de estilos de programação, apresentando desde as características de mais alto nível de orientação por objetos até a programação de baixo nível em C, como manipulação direta de memória e uso intensivo de ponteiros, e com isso podendo ser usada em praticamente todos os níveis de abstração. Os recursos de orientação por objetos seriam usadas para estruturar e organizar o código enquanto que as funções-membro das classes seriam implementadas em baixo nível. Nessa referência, por exemplo, reporta-se maior eficiência de código em C++ na solução de sistemas lineares quando comparado a implementação em Fortran justamente devido ao uso intensivo de ponteiros da sintaxe em C. Por outro lado, é menos eficiente na etapa de construção de sistemas devido às estruturas de dados mais complexas e à ligação dinâmica (acesso à tabelas de virtualidade).

(Zeglinski *et al.*, 1994) observaram que, com o C++, as classes para o MEF podem ser contruídas sobre uma base de entidades matemáticas básicas incorporando algoritmos de solução genéricos implementados em linguagem de baixo nível para alta eficiência, especialmente bibliotecas de matrizes com armazenamento esparso associados a métodos de solução de sistemas de equações lineares. Com relação às estruturas de dados, observaram a necessidade da aplicação criteriosa de hierarquias e virtualidade para amenizar o custo da ligação dinâmica. Em outras palavras, uma maior eficiência estaria, a princípio, associada a classes menos genéricas. Esse conceito, porém, tem sido revisto nos desenvolvimentos mais atuais de computação científica, nos quais busca-se extensa generalidade e alta performance simultaneamente (Haney e Crotinger, 1999; Veldhuizen, 2000).

No mesmo período, ocorreram a diversificação das aplicações e um princípio de incorporação de capacidades não-lineares. (Menétrey e Zimmermann, 1993) discutiram a extensão de um código de elasticidade linear para plasticidade J_2. (Fancello, 1993) aplicou a POO em análise de resposta, análise de sensibilidade e otimização em contato sem atrito e mecânica da fratura, juntamente com análise adaptável e geração automática de malhas. Esses trabalhos demonstram que a orientação por objetos pode ser aplicada com sucesso em contextos nos quais o enfoque algorítmico é dominante, e não apenas na modelagem de estruturas de dados complexas. Posteriormente, as características da POO passaram a ser utilizadas justamente para facilitar a implementação de algoritmos de análise adaptável (Leinen, 1995), métodos multigrid em malhas não-estruturadas (Bittencourt e Feijóo, 1996; Bittencourt e Feijóo, 1998), análise de sensibilidade e otimização (Silva, 1997; Silva e Bittencourt, 1997; Silva e Bittencourt, 1998; Silva e Bittencourt, 1999b; Silva e Bittencourt, 1999a).

(Bittencourt, 2000) construiu as classes do MEF com *arrays template*, o que possibilitou concentrar a maior parte das responsabilidades de manipulação de memória. Além disso, foram implementados diversos esquemas de armazenamento de matrizes e métodos diretos e iterativos de solução de sistemas lineares. Esse conjunto de classes serviu de base para os trabalhos (Silva, 1997; Silva e Bittencourt, 1997; Silva e Bittencourt, 1998; Silva e Bittencourt, 1999b; Silva e Bittencourt, 1999a; Silva e Bittencourt, 2000).

No final da década de 1990, algumas abstrações e estilos de modelagem já eram mais ou menos comuns: hieraquias de tipos elementos, funções de forma, materiais e regras de integração associadas a listas centralizadas de nós, incidências, restrições e carregamentos nodais, e graus de liberdade. Nessa época, ocorreram ainda diversificação dos temas (Oñate *et al.*, 1998) e um crescimento acentuado das aplicações. (Mackerle, 2000), por exemplo, reportou 111 referências de 1996 a 1999 acerca de orientação por objetos em aplicações de elementos finitos em tópicos tais como computação simbólica, geração automática de códigos, arquitetura de sistemas, visualização de dados, processamento distribuído, paralelismo e análise adaptável, em aplicações estruturais estáticas e dinâmicas, lineares e não-lineares, além de otimização e problemas multidisciplinares.

Em termos de contribuições na aplicação da POO no núcleo de solução, podem ser citados (Boogaard *et al.*, 1998) pela utilização intensiva de bibliotecas *template* de domínio público STL[12] e TNT[13] para acelerar o desenvolvimento e aplicação de linguagem de modelagem, (Dubois-Pèlerin e Pegon, 1998) no projeto de *solvers* não-lineares genéricos, aplicação de *templates* e simplificação de interfaces entre as classes. Entretanto, não há nesses trabalhos preocupação com a escala dos problemas a serem resolvidos. Em (Boogaard *et al.*, 1998), as hierarquias de elementos são extensas e faz-se uso de herança múltipla para simplificar a implementação de problemas acoplados e em (Dubois-Pèlerin e Pegon, 1998), existem apenas elementos uni e bidimensionais, regras de integração restritas e algoritmos pouco eficientes para a solução de sistemas lineares. Essas características degradam a performance do sistema com o crescimento das dimensões dos problemas a serem resolvidos.

A busca de se conciliar eficiência e generalidade de código em aplicações de complexidade e escala crescentes tem mostrado que abordagens mais genéricas de desenvolvimento de software devem ser adotadas. Paralelamente ao desenvolvimento descrito acima, houve a maturação e disseminação de técnicas modernas de engenharia de software compreendendo linguagens de modelagem, processos de desenvolvimento e ferramentas CASE[14] (Booch, 1991; Booch *et al.*, 1999; Fowler e Scott, 1997; Gamma *et al.*, 1995; Humphrey, 2000; Jacobson *et al.*, 1994; Jacobson *et al.*, 1999; Larman, 1998; Larman, 2001; Martin e Odell, 1998; Rational Software Corporation Inc., 2000; Rumbaugh *et al.*, 1991; Rumbaugh *et al.*, 1999). Esse corpo de conhecimentos em análise e projeto orientados por objetos fornece recursos de planejamento, construção e manutenção de sistemas mais poderosos que a programação orientada por objetos isoladamente.

Com o auxílio dessas técnicas, passa-se a encarar o desenvolvimento de software para o MEF de forma mais sistêmica e global. A atenção passa a se concentrar na construção de infraestruturas genéricas de software denominadas *frameworks* que sirvam de suporte para o desenvolvimento de aplicações específicas (Archer *et al.*, 1999; Besson e Foerch, 1997; Bittencourt, 2000; Devloo, 1997;

[12]Standard Template Library, também denominada Standard C++ Library.
[13]Template Numerical Toolkit.
[14]Computer Aided Software Engineering.

Devloo e Longhin, 2002; Rucki e Miller, 1996; Rucki e Miller, 1998; Sotelino *et al.*, 1998; Yu e Kumar, 2001). Um *framework* consiste de um modelo orientado por objetos da totalidade de um domínio de problema, incluindo a definição dos mecanismos e protocolos para a interação de objetos pertencendo às classes contidas nele, enquanto que bibliotecas de classes são apenas um grupo de classes relacionadas, as quais possuem algumas interações e que podem ser instanciadas como partes de um aplicação (Sotelino *et al.*, 1998). Deve-se notar, porém, que o desenvolvimento de *frameworks* envolve grande esforço de planejamento, uma vez que as relações entre as classes aumentam de complexidade com o crescimento do sistema, sendo a programação, ainda hoje, a principal fonte de custo e atrasos na simulação de problemas (Devloo e Longhin, 2002). A primeira discussão acerca da aplicação das técnicas atuais de engenharia de software foi feita por (Besson e Foerch, 1997), os quais mencionaram conceitos tais como desenvolvimento iterativo, casos de uso e modelagem de sistemas, e apresentaram uma arquitetura de grande flexibilidade baseada em composição de classes e herança dinâmica.

A construção de *framework* para elementos finitos com aplicação intensiva de *templates* no projeto das estruturas de dados foi apresentada por (Bittencourt, 2000). (Sotelino *et al.*, 1998) discutiram o desenvolvimento de *framework* orientado por objetos para a utilização de computação paralela na solução de problemas pelo MEF. Esse *framework* inclui, além da parte de solução numérica, componentes para a construção de interfaces gráficas. É interessante notar que o esforço de desenvolvimento de *frameworks* genéricos leva à identificação de novas abstrações tais como as classes de transformação e manipulação introduzidas por (Archer *et al.*, 1999) para isolar modelo de elementos finitos e o núcleo de análise. (Devloo e Longhin, 2002) discutiram o desenvolvimento de *framework* para análise adaptável *hp*, métodos multigrid de solução e computação paralela.

A recente disponibilidade de modernas bibliotecas orientadas por objetos voltadas para a solução de equações diferenciais parciais tem se mostrado como uma das tendências mais promissoras, juntamente com os recursos de engenharia de software, para a simplificação da construção de *frameworks* para aplicações de elementos finitos. De fato, por apresentarem o estado da arte em eficiência numérica e paralelização de algoritmos, essas bibliotecas facilitam o desenvolvimento de novos algoritmos, especialmente em aplicações não-lineares (Adams e Demmel, 1999; Adams, 2000a; Adams, 2000b). Podem ser citadas as bibliotecas PETSc[15] (Balay *et al.*, 2001), Diffpack[16] (Langtangen, 1999) e POOMA[17] (Oldham, 2002), além do TAO (Benson *et al.*, 2002) para otimização de grande escala.

Em quase todos as referências anteriores, as arquitetutas eram tais que o núcleo de solução acessava apenas informações do modelo discreto. Nesses casos, a interação com o usuário ocorre apenas nas etapas de pré e pós-processamento, sendo que a etapa de solução ocorre sem intervenção do usuário, o qual apenas recebe informações sobre o progresso da solução. Existe uma abordagem alternativa, na qual as entidades de elementos finitos são associadas diretamente a entidades geométricas (Mackie, 1997; Mackie, 2000). Busca-se com isso alta interatividade com os usuários. Esse tipo de arquitetura tem sido usada, por exemplo, em sistemas de CAD que passam a incorporar recursos de análise. A eficiência numérica nessas implementações é atingida através de computação concorrente (*threads*) e técnicas de subestruturação ao invés do processamento de listas centralizadas de informações. Nas estações de trabalho atuais, problemas lineares de dimensões moderadas são tratados de modo satisfatório com sistemas construídos dessa forma. Entretanto, esse tipo de abordagem ainda não é conveniente para problemas não-lineares complexos ou para problemas de grandes dimensões devido ao alto tempo de processamento, dificultando a interatividade. A viabilidade desse tipo de arquitetura para problemas de maior complexidade está associada ao aumento da capacidade de processamento e recursos de visualização além do disponível atualmente.

Existem poucas referências sobre a aplicação de técnicas de orientação por objetos na implementação de algoritmos de análise de sensibilidade e otimização. Algums fundamentos básicos para arquiteturas dessa classe de problemas foram estabelecidos por (Cardoso e Santos, 1993), os quais propuseram, por exemplo, hieraquias de variáveis de projeto e funcionais de performance. Implementando os algoritmos a partir das classes base de cada hierarquia, obtem-se uma infraestrutura genérica na qual novos tipos de variáveis e medidas de performance podem ser incluídos sem afetar o restante do código.

[15] O PETSc (Portable, Extensible Toolkit for Scientific Computation), desenvolvido no Argonne National Labs, é um dos mais avançados produtos para computação paralela da atualidade. É implementado em C por motivos de eficiência numérica mas foi projetado seguindo princípios de orientação por objetos. Concentra-se em bibliotecas para álgebra linear, incluindo pré-condicionadores, métodos de solução de sistemas e determinação de autovalores.

[16] O Diffpack consiste numa coleção de classes em C++ para a solução numérica de equações diferenciais parciais com foco particular em aplicações de elementos finitos e diferenças finitas. Essa biblioteca foi projetada para a prototipagem rápida para a simulação de novos problemas usando alto nível de abstração (Sampath e Zabaras, 2000). Essa biblioteca é originalmente seqüencial mas tem sido aplicada em paralelização (Acklam e Langtangen, 1999).

[17] O POOMA (Parallel Object-Oriented Methods and Applications Toolkit) é produto de um projeto do Los Alamos National Labs para o desenvolvimento de técnicas orientadas por objetos para computação paralela. É menos genérico que o PETSc, porém, combinado com as modernas arquiteturas de hardware para supercomputação, tem alterado os paradigmas da computação científica atual.

Além disso, parte dos cálculos de sensibilidade é de responsabilidade das classes de elementos finitos. Essa abordagem foi extendida nos trabalhos (Silva, 1997; Silva e Bittencourt, 1998; Silva e Bittencourt, 2000), nos quais propôs-se ainda a generalização dos campos de velocidade, maior integração com as classes de análise de resposta, recursos para manipulação de dados de entidades NURBS e classe de modelo parametrizado, da qual derivam os algoritmos de minimização. A integração entre análise de resposta e análise de sensibilidade é importante no tratamento de não-linearidades irreversíveis (Kim, 1999).

(Miki, 1995) aplicou os conceitos de orientação por objetos para implementar algoritmo de otimização de estruturas baseado em conhecimento. A implementação de ambiente para análise de sensibilidade e otimização em problemas de escoamento de fluidos e transferência de calor foi apresentado por (Tiller e Dantzig, 1996), no qual foram usadas tanto linguagens procedurais (C e Fortran) quanto linguagens orientadas por objetos (C++ e TCL). (Becker *et al.*, 1997) aplicou a linguagem Java no gerenciamento de otimização multidisciplinar em ambiente de computação distribuída.

A situação atual é tal que não se justifica mais utilizar linguagens procedurais em novos projetos e desconsiderar as técnicas mais modernas de engenharia de software. No contexto da Mecânica Computacional, exige-se a combinação do trabalho de várias pessoas, a incorporação freqüente de novas técnicas e a garantia da confiabilidade. Soma-se a isso a crescente complexidade dos problemas a serem tratados. As técnicas de análise, projeto e programação orientadas por objetos têm se mostrado como os recursos mais convenientes para o desenvolvimento de sistemas que explorem da melhor maneira o potencial das modernas arquiteturas de computação – incluindo os recursos de alta performance, paralelismo e sistemas distribuídos – para a solução de problemas complexos.

1.3 Definição do Problema

1.3.1 Otimização de Estruturas Hiperelásticas Não-Lineares Quasi-Incompressíveis

Vários problemas de otimização estrutural podem ser descritos pelo enunciado

$$\begin{aligned} &\min \; f(\mathsf{s}, \nabla \mathsf{s}, \mathbf{X}; \mathbf{d}) \\ &\qquad \text{sujeito a} \\ &g_i(\mathsf{s}, \nabla \mathsf{s}, \mathbf{X}; \mathbf{d}) \leq 0, \qquad i = 1, \ldots, N_g, \\ &h_j(\mathsf{s}, \nabla \mathsf{s}, \mathbf{X}; \mathbf{d}) = 0, \qquad j = 1, \ldots, N_h, \end{aligned} \tag{1.1}$$

sendo f a função objetivo, g_i e h_j os funcionais de restrição, e $\mathbf{d} \in \Re^{Nd}$ o vetor de variáveis de projeto. Os funcionais f, g_i, e h_j são genericamente denominados *funcionais de performance*, (Arora e Haug, 1979; Arora, 1989). O campo vetorial s é a resposta do problema estrutural definido no domínio não-deformado $\mathcal{B} \subset \Re^3$ e $\mathbf{X} \in \mathcal{B}$ é um ponto material. De fato, mesmo alguns problemas a rigor multiobjetivos podem ser convenientemente tratados dessa forma.

A região Ω na qual todas as restrições são simultaneamente satisfeitas é chamada de região viável e é escrita como

$$\Omega = \left\{ \mathbf{d} \in \Re^{Nd} | \; g_i \leq 0, \; h_j = 0; \; i = 1, \ldots, N_g, \;\; j = 1, \ldots, N_h \right\}. \tag{1.2}$$

No enunciado (1.1), f, g_i, e h_j foram definidos como funcionais implícitos das variáveis de projeto, *i.e.*, esses funcionais dependem da resposta estrutural s, a qual é função das variáveis de projeto. O significado do vetor $\mathbf{d}$ depende do tipo de otimização a ser executada. No caso de otimização topológica, o próprio domínio $\mathcal{B}$ é a variável de projeto e, nesse caso, os componentes de $\mathbf{d}$ controlam a ativação/desativação de elementos utilizados na discretização de $\mathcal{B}$. Em otimização de configuração, os componentes de $\mathbf{d}$ parametrizam ângulos entre elementos estruturais tais como barras, vigas e placas. No contexto deste trabalho, o vetor $\mathbf{d}$ parametriza a forma de $\mathcal{B}$ (de topologia fixa) e características discretas do problema, tais como propriedades de material e propriedades geométricas de elementos estruturais. Logo, $\mathsf{s} \equiv \mathsf{s}(\mathcal{B}(\mathbf{d}), \mathbf{d})$, sendo o problema estrutural um subproblema do enunciado de programação não-linear[18].

[18]Uma formulação alternativa seria incluir o problema estrutural como uma restrição adicional do enunciado de otimização e incluir as variáveis de resposta estrutural no conjunto de variáveis de projeto (Dias *et al.*, 1998; Herskovits *et al.*, 1998). Entretanto, essa formulação é aplicável apenas a problemas estruturais reversíveis e não será utilizada neste trabalho.

Os problemas estruturais a serem considerados neste trabalho são tema do Capítulo 2 deste texto e são descritos genericamente pelo enunciado variacional misto (2.78)

$$D\Pi(\mathsf{s})[\delta\mathsf{s}] = \begin{cases} a(\mathsf{x}, \delta\mathsf{x}) + b_1(\delta\mathsf{x}, p) - l(\delta\mathsf{x}) & = 0 \\ b_2(\mathsf{x}, \delta p) - g(p, \delta p) & = 0 \end{cases},$$

sendo $\mathsf{s} = \begin{bmatrix} \mathsf{x} & p \end{bmatrix}$, x campo vetorial de coordenadas espaciais e p campo escalar de pressão hidrostática. Observa-se que $a(\cdot,\cdot)$, $b_1(\cdot,\cdot)$, $b_2(\cdot,\cdot)$ e $g(\cdot,\cdot)$ são, geralmente, formas não-lineares.

Esse enunciado é uma forma matemática mais conveniente para tratar estruturas cujas propriedades físicas de material incluem restrições internas, como é o caso da incompressibilidade e quasi-incompressibilidade, características de sólidos reais sujeitos à deformação finita.

Uma classe importante de algoritmos de solução de problemas descritos pelo enunciado (1.1), a qual se destaca pela sua eficiência, requer as derivadas de primeira ordem dos funcionais de performance (Bazaraa *et al.*, 1993; Belegundu e Arora, 1985a; Belegundu e Arora, 1985b; Luenberger, 1989). Métodos de ordem zero (que utilizam somente os valores dos funcionais) são recomendados apenas para os casos nos quais a informação de primeira ordem não pode ser obtida de forma confiável ou quando os funcionais são não-diferenciáveis devido a seu custo computacional extremamente alto (Haslinger e Jedelský, 1996; Huang e Arora, 1997).

A relação implícita entre funcionais e variáveis de projeto através das equações diferenciais do problema estrutural, cuja solução em casos práticos de interesse requer tratamento numérico, exige metodologia específica de determinação de gradientes, denominada de *análise de sensibilidade de projeto*.

As referências citadas na Seção 1.2.2 mostram que foram desenvolvidas formulações de análise de sensibilidade para a grande parte das não-linearidades estruturais. Porém, um dos problemas não considerados em análise de sensibilidade e otimização foi a hiperelasticidade não-linear quasi-incompressível pelo MEF. Quando esse método foi aplicado, tratou-se apenas a restrição de incompressibilidade (Tortorelli, 1992; Choi e Duan, 2000), muito severa e menos realista. Além disso, foram aplicados elementos finitos simples, nos quais o campo de deslocamentos foi interpolado por quadriláteros ou hexaedros lineares. Formulações de quasi-incompressibilidade foram apenas consideradas em trabalhos com método *meshfree* RKPM (Grindeanu *et al.*, 1998; Kim, 1999).

A quasi-incompressibilidade requer um enunciado variacional modificado e uma formulação mais precisa da função de densidade de energia de deformação, a qual dificulta a derivação de expressões de componentes de tensão e de propriedades tangentes de rigidez. Formulações de análise de resposta mais apuradas (Chen e Pan, 1996; Chen *et al.*, 1996; Chen *et al.*, 1997a) não foram extendidas para análise de sensibilidade e otimização com o MEF.

Por outro lado, um aspecto muitas vezes negligenciado pela literatura consiste na coerência entre a avaliação da sensibilidade e da técnica de atualização da discretização de $\mathcal{B}$ (Bugeda e Oliver, 1993; Canales *et al.*, 1993; Dufeu *et al.*, 1997; Fancello, 1993; Özakça *et al.*, 1993; Younsi *et al.*, 1996). A manutenção dessa coerência tem influência na taxa de convergência real observada nos algoritmos de otimização de primeira ordem, os quais dependem do grau de precisão dos gradientes dos funcionais de performance na vizinhaça de cada ponto do espaço de variáveis de projeto. Enquanto que no sentido mais estrito o campo de velocidades deveria sempre ser utilizado na atualização da malha, é óbvio também que grandes alterações do domínio podem causar graves distorções nos elementos, degradando a solução da análise de resposta. Nesse caso, uma abordagem é interromper a análise, redefinir a malha e reiniciar o problema de otimização (Choi e Chang, 1994), o que é muito restritivo. De outro lado, estão as técnicas nas quais a malha é atualizada sem a utilização do campo de velocidades tais como a geração de uma nova malha para o domínio modificado (possivelmente aplicando-se análise adaptável) ou técnicas de suavização de Laplace, cujo objetivo é garantir a qualidade da discretização. Nesses casos, o comportamento dos funcionais de performance, principalmente na vizinhança do ponto de avaliação da sensibilidade fica dissociada da previsão linear fornecida pela expressão de sensibilidade. Fora dessa vizinhança, porém, a informação de primeira ordem deixa de ser determinante.

Do ponto de vista de uma ferramenta de otimização, é importante dispor das duas estratégias de atualização de discretização, determinando apenas quando cada uma é aplicável. Respostas para essas questões passam pela determinação de campos de velocidades com boas características para atualização de discretizações de domínio, recursos de geração automática de malhas e técnicas eficientes de verificação de qualidade de discretização. No caso de problemas não-lineares, técnicas *a priori* são mais convenientes devido ao custo da análise adaptável.

1.3.2 Desenvolvimento de Sistema para Análise de Resposta, Análise de Sensibilidade e Otimização

O desenvolvimento de código razoavelmente genérico para análise de resposta de estruturas incluindo comportamento não-linear é tarefa complexa. É preciso considerar diversas opções de tipos de elementos, materiais, cinemática, restrições internas, condições de contorno, pontos materiais e tipos de algoritmos de solução. Opções diferentes podem estar presentes num mesmo problema, sendo algumas conhecidas em tempo de compilação e outras apenas no momento da execução. O recurso de otimização requer ainda análise de sensibilidade, algoritmo de programação matemática, manipulação de descrição geométrica, cálculo de campos de velocidades, atualização de malhas e de condições de contorno. Somam-se também preocupações com extensibilidade, mantenabilidade e, de modo crítico, com a eficiência numérica pois certos casos de otimização de estruturas não-lineares podem ser classificados como problemas de computação de alta performance. O investimento em tempo para implementar e integrar todos esses recursos é alto, sendo necessário desenvolver o sistema numa seqüência de complexidade crescente.

Um engano dos primeiros pesquisadores de análise de sensibilidade foi considerar desnecessário investir em desenvolvimento de código. Uma das premissas iniciais era que a análise de sensibilidade consistia num pós-processamento da análise de resposta de um programa de análise de resposta qualquer, considerado como caixa-preta. Essa hipótese não se mostrou totalmente verdadeira em problemas não-lineares, observando-se um período relativamente longo entre trabalhos teóricos pioneiros, tais como (Ryu *et al.*, 1985; Choi e Santos, 1987) que já apresentaram conjunto bastante completo de conclusões, e a publicação de resultados numéricos importantes. Como citado anteriormente, a análise de sensibilidade de problemas não-lineares irreversíveis não pode ser feita com pacotes fechados pois é necessário ter acesso a todos os detalhes da execução da análise de resposta. Além disso, a execução eficiente da análise de sensibilidade requer que esta seja feita concorrentemente à análise de resposta.

Atualmente, desconsideram-se práticas comprovadas de engenharia de software, as quais possibilitariam acelerar o desenvolvimento de sistemas através da aplicação de recursos, bibliotecas e *frameworks* já desenvolvidos, eficientes e de disponibilidade pública, principalmente para atividades acadêmicas. Tais técnicas de desenvolvimento orientam a organização e a compresensão da arquitetura de sistemas, partindo-se de uma visão global de distribuição de responsabilidades entre módulos, identificação de necessidades e possibilidades de reutilização de códigos, padrões e bibliotecas.

Neste trabalho são consideradas apenas não-linearidades reversíveis. Porém, para que os resultados e conclusões aqui obtidos sejam extensíveis para problemas irreversíveis, tornou-se necessário investir parte do trabalho em técnicas específicas de desenvolvimento de código, especialmente na definição de uma arquitetura genérica que pudesse ser extendida para outros tipos de problemas. Apenas o investimento em arquitetura permite alcançar as promessas de alta produtividade de desenvolvimento de código prometidas pela tecnologia de orientação por objetos.

A utilização recente de linguagens de modelagem, tais como a UML, em publicações de computação científica não necessaricamente significa uma ênfase ou preocupação maior com arquitetura de sistemas. Arquitetura não pode ser confundida com modelagem, ainda que parte da arquitetura seja descrita com o auxílio de modelos. Além disso, grande parte dos trabalhos trazem apenas diagramas de classes, os quais são diagramas estáticos, neglicenciando descrições dinâmicas que realmente descrevam o funcionamento dos sistemas. A adoção de um processo formal de desenvolvimento de software é, em geral, a forma mais conveniente de orientar o trabalho na direção da definição da arquitetura do sistema, utilizando corretamente os recursos de desenvolvimento disponíveis e produzindo resultados de forma planejada e previsível.

1.4 Objetivos

De acordo com as considerações expostas nas seções anteriores, colocam-se os objetivos deste trabalho e a organização adotada para cumpri-los:

- Formulação de análise de resposta de problemas de cinemática não-linear (grandes deslocamentos, rotações e deformações) com material hiperelástico não-linear quasi-incompressível (*e.g.* elastômeros) pelo MEF. A inclusão da restrição interna é discutida através das formulações lagrangeana perturbada, mista e método de projeção da pressão. Esses temas são apresentados no Capítulo 2. Conceitos de Mecânica do Contínuo adotados nesse capítulo constam nos Apêndices A e B. No Apêndice C, tem-se detalhes da discretização em elementos finitos.

- Formulação de análise de sensibilidade a parâmetros discretos e variação da forma para a classe de problemas descrita no item anterior. Essa é uma das contribuições deste trabalho, uma vez que essa formulação não havia sido aplicada juntamente com o MEF para quasi-incompressibilidade com discretizações respeitando a condição de Babuska-Brezzi. Aplicação em elementos triangulares, tetraédricos, quadrangulares e hexaédricos conforme discutido no Capítulo 3. Aspectos da discretização são apresentados no Apêndice C.

- Métodos de geração de campos de velocidades de projeto no contorno e no interior de domínios bi e tridimensionais. Verificação de características de precisão dos resultados de análise de sensibilidade, manutenção da qualidade da discretização durante atualizações da geometria, eficiência numérica e influência na seqüência de processos iterativos de otimização. São apresentados critérios para verificação *a priori* da qualidade da discretização em triângulos, quadrados, tetraedros e hexaedros. Como o objetivo de utilizar a saída padrão de geradores de malha (coordenadas de nós e incidências), também foi desenvolvido algoritmo de recuperação de coordenadas paramétricas de nós situados sobre superfícies NURBS. Essas discussões, presentes no Capítulo 4, são contribuições a implementação da análise de sensibilidade e atualização de discretizações. Por fim, são citadas resumidamente características do algoritmo de programação matemática utilizado.

- Projeto e implementação de arquitetura de sistema para análise de resposta, análise de sensibilidade e otimização de estruturas utilizando recursos do processo de desenvolvimento de software *Unified Process* (Jacobson *et al.*, 1999), tal como a linguagem de modelagem UML, partindo de um conjunto de classes desenvolvida em trabalhos anteriores (Bittencourt, 2000; Silva, 1997). As funcionalidades desse conjunto inicial de classes estavam restritas a análise de resposta, sensibilidade e otimização de problemas lineares, sendo que os recursos de otimização estavam restritos a domínios bidimensionais. O núcleo de análise de resposta foi reconstruído para suportar os novos tipos de problemas a serem implementados, incluindo a definição de conjunto de elementos para problemas de cinemática finita e quasi-incompressibilidade, revisão das regras de integração numérica e funções de forma, e método de solução incremental de problemas lineares por algoritmo de Newton-Raphson com busca linear. Novos algoritmos de geração de campos de velocidades e atualização de malhas para domínios bi e tridimensionais foram incluídos com o objetivo de garantir a precisão dos cálculos de sensibilidade. As metodologias de desenvolvimento de software aplicadas são descritas no Capítulo 5.

- Estudos de casos e resultados numéricos são apresentados nos Capítulos 2, 3 e 4. Conclusões e perspectivas futuras no Capítulo 6.

Capítulo 2

Análise de Resposta para Não-Linearidades Geométricas e Hiperelasticidade

Na mecânica clássica, a força em uma mola depende somente da mudança de comprimento da mola, sendo independente da história passada de seu comprimento, bem como da taxa na qual o comprimento está mudando. Essa hipótese clássica de comportamento de uma mola em seu domínio elástico será extendida para o caso geral de corpos elásticos. Portanto, a resposta de um material elástico é independente da taxa em que ocorre a deformação; a variação da tensão é independente do intervalo de tempo decorrido entre dois estados e da trajetória de estados de deformação entre eles, sendo determinada apenas pelo estado de deformação do ponto material em relação a uma configuração de referência arbitrária.

O enunciado anterior, sem exigências termodinâmicas adicionais, define um *material elástico de Cauchy*, no qual a tensão de Cauchy $\boldsymbol{\sigma}$ depende apenas da deformação instantânea do ponto material. Como não se exige nenhuma propriedade para a equação constitutiva[1], a elasticidade de Cauchy pode apresentar uma estrutura não-conservativa. Assim, apesar de $\boldsymbol{\sigma}$ ser independente da trajetória de deformação, o trabalho realizado pela tensão é, no caso geral, dependente do caminho realizado. A

[1]Em particular, a restrição de que a equação constitutiva seja obtida a partir da derivação de uma função potencial escalar define a classe de materiais hiperelásticos ou materiais elásticos de Green.

consideração de princípios termodinâmicos leva a um subconjunto da elasticidade de Cauchy, denominada *elasticidade de Green* ou *hiperelasticidade*. Na literatura, quando se aplica o termo 'elasticidade' geralmente refere-se a essa classe de materiais.

A elasticidade linear infinitesinal talvez seja o aspecto mais conhecido da hiperelasticidade. Contudo um número considerável de aplicações práticas envolve grandes deslocamentos, rotações e deformações – conjunto chamado genericamente de *não-linearidades geométricas*[2] – e materiais cuja relação entre tensão e deformação escapa do modelo puramente linear. Elementos de suspensão e isolamento de vibrações de veículos, tais como molas e coxins, e as aplicações de elastômeros em componentes de vedação (selos, anéis, juntas, etc.) são exemplos típicos de aplicações de engenharia cujas características não-lineares não podem ser completamente desprezadas. Molas helicoidais metálicas apresentam deslocamentos finitos associados a deformações locais infinitesimais (e portanto comportamento linear do material). As características de rigidez de coxins são dominadas pela propriedade de quasi-incompressibilidade dos elastômeros, a qual associada à possibilidade de se deformarem acentuadamente dentro do regime elástico, os tornam bastante atraentes para as aplicações de vedação.

Neste capítulo, as equações constitutivas de corpos deformáveis hiperelásticos são formuladas em termos de princípios variacionais, conduzindo a expressões convenientes para tratamento numérico. Como técnica de solução, o foco principal está na formulação de elementos finitos, mas o desenvolvimento, em sua grande parte, é aplicável a qualquer método. A abordagem adotada permite incluir não-linearidades geométricas e de material, em especial características específicas dos elastômeros, tais como a incompressibilidade.

A formulação cinemática básica de corpos deformáveis é considerada pré-requisito para este capítulo, sendo apresentada no Apêndice A. Adota-se aqui uma descrição Lagrangeana total para as equações.

Inicialmente, discutem-se aspectos da equação constitutiva da hiperelasticidade: funções de densidade de energia de deformação, suas parcelas distorcionais, volumétricas e restrições internas. Em seguida, têm-se as técnicas numéricas para análise de resposta desse problema em termos de enunciados variacionais, tanto na forma padrão de minimização de um funcional de único campo como na forma mais geral de enunciados de múltiplos campos. Esse é o caso dos materiais apresentando restrições

[2] A formulação de não-linearidade geométrica decorre da não adoção da hipótese de coincidência entre a geometria do domínio inicial e a geometria obtida após a aplicação das ações externas.

internas, tais como incompressibilidade.

A formulação linear de elementos finitos é discutida suscintamente, tanto em sua forma padrão de único campo quanto na forma mista. Contudo, um material hiperelástico genérico apresenta relação tensão-deformação, e conseqüentemente enunciados variacionais, não-lineares. O método de Newton-Raphson é então aplicado, convertendo o problema original não-linear em uma seqüência de subproblemas lineares. A discretização das expressões para implementação é apresentada no Apêndice C.

A situação de grandes deslocamentos associados a pequenas deformações é tratada como caso particular no contexto geral de hiperelasticidade apenas generalizando-se a lei de Hooke. Por se tratar do caso mais simples de não-linearidade em mecânica dos sólidos é muito útil para testes de implementação.

2.1 Material Hiperelástico

2.1.1 Definição

Como apresentado na seção anterior, um material elástico é aquele no qual o comportamento constitutivo depende apenas do estado corrente da deformação. Tomando-se o tensor gradiente de deformação $\mathbf{F}(\mathbf{X})$ numa partícula $\mathbf{X}$ do corpo e seu tensor de tensões conjugado, ou seja, o primeiro tensor de Piola-Kirchhoff $\mathbf{T}$, a elasticidade pode ser descrita de forma geral pela seguinte equação constitutiva

$$\mathbf{T} = \boldsymbol{\Upsilon}\left(\mathbf{F}(\mathbf{X}), \mathbf{X}\right), \tag{2.1}$$

sendo que a dependência direta de $\mathbf{X}$ permite o tratamento de materiais não-homogêneos. Neste trabalho, porém, considera-se apenas o caso de materiais homogêneos e

$$\mathbf{T} = \boldsymbol{\Upsilon}\left(\mathbf{F}(\mathbf{X})\right).$$

Se o trabalho interno realizado pelas tensões durante um processo de deformação depende apenas do estado do corpo nos instantes de tempo inicial t_0 e final t, o comportamento do material é independente do caminho (ou histórico) das ações externas e o material é denominado *hiperelástico* (Bonet e Wood, 1997; Gurtin, 1981).

Sob essas condições, pode-se definir uma função escalar $\dot{W}$ para a densidade de potência de deformação. Lembrando que $\mathbf{T}$ é conjugado com $\dot{\mathbf{F}}$, tem-se que

$$\dot{W}(\mathbf{F}) = \mathbf{T} \cdot \dot{\mathbf{F}} = \boldsymbol{\Upsilon}(\mathbf{F}) \cdot \dot{\mathbf{F}}, \tag{2.2}$$

de tal forma que o trabalho realizado pelas tensões entre as posições inicial e corrente é

$$W(\mathbf{F}) = \int_{t_0}^{t} \boldsymbol{\Upsilon}(\mathbf{F}) \cdot \dot{\mathbf{F}}\, dt.$$

Assumindo que é possível construir uma função escalar $W(\mathbf{F})$ a partir de experimentos, então a taxa de variação de W é

$$\dot{W}(\mathbf{F}) = DW(\mathbf{F}, \mathbf{X})[\dot{\mathbf{F}}] = \frac{\partial W}{\partial \mathbf{F}}(\mathbf{F}) \cdot \dot{\mathbf{F}} = \boldsymbol{\Upsilon}(\mathbf{F}) \cdot \dot{\mathbf{F}}, \tag{2.3}$$

Comparando (2.2) e (2.3), obtem-se que o primeiro tensor de tensões de Piola-Kirchhoff é a derivada de uma função escalar $W(\mathbf{F}, \mathbf{X})$ com respeito a $\mathbf{F}$ mantendo $\mathbf{X}$ fixo, isto é,

$$\boxed{\mathbf{T} = \boldsymbol{\Upsilon}(\mathbf{F}, \mathbf{X}) = DW(\mathbf{F}, \mathbf{X}) = \frac{\partial W}{\partial \mathbf{F}}(\mathbf{F}, \mathbf{X}).} \tag{2.4}$$

A expressão anterior é valida para um material sem restrições internas, no qual as componentes de $\dot{\mathbf{F}}$ são independentes. Assim, os modelos hiperelásticos são não-dissipativos e não requerem variáveis internas no seu modelo constitutivo[3]. O funcional $W(\mathbf{F})$ é chamado de *densidade de energia de deformação.*

Demonstra-se que a propriedade (2.3) é necessária para que se respeite o axioma termodinâmico usual de que o trabalho seja não-negativo em processos fechados (Gurtin, 1981; Lemaitre e Chaboche, 1990). De fato, o trabalho é nulo em processos fechados para um material hiperelástico, ou seja, a hiperelasticidade apresenta um estrutura conservativa, caracterizando problemas potenciais.

Como apresentado no Apêndice A, a resposta de um material deve ser objetiva, ou seja, independente de qualquer movimento rígido a que o corpo possa ser submetido. No caso de um corpo elástico, demonstra-se em (Gurtin, 1981) que uma condição necessária e suficiente de objetividade é

$$\mathbf{Q}\boldsymbol{\Upsilon}(\mathbf{F})\mathbf{Q}^T = \boldsymbol{\Upsilon}(\mathbf{Q}\mathbf{F}), \qquad \forall \mathbf{Q} \in \mathrm{Orth}^+. \tag{2.5}$$

[3]Em certos processos dinâmicos observa-se aquecimento dos componentes devido ao atrito interno entre constituintes do material. Tais efeitos caracterizam a presença de irreversibilidades internas e portanto escapam ao domínio de validade da hiperelasticidade, exigindo uma formulação mais geral. Quando a viscosidade interna do material é relevante deve-se incluir termos associados à viscoleasticidade na relação constitutiva do material (Lemaitre e Chaboche, 1990; Simo e Hughes, 1998).

No caso de material hiperelástico, a condição anterior implica que a função de densidade de energia de deformação deve permanecer invariante quando o corpo sofre uma rotação rígida, ou seja,

$$W(\mathbf{QF}) = W(\mathbf{F}), \qquad \forall \mathbf{Q} \in \mathrm{Orth}^{+}.$$

Considerando a decomposição polar $\mathbf{F} = \mathbf{RU}$ dada em (A.13), tem-se então que W depende de $\mathbf{F}$ apenas através do tensor de alongamento $\mathbf{U}$. Dessa forma, para materiais elásticos com propriedades independentes do observador, $W(\mathbf{F})$ é independente do tensor de rotações, podendo ser determinado pela sua restrição a Psym, isto é,

$$W(\mathbf{F}) = W(\mathbf{RU}) = W(\mathbf{U}).$$

Por conveniência, é usual escrever W em função de $\mathbf{C} = \mathbf{U}^2 = \mathbf{F}^T\mathbf{F}$. Portanto,

$$W(\mathbf{F}) = W(\mathbf{C}).$$

A partir daí, a seguinte relação análoga a (2.3) é válida

$$\dot{W}(\mathbf{C}) = DW(\mathbf{C}, \mathbf{X})[\dot{\mathbf{C}}] = \frac{\partial W}{\partial \mathbf{C}}(\mathbf{C}) \cdot \dot{\mathbf{C}}.$$

Além disso[4],

$$\dot{W}(\mathbf{F}) = \frac{\partial W}{\partial \mathbf{F}}(\mathbf{F}) \cdot \dot{\mathbf{F}} = \mathbf{T} \cdot \dot{\mathbf{F}} = \mathbf{S} \cdot \dot{\mathbf{E}} = \frac{\partial W}{\partial \mathbf{C}}(\mathbf{C}) \cdot \dot{\mathbf{C}} = \dot{W}(\mathbf{C}).$$

A igualdade acima juntamente com a definição (A.18), de onde se observa que $\frac{1}{2}\dot{\mathbf{C}} = \dot{\mathbf{E}}$, permitem a construção da seguinte equação constitutiva totalmente Lagrangeana relacionando o segundo tensor de Piola-Kirchhoff e o tensor de deformação finita de Lagrange $\mathbf{E}$

$$\boxed{\mathbf{S}(\mathbf{C}, \mathbf{X}) = 2\frac{\partial W}{\partial \mathbf{C}} = \frac{\partial W}{\partial \mathbf{E}}.} \tag{2.6}$$

A relação entre $\mathbf{S}$ e $\mathbf{C}$ ou $\mathbf{E}$ dada pela expressão (2.6) é, em geral, não-linear. Como será visto posteriormente, num processo de solução por Newton-Raphson, essa relação deve ser linearizada com respeito a um incremento $\Delta\mathbf{u}$ na configuração corrente. Empregando a regra da cadeia, uma relação linear entre a derivada direcional de $\mathbf{S}$ e a deformação linearizada $D\mathbf{E}[\Delta\mathbf{u}]$ pode ser obtida como se segue

$$D\mathbf{S}[\Delta\mathbf{u}] = \left.\frac{d}{d\alpha}\right|_{\alpha=0} \mathbf{S}[\mathbf{E}(\mathbf{X} + \alpha\Delta\mathbf{u})] = \frac{\partial \mathbf{S}}{\partial \mathbf{E}} : \left.\frac{d}{d\alpha}\right|_{\alpha=0} \mathbf{E}(\mathbf{X} + \alpha\Delta\mathbf{u}) = \frac{\partial \mathbf{S}}{\partial \mathbf{E}} : D\mathbf{E}[\Delta\mathbf{u}],$$

[4] Ver seção A.8.

ou ainda,

$$D\mathbf{S}[\Delta\mathbf{u}] = \mathsf{C} : D\mathbf{E}[\Delta\mathbf{u}],$$

sendo

$$\mathsf{C} = \frac{\partial \mathbf{S}}{\partial \mathbf{E}} = 2\frac{\partial \mathbf{S}}{\partial \mathbf{C}} = 4\frac{\partial^2 W}{\partial \mathbf{C}\partial \mathbf{C}} \tag{2.7}$$

o tensor simétrico de quarta ordem denominado tensor de elasticidade material ou Lagrangeano.

2.1.2 Materiais Hiperelásticos Isotrópicos

As equações constitutivas de materiais hiperelásticos apresentadas anteriormente são de aplicação irrestrita. Considera-se agora o caso de material hiperelástico isotrópico. Isotropia é definida impondo-se que o comportamento constitutivo do material seja o mesmo em qualquer direção. Por exemplo, considere um corpo de prova o qual foi submetido a uma rotação rígida qualquer e posteriormente ensaiado experimentalmente. Se o resultado do ensaio para toda rotação rígida for o mesmo obtido para o corpo de prova na posição original, então o material do corpo é isotrópico.

Para o caso hiperelástico, isso implica a seguinte condição para a função de densidade de energia de deformação:

$$W(\mathbf{FQ}) = W(\mathbf{F}), \qquad \forall \mathbf{Q} \in \mathrm{Orth}^+. \tag{2.8}$$

De forma análoga à objetividade, pode-se caracterizar completamente o comportamento de W quanto à isotropia pela sua resposta ao conjunto Psym. Tomando-se a decomposição polar (A.13) tem-se que

$$W(\mathbf{F}) = W(\mathbf{RU}) = W(\mathbf{U}) = W(\mathbf{VR}) = W(\mathbf{V}).$$

Logo, W pode também ser escrito tanto em função de $\mathbf{C}$ como de $\mathbf{B}$, isto é,

$$W(\mathbf{F}) = W(\mathbf{C}) = W(\mathbf{B}).$$

Combinando isotropia e objetividade mostra-se que

$$W\left(\mathbf{QUQ}^T\right) = W(\mathbf{U}), \qquad \forall \mathbf{Q} \in \mathrm{Orth}^+.$$

Portanto, W é uma função escalar isotrópica de $\mathbf{U}$ (Ogden, 1984). Pode-se então considerála função dos invariantes $\iota_1(\mathbf{U}), \iota_2(\mathbf{U}), \iota_3(\mathbf{U})$, ou de forma equivalente, uma função simétrica dos autovalores $\lambda_1, \lambda_2, \lambda_3$ de $\mathbf{U}$ e $\mathbf{V}$, ou seja,

$$W(\mathbf{U}) \equiv W(\mathbf{V}) = W(\lambda_1, \lambda_2, \lambda_3) = W(\lambda_1, \lambda_3, \lambda_2) = W(\lambda_3, \lambda_1, \lambda_2).$$

Observa-se que $W(1,1,1) = 0$ e $\partial W(1,1,1)/\partial \lambda_i = 0$ são propriedades convenientes para representar o estado natural, correspondente ao estado não-distorcido e livre de tensões.

Além dessa última consideração, como as únicas exigências quanto à forma da função de densidade de energia de deformação foram a isotropia e objetividade, a observação do comportamento dos materiais por meio de ensaios é fundamental para escrever leis constitutivas que possam ser utilizadas na prática em simulações que reproduzam razoavelmente os comportamentos reais dos materiais.

De acordo como trabalho de Mooney depois extendido por Rivlin, a função de energia de deformação pode ser escrita em função dos três invariantes

$$\begin{aligned} I_1 &= \operatorname{tr}\mathbf{C} = \lambda_1^2 + \lambda_2^2 + \lambda_3^2, \\ I_2 &= \frac{1}{2}\left[(\operatorname{tr}\mathbf{C})^2 - \operatorname{tr}\left(\mathbf{C}^2\right)\right] = \lambda_1^2\lambda_2^2 + \lambda_2^2\lambda_3^2 + \lambda_3^2\lambda_1^2, \\ I_3 &= \det\mathbf{C} = \lambda_1^2\lambda_2^2\lambda_3^2 = (\det\mathbf{F})^2, \end{aligned} \tag{2.9}$$

do tensor de deformação direito de Cauchy-Green $\mathbf{C} = \mathbf{U}^2 = \mathbf{F}^T\mathbf{F}$, ou seja,

$$\boxed{W(\mathbf{C}) = W(I_1, I_2, I_3).} \tag{2.10}$$

A partir de (2.6), (2.10) e da regra da cadeia, o segundo tensor de Piola-Kirchhoff é dado por

$$\mathbf{S} = 2\frac{\partial W}{\partial \mathbf{C}} = 2\frac{\partial W}{\partial \mathbf{I}_1}\frac{\partial \mathbf{I}_1}{\partial \mathbf{C}} + 2\frac{\partial W}{\partial \mathbf{I}_2}\frac{\partial \mathbf{I}_2}{\partial \mathbf{C}} + 2\frac{\partial W}{\partial \mathbf{I}_3}\frac{\partial \mathbf{I}_3}{\partial \mathbf{C}}. \tag{2.11}$$

As derivadas dos invariantes em relação a $\mathbf{C}$ são dadas por (Bonet e Wood, 1997)

$$\frac{\partial \mathbf{I}_1}{\partial \mathbf{C}} = \mathbf{I}, \qquad \frac{\partial \mathbf{I}_2}{\partial \mathbf{C}} = 2\mathbf{C}, \qquad \frac{\partial \mathbf{I}_3}{\partial \mathbf{C}} = J^2\mathbf{C}^{-1},$$

sendo

$$J = \det\mathbf{F}. \tag{2.12}$$

Substituindo as relações anteriores em (2.11), tem-se

$$\mathbf{S} = 2\frac{\partial W}{\partial \mathbf{I}_1}\mathbf{I} + 4\frac{\partial W}{\partial \mathbf{I}_2}\mathbf{C} + 2J^2\frac{\partial W}{\partial \mathbf{I}_3}\mathbf{C}^{-1}. \tag{2.13}$$

O tensor de tensões de Cauchy é obtido substituindo-se (2.13) em (A.48). Portanto,

$$\boldsymbol{\sigma} = 2J^{-1}\frac{\partial W}{\partial \mathbf{I}_1}\mathbf{B} + 4J^{-1}\frac{\partial W}{\partial \mathbf{I}_2}\mathbf{B}^2 + 2J\frac{\partial W}{\partial \mathbf{I}_3}\mathbf{I}.$$

Observa-se que os invariantes de $\mathbf{C}$ e $\mathbf{B}$ são coincidentes.

2.1.3 Restrições Internas e Incompressibilidade

Certos materiais exibem padrões de comportamento que impõem restrições físicas às formas genéricas de equações constitutivas discutidas até aqui. Uma hipótese clássica é a conservação de volume em sólidos, uma vez que a maioria dos processos práticos de grandes deformações acontecem sob incompressibilidade ou *quasi-incompressibilidade*[5]. A consideração de comportamento incompressível é feita através da inclusão de um conjunto de equações de restrição no modelo constitutivo do material. Apesar desse método também ser uma idealização, seu objetivo é introduzir uma melhor aproximação do modelo real em relação à equação constitutiva não restrita.

A forma mais simples de restrição é uma equação escalar do tipo

$$h(\mathbf{F}) = 0 \tag{2.14}$$

com h suficientemente regular.

Supõe-se que h seja uma função objetiva, portanto

$$h(\mathbf{QF}) = h(\mathbf{F}), \qquad \forall \mathbf{Q} \in \mathrm{Orth}^+, \mathbf{F} \in \mathrm{Lin}^+ .$$

Considerando as mesmas hipóteses assumidas para a função de densidade de energia de deformação, pode-se escrever h como função apenas da deformação, em termos do tensor $\mathbf{U}$ ou mesmo de $\mathbf{C}$, como se segue,

$$h(\mathbf{F}) = h(\mathbf{RU}) = h(\mathbf{U}) = 0, \qquad \mathbf{R} \in \mathrm{Orth}^+,$$

$$\tilde{h}(\mathbf{C}) = \tilde{h}\left(\mathbf{U}^2\right) = h\left(\mathbf{C}^{1/2}\right) = 0.$$

[5] A quasi-incompressibilidade é empregada em materiais para os quais a restrição de incompressibilidade é muito severa, sendo obtidos campos de deslocamentos mais precisos quando alguma deformação volumétrica é assumida. Vide Seção 2.1.6.

A diferenciação de (2.14) fornece

$$\dot{h}(\mathbf{F}) \equiv \frac{\partial h}{\partial \mathbf{F}}(\mathbf{F}) \cdot \dot{\mathbf{F}} = 0. \tag{2.15}$$

Portanto, do ponto de vista variacional[6], um múltiplo de $\partial h / \partial \mathbf{F}$ pode ser adicionado à equação constitutiva (2.1) sem afetar a potência de deformação para todos os gradientes de deformação de movimentos compatíveis com (2.14). Assim, obtem-se a equação constitutiva restrita

$$\mathbf{T} = \mathbf{\Upsilon}(\mathbf{F}) + q\frac{\partial h}{\partial \mathbf{F}}(\mathbf{F}), \qquad q \in \Re, \tag{2.16}$$

sendo q um escalar que sob certas condições coincide com a pressão hisdrostática e será determinado impondo-se a restrição de incompressibilidade $J = 1$ (Bonet e Wood, 1997).

A restrição[7]

$$\boxed{h(\mathbf{F}) = \det \mathbf{F} - 1 = 0} \tag{2.17}$$

define a exigência de incompressibilidade.

Tomando-se a derivada direcional de $J = \det \mathbf{F}$ na direção de $\Delta\mathbf{F}$, obtem-se (Gurtin, 1981)

$$DJ[\Delta\mathbf{F}] = \frac{\partial J}{\partial \mathbf{F}} \cdot \Delta\mathbf{F} = (\det \mathbf{F}) \operatorname{tr}\left(\mathbf{F}^{-1}\Delta\mathbf{F}\right) = (\det \mathbf{F})\left(\mathbf{F}^{-T} \cdot \Delta\mathbf{F}\right) \quad \Rightarrow \frac{\partial J}{\partial \mathbf{F}} = (\det \mathbf{F})\,\mathbf{F}^{-T}. \tag{2.18}$$

Assim, para a restrição (2.17),

$$\frac{\partial h}{\partial \mathbf{F}}(\mathbf{F}) = (\det \mathbf{F})\,\mathbf{F}^{-T} = J\mathbf{F}^{-T}.$$

Apesar de $\det \mathbf{F} = 1$ no caso incompressível, torna-se interessante mantê-lo na expressão anterior para que a mesma também seja válida para o caso quasi-incompressível.

Substituindo a expressão acima em (2.16), obtem-se a seguinte equação para o primeiro tensor de Piola-Kirchhoff em material incompressível

$$\mathbf{T} = \mathbf{\Upsilon}(\mathbf{F}) + qJ\mathbf{F}^{-T}.$$

---

[6] De maneira simplificada, essa consideração equivale a solução do problema

$$\min f(\mathbf{u}) = \frac{1}{2}a(\mathbf{u},\mathbf{u}) - l(\mathbf{u}), \qquad s.a \quad h(\mathbf{u}) = h(\nabla\mathbf{u}) = 0.$$

Esse problema tem Lagrangeano

$$L(\mathbf{u}, q) = \frac{1}{2}a(\mathbf{u},\mathbf{u}) - l(\mathbf{u}) + qh(\nabla\mathbf{u}),$$

sendo q o multiplicador de Lagrange. A solução variacional desse problema é justamente

$$a(\mathbf{u}, \delta\mathbf{v}) + q\frac{\partial h(\nabla\mathbf{u})}{\partial \nabla\mathbf{u}} \cdot \nabla\delta\mathbf{v} = l(\delta\mathbf{v}).$$

[7] Também poderia ser escrita nas formas $\det \mathbf{U} - 1 = 0$ ou $\det \mathbf{C} - 1 = 0$.

Para o segundo tensor de Piola-Kirchhoff, tem-se

$$\mathbf{S} = \mathbf{F}^{-1}\mathbf{T} = \mathbf{F}^{-1}\boldsymbol{\Upsilon}(\mathbf{F}) + qJ\mathbf{C}^{-1}.$$

Aplicando a relação $\boldsymbol{\sigma} = \mathbf{T}\mathbf{F}^{T}(\det \mathbf{F})^{-1}$ entre o tensor de Cauchy e o primeiro tensor de Piola-Kirchhoff, determina-se a equação constitutiva da tensão de Cauchy com restrição de incompressibilidade

$$\boldsymbol{\sigma} = \boldsymbol{\Sigma}(\mathbf{F}) + qJ\mathbf{I}, \qquad \boldsymbol{\Sigma}(\mathbf{F}) = \boldsymbol{\Upsilon}(\mathbf{F})\,\mathbf{F}^{T}. \tag{2.19}$$

A expressão (2.19) define um modelo de material cuja descrição exige a inclusão de uma nova variável independente no problema de deformação original, pois a variável q somente pode ser determinada simultaneamente às demais variáveis de deslocamento através da solução da equação de Cauchy.

A consideração de equações constitutivas não-restritas reconduz o problema à forma padrão de formulação variacional, enquanto que a introdução de restrições internas (e conseqüentemente de multiplicadores de Lagrange na forma de variáveis adicionais) dá origem a formulações variacionais mistas (Brezzi e Fortin, 1991; Brenner e Scott, 1994).

2.1.4 Parcelas Distorcional e Volumétrica da Equação Constitutiva

De forma análoga à decomposição do tensor de deformação infinitesimal na soma de uma parcela volumétrica e outra distorcional[8], pode-se também decompor o tensor gradiente de defomação $\mathbf{F}$. Nesse caso, a decomposição é multiplicativa, ou seja, particiona-se $\mathbf{F}$ como uma parcela volumétrica ($\tilde{\mathbf{F}}$) seguida de uma parcela distorcional ($\bar{\mathbf{F}}$) e vice-versa. Logo,

$$\mathbf{F} = \tilde{\mathbf{F}}\bar{\mathbf{F}} = \bar{\mathbf{F}}\tilde{\mathbf{F}},$$

sendo

$$\tilde{\mathbf{F}} = J^{1/3}\mathbf{I} \qquad \text{e} \qquad \bar{\mathbf{F}} = J^{-1/3}\mathbf{F}. \tag{2.20}$$

Observa-se que a parcela distorcional satisfaz $\det \bar{\mathbf{F}} = 1$.

[8] Vide Seção A.9.

Decomposições similares podem ser efetuadas para os outros tensores de deformação. Em particular, para o tensor de Cauchy-Green $\mathbf{C}$, tem-se que a componente distorcional é

$$\bar{\mathbf{C}} = \bar{\mathbf{F}}^T\bar{\mathbf{F}} = J^{-2/3}\mathbf{F}^T\mathbf{F} = \det\mathbf{C}, \tag{2.21}$$

com $\det\mathbf{C} = J^2$.

A seguinte relação é válida entre os invariantes principais de $\bar{\mathbf{C}}$ e $\mathbf{C}$ definidos em (2.9)

$$\bar{I}_1 = I_1 I_3^{-1/3}, \quad \bar{I}_2 = I_2 I_3^{-2/3}, \quad \bar{I}_3 = 1. \tag{2.22}$$

Usando as relações anteriores, as parcelas relativas à distorção e à dilatação da equação constitutiva podem ser separadas.

Para isso, considere inicialmente a seguinte alteração de configuração x_0 de um corpo de material incompressível

$$\nabla\mathsf{x}_0 = \mathbf{F}_0, \qquad \det\mathbf{F}_0 = 1,$$

$$W_0 \equiv W_0\left(\mathbf{F}_0\right),$$

$$\mathbf{T}_0 = \frac{\partial W_0}{\partial \mathbf{F}_0} + q_0\mathbf{F}_0^{-T}. \tag{2.23}$$

Seja agora a alteração de configuração de um corpo de material compressível genérico, porém expressa em termos do representante isocórico do gradiente de seu campo de deslocamento, ou seja, $\bar{\mathbf{F}} = J^{-1/3}\mathbf{F}$. Nesse caso, define-se a energia de deformação como $W^*\left(\bar{\mathbf{F}}, J\right) \equiv W\left(\mathbf{F}\right)$.

Para materiais hiperelásticos, empregando a regra da cadeia, tem-se

$$\mathbf{T} = \frac{\partial W}{\partial \mathbf{F}} = \frac{\partial W^*}{\partial \mathbf{F}} = \frac{\partial W^*}{\partial \bar{\mathbf{F}}}\frac{\partial \bar{\mathbf{F}}}{\partial \mathbf{F}} + \frac{\partial W^*}{\partial J}\frac{\partial J}{\partial \mathbf{F}}. \tag{2.24}$$

Aplicando a definição de $\bar{\mathbf{F}}$ e a relação (2.18), tem-se

$$\begin{aligned} \frac{\partial \bar{\mathbf{F}}}{\partial \mathbf{F}} &= \frac{\partial\left(J^{-1/3}\mathbf{F}\right)}{\partial \mathbf{F}} = \frac{\partial\left(J^{-1/3}\right)}{\partial \mathbf{F}} \otimes \mathbf{F} + J^{-1/3}\left(\frac{\partial \mathbf{F}}{\partial \mathbf{F}}\right) \\ &= J^{-1/3}\left(\mathsf{I} - \frac{1}{3}\mathbf{F}^{-T} \otimes \mathbf{F}\right) = J^{-1/3}\left(\mathsf{I} - \frac{1}{3}\bar{\mathbf{F}}^{-T} \otimes \bar{\mathbf{F}}\right), \end{aligned} \tag{2.25}$$

sendo I o *tensor identidade simétrico de quarta ordem* definido em (A.5).

Substituindo (2.18) e (2.25) em (2.24)

$$\begin{aligned}
\mathbf{T} &= \frac{\partial W^*}{\partial \bar{\mathbf{F}}}\left[J^{-1/3}\left(\mathsf{I}-\frac{1}{3}\bar{\mathbf{F}}^{-T}\otimes\bar{\mathbf{F}}\right)\right]+\frac{\partial W^*}{\partial J}J\bar{\mathbf{F}}^{-T}J^{-1/3}\\
&= J^{-1/3}\left[\frac{\partial W^*}{\partial \bar{\mathbf{F}}}-\frac{1}{3}\left(\bar{\mathbf{F}}^{-T}\otimes\bar{\mathbf{F}}\right)\frac{\partial W^*}{\partial \bar{\mathbf{F}}}+J\frac{\partial W^*}{\partial J}\bar{\mathbf{F}}^{-T}\right]\\
&= J^{-1/3}\left[\frac{\partial W^*}{\partial \bar{\mathbf{F}}}-\frac{1}{3}\left(\bar{\mathbf{F}}\cdot\frac{\partial W^*}{\partial \bar{\mathbf{F}}}\right)\bar{\mathbf{F}}^{-T}+J\frac{\partial W^*}{\partial J}\bar{\mathbf{F}}^{-T}\right]\\
&= J^{-1/3}\left[\frac{\partial W^*}{\partial \bar{\mathbf{F}}}-\frac{1}{3}\operatorname{tr}\left(\frac{\partial W^*}{\partial \bar{\mathbf{F}}}\bar{\mathbf{F}}^{T}\right)\bar{\mathbf{F}}^{-T}+J\frac{\partial W^*}{\partial J}\bar{\mathbf{F}}^{-T}\right].
\end{aligned}$$

Logo, o primeiro tensor de Piola-Kirchhoff passa a ser escrito como (Ogden, 1984)

$$\mathbf{T}=\frac{\partial W}{\partial \mathbf{F}}=J^{-1/3}\left(\frac{\partial W^*}{\partial \bar{\mathbf{F}}}+\bar{q}\bar{\mathbf{F}}^{-T}\right), \tag{2.26}$$

sendo

$$\bar{q}=-\frac{1}{3}\operatorname{tr}\left(\frac{\partial W^*}{\partial \bar{\mathbf{F}}}\bar{\mathbf{F}}^{T}\right)+J\frac{\partial W^*}{\partial J}. \tag{2.27}$$

Nota-se a semelhança entre as expressões (2.23) e (2.26).

Observando que $\mathbf{C}=J^{2/3}\bar{\mathbf{C}}$, por um processo análogo, determina-se a forma correspondente para o segundo tensor de Piola-Kirchhoff, ou seja,

$$\begin{aligned}
\mathbf{S} &= J^{-2/3}\left[2\frac{\partial W^*}{\partial \bar{\mathbf{C}}}-\frac{1}{3}\left(\bar{\mathbf{C}}\cdot 2\frac{\partial W^*}{\partial \bar{\mathbf{C}}}\right)\bar{\mathbf{C}}^{-1}+J\frac{\partial W^*}{\partial J}\bar{\mathbf{C}}^{-1}\right]\\
&= J^{-2/3}\left[\frac{\partial W^*}{\partial \bar{\mathbf{E}}}-\frac{1}{3}\left(\bar{\mathbf{C}}\cdot\frac{\partial W^*}{\partial \bar{\mathbf{E}}}\right)\bar{\mathbf{C}}^{-1}+J\frac{\partial W^*}{\partial J}\bar{\mathbf{C}}^{-1}\right].
\end{aligned}$$

Como a tensão de Cauchy é dada por $\boldsymbol{\sigma}=J^{-1}\mathbf{T}\mathbf{F}^{T}$, utilizando (2.26), obtem-se

$$\boldsymbol{\sigma}=J^{-1}\left[\frac{\partial W^*}{\partial \bar{\mathbf{F}}}\bar{\mathbf{F}}^{T}-\frac{1}{3}\operatorname{tr}\left(\frac{\partial W^*}{\partial \bar{\mathbf{F}}}\bar{\mathbf{F}}^{T}\right)\right]+\frac{\partial W^*}{\partial J}\mathbf{I}.$$

A parcela da tensão $\boldsymbol{\sigma}$ relacionada à dilatação é a tensão normal média $\bar{p}$, que a partir da expressão acima assume a forma

$$\bar{p}=\frac{1}{3}\operatorname{tr}(\boldsymbol{\sigma})=\frac{\partial W^*}{\partial J}.$$

Em outras palavras, o termo $\dfrac{\partial W^*}{\partial J}$ carrega apenas informações sobre dilatação. Já o termo

$$J^{-1}\left[\frac{\partial W^*}{\partial \bar{\mathbf{F}}}\bar{\mathbf{F}}^{T}-\frac{1}{3}\operatorname{tr}\left(\frac{\partial W^*}{\partial \bar{\mathbf{F}}}\bar{\mathbf{F}}^{T}\right)\right]$$

é a parte relacionada à distorção.

Demonstra-se que (2.23) é o limite da seqüência de problemas compressíveis (2.26) para $J \to 1$ (Ogden, 1984).

2.1.5 Modelos de Materiais Hiperelásticos

Na hipótese de Mooney e Rivlin, assume-se que W é continuamente diferenciável com respeito a I_1, I_2, I_3 e escreve-se W como um série infinita de potências[9] de $(I_1 - 3)$, $(I_2 - 3)$ e $(I_3 - 1)$

$$\boxed{W(\mathbf{C}) = W(I_1, I_2, I_3) = \sum_{i,j,k=0}^{\infty} A_{ijk} (I_1 - 3)^i (I_2 - 3)^j (I_3 - 1)^k ,} \tag{2.28}$$

que significa fazer a expansão em série de Taylor em torno da configuração não-deformada de referência $(\lambda_1 = 1, \lambda_2 = 1, \lambda_3 = 1 \Rightarrow I_1 = I_2 = 3, I_3 = 1)$. Os índices i, j, k assumem valores $0, 1, 2, \ldots$ e os coeficientes A_{ijk} são independentes da deformação em qualquer região fechada que inclua a configuração de referência. A exigência de que a energia se anule na configuração de referência é facilmente cumprida adotando $A_{000} = 0$, podendo-se provar que a configuração de referência é livre de tensões se e somente se $A_{100} + 2A_{010} + A_{001} = 0$. Uma função isotrópica pode ser aproximada com a precisão desejada por uma expansão desse tipo com número finito de termos.

Formas usualmente aplicadas são obtidas para $A_{ijk} = 0$ $(k = 1, 2, 3, \ldots / i, j \neq 0)$

$$W(I_1, I_2, I_3) = \sum_{i,j=0}^{\infty} A_{ij0} (I_1 - 3)^i (I_2 - 3)^j + \sum_{r=1}^{\infty} A_{00k} (I_3 - 1)^k \tag{2.29}$$

ou $A_{ijk} = 0$ $(j, k = 1, 2, 3, \ldots / i, j \neq 0)$

$$W(I_1, I_2, I_3) = \sum_{i,j=0}^{\infty} A_{i00} (I_1 - 3)^i + \sum_{r=1}^{\infty} A_{00k} (I_3 - 1)^k . \tag{2.30}$$

No caso de materiais incompressíveis, $I_3 = 1$, sendo W dependente apenas dos outros dois invariantes. Nesse caso, a série de Rivlin (2.28) se reduz a

$$\bar{W}(I_1, I_2) = \sum_{i,j=0}^{\infty} A_{ij} (I_1 - 3)^i (I_2 - 3)^j . \tag{2.31}$$

A série infinita de Rivlin é normalmente truncada na forma de Mooney-Rivlin

$$\bar{W}(I_1, I_2) = A_{10} (I_1 - 3) + A_{01} (I_2 - 3) \tag{2.32}$$

[9] O mesmo poderia ter sido feito com os invariantes de $\mathbf{B} = \mathbf{V}^2 = \mathbf{F}\mathbf{F}^T$.

ou neo-Hooke

$$\bar{W}(I_1, I_2) = A_{10}(I_1 - 3). \tag{2.33}$$

Observa-se, porém, que as constantes de material de (2.32) e (2.33) obtidas de ensaios de tração são, em geral, adequadas somente para a reprodução de deformações finitas moderadas, podendo não reproduzir com precisão dados experimentais em deformações muito acentuadas (Chen e Pan, 1996). A função cúbica

$$\bar{W}(I_1, I_2) = A_{10}(I_1 - 3) + A_{20}(I_1 - 3)^2 + A_{30}(I_1 - 3)^3 \tag{2.34}$$

e a função cúbica modificada

$$\bar{W}(I_1, I_2) = \frac{\alpha}{\beta}\left[1 - e^{-\beta(I_1-3)}\right] + A_{10}(I_1 - 3) + A_{20}(I_1 - 3)^2 + A_{30}(I_1 - 3)^3 \tag{2.35}$$

têm fornecido bons resultados na simulação numérica de borracha (Chen *et al.*, 1996; Chen *et al.*, 2000). Por outro lado, (2.32) e (2.33) permitem estudar satisfatoriamente o comportamento de materiais hiperelásticos do ponto de vista do desenvolvimento de algoritmos numéricos (Kim, 1999; Kim *et al.*, 2001b).

2.1.6 Quasi-Incompressibilidade

Verifica-se na prática que a condição de incompressibilidade é um idealização matemática que corresponde a situação limite na qual certas grandezas físicas são consideradas infinitas. Como isso não é fisicamente possível, todos os materiais apresentam, na prática, certo grau de compressibilidade. Além disso, existem materiais para os quais a condição de incompressibilidade se mostra muito severa e não reproduz corretamente seu comportamento real. Tais materiais são denominados *aproximadamente incompressíveis* ou *quasi-incompressíveis* e um exemplo importante de materiais de engenharia desse tipo são os elastômeros ou borrachas.

Demonstra-se (Chen e Pan, 1996) que nesses casos a densidade de energia de deformação pode ser decomposta na soma de uma parcela distorcional e outra volumétrica, ou seja,

$$\boxed{W(\bar{I}_1, \bar{I}_2, I_3) = \bar{W}(\bar{I}_1, \bar{I}_2) + \tilde{W}(I_3),} \tag{2.36}$$

sendo $\bar{W}$ a *energia de distorção* e $\tilde{W}$ a *energia de dilatação.*

Para materiais quasi-incompressíveis, a energia de dilatação pode-se ser reescrita como (Chen *et al.*, 1997a)

$$\tilde{W}(I_3) = \frac{1}{\varepsilon} G(J) \tag{2.37}$$

sendo ε um parâmetro pequeno indicando o grau de compressibilidade. Formalmente, $\varepsilon \to 0$ e $G(J) \to 0$ (mais rapidamente que ε) corresponde à incompressibilidade[10]. Nesse caso, elimina-se $\tilde{W}(I_3)$ de (2.36).

Um exemplo de função $\tilde{W}(I_3)$ é (Chen *et al.*, 1997a)

$$\tilde{W}(I_3) = \frac{1}{\varepsilon}\left[\frac{1}{2}\left(J^2 - 1 - 2\ln J\right)\right]. \tag{2.38}$$

Como mencionado anteriormente, a quantidade

$$\bar{p} = \frac{\partial \tilde{W}}{\partial J} = \frac{1}{\varepsilon}\left(J - \frac{1}{J}\right)$$

corresponde à pressão hidrostática. Para materiais aproximadamente incompressíveis, J é próximo de 1 e portanto

$$\bar{p} \approx \frac{1}{\varepsilon}(J-1) = \frac{1}{\varepsilon}\left(\frac{\rho_0}{\rho} - 1\right).$$

Uma solução usual aplicada na literatura (Bonet e Wood, 1997; Chen e Pan, 1996; Chen *et al.*, 1996; Chen *et al.*, 1997a; Chen *et al.*, 1997b; Chen *et al.*, 2000; Grindeanu *et al.*, 1998; Kim, 1999) é considerar a aproximação acima, o que corresponde à seguinte função de densidade de energia de dilatação

$$\tilde{W}(I_3) = \frac{1}{2\varepsilon}(J-1)^2. \tag{2.39}$$

Juntamente com (2.32) a (2.35), que assumem material incompressível e portanto são funções de invariantes de gradientes de deformação isocóricos como $\bar{\mathbf{F}}$, geram formas aproximadamente incompressíveis de modelos de materiais como, por exemplo, (Kim, 1999)

$$W\left(\bar{I}_1, \bar{I}_2, J\right) = A_{10}\left(\bar{I}_1 - 3\right) + A_{01}\left(\bar{I}_2 - 3\right) + \frac{\tilde{\kappa}}{2}(J-1)^2. \tag{2.40}$$

O termo

$$\tilde{\kappa} = 1/\varepsilon \tag{2.41}$$

[10] Os problemas numéricos têm origem na dificuldade prática de se avaliar com a precisão necessária o termo $(1/\varepsilon)\,G(J)$ quando $(1/\varepsilon) \to \infty$ e $G(J) \to 0$, dando origem à oscilações de valor entre elementos. Ou seja, é o efeito numérico de uma indeterminação matemática.

é interpretado fisicamente como módulo volumétrico (*bulk modulus*), representando a constante de material na relação entre pressão hidrostática e deformação volumétrica (dilatação).

2.2 Técnicas Numéricas para Análise de Resposta

2.2.1 Formulação Clássica de Elementos Finitos para Problemas Lineares

Inicialmente, considere a formulação clássica de elementos finitos em problemas lineares para exemplificar o processo de solução (Brezzi e Fortin, 1991).

Seja a situação usual na qual a solução $\mathbf{u}$ de um problema físico minimiza um certo funcional Π, ou seja,

$$\mathbf{u} = \inf_{\mathbf{w}\in\mathcal{V}} \Pi\left(\mathbf{w}\right), \tag{2.42}$$

sendo $\mathcal{V}$ o espaço funcional (de Hilbert) conveniente.

Se Π for diferenciável, o mínimo será caracterizado por uma equação variacional

$$\left\langle \Pi'\left(\mathbf{u}\right), \mathbf{w}\right\rangle_{\mathcal{V}'\times\mathcal{V}} = D\Pi\left(\mathbf{u}\right)\left[\mathbf{w}\right] = 0, \qquad \forall \mathbf{w}\in\mathcal{V}, \tag{2.43}$$

sendo $\langle\cdot,\cdot\rangle_{\mathcal{V}'\times\mathcal{V}}$ usado para denotar a dualidade entre $\mathcal{V}$ e seu dual topológico $\mathcal{V}'$. A derivada $\Pi'\left(\mathbf{u}\right)$ em $\mathbf{u}$ é uma forma linear em $\mathcal{V}$, de acordo com (B.3) e a Proposição B.4.

O método clássico de Ritz para aproximar a solução de (2.42) consiste em procurar $\mathbf{u}_h \in \mathcal{V}_h$, com $\mathcal{V}_h$ um subespaço de dimensão finita de $\mathcal{V}$, que é a solução de

$$\inf_{\mathbf{w}_h\in\mathcal{V}_h} \Pi\left(\mathbf{w}_h\right) \tag{2.44}$$

ou

$$\left\langle \Pi'\left(\mathbf{u}_h\right), \mathbf{w}_h\right\rangle_{\mathcal{V}'\times\mathcal{V}} = D\Pi\left(\mathbf{u}_h\right)\left[\mathbf{w}_h\right] = 0, \qquad \forall \mathbf{w}_h\in\mathcal{V}_h. \tag{2.45}$$

Um exemplo de problema bastante conhecido é definido pela função potencial

$$\Pi\left(\mathbf{w}\right) = \frac{1}{2}a\left(\mathbf{w},\mathbf{w}\right) - l\left(\mathbf{w}\right), \tag{2.46}$$

com $a\left(\cdot,\cdot\right)$ uma forma *bilinear* em $\mathcal{V}$, suposta contínua e simétrica, e $l\left(\mathbf{w}\right)$ uma forma *linear* em $\mathcal{V}$.

Logo, (2.45) pode ser escrita como

$$a(\mathbf{u}_h, \mathbf{w}_h) = l(\mathbf{w}_h), \qquad \forall \mathbf{w}_h \in \mathcal{V}_h. \tag{2.47}$$

Seja $N = \dim \mathcal{V}_h$. Se uma base $\mathbf{w}_{1h}, \ldots, \mathbf{w}_{Nh}$ de $\mathcal{V}_h$ é escolhida, pode-se escrever $\mathbf{u}_h$ como uma combinação linear de $\mathbf{w}_{ih}$

$$\mathbf{u}_h = \sum_{i=1}^{N} \hat{u}_i \mathbf{w}_{ih}. \tag{2.48}$$

O problema é então reduzido à solução do seguinte sistema de equações

$$\sum_{i=1}^{N} K_{ij} \hat{u}_i = \hat{f}_j, \qquad 1 \leq j \leq N,$$

ou em forma matricial

$$\mathbf{K}\hat{\mathbf{u}} = \hat{\mathbf{f}}, \tag{2.49}$$

sendo $K_{ij} = a(\mathbf{w}_{ih}, \mathbf{w}_{jh})$ e $\hat{f}_j = l(\mathbf{w}_{jh})$.

A formulação pode ser extendida para o caso de $a(\cdot,\cdot)$ não-simétrico. Nesse caso, a equação (2.47) não corresponde mais a um problema de minimização. Essa generalização é usualmente denominada método de Galerkin.

Problemas na forma (2.47) têm geralmente solução única se, em particular, a forma $a(\cdot,\cdot)$ é coerciva, ou seja, se existe um número positivo real α tal que para $\forall \mathbf{w} \in \mathcal{V}$ tem-se $a(\mathbf{w}, \mathbf{w}) \geq \alpha \|\mathbf{w}\|^2_{\mathcal{V}}$ (teorema de Lax-Milgram).

Esta é a teoria geral clássica da qual os diversos métodos numéricos de solução de equações diferenciais (como elementos finitos ou *meshfree*) são casos especiais. Nesse sentido, o MEF é uma técnica genérica para construção de subespaços de dimensão finita de um espaço $\mathcal{V}$ de Hilbert de modo a aplicar o método de Ritz-Galerkin ao problema variacional. Nesse caso, $\mathcal{V}_h$ é um subespaço de $\mathcal{V}$ de dimensão finita associado a uma discretização do domínio $\mathcal{B}$, geralmente denominada *malha de elementos finitos com subdomínios* $\mathcal{B}_e$, a qual corresponde a uma triangulação[11] $\mathcal{T}_h$ de $\mathcal{B}$ tal que $\mathcal{B} = \cup_e \mathcal{B}_e$ a partir da qual essa aproximação finita é obtida.

Dessa forma, o MEF é dependente de um princípio variacional e um espaço funcional. Mudando-

[11]O termo *triangulação* é usado para denominar qualquer discretização de domínio em subdomínios côncavos de geometria simples (Brenner e Scott, 1994). Esse termo não é exclusivo a discretizações em triângulos.

se o princípio variacional e/ou o espaço em que o princípio é colocado, obtem-se uma aproximação de elementos finitos diferente, mesmo que a solução do problema contínuo seja a mesma.

2.2.2 Formulação Mista de Elementos Finitos para Problemas Lineares

Na seção anterior, o problema de minização do funcional era não-restrito. Entretanto, é usual a presença de restrições nas formulações de problemas físicos, sendo necessário desenvolver procedimentos de solução que as incluam. Assim como na Seção 2.2.1, a intenção aqui é expor de maneira genérica os aspectos gerais da formulação, exemplificando a solução de elementos finitos com restrições lineares e com isso estabelecer um contexto no qual problemas mais gerais de mecânica do contínuo possam ser analisados e formulados numericamente. Os teoremas de existência e unicidade da solução, definição precisa de espaços e operadores podem ser encontrados nas referências (Brenner e Scott, 1994; Brezzi e Fortin, 1991).

Seja $\mathcal{V}$ um espaço de Hilbert com norma $\|\cdot\|_{\mathcal{V}}$ e produto escalar $(\cdot,\cdot)_{\mathcal{V}}$. Considere uma forma bilinear contínua $a(\mathbf{u},\mathbf{w})$ (não necessariamente simétrica) em $\mathcal{V}\times\mathcal{V}$, e portanto limitada.

Seja agora outro espaço $\mathcal{Q}$ de Hilbert, com norma $\|\cdot\|_{\mathcal{Q}}$ e produto escalar $(\cdot,\cdot)_{\mathcal{Q}}$, e uma forma bilinear contínua $b(\mathbf{w},\mathbf{p})$ em $\mathcal{V}\times\mathcal{Q}$ com $|b(\mathbf{w},\mathbf{p})|\leq\|b\|\,\|\mathbf{w}\|_{\mathcal{V}}\,\|\mathbf{q}\|_{\mathcal{Q}}$.

Sejam $\mathbf{l}\in\mathcal{V}'$, $\mathbf{g}\in\mathcal{Q}'$ dados sendo $l(\mathbf{w})=\langle\mathbf{l},\mathbf{w}\rangle_{\mathcal{V}'\times\mathcal{V}}$ e $g(\mathbf{q})=\langle\mathbf{g},\mathbf{q}\rangle_{\mathcal{Q}'\times\mathcal{Q}}$. Deseja-se determinar $\mathbf{u}\in\mathcal{V}$ e $\mathbf{p}\in\mathcal{Q}$ soluções de

$$\begin{cases} a(\mathbf{u},\mathbf{w})+b(\mathbf{w},\mathbf{p}) & =l(\mathbf{w}) \qquad \forall\mathbf{w}\in\mathcal{V} \\ b(\mathbf{u},\mathbf{q}) & =g(\mathbf{q}) \qquad \forall\mathbf{q}\in\mathcal{Q} \end{cases} \tag{2.50}$$

Se a forma bilinear $a(\mathbf{u},\mathbf{w})$ é simétrica, as equações (2.50) são as condições de otimalidade do problema de ponto de sela

$$\inf_{\mathbf{w}\in\mathcal{V}}\sup_{\mathbf{q}\in\mathcal{Q}}\frac{1}{2}a(\mathbf{w},\mathbf{w})+b(\mathbf{w},\mathbf{q})-l(\mathbf{w})-g(\mathbf{q}).$$

Em outras palavras, (2.50) significa $\delta\Pi=0$ para

$$\Pi=\sup_{\mathbf{q}\in\mathcal{Q}}\frac{1}{2}a(\mathbf{w},\mathbf{w})+b(\mathbf{w},\mathbf{q})-l(\mathbf{w})-g(\mathbf{q}).$$

Da mesma forma que na formulação clássica de elementos finitos, definem-se subespaços de dimensão finita $\mathcal{V}_h \subset \mathcal{V}$ e $\mathcal{Q}_h \subset \mathcal{Q}$. Procura-se o par $\{\mathbf{u}_h, \mathbf{p}_h\} \in \mathcal{V}_h \times \mathcal{Q}_h$, solução de

$$\left\{ \begin{array}{lll} a(\mathbf{u}_h, \mathbf{w}_h) + b(\mathbf{w}_h, \mathbf{p}_h) & = l(\mathbf{w}_h) & \forall \mathbf{w}_h \in \mathcal{V}_h \\ b(\mathbf{u}_h, \mathbf{q}_h) & = g(\mathbf{q}_h) & \forall \mathbf{q}_h \in \mathcal{Q}_h \end{array} \right. . \tag{2.51}$$

Considerando $N = \dim \mathcal{V}_h$ e $M = \dim \mathcal{Q}_h$ e suas respectivas bases $\{\mathbf{w}_{ih} | 1 \leq i \leq N\}$ e $\{\mathbf{q}_{ih} | 1 \leq i \leq M\}$, definem-se

$$K_{ij} = a(\mathbf{w}_{ih}, \mathbf{w}_{jh}),$$

$$P_{ij} = b(\mathbf{w}_{ih}, \mathbf{q}_{jh}),$$

$$\hat{f}_i = l(\mathbf{w}_{ih}) = \langle \mathbf{l}, \mathbf{w}_{ih} \rangle,$$

$$\hat{g}_i = g(\mathbf{q}_{ih}) = \langle \mathbf{g}, \mathbf{q}_{ih} \rangle,$$

$$\mathbf{u}_h = \sum_{i=1}^{N} \hat{u}_i \mathbf{w}_{ih},$$

$$\mathbf{p}_h = \sum_{i=1}^{M} \hat{p}_i \mathbf{q}_{ih}.$$

A partir das expressões anteriores, (2.51) pode ser escrito na seguinte forma matricial

$$\left\{ \begin{array}{rl} \mathbf{K}\hat{\mathbf{u}} + \mathbf{P}^T \hat{\mathbf{p}} & = \hat{\mathbf{f}} \\ \mathbf{P}\hat{\mathbf{u}} & = \hat{\mathbf{g}} \end{array} \right. \tag{2.52}$$

ou

$$\left[\begin{array}{cc} \mathbf{K} & \mathbf{P}^T \\ \mathbf{P} & \mathbf{0} \end{array} \right] \left\{ \begin{array}{c} \hat{\mathbf{u}} \\ \hat{\mathbf{p}} \end{array} \right\} = \left\{ \begin{array}{c} \hat{\mathbf{f}} \\ \hat{\mathbf{g}} \end{array} \right\} \tag{2.53}$$

Assim como antes, considera-se que as bases $\{\mathbf{w}_{ih}\}$ e $\{\mathbf{q}_{ih}\}$ foram construídas usando uma técnica de elementos finitos. Para uma forma bilinear $a(\cdot, \cdot)$ simétrica, o sistema linear (2.53) é simétrico mas pode ser indefinido, propriedade diretamente relacionada ao fato de se tratar de um problema de sela.

Observa-se ainda que dependendo do tipo de função de densidade de energia de deformação W utilizada e/ou a escala de deformação, as formas $a(\cdot, \cdot)$ e $b(\cdot, \cdot)$ passam a ser não-lineares (Chen *et al.*, 1997a). Do ponto de vista da solução numérica, entretanto, o enunciado é linearizado, convertendo-se o problema original não-linear numa seqüência de subproblemas lineares. Cada um desses subproblemas é então resolvido de acordo com as técnicas apresentadas nesta seção e na anterior.

2.2.3 Materiais Quasi-incompressíveis e Incompressíveis

Assumindo que o material é hiperelástico, isotrópico e aproximadamente incompressível (quasi-incompressível), a função de densidade de energia de deformação pode ser expressa, de acordo com a Seção 2.1.6, como

$$W(\mathbf{F}) = \bar{W}(\bar{I}_1, \bar{I}_2) + \tilde{W}(I_3) = \bar{W}(\bar{I}_1, \bar{I}_2) + \frac{1}{\varepsilon} G(J). \tag{2.54}$$

Demonstra-se que existe solução para o problema hiperelástico na forma variacional padrão (B.24) com $\bar{W}$ e G respeitando certas hipóteses[12] (Chen *et al.*, 1997a).

Consideremos agora o comportamento limite de (B.24) quando o parâmetro $\varepsilon \to 0+$. Nesse caso, para enfatizar a dependência em ε, reescreve-se (B.24) como

$$\Pi^{\varepsilon}(\mathsf{x}^{\varepsilon}) = \inf_{\chi \in \mathcal{V}} \Pi^{\varepsilon}(\chi), \tag{2.55}$$

sendo

$$\Pi^{\varepsilon}(\chi) = \int_{\mathcal{B}} \left[\bar{W}(\bar{I}_1(\chi), \bar{I}_2(\chi)) + \frac{1}{\varepsilon} G(J(\chi)) \right] dV - W_{ext}(\mathbf{X}, \chi(\mathbf{X}), \nabla \chi(\mathbf{X})). \tag{2.56}$$

Na obtenção do espaço $\mathcal{V}$ apenas se exige que $\det \mathbf{F} > 0$.

O problema limite relacionado é então

$$\Pi^{(0)}\left(\mathsf{x}^{(0)}\right) = \inf_{\chi \in \mathcal{V}^{(0)}} \Pi^{(0)}(\chi),$$

sendo

$$\Pi^{(0)}(\chi) = \int_{\mathcal{B}} \bar{W}(\bar{I}_1(\chi), \bar{I}_2(\chi)) \, dV - W_{ext}(\mathbf{X}, \chi(\mathbf{X}), \nabla \chi(\mathbf{X})).$$

Nesse caso, o espaço $\mathcal{V}^{(0)}$ incorpora a restrição $\det \mathbf{F} = 1$ em sua construção.

Demonstra-se que uma seqüência de problemas quasi-incompressíveis gerados por $\varepsilon \to 0+$ dá origem a uma subseqüência de soluções $\{\mathsf{x}^{\varepsilon}\}$ que converge para $\mathsf{x}^{(0)}$ (Chen *et al.*, 1997a).

Em outras palavras, a princípio analiticamente, é possível obter uma solução de problema incompressível tão precisa quanto se queira a partir de uma uma seqüência de problemas aproximadamante

[12] Contudo, nem todos os modelos de materiais usualmente empregados cumprem todas as hipóteses do teorema, como é o exemplo do material de Mooney-Rivlin. Exemplos numéricos, entretanto, mostram que existe um mínimo para a energia potencial mesmo se certas condições são violadas.

incompressíveis, cuja solução tem a existência demonstrada.

Considere uma discretização de elementos finitos para o problema de minimização (B.24). Seja dada então uma triangulação $\mathcal{T}_h$ de $\mathcal{B}$ tal que $\mathcal{B} = \cup_e \mathcal{B}_e$ e funções de forma polinomiais definidas em um domínio padrão $\mathcal{B}_{ref}$ para o qual existe uma transformação bem conhecida $\mathcal{B}_e \to \mathcal{B}_{ref}$ inversível. Polinômios são definidos em $\mathcal{B}_{ref}$ e a partir daí um subespaço discreto $\mathcal{V}_h$ de $\mathcal{V}$ é obtido. A discretização clássica em elementos finitos de (B.24) é então

$$\Pi\left(\mathsf{x}_h\right) = \inf_{\chi_h \in \mathcal{V}_h} \Pi\left(\chi_h\right). \tag{2.57}$$

Demonstra-se que existe solução de (2.57) se existir solução de (B.24). Além disso, existe uma analogia discreta para a convergência de problemas quasi-incompressíveis, ou seja,

$$\lim_{\varepsilon \to 0+} \mathsf{x}_h^{\varepsilon}\left(\mathbf{X}\right) = \mathsf{x}_h^{(0)}\left(\mathbf{X}\right), \qquad \forall \mathbf{X} \in \mathcal{B},$$

$$\lim_{\varepsilon \to 0+} \det \mathbf{F}^{\varepsilon}\left(\mathbf{X}\right) = 1, \qquad \forall \mathbf{X} \in \mathcal{B},$$

sendo que $\mathsf{x}_h^{(0)} \in \mathcal{V}_h^{(0)}$, espaço discreto de funções incorporando a restrição $\det \mathbf{F} = 1$.

2.2.4 Formulação Lagrangeana Perturbada

Apesar dos teoremas demonstrarem a existência de solução, os mesmos não implicam nenhum procedimento para a sua determinação. De fato, à medida que $\varepsilon > 0$ se torna pequeno, a dificuldade envolvida na solução do problema discreto formulado a partir do enunciado variacional (B.24) aumenta bastante, conduzindo ao fenômeno conhecido na literatura como 'travamento' (*volumetric locking*) (Brezzi e Fortin, 1991; Chen e Pan, 1996; Chen *et al.*, 1997a; Sussman e Bathe, 1987), no qual a precisão do campo de deslocamentos é perdida.

Como na prática é complicado (e não necessário) construir um espaço $\mathcal{V}_h^{(0)}$ satisfazendo exatamente a restrição de incompressibilidade $\det \mathbf{F} = 1$, é fácil observar que empregar espaços usuais de elementos finitos para $\mathcal{V}_h^{\varepsilon}$ quando $\varepsilon \to 0+$ também não é conveniente. Os resultados disponíveis na literatura mostram que é mais eficiente utilizar o enunciado variacional modificado (B.31) e a formulação mista de elementos finitos.

O enunciado de quasi-incompressibilidade é obtido modificando-se ligeiramente (B.31) para o problema de sela em (x, p)

$$\Pi(\mathsf{x}, p) = \inf_{\chi \in \mathcal{V}} \sup_{q \in \mathcal{Q}} \Pi(\chi, q), \tag{2.58}$$

sendo

$$\begin{aligned}
&\Pi(\chi, q) = \int_{\mathcal{B}} \left[\bar{W}\left(\bar{I}_1(\chi), \bar{I}_2(\chi)\right) + q\left(J(\chi) - 1\right) - G^*(q)\right] \, dV - W_{ext}\left(\mathbf{X}, \chi(\mathbf{X}), \nabla\chi(\mathbf{X})\right), \\
&G^*(p) = \sup_{J>0} \left[ph(J) - \frac{1}{\varepsilon}G(J)\right], \\
&h(J(\mathsf{x})) \equiv h(\nabla \mathsf{x}) = \det \mathbf{F} - 1 = J - 1 = 0.
\end{aligned} \tag{2.59}$$

O enunciado variacional (2.58) é denominado *formulação Lagrangeana perturbada*, pois se relaciona com a formulação Lagrangeana tradicional (B.31) através da adição do termo $-G^*(q)$ que relaxa a condição de incompressibilidade. Nesse sentido, também (B.31) pode ser considerado caso particular de (2.58).

Seja (x, p) um ponto estacionário do funcional Π. Então a variação $\delta\Pi$ se anula, ou seja,

$$\int_{\mathcal{B}} \left[\frac{\partial \bar{W}(\mathsf{x})}{\partial \bar{I}_1}\delta\bar{I}_1 + \frac{\partial \bar{W}(\mathsf{x})}{\partial \bar{I}_2}\delta\bar{I}_2 + p\delta J + \left(J(\mathsf{x}) - 1\right)\delta p - G^{*\prime}(p)\,\delta p\right] \, dV - \left\langle W'_{ext}(\mathsf{x}), \delta\mathsf{x}\right\rangle = 0.$$

Como $\delta\mathsf{x}$ e δp são variações independentes, a equação acima pode ser reescrita na forma do seguinte sistema

$$\begin{cases}
\displaystyle\int_{\mathcal{B}} \left[\frac{\partial \bar{W}(\mathsf{x})}{\partial \bar{I}_1}\delta\bar{I}_1 + \frac{\partial \bar{W}(\mathsf{x})}{\partial \bar{I}_2}\delta\bar{I}_2 + p\delta J\right] \, dV - \left\langle W'_{ext}(\mathsf{x}), \delta\mathsf{x}\right\rangle = 0, & \forall \delta\mathsf{x} \in \mathcal{V} \\
\displaystyle\int_{\mathcal{B}} \left(J(\mathsf{x}) - 1\right)\delta p \, dV - \int_{\mathcal{B}} G^{*\prime}(p)\,\delta p \, dV = 0, & \forall \delta p \in \mathcal{Q}
\end{cases} \tag{2.60}$$

que é o ponto de partida para a aplicação da formulação mista ao problema. O sistema acima é semelhante a (2.50), porém não-linear.

Das propriedades assumidas para $W\left(\bar{I}_1, \bar{I}_2, J\right)$, tem-se que

$$\frac{\partial W}{\partial \bar{I}_1}\delta\bar{I}_1 + \frac{\partial W}{\partial \bar{I}_2}\delta\bar{I}_2 + \frac{\partial W}{\partial J}\delta J = \left(\frac{\partial W}{\partial \bar{I}_1}\frac{\partial \bar{I}_1}{\partial \mathbf{E}} + \frac{\partial W}{\partial \bar{I}_2}\frac{\partial \bar{I}_2}{\partial \mathbf{E}} + \frac{\partial W}{\partial J}\frac{\partial J}{\partial \mathbf{E}}\right) \cdot \delta\mathbf{E} = \mathbf{S} \cdot \delta\mathbf{E}.$$

Já foi comentado que $\frac{\partial W}{\partial J} = p$, apenas sendo mais conveniente do ponto de vista numérico tratar esse termo como variável independente. Assim, $(2.60)_1$ é a equação obtida a partir de (B.24) com função de energia de deformação na forma (2.54) e introduzindo a variável adicional p.

A exigência de respeito à restrição, representada variacionalmente pela equação $(2.60)_2$, pode ser resolvida simultaneamente a $(2.60)_1$ (Sussman e Bathe, 1987) ou incluída no problema através de técnicas tais como integração seletiva reduzida (Hughes, 1980) e métodos de projeção (Chen e Pan, 1996).

2.2.5 Estratégia de Solução Numérica de Problemas Não-Lineares

A estratégia básica de solução de problemas não-lineares genéricos consiste no particionamento do problema original numa seqüência de subproblemas não-lineares consecutivos. Cada subproblema é então resolvido, sendo sua solução o estado inicial do subproblema seguinte (Simo e Hughes, 1998).

Em problemas não-estacionários, esse particionamento corresponde à discretização do intervalo de ocorrência do fenômeno em incrementos de tempo por um algoritmo de integração. Procedimento análogo é aplicado a problemas estacionários, sendo que os incrementos de tempo se traduzem na aplicação de incrementos de carregamentos e deslocamentos prescritos. Dessa forma, todos os problemas são tratados numericamente como se ocorressem durante um intervalo de tempo finito discretizado em incrementos.

Formalmente, considera-se $I = \left[0, \hat{t}\right] \subset \Re$ o intervalo total de ocorrência do fenômeno, sendo conhecido o estado do sistema em $t = 0$. Particiona-se I em subintervalos $I_{n+1} =]t_n, t_{n+1}]$ de comprimento $\Delta t_{n+1} = t_{n+1} - t_n$, $j = 0, \ldots, Nl - 1$, tais que $0 = t_0 < t_1 < \ldots < t_{Nl} = \hat{t}$. Esta partição fornece a base para estratégias de integração incremental nas quais são procuradas soluções para as equações de evolução em tempos discretos t_{n+1}, $n = 0, \ldots, Nl - 1$. Num incremento típico, o estado no instante t_n é conhecido e usa-se um procedimento de integração para determinar a solução no tempo $t_{n+1} = t_n + \Delta t_{n+1}$ (Vidal e Haber, 1993).

No problema estrutural em questão, tem-se a seqüência de problemas não-lineares

$$\operatorname{Div} \mathbf{T}_{n+1} + \mathbf{b}_{0n+1} = \mathbf{0}, \tag{2.61}$$

$$\mathsf{x}_{n+1} = \bar{\mathsf{x}}_{n+1} \qquad \text{em } \Gamma^1, \tag{2.62}$$

$$\mathbf{T}_{n+1}\mathbf{n} = \bar{\mathbf{t}}_{n+1} \qquad \text{em } \Gamma^2, \tag{2.63}$$

sendo conhecida a alteração configuração x_n, correspondente ao instante anterior.

Dessa forma, cada intervalo I_{n+1} está associado a um par de esforços externos $\mathbf{b}_{0n+1}$ e $\bar{\mathbf{t}}_{n+1}$, bem como a um espaço de deslocamentos admissíveis

$$\mathcal{X}_{n+1} = \left\{ \mathsf{x}_{n+1} : \mathcal{B} \rightarrow \Re^{\dim} | \ \mathsf{x}_{n+1} \in \left[\mathfrak{H}^1 \left(\mathcal{B} \right) \right]^{\dim} ; \mathsf{x}_{n+1}|_{\Gamma 1} = \bar{\mathsf{x}}_{n+1} \right\} . \tag{2.64}$$

O espaço de variações cinematicamente admissíveis $\mathcal{V}$ é definido em todos os instantes como

$$\mathcal{V} = \left\{ \delta\mathsf{x} : \mathcal{B} \rightarrow \Re^{\dim} | \ \delta\mathsf{x} \in \left[\mathfrak{H}^1 \left(\mathcal{B} \right) \right]^{\dim} ; \delta\mathsf{x}|_{\Gamma 1} = \mathbf{0} \right\} , \tag{2.65}$$

sendo $\dim \leq 3$ a dimensão do domínio do problema e $\mathfrak{H}^1 \left(\mathcal{B} \right)$ o espaço de Sobolev de funções com derivadas de primeira ordem quadrado integráveis; Γ^1 é a região de $\partial\mathcal{B}$ sobre a qual são aplicadas condições de contorno essenciais. O espaço $\mathfrak{H}^1 \left(\mathcal{B} \right)$ é suficiente no caso de problemas planos bidimensionais, sólidos axissimétricos e sólidos tridimensionais, que será o caso dos problemas deste trabalho. No caso de vigas e cascas, por exemplo, usa-se $\mathfrak{H}^2 \left(\mathcal{B} \right)$.

Num problema não-linear em equilíbrio estático, a discretização temporal do problema estrutural não-linear corresponde à aplicação incremental de condições de contorno e carregamentos, sendo os deslocamentos determinados em incrementos seqüenciais. Nesse contexto, entende-se o tempo como um parâmetro sem significado físico – uma vez que o sistema está em equilíbrio estático – que controla a seqüência de eventos na aproximação numérica.

Um procedimento usual para integrar o problema (2.61)–(2.63) no intervalo I_{n+1} é a aplicação do método de Newton-Raphson à forma variacional do problema. O método de solução assim obtido é denominado *solução em passos de carregamento* (Oden, 1972), consistindo num método de carregamento incremental modificado no qual a estimativa inicial linear é refinada por iterações de Newton-Raphson.

2.2.6 Método de Newton-Raphson para a Formulação de Elementos Finitos de Único Campo

Nas Seções 2.2.1 e 2.2.2 a formulação de elementos finitos era linear e portanto a solução dos problemas era obtida em único passo através da solução de um sistema de equações lineares. Obviamente a introdução de qualquer não-linearidade, seja de material, geométrica ou de condição de contorno altera essa situação, exigindo um procedimento de solução diferente.

O método de Newton-Raphson é um método genérico de solução de equações não-lineares, consistindo num procedimento iterativo de ponto fixo (Kreyszig, 1989) que aproxima a solução do problema original através de uma seqüência de soluções de aproximações lineares.

Considere o problema (B.24), cuja solução é obtida resolvendo-se (B.23), que dá origem à equação não-linear

$$\boxed{\delta\Pi\left(\mathsf{x}_{n+1},\delta\mathsf{x}\right)=D\Pi\left(\mathsf{x}_{n+1}\right)\left[\delta\mathsf{x}\right]=a\left(\mathsf{x}_{n+1},\delta\mathsf{x}\right)-l_{n+1}\left(\delta\mathsf{x}\right)=0,\qquad \mathsf{x}_{n+1}\in\mathcal{X}_{n+1},\forall\delta\mathsf{x}\in\mathcal{V},} \tag{2.66}$$

sendo

$$a\left(\mathsf{x}_{n+1},\delta\mathsf{x}\right)=\delta W_{int}\left(\mathsf{x}_{n+1},\delta\mathsf{x}\right)=\int_{\mathcal{B}}\mathbf{S}_{n+1}\cdot\delta\mathbf{E}\;dV, \tag{2.67}$$

$$l_{n+1}\left(\delta\mathsf{x}\right)=\delta W_{ext}\left(\mathsf{x}_{n+1},\delta\mathsf{x}\right)=\left\langle W'_{ext},\delta\mathsf{x}\right\rangle_{n+1}=\int_{\Gamma 2}\bar{\mathbf{t}}_{n+1}\cdot\delta\mathsf{x}\;dA+\int_{\mathcal{B}}\mathbf{b}_{0n+1}\cdot\delta\mathsf{x}\;dV. \tag{2.68}$$

Aplicando Newton-Raphson a (2.66), tem-se o procedimento iterativo

$$\boxed{\begin{array}{l}\delta\Pi\left({}^{(k)}\mathsf{x}_{n+1},\delta\mathsf{x}\right)+D\delta\Pi\left({}^{(k)}\mathsf{x}_{n+1},\delta\mathsf{x}\right)\left[{}^{(k+1)}\Delta\mathsf{x}\right]=0,\\[2em] {}^{(k+1)}\mathsf{x}_{n+1}={}^{(k)}\mathsf{x}_{n+1}+{}^{(k+1)}\Delta\mathsf{x},\\[2em] k+1\rightarrow k,\qquad \text{até que }\left\|{}^{(k+1)}\Delta\mathsf{x}\right\|\leq\varepsilon>0,\end{array}} \tag{2.69}$$

tal que

$$\begin{aligned}{}^{(k+1)}\mathsf{x}_{n+1}&=\mathbf{X}+{}^{(k+1)}\mathbf{u}_{n+1}={}^{(k)}\mathsf{x}_{n+1}+{}^{(k+1)}\Delta\mathsf{x}\\&=\mathbf{X}+{}^{(k)}\mathbf{u}_{n+1}+{}^{(k+1)}\Delta\mathsf{x}=\mathbf{X}+{}^{(k)}\mathbf{u}_{n+1}+{}^{(k+1)}\Delta\mathbf{u},\end{aligned}$$

$${}^{(k+1)}\Delta\mathsf{x}={}^{(k+1)}\Delta\mathbf{u}.$$

Em (2.69) (omitindo os índices da discretização temporal e da iteração de Newton-Raphson),

$$D\delta\Pi\left(\mathsf{x},\delta\mathsf{x}\right)\left[\Delta\mathbf{u}\right]=D\delta W_{int}\left(\mathsf{x},\delta\mathsf{x}\right)\left[\Delta\mathbf{u}\right]-D\delta W_{ext}\left(\mathsf{x},\delta\mathsf{x}\right)\left[\Delta\mathbf{u}\right], \tag{2.70}$$

sendo

$$\begin{aligned}D\delta W_{int}\left(\mathsf{x},\delta\mathsf{x}\right)\left[\Delta\mathbf{u}\right]&=\int_{\mathcal{B}}D\left(\mathbf{S}\cdot\delta\mathbf{E}\right)\left[\Delta\mathbf{u}\right]\;dV\\&=\int_{\mathcal{B}}D\mathbf{S}\left[\Delta\mathbf{u}\right]\cdot\delta\mathbf{E}\;dV+\int_{\mathcal{B}}\mathbf{S}\cdot D\delta\mathbf{E}\left[\Delta\mathbf{u}\right]\;dV\\&=\int_{\mathcal{B}}\mathsf{C}\left(\mathsf{x}\right):D\mathbf{E}\left[\Delta\mathbf{u}\right]\cdot\delta\mathbf{E}\;dV+\int_{\mathcal{B}}\mathbf{S}\cdot D\delta\mathbf{E}\left[\Delta\mathbf{u}\right]\;dV.\end{aligned} \tag{2.71}$$

Pode-se observar na expressão anterior a seguinte aplicação da regra da cadeia:

$$D\mathbf{S}(\mathsf{x})[\Delta\mathbf{u}] = \frac{\partial\mathbf{S}}{\partial\mathbf{E}}(\mathsf{x}) : D\mathbf{E}[\Delta\mathbf{u}] = \mathsf{C}(\mathsf{x}) : D\mathbf{E}[\Delta\mathbf{u}]. \tag{2.72}$$

O termo $D\mathbf{E}[\Delta\mathbf{u}]$ descreve a variação no tensor de deformação finita devido à variação de deslocamento $\Delta\mathbf{u}$, ou seja,

$$\begin{aligned} D\mathbf{E}[\Delta\mathbf{u}] &= \Delta\mathbf{E} = \frac{1}{2}\left(\mathbf{F}^T D\mathbf{F}[\Delta\mathbf{u}] + D\mathbf{F}^T[\Delta\mathbf{u}]\,\mathbf{F}\right) = \frac{1}{2}\left(\mathbf{F}^T\nabla(\Delta\mathbf{u}) + \nabla(\Delta\mathbf{u})^T\mathbf{F}\right) \\ &= \frac{1}{2}\left(\mathbf{F}^T(\mathrm{grad}\,\Delta\mathbf{u})_m\,\mathbf{F} + \mathbf{F}^T(\mathrm{grad}\,\Delta\mathbf{u})_m^T\,\mathbf{F}\right) \\ &= \mathbf{F}^T\left[\frac{1}{2}\left(\mathrm{grad}\,\Delta\mathbf{u} + \mathrm{grad}^T\,\Delta\mathbf{u}\right)_m\right]\mathbf{F} = \mathbf{F}^T\epsilon_m\mathbf{F}. \end{aligned}$$

Esse termo pode ser interpretado como a transformação da deformação infinitesimal devido a $\Delta\mathbf{u}$ na configuração deformada x para uma medida de deformação material.

O termo $\delta\mathbf{E}$ tem forma e interpretação semelhantes

$$\delta\mathbf{E} = \frac{1}{2}\left(\mathbf{F}^T\delta\mathbf{F} + \delta\mathbf{F}^T\mathbf{F}\right) = D\mathbf{E}[\delta\mathsf{x}], \qquad \delta\mathbf{F} = \nabla\delta\mathsf{x},$$

sendo entretanto a variação devido a $\delta\mathsf{x}$.

Da expressão acima, obtem-se $D\delta\mathbf{E}[\Delta\mathbf{u}]$, sua derivada direcional (lembrando que $\delta\mathsf{x}$ *não* é função da configuração), ou seja,

$$D\delta\mathbf{E}[\Delta\mathbf{u}] = \frac{1}{2}\left(D\mathbf{F}^T[\Delta\mathbf{u}]\,\delta\mathbf{F} + \delta\mathbf{F}^T D\mathbf{F}[\Delta\mathbf{u}]\right) = \frac{1}{2}\left(\nabla(\Delta\mathbf{u})^T\delta\mathbf{F} + \delta\mathbf{F}^T\nabla(\Delta\mathbf{u})\right).$$

Em resumo,

$$\begin{aligned} D\delta W_{int}(\mathsf{x},\delta\mathsf{x})[\Delta\mathbf{u}] &= \int_{\mathcal{B}} \mathsf{C}(\mathsf{x}) : D\mathbf{E}[\Delta\mathbf{u}]\cdot D\mathbf{E}[\delta\mathsf{x}]\;dV \\ &+ \int_{\mathcal{B}} \mathbf{S}(\mathsf{x})\cdot\left[\frac{1}{2}\left(\nabla(\Delta\mathbf{u})^T\nabla\delta\mathsf{x} + \nabla\delta\mathsf{x}^T\nabla(\Delta\mathbf{u})\right)\right]\;dV = \\ &= Da(\mathsf{x},\delta\mathsf{x})[\Delta\mathbf{u}] = \delta a(\mathsf{x};\Delta\mathbf{u},\delta\mathsf{x}). \end{aligned} \tag{2.73}$$

Observa-se que $\delta a(\mathsf{x};\Delta\mathbf{u},\delta\mathsf{x})$ é uma forma bilinear simétrica em $\Delta\mathbf{u}$ e $\delta\mathsf{x}$ obtida na configuração x.

Considerando que o carregamento no contorno seja independente da deformação (o caso dependente da deformação será tratado na Seção 2.2.10), ou seja, $D\delta W_{ext}(\mathsf{x},\delta\mathsf{x})[\Delta\mathbf{u}] = 0$, o método iterativo

(2.69) passa a ser escrito como,

$$\boxed{\begin{aligned}
&\delta a\left({}^{(k)}\mathsf{x}_{n+1};\ {}^{(k+1)}\Delta\mathbf{u},\delta\mathsf{x}\right) = l_{n+1}(\delta\mathsf{x}) - a\left({}^{(k)}\mathsf{x}_{n+1},\delta\mathsf{x}\right),\\
&{}^{(k+1)}\mathsf{x}_{n+1} = {}^{(k)}\mathsf{x}_{n+1} + {}^{(k+1)}\Delta\mathbf{u},\\
&k+1\rightarrow k, \qquad \text{até que } \left\|{}^{(k+1)}\Delta\mathbf{u}\right\| \leq \varepsilon > 0.
\end{aligned}} \tag{2.74}$$

Como $l_{n+1}(\delta\mathsf{x}) - a\left({}^{(k)}\mathsf{x}_{n+1},\delta\mathsf{x}\right)$ é linear em $\delta\mathsf{x}$, a solução de (2.74) em cada passo $k+1$ segue o procedimento apresentado na Seção 2.2.1.

O critério de convergência acima decorre da formulação usual de algoritmo de ponto interior (Kreyszig, 1989). Dada a diversidade de comportamentos que podem ser observados em problemas estruturais não-lineares, critérios de convergência alternativos podem ser aplicados, tais como (Bathe, 1996)

$$\left\|\left[\,\hat{\mathbf{f}}_{ext}\right]_{n+1} - {}^{(k)}\left[\hat{\mathbf{f}}_{int}\right]_{n+1}\right\| \Big/ \left\|\left[\,\hat{\mathbf{f}}_{ext}\right]_{n+1} - {}^{(0)}\left[\hat{\mathbf{f}}_{int}\right]_{n+1}\right\| \leq \varepsilon, \tag{2.75}$$

que mede a proximidade da raiz de $a\left({}^{(k)}\mathsf{x}_{n+1},\delta\mathsf{x}\right) - l_{n+1}(\delta\mathsf{x})$, ou (na notação do Apêndice C)

$$\left|\left(\left[\,\hat{\mathbf{f}}_{ext}\right]_{n+1} - {}^{(k)}\left[\hat{\mathbf{f}}_{int}\right]_{n+1}\right)\cdot{}^{(k+1)}\left[\Delta\hat{\mathbf{u}}\right]\right| \Big/ \left|\left(\left[\,\hat{\mathbf{f}}_{ext}\right]_{n+1} - {}^{(0)}\left[\hat{\mathbf{f}}_{int}\right]_{n+1}\right)\cdot{}^{(1)}\left[\Delta\hat{\mathbf{u}}\right]\right| \leq \varepsilon, \tag{2.76}$$

que monitora o módulo relativo da derivada direcional do problema de minimização descrito por (2.74). Nesse critério, controla-se simultaneamente o módulo do incremento do campo de deslocamentos e o módulo do desbalanço.

2.2.7 Não-Linearidade Geométrica com Material Elástico Linear

Esse tipo de não-linearidade ocorre quando um corpo apresenta grandes deslocamentos mantendo, ao mesmo tempo, pequenas deformações em seus pontos materiais.

Uma descrição simples e largamente empregada é a generalização da lei de Hooke (A.52)

$$\mathbf{S} = \mathsf{C} : \mathbf{E} \tag{2.77}$$

sendo C um tensor *constante* dado por

$$\mathsf{C} = 2\tilde{\mu}\mathsf{I} + \tilde{\lambda}\left(\mathbf{I}\otimes\mathbf{I}\right) = 2\tilde{\mu}\mathsf{I}_{\text{dev}} + \tilde{\kappa}\left(\mathbf{I}\otimes\mathbf{I}\right),$$

cujas constantes foram definidas em (A.68).

Obviamente, (2.77) se reduz a lei de Hooke original no caso de pequenas deformações. Entretanto, o ponto importante dessa relação constitutiva está no fato de $\mathbf{E}$ ser insensível a movimentos rígidos. Por isso, os grandes deslocamentos e rotações são automaticamente desprezados e, como as deformações são pequenas, retorna-se pontualmente à lei de Hooke.

Em termos do algoritmo de Newton-Raphson (2.69), a versão incremental (2.72) da relação tensão-deformação (2.77) é

$$\Delta \mathbf{S} = D\mathbf{S}\left(\mathsf{x}\right)\left[\Delta \mathbf{u}\right] = \frac{\partial \mathbf{S}}{\partial \mathbf{E}} : D\mathbf{E}\left[\Delta \mathbf{u}\right] = \mathsf{C} : D\mathbf{E}\left[\Delta \mathbf{u}\right] = \mathsf{C} : \Delta \mathbf{E}.$$

2.2.8 Método de Newton-Raphson para a Formulação Lagrangeana Perturbada

Para o problema de sela (2.58) definido em um intervalo de tempo I_{n+1}, a solução é obtida também resolvendo-se uma condição de ponto estacionário, só que na forma do sistema de equações variacionais (2.60), reescrito abaixo:

$$\boxed{\begin{array}{l} \delta\Pi\left(\mathsf{s}_{n+1}, \delta\mathsf{s}\right) = D\Pi\left(\mathsf{s}_{n+1}\right)\left[\delta\mathsf{s}\right] = \begin{cases} a\left(\mathsf{x}_{n+1}, \delta\mathsf{x}\right) + b_1\left(\delta\mathsf{x}, p_{n+1}\right) - l\left(\delta\mathsf{x}\right) & = 0 \\ b_2\left(\mathsf{x}_{n+1}, \delta p\right) - g\left(p_{n+1}, \delta p\right) & = 0 \end{cases}, \\ \\ \mathsf{x}_{n+1} \in \mathcal{X}_{n+1}, \forall \delta\mathsf{x} \in \mathcal{V},\ p_{n+1}, \delta p \in \mathcal{Q}, \end{array}} \tag{2.78}$$

sendo

$$\mathsf{s}_{n+1} = \begin{bmatrix} \mathsf{x}_{n+1} & p_{n+1} \end{bmatrix}^T, \tag{2.79}$$

$$\delta\mathsf{s} = \begin{bmatrix} \delta\mathsf{x} & \delta p \end{bmatrix}^T, \tag{2.80}$$

$$a\left(\mathsf{x}_{n+1}, \delta\mathsf{x}\right) = \int_{\mathcal{B}} \left[\frac{\partial \bar{W}\left(\mathsf{x}_{n+1}\right)}{\partial \bar{I}_1}\delta \bar{I}_1 + \frac{\partial \bar{W}\left(\mathsf{x}_{n+1}\right)}{\partial \bar{I}_2}\delta \bar{I}_2\right] dV = \int_{\mathcal{B}} \bar{\mathbf{S}}_{n+1} \cdot \delta\mathbf{E}\ dV, \tag{2.81}$$

$$b_1\left(\delta\mathsf{x}, p_{n+1}\right) = \int_{\mathcal{B}} p_{n+1}\delta J\ dV, \tag{2.82}$$

$$b_2\left(\mathsf{x}_{n+1}, \delta p\right) = \int_{\mathcal{B}} \left(J\left(\mathsf{x}_{n+1}\right) - 1\right)\delta p\ dV, \tag{2.83}$$

$$g\left(p_{n+1}, \delta p\right) = \int_{\mathcal{B}} G^{*\prime}\left(p_{n+1}\right)\delta p\ dV. \tag{2.84}$$

Aplicando Newton-Raphson a (2.78), tem-se o seguinte procedimento iterativo,

$$\boxed{\begin{aligned} &\delta\Pi\left({}^{(k)}\mathsf{s}_{n+1},\delta\mathsf{s}\right)+D\delta\Pi\left({}^{(k)}\mathsf{s}_{n+1},\delta\mathsf{s}\right)\left[{}^{(k+1)}\Delta\mathsf{s}\right]=0,\\ &{}^{(k+1)}\mathsf{s}_{n+1}={}^{(k)}\mathsf{s}_{n+1}+{}^{(k+1)}\Delta\mathsf{s},\\ &k+1\rightarrow k,\qquad \text{até que }\left\|{}^{(k+1)}\Delta\mathsf{s}\right\|\leq\varepsilon>0,\end{aligned}} \tag{2.85}$$

observando que ${}^{(k+1)}\Delta\mathsf{s}=\left[\begin{array}{cc}{}^{(k+1)}\Delta\mathsf{x} & {}^{(k+1)}\Delta p\end{array}\right]^T=\left[\begin{array}{cc}{}^{(k+1)}\Delta\mathbf{u} & {}^{(k+1)}\Delta p\end{array}\right]^T$ para a variação finita aplicada a ${}^{(k)}\mathsf{s}_{n+1}$.

O termo linearizado de (2.85) tem a seguinte forma geral (omitindo os índices da discretização temporal e da iteração de Newton-Raphson):

$$D\delta\Pi\left(\mathsf{s},\delta\mathsf{s}\right)\left[\Delta\mathsf{s}\right]=\left\{\begin{array}{l}D\delta W_{int}\left(\mathsf{s},\delta\mathsf{s}\right)\left[\Delta\mathsf{s}\right]-D\delta W_{ext}\left(\mathsf{s},\delta\mathsf{s}\right)\left[\Delta\mathsf{s}\right]\\ D\delta W_{p}\left(\mathsf{s},\delta\mathsf{s}\right)\left[\Delta\mathsf{s}\right]\end{array}\right. \tag{2.86}$$

Desenvolvendo o primeiro termo de (2.86), obtem-se

$$\begin{aligned} D\delta W_{int}\left(\mathsf{s},\delta\mathsf{s}\right)\left[\Delta\mathsf{s}\right] &= \int_{\mathcal{B}}\bar{\mathsf{C}}\left(\mathsf{x}\right):D\mathbf{E}\left[\Delta\mathbf{u}\right]\cdot D\mathbf{E}\left[\delta\mathsf{x}\right]\ dV\\ &+\int_{\mathcal{B}}\bar{\mathbf{S}}\cdot\left[\frac{1}{2}\left(\nabla(\Delta\mathbf{u})^T\ \nabla\delta\mathsf{x}+\nabla\delta\mathsf{x}^T\ \nabla(\Delta\mathbf{u})\right)\right]\ dV+\\ &+\int_{\mathcal{B}}\delta J Dp\left[\Delta p\right]\ dV+\int_{\mathcal{B}}pD\delta J\left[\Delta\mathbf{u}\right]\ dV. \end{aligned} \tag{2.87}$$

Os termos de tensão e do tensor de elasticidade tangente são definidos abaixo como

$$\mathbf{S}=W_{,\mathbf{E}}=\frac{\partial W}{\partial\bar{I}_1}\frac{\partial\bar{I}_1}{\partial\mathbf{E}}+\frac{\partial W}{\partial\bar{I}_2}\frac{\partial\bar{I}_2}{\partial\mathbf{E}}+p\frac{\partial J}{\partial\mathbf{E}}, \tag{2.88}$$

$$\bar{\mathbf{S}}=\frac{\partial W}{\partial\bar{I}_1}\frac{\partial\bar{I}_1}{\partial\mathbf{E}}+\frac{\partial W}{\partial\bar{I}_2}\frac{\partial\bar{I}_2}{\partial\mathbf{E}}, \tag{2.89}$$

$$\tilde{\mathbf{S}}=p\frac{\partial J}{\partial\mathbf{E}}, \tag{2.90}$$

$$\bar{\mathsf{C}}\left(\mathsf{x}\right)=\frac{\partial\bar{\mathbf{S}}}{\partial\mathbf{E}}\left(\mathsf{x}\right)=\frac{\partial^2\bar{W}}{\partial\mathbf{E}^2}\left(\mathsf{x}\right), \tag{2.91}$$

$$\tilde{\mathsf{C}}\left(\mathsf{s}\right)=\frac{\partial\tilde{\mathbf{S}}}{\partial\mathbf{E}}\left(\mathsf{s}\right)=p\frac{\partial^2 J}{\partial\mathbf{E}^2}\left(\mathsf{x}\right). \tag{2.92}$$

Desenvolvendo os demais termos de (2.87), tem-se

$$\delta J = \frac{\partial J}{\partial \mathbf{E}} \cdot \delta \mathbf{E} = \frac{\partial J}{\partial \mathbf{E}} \cdot D\mathbf{E}\left[\delta \mathsf{x}\right], \tag{2.93}$$

$$D\delta J\left[\Delta \mathbf{u}\right] = D\left[\frac{\partial J}{\partial \mathbf{E}} \cdot \delta \mathbf{E}\right]\left[\Delta \mathbf{u}\right] = \frac{\partial J}{\partial \mathbf{E}} \cdot D\delta \mathbf{E}\left[\Delta \mathbf{u}\right] + D\mathbf{E}\left[\delta \mathsf{x}\right] \cdot D\left[\frac{\partial J}{\partial \mathbf{E}}\right]\left[\Delta \mathbf{u}\right], \tag{2.94}$$

$$\begin{aligned} pD\delta J\left[\Delta \mathbf{u}\right] &= p\frac{\partial J}{\partial \mathbf{E}} \cdot D\delta \mathbf{E}\left[\Delta \mathbf{u}\right] + D\mathbf{E}\left[\delta \mathsf{x}\right] \cdot pD\left[\frac{\partial J}{\partial \mathbf{E}}\right]\left[\Delta \mathbf{u}\right] \\ &= \tilde{\mathbf{S}} \cdot \left[\frac{1}{2}\left(\nabla(\Delta \mathbf{u})^T \nabla \delta \mathsf{x} + \nabla \delta \mathsf{x}^T \nabla(\Delta \mathbf{u})\right)\right] + p\frac{\partial^2 J}{\partial \mathbf{E}^2} : D\mathbf{E}\left[\Delta \mathbf{u}\right] \cdot D\mathbf{E}\left[\delta \mathsf{x}\right] \\ &= \tilde{\mathbf{S}} \cdot \left[\frac{1}{2}\left(\nabla(\Delta \mathbf{u})^T \nabla \delta \mathsf{x} + \nabla \delta \mathsf{x}^T \nabla(\Delta \mathbf{u})\right)\right] + \tilde{\mathsf{C}}\left(\mathsf{s}\right) : D\mathbf{E}\left[\Delta \mathbf{u}\right] \cdot D\mathbf{E}\left[\delta \mathsf{x}\right]. \end{aligned} \tag{2.95}$$

Substituindo (2.90) a (2.95) em (2.87) e agrupando os termos, reescreve-se $D\delta W_{int}\left(\mathsf{s}, \delta \mathsf{s}\right)\left[\Delta \mathsf{s}\right]$ na forma mais conveniente seguinte

$$\begin{aligned} D\delta W_{int}\left(\mathsf{s}, \delta \mathsf{s}\right)\left[\Delta \mathsf{s}\right] &= \int_{\mathcal{B}} \mathsf{C}\left(\mathsf{s}\right) : D\mathbf{E}\left[\Delta \mathbf{u}\right] \cdot D\mathbf{E}\left[\delta \mathsf{x}\right] \, dV \\ &+ \int_{\mathcal{B}} \mathbf{S} \cdot \left[\frac{1}{2}\left(\nabla(\Delta \mathbf{u})^T \nabla \delta \mathsf{x} + \nabla \delta \mathsf{x}^T \nabla(\Delta \mathbf{u})\right)\right] \, dV \\ &+ \int_{\mathcal{B}} D\mathbf{E}\left[\delta \mathsf{x}\right] \cdot \frac{\partial J}{\partial \mathbf{E}} Dp\left[\Delta p\right] \, dV, \end{aligned} \tag{2.96}$$

sendo $\mathsf{C} = \bar{\mathsf{C}} + \tilde{\mathsf{C}}$ e $\mathbf{S} = \bar{\mathbf{S}} + \tilde{\mathbf{S}}$.

Desenvolvendo a segunda equação de (2.86), obtem-se

$$\begin{aligned} D\delta W_p\left(\mathsf{s}, \delta \mathsf{s}\right)\left[\Delta \mathsf{s}\right] &= \int_{\mathcal{B}} \delta p\left[DJ\left(\mathsf{x}\right)\left[\Delta \mathbf{u}\right] - DG^{*\prime}\left(p\right)\left[\Delta p\right]\right] \, dV \\ &= \int_{\mathcal{B}} \delta p\left[D\mathbf{E}\left[\Delta \mathbf{u}\right] \cdot \frac{\partial J}{\partial \mathbf{E}} - DG^{*\prime}\left(p\right)\left[\Delta p\right]\right] \, dV, \end{aligned} \tag{2.97}$$

observando-se que $D\delta p\left[\Delta \mathsf{s}\right] = 0$, pois δp não depende de s.

Se o carregamento é independente da deformação então $D\delta W_{ext}(\mathsf{s}, \delta\mathsf{s})[\Delta\mathsf{s}] = 0$ e o algoritmo (2.85) assume a forma

$$\boxed{\begin{array}{l} \left\{ \begin{array}{ll} \delta a\left({}^{(k)}\mathsf{s}_{n+1};\ {}^{(k+1)}\Delta\mathsf{x}, \delta\mathsf{x}\right) + \delta b\left({}^{(k)}\mathsf{s}_{n+1}; \delta\mathsf{x},\ {}^{(k+1)}\Delta p\right) & = l_{n+1}(\delta\mathsf{x}) - a\left({}^{(k)}\mathsf{x}_{n+1}, \delta\mathsf{x}\right) \\ & \quad -b_1\left(\delta\mathsf{x},\ {}^{(k)}p_{n+1}\right) \\ \delta b\left({}^{(k)}\mathsf{s}_{n+1};\ {}^{(k+1)}\Delta\mathsf{x}, \delta p\right) & = \delta g\left({}^{(k)}p_{n+1},\ {}^{(k+1)}\Delta p, \delta p\right) \\ & \quad -b_2\left({}^{(k)}\mathsf{x}_{n+1}, \delta p\right) \\ & \quad +g\left({}^{(k)}p_{n+1}, \delta p\right) \end{array} \right. \\ \\ {}^{(k+1)}\mathsf{s}_{n+1} = {}^{(k)}\mathsf{s}_{n+1} + {}^{(k+1)}\Delta\mathsf{s}, \\ \\ k+1 \to k, \qquad \text{até que } \left\| {}^{(k+1)}\Delta\mathsf{s} \right\| \leq \varepsilon > 0. \end{array}} \tag{2.98}$$

Da mesma forma que em (2.74), também se aplicam os critérios de convergência (2.75) e (2.76).

Os termos de (2.98) são dados por

$$\begin{aligned} \delta a\left({}^{(k)}\mathsf{s}_{n+1};\ {}^{(k+1)}\Delta\mathsf{x}, \delta\mathsf{x}\right) &= \int_{\mathcal{B}} \mathsf{C}\left({}^{(k)}\mathsf{s}_{n+1}\right) : D\mathbf{E}\left[{}^{(k+1)}\Delta\mathbf{u}\right] \cdot D\mathbf{E}[\delta\mathsf{x}]\ dV \\ &+ \int_{\mathcal{B}} \mathbf{S}\left({}^{(k)}\mathsf{s}_{n+1}\right) \cdot \frac{1}{2}\left[\nabla^{(k+1)}(\Delta\mathbf{u})^T \nabla\delta\mathsf{x} + \nabla\delta\mathsf{x}^T \nabla^{(k+1)}(\Delta\mathbf{u})\right]\ dV, \end{aligned} \tag{2.99}$$

$$\delta b\left({}^{(k)}\mathsf{s}_{n+1}; \delta\mathsf{x},\ {}^{(k+1)}\Delta p\right) = \int_{\mathcal{B}} D\mathbf{E}[\delta\mathsf{x}] \cdot \frac{\partial J}{\partial \mathbf{E}}\ {}^{(k+1)}\Delta p\ dV, \tag{2.100}$$

$$\delta b\left({}^{(k)}\mathsf{s}_{n+1};\ {}^{(k+1)}\Delta\mathsf{x}, \delta p\right) = \int_{\mathcal{B}} \delta p \left\{ D\mathbf{E}\left[{}^{(k+1)}\Delta\mathbf{u}\right] \cdot \frac{\partial J}{\partial \mathbf{E}} \right\} dV, \tag{2.101}$$

$$\delta g\left({}^{(k)}p_{n+1}, {}^{(k+1)}\Delta p, \delta p\right) = \int_{\mathcal{B}} \delta p \left\{ DG^{*\prime}\left({}^{(k)}p_{n+1}\right)\left[{}^{(k+1)}\Delta p\right] \right\}\ dV. \tag{2.102}$$

Como

$$\delta a\left({}^{(k)}\mathsf{s}_{n+1}; \cdot, \cdot\right)$$

é bilinear em $\mathcal{V}$,

$$\delta b\left({}^{(k)}\mathsf{s}_{n+1}; \cdot, \cdot\right)$$

é bilinear em $\mathcal{V} \times \mathcal{Q}$,

$$l_{n+1}(\delta \mathsf{x}) - a\left({}^{(k)}\mathsf{x}_{n+1}, \delta \mathsf{x}\right) - b_1\left(\delta \mathsf{x},\ {}^{(k)}p_{n+1}\right)$$

é linear em $\delta \mathsf{x}$ e

$$\delta g\left({}^{(k)}p_{n+1}, \delta p\right) - b_2\left({}^{(k)}\mathsf{x}_{n+1}, \delta p\right) + g\left({}^{(k)}p_{n+1}, \delta p\right)$$

é linear em δp, o sistema (2.98) é do tipo (2.50).

2.2.9 Método de Projeção da Pressão

Na Seção 2.2.8, o problema não-linear de deslocamentos sujeito a restrição de incompressibilidade foi formulado através de uma forma variacional mista. Entretanto, como observado na Seção 2.2.2, as formas matriciais obtidas são diferentes daquelas decorrentes da forma variacional de único campo, e portanto códigos desenvolvidos para essa situação não podem ser aplicados a formas mistas. Técnicas alternativas foram então propostas com o objetivo de, reproduzindo a precisão da formulação mista, originarem formas matriciais compatíveis com os códigos já existentes, desenvolvidos para problemas decorrentes da formulação clássica de único campo. Nesse contexto, podem ser citadas as técnicas de integração reduzida e os métodos de projeção (Brezzi e Fortin, 1991; Chen e Pan, 1996; Hughes, 1980).

O método de projeção da pressão (Chen e Pan, 1996) é uma generalização para hiperelasticidade não-linear quasi-incompressível do método B-bar (Hughes, 1980; Simo e Hughes, 1998), proposto para elasticidade linear incompressível. Tais técnicas são denominadas *métodos de deformação suposta*, pois assumem um tipo de interpolação *a priori* para os campos de tensão e deformação, independentemente da aproximação adotada para o campo de deslocamentos. A pressão hidrostática é projetada no sentido de mínimos quadrados num espaço funcional previamente escolhido e então reescrita em termos dos deslocamentos.

A técnica de projeção consiste no problema de aproximar uma função escalar quadrado-integrável $p_e(\mathbf{x})$ no domínio de um elemento e no sentido de mínimos quadrados por uma combinação linear de uma seqüência de funções $\{Q_1(\mathbf{x}), Q_2(\mathbf{x}), \ldots, Q_n(\mathbf{x})\}$. Logo, deve-se determinar o vetor $\mathbf{p}_e = [p_1, p_2, \ldots, p_n]^T$ que minimize

$$\|p_e(\mathbf{x}) - \mathbf{Q}\mathbf{p}_e\|^2_{L_2(\mathcal{B}e)}$$

sendo $\|\cdot\|_{L_2(\mathcal{B}e)}$ a norma L_2 no domínio do elemento e e $\mathbf{Q}(\mathbf{x}) = \begin{bmatrix} Q_1(\mathbf{x}) & Q_2(\mathbf{x}) & \ldots & Q_n(\mathbf{x}) \end{bmatrix}$.

No método de projeção da pressão, assume-se a hipótese de 'pressão hidrostática/deformação volumétrica' linear ($\tilde{\kappa}$ constante), o que é razoável em problemas quasi-incompressíveis, pois a deformação volumétrica é muito pequena. 'Projetar a pressão' corresponde a assumir uma expansão polinomial para p dentro de uma zona de integração após a discretização do problema. A seguir, $(2.78)_2$ e $(2.98)_2$ são resolvidas para a variável p em função do deslocamento. Esse resultado é então substituído nas equações precedentes dos respectivos sistemas de equações de modo que o sistema discreto resultante é função apenas do deslocamento.

As condições de Babuska-Brezzi fornecem a base para escolher o espaço funcional de pressão de forma coerente com o espaço funcional de deslocamentos (Brezzi e Fortin, 1991). Uma escolha usual é utilizar elementos com interpolação quadrática de Lagrange do campo de deslocamentos associados a expansão linear da pressão (Chen e Pan, 1996; Chen *et al.*, 1996; Sussman e Bathe, 1987). Nesse caso, tem-se os elementos quadrangulares de 9 nós para deslocamentos com 3 variáveis de pressão, ou seja,

$$\mathbf{Q}(\mathbf{x}) = \begin{bmatrix} 1 & x & y \end{bmatrix}$$

e hexaedros de 27 nós com 4 variáveis de pressão, sendo

$$\mathbf{Q}(\mathbf{x}) = \begin{bmatrix} 1 & x & y & z \end{bmatrix}.$$

Resultados numéricos têm mostrado também que a utilização de interpolação constante dentro de uma zona de integração é satisfatória para aplicações de hiperelasticidade (Kim, 1999; Chen *et al.*, 2000). Por exemplo, os elementos com interpolação quadrática de Serendipty do campo de deslocamentos e pressão constante também respeitam as condições de Babuska-Brezzi. Essa escolha facilita a condensação da pressão dentro de cada elemento.

Aplicando $(2.60)_2$ em um domínio de integração $\mathcal{B}_e \subset \mathcal{B}$ com pressão constante $\hat{p}_e$, obtem-se o seguinte resultado,

$$\int_{\mathcal{B}_e} (J-1)\,\delta p\; dV - \int_{\mathcal{B}_e} G^{*\prime}(\hat{p}_e)\,\delta p\; dV = 0, \qquad \hat{p}_e \text{ constante} \Rightarrow \quad G^{*\prime}(\hat{p}_e)\int_{\mathcal{B}_e} dV = \int_{\mathcal{B}_e} (J-1)\; dV.$$

No caso de material quasi-incompressível,

$$G(J) = \frac{1}{2}(J-1)^2$$

é uma opção razoável[13] para $G(J)$ na expressão (2.54) (Chen *et al.*, 1997a; Kim, 1999). Portanto,

$$G^*(p) = \frac{1}{2}\varepsilon p^2 = \frac{1}{2\tilde{\kappa}}p^2 \Rightarrow G^{*\prime}(p) = \frac{p}{\tilde{\kappa}}.$$

Por fim obtém-se,

$$\boxed{{}^{(k)}\hat{p}_{e_{n+1}} = \frac{\tilde{\kappa}}{\int_{\mathcal{B}_e} dV} \int_{\mathcal{B}_e} \left[J\left({}^{(k)}\mathsf{x}_{n+1}\right) - 1\right] \, dV,} \tag{2.103}$$

que define a condensação da pressão hidrostática no domínio $\mathcal{B}_e$. Este valor pode ser então substituído em (2.98) eliminando-se a variável ${}^{(k)}p_{n+1}$.

Observa-se em (2.103) que o método de projeção é um técnica de suavização usada para eliminar a oscilação pontual do valor de pressão. Coerente, portanto, com a formulação mista, na qual o cálculo de pressão é tratado como uma restrição a ser respeitada em média.

Aplicando-se procedimento semelhante ao sistema linearizado (2.98), resolve-se a segunda equação do sistema para a variável Δp, ou seja,

$$\int_{\mathcal{B}} \delta p \left\{D\mathbf{E}[\Delta\mathbf{u}] \cdot \frac{\partial J}{\partial \mathbf{E}}\right\} dV = \int_{\mathcal{B}} \delta p \left\{DG^{*\prime}(p)[\Delta p]\right\} \, dV - \int_{\mathcal{B}} (J-1)\,\delta p \, dV + \int_{\mathcal{B}} G^{*\prime}(p)\,\delta p \, dV$$

$$\int_{\mathcal{B}} D\mathbf{E}[\Delta\mathbf{u}] \cdot \frac{\partial J}{\partial \mathbf{E}} dV = \int_{\mathcal{B}} DG^{*\prime}(p)[\Delta p] \; - \int_{\mathcal{B}} (J-1) \; dV + \int_{\mathcal{B}} G^{*\prime}(p) \; dV$$

$$\int_{\mathcal{B}} DG^{*\prime}(p)[\Delta p] \; dV = \int_{\mathcal{B}} \left\{D\mathbf{E}[\Delta\mathbf{u}] \cdot \frac{\partial J}{\partial \mathbf{E}} + J - 1 - G^{*\prime}(p)\right\} \, dV$$

Assumindo $\hat{p}_e$ constante e utilizando a mesma expressão anterior para $G^*(p)$, tem-se

$$\boxed{{}^{(k+1)}\Delta\hat{p}_e = \frac{\tilde{\kappa}}{\int_{\mathcal{B}_e} dV} \int_{\mathcal{B}_e} \left\{D\mathbf{E}\left[{}^{(k+1)}\Delta\mathbf{u}\right] \cdot \frac{\partial J}{\partial \mathbf{E}}\left({}^{(k)}\mathsf{x}_{n+1}\right) + J\left({}^{(k)}\mathsf{x}_{n+1}\right) - 1 - \frac{{}^{(k)}\hat{p}_{e_{n+1}}}{\tilde{\kappa}}\right\} \, dV,} \tag{2.104}$$

que é posteriormente substituído na primeira equação do sistema.

2.2.10 Carregamento Dependente da Deformação

Na formulação de Newton-Raphson das seções anteriores considerou-se que o carregamento era independente da deformação, ou seja, este dependia somente da geometria inicial. Formalmente, $D\delta W_{ext}(\mathsf{r}, \delta\mathsf{r})[\Delta\mathbf{u}] = 0$. Entretanto, um caso freqüente de interesse prático consiste de pressão normal a certa região do contorno em todos os momentos do deslocamento.

[13]Vide Seção 2.1.6.

No caso de uma pressão normal $\tilde{p}$ sobre o contorno do corpo, tem-se que

$$l(\mathsf{s}, \delta\mathsf{x}) = \int_{\Gamma 2t} -\tilde{p}\mathbf{m} \cdot \delta\mathsf{x}\, dA, \tag{2.105}$$

sendo $\mathbf{m}$ o vetor normal positivo à superfície espacial (deformada) Γ_t^2. Esse tipo de carregamento pode ser definido em termos da geometria inicial da seguinte forma

$$l(\mathsf{s}, \delta\mathsf{x}) = \int_{\Gamma 2t} -\tilde{p}\mathbf{m} \cdot \delta\mathsf{x}\, dA = \int_{\Gamma 2} -\tilde{p}\,(\det \mathbf{F})\, \mathbf{F}^{-T}\mathbf{n} \cdot \delta\mathsf{x}\, dA, \tag{2.106}$$

sendo $\mathbf{n}$ a normal à superfície material Γ^2.

Como a linearização da expressão anterior (necessária para a aplicação do método de Newton-Raphson) não é um processo trivial, é mais eficiente adotar-se uma representação paramétrica para o contorno e utilizá-la para descrever o vetor normal. Utiliza-se então a discretização da geometria para aproximar a representação paramétrica da superfície Γ^2. Nessa situação de carregamentos,

$$D\delta W_{ext}\left({}^{(k)}\mathsf{s}_{n+1}, \delta\mathsf{s}\right)\left[{}^{(k+1)}\Delta\mathbf{u}\right] = \delta l\left({}^{(k)}\mathsf{s}_{n+1};\ {}^{(k+1)}\Delta\mathbf{u}, \delta\mathsf{x}\right).$$

A equação de (2.74) é então modificada para

$$\delta a\left({}^{(k)}\mathsf{x}_{n+1};\ {}^{(k+1)}\Delta\mathbf{u}, \delta\mathsf{x}\right) - \delta l\left({}^{(k)}\mathsf{x}_{n+1};\ {}^{(k+1)}\Delta\mathbf{u}, \delta\mathsf{x}\right) = l\left({}^{(k)}\mathsf{x}_{n+1}, \delta\mathsf{x}\right) - a\left({}^{(k)}\mathsf{x}_{n+1}, \delta\mathsf{x}\right) \tag{2.107}$$

e a primeira equação do sistema (2.98) para

$$\begin{aligned} \delta a\left({}^{(k)}\mathsf{s}_{n+1};\ {}^{(k+1)}\Delta\mathbf{u}, \delta\mathsf{x}\right) &+ \delta b\left({}^{(k)}\mathsf{s}_{n+1}; \delta\mathsf{x},\ {}^{(k+1)}\Delta p\right) - \delta l\left({}^{(k)}\mathsf{s}_{n+1};\ {}^{(k+1)}\Delta\mathbf{u}, \delta\mathsf{x}\right) = \\ &= l\left({}^{(k)}\mathsf{s}_{n+1}, \delta\mathsf{x}\right) - a\left({}^{(k)}\mathsf{x}_{n+1}, \delta\mathsf{x}\right) - b_1\left(\delta\mathsf{x},\ {}^{(k)}p_{n+1}\right). \end{aligned} \tag{2.108}$$

Domínio Bidimensional

No caso bidimensional, o contorno pode ser descrito por curvas uniparamétricas $r \to \mathbf{x}(r)$. O vetor tangente $\mathbf{t}_r$ é então definido pela derivada parcial do vetor posição $\mathbf{x}$ em relação ao parâmetro r e o vetor normal *negativo*[14] $\mathbf{n}_r$ é obtido a partir de seu produto vetorial com o vetor unitário normal ao plano contendo o domínio $\mathcal{B}$, ou seja,

$$\mathbf{t}_r = \frac{\partial \mathbf{x}}{\partial r} = \mathbf{x}_{,r}, \tag{2.109}$$

$$\mathbf{n}_r = -\mathbf{e}_3 \times \mathbf{t}_r. \tag{2.110}$$

[14] Na notação usual de sinais de componentes de tensão, tensão normal de tração é positiva e de compressão é negativa.

Logo, demonstra-se que o carregamento e sua linearização são escritos em representação paramétrica como (Marsden e Tromba, 1988)

$$l(\mathsf{s}, \delta\mathsf{x}) = \int_{\Gamma 2t} -\tilde{p}\mathbf{m} \cdot \delta\mathsf{x}\, dL = \int_r \tilde{p}\mathbf{n}_r \cdot \delta\mathsf{x}\, dr = \int_r \tilde{p} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \mathbf{x}_{,r} \cdot \delta\mathsf{x}\, dr, \tag{2.111}$$

$$Dl(\mathsf{s}, \delta\mathsf{x})[\Delta\mathbf{u}] = \delta l(\mathsf{x}; \Delta\mathbf{u}, \delta\mathsf{x}) = \int_r \tilde{p} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} D\mathbf{x}_{,r}[\Delta\mathbf{u}] \cdot \delta\mathsf{x}\, dr, \tag{2.112}$$

sendo

$$D\mathbf{x}_{,r}[\Delta\mathbf{u}] = D(\mathbf{X} + \mathbf{u})_{,r}[\Delta\mathbf{u}] = \Delta\mathbf{u}_{,r}.$$

Assim,

$$\boxed{\delta l(\mathsf{s}; \Delta\mathbf{u}, \delta\mathsf{x}) = \int_r \tilde{p} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \Delta\mathbf{u}_{,r} \cdot \delta\mathsf{x}\, dr,} \tag{2.113}$$

sendo os termos $\mathbf{x}_{,r}$ e $\Delta\mathbf{u}_{,r}$ dados por

$$\mathbf{x}_{,r} = \sum_{i=1}^{Nn} \left(\mathbf{X}_e^i + \mathbf{u}_e^i\right) N_{,r}^i, \tag{2.114}$$

$$\Delta\mathbf{u}_{,r} = \sum_{i=1}^{Nn} \Delta\mathbf{u}_e^i N_{,r}^i. \tag{2.115}$$

Observa-se que $N_{,r}^i$ é calculado na restrição da função N^i a $\partial\mathcal{B}_e \cap \Gamma^2 \equiv \Gamma_e^2$.

Domínio Tridimensional

A parametrização do contorno de um domínio tridimensional requer um par de coordenadas, por exemplo (r, s). Portanto, as tangentes unitárias à superfície no ponto (r, s) são

$$\mathbf{t}_r = \frac{\partial\mathbf{x}}{\partial r} = \mathbf{x}_{,r}, \tag{2.116}$$

$$\mathbf{t}_s = \frac{\partial\mathbf{x}}{\partial s} = \mathbf{x}_{,s}. \tag{2.117}$$

Assumindo que a parametrização é positivamente orientada,

$$\mathbf{n}_{rs} = -\mathbf{t}_r \times \mathbf{t}_s \tag{2.118}$$

define a normal paramétrica negativa no ponto (r, s).

A partir daí tem-se que,

$$\begin{aligned} l\,(\mathsf{r}, \delta\mathsf{x}) &= \int_{\Gamma 2t} -\tilde{p}\mathbf{m} \cdot \delta\mathsf{x}\; dA = \int_r \int_s \tilde{p}\mathbf{n}_{rs} \cdot \delta\mathsf{x}\; drds = -\int_r \int_s \tilde{p}\,(\mathbf{t}_r \times \mathbf{t}_s) \cdot \delta\mathsf{x}\; drds \\ &= -\int_r \int_s \tilde{p}\,(\mathbf{x}_{,r} \times \mathbf{x}_{,s}) \cdot \delta\mathsf{x}\; drds. \end{aligned} \tag{2.119}$$

Analogamente ao caso bidimensional,

$$D\mathbf{x}_{,r}\,[\Delta\mathbf{u}] = \Delta\mathbf{u}_{,r}, \qquad D\mathbf{x}_{,s}\,[\Delta\mathbf{u}] = \Delta\mathbf{u}_{,s},$$

$$\begin{aligned} D\,[(\mathbf{x}_{,r} \times \mathbf{x}_{,s})]\,[\Delta\mathbf{u}] &= D\mathbf{x}_{,r}\,[\Delta\mathbf{u}] \times \mathbf{x}_{,s} + \mathbf{x}_{,r} \times D\mathbf{x}_{,s}\,[\Delta\mathbf{u}] = \Delta\mathbf{u}_{,r} \times \mathbf{x}_{,s} + \mathbf{x}_{,r} \times \Delta\mathbf{u}_{,s} \\ &= -\mathbf{x}_{,s} \times \Delta\mathbf{u}_{,r} + \mathbf{x}_{,r} \times \Delta\mathbf{u}_{,s} = -\operatorname{Sk}[\mathbf{x}_{,s}]\,\Delta\mathbf{u}_{,r} + \operatorname{Sk}[\mathbf{x}_{,r}]\,\Delta\mathbf{u}_{,s}. \end{aligned}$$

O operador $\operatorname{Sk}[\cdot]$ associa a cada vetor $\mathbf{a} \in \Re^3$ um tensor antissimétrico

$$\operatorname{Sk}[\mathbf{a}] = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}.$$

Conclui-se então que a forma linearizada de (2.119) é

$$\boxed{\delta l\,(\mathsf{s}; \Delta\mathbf{u}, \delta\mathsf{x}) = \int_r \int_s \tilde{p}\operatorname{Sk}[\mathbf{x}_{,s}]\,\Delta\mathbf{u}_{,r} \cdot \delta\mathsf{x}\; drds - \int_r \int_s \tilde{p}\operatorname{Sk}[\mathbf{x}_{,r}]\,\Delta\mathbf{u}_{,s} \cdot \delta\mathsf{x}\; drds.} \tag{2.120}$$

2.3 Estudo de Casos de Análise de Resposta

2.3.1 Grandes Deslocamentos com Material Linear

Uma mola de aço com diâmetro do arame de 10 mm e hélice definida pela expressão vetorial $\mathbf{x} = \begin{bmatrix} 60\cos t & 60\operatorname{sen} t & 10t \end{bmatrix}^T$, $0 \leq t \leq 12$ rad é discretizada em tetraedros, como mostrado na Figura 2.1. A mola é engastada na sua extremidade inferior e aplica-se uma força vertical de -0.375 kN na extremidade superior. São utilizados elementos quadráticos para cinemática finita e modelo de material elástico linear.

O modelo apresenta 6231 nós, 2994 elementos e 18636 graus de liberdade, sendo resolvido em 5 passos de carregamento e um total de 34 iterações de Newton-Raphson. Em todas as iterações, a matriz de rigidez é completamente reconstruída (Newton-Raphson completo), não há busca linear e utiliza-se o critério de convergência (2.75).

Na Figura 2.2, tem-se um deslocamento máximo de 31.576 mm, correspondendo a uma tensão equivalente máxima de 0.74343 kN/mm^2.

A diferença entre a aplicação das cinemáticas finita e infinitesimal neste problema pode ser observado nos gráficos das Figuras 2.3(a) e 2.3(b), nas quais comparam-se os valores máximos de deslocamento vertical e a tensão equivalente de von Mises para uma força vertical de até -0.750 kN.

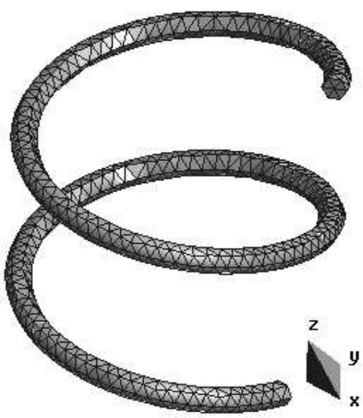


Figura 2.1: Geometria não-deformada ($E = 207 \times 10^3$ kN/mm^2, $\nu = 0.3$, $\rho = 7.81 \times 10^{-6}$kg/mm^3).

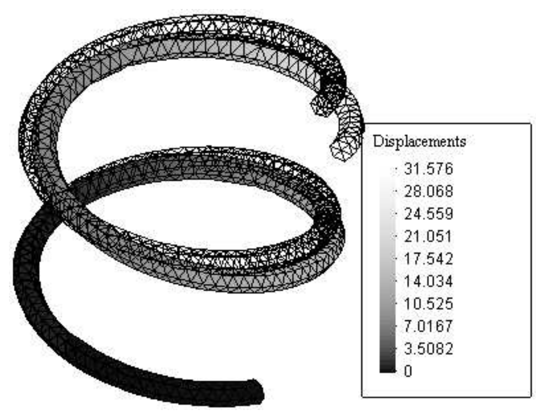
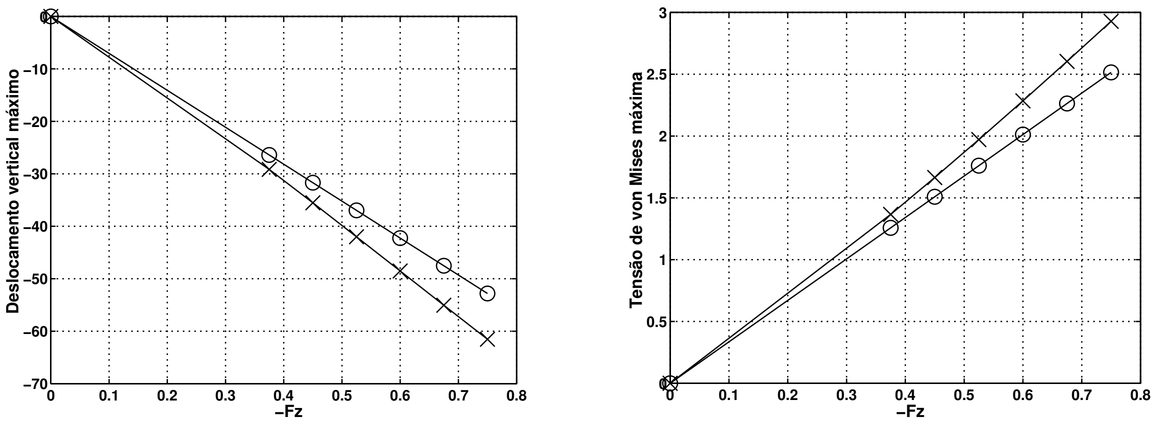


Figura 2.2: Configuração final em escala 1:1 [mm].

Neste problema, uma parcela significativa do campo de deslocamentos decorre das rotações finitas dos pontos materiais, o que não é contabilizado pela cinemática linear. Dessa forma, a solução linear fornece valores inferiores de deslocamentos e tensões e portanto não permite prever com precisão o comportamento e a durabilidade dos sistemas com cinemática finita.

2.3.2 Material Hiperelástico Quasi-Incompressível

Os exemplos numéricos envolvendo material hiperelástico incompressível foram retirados da referência (Pantuso e Bathe, 1997). Foi utilizada interpolação quadrática (de Serendipty) do campo de deslocamentos associada a interpolação constante da pressão hidrostática em cada elemento. Portanto, na notação u/p, tem-se elementos 8/1, 6/1, 20/1 e 10/1 para elementos quadrangulares, triangulares,



(a) (b)

Figura 2.3: Soluções do problema da mola com cinemáticas finita (−×−) e infinitesimal (−○−) em relação à força aplicada.

hexaédricos e tetraédricos respectivamente. Todos os elementos apresentam cinemática finita e aplica-se integração numérica consistente nos cálculos elementares.

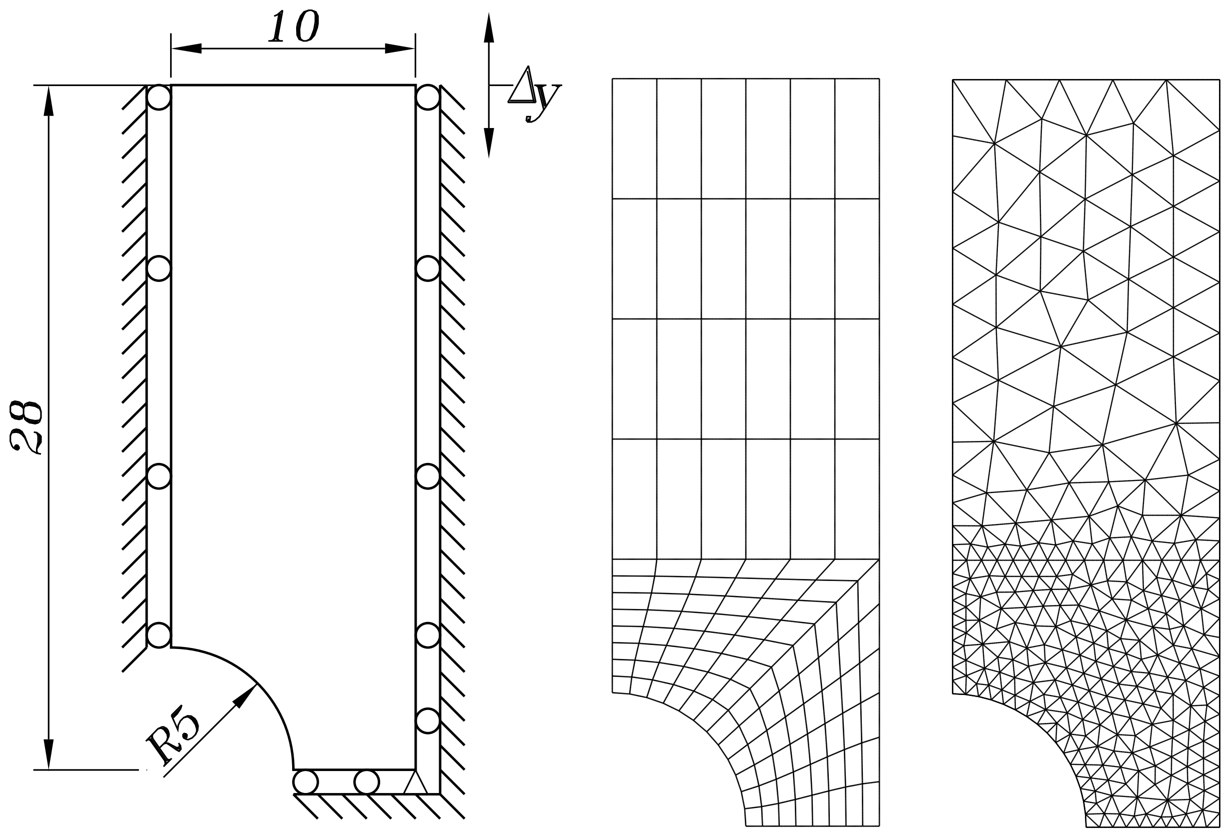
Nas iterações de Newton-Raphson (completo) com busca linear, os sistemas lineares foram resolvidos por eliminação de Gauss. Pré-condicionadores do método de gradiente conjugado que se mostraram eficientes em problemas elásticos lineares, como o Gauss-Seidel simétrico (Bittencourt e Feijóo, 1996), apresentaram performance bastante baixa em problemas de hiperelasticidade incompressível devido ao mal-condicionamento das matrizes de rigidez tangente obtidas desses problemas (Adams, 2000a). Além disso, no decorrer do processo de solução, o condicionamento piora com a compressão dos elementos.

Placa com furo

Uma placa com furo central nas condições da Figura 2.4(a) é sujeita a deslocamentos de compressão e tração prescritos em sua face superior.

As Figuras 2.5 e 2.6 mostram os resultados em termos do deslocamento vertical para a discretização em elementos quadrangulares (Figura 2.4(b)), enquanto que os resultados das Figuras 2.7 e 2.8 foram obtidos com elementos triangulares (Figura 2.4(c)). Em ambos os casos, o comportamento é semelhante, apenas observando-se rigidez ligeiramente maior na discretização em triângulos.

As curvas de força por deslocamento em compressão concordam com os dados apresentados na

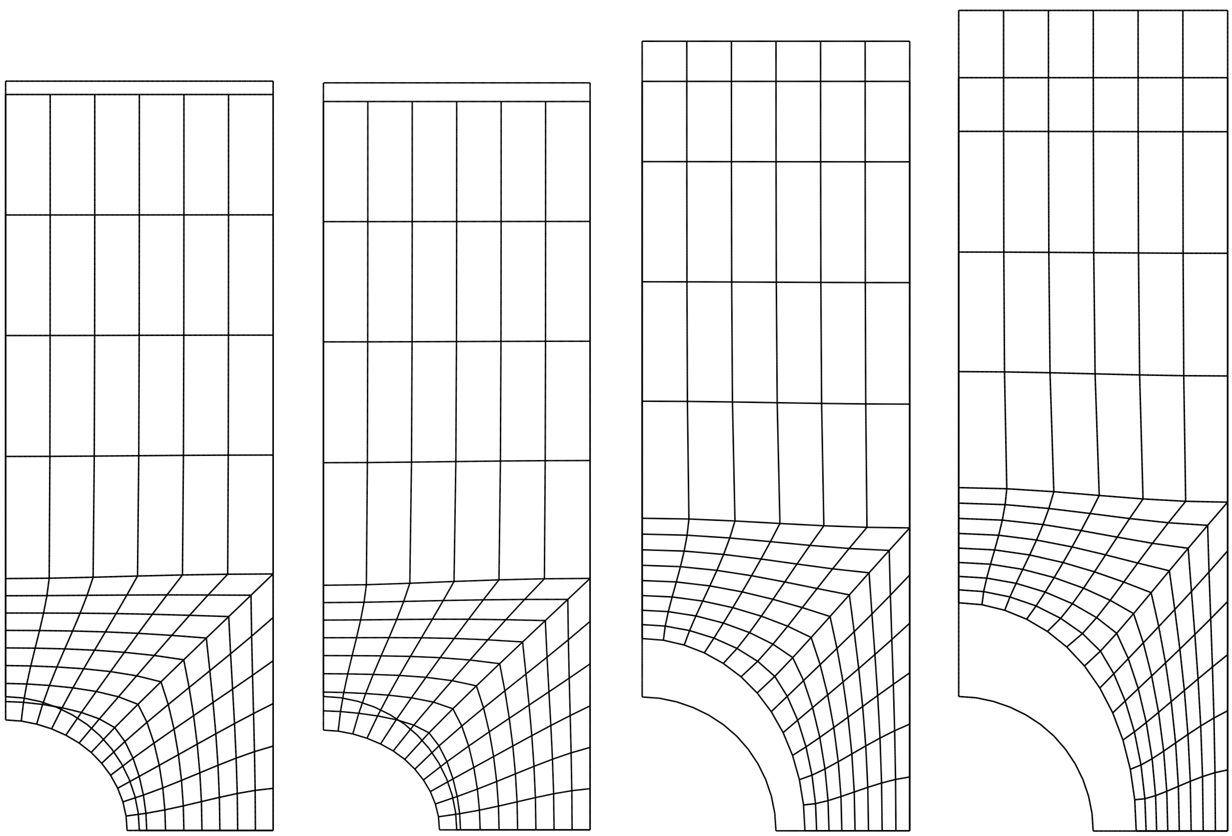


(a) Geometria e condições de contorno.

(b) Elementos quadrangulares.

(c) Elementos triangulares.

Figura 2.4: Placa com furo em modelo de deformação plana. Material Mooney-Rivlin: $A_{10} = 1.777$, $A_{01} = 0.045$, $\tilde{\kappa} = 666.66$.



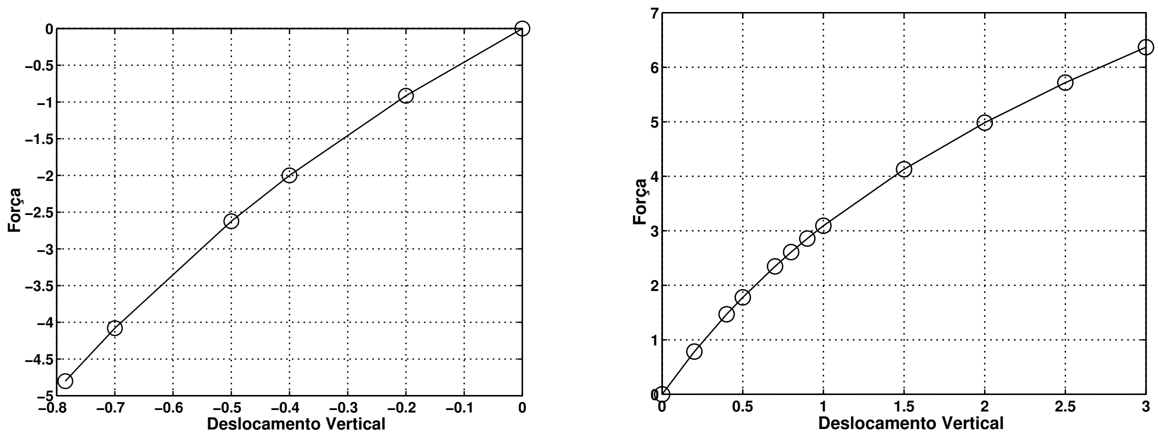
(a)
$\Delta y = -0.50000.$
$ux_{max} = -0.45807.$
$uy_{max} = -0.87074.$
$\sigma_{VMmax} = 0.29736.$

(b)
$\Delta y = -0.70000.$
$ux_{max} = -0.64817.$
$uy_{max} = -1.26100.$
$\sigma_{VMmax} = 0.45583.$

(c)
$\Delta y = 1.50000.$
$ux_{max} = 1.08379.$
$uy_{max} = 2.18389.$
$\sigma_{VMmax} = 0.64615.$

(d)
$\Delta y = 2.50000.$
$ux_{max} = 1.60064.$
$uy_{max} = 3.48273.$
$\sigma_{VMmax} = 1.05676.$

Figura 2.5: Elementos quadrangulares: deformações de compressão e tração (escala 1:1).



(a) Compressão. **(b)** Tração.

Figura 2.6: Curvas força×deslocamento da placa com furo em elementos quadrangulares.

referência (Pantuso e Bathe, 1997) até o deslocamento de -0.785. A partir daí são observados modos espúrios de deformação em torno do furo, o que é esperado na solução deste problema através de elementos com interpolação constante da pressão hidrostática sob compressão.

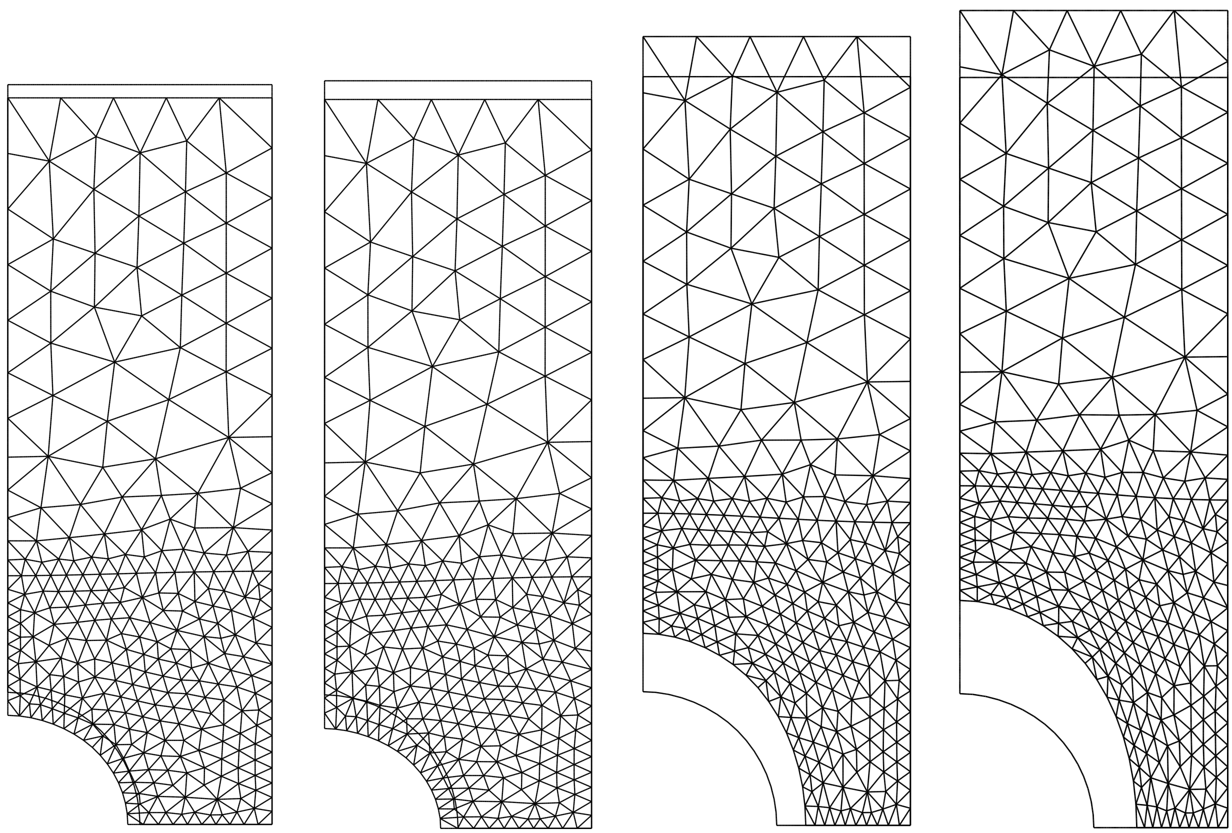
Cilindro de borracha

Numa aplicação, tem-se um eixo de material metálico montado no interior de um cilindro de borracha, o qual tem sua superfície externa engastada, como mostrado na Figura 2.9. As propriedades são tais que o eixo apresenta rigidez muito superior à borracha, correspondendo, na prática, a um corpo rígido.

Ao se impor deslocamentos verticais ao eixo metálico, obtem-se a curva de força por deslocamento da Figura 2.10, observando-se as deformações da Figura 2.11. Até o nível de deformação considerado, a relação entre força e deslocamento reproduz os resultados da referência (Pantuso e Bathe, 1997). A solução é obtida em 5 passos de carregamento.

2.4 Conclusões e Análise dos Resultados

Os resultados obtidos mostram que a formulação apresentada neste capítulo permitem a solução de problemas envolvendo hiperelasticidade não-linear quasi-incompressível de forma satisfatória, repro-



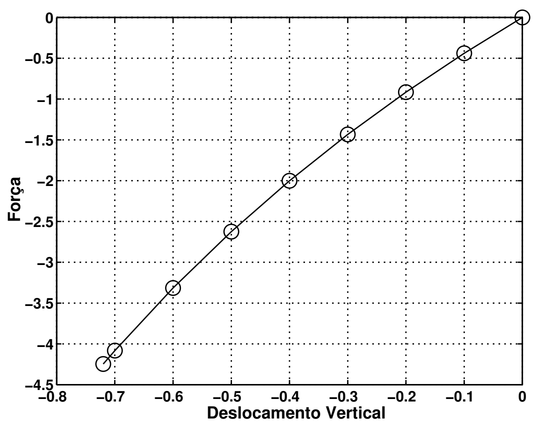
(a)
$\Delta y = -0.50000.$
$ux_{max} = -0.46033.$
$uy_{max} = -0.86988.$
$\sigma_{VMmax} = 0.29787.$

(b)
$\Delta y = -0.70000.$
$ux_{max} = -0.65280.$
$uy_{max} = -1.25980.$
$\sigma_{VMmax} = 0.46237.$

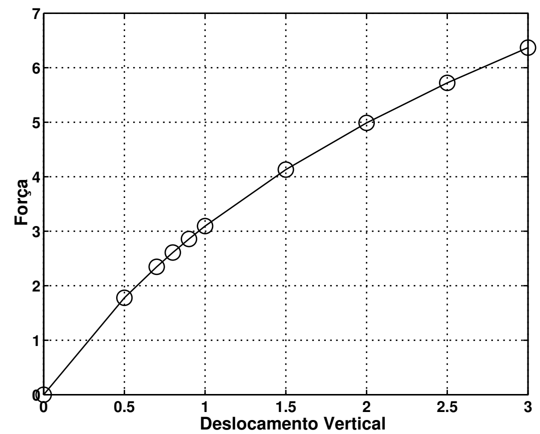
(c)
$\Delta y = 1.50000.$
$ux_{max} = 1.08652.$
$uy_{max} = 2.18246.$
$\sigma_{VMmax} = 0.65510.$

(d)
$\Delta y = 2.50000.$
$ux_{max} = 1.60484.$
$uy_{max} = 3.48076.$
$\sigma_{VMmax} = 1.07552.$

Figura 2.7: Elementos triangulares: deformações de compressão e tração (escala 1:1).



(a) Compressão.



(b) Tração.

Figura 2.8: Curvas força×deslocamento da placa com furo em elementos triangulares.

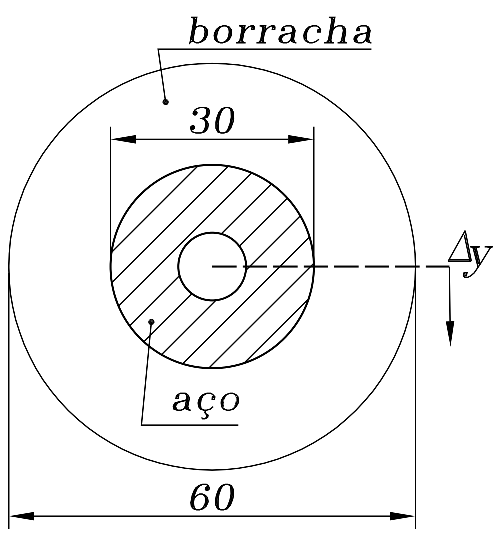


Figura 2.9: Coxim cilíndrico em modelo de deformação plana. Coxim de borracha: material Mooney-Rivlin, $A_{10} = 1.777$, $A_{01} = 0.045$, $\tilde{\kappa} = 666.66$. Eixo de aço: $E = 2.1 \times 10^{11}$, $\nu = 0.3$.

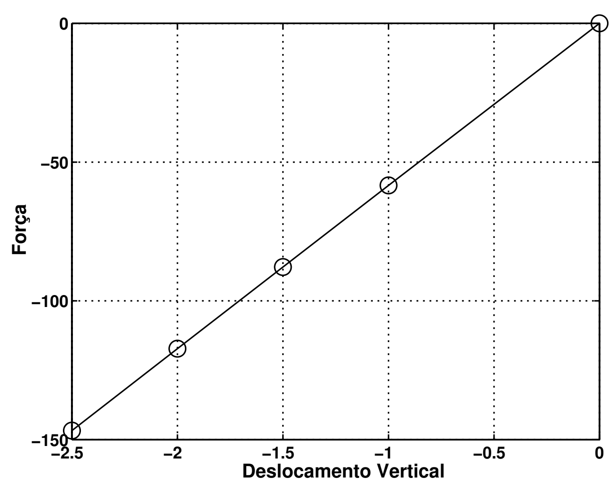
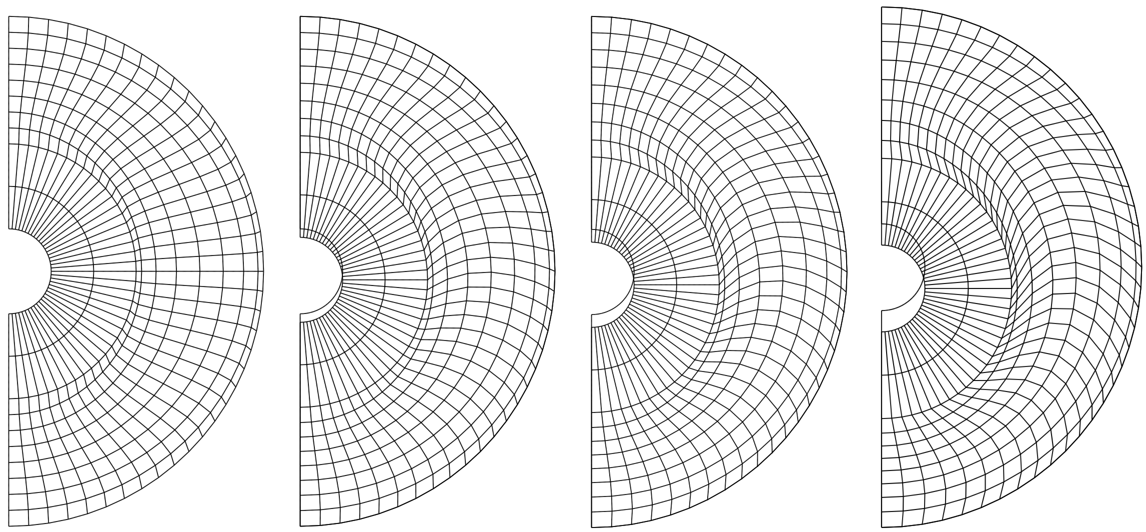


Figura 2.10: Coxim cilíndrico: curva força× deslocamento.



(a) Inicial. **(b)** $\Delta y = -1.0$. **(c)** $\Delta y = -1.5$. **(d)** $\Delta y = -2.5$.

Figura 2.11: Seqüência de deformações (escala 1:1).

duzindo comportamentos observados na literatura.

Capítulo 3

Análise de Sensibilidade para Não-Linearidades Geométricas e Hiperelasticidade

Apresenta-se, neste capítulo, a formulação contínua de análise de sensibilidade de problemas estruturais não-lineares conservativos devido a alteração de parâmetros discretos e modificação da forma. Através dessa técnica, determina-se a sensibilidade do campo de deslocamentos necessária para avaliação das expressões de sensibilidade de funcionais de performance em problemas de otimização de parâmetros discretos e forma.

Como conceito resultante dessa formulação, tem-se que o cálculo de sensibilidade é sempre linear, mesmo para sistemas não-lineares (Arora e Cardoso, 1992; Choi e Santos, 1987; Kim, 1999; Ryu *et al.*, 1985). Portanto, o mesmo é efetuado em passo único, ao contrário do processo iterativo de análise, tornando as características de eficiência da formulação de análise de sensibilidade muito mais evidentes em problemas não-lineares que nos lineares.

3.1 Sensibilidade de Funcionais a Parâmetros Discretos

Considerando um problema estrutural não-linear, supõe-se que sua solução seja obtida de forma incremental. Logo, seja s_{n+1} a configuração de equilíbrio do sistema mecânico ao final do passo de carregamento $I_{n+1} =]t_n, t_{n+1}]$, determinada através do algoritmo do tipo Newton-Raphson (2.74), ou seja, o problema de análise de resposta está resolvido até o instante t_{n+1}.

Durante o projeto de um sistema, definem-se funcionais de performance de modo a julgar objetivamente o comportamento do sistema em análise. No caso de sistemas estruturais, uma forma razoavelmente geral de funcional é (Silva e Bittencourt, 1999b)

$$\boxed{\psi(\mathbf{d}) = \int_{\mathcal{B}} \mathcal{G}(\mathbf{u}_{n+1}, \nabla\mathbf{u}_{n+1}, \mathbf{d})\ dV, \qquad \mathsf{x}_{n+1}(\mathbf{X}, \mathbf{d}) = \mathbf{X} + \mathbf{u}_{n+1}(\mathbf{X}, \mathbf{d}),} \tag{3.1}$$

sendo $\mathbf{d} \in \Re^{Nd}$ um conjunto de variáveis que parametrizam características *discretas* do sistema, ou seja, o domínio $\mathcal{B}$ permanece constante.

A informação mais útil para se efetuar melhoramentos no projeto (tanto interativamente quanto dentro de um algoritmo de otimização), entretanto, é a medida da sensibilidade do funcional de performance a perturbações nos parâmetros de projeto. Em outras palavras, busca-se a variação de primeira ordem do funcional em relação a $\mathbf{d}$, ou seja,

$$\dot{\psi}(\mathbf{d}) \equiv D\psi(\mathbf{d})[\delta\mathbf{d}] = D\left\{\int_{\mathcal{B}} \mathcal{G}\ dV\right\}[\delta\mathbf{d}] = \nabla_{\mathbf{d}}\psi \cdot \delta\mathbf{d}.$$

Como o domínio é fixo, pode-se comutar a operação de diferenciação com a integração. Portanto, aplicando a regra da cadeia (Gurtin, 1981), tem-se

$$\boxed{\dot{\psi}(\mathbf{d}) = \int_{\mathcal{B}} D\mathcal{G}[\delta\mathbf{d}]\ dV = \int_{\mathcal{B}} \left(\mathcal{G}_{,\mathbf{u}} \cdot \dot{\mathbf{u}}_{n+1} + \mathcal{G}_{,\nabla\mathbf{u}} \cdot \nabla\dot{\mathbf{u}}_{n+1} + \mathcal{G}_{,\mathbf{d}} \cdot \delta\mathbf{d}\right)\ dV} \tag{3.2}$$

como expressão de sensibilidade do funcional, a qual depende de $\dot{\mathbf{u}}_{n+1}$, variação da resposta estrutural $\mathbf{u}_{n+1}$ no instante t_{n+1} obtida da sensibilidade da equação variacional de equilíbrio do sistema.

3.2 Alteração de Forma do Domínio

Considere agora que o domínio material $\mathcal{B}$ é modificado segundo a transformação geométrica

$$\mathsf{T}_\tau : \mathcal{B} \times \Re \rightarrow \mathcal{B}_\tau \subset \mathcal{E}, \tag{3.3}$$

dando origem a um *segundo domínio material*[1] $\mathcal{B}_\tau$, ou seja,

$$\mathsf{T}_\tau(\mathbf{X},\tau) = \mathbf{X}_\tau, \qquad \mathsf{T}(\mathbf{X}_\tau,\tau) = \mathbf{X}, \tag{3.4}$$

$$\mathsf{x}_\tau(\mathbf{X}_\tau) = \mathbf{x}_\tau = \mathbf{X}_\tau + \mathbf{u}_\tau, \qquad \mathbf{u}_\tau \equiv \mathbf{u}_\tau(\mathbf{X}_\tau,\tau). \tag{3.5}$$

Essa transformação de domínio define uma *modificação de projeto*, ou seja, alterações na geometria inicial $\mathcal{B}$ visando atingir algum objetivo funcional (Silva e Bittencourt, 1999b).

A expansão em série de Taylor da transformação (3.3) em torno de $\mathcal{B}$ (ou seja, $\tau = 0$) pode ser escrita como

$$\mathsf{T}_\tau(\mathbf{X},\tau) = \mathsf{T}_\tau(\mathbf{X},0) + \tau\frac{\partial}{\partial\tau}\mathsf{T}_\tau(\mathbf{X},0) + o(\tau^n), \qquad n = 2,3,\ldots \tag{3.6}$$

A partir da definição usual de campo espacial de velocidades

$$\mathbf{V}_s(\mathbf{X}_\tau,\tau) = \frac{\partial}{\partial\tau}\mathsf{T}_\tau(\mathsf{T}(\mathbf{X}_\tau,\tau),\tau) = \dot{\mathsf{T}}_\tau(\mathsf{T}(\mathbf{X}_\tau,\tau),\tau), \tag{3.7}$$

obtem-se o *campo de velocidades de projeto*, que é o campo de velocidades em torno da geometria de referência $\mathcal{B}$, ou seja,

$$\mathbf{V}(\mathbf{X}) = \mathbf{V}_s(\mathbf{X}_\tau,\tau)\big|_{\tau=0} = \frac{\partial\mathbf{X}}{\partial\tau}. \tag{3.8}$$

Portanto, a expansão em séries em torno de $\mathcal{B}$ desprezando os termos de maior ordem é

$$\mathbf{X}_\tau = \mathsf{T}_\tau(\mathbf{X},\tau) = \mathbf{X} + \tau\mathbf{V}(\mathbf{X}), \qquad \mathbf{X}\in\mathcal{B}. \tag{3.9}$$

Em outras palavras, o campo de velocidades $\mathbf{V}$ descreve como a geometria varia numa aproximação de primeira ordem.

A forma mais simples de descrever exatamente alterações de domínio para qualquer valor do parâmetro é adotar transformações $\mathsf{T}_\tau(\mathbf{X},\tau)$ lineares, ou seja, tais que $o(\tau^n) = 0$.

A partir da definição original de gradiente da alteração de configuração

$$\mathbf{F} = \nabla\mathsf{x} = \mathbf{I} + \nabla\mathbf{u}, \qquad \nabla\mathbf{u}(\mathbf{X}) = \frac{\partial u_i}{\partial X_j},$$

[1] Fala-se em 'segundo domínio material' pois a modificação do domíno não é causada pelas forças externas do problema físico sob análise. Ao contrário, o novo domínio obtido será aplicado ao mesmo problema físico original.

tem-se o gradiente da alteração de configuração x_τ como

$$\mathbf{F}_\tau = \nabla_\tau \mathsf{x}_\tau = \mathbf{I} + \nabla_\tau \mathbf{u}_\tau, \qquad \nabla_\tau \mathbf{u}_\tau \left(\mathbf{X}_\tau\right) = \frac{\partial u_{\tau i}}{\partial X_{\tau j}}, \tag{3.10}$$

observando-se que $\nabla \mathbf{u} \equiv \nabla_\tau \mathbf{u}_\tau|_{\tau=0}$.

O mesmo formalismo desenvolvido para movimentos (A.21) é aplicado aqui, uma vez que a estrutura matemática de modificação da geometria é a mesma, apenas que neste caso não são ações externas as causas das mudanças de configuração. Portanto, a derivada total da alteração de configuração pode ser obtida diretamente da definição (3.5), ou seja,

$$\dot{\mathsf{x}}_\tau = \dot{\mathbf{x}}_\tau = \dot{\mathbf{X}}_\tau + \dot{\mathbf{u}}_\tau, \qquad \dot{\mathsf{x}} = \dot{\mathbf{x}} = \mathbf{V} + \dot{\mathbf{u}}, \quad \tau = 0, \tag{3.11}$$

ou da expressão de derivada total (Gurtin, 1981)

$$\dot{\mathsf{x}} = \mathsf{x}' + \nabla \mathsf{x} \mathbf{V}, \qquad \mathsf{x}' = \mathbf{u}', \tag{3.12}$$

enfatizando-se o significado da notação

$$\boxed{\dot{\mathsf{x}} = \frac{d}{d\tau}\mathsf{x}, \qquad \mathsf{x}' = \frac{\partial}{\partial \tau}\mathsf{x},}$$

sendo que $(\dot{\ })$ e $(')$ indicam, respectivamente, derivada total e parcial em relação a τ.

Da mesma forma,

$$\dot{\mathbf{u}} = \mathbf{u}' + \nabla \mathbf{u} \mathbf{V}. \tag{3.13}$$

Derivando-se $\dot{\mathsf{x}}_\tau$ e $\dot{\mathbf{u}}_\tau$ em relação ao espaço, aplicando (3.12) e fazendo $\tau \to 0$ obtem-se

$$\nabla \dot{\mathsf{x}} = \nabla \mathsf{x}' + \nabla \left(\nabla \mathsf{x} \mathbf{V}\right) = \nabla \mathsf{x}' + \nabla \mathsf{x} \nabla \mathbf{V} + \nabla \left(\nabla \mathsf{x}\right) \mathbf{V}, \tag{3.14}$$

$$\nabla \dot{\mathbf{u}} = \nabla \mathbf{u}' + \nabla \left(\nabla \mathbf{u} \mathbf{V}\right) = \nabla \mathbf{u}' + \nabla \mathbf{u} \nabla \mathbf{V} + \nabla \left(\nabla \mathbf{u}\right) \mathbf{V}. \tag{3.15}$$

Observando-se que as derivadas *parciais* são comutativas, conclui-se que

$$\nabla \mathsf{x}' = \nabla \mathbf{u}'. \tag{3.16}$$

É importante enfatizar que a derivada total *não* comuta com derivadas parciais. Logo, $\nabla \dot{\mathsf{x}} \neq (\nabla \mathsf{x})^{\cdot}$.

Observando que

$$(\nabla \mathbf{u})^{\cdot} = (\nabla \mathbf{u})' + \nabla \left(\nabla \mathbf{u}\right) \mathbf{V} = \nabla \mathbf{u}' + \nabla \left(\nabla \mathbf{u}\right) \mathbf{V} \tag{3.17}$$

e que, a partir de (3.15),

$$\nabla \mathbf{u}' + \nabla (\nabla \mathbf{u}) \mathbf{V} = \nabla \dot{\mathbf{u}} - \nabla \mathbf{u} \nabla \mathbf{V}, \tag{3.18}$$

obtem-se que (Haug *et al.*, 1986)

$$\boxed{\dot{\mathbf{F}} = (\nabla \mathsf{x})^{\cdot} = (\mathbf{I} + \nabla \mathbf{u})^{\cdot} = (\nabla \mathbf{u})^{\cdot} = \nabla \dot{\mathbf{u}} - \nabla \mathbf{u} \nabla \mathbf{V}.} \tag{3.19}$$

Ambas as expressões (3.14) ou (3.19) podem ser usadas no desenvolvimento de expressões de análise de sensibilidade. Contudo a expressão final de sensibilidade será a mesma.

A expressão x_τ pode ser escrita em termos do domínio original à semelhança da descrição material de funções, ou seja,

$$\mathsf{x}_{\tau m} \equiv \mathsf{x}_\tau \left(\mathsf{T}_\tau \left(\mathbf{X}, \tau \right), \tau \right).$$

E da mesma forma tem-se a descrição material de seu gradiente em $\mathcal{B}_\tau$ e as derivadas associadas ao gradiente da transformação $\mathsf{T}_\tau : \mathcal{B} \to \mathcal{B}_\tau$:

$$\nabla \mathsf{x}_{\tau m} = \left(\nabla_\tau \mathsf{x}_\tau \right)_m \frac{\partial \mathbf{X}_\tau}{\partial \mathbf{X}} = \left(\nabla_\tau \mathsf{x}_\tau \right)_m \mathsf{F}, \qquad \mathsf{F} = \frac{\partial \mathbf{X}_\tau}{\partial \mathbf{X}}, \qquad \left(\nabla_\tau \mathsf{x}_\tau \right)_m = \nabla \mathsf{x}_{\tau m} \mathsf{F}^{-1}, \tag{3.20}$$

$$\dot{\mathsf{F}} = \left(\nabla_\tau \mathbf{V}_s \right)_m \mathsf{F}, \tag{3.21}$$

$$(\det \mathsf{F})^{\cdot} = \det \mathsf{F} \left(\operatorname{div} \mathbf{V}_s \right)_m. \tag{3.22}$$

3.3 Sensibilidade de Funcionais Definidos num Domínio Variável

Seja um funcional ψ_τ, definido na configuração $\mathcal{B}_\tau$, com a seguinte expressão genérica:

$$\psi_\tau = \int_{\mathcal{B}_\tau} \mathcal{G} \left(\mathbf{u}_\tau, \nabla_\tau \mathbf{u}_\tau, \mathbf{X}_\tau \right) \, dV, \tag{3.23}$$

sendo $\mathsf{x}_\tau = \mathbf{X}_\tau + \mathbf{u}_\tau$ a configuração de equilíbrio do sistema mecânico de domínio $\mathcal{B}_\tau$ que foi determinada através dos algoritmos (2.74) ou (2.98).

Altera-se o domínio de integração $\mathcal{B}_\tau$ para $\mathcal{B}$ da seguinte forma:

$$\psi_\tau = \int_{\mathcal{B}_\tau} \mathcal{G} \left(\mathbf{u}_\tau, \nabla_\tau \mathbf{u}_\tau, \mathbf{X}_\tau \right) \, dV = \int_{\mathcal{B}} \mathcal{G}_m \left(\mathbf{u}_{\tau m}, \nabla \mathbf{u}_{\tau m}, \mathbf{X} \right) \det \mathsf{F} \; dV. \tag{3.24}$$

Logo, a derivada total de (3.24) em relação a τ, em $\tau = 0$, é (Haug *et al.*, 1986; Santos e Choi, 1992; Silva e Bittencourt, 1999b)

$$\dot{\psi} = \int_{\mathcal{B}} \left(\dot{\mathcal{G}} + \mathcal{G} \operatorname{Div} \mathbf{V} \right) \, dV,$$

com

$$\dot{\mathcal{G}} = \mathcal{G}_{,\mathbf{u}} \cdot \dot{\mathbf{u}} + \mathcal{G}_{,\nabla\mathbf{u}} \cdot (\nabla\mathbf{u})^{\cdot} = \mathcal{G}_{,\mathbf{u}} \cdot \dot{\mathbf{u}} + \mathcal{G}_{,\nabla\mathbf{u}} \cdot \nabla\dot{\mathbf{u}} - \mathcal{G}_{,\nabla\mathbf{u}} \cdot \nabla\mathbf{u}\nabla\mathbf{V} + \mathbf{V} \cdot \nabla\mathcal{G},$$

sendo que $\nabla\mathcal{G}$ *não envolve* dependência implícita através de $\mathbf{u}$. Dessa forma,

$$\dot{\psi} = \int_{\mathcal{B}} \left(\mathcal{G}_{,\mathbf{u}} \cdot \dot{\mathbf{u}} + \mathcal{G}_{,\nabla\mathbf{u}} \cdot \nabla\dot{\mathbf{u}} - \mathcal{G}_{,\nabla\mathbf{u}} \cdot \nabla\mathbf{u}\nabla\mathbf{V} \right) \, dV + \int_{\mathcal{B}} \left(\mathcal{G} \operatorname{Div} \mathbf{V} + \mathbf{V} \cdot \nabla\mathcal{G} \right) \, dV, \tag{3.25}$$

Portanto,

$$\boxed{\dot{\psi} = \int_{\mathcal{B}} \left(\mathcal{G}_{,\mathbf{u}} \cdot \dot{\mathbf{u}}_{n+1} + \mathcal{G}_{,\nabla\mathbf{u}} \cdot \nabla\dot{\mathbf{u}}_{n+1} \right) \, dV + \int_{\mathcal{B}} \left(\mathcal{G} \operatorname{Div} \mathbf{V} + \mathbf{V} \cdot \nabla\mathcal{G} - \mathcal{G}_{,\nabla\mathbf{u}} \cdot \nabla\mathbf{u}_{n+1}\nabla\mathbf{V} \right) \, dV} \tag{3.26}$$

Novamente, o enunciado da sensibilidade do funcional envolve os termos $\dot{\mathbf{u}}_{n+1}$ e $\nabla\dot{\mathbf{u}}_{n+1}$, obtidos da sensibilidade da equação variacional do problema em relação à variação do domínio $\mathcal{B}$.

Tanto a expressão (3.26) quanto a sensibilidade da equação de equilíbrio dependem intensivamente do campo de velocidades $\mathbf{V}$, o qual somente é conhecido univocamente no contorno em grande parte das situações. Porém, uma vez respeitados os requisitos teóricos dessa grandeza descritos por (Choi e Chang, 1994), garante-se a correção no uso de (3.26).

3.4 Análise de Sensibilidade em Hiperelasticidade sem Restrições

3.4.1 Análise de Sensibilidade a Parâmetros Discretos

Considere o corpo em sua configuração de equilíbrio x_{n+1} e sua equação de equilíbrio variacional

$$a_{\mathbf{d}} \left(\mathsf{x}_{n+1}, \delta\mathsf{x} \right) = l_{\mathbf{d}n+1} \left(\delta\mathsf{x} \right), \qquad \mathsf{x}_{n+1} \in \mathcal{X}_{n+1}, \forall \delta\mathsf{x} \in \mathcal{V}. \tag{3.27}$$

A expressão (3.27) é a mesma discutida anteriormente, sendo o subscrito $\mathbf{d}$ apenas uma indicação explícita da dependência do sistema em relação às variáveis de projeto, ou seja,

$$a_{\mathbf{d}} \left(\mathsf{x}_{n+1}, \delta\mathsf{x} \right) \equiv a \left(\mathbf{d}; \mathsf{x}_{n+1} \left(\mathbf{d} \right), \delta\mathsf{x} \right),$$

$$l_{\mathbf{d}n+1} \left(\delta\mathsf{x} \right) \equiv l_{n+1} \left(\mathbf{d}; \delta\mathsf{x} \right).$$

Os deslocamentos virtuais $\delta \mathsf{x}$, por sua vez, são independentes de $\mathbf{d}$ (Haug *et al.*, 1986; Santos e Choi, 1992; Park e Choi, 1996b). Assim, observa-se que a forma bilinear $a_{\mathbf{d}}(\cdot,\cdot)$ depende tanto explicitamente das variáveis de projeto $\mathbf{d}$ quanto implicitamente através da resposta estrutural x_{n+1} na configuração de equilíbrio $\mathbf{X}_{n+1}$.

Quando o projeto corrente $\mathbf{d}$ sofre uma perturbação $\tau\delta\mathbf{d}$, origina-se um novo projeto $\mathbf{d}+\tau\delta\mathbf{d}$. Logo, tem-se um novo sistema e uma nova equação de equilíbrio,

$$a\left(\mathbf{d}+\tau\delta\mathbf{d};\mathsf{x}_{n+1}\left(\mathbf{d}+\tau\delta\mathbf{d}\right),\delta\mathsf{x}\right)=l_{n+1}\left(\mathbf{d}+\tau\delta\mathbf{d};\delta\mathsf{x}\right),\qquad\forall\delta\mathsf{x}\in\mathcal{V}.\tag{3.28}$$

A sensibilidade da resposta estrutural x_{n+1} em relação aos parâmetros de projeto pode ser obtida da primeira derivada de (3.28) em relação a τ. A sensibilidade procurada consiste na derivada direcional de x_{n+1} na direção $\delta\mathbf{d}$, ou seja,

$$\dot{\mathsf{x}}_{n+1}\left(\mathbf{d}\right)=D\mathsf{x}_{n+1}\left(\mathbf{d}\right)\left[\delta\mathbf{d}\right]=\lim_{\tau\to0}\frac{\mathsf{x}_{n+1}\left(\mathbf{d}+\tau\delta\mathbf{d}\right)-\mathsf{x}_{n+1}\left(\mathbf{d}\right)}{\tau}=\left(\nabla_{\mathbf{d}}\mathsf{x}_{n+1}\right)\delta\mathbf{d}.\tag{3.29}$$

A sensibilidade do campo de deslocamentos está relacionada a (3.29) através de (A.25), ou seja,

$$\dot{\mathbf{u}}_{n+1}\left(\mathbf{d}\right)=\dot{\mathsf{x}}_{n+1}\left(\mathbf{d}\right)=\lim_{\tau\to0}\frac{\mathbf{u}_{n+1}\left(\mathbf{d}+\tau\delta\mathbf{d}\right)-\mathbf{u}_{n+1}\left(\mathbf{d}\right)}{\tau}=\left(\nabla_{\mathbf{d}}\mathbf{u}_{n+1}\right)\delta\mathbf{d}.\tag{3.30}$$

As derivadas das formas bilinear e linear da equação de estado (3.27) devido à dependência explícita em relação às variáveis de projeto $\mathbf{d}$ são dadas, respectivamente, por

$$\begin{aligned}a'\left(\mathbf{d};\mathsf{x}_{n+1}\left(\mathbf{d}\right),\delta\mathsf{x}\right) &= D\left[a\left(\mathbf{d};\mathsf{x}_{n+1}\left(\mathbf{d}\right),\delta\mathsf{x}\right)\right]\left[\delta\mathbf{d}\right]\\ &= \lim_{\tau\to0}\frac{a\left(\mathbf{d}+\tau\delta\mathbf{d};\mathsf{x}_{n+1}\left(\mathbf{d}\right),\delta\mathsf{x}\right)-a\left(\mathbf{d};\mathsf{x}_{n+1}\left(\mathbf{d}\right),\delta\mathsf{x}\right)}{\tau}\\ &= \frac{\partial a_{n+1}}{\partial\tau}\\ &= \nabla_{\mathbf{d}}a\left(\mathsf{x}_{n+1},\delta\mathsf{x}\right)\cdot\delta\mathbf{d},\end{aligned}\tag{3.31}$$

$$\begin{aligned}l'_{n+1}\left(\mathbf{d};\delta\mathsf{x}\right) &= Dl_{n+1}\left(\mathbf{d};\delta\mathsf{x}\right)\left[\delta\mathbf{d}\right]\\ &= \lim_{\tau\to0}\frac{l_{n+1}\left(\mathbf{d}+\tau\delta\mathbf{d};\delta\mathsf{x}\right)-l_{n+1}\left(\mathbf{d};\delta\mathsf{x}\right)}{\tau}\\ &= \frac{\partial l_{n+1}}{\partial\tau}\\ &= \nabla_{\mathbf{d}}l_{n+1}\left(\delta\mathsf{x}\right)\cdot\delta\mathbf{d}.\end{aligned}\tag{3.32}$$

Como a forma bilinear $a_{\mathbf{d}}(\cdot,\cdot)$ também depende implicitamente de $\mathbf{d}$, sua derivada total é

$$\begin{aligned}\left[a\left(\mathbf{d};\mathsf{x}_{n+1}(\mathbf{d}),\delta\mathsf{x}\right)\right]^{\cdot} &= D\left[a\left(\mathbf{d};\mathsf{x}_{n+1}(\mathbf{d}),\delta\mathsf{x}\right)\right][\delta\mathbf{d}] = \frac{da_{n+1}}{d\tau} = \\ &= \lim_{\tau\to 0}\frac{a\left(\mathbf{d}+\tau\delta\mathbf{d};\mathsf{x}_{n+1}(\mathbf{d}+\tau\delta\mathbf{d}),\delta\mathsf{x}\right)-a\left(\mathbf{d};\mathsf{x}_{n+1}(\mathbf{d}),\delta\mathsf{x}\right)}{\tau}.\end{aligned} \tag{3.33}$$

Empregando a regra da cadeia e a relação (3.31), tem-se que

$$\begin{aligned}\left[a\left(\mathbf{d};\mathsf{x}_{n+1}(\mathbf{d}),\delta\mathsf{x}\right)\right]^{\cdot} &= a'_{\mathbf{d}}\left(\mathsf{x}_{n+1},\delta\mathsf{x}\right)+D_{\mathsf{x}}\left[a_{\mathbf{d}}\left(\mathsf{x}_{n+1},\delta\mathsf{x}\right)\right]\left[\dot{\mathsf{x}}_{n+1}\right] \\ &= a'\left(\mathbf{d};\mathsf{x}_{n+1}(\mathbf{d}),\delta\mathsf{x}\right)+\delta a\left(\mathbf{d};\mathsf{x}_{n+1};\dot{\mathsf{x}}_{n+1}(\mathbf{d}),\delta\mathsf{x}\right) \\ &= \nabla_{\mathbf{d}}a\left(\mathsf{x}_{n+1},\delta\mathsf{x}\right)\cdot\delta\mathbf{d}+\delta a\left(\mathsf{x}_{n+1};\dot{\mathsf{x}}_{n+1},\delta\mathsf{x}\right) \qquad (3.34) \\ &= \nabla_{\mathbf{d}}a\left(\mathsf{x}_{n+1},\delta\mathsf{x}\right)\cdot\delta\mathbf{d}+\delta a\left(\mathsf{x}_{n+1};\dot{\mathbf{u}}_{n+1},\delta\mathsf{x}\right). \qquad (3.35)\end{aligned}$$

Como a forma linear depende apenas explicitamente de $\mathbf{d}$, as suas derivadas parcial e total coincidem, ou seja, $\left[l_{n+1}(\mathbf{d};\delta\mathsf{x})\right]^{\cdot} = D\left[l_{n+1}(\mathbf{d};\delta\mathsf{x})\right][\delta\mathbf{d}] = l'_{n+1}(\mathbf{d};\delta\mathsf{x})$. Portanto, a derivada total da equação de equilíbrio

$$\left[a\left(\mathbf{d};\mathsf{x}_{n+1}(\mathbf{d}),\delta\mathsf{x}\right)\right]^{\cdot} = \left[l_{n+1}(\mathbf{d};\delta\mathsf{x})\right]^{\cdot}$$

resulta em

$$\delta a\left(\mathbf{d};\mathsf{x}_{n+1};\dot{\mathbf{u}}_{n+1}(\mathbf{d}),\delta\mathsf{x}\right) = l'_{n+1}(\mathbf{d};\delta\mathsf{x}) - a'\left(\mathbf{d};\mathsf{x}_{n+1}(\mathbf{d}),\delta\mathsf{x}\right).,$$

De forma sintética,

$$\boxed{\delta a\left(\mathsf{x}_{n+1};\dot{\mathbf{u}}_{n+1},\delta\mathsf{x}\right) = l'_{\mathbf{d}n+1}(\delta\mathsf{x}) - a'_{\mathbf{d}}\left(\mathsf{x}_{n+1},\delta\mathsf{x}\right).} \tag{3.36}$$

Deve-se observar que a forma bilinear $\delta a\left(\mathsf{x}_{n+1};\cdot,\cdot\right)$ em (3.36) é *exatamente* a mesma de (2.74) uma vez que o procedimento de derivação é o mesmo[2]. Assim, como a configuração de equilíbrio x_{n+1} já foi determinada, a matriz correspondente a $\delta a\left(\mathsf{x}_{n+1};\cdot,\cdot\right)$ também já foi construída (e, se um método direto de solução de sistemas lineares estiver sendo empregado, essa matriz já estará inclusive fatorada). Portanto, a determinação da sensibilidade $\dot{\mathsf{x}}_{n+1}$ envolve somente a solução de um único sistema linear, cuja matriz coincide com a matriz de rigidez tangente da última iteração de análise de resposta do passo I_{n+1}, apesar do procedimento iterativo exigido para determinar a resposta. Aliás,

[2]Se o método de solução empregado não utilizar $\delta a\left(\mathsf{x}_{n+1};\cdot,\cdot\right)$ (que dá origem à matriz de rigidez tangente) explicitamente, ela deve ser calculada para efeito da análise de sensibilidade.

existe razoável consenso de que a análise de sensibilidade deve sempre ser linear, quaisquer que sejam as não-linearidades envolvidas na análise de resposta (Kim, 1999).

A solução do sistema (3.36) é então aplicado em (3.2), resultando na sensibilidade do funcional de performance à alteração do conjunto de parâmetros $\mathbf{d}$ na direção $\delta\mathbf{d}$. O procedimento descrito é denominado *método direto de análise de sensibilidade* e requer a solução de um sistema linear (3.36) por variável de projeto para determinar todas as Nd componentes de $\nabla_{\mathbf{d}}\psi$, como mostrado abaixo:

$$\boxed{\begin{aligned} &\delta a\left(\mathsf{x}_{n+1};\frac{\partial \mathbf{u}_{n+1}}{\partial d_k},\delta\mathsf{x}\right)=\frac{\partial l_{n+1}}{\partial d_k}\left(\delta\mathsf{x}\right)-\frac{\partial a}{\partial d_k}\left(\mathsf{x}_{n+1},\delta\mathsf{x}\right),\\ &\left(\nabla_{\mathbf{d}}\psi\right)_k=\frac{\partial\psi}{\partial d_k}=\int_{\mathcal{B}}\left[\mathcal{G}_{,\mathbf{u}}\cdot\frac{\partial \mathbf{u}_{n+1}}{\partial d_k}+\mathcal{G}_{,\nabla\mathbf{u}}\cdot\nabla\left(\frac{\partial \mathbf{u}_{n+1}}{\partial d_k}\right)+\frac{\partial\mathcal{G}_{n+1}}{\partial d_k}\right]\,dV,\\ &k=1,\ldots,Nd.\end{aligned}} \tag{3.37}$$

Quando $\delta a\left(\mathsf{x}_{n+1};\cdot,\cdot\right)$ é simétrico, pode-se aplicar um procedimento alternativo denominado *método adjunto de análise de sensibilidade*. Nesse método, elimina-se a dependência direta de (3.2) em relação a $\dot{\mathsf{x}}_{n+1}$ introduzindo-se a variável adjunta $\boldsymbol{\lambda}$ e a seguinte equação adjunta:

$$\boxed{\delta a\left(\mathsf{x}_{n+1};\boldsymbol{\lambda},\delta\boldsymbol{\lambda}\right)=\int_{\mathcal{B}}\left(\mathcal{G}_{,\mathbf{u}}\cdot\delta\boldsymbol{\lambda}+\mathcal{G}_{,\nabla\mathbf{u}}\cdot\nabla\delta\boldsymbol{\lambda}\right)\,dV,\qquad\forall\delta\boldsymbol{\lambda}\in\mathcal{V}.} \tag{3.38}$$

Das propriedades de diferenciabilidade usualmente assumidas para $\mathcal{G}$ resulta que o lado direito de (3.38) é uma forma linear, podendo-se aplicar o teorema de Lax-Milgram.

Como $\dot{\mathbf{u}}_{n+1}\in\mathcal{V}$, então assumindo $\delta\boldsymbol{\lambda}=\dot{\mathbf{u}}_{n+1}$,

$$\delta a\left(\mathsf{x}_{n+1};\boldsymbol{\lambda},\dot{\mathbf{u}}_{n+1}\right)=\int_{\mathcal{B}}\left(\mathcal{G}_{,\mathbf{u}}\cdot\dot{\mathbf{u}}_{n+1}+\mathcal{G}_{,\nabla\mathbf{u}}\cdot\nabla\dot{\mathbf{u}}_{n+1}\right)\,dV.$$

Se $\delta a\left(\mathsf{x}_{n+1};\cdot,\cdot\right)$ é simétrico, então

$$\begin{aligned}\delta a\left(\mathsf{x}_{n+1};\boldsymbol{\lambda},\dot{\mathbf{u}}_{n+1}\right)&=\delta a\left(\mathsf{x}_{n+1};\dot{\mathbf{u}}_{n+1},\boldsymbol{\lambda}\right)\\&\Rightarrow\int_{\mathcal{B}}\left(\mathcal{G}_{,\mathbf{u}}\cdot\dot{\mathbf{u}}_{n+1}+\mathcal{G}_{,\nabla\mathbf{u}}\cdot\nabla\dot{\mathbf{u}}_{n+1}\right)\,dV=l'_{\mathbf{d}n+1}\left(\boldsymbol{\lambda}\right)-a'_{\mathbf{d}}\left(\mathsf{x}_{n+1},\boldsymbol{\lambda}\right).\end{aligned} \tag{3.39}$$

Substituindo (3.39) em (3.2) tem-se

$$\boxed{\dot{\psi}\left(\mathbf{d}\right)=l'_{\mathbf{d}n+1}\left(\boldsymbol{\lambda}\right)-a'_{\mathbf{d}}\left(\mathsf{x}_{n+1},\boldsymbol{\lambda}\right)+\int_{\mathcal{B}}\mathcal{G}_{,\mathbf{d}}\cdot\delta\mathbf{d}\,dV} \tag{3.40}$$

como expressão de sensibilidade do funcional de performance. Em resumo, resolve-se a equação adjunta para cada um dos Nf funcionais e, em seguida, aplica-se (3.40). Em termos de derivadas parciais tem-se:

$$(\nabla_{\mathbf{d}}\psi)_k = \frac{\partial \psi}{\partial d_k} = \frac{\partial l_{n+1}}{\partial d_k}(\boldsymbol{\lambda}) - \frac{\partial a}{\partial d_k}(\mathsf{x}_{n+1}, \boldsymbol{\lambda}) + \int_{\mathcal{B}} \frac{\partial \mathcal{G}}{\partial d_k}\, dV, \qquad k = 1, \ldots, Nf. \tag{3.41}$$

No método adjunto, resolve-se um sistema linear correspondente a $\delta a\,(\mathsf{x}_{n+1}; \cdot, \cdot)$ por funcional, enquanto no método direto tem-se um sistema linear por variável. É importante observar, entretanto, que ambos os métodos são equivalentes em termos dos resultados.

3.4.2 Análise de Sensibilidade à Forma

Pode-se definir uma família de problemas estruturais associando-se a cada domínio $\mathcal{B}_\tau$ um problema variacional da forma

$$a_{\mathcal{B}_\tau}(\mathsf{x}_\tau, \delta\mathsf{x}_\tau) = l_{\mathcal{B}_\tau}(\delta\mathsf{x}_\tau), \qquad \forall \delta\mathsf{x}_\tau \in \mathcal{V}_\tau. \tag{3.42}$$

De posse das expressões anteriores, pode-se retornar à (3.42) e obter sua sensibilidade à alteração da forma. Sendo

$$a_{\mathcal{B}_\tau}(\mathsf{x}_\tau, \delta\mathsf{x}_\tau) = \int_{\mathcal{B}_\tau} \mathbf{S}_\tau \cdot \delta\mathbf{E}_\tau \, dV = \int_{\mathcal{B}_\tau} \Theta_\tau \left(\nabla_\tau \mathbf{u}_\tau, \nabla_\tau \delta\mathbf{u}_\tau\right) \, dV, \tag{3.43}$$

sua derivada total em relação ao parâmetro τ é

$$\begin{aligned} \left[a_{\mathcal{B}_\tau}(\mathsf{x}_\tau, \delta\mathsf{x}_\tau)\right]^{\cdot} &= \int_{\mathcal{B}} \left(\dot{\Theta}_\tau + \Theta_\tau \operatorname{div} \mathbf{V}_s\right)_m (\det \mathsf{F}) \, dV & (3.44)\\ &= \int_{\mathcal{B}} \left(\Theta_\tau' + \mathbf{V}_s \cdot \nabla_\tau \Theta_\tau + \Theta_\tau \operatorname{div} \mathbf{V}_s\right)_m (\det \mathsf{F}) \, dV. & (3.45) \end{aligned}$$

Segundo (3.8), em $\tau = 0$ tem-se

$$\left[a\,(\mathsf{x}, \delta\mathsf{x})\right]^{\cdot} = \int_{\mathcal{B}} \left(\Theta' + \mathbf{V} \cdot \nabla\Theta + \Theta \operatorname{Div} \mathbf{V}\right) \, dV, \tag{3.46}$$

sendo

$$\Theta' = (\mathbf{S} \cdot \delta\mathbf{E})' = \mathbf{S}' \cdot \delta\mathbf{E} + \mathbf{S} \cdot \delta\mathbf{E}', \tag{3.47}$$

$$\mathbf{S}' = \frac{\partial \mathbf{S}}{\partial \tau} = D_{\mathbf{E}}\mathbf{S}\left[\mathbf{E}'\right] = W_{,\mathbf{E},\mathbf{E}} : \mathbf{E}' = \mathsf{C}\,(\mathsf{x}) : \mathbf{E}', \tag{3.48}$$

$$\mathbf{F}' = \frac{\partial \mathbf{F}}{\partial \tau} = \frac{\partial\,(\nabla \mathsf{x})}{\partial \tau} = \nabla \mathsf{x}' = \frac{\partial\,(\nabla \mathbf{u})}{\partial \tau} = \nabla \mathbf{u}',$$

$$\mathbf{E}' = \frac{\partial \mathbf{E}}{\partial \tau} = \frac{1}{2}\left(\mathbf{F}'^T\mathbf{F} + \mathbf{F}^T\mathbf{F}'\right) = \frac{1}{2}\left[\left(\nabla\mathbf{u}'\right)^T\mathbf{F} + \mathbf{F}^T\left(\nabla\mathbf{u}'\right)\right].$$

A partir de (3.15),

$$\mathbf{E}' = \frac{1}{2}\left(\nabla\dot{\mathbf{u}}^T\mathbf{F} + \mathbf{F}^T\nabla\dot{\mathbf{u}}\right) - \frac{1}{2}\left[\left(\nabla\mathbf{u}\nabla\mathbf{V}\right)^T\mathbf{F} + \mathbf{F}^T\left(\nabla\mathbf{u}\nabla\mathbf{V}\right)\right] - \frac{1}{2}\left[\left(\nabla\nabla\mathbf{u}\mathbf{V}\right)^T\mathbf{F} + \mathbf{F}^T\left(\nabla\nabla\mathbf{u}\mathbf{V}\right)\right].$$

A expressão anterior pode ser denotada como

$$\mathbf{E}' = D\mathbf{E}\left[\dot{\mathbf{u}}\right] + \mathbf{E}_{\mathbf{V}}\left(\mathbf{u}\right) - \frac{1}{2}\left[\left(\nabla\nabla\mathbf{u}\mathbf{V}\right)^T\mathbf{F} + \mathbf{F}^T\left(\nabla\nabla\mathbf{u}\mathbf{V}\right)\right], \tag{3.49}$$

sendo

$$\mathbf{E}_{\mathbf{V}}\left(\mathbf{u}\right) = -\frac{1}{2}\left[\left(\nabla\mathbf{u}\nabla\mathbf{V}\right)^T\mathbf{F} + \mathbf{F}^T\left(\nabla\mathbf{u}\nabla\mathbf{V}\right)\right]. \tag{3.50}$$

A derivada do deslocamento virtual $\delta\mathbf{u}$ é obtida de (3.13). Lembrando que essa derivada é nula (Haug *et al.*, 1986), obtem-se a seguinte relação

$$\delta\dot{\mathbf{u}} = \delta\mathbf{u}' + \nabla\delta\mathbf{u}\mathbf{V} = 0 \Rightarrow \delta\mathbf{u}' = -\nabla\delta\mathbf{u}\mathbf{V}.$$

Tomando o gradiente da expressão anterior, tem-se

$$\nabla\delta\mathbf{u}' = -\nabla\delta\mathbf{u}\nabla\mathbf{V} - \nabla\nabla\delta\mathbf{u}\mathbf{V}.$$

Logo,

$$\delta\mathbf{E}' = \frac{\partial\delta\mathbf{E}}{\partial\tau} = \frac{1}{2}\left(\delta\mathbf{F}^{T\prime}\mathbf{F} + \mathbf{F}^T\delta\mathbf{F}'\right) + \frac{1}{2}\left(\delta\mathbf{F}^T\mathbf{F}' + \mathbf{F}^{T\prime}\delta\mathbf{F}\right).$$

Como $\delta\mathbf{F}' = \nabla\delta\mathbf{u}'$ e substituindo (3.18), conclui-se que

$$\begin{aligned}
\delta\mathbf{E}' &= \frac{1}{2}\left[\left(-\nabla\delta\mathbf{u}\nabla\mathbf{V}\right)^T\mathbf{F} + \mathbf{F}^T\left(-\nabla\delta\mathbf{u}\nabla\mathbf{V}\right)\right] + \frac{1}{2}\left[\delta\mathbf{F}^T\left(\nabla\dot{\mathbf{u}} - \nabla\mathbf{u}\nabla\mathbf{V}\right) + \left(\nabla\dot{\mathbf{u}} - \nabla\mathbf{u}\nabla\mathbf{V}\right)^T\delta\mathbf{F}\right] + \\
&\quad + \frac{1}{2}\left[\left(-\nabla\nabla\delta\mathbf{u}\mathbf{V}\right)^T\mathbf{F} + \mathbf{F}^T\left(-\nabla\nabla\delta\mathbf{u}\mathbf{V}\right)\right] + \frac{1}{2}\left[\delta\mathbf{F}^T\left(-\nabla\nabla\mathbf{u}\mathbf{V}\right) + \left(-\nabla\nabla\mathbf{u}\mathbf{V}\right)^T\delta\mathbf{F}\right] \\
&= \frac{1}{2}\left(\delta\mathbf{F}^T\nabla\dot{\mathbf{u}} + \nabla\dot{\mathbf{u}}^T\delta\mathbf{F}\right) + \\
&\quad + \frac{1}{2}\left[\left(-\nabla\delta\mathbf{u}\nabla\mathbf{V}\right)^T\mathbf{F} + \mathbf{F}^T\left(-\nabla\delta\mathbf{u}\nabla\mathbf{V}\right)\right] + \frac{1}{2}\left[\delta\mathbf{F}^T\left(-\nabla\mathbf{u}\nabla\mathbf{V}\right) + \left(-\nabla\mathbf{u}\nabla\mathbf{V}\right)^T\delta\mathbf{F}\right] \\
&\quad + \frac{1}{2}\left[\left(-\nabla\nabla\delta\mathbf{u}\mathbf{V}\right)^T\mathbf{F} + \mathbf{F}^T\left(-\nabla\nabla\delta\mathbf{u}\mathbf{V}\right)\right] + \frac{1}{2}\left[\delta\mathbf{F}^T\left(-\nabla\nabla\mathbf{u}\mathbf{V}\right) + \left(-\nabla\nabla\mathbf{u}\mathbf{V}\right)^T\delta\mathbf{F}\right].
\end{aligned}$$

Reescreve-se a relação anterior como

$$\delta\mathbf{E}' = D\delta\mathbf{E}\left[\dot{\mathbf{u}}\right] + \delta\mathbf{E}_{\mathbf{V}}\left(\mathbf{u}\right) +$$

$$+\frac{1}{2}\left[\left(-\nabla\nabla\delta\mathbf{u}\mathbf{V}\right)^{T}\mathbf{F}+\mathbf{F}^{T}\left(-\nabla\nabla\delta\mathbf{u}\mathbf{V}\right)\right]+\frac{1}{2}\left[\delta\mathbf{F}^{T}\left(-\nabla\nabla\mathbf{u}\mathbf{V}\right)+\left(-\nabla\nabla\mathbf{u}\mathbf{V}\right)^{T}\delta\mathbf{F}\right], \tag{3.51}$$

sendo

$$\delta\mathbf{E}_{\mathbf{V}}\left(\mathbf{u}\right)=\frac{1}{2}\left[\left(-\nabla\delta\mathbf{u}\nabla\mathbf{V}\right)^{T}\mathbf{F}+\mathbf{F}^{T}\left(-\nabla\delta\mathbf{u}\nabla\mathbf{V}\right)\right]+\frac{1}{2}\left[\delta\mathbf{F}^{T}\left(-\nabla\mathbf{u}\nabla\mathbf{V}\right)+\left(-\nabla\mathbf{u}\nabla\mathbf{V}\right)^{T}\delta\mathbf{F}\right]. \tag{3.52}$$

O termo $\nabla\Theta$ em (3.46) é dado por

$$\nabla\Theta=\nabla\left(\mathbf{S}\cdot\delta\mathbf{E}\right)=\nabla\mathbf{S}\cdot\delta\mathbf{E}+\mathbf{S}\cdot\nabla\delta\mathbf{E}, \tag{3.53}$$

sendo

$$\begin{aligned}\nabla\delta\mathbf{E} &= \frac{1}{2}\left[\nabla\left(\delta\mathbf{F}^{T}\mathbf{F}+\mathbf{F}^{T}\delta\mathbf{F}\right)\right]\\ &= \frac{1}{2}\left[\left(\nabla\nabla\delta\mathbf{u}\right)^{T}\mathbf{F}+\mathbf{F}^{T}\left(\nabla\nabla\delta\mathbf{u}\right)\right]+\frac{1}{2}\left[\delta\mathbf{F}^{T}\left(\nabla\nabla\mathbf{u}\right)+\left(\nabla\nabla\mathbf{u}\right)^{T}\delta\mathbf{F}\right],\end{aligned}$$

$$\nabla\mathbf{S}=\mathsf{C}\left(\mathsf{x}\right):\nabla\mathbf{E},$$

$$\nabla\mathbf{E}=\frac{1}{2}\left[\nabla\left(\mathbf{F}^{T}\mathbf{F}\right)\right]=\frac{1}{2}\left[\left(\nabla\nabla\mathbf{u}\right)^{T}\mathbf{F}+\mathbf{F}^{T}\left(\nabla\nabla\mathbf{u}\right)\right].$$

Portanto, substituindo as relações anteriores em (3.46) e rearranjando, obtém-se

$$\begin{aligned}\left[a\left(\mathsf{x},\delta\mathsf{x}\right)\right]^{\cdot} &= \int_{\mathcal{B}}\left[\mathsf{C}\left(\mathsf{x}\right):D\mathbf{E}\left[\dot{\mathbf{u}}\right]\cdot\delta\mathbf{E}+\mathbf{S}\cdot D\delta\mathbf{E}\left[\dot{\mathbf{u}}\right]\right]\,dV+\\ &+ \int_{\mathcal{B}}\left[\mathsf{C}\left(\mathsf{x}\right):\mathbf{E}_{\mathbf{V}}\left(\mathbf{u}\right)\cdot\delta\mathbf{E}+\mathbf{S}\cdot\delta\mathbf{E}_{\mathbf{V}}\left(\mathbf{u}\right)+\mathbf{S}\cdot\delta\mathbf{E}\,\mathrm{Div}\,\mathbf{V}\right]\,dV.\end{aligned} \tag{3.54}$$

A primeira integral de (3.54) é idêntica a (2.71), trocando-se apenas $\Delta\mathbf{u}$ por $\dot{\mathbf{u}}$. Logo é usual reescrever (3.54) como a soma

$$\left[a\left(\mathsf{x},\delta\mathsf{x}\right)\right]^{\cdot}=\delta a\left(\mathsf{x};\dot{\mathbf{u}},\delta\mathsf{x}\right)+a_{V}'\left(\mathsf{x},\delta\mathsf{x}\right), \tag{3.55}$$

sendo

$$a_{V}'\left(\mathsf{x},\delta\mathsf{x}\right)=\int_{\mathcal{B}}\left[\mathsf{C}\left(\mathsf{x}\right):\mathbf{E}_{\mathbf{V}}\left(\mathbf{u}\right)\cdot\delta\mathbf{E}+\mathbf{S}\cdot\delta\mathbf{E}_{\mathbf{V}}\left(\mathbf{u}\right)+\mathbf{S}\cdot\delta\mathbf{E}\,\mathrm{Div}\,\mathbf{V}\right]\,dV. \tag{3.56}$$

No caso de carregamentos independentes da geometria, a sensibilidade de $l\left(\delta\mathsf{x}\right)$ em relação à modificação de $\mathcal{B}$ é

$$\begin{aligned}\dot{l}\left(\delta\mathsf{x}\right) &= l_{V}'\left(\delta\mathsf{x}\right)=\\ &= \int_{\mathcal{B}}\left[\left(\nabla\mathbf{b}_{0}\right)\mathbf{V}+\mathbf{b}_{0}\,\mathrm{Div}\,\mathbf{V}\right]\cdot\delta\mathsf{x}\,dV+\int_{\Gamma^{2}}\left[\left(\nabla\mathbf{t}\right)\mathbf{V}+\mathbf{t}\ \left(\mathbf{V}\cdot\mathbf{n}\right)\mathrm{Div}\,\mathbf{n}\right]\cdot\delta\mathsf{x}\,dA\end{aligned} \tag{3.57}$$

Como resultado, tem-se que a sensibilidade da equação variacional é

$$[a(\mathsf{x}_{n+1}, \delta\mathsf{x})]^{\cdot} = \dot{l}_{n+1}(\delta\mathsf{x}) \Rightarrow \delta a(\mathsf{x}_{n+1}; \dot{\mathbf{u}}_{n+1}, \delta\mathsf{x}) + a'_V(\mathsf{x}_{n+1}, \delta\mathsf{x}) = l'_{V n+1}(\delta\mathsf{x}).$$

A partir daí, obtem-se a equação variacional do método direto para sensibilidade à forma como

$$\boxed{\delta a(\mathsf{x}_{n+1}; \dot{\mathbf{u}}_{n+1}, \delta\mathsf{x}) = l'_{V n+1}(\delta\mathsf{x}) - a'_V(\mathsf{x}_{n+1}, \delta\mathsf{x}),} \tag{3.58}$$

sendo $\delta a(\mathsf{x}_{n+1}; \cdot, \cdot)$ a forma bilinear da última iteração de análise e x_{n+1} a solução da análise de resposta na iteração I_{n+1}.

O método adjunto para sensibilidade a forma é idêntico ao desenvolvido para sensibilidade à parâmetros discretos, enfatizando-se a hipótese de $\delta a(\mathsf{x}_{n+1}; \cdot, \cdot)$ simétrico. Portanto, aplicando a equação adjunta (3.38), elimina-se a dependência direta em relação a $\dot{\mathbf{u}}_{n+1}$ e $\nabla\dot{\mathbf{u}}_{n+1}$ e, utilizando (3.26), reescreve-se a expressão de sensibilidade do funcional ψ como

$$\boxed{\dot{\psi} = l'_{V n+1}(\boldsymbol{\lambda}) - a'_V(\mathsf{x}_{n+1}, \boldsymbol{\lambda}) + \int_{\mathcal{B}} \left(\mathcal{G}\,\mathrm{Div}\,\mathbf{V} + \mathbf{V} \cdot \nabla\mathcal{G} - \mathcal{G}_{,\nabla\mathbf{u}} \cdot \nabla\mathbf{u}_{n+1}\nabla\mathbf{V}\right)\, dV.} \tag{3.59}$$

3.5 Análise de Sensibilidade em Hiperelasticidade Quasi-Incompressível

3.5.1 Análise de Sensibilidade a Parâmetros Discretos

Considere a mesma situação discutida na Seção 3.4 de determinar a sensibilidade de funcionais de performance à variação de parâmetros discretos. Obtém-se expressão semelhante a (3.2), apenas trocando-se a dependência em x_{n+1} por $\mathsf{s}_{n+1} = \begin{bmatrix} \mathsf{x}_{n+1} & p_{n+1} \end{bmatrix}^T$.

Da mesma forma, deve-se retornar à equação variacional do sistema e determinar sua sensibilidade às variáveis de projeto. Seja então

$$\begin{cases} a_{\mathbf{d}}(\mathsf{x}_{n+1}, \delta\mathsf{x}) + b_{1\mathbf{d}}(\delta\mathsf{x}, p_{n+1}) - l_{\mathbf{d}}(\delta\mathsf{x}) & = 0 \\ b_{2\mathbf{d}}(\mathsf{x}_{n+1}, \delta p) - g_{\mathbf{d}}(p_{n+1}, \delta p) & = 0 \end{cases}, \qquad \mathsf{x}_{n+1} \in \mathcal{X}_{n+1}, \forall \delta\mathsf{x} \in \mathcal{V}, \tag{3.60}$$

a equação de equilíbrio do sistema parametrizado por $\mathbf{d}$ com solução $\mathsf{s}_{n+1} = \begin{bmatrix} \mathsf{x}_{n+1} & p_{n+1} \end{bmatrix}^T$.

As derivadas totais dos termos do sistema (3.60) são

$$\begin{aligned}
\left[a_{\mathbf{d}}(\mathsf{x},\delta\mathsf{x})+b_{1\mathbf{d}}(\delta\mathsf{x},p)\right]^{\cdot} &= \nabla_{\mathbf{d}}a(\mathsf{x},\delta\mathsf{x})\cdot\delta\mathbf{d}+\nabla_{\mathbf{d}}b_1(\delta\mathsf{x},p)\cdot\delta\mathbf{d}+\delta a(\dot{\mathbf{u}},\delta\mathsf{x})+\delta b(\delta\mathsf{x},\dot{p})\\
&= \dot{a}_{\mathbf{d}}(\mathsf{x},\delta\mathsf{x})+\dot{b}_{1\mathbf{d}}(\delta\mathsf{x},p)+\delta a(\dot{\mathbf{u}},\delta\mathsf{x})+\delta b(\delta\mathsf{x},\dot{p}),
\end{aligned}$$

$$\begin{aligned}
\left[b_{2\mathbf{d}}(\mathsf{x},\delta p)-g_{\mathbf{d}}(p,\delta p)\right]^{\cdot} &= \nabla_{\mathbf{d}}b_2(\mathsf{x},\delta p)\cdot\delta\mathbf{d}-\nabla_{\mathbf{d}}g(p,\delta p)\cdot\delta\mathbf{d}+\delta b(\dot{\mathbf{u}},\delta p)-\delta g(\dot{p},\delta p)\\
&= \dot{b}_{2\mathbf{d}}(\mathsf{x},\delta p)-\dot{g}_{\mathbf{d}}(p,\delta p)+\delta b(\dot{\mathbf{u}},\delta p)-\delta g(\dot{p},\delta p).
\end{aligned}$$

Logo, tem-se o seguinte sistema

$$\left\{\begin{array}{ll}
\delta a(\mathsf{s}_{n+1};\dot{\mathbf{u}}_{n+1},\delta\mathsf{x})+\delta b(\mathsf{s}_{n+1};\delta\mathsf{x},\dot{p}) & =\dot{l}_{\mathbf{d}n+1}(\delta\mathsf{x})-\dot{a}_{\mathbf{d}}(\mathsf{x}_{n+1},\delta\mathsf{x})-\dot{b}_{1\mathbf{d}}(\delta\mathsf{x},p_{n+1})\\
\delta b(\mathsf{s}_{n+1};\dot{\mathbf{u}}_{n+1},\delta p) & =\delta g(p_{n+1},\dot{p}_{n+1},\delta p)-\dot{b}_{2\mathbf{d}}(\mathsf{x}_{n+1},\delta p)+\dot{g}_{\mathbf{d}}(p_{n+1},\delta p)
\end{array}\right. . \quad (3.61)$$

Utilizando a mesma hipótese da Seção 2.2.9, resolve-se a segunda equação de (3.61), reescrevendo $\dot{p}_{n+1}$ em função de $\dot{\mathsf{x}}_{n+1}$ através da projeção da derivada da pressão no elemento e da seguinte forma

$$\dot{\hat{p}}_{en+1}=\frac{\tilde{\kappa}}{\delta\hat{p}_e\int_{\mathcal{B}_e}dV}\left[\delta b(\mathsf{s}_{n+1};\dot{\mathbf{u}}_{n+1},\delta p)+\dot{b}_{2\mathbf{d}}(\mathsf{x}_{n+1},\delta p)-\dot{g}_{\mathbf{d}}(p_{n+1},\delta p)\right]. \quad (3.62)$$

3.5.2 Análise de Sensibilidade a Forma

Segue-se o mesmo procedimento da Seção 3.4.2, considerando-se entretanto, as seguintes expressões:

$$\begin{aligned}
\mathbf{S}' &= \frac{\partial\mathbf{S}}{\partial\tau}=\bar{\mathbf{S}}'+\tilde{\mathbf{S}}'=D_{\mathbf{E}}\bar{\mathbf{S}}[\mathbf{E}']+D_{\mathbf{E}}\tilde{\mathbf{S}}[\mathbf{E}']+D_p\tilde{\mathbf{S}}[p']=\\
&= \bar{W}_{,\mathbf{E},\mathbf{E}}:\mathbf{E}'+\tilde{W}_{,\mathbf{E},\mathbf{E}}:\mathbf{E}'+\tilde{W}_{,\mathbf{E},p}:p'=\bar{\mathsf{C}}(\mathsf{x}):\mathbf{E}'+\tilde{\mathsf{C}}(\mathsf{s}):\mathbf{E}'+J_{,\mathbf{E}}p',
\end{aligned}$$

$$\mathbf{S}'=\mathsf{C}(\mathsf{s}):\mathbf{E}'+J_{,\mathbf{E}}p', \quad (3.63)$$

$$\dot{p}=p'+\nabla p\cdot\mathbf{V}\Rightarrow p'=\dot{p}-\nabla p\cdot\mathbf{V}, \quad (3.64)$$

$$\nabla\mathbf{S}=\mathsf{C}(\mathsf{x}):\nabla\mathbf{E}+J_{,\mathbf{E}}\otimes\nabla p. \quad (3.65)$$

Portanto, a derivada total da equação de estado é

$$\begin{aligned}
\left[a\left(\mathsf{x},\delta\mathsf{x}\right)+b_{1}\left(\delta\mathsf{x},p\right)\right]^{\cdot} &= \int_{\mathcal{B}}\left[\mathsf{C}\left(\mathsf{s}\right):D\mathbf{E}\left[\dot{\mathbf{u}}\right]\cdot\delta\mathbf{E}+\mathbf{S}\cdot D\delta\mathbf{E}\left[\dot{\mathbf{u}}\right]\right]\,dV+\int_{\mathcal{B}}J_{,\mathbf{E}}\dot{p}\cdot\delta\mathbf{E}\,dV \\
&+ \int_{\mathcal{B}}\left[\mathsf{C}\left(\mathsf{s}\right):\mathbf{E}_{\mathbf{V}}\left(\mathsf{x}\right)\cdot\delta\mathbf{E}+\mathbf{S}\cdot\delta\mathbf{E}_{\mathbf{V}}\left(\mathsf{x}\right)+\mathbf{S}\cdot\delta\mathbf{E}\,\mathrm{Div}\,\mathbf{V}\right]\,dV \\
&= \dot{l}\left(\delta\mathsf{x}\right)=l_{V}'\left(\delta\mathsf{x}\right),
\end{aligned}$$

ou ainda

$$\left[a\left(\mathsf{x},\delta\mathsf{x}\right)+b_{1}\left(\delta\mathsf{x},p\right)\right]^{\cdot}=\delta a\left(\mathsf{s};\dot{\mathbf{u}},\delta\mathsf{x}\right)+\delta b\left(\mathsf{s};\delta\mathsf{x},\dot{p}\right)+a_{V}'\left(\mathsf{x},\delta\mathsf{x}\right)+b_{1V}'\left(\delta\mathsf{x},p\right)=l_{V}'\left(\delta\mathsf{x}\right). \tag{3.66}$$

Analogamente, para a segunda equação em (3.60),

$$\begin{aligned}
\left[b_{2}\left(\mathsf{x},\delta p\right)-g\left(p,\delta p\right)\right]^{\cdot} &= \int_{\mathcal{B}_{e}}\left(J_{,\mathbf{E}}\cdot D\mathbf{E}\left[\dot{\mathbf{u}}\right]-DG^{*\prime}\left[\dot{p}\right]\right)\delta p\,dV \\
&\quad+\int_{\mathcal{B}_{e}}\left[J_{,\mathbf{E}}\cdot\mathbf{E}_{\mathbf{V}}\left(\mathsf{x}\right)+\left(J-1-G^{*\prime}\right)\mathrm{Div}\,\mathbf{V}\right]\delta p\,dV=0,
\end{aligned}$$

$$\left[b_{2}\left(\mathsf{x},\delta p\right)-g\left(p,\delta p\right)\right]^{\cdot}=\delta b\left(\mathsf{s};\dot{\mathbf{u}},\delta p\right)-\delta g\left(p,\dot{p},\delta p\right)+b_{2V}'\left(\mathsf{x},\delta p\right)-g_{V}'\left(p,\delta p\right). \tag{3.67}$$

Tem-se então um sistema de incógnitas $\dot{\mathsf{x}}_{n+1}$ e $\dot{p}_{n+1}$, ou seja,

$$\boxed{\left\{\begin{array}{ll}
\delta a\left(\mathsf{s}_{n+1};\dot{\mathbf{u}}_{n+1},\delta\mathsf{x}\right)+\delta b\left(\mathsf{s}_{n+1};\delta\mathsf{x},\dot{p}\right) & =l_{Vn+1}'\left(\delta\mathsf{x}\right)-a_{V}'\left(\mathsf{x}_{n+1},\delta\mathsf{x}\right)-b_{1V}'\left(\delta\mathsf{x},p_{n+1}\right) \\
\delta b\left(\mathsf{s}_{n+1};\dot{\mathbf{u}}_{n+1},\delta p\right) & =\delta g\left(p_{n+1},\dot{p}_{n+1},\delta p\right)-b_{2V}'\left(\mathsf{x}_{n+1},\delta p\right) \\
 & \quad+g_{V}'\left(p_{n+1},\delta p\right)
\end{array}\right. .} \tag{3.68}$$

Aplicando o método de projeção de pressão, elimina-se $\dot{p}$ do sistema (3.68), através da expressão

$$\boxed{\dot{\hat{p}}_{en+1}=\frac{\tilde{\kappa}}{\int_{\mathcal{B}_{e}}dV}\int_{\mathcal{B}_{e}}\left[J_{,\mathbf{E}}\cdot D\mathbf{E}\left[\dot{\mathbf{u}}_{n+1}\right]+J_{,\mathbf{E}}\cdot\mathbf{E}_{\mathbf{V}}\left(\mathsf{x}_{n+1}\right)+\left(J-1-G_{n+1}^{*\prime}\right)\mathrm{Div}\,\mathbf{V}\right]\,dV.} \tag{3.69}$$

3.6 Sensibilidade de Carregamento Dependente da Deformação

3.6.1 Análise de Sensibilidade a Parâmetros Discretos

Quando o carregamento depende do estado de deformação, a expressão de sensibilidade do carregamento (3.32) é substituída por

$$\begin{aligned}
\left[l\left(\mathbf{d};\mathsf{x}\left(\mathbf{d}\right),\delta\mathsf{x}\right)\right]^{\cdot} &= D\left[l_{\mathbf{d}}\left(\mathsf{x},\delta\mathsf{x}\right)\right]\left[\delta\mathbf{d}\right]=\dot{l}_{\mathbf{d}}\left(\mathsf{x},\delta\mathsf{x}\right)+D_{\mathsf{x}}\left[l_{\mathbf{d}}\left(\mathsf{x},\delta\mathsf{x}\right)\right]\left[\dot{\mathsf{x}}\right]= \\
&= \dot{l}\left(\mathbf{d};\mathsf{x}\left(\mathbf{d}\right),\delta\mathsf{x}\right)+\delta l\left(\mathbf{d};\dot{\mathsf{x}}\left(\mathbf{d}\right),\delta\mathsf{x}\right)=\nabla_{\mathbf{d}}l\left(\mathsf{x},\delta\mathsf{x}\right)\cdot\delta\mathbf{d}+\delta l\left(\dot{\mathbf{u}},\delta\mathsf{x}\right),
\end{aligned} \tag{3.70}$$

sendo

$$\dot{l}\left(\mathbf{d};\mathsf{x}\left(\mathbf{d}\right),\delta\mathsf{x}\right)=\lim_{\tau\to 0}\frac{l\left(\mathbf{d}+\tau\delta\mathbf{d};\mathsf{x}\left(\mathbf{d}\right),\delta\mathsf{x}\right)-l\left(\mathbf{d};\mathsf{x}\left(\mathbf{d}\right),\delta\mathsf{x}\right)}{\tau}=\nabla_{\mathbf{d}}l\left(\mathsf{x},\delta\mathsf{x}\right)\cdot\delta\mathbf{d}.$$

Isso altera (3.36) para

$$\delta a\left(\mathsf{x}_{n+1};\dot{\mathbf{u}}_{n+1},\delta\mathsf{x}\right)-\delta l\left(\mathsf{x}_{n+1};\dot{\mathbf{u}}_{n+1},\delta\mathsf{x}\right)=\dot{l}_{\mathbf{d}n+1}\left(\mathsf{x}_{n+1},\delta\mathsf{x}\right)-\dot{a}_{\mathbf{d}}\left(\mathsf{x}_{n+1},\delta\mathsf{x}\right), \tag{3.71}$$

e a primeira equação do sistema (3.61) para

$$\begin{aligned}\delta a\left(\mathsf{s}_{n+1};\dot{\mathbf{u}}_{n+1},\delta\mathsf{x}\right)+\delta b\left(\mathsf{s}_{n+1};\delta\mathsf{x},\dot{p}_{n+1}\right)-\delta l\left(\mathsf{s}_{n+1};\dot{\mathbf{u}}_{n+1},\delta\mathsf{x}\right) &= \dot{l}_{\mathbf{p}n+1}\left(\mathsf{s}_{n+1},\delta\mathsf{x}\right)-\dot{a}_{\mathbf{p}}\left(\mathsf{x}_{n+1},\delta\mathsf{x}\right)\\ &\quad -\dot{b}_{1\mathbf{p}}\left(\delta\mathsf{x},p_{n+1}\right).\end{aligned} \tag{3.72}$$

As equações de sensibilidade (3.71) e (3.72) são associadas respectivamente às equações iterativas de análise de resposta (2.107) e (2.108).

3.6.2 Análise de Sensibilidade a Forma

Em domínios variáveis, (3.36) é substituída por

$$\delta a\left(\mathsf{x}_{n+1};\dot{\mathbf{u}}_{n+1},\delta\mathsf{x}\right)-\delta l\left(\mathsf{x}_{n+1};\dot{\mathbf{u}}_{n+1},\delta\mathsf{x}\right)=l'_{Vn+1}\left(\delta\mathsf{x}\right)-a'_{V}\left(\mathsf{x}_{n+1},\delta\mathsf{x}\right), \tag{3.73}$$

e a primeira equação do sistema (3.61) por

$$\begin{aligned}\delta a\left(\mathsf{s}_{n+1};\dot{\mathbf{u}}_{n+1},\delta\mathsf{x}\right)+\delta b\left(\mathsf{s}_{n+1};\delta\mathsf{x},\dot{p}_{n+1}\right)-\delta l\left(\mathsf{s}_{n+1};\dot{\mathbf{u}}_{n+1},\delta\mathsf{x}\right) &= l'_{Vn+1}\left(\mathsf{s}_{n+1},\delta\mathsf{x}\right)-a'_{V}\left(\mathsf{x}_{n+1},\delta\mathsf{x}\right)\\ &\quad -b'_{1V}\left(\delta\mathsf{x},p_{n+1}\right).\end{aligned} \tag{3.74}$$

Para determinar $\delta l\left(\mathsf{x}_{n+1};\dot{\mathbf{u}}_{n+1},\delta\mathsf{x}\right)$ e $l'_{Vn+1}\left(\mathsf{s}_{n+1},\delta\mathsf{x}\right)$ é preciso considerar a dimensão do domínio $\mathcal{B}$, a exemplo da Seção 2.2.10.

Domínio Bidimensional

Considere o carregamento de acordo com (2.111)

$$l\left(\mathsf{s},\delta\mathsf{x}\right)=\int_{\Gamma 2t}-\tilde{p}\mathbf{m}\cdot\delta\mathsf{x}\,dL=\int_{r}\tilde{p}\mathbf{n}_{r}\cdot\delta\mathsf{x}\,dr.$$

A derivada total em relação a τ é

$$\frac{d}{d\tau}\left[l\left(\mathsf{s},\delta\mathsf{x}\right)\right]=\left[l\left(\mathsf{s},\delta\mathsf{x}\right)\right]^{\cdot}=\int_{r}\tilde{p}\frac{d}{d\tau}\left(\mathbf{n}_{r}\right)\cdot\delta\mathsf{x}\;dr.$$

Da definição do vetor normal $\mathbf{n}_r$, tem-se que

$$\frac{d}{d\tau}\left(\mathbf{n}_{r}\right)=-\mathbf{e}_{3}\times\frac{d}{d\tau}\left(\mathbf{t}_{r}\right).$$

Logo,

$$\frac{d}{d\tau}\left(\mathbf{t}_{r}\right)=\frac{d}{d\tau}\left(\mathbf{x}_{,r}\right)=\mathbf{V}_{,r}+\dot{\mathbf{u}}_{,r}$$

e

$$\left[l\left(\mathsf{s},\delta\mathsf{x}\right)\right]^{\cdot}=\int_{r}\tilde{p}\begin{bmatrix}0&1\\-1&0\end{bmatrix}\left(\mathbf{V}_{,r}+\dot{\mathbf{u}}_{,r}\right)\cdot\delta\mathsf{x}\;dr=\delta l\left(\mathsf{s};\dot{\mathbf{u}},\delta\mathsf{x}\right)+l_{V}^{\prime}\left(\mathsf{s},\delta\mathsf{x}\right). \tag{3.75}$$

O termo

$$\delta l\left(\mathsf{s};\dot{\mathbf{u}},\delta\mathsf{x}\right)=\int_{r}\tilde{p}\begin{bmatrix}0&1\\-1&0\end{bmatrix}\dot{\mathbf{u}}_{,r}\cdot\delta\mathsf{x}\;dr \tag{3.76}$$

é idêntico a (2.113), apenas substituindo-se $\Delta\mathbf{u}_{,r}$ por $\dot{\mathbf{u}}_{,r}$. Como o carregamento depende do estado de deformação, (3.76) é a medida da sensibilidade de $l\left(\mathsf{s},\delta\mathsf{x}\right)$ à mudança do deslocamento devida à alteração do domínio. Já a parcela

$$l_{V}^{\prime}\left(\mathsf{s},\delta\mathsf{x}\right)=\int_{r}\tilde{p}\begin{bmatrix}0&1\\-1&0\end{bmatrix}\mathbf{V}_{,r}\cdot\delta\mathsf{x}\;dr \tag{3.77}$$

descreve a alteração do carregamento devida à mudança da forma do contorno no qual o carregamento é aplicado. Se o contorno no qual se aplica o carregamento não está parametrizado, o termo (3.77) se anula.

Domínio Tridimensional

Segue-se o mesmo procedimento para domínios tridimensionais, ou seja, a partir da expressão dos carregamentos em termos de variáveis paramétricas

$$l\left(\mathsf{s},\delta\mathsf{x}\right)=\int_{\Gamma 2t}-\tilde{p}\mathbf{m}\cdot\delta\mathsf{x}\;dA=\int_{r}\int_{s}\tilde{p}\mathbf{n}_{rs}\cdot\delta\mathsf{x}\;drds=-\int_{r}\int_{s}\tilde{p}\left(\mathbf{t}_{r}\times\mathbf{t}_{s}\right)\cdot\delta\mathsf{x}\;drds,$$

A partir daí, obtém-se a derivada total em relação a τ

$$\begin{aligned} \left[l\left(\mathsf{s},\delta\mathsf{x}\right)\right]^{\cdot} &= \int_r\int_s \tilde{p}\,\mathrm{Sk}\left[\mathbf{x}_{,s}\right]\left(\mathbf{V}_{,r}+\dot{\mathbf{u}}_{,r}\right)\cdot\delta\mathsf{x}\;drds-\int_r\int_s \tilde{p}\,\mathrm{Sk}\left[\mathbf{x}_{,r}\right]\left(\mathbf{V}_{,s}+\dot{\mathbf{u}}_{,s}\right)\cdot\delta\mathsf{x}\;drds \\ &= \delta l\left(\mathsf{s};\dot{\mathbf{u}},\delta\mathsf{x}\right)+l'_V\left(\mathsf{s},\delta\mathsf{x}\right), \end{aligned} \tag{3.78}$$

sendo

$$\delta l\left(\mathsf{s};\dot{\mathbf{u}},\delta\mathsf{x}\right)=\int_r\int_s \tilde{p}\,\mathrm{Sk}\left[\mathbf{x}_{,s}\right]\dot{\mathbf{u}}_{,r}\cdot\delta\mathsf{x}\;drds-\int_r\int_s \tilde{p}\,\mathrm{Sk}\left[\mathbf{x}_{,r}\right]\dot{\mathbf{u}}_{,s}\cdot\delta\mathsf{x}\;drds \tag{3.79}$$

a variação do carregamento devida à mudança do deslocamento e

$$l'_V\left(\mathsf{s},\delta\mathsf{x}\right)=\int_r\int_s \tilde{p}\,\mathrm{Sk}\left[\mathbf{x}_{,s}\right]\mathbf{V}_{,r}\cdot\delta\mathsf{x}\;drds-\int_r\int_s \tilde{p}\,\mathrm{Sk}\left[\mathbf{x}_{,r}\right]\mathbf{V}_{,s}\cdot\delta\mathsf{x}\;drds \tag{3.80}$$

a variação devida a mudança da geometria do contorno no qual o carregamento é aplicado.

3.7 Considerações de Análise de Sensibilidade em Hiperelasticidade Não-Linear

Em todos os casos anteriores, a sensibilidade da resposta do sistema até o tempo t_{n+1} apenas envolve grandezas em t_{n+1}, independendo dos estados de equilíbrio nos instantes anteriores. Dessa forma, a sensibilidade da resposta final do sistema pode ser determinada através de um pós-processamento da análise de resposta, utilizando apenas os dados da equação de equilíbrio do último passo de carregamento. Isso decorre do fato do problema hiperelástico ser potencial, ou seja, a resposta do sistema pode ser caracterizada completamente apenas pelos estados inicial e final do sistema.

Por envolver apenas a solução de um sistema linear, em oposição ao procedimento iterativo de análise de resposta, torna-se evidente a eficiência do método de análise de sensibilidade para cálculo de gradientes de funcionais quando comparado com métodos de diferenças finitas, especialmente para o nível de precisão alcançado pela análise de sensibilidade.

Capítulo 4

Técnicas de Otimização de Forma de Estruturas

O desenvolvimento de um ambiente de otimização de forma de estruturas requer a integração de diversas ferramentas tais como análise de resposta, análise de sensibilidade, descrição geométrica, geração de malhas, parametrização e algoritmo de minimação. Além das dificuldades práticas de implementação, é necessário detectar na formulação as diversas conexões entre esses vários componentes do problema de otimização e garantir que essas dependências sejam respeitadas.

Nesse sentido, um conceito central de ligação e comunicação são os campos de velocidade de projeto. As expressões de sensibilidade, por exemplo, não enxergam as variáveis de forma diretamente, apenas os campos de velocidade induzidos por elas. Logo, as expressões de sensibilidade impõem requisitos teóricos ao formato dos campos de velocidade. A descrição geométrica, por sua vez, deve fornecer uma regra geral e reutilizável para obtenção desses campos. Deve-se ainda garantir que o contorno da malha coincida sempre com o contorno da geometria em todos os momentos da otimização.

A ligação aparentemente mais simples estaria no mecanismo de comunicação entre o algoritmo de minimização e o restante dos componentes: o algoritmo de minimização apenas recebe valores de funcionais, gradientes e variáveis de projeto, pouco importando seu significado ou forma de obtenção. Ocorre, porém, que o correto funcionamento desse algoritmo depende de que hipóteses de continuidade e suavidade dos funcionais do problema sejam válidas. A garantia prática efetiva dessas hipóteses depende de procedimentos e controles específicos da atualização de malhas devido à modificações do domínio $\mathcal{B}$. Os campos de velocidade também desempenham papel fundamental nesse ponto.

Neste capítulo, são discutidas as dependências e conexões relacionadas campos de velocidades e atualização de malhas. Por último, alguns aspectos do algoritmo de minimização utilizado são apresentados.

4.1 Campos de Velocidades de Projeto

Uma ferramenta genérica de otimização de forma de estruturas requer a integração e coordenação entre modelagem geométrica, discretização do modelo para solução numérica, análise de resposta, análise de sensibilidade e algoritmo de minimização. A coordenação primária da execução das atividades fica a cargo do algoritmo de minimização, porém a integração entre elas é feita pelos campos de velocidades de projeto, definidos em (3.8).

Cada campo de velocidades define a variação de primeira ordem da geometria do domínio corrente associada a uma determinada variável de forma. Considere o vetor de variáveis de projeto $\mathbf{d} \in \Re^n$ parametrizando a forma do domínio corrente $\mathcal{B}$. Essa parametrização define n campos de velocidades sobre o contorno $\partial\mathcal{B}$ da seguinte forma:

$$\begin{aligned} &\tilde{d}_i(\tau): \qquad \tilde{d}_i = d_i + \tau \Delta d_i, \qquad 1, \ldots, n, \\ &\mathbf{V}_i(\mathbf{X}) = \mathbf{V}_s(\mathbf{X}_\tau, \tau)\big|_{\tau=0} = \frac{\partial \mathbf{X}}{\partial \tau} = \mathbf{V}_s\left(\mathbf{X}_{\mathbf{d}}, \tilde{d}_i\right)\Big|_{\tilde{d}_i = d_i} = \frac{\partial \mathbf{X}}{\partial d_i}. \end{aligned} \tag{4.1}$$

Num domínio contínuo, os campos de velocidades associam a parametrização da geometria com as possibilidades de movimentação dos pontos materiais. Num modelo de elementos finitos, a associação é entre parametrização e movimentação dos nós da malha. Em outras palavras, os campos de velocidades conectam geometria e malha de elementos. Por sua vez, as expressões de análise de sensibilidade de funcionais de performance estão associadas às variáveis de projeto apenas pelos campos de velocidades $\mathbf{V}$. Uma expressão como (3.26) mede a sensibilidade do funcional devido à modificação de $\mathcal{B}$ com um dado campo de velocidades. Logo, para um campo $\mathbf{V}_i(\mathbf{X})$, as expressões de sensibilidade fornecem a derivada parcial do funcional em relação a d_i. Por isso, *a previsão linear fornecida pela análise de sensibilidade somente é válida quando a malha de elementos finitos de um novo projeto é obtida perturbando-se a malha corrente utilizando os campos de velocidades.* Deve-se considerar ainda, que a qualidade das malhas produzidas dessa forma influenciam diretamente a precisão dos resultados de análise de resposta e análise de sensibilidade, os quais determinam a taxa de convergência real do algoritmo de minimização.

Entretanto, apesar das expressões de análise de resposta e análise de sensibilidade estarem bem definidas nos pontos materiais do corpo $\mathcal{B}$, o campo de velocidades somente é definido univocamente no contorno $\partial\mathcal{B}$ (Silva, 1997; Silva e Bittencourt, 1999a), não existindo expressão fechada para o interior do domínio. Apesar de seu aspecto central neste contexto, o cálculo de campos de velocidades no interior de $\mathcal{B}$ se baseia apenas em requisitos teóricos decorrentes das expressões de análise de sensibilidade e requisitos práticos obtidos de características da solução numérica por elementos finitos (Choi e Chang, 1994).

4.1.1 Requisitos Teóricos para Campos de Velocidades

Do ponto de vista teórico, cada campo de velocidades deve respeitar dois requisitos:

1. ter a mesma regularidade do campo de deslocamentos e
2. fornecer uma variação linear do domínio $\mathcal{B}$ em relação à respectiva variável de projeto.

A primeira exigência decorre de (3.26) e das expressões de análise de sensibilidade desenvolvidas no Capítulo 3. Observam-se as mesmas exigências de regularidade para o campo de velocidades $\mathbf{V}$ e para o campo de deslocamentos $\mathbf{u}$ em expressões derivadas da equação variacional do problema. Portanto, a análise de sensibilidade requer que o campo de velocidades tenha a mesma regularidade da solução fraca do problema. Dessa forma, para problemas de deformação plana, tensão plana, sólidos axissimétricos e sólidos tridimensionais, deve-se ter $\mathbf{V} \in \mathfrak{H}^1(\mathcal{B})$, ou seja, a primeira derivada deve ser quadrado-integrável. No caso de modelos de viga e casca, $\mathbf{V} \in \mathfrak{H}^2(\mathcal{B})$. Numericamente, esse requisito é cumprido apenas aplicando aos campos de velocidades a mesma interpolação usada no campo de deslocamentos.

O segundo requisito é apenas a exigência da validade de (3.9), ou seja, numa dada configuração de $\mathcal{B}$, o campo de velocidades não deve depender do valor das variáveis de projeto $\mathbf{d}$, mas somente do ponto material $\mathbf{X}$. De acordo com (3.9), uma perturbação da malha de elementos finitos do projeto está relacionada com a perturbação de cada variável de projeto através do campo de velocidades dessa variável agindo como constante de proporcionalidade em cada nó. Malhas obtidas dessa forma são lineares em relação às alterações nas variáveis de projeto e, nesses casos, a previsão de variação de performance fornecida pela análise de sensibilidade apresenta a maior concordância possível com as variações dos valores dos funcionais de performance, independentemente do tamanho da perturbação. Não-linearidades de um campo de velocidades podem se originar já em sua definição sobre o contorno ou em sua extensão para o interior do domínio. Não-linearidades sobre o contorno decorrem da parametrização adotada[1] e, nesse caso, a qualidade da previsão de variação do domínio dependerá do tamanho da perturbação. Em conseqüência, os passos do algoritmo de minimização têm de ser limitados. Por outro lado, dado um campo linear sobre o contorno, a linearidade entre as perturbações do domínio discretizado e das variáveis de projeto é perdida no interior do domínio, por exemplo, quando a nova malha é obtida através de geração automática ou é empregado procedimento de suavização de Laplace (Choi e Chang, 1994; Bugeda e Oliver, 1993; Fancello, 1993). Isso significa que a malha de elementos finitos para um novo domínio deve, sempre que possível, ser obtida combinando-se linearmente as variações em cada componente de $\mathbf{d}$ e os respectivos campos de velocidade. Para pequenas modificações de $\mathcal{B}$, a informação de primeira ordem fornecida pela análise de sensibilidade é dominante e, portanto, o respeito a esta exigência é especialmente importante.

Se essas duas exigências teóricas são respeitadas, soluções analíticas com qualquer campo levariam aos mesmos resultados de sensibilidade. Numericamente, se os campos respeitam tais condições, os resultados serão precisos no sentido padrão de verificação de resultados de análise de sensibilidade (Haug *et al.*, 1986; Santos e Choi, 1988; Kim, 1999; Tseng e Arora, 1989), que consiste em compará-los com derivadas obtidas por diferença finita central na qual as malhas das perturbações do domínio são obtidas com os mesmos campos de velocidades da análise de sensibilidade. Observa-se também que quanto mais refinada a malha, os resultados são menos sensíveis ao tipo de campo de velocidades empregado.

4.1.2 Requisitos Práticos para Campos de Velocidades

Como discutido acima, a discretização do domínio de um novo projeto deve ser obtida, sempre que possível, a partir da discretização corrente aplicando-se (3.9). Além da justificativa teórica, efeitos não-lineares imprevistos nos funcionais de performance do sistema podem surgir de alterações na topologia da malha ou da precisão do gerador de malhas. Observa-se que a precisão aceita por um gerador

[1] Vide Seção 4.1.3.

de malhas externo em seus dados de entrada pode ser limitada e menor que a perturbação imposta ao domínio pelo algoritmo de otimização: se a perturbação do domínio é menor que a precisão do gerador, este passa a fornecer sempre a mesma discretização independentemente da geometria fornecida, causando mudança no comportamento esperado do funcional e perda de continuidade do funcional no limite $\tau \to 0$, como observado na Figura 4.1[2]. Como a informação de sensibilidade passa a não coincidir com as avaliações dos funcionais, a etapa de busca linear do algoritmo de minimização é bastante prejudicada, exigindo grande número de avaliações da performance do sistema, o que está sempre associado à solução de sistemas lineares de grandes dimensões no caso de discretizações em elementos finitos. A ocorrência de buscas lineares com baixa taxa de convergência *efetiva*[3] pode ser considerada como a principal fonte de custo computacional num procedimento de otimização.

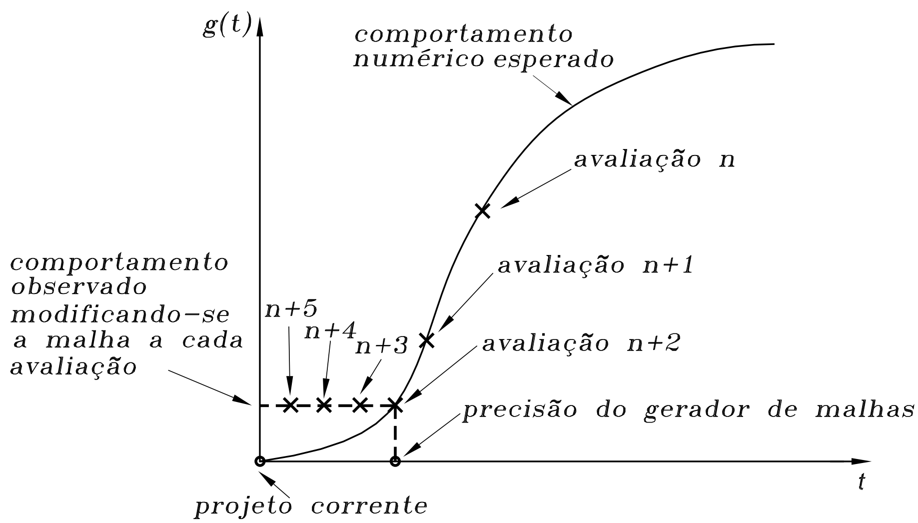


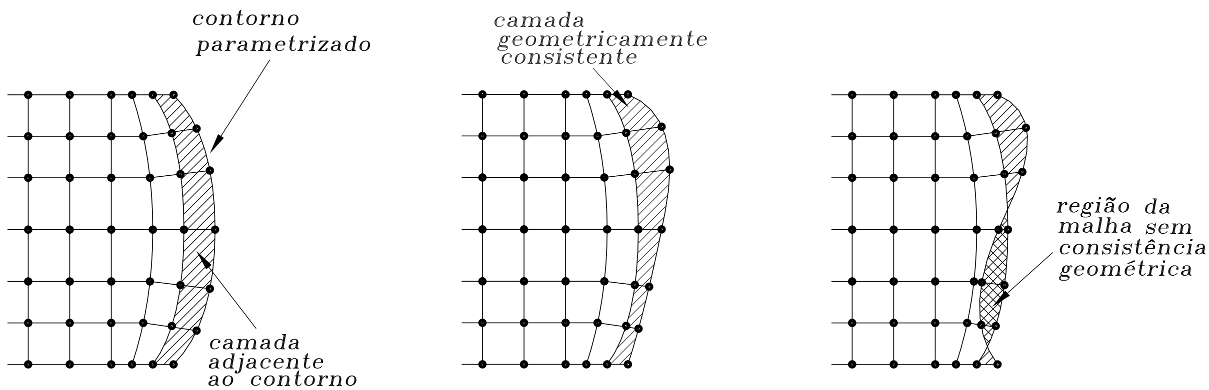
Figura 4.1: Comportamento numérico de funcionais de performance determinados a partir de modelos de elementos finitos.

Dessa forma, o campo de velocidades se torna também uma ferramenta de geração de malhas e, obviamente, o principal requisito prático para qualquer procedimento desse tipo está relacionado à qualidade dos elementos produzidos. Assumindo que a malha inicial tenha qualidade tal que forneça resultados corretos de análise de resposta e sensibilidade, o campo de velocidades deve permitir que essa qualidade não seja degradada para algum intervalo de modificação das variáveis de projeto. É

[2]Um procedimento usual e eficiente de busca linear consiste iniciar-se com um passo grande e seguidamente reduzi-lo sempre que os projetos sejam reprovados nos critérios do algoritmo de minimização (Evsukoff, 1992; Silva, 1997). A perda da continuidade do funcional, torna imprevisível o comportamento de algoritmos de determinação de passo ótimo baseadas na interpolação dos valores dos funcionais na direção de busca.

[3]A taxa de convergência teórica geralmente é determinada considerando funcionais explícitos. No caso de funcionais implícitos associados a resultados de métodos numéricos, a convergência efetiva dependerá do comportamento numérico real da implementação.

claro porém que, qualquer que seja o campo de velocidades, sempre haverá distorção da discretização para grandes alterações do domínio e portanto a necessidade de regeneração da malha.



(a) Malha não perturbada. **(b)** Perturbação menor que a espessura da camada. **(c)** Perturbação maior que a espessura da camada.

Figura 4.2: Perturbações de malha de elementos utilizando campo de velocidades definido em camada unitária de elementos adjacente ao contorno parametrizado.

Quando um gerador automático de malhas está integrado ao ambiente de otimização, a malha pode ser regenerada automaticamente, o que, no entanto, restringe os tipos de elementos que podem ser usados[4]. Se um gerador não está disponível ou não pode ser aplicado, o processo de otimização deve ser interrompido e a malha reconstruída. Campos de velocidades de maior qualidade permitem que os intervalos de modificação das variáveis sejam maiores sem a necessidade de regeneração da malha, sendo esse o caso de campos de velocidades definidos em todo interior do domínio (Yao e Choi, 1989). Campos definidos em apenas uma camada de elementos adjacente ao contorno parametrizado (Fancello, 1993; Seong e Choi, 1987; Silva, 1997) permitem apenas alterações da ordem da espessura da camada (observar as Figuras 4.2(a) e 4.2(b)). É importante observar, entretanto, que a possibilidade – maior ou menor – de distorcer os elementos da malha ao se aplicar (3.9) não tem nenhuma influência na precisão dos cálculos de sensibilidade, os quais são efetuados na configuração corrente e portanto antes de qualquer modificação na malha.

Uma outra exigência está relacionada com a manutenção da consistência do domínio do proble-

[4]Geradores automáticos geralmente estão restritos a tetraedros ou à combinação de triângulos e quadriláteros em domínios bidimensionais. Entretanto, modelos de estruturas reais muitas vezes incluem associações de barras, vigas e equações de restrição, além dos elementos bi e tridimensionais. Tais associações dificilmente podem ser reconstruídas sem interação intensiva do usuário.

ma, ou seja, a seqüência de transformações do domínio discretizado deve coincidir com a seqüência de transformações do domínio contínuo. Formalmente, a transformação de domínio T_τ induzida pelo campo de velocidades definida em (3.3), deve ser tal que $\mathring{\mathcal{B}} \rightarrow \mathsf{T}_\tau(\mathring{\mathcal{B}})$ e $\partial\mathcal{B} \rightarrow \mathsf{T}_\tau\,(\partial\mathcal{B})$ seja respeitado em todos nós da malha, sendo de especial importância que os nós do contorno da malha sempre se mantenham sobre o contorno geométrico. Considerando que $\partial\mathcal{B}$ é definido de forma paramétrica, uma solução possível é impor que nós sobre o contorno mantenham sempre as mesmas coordenadas paramétricas durante as iterações de otimização (Choi e Chang, 1994). Observa-se, porém, que isso implica em restringir os movimentos dos nós localizados sobre o contorno, limitando as possibilidades de parametrização do problema de otimização. Por outro lado, empregando entidades NURBS na definição de $\partial\mathcal{B}$, adotando como variáveis as coordenadas de seus pontos de controle e aplicando técnica de recuperação de coordenadas paramétricas, tem-se recursos bastante flexíveis de definição de variáveis de projeto associadas à geometria. Garante-se ainda coincidência entre malha e geometria em $\partial\mathcal{B}$ através da própria formulação de campos de velocidade[5], sem a necessidade de se impor controles sobre os nós do contorno.

Uma preocupação adicional de consistência, não relacionada diretamente aos campos de velocidades, consiste em garantir que T_τ não degenere o domínio $\mathcal{B}$. A adoção de uma parametrização correta de $\mathcal{B}$, incluindo-se aí a definição conveniente de limites máximos e mínimos para as variáveis de projeto é o procedimento recomendado para se evitar as ocorrências de perda de consistência geométrica do domínio contínuo. Se a perda de consistência geométrica ocorre devido a incorreções na parametrização, torna-se impossível prosseguir com o procedimento de otimização. Porém, se a geometria continua consistente e apenas houve perda de consistência na malha, ainda é possível continuar a seqüência de otimização se houver recursos automáticos de reconstrução da malha, como observado na Figura 4.2(c), na qual a perda de consistência ocorre apenas na discretização.

Também do ponto de vista do desenvolvimento de um ambiente de otimização estrutural, é fundamental que a técnica de geração de campos de velocidades no interior de $\mathcal{B}$ seja totalmente automática, não exigindo nenhuma interação com o usuário durante o ciclo de otimização. A determinação de campos de velocidades deve ser então baseada num algoritmo genérico e reutilizável cujas entradas sejam somente a discretização de $\mathcal{B}$ e os campos de velocidades no contorno em toda seqüência de geometrias produzidas num processo de otimização. Por último, busca-se também eficiência computacional como

[5] Seção 4.1.3.

em qualquer procedimento numérico.

4.1.3 Geração de Campos de Velocidades no Contorno

Neste trabalho, curvas e superfícies NURBS (*Non-Uniform Rational B-Splines*) (Rogers e Adams, 1990) são aplicadas na definição da geometria de estruturas bi e tridimensionais para análise de resposta, análise de sensibilidade e otimização de forma.

Curvas NURBS

As coordenadas de um ponto sobre uma curva B-spline racional de $q+1$ pontos de controle de coordenadas $\mathbf{X}^i = \left\{X_1^i, X_2^i, X_3^i\right\}^T$ são dadas por

$$\boxed{\mathbf{X}(r) = \sum_{i=1}^{q+1} \mathbf{X}^i R_{i,\chi}(r), \qquad R_{i,\chi}(r) = \frac{\beta_i N_{i,\chi}(r)}{\displaystyle\sum_{j=1}^{q+1} \beta_j N_{j,\chi}(r)}.} \tag{4.2}$$

O conjunto $\{R_{i,\chi}(r)\}$ é a base B-spline racional e os β_i's são os pesos associados aos pontos de controle. Os pesos e as coordenados físicas de um ponto pertencem a um espaço de coordenadas de quatro dimensões. A expressão (4.2) define uma razão de polinômios de ordem χ (grau $\chi - 1$). A quantidade adimensional r é o parâmetro interno da curva.

A base B-spline $\{N_{i,\chi}(r)\}$ pode ser avaliada pela fórmula recursiva de Cox-de Boor (Rogers e Adams, 1990),

$$\boxed{\begin{array}{l} \begin{cases} N_{i,1}(r) = 1, & \kappa_i \leq r \leq \kappa_{i+1}, \\ N_{i,1}(r) = 0, & \text{outros casos,} \end{cases} \\ \\ N_{i,k}(r) = \dfrac{(r-\kappa_i)\, N_{i,k-1}(r)}{\kappa_{i+k-1}-\kappa_i} + \dfrac{(\kappa_{i+k}-r)\, N_{i+1,k-1}(r)}{\kappa_{i+k}-\kappa_{i+1}}, \qquad k = 2, \ldots, \chi, \end{array}} \tag{4.3}$$

sendo os κ_i's elementos de um *vetor de nodos* $\boldsymbol{\kappa}$ com dimensão $q + \chi + 1$.

O vetor de nodos deve ser uma série monotonicamente crescente de números reais. Três tipos de vetores de nodos são geralmente usados: periódico (uniforme), uniforme aberto e não-uniforme. Um vetor de nodos periódico tem valores que são igualmente espaçados e distribuídos entre 0 e algum valor máximo com incrementos de 1. Esse tipo é usado para gerar curvas fechadas. Vetores uniformes abertos são gerados pela regra

$$\boxed{\begin{cases} \kappa_i = 0, & 1 \leq i \leq \chi, \\ \kappa_i = i - \chi, & \chi + 1 \leq i \leq q + 1, \\ \kappa_i = q - \chi + 2, & q + 2 \leq i \leq q + \chi + 1. \end{cases}} \tag{4.4}$$

Vetores não-uniformes podem tanto ter valores com espaçamento desigual e/ou valores internos múltiplos. Geralmente os vetores de nodos são normalizados para o intervalo $[0, 1]$.

Rigorosamente, uma curva NURBS é uma curva B-spline racional gerada com vetor de nodos não-uniforme. Todas as curvas B-splines racionais e não-racionais podem ser denominadas NURBS, uma vez que esta é a forma mais geral de B-spline.

Superfícies NURBS

Superfícies NURBS são obtidas da generalização dos conceitos anteriores para as coordenadas bi-paramétricas (r, s). Um superfície B-spline racional é dada por

$$\boxed{\mathbf{X}(r,s) = \sum_{i=1}^{q+1}\sum_{j=1}^{t+1} \mathbf{X}^{ij} S_{ij}(r,s), \qquad S_{ij}(r,s) = \frac{\beta_{ij} N_{i,\chi}(r) M_{j,\vartheta}(s)}{\displaystyle\sum_{k=1}^{q+1}\sum_{l=1}^{t+1} \beta_{kl} N_{k,\chi}(r) M_{l,\vartheta}(s)},} \tag{4.5}$$

sendo os $\mathbf{X}^{ij}$'s vértices de uma rede poligonal tridimensional e os $S_{ij}(r,s)$'s funções de base racionais de superfície. $N_{i,\chi}(r)$ e $M_{j,\vartheta}(s)$ são funcões de base B-spline não-racionais dadas pela fórmula recursiva de Cox-de Boor em cada direção paramétrica utilizando vetores de nodos periódicos, uniformes abertos ou não-uniformes. Observa-se que as fronteiras de superfícies NURBS são curvas NURBS.

Campos de Velocidades

Assumindo que a variável de forma é uma coordenada l $(l = 1, 2, 3)$ de um ponto de controle *interno* de uma superfície NURBS (4.5), ou seja, $d_i = X_l^{ij}$, tem-se

$$\frac{\partial \mathbf{X}}{\partial d_i} = \frac{\partial \mathbf{X}}{\partial X_l^{ij}}, \tag{4.6}$$

sendo

$$\frac{\partial \mathbf{X}}{\partial X_1^{ij}} = \left\{ \begin{array}{c} S_{ij}(r,s) \\ 0 \\ 0 \end{array} \right\}, \qquad \frac{\partial \mathbf{X}}{\partial X_2^{ij}} = \left\{ \begin{array}{c} 0 \\ S_{ij}(r,s) \\ 0 \end{array} \right\}, \qquad \frac{\partial \mathbf{X}}{\partial X_3^{ij}} = \left\{ \begin{array}{c} 0 \\ 0 \\ S_{ij}(r,s) \end{array} \right\}.$$

Se a variável é o peso ij, i.e., $d_i = \beta_{ij}$, então

$$\frac{\partial \mathbf{X}}{\partial d_i} = \frac{\partial \mathbf{X}}{\partial \beta_{ij}} = \frac{1}{\beta_{ij}} \left[\mathbf{X}^{ij} - \mathbf{X}(r,s) \right] S_{ij}(r,s)\,. \tag{4.7}$$

Em ambos os casos, o campo de velocidades no contorno do modelo discreto é obtido avaliando-se (4.6) e (4.7) nas coordenadas paramétricas (r,s) de cada nó da superfície.

A relação linear entre (4.6) e (4.5) mostra que parametrizações baseadas em pontos de controle dão origem a campos de velocidades que, quando utilizados na atualização de coordenadas nodais, garantem que nós sobre o contorno $\partial\mathcal{B}$ permanecerão sempre sobre esse contorno uma vez que nós e geometria serão atualizados exatamente com a mesma expressão independentemente do tamanho da perturbação.

Considere, por exemplo, a parametrização de uma coordenada do ponto de controle $\mathbf{X}^{mn}$. Logo,

$$\mathbf{X}(r,s) + \delta d \mathbf{V}(r,s) = \left\{ \begin{array}{ll} \sum_{i=1}^{q+1} \sum_{j=1}^{t+1} \breve{\mathbf{X}}^{ij} S_{ij}(r,s)\,, & \\ & \\ \breve{\mathbf{X}}^{ij} = \mathbf{X}^{ij}, & i \neq m,\ j \neq n, \\ \breve{\mathbf{X}}^{ij} = \mathbf{X}^{ij} + \delta d, & i = m,\ j = n. \end{array} \right.$$

O lado esquerdo da expressão anterior representa o algoritmo de atualização das coordenadas dos nós sobre o contorno, ou seja, a cada posição nodal soma-se a respectiva velocidade multiplicada pela perturbação. O lado direito da mesma expressão indica a regra de atualização da representação geométrica.

Por outro lado, (4.7) e (4.5) se relacionam de modo não-linear significando que atualizações de nós por meio de campos de velocidades apenas aproximarão a geometria atualizada quando a parametrização envolver pesos. Essa aproximação será tanto pior quanto maior for a perturbação. Por esse motivo, parametrizações de pesos de curvas e superfícies NURBS não serão adotadas neste trabalho. Representações baseadas em geometria variacional, tais como as apresentadas em (Lindby e Santos, 1997; Chen e Tortorelli, 1997), têm a mesma característica. Nesses casos, é necessário limitar o tamanho da perturbação a cada passo do algoritmo de otimização.

No caso da variável estar associada a um ponto de controle de uma curva NURBS que é a fronteira entre duas superfícies NURBS, qualquer modificação se propaga para ambas as superfícies. Tais modificações são descritas por campos de velocidades parciais $\mathbf{V}_1$ and $\mathbf{V}_2$ em cada uma das superfícies. O campo de velocidades resultante é a união $\mathbf{V} = \mathbf{V}_1 \cup \mathbf{V}_2$. Da mesma forma, se a variável está associada a um ponto que é um vértice de n_{surf} superfícies, o campo de velocidades resultante é a união de campos parciais, ou seja, $\mathbf{V} = \mathbf{V}_1 \cup \mathbf{V}_2 \cup \ldots \cup \mathbf{V}_{n_{surf}}$.

Componentes mecânicos podem também apresentar características, tais como simetria, que devem ser mantidas durante a otimização. Isso é feito através da correta definição das variáveis de projeto e dos campos de velocidades. Por exemplo, para uma relação de simetria entre duas variáveis, o campo de velocidades é $\mathbf{V} = \mathbf{V}_1 + (-\mathbf{V}_2)$. Para relações de igualdade de n_{var} variáveis de forma, o campo de velocidades é $\mathbf{V} = \mathbf{V}_1 + \mathbf{V}_2 + \ldots + \mathbf{V}_{n_{var}}$.

As expressões anteriores são todas análogas para curvas NURBS.

Recuperação de Coordenadas Paramétricas

As expressões (4.6) e (4.7) mostram que o campo de velocidades de variáveis relacionadas a características de curvas e superfícies NURBS exigem o cálculo das funções de base nos nós localizados sobre o contorno. Para isso, torna-se necessário conhecer as coordenadas paramétricas (r, s) correspondentes a cada nó sobre o contorno $\partial\mathcal{B}$. Tais informações não são geralmente fornecidas pelos geradores de malha, sendo necessário dispor de recursos para determiná-las. Além disso, mesmo que os parâmetros sejam conhecidos, a aplicação da transformação de domínio (3.3) altera seus valores, os quais precisam ser atualizados antes da avaliação dos campos de velocidades.

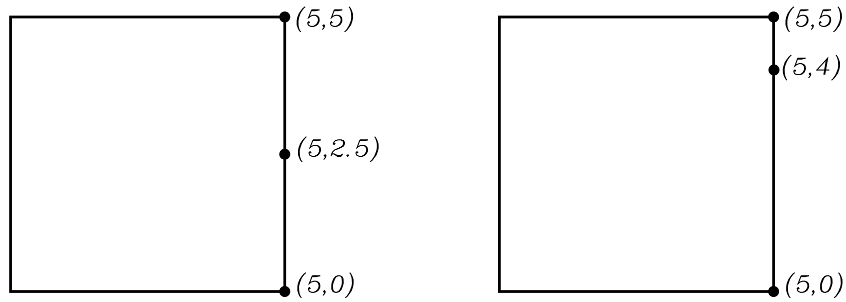
Para geometrias bidimensionais, uma abordagem inicial seria a aproximação linear, baseada na aproximação dos parâmetros r a partir das distâncias entre os nós sobre o contorno. Considere a distância entre dois nós consecutivos do contorno

$$\delta_i = \|\mathbf{X}_i - \mathbf{X}_{i-1}\| ,$$

sendo $\mathbf{X}_i$ as coordenadas do nó i. A aproximação $\tilde{r}_i$ do parâmetro r_i correspondente seria então,

$$\tilde{r}_i = \left(\sum_{k=1}^{i} \delta_k\right) / \left(\sum_{k=1}^{Nc} \delta_k\right), \qquad \tilde{r}_0 = 0.0, \tag{4.8}$$

sendo Nc o número total de nós na região considerada do contorno.



(a) Ponto de controle central em posição simétrica.

(b) Ponto de controle central deslocado.

Figura 4.3: Exemplos para teste de cálculo de curvas NURBS.

Essa aproximação, entretanto, mostra-se bastante imprecisa, dando origem a erros sistemáticos de análise de sensibilidade mesmo em casos simples como se pode observar no exemplo a seguir. Considere o domínio quadrado mostrado na Figura 4.3(a), no qual um dos lados é descrito por uma B-spline quadrática, estando as coordenadas dos 3 pontos de controle indicadas na mesma figura. Aplicando (4.4), o vetor de nodos é

$$\boldsymbol{\kappa} = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}.$$

A partir de (4.3) são obtidas as funções de base

$$\begin{array}{lllll} N_{1,1} = 0, & N_{2,1} = 0, & N_{3,1} = 1, & N_{4,1} = 0, & N_{5,1} = 0, \\ N_{1,2} = 0, & N_{2,2} = 1 - r, & N_{3,2} = r, & N_{4,2} = 0, & \\ N_{1,3} = (1-r)^2, & N_{2,3} = 2r(1-r), & N_{3,3} = r^2. & & \end{array}$$

A última linha é aplicada em (4.2), originando a seguinte expressão seguinte das coordenadas dos pontos sobre o lado do domínio indicado na Figura 4.3(a):

$$\mathbf{x}(r) = \left\{ \begin{array}{c} 5 \\ 0 \end{array} \right\} (1-r)^2 + \left\{ \begin{array}{c} 5 \\ 2.5 \end{array} \right\} 2r(1-r) + \left\{ \begin{array}{c} 5 \\ 5 \end{array} \right\} r^2 = \left\{ \begin{array}{c} 5 \\ 5r \end{array} \right\}, \qquad 0 \leq r \leq 1.$$

A derivada da área ψ do quadrado em relação à variável d, coordenada x do segundo ponto de controle, é então,

$$\frac{\partial \psi}{\partial d} = \int_{\mathcal{B}} \mathrm{Div}\, \mathbf{V} \; dV = \int_{\partial \mathcal{B}} \mathbf{V} \cdot \mathbf{n} \; dL = \int_0^1 2r(1-r)(5) \; dr = \frac{10}{6} = 1.666\ldots,$$

pois o campo de velocidades no contorno é a própria segunda função de base da curva. Utilizando (4.8), obtem-se o mesmo resultado utilizando-se elementos triangulares quadráticos (que descrevem exatamente o campo de velocidades no contorno): $\frac{\partial \psi}{\partial d} = 1.666666667$. Porém, se o segundo ponto de controle for alterado, mantendo-se o mesmo contorno final (como mostrado na Figura 4.3(b)), as coordenadas dos pontos desse lado passam a ser fornecidas pela expressão

$$\mathbf{x}\left(r\right)=\left\{\begin{array}{c}5\\0\end{array}\right\}\left(1-r\right)^{2}+\left\{\begin{array}{c}5\\4.0\end{array}\right\}2r\left(1-r\right)+\left\{\begin{array}{c}5\\5\end{array}\right\}r^{2}=\left\{\begin{array}{c}5\\8r-3r^{2}\end{array}\right\},\qquad 0\leq r\leq 1.$$

Observa-se que o campo de velocidades também é fornecido pela segunda função de base da curva, portanto espera-se o mesmo resultado. De fato,

$$\frac{\partial \psi}{\partial d}=\int_{0}^{1}2r\left(1-r\right)\left(8-6r\right)\ dr=\frac{10}{6}=1.666\ldots$$

O resultado numérico foi então de $\frac{\partial \psi}{\partial d} = 1.666585859$ ao invés do resultado exato e, quanto mais deslocado em relação ao centro estiver este ponto de controle, maior o erro obtido. Além disso, trata-se de um erro não controlável e sistemático que se torna maior com a modificação da posição dos pontos de controle, geometrias mais complexas e outros funcionais de performance. O impacto é ainda mais significativo quando o próprio campo de velocidades é usado para atualizar os nós da malha.

Tem-se então o problema de determinar o valor $\tilde{r}_i$ tal que

$$\left\{\begin{array}{c}\tilde{x}_i\\\tilde{y}_i\end{array}\right\}=\sum_{j=1}^{q+1}\mathbf{X}^{j}R_{j,\chi}\left(\tilde{r}_i\right)$$

sendo $\mathbf{X}^j$ as coordenadas dos pontos de controle e $\left[\begin{array}{cc}\tilde{x}_i & \tilde{y}_i\end{array}\right]^T$ as coordenadas do nó. Assim, deve-se resolver a equação vetorial

$$\mathbf{f}\left(r_i\right)=\sum_{j=1}^{q+1}\mathbf{X}^{j}R_{j,\chi}\left(r_i\right)-\left\{\begin{array}{c}\tilde{x}_i\\\tilde{y}_i\end{array}\right\}=\mathbf{0},\qquad D\mathbf{f}\left(r_i\right)=\sum_{j=1}^{q+1}\mathbf{X}^{j}\frac{\partial}{\partial r}R_{j,\chi}\left(r_i\right).$$

Essa equação será resolvida através de iterações de Newton-Raphson.

A aproximação de Taylor de primeira ordem para $\mathbf{f}\left(r\right)$ em torno de r_i^*,

$$\mathbf{f}\left(r_i\right)=\mathbf{f}\left(r_i^*\right)+\left\{D\mathbf{f}\left(r_i^*\right)\right\}\left(r_i-r_i^*\right)=\mathbf{0},\tag{4.9}$$

é o ponto de partida para uma forma algorítmica de solução.

Multiplicando (4.9) por $\left\{D\mathbf{f}\left(r_i^*\right)\right\}^T$,

$$\left\{D\mathbf{f}\left(r_i^*\right)\right\}^T\left\{D\mathbf{f}\left(r_i^*\right)\right\}\left(r_i-r_i^*\right)+\left\{D\mathbf{f}\left(r_i^*\right)\right\}^T\mathbf{f}\left(r_i^*\right)=\mathbf{0},$$

converte-se a equação vetorial em escalar. Isolando r_i, ou seja,

$$r_i = r_i^* - \frac{\{D\mathbf{f}\left(r_i^*\right)\}^T \mathbf{f}\left(r_i^*\right)}{\{D\mathbf{f}\left(r_i^*\right)\}^T \{D\mathbf{f}\left(r_i^*\right)\}},$$

obtém-se a fórmula recursiva para

$$\boxed{r_i^{k+1} = r_i^k - \frac{\left\{D\mathbf{f}\left(r_i^k\right)\right\}^T \mathbf{f}\left(r_i^k\right)}{\{D\mathbf{f}\left(r_i^k\right)\}^T \{D\mathbf{f}\left(r_i^k\right)\}}, \quad k+1 \rightarrow k, \quad \text{até que } \left|r_i^{k+1} - r_i^k\right| \leq \varepsilon > 0.} \tag{4.10}$$

Se houve convergência para $k+1$, adota-se $\tilde{r}_i = r_i^{k+1}$.

Este procedimento deve ser realizado para cada nó da região do contorno associada a variáveis de projeto. Entretanto, este algoritmo é eficiente pois não há necessidade de resolver sistemas lineares uma vez que o problema para cada nó é desacoplado dos demais e todas as funções envolvidas são conhecidas (as funções de base e suas derivadas são obtidas de maneira semelhante). Testes numéricos têm mostrado que o algoritmo converge rapidamente mesmo para aproximação $r_i^k = 0.0$ para todos os nós de uma curva.

Aplicando este procedimento ao exemplo do quadrado, obtem-se $\frac{\partial \psi}{\partial d} = 1.666666667$ para qualquer coordenada y, $0.0 < y < 5.0$, do segundo ponto de controle.

O método anterior é indispensável em problemas tridimensionais pois em superfícies não há ordenação preferencial dos nós que permita mesmo uma aproximação linear para as coordenadas paramétricas. Nesse caso, o algoritmo envolve a solução de um sistema linear 2×2 simétrico em cada nó da região do contorno associada a variáveis de projeto.

4.1.4 Geração de Campos de Velocidades do Interior do Domínio

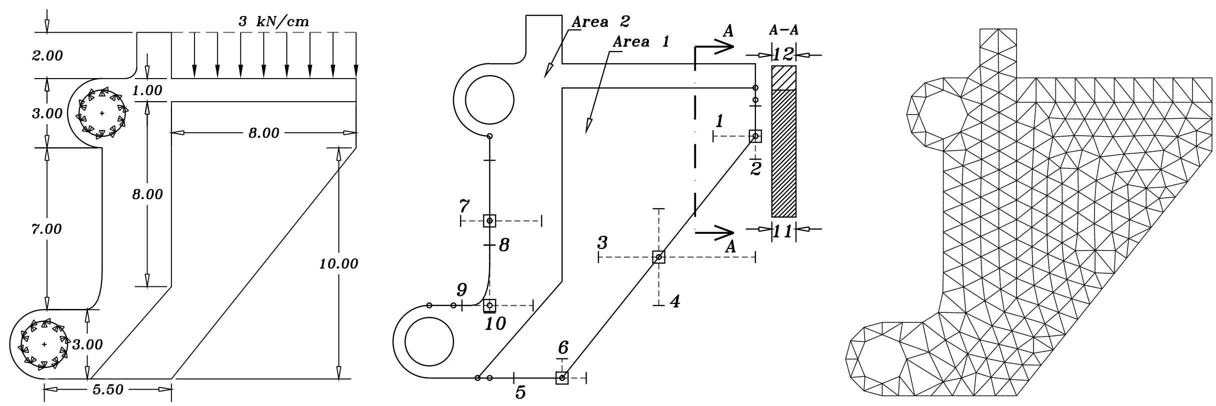
Duas técnicas de geração de campos de velocidades no interior de domínios discretizados em elementos finitos foram aplicadas na implementação dos algoritmos de análise de sensibilidade e otimização: *método da camada unitária de contorno* (Silva, 1997) e *método de deslocamentos fictícios do contorno* (Choi e Chang, 1994). Os requisitos teóricos são cumpridos por campos obtidos por ambas as técnicas e, dentro de certo intervalo de perturbação, também são cumpridos os requisitos práticos.

Método da Camada Unitária de Contorno

Neste método, o campo de velocidades é não-nulo somente na primeira camada de elementos adjacentes ao contorno parametrizado.

A partir de um campo de velocidades conhecido nos nós do contorno, o campo de velocidades nos nós da camada adjacente é obtida através de interpolação com funções de forma lineares das velocidades do contorno e de velocidades nulas na fronteira interna da camada. É importante observar que a interpolação linear somente ocorre na direção da espessura da camada. Sobre o contorno, o campo de velocidades tem a mesma ordem da interpolação da malha de elementos.

Considerando o projeto parametrizado e discretizado em elementos finitos da Figura 4.4, tem-se os campos de velocidades sobre o contorno e no interior do domínio mostrados na Figura 4.5 comparados com os campos obtidos com o método de deslocamentos fictícios do contorno descrito na próxima seção.



(a) Geometria e condições de contorno.

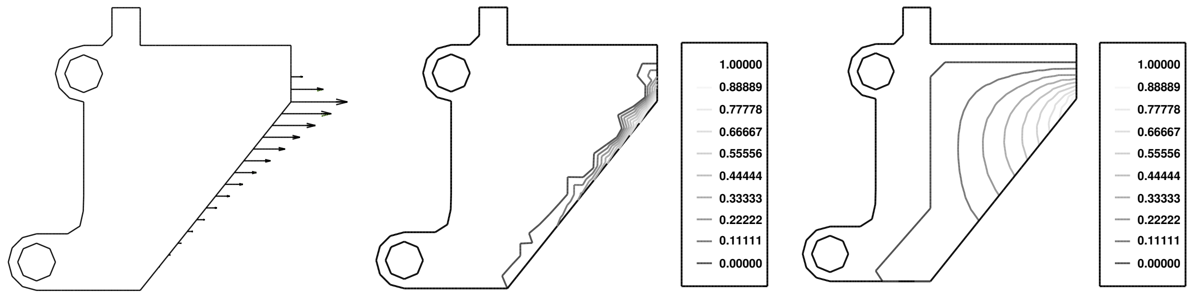
(b) Variáveis de projeto controlando forma e espessura.

(c) Discretização em triângulos quadráticos.

Figura 4.4: Projeto parametrizado para análise de sensibilidade e otimização.

O principal ponto positivo desse método é sua eficiência computacional uma vez que não envolve nenhum cálculo numérico. Além disso, as expressões de análise de sensibilidade somente precisam ser avaliadas na camada adjacente ao contorno, o que é bastante vantajoso em geometrias tridimensionais.

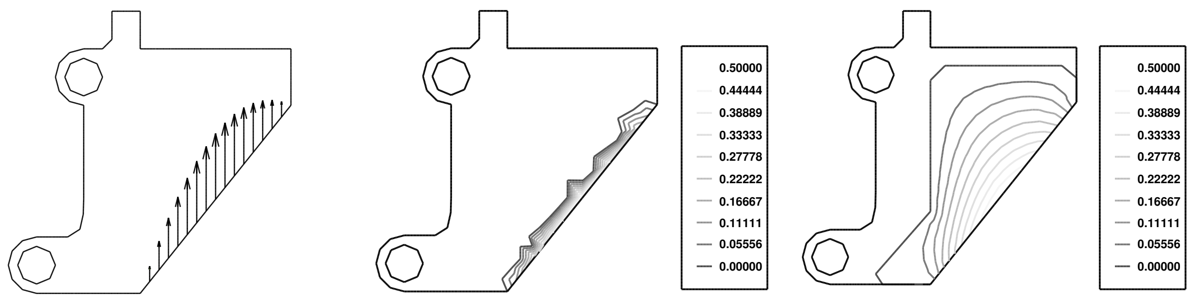
Contudo, ao se concentrar toda a possibilidade de variação do domínio numa única camada de elementos, há bastante probabilidade de que sejam produzidos elementos distorcidos ou mesmo perda



(a) Variável 1: velocidades sobre o contorno.

(b) Interior com método da camada unitária de contorno.

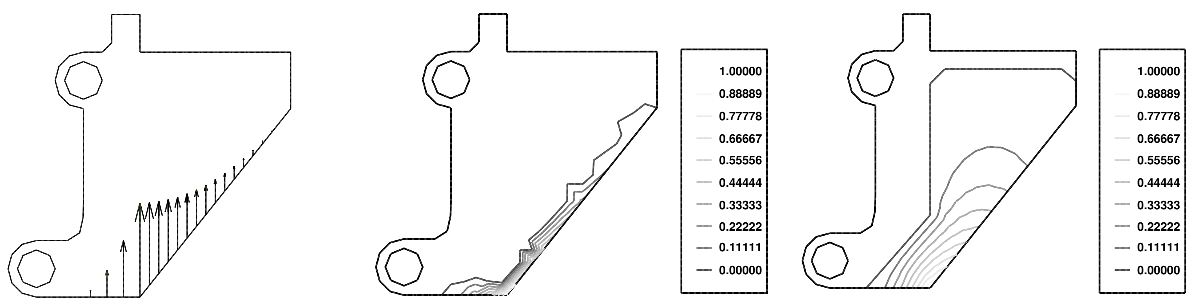
(c) Interior com método de deslocamentos fictícios de contorno.



(d) Variável 4: velocidades sobre o contorno.

(e) Interior com método da camada unitária de contorno.

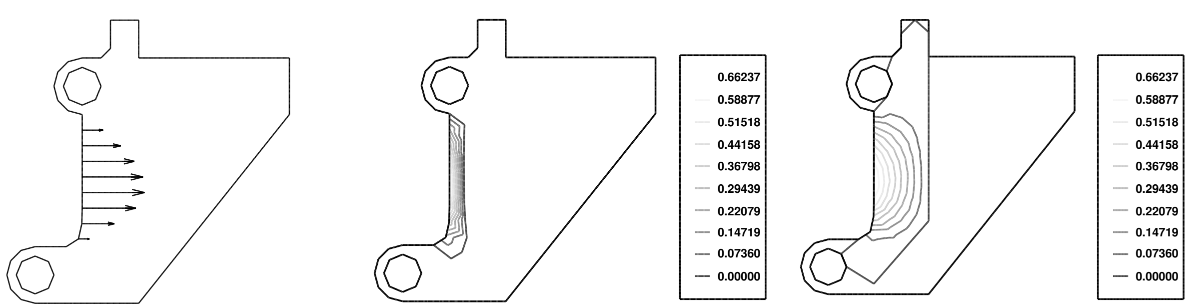
(f) Interior com método de deslocamentos fictícios de contorno.



(g) Variável 6: velocidades sobre o contorno.

(h) Interior com método da camada unitária de contorno.

(i) Interior com método de deslocamentos fictícios de contorno.



(j) Variável 9: velocidades sobre o contorno.

(k) Interior com método da camada unitária de contorno.

(l) Interior com método de deslocamentos fictícios de contorno.

Figura 4.5: Campos de velocidades devido à parametrização da Figura 4.4(b).

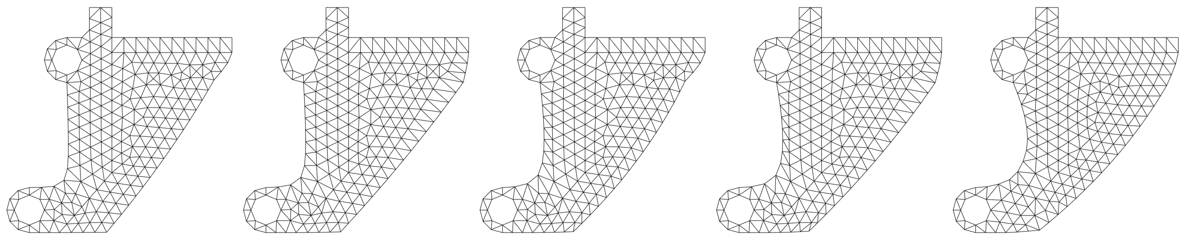
da consistência geométrica da malha ao se aplicar (3.9), como na Figura 4.2(c). E quanto mais refinada for a malha adjacente ao contorno, maior a incidência de tais efeitos. Ambos os problemas, porém, não são um impedimento para a aplicação deste método, pois é possível detectá-los numericamente e interromper o processo de otimização ou aplicar um gerador de malhas na reconstrução da discretização. Quando a malha perde sua consistência geométrica, o jacobiano da transformação global-local de algum elemento da camada de contorno torna-se negativo. A distorção de elementos, por sua vez, pode ser indicada por medidas de erro *a priori* baseadas na geometria dos elementos[6].

Na Figura 4.6, apresenta-se a seqüência de iterações de otimização com o objetivo de minimizar o volume considerando as definições da Figura 4.4 e método de camada unitária de contorno para geração de campos de velocidades. Devido à distorção dos elementos ou perda de consistência da discretização, o gerador de malhas integrado ao ambiente de otimização estrutural foi acionado automaticamente para reconstruir a discretização nas iterações 1, 5, 7 e 9. Nas demais, apenas a informação dos campos de velocidades foi suficiente para atualizar a malha para a iteração seguinte.

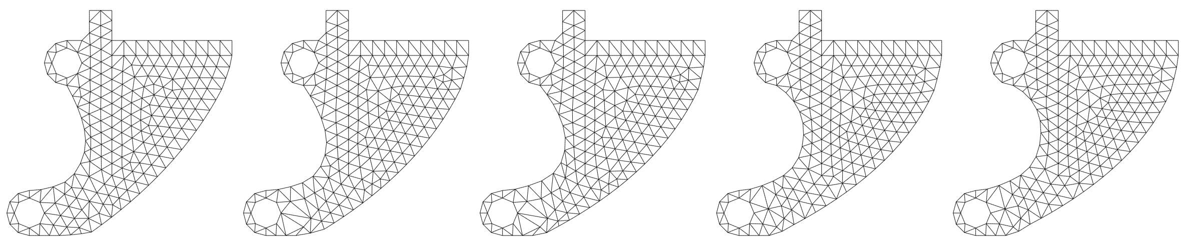
Os resultados de sensibilidade mostrados nas Tabelas 4.1 a 4.4 demonstram a precisão alcançada aplicando-se os campos obtidos pelo método de camada unitária de contorno. A precisão é bastante alta uma vez que o campo é obtido sem cálculo numérico, eliminado-se a possibilidade de acúmulo de erros. A precisão de análise de sensibilidade é verificada comparando-se seus resultados com os valores obtidos por diferença finita central em que as perturbações são obtidas por (3.9) utilizando os mesmos campos de velocidades usados na avaliação das expressões de sensibilidade. Esse procedimento permite comparar a previsão de comportamento fornecida pela medida de sensibilidade com o comportamento real observado em pequenas perturbações da malha.

Por combinar simplicidade, precisão e eficiência computacional, o método de camada unitária de contorno para cálculo de campos de velocidades é o mais atrativo. Entretanto, para se obter um ambiente automático de otimização de estruturas, é necessário dispor de recursos de geração automática de malhas integrados e que a modelagem do problema em elementos finitos possa ser obtida também de forma automática. Caso contrário, é necessário aplicar algoritmos de geração de campos de velocidades como o descrito a próxima seção.

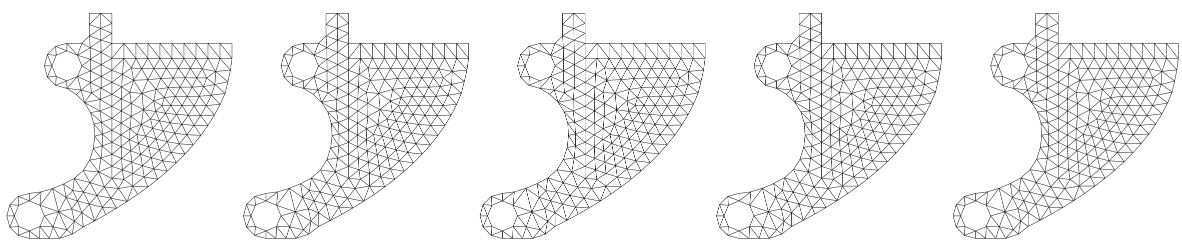
[6] Seção 4.2.1.



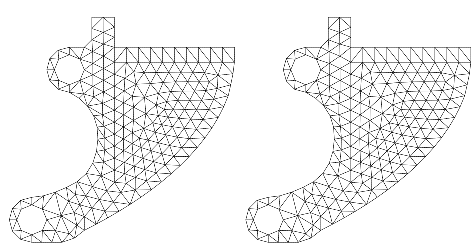
(a) Iteração 1. **(b)** Iteração 2. **(c)** Iteração 3. **(d)** Iteração 4. **(e)** Iteração 5.



(f) Iteração 6. **(g)** Iteração 7. **(h)** Iteração 8. **(i)** Iteração 9. **(j)** Iteração 10.



(k) Iteração 11. **(l)** Iteração 12. **(m)** Iteração 13. **(n)** Iteração 14. **(o)** Iteração 15.



(p) Iteração 16. **(q)** Iteração 17.

Figura 4.6: Seqüência de otimização do problema definido na Figura 4.4 com campos de velocidades gerados com o método de camada unitária de contorno (volume inicial: 114.563 cm^3; volume final: 69.499 cm^3). A malha é reconstruída com gerador de malhas nas iterações 1, 5, 7, 9. Nas demais iterações, o campo de velocidades é utilizado para atualizar a dicretização.

Tabela 4.1: Gradientes do funcional de volume (114.5626887 cm^3) avaliado no problema da Figura 4.4. Derivadas parciais de variáveis de forma são avaliadas com dois tipos de campos de velocidades e comparados com valores obtidos por diferenças finitas. AS: análise de sensibilidade. MDF: diferenças finitas usando o respectivo campo de velocidades. A concordância considera o valor MDF como padrão.

Método da Camada Unitária de Contorno			Método de Deslocamentos Fictícios do Contorno		
AS	**MDF**	**Concordância %**	**AS**	**MDF**	**Concordância %**
4.1666663111	4.1666666675	99.999991447147	4.1641612964	4.1666666675	99.939871093052
-2.6666666667	-2.6666666667	99.999999999997	-2.6683978726	-2.6666666667	99.935079776217
3.3333333333	3.3333333333	99.999999999997	3.3333960870	3.3333333333	99.998117388763
-2.6666666667	-2.6666666667	100.000000000000	-2.6667928271	-2.6666666667	99.995268984471
3.3333333333	3.3333333333	99.999999999999	3.3342553455	3.3333333333	99.972339636020
-3.9173091868	-3.9173115371	99.999940000273	-3.9159820977	-3.9173115371	99.966062452016
-1.8968325399	-1.8968343011	99.999907145649	-1.8942774906	-1.8968343011	99.865206435523
0.7714256686	0.7714257012	99.999995776876	0.7799399212	0.7714257012	98.896300705911
-2.9159447583	-2.9159441927	99.999980604524	-2.9083487776	-2.9159441927	99.739521244785
0.0622272689	0.0622279448	99.998913697820	0.0654958962	0.0622279448	94.748418302942
63.0000000000	63.0000000000	100.000000000000	63.0000000000	63.0000000000	100.000000000000
51.5626886617	51.5626886617	100.000000000000	51.5626886617	51.5626886617	100.000000000000

Tabela 4.2: Gradientes do funcional de deslocamento máximo (0.0066240 cm) avaliado no problema da Figura 4.4. Derivadas parciais de variáveis de forma são avaliadas com dois tipos de campos de velocidades e comparados com valores obtidos por diferenças finitas. AS: análise de sensibilidade. MDF: diferenças finitas usando o respectivo campo de velocidades. A concordância considera o valor MDF como padrão.

Método da Camada Unitária de Contorno			Método de Deslocamentos Fictícios do Contorno		
AS	**MDF**	**Concordância %**	**AS**	**MDF**	**Concordância %**
-0.0004906680	-0.0004907736	99.978495869766	-0.0004903594	-0.0004905262	99.966002157550
0.0003514299	0.0003512985	99.962621228525	0.0003512055	0.0003511043	99.971199203442
-0.0004687693	-0.0004684957	99.941613751385	-0.0004687172	-0.0004684722	99.947715770464
0.0003750564	0.0003749502	99.971683717945	0.0003750172	0.0003749874	99.992052180418
-0.0002464274	-0.0002462715	99.936675215907	-0.0002464322	-0.0002463835	99.980236887704
0.0002020671	0.0002023981	99.836451371037	0.0002020728	0.0002025115	99.783330323216
0.0002229800	0.0002228742	99.952523838024	0.0002186078	0.0002184983	99.949852974351
-0.0002406294	-0.0002407878	99.934195597654	-0.0002381976	-0.0002383269	99.945736778083
0.0000876989	0.0000876953	99.995917237224	0.0000869478	0.0000869279	99.977098346138
-0.0000125219	-0.0000125292	99.942050864593	-0.0000123988	-0.0000124187	99.840224604620
-0.0033921791	-0.0034181309	99.240760163367	-0.0033921791	-0.0034181309	99.240760163367
-0.0032318361	-0.0032570971	99.224432317577	-0.0032318361	-0.0032570971	99.224432317577

Tabela 4.3: Gradientes do funcional de energia de deformação (0.0490047 kNcm) avaliado no problema da Figura 4.4. Derivadas parciais de variáveis de forma são avaliadas com dois tipos de campos de velocidades e comparados com valores obtidos por diferenças finitas. AS: análise de sensibilidade. MDF: diferenças finitas usando o respectivo campo de velocidades. A concordância considera o valor MDF como padrão.

Método da Camada Unitária de Contorno			**Método de Deslocamentos Fictícios do Contorno**		
AS	**MDF**	**Concordância %**	**AS**	**MDF**	**Concordância %**
-0.0016886893	-0.0016902836	99.905680363374	-0.0016885152	-0.0016900781	99.907524639923
0.0013086774	0.0013088935	99.983490315339	0.0013085659	0.0013087349	99.987087780716
-0.0022959217	-0.0022963636	99.980758899327	-0.0022958237	-0.0022963060	99.978995283404
0.0018368352	0.0018370796	99.986696802376	0.0018367988	0.0018370432	99.986696501040
-0.0014093507	-0.0014094749	99.991189807308	-0.0014093114	-0.0014094556	99.989771780048
0.0011556109	0.0011561855	99.950298782420	0.0011555686	0.0011561614	99.948731509059
0.0019562049	0.0019555812	99.968107313567	0.0019197023	0.0019191985	99.973749288458
-0.0019756050	-0.0019765653	99.951418535305	-0.0019542394	-0.0019550601	99.958021153077
0.0008621585	0.0008624104	99.970788918893	0.0008559726	0.0008562043	99.972943541567
-0.0001122130	-0.0001122661	99.952715073340	-0.0001121678	-0.0001122260	99.948132667627
-0.0233141002	-0.0235038244	99.192794063707	-0.0233141002	-0.0235038244	99.192794063707
-0.0256906110	-0.0259119562	99.145779402283	-0.0256906110	-0.0259119562	99.145779402283

Tabela 4.4: Gradientes do funcional de tensão máxima de von Mises (11.7011104 N/cm^2) avaliado no problema da Figura 4.4. Derivadas parciais de variáveis de forma são avaliadas com dois tipos de campos de velocidades e comparados com valores obtidos por diferenças finitas. AS: análise de sensibilidade. MDF: diferenças finitas usando o respectivo campo de velocidades. A concordância considera o valor MDF como padrão.

Método da Camada Unitária de Contorno			**Método de Deslocamentos Fictícios do Contorno**		
AS	**MDF**	**Concordância %**	**AS**	**MDF**	**Concordância %**
-0.0596684873	-0.0621274321	96.042094914352	-0.0599656247	-0.0623681632	96.147812618241
0.0481658170	0.0501555585	96.032859550390	0.0483094426	0.0501306087	96.367157336881
-0.1532153492	-0.1555217072	98.517018602167	-0.1534513353	-0.1554357701	98.723308832087
0.1224348377	0.1241417316	98.625044208681	0.1225078973	0.1240582473	98.750304779541
-0.1236379556	-0.1251238215	98.812483564333	-0.1236767565	-0.1249471797	98.983231825032
0.1016567561	0.1016975552	99.959881864333	0.1016275700	0.1014516574	99.826604524256
0.4278937073	0.4022925979	93.636196770174	0.4204282779	0.3953915539	93.667865770007
-0.3667297030	-0.3453067820	93.795974433030	-0.3612723435	-0.3401027828	93.775540293550
0.2081579908	0.1968208223	94.239853068907	0.2055911879	0.1948217777	94.472173326776
-0.0263543508	-0.0246979281	93.293272826856	-0.0340166228	-0.0293791851	84.215226807699
-1.2163386971	-1.1911893116	97.888716328250	-1.2163386971	-1.1911893116	97.888716328250
-10.8280721649	-10.6200089826	98.040837982254	-10.8280721649	-10.6200089826	98.040837982254

Método de Deslocamentos Fictícios do Contorno

O método de deslocamentos fictícios do contorno consiste em interpretar cada campo de velocidades no contorno como condições de Dirichlet impostas num problema elastostático linear fictício com as seguintes características:

- corpo $\mathcal{B}$ composto de material elástico linear fictício com módulo de elasticidade unitário e coeficiente de Poisson nulo;
- cada variável de forma dá origem a um campo de velocidades não-nulas numa parcela de $\partial\mathcal{B}$, sendo o restante do contorno associado a velocidades nulas (o que garante o equilíbrio de cada problema auxiliar);
- usa-se a mesma discretização da análise de resposta para o campo de deslocamentos, devido à primeira exigência teórica;
- as incógnitas são as velocidades de todos os nós internos de $\mathcal{B}$.

Cada campo é, portanto, determinado a partir da solução de um problema linear, garantindo o respeito à segunda exigência teórica. Numericamente, cada problema linear corresponde à solução de um sistema de equações lineares da ordem de uma iteração de equilíbrio de Newton-Raphson. Considerando o projeto parametrizado e discretizado em elementos finitos da Figura 4.4, tem-se os campos de velocidades sobre o contorno e no interior do domínio mostrados na Figura 4.5 comparados com os campos obtidos com o método da camada unitária de contorno.

A precisão dos cálculos de sensibilidade usando campos de velocidades obtidos desta forma é praticamente idêntica[7] ao que se observa usando campos gerados pelo método da camada unitária de contorno (Tabelas 4.1 a 4.4). Nos dois casos, a verificação componente a componente, que é a mais exigente (Tseng e Arora, 1989), mostra resultados precisos mesmo quando o valor absoluto da derivada parcial é pequeno.

Entretanto, a principal característica deste método é originar campos que possibilitam aplicar (3.9) com maior flexibilidade, preservando a qualidade da malha original para intervalos maiores de perturbação. Na seqüência de otimização mostrada na Figura 4.7, foi necessário reconstruir a malha

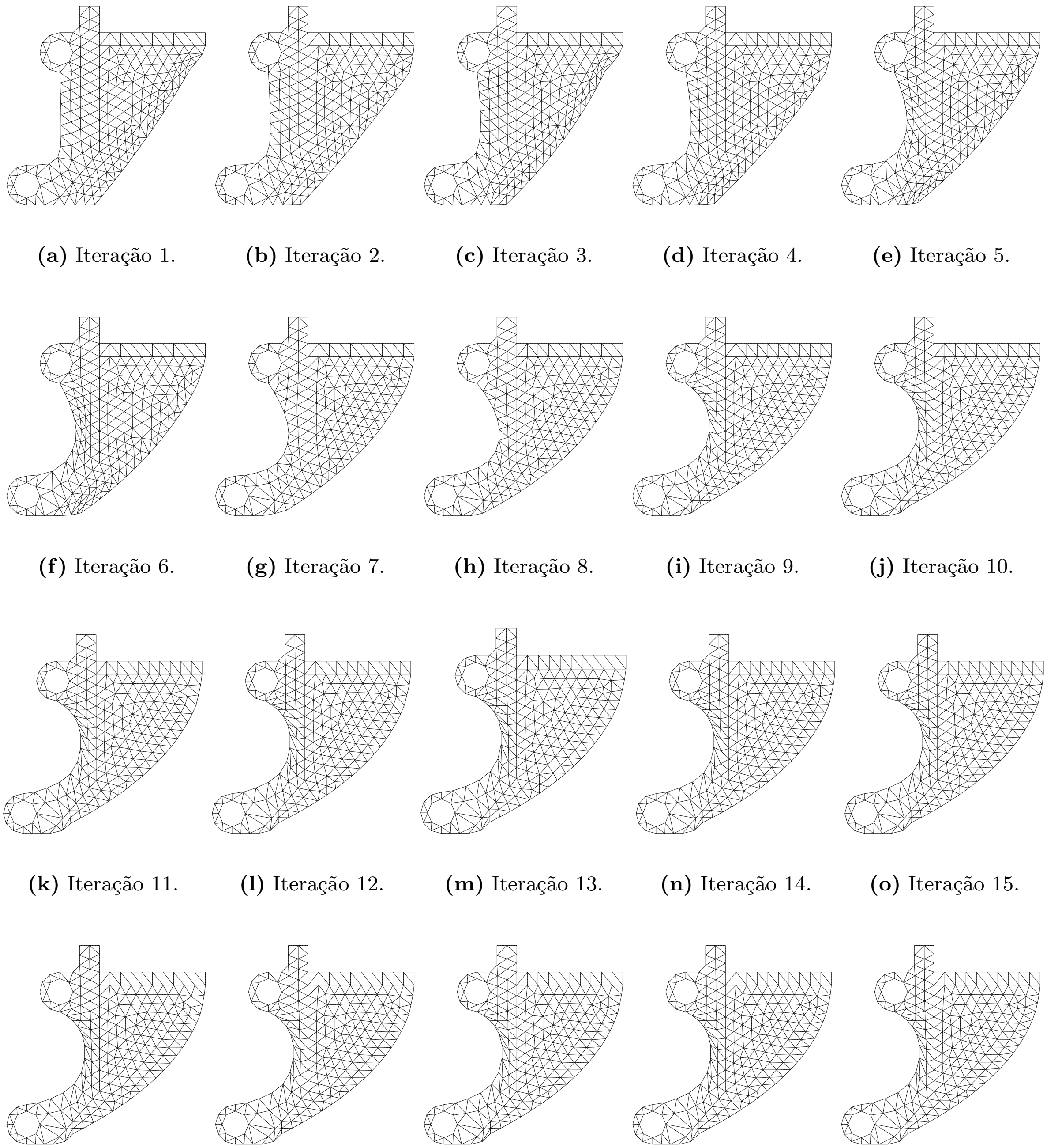
[7] Em algumas variáveis, a precisão é pouco menor devido ao erro numérico envolvido na solução de sistemas lineares.

apenas na iteração 7. Em alguns casos, é possível executar toda seqüência de iterações de otimização sem a necessidade de reconstruir a malha de elementos. Um exemplo disso é mostrado nas Figuras 4.8 a 4.10. Em casos para os quais não se dispõe de geração automática de malhas ou não é possível integrar o gerador ao ambiente de otimização, a aplicação deste método é necessária apesar de seu custo computacional claramente alto quando implementado no formato original, ou seja, resolvendo-se os problemas auxiliares exatamente.

Observa-se, porém, que é possível aumentar bastante a eficiência do método modificando-se ligeiramente a estratégia de solução dos problemas fictícios. Simulações numéricas mostram que os resultados de procedimentos de otimização sofrem apenas pequenas modificações quando os problemas auxiliares de determinação de campos de velocidades são resolvidos com métodos iterativos baseados em gradientes conjugados (Bittencourt e Feijóo, 1997) exigindo-se baixa precisão na solução. Ao se exigir baixa precisão, o custo associado à solução iterativa de sistemas lineares cai bastante, podendo se tornar menor até que o custo de aplicação de condições de contorno Dirichlet na matriz global do problema fictício (o que é especialmente crítico em problemas tridimensionais devido ao número de nós da malha) dependendo da estrutura de dados usada no armazenamento dessa matriz. Nessa situação, armazenamentos em colunas ascendentes (*skyline*) ou esparso por colunas são superiores ao esparso por linhas (Bittencourt e Feijóo, 1997), nos quais o acesso aos elementos das colunas é feito de forma indireta através de laços condicionais.

Resultados para o problema das Figuras 4.4 a 4.7 são mostrados na Figura 4.11. Comparam-se as evoluções da função objetivo com a técnica de camada unitária de contorno e com deslocamentos fictícios do contorno para três diferentes precisões de método de gradiente conjugado com pré-condicionador Gauss-Seidel simétrico para solução dos subproblemas. Aplicando camada unitária de contorno e deslocamentos fictícios com precisões 10^{-4} e 10^{-2}, os resultados são praticamente idênticos em número de iterações e solução final. Reduzindo a precisão para 10^{-1}, obtem-se ainda a mesma solução. Porém a taxa de convergência efetiva é reduzida e com isso o número de iterações até a convergência aumenta de 17 para 20. A contrapartida em termos de redução de custo de obtenção dos campos de velocidade é ilustrada nas Tabelas 4.5 a 4.7, que mostram o número de iterações de gradiente conjugado necessárias para a solução dos subproblemas associados a cada variável de forma.

Resultados para o problema das Figuras 4.8 e 4.9 são ilustrados na Figura 4.12, podendo-se observar apenas uma pequena variação no volume ótimo para 3 precisões diferentes do método de



(a) Iteração 1. **(b)** Iteração 2. **(c)** Iteração 3. **(d)** Iteração 4. **(e)** Iteração 5.

(f) Iteração 6. **(g)** Iteração 7. **(h)** Iteração 8. **(i)** Iteração 9. **(j)** Iteração 10.

(k) Iteração 11. **(l)** Iteração 12. **(m)** Iteração 13. **(n)** Iteração 14. **(o)** Iteração 15.

(p) Iteração 16. **(q)** Iteração 17. **(r)** Iteração 18. **(s)** Iteração 19. **(t)** Iteração 20.

Figura 4.7: Seqüência de otimização do problema definido na Figura 4.4 com campos de velocidades gerados com o método de deslocamentos fictícios do contorno (volume inicial: 114.563 cm^3; volume final: 69.543 cm^3). Problemas auxiliares resolvidos por gradiente conjugado com pré-condicionamento Gauss-Seidel simétrico com precisão 10^{-1}. Na iteração 7, a malha é reconstruída com gerador de malhas. Nas demais iterações, o campo de velocidades é utilizado para atualizar a dicretização.

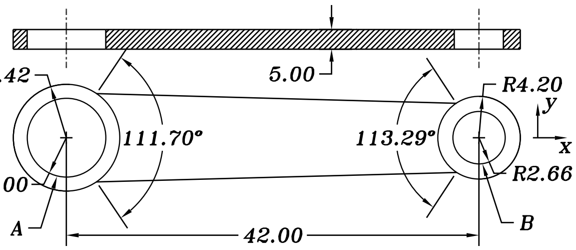


Figura 4.8: Dimensões [cm] do projeto inicial ($E = 21.0 \times 10^3$ kN/cm^2, $\nu = 0.3$, $\rho = 7.81 \times 10^{-3}$kg/cm^3). Os nós da superfície A estão engastados. A força distribuída ($F_x = 0.12$ kN/cm^2, $F_y = 0.06$ kN/cm^2, $F_z = 0.04$ kN/cm^2) é aplicada na superfície B.

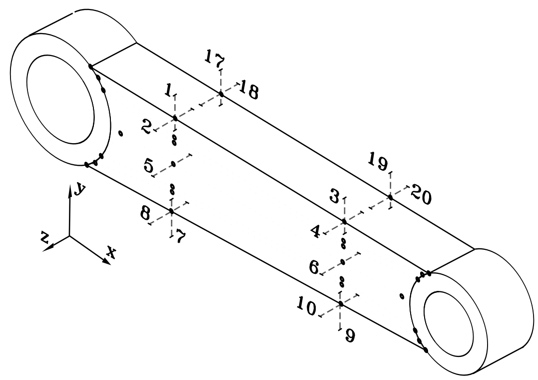
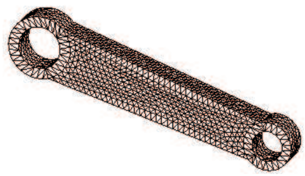


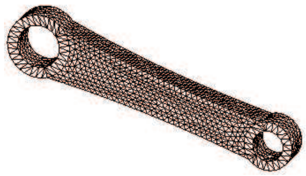
Figura 4.9: Variáveis de projeto. As linhas tracejadas indicam a direção e a amplitude de variação de cada variável de projeto. O objetivo é minimizar o volume com restrição na máxima energia de deformação (0.2 kNcm).

Tabela 4.5: Custo de solução dos subproblemas auxiliares de determinação de campos de velocidades do problema definido na Figura 4.4 em termos de iterações de gradiente conjugado pré-condicionado por Gauss-Seidel simétrico com precisão 10^{-4}.

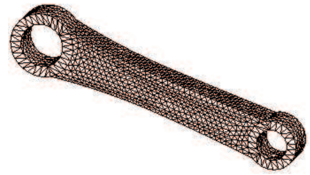
	Variáveis de forma									
	1	2	3	4	5	6	7	8	9	10
Iteração de otimização	Número de iterações de solução de subproblemas auxiliares									
0	33	33	33	33	27	33	22	23	21	24
1	32	32	32	32	29	32	21	21	21	21
2	33	33	32	32	29	32	21	21	21	24
3	32	32	32	32	29	32	21	21	21	24
4	32	32	32	34	29	32	21	21	21	24
5	32	32	34	34	29	34	21	20	19	23
6	32	32	34	34	29	34	22	20	19	22
7	33	37	33	33	30	33	17	19	15	17
8	33	39	33	33	29	33	16	17	15	17
9	33	39	36	35	29	33	19	19	16	19
10	33	39	35	35	29	33	19	19	16	19
11	33	33	35	35	29	33	19	19	16	19
12	33	36	35	35	29	35	16	19	16	19
13	33	36	35	35	29	35	16	19	16	19
14	33	36	35	35	29	35	16	19	16	19
15	33	36	35	35	29	35	16	19	16	19
16	33	36	35	35	29	35	16	19	16	19
Total	556	593	576	577	492	569	319	335	301	348
										4666



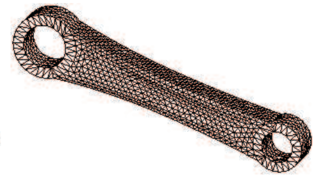
(a) Projeto inicial.



(b) Iteração 1.

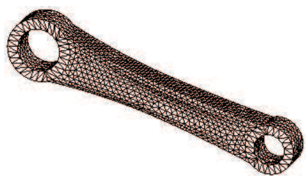


(c) Iteração 2.

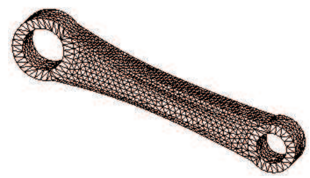


(d) Iteração 3.

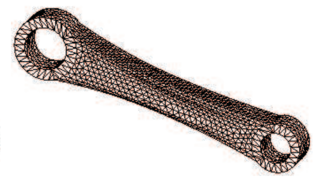
(e) Iteração 4.



(f) Iteração 5.



(g) Iteração 6.

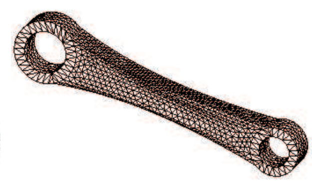


(h) Iteração 7.

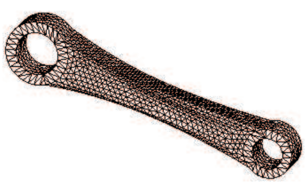
(i) Iteração 8.

(j) Iteração 9.

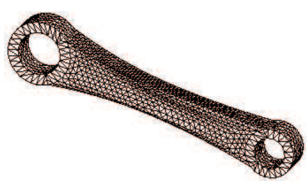
(k) Iteração 10.



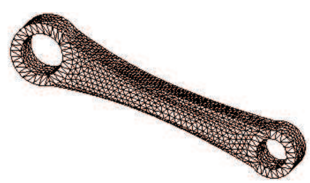
(l) Iteração 11.



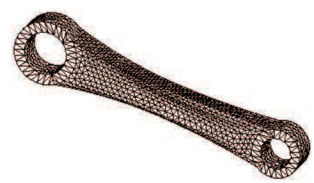
(m) Iteração 12.



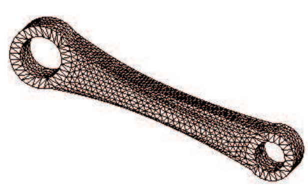
(n) Iteração 13.



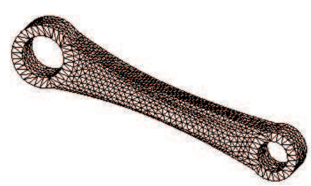
(o) Iteração 14.



(p) Iteração 15.

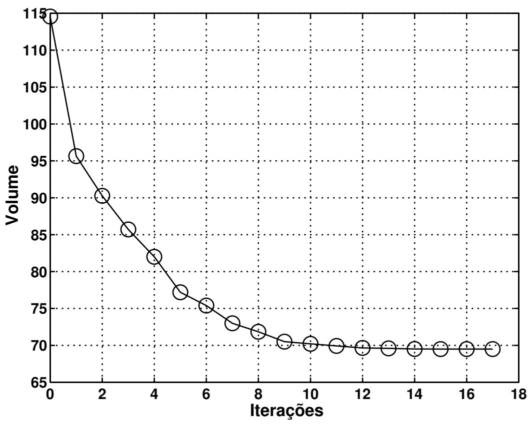


(q) Iteração 16.

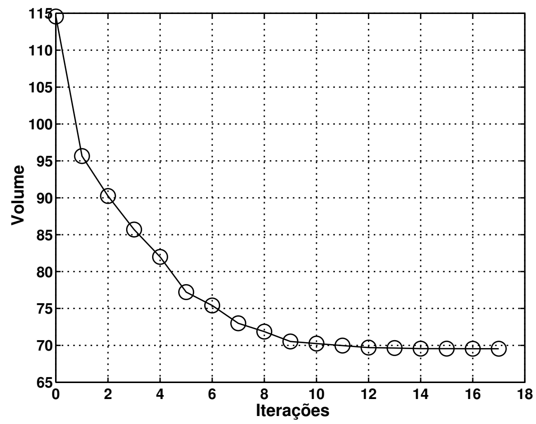


(r) Iteração 17: projeto ótimo.

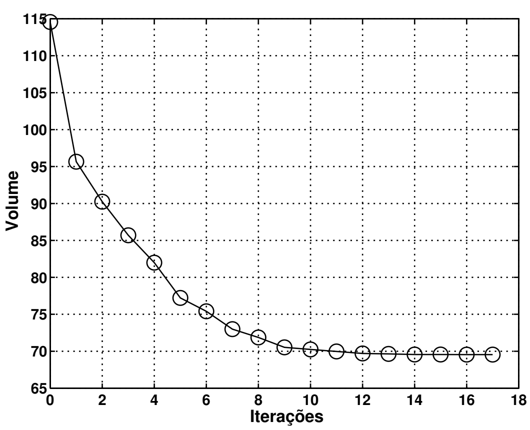
Figura 4.10: Seqüência de otimização sem reconstrução da malha (volume inicial: 1763.98 cm^3; volume final: 1295.20 cm^3). Problemas auxiliares resolvidos com precisão 10^{-1}. Campos de velocidades gerados com o método de deslocamentos fictícios do contorno foram usados para reposicionar os nós a cada modificação da geometria.



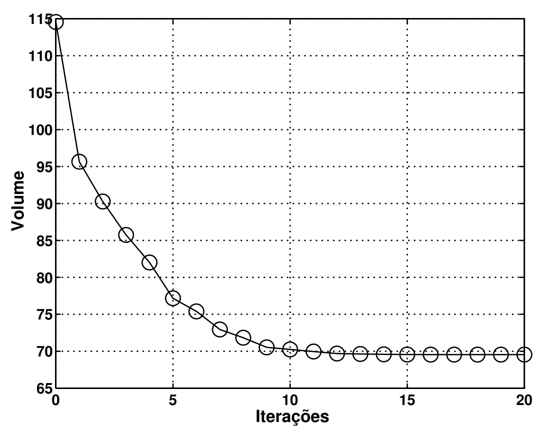
(a) Camada unitária de contorno. Volume final: 69.50 cm^3. 17 iterações (Figura 4.6).



(b) Deslocamentos fictícios do contorno. Problemas auxiliares resolvidos com precisão 10^{-4}. Volume final: 69.55 cm^3. 17 iterações.



(c) Deslocamentos fictícios do contorno. Problemas auxiliares resolvidos com precisão 10^{-2}. Volume final: 69.55 cm^3. 17 iterações.



(d) Deslocamentos fictícios do contorno. Problemas auxiliares resolvidos com precisão 10^{-1}. Volume final: 69.54 cm^3. 20 iterações (Figura 4.7).

Figura 4.11: Evolução da função objetivo em seqüências de otimização do problema definido na Figura 4.4. Volume inicial: 114.56 cm^3. Os sistemas lineares dos problemas auxiliares foram resolvidos com gradiente conjugado com precondicionamento Gauss-Seidel simétrico.

Tabela 4.6: Custo de solução dos subproblemas auxiliares de determinação de campos de velocidades do problema definido na Figura 4.4 em termos de iterações de gradiente conjugado pré-condicionado por Gauss-Seidel simétrico com precisão 10^{-2}.

	Variáveis de forma									
	1	2	3	4	5	6	7	8	9	10
Iteração de otimização	Número de iterações de solução de subproblemas auxiliares									
0	14	11	14	14	11	11	8	11	8	15
1	14	14	14	14	10	11	8	11	8	15
2	14	11	14	14	10	13	8	11	8	15
3	14	14	13	14	10	13	8	11	8	15
4	14	11	14	16	10	13	8	11	8	15
5	14	11	13	16	10	13	8	8	7	11
6	14	11	13	16	10	13	9	8	7	11
7	14	17	11	14	8	11	7	8	7	8
8	14	17	11	14	10	11	8	8	6	8
9	14	17	11	14	10	14	8	7	7	10
10	14	14	11	14	10	14	7	7	7	10
11	14	14	11	14	10	14	8	7	7	10
12	14	14	11	17	10	13	8	7	7	10
13	14	14	14	17	10	13	8	7	7	10
14	14	14	14	17	10	13	10	7	7	10
15	14	14	14	17	10	13	10	7	7	10
16	14	14	14	17	10	13	10	7	7	10
Total	238	232	217	259	169	216	141	143	123	193
										1931

gradiente conjugado.

O método de deslocamentos fictícios de contorno associado à solução iterativa com baixa precisão fornece campos de velocidades com características muito semelhantes às dos campos obtidos pelo método original, porém com custo intermediário entre o mesmo e o método de camada unitária de contorno.

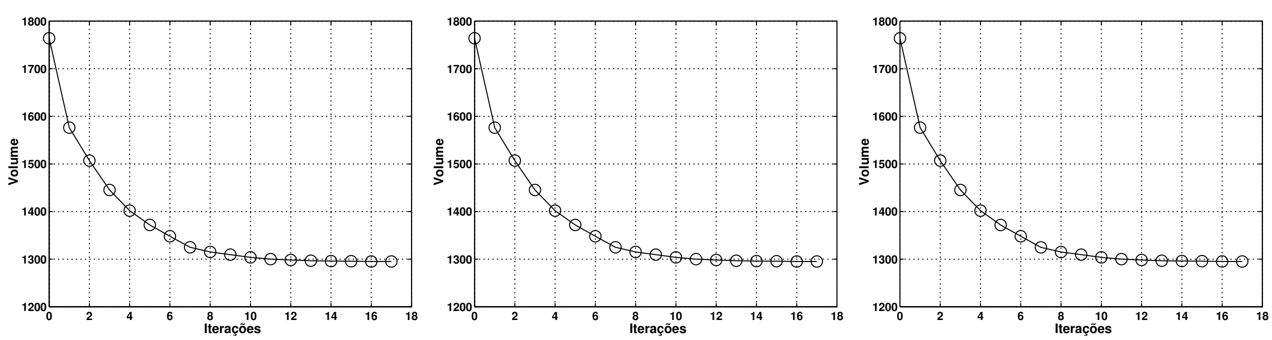
4.2 Atualização de Malhas em Iterações de Otimização

Qualquer que seja o campo de velocidades, existe a possibilidade de que a distorção da malha causada pela aplicação de (3.9), numa mesma iteração ou durante a seqüência de iterações, possa comprometer a qualidade da análise de resposta e análise de sensibilidade. O ambiente de otimização deve estar preparado para identificar essa ocorrência e requisitar a reconstrução da malha ou a interrupção do processo quando isso não for possível.

Justificativas teóricas e práticas mostram ser necessário aplicar (3.9) sempre que as perturbações impostas à malha pelas modificações nas variáveis de projeto não comprometam a qualidade da malha.

Tabela 4.7: Custo de solução dos subproblemas auxiliares de determinação de campos de velocidades do problema definido na Figura 4.4 em termos de iterações de gradiente conjugado pré-condicionado por Gauss-Seidel simétrico com precisão 10^{-1}.

	Variáveis de forma									
	1	2	3	4	5	6	7	8	9	10
Iteração de otimização	Número de iterações de solução de subproblemas auxiliares									
0	4	4	4	4	4	4	4	3	4	9
1	4	4	4	4	4	4	4	3	4	9
2	4	4	4	4	4	4	2	5	4	9
3	4	4	4	4	4	4	2	5	4	9
4	4	4	4	4	4	4	4	3	4	9
5	4	4	4	4	4	4	4	4	4	5
6	4	4	4	4	4	4	4	2	4	2
7	4	2	4	4	4	4	3	2	3	2
8	4	2	4	4	4	4	4	2	3	5
9	4	2	4	4	4	4	4	2	3	5
10	2	2	4	4	4	4	4	2	4	5
11	2	2	4	4	4	4	4	2	4	5
12	2	2	4	4	4	4	4	2	4	5
13	2	2	4	4	4	4	4	2	4	5
14	2	2	4	4	4	4	4	2	4	5
15	2	2	4	5	4	4	4	2	4	5
16	2	2	4	5	4	4	4	2	4	5
17	2	2	4	5	4	4	4	2	4	5
18	2	2	4	5	4	4	4	2	4	5
19	2	2	4	5	4	4	4	2	4	5
Total	060	054	080	085	080	080	075	051	077	114
										756



(a) Problemas auxiliares resolvidos com precisão 10^{-4}. Volume final: 1295.16 cm^3. 17 iterações.

(b) Problemas auxiliares resolvidos com precisão 10^{-2}. Volume final: 1295.22 cm^3. 17 iterações.

(c) Problemas auxiliares resolvidos com precisão 10^{-1}. Volume final: 1295.20 cm^3. 17 iterações.

Figura 4.12: Evolução da função objetivo em seqüências de otimização utilizando método de deslocamentos fictícios do contorno para o problema das Figuras 4.8 e 4.9. Volume inicial: 1763.98 cm^3. Os sistemas lineares dos problemas auxiliares foram resolvidos com gradiente conjugado com precondicionamento Gauss-Seidel simétrico.

Tais perturbações são consideradas *pequenas* neste trabalho. Quando a malha é distorcida além do limite aceitável ou perde sua consistência geométrica, tem-se perturbações *grandes*. Portanto, não é a quantidade de alteração do domínio $\mathcal{B}$ ou do valor das variáveis de projeto que determina se uma perturbação é denominada grande ou pequena e sim a qualidade e a consistência da malha produzida pela regra linear de reposicionamento dos nós. Uma pequena alteração de $\mathcal{B}$ está sempre associada a uma pequena perturbação, mas o inverso não é verdadeiro.

No caso de pequenas perturbações, a taxa de convergência da otimização é altamente influenciada pelo respeito estrito aos requisitos já discutidos. Em outras palavras, a observância da aplicação precisa da formulação durante a manipulação do modelo discreto é fundamental. Porém, mesmo do ponto de vista analítico, é razoável esperar que, quanto maior a perturbação, maior a discrepância entre a performance do sistema e a previsão linear fornecida pela análise de sensibilidade. Também a interpolação quadrática dos funcionais pode deixar de ser uma estimativa confiável nesses casos. Soma-se a isso a crescente importância do efeito de distorção dos elementos na resposta do sistema. Por isso, o respeito à (3.9) se torna menos importante que a garantia da precisão das análises de resposta e sensibilidade quando ocorrem grandes perturbações da malha. Nessas situações, *e somente nessas*, a reconstrução da malha deve ser feita, aplicando inclusive procedimentos de estimação de erros e análise adaptável se necessário (Fancello, 1993; Bugeda e Oliver, 1993; Chang e Choi, 1992; Yao e Choi, 1989; Özakça *et al.*, 1993; Canales *et al.*, 1993; Buscaglia *et al.*, 1995; Buscaglia *et al.*, 1998; Dufeu *et al.*, 1997). Para isso, também é necessário que a geometria – e não a malha – seja a entidade central da parametrização. Se a malha não pode ser reconstruída automaticamente de forma satisfatória, o procedimento de otimização deve ser interrompido para sua reconstrução.

Por fim, resta discutir critérios para aceitação de malhas de elementos finitos, ou seja, critérios para diferenciar perturbações grandes e pequenas. Propõe-se a aplicação de estimativas de erro *a priori*, baseadas na verificação da distorção da geometria dos elementos em relação a uma geometria padrão. Com isso é possível verificar a qualidade da malha de forma eficiente e com razoável precisão, considerando a não necessidade de se executar a análise de resposta. Uma malha é rejeitada quando:

1. há perda de consistência geométrica da malha, de acordo com a Figura 4.2,
2. pelo menos um elemento excede o limite máximo aceitável de distorção ou
3. uma porcentagem dos elementos da malha excede um nível médio.

O item 1 pode ser facilmente detectado pela ocorrência de elementos com jacobiano da transformação para o elemento padrão com sinal negativo. Critérios para a avaliação dos itens 2 e 3 são discutidos a seguir.

4.2.1 Medidas de Distorção de Malhas

Triângulos

A distorção de elementos triangulares é determinada comparando-se sua geometria com um triângulo equilátero. Uma forma eficiente de efetuar essa comparação é através da relação adimensional

$$\frac{L_e^2}{A_e}, \tag{4.11}$$

sendo L_e o perímetro do elemento e A_e sua área. Aplicando essa relação num triângulo equilátero, tem-se que

$$\frac{L_e^2}{A_e} = 12\sqrt{3}.$$

Define-se aqui a distorção de um triângulo qualquer em relação a um triângulo equilátero como a medida

$$\boxed{d_{tri} = \frac{L_e^2}{12\sqrt{3}A_e}} \tag{4.12}$$

cujo valor mínimo 1 corresponde a uma geometria não distorcida. Qualquer variação em relação à geometria equilátera implica em $d_{tri} > 1$. Através dessa relação, é possível detectar tanto a ocorrência de ângulos excessivamente obtusos (maiores que 90°) quanto problemas de razão de aspecto.

Observa-se ainda que a taxa de crescimento da medida d_{tri} em relação à variação da geometria triangular é bastante acentuada, sendo possível identificar geometrias distorcidas para valores moderados de d_{tri}. Para o triângulo isósceles da Figura 4.13, por exemplo, tem-se

$$d_{tri} = \frac{(2+\lambda)^2}{12\sqrt{3\left(\frac{\lambda^2}{4} - \frac{\lambda^4}{16}\right)}}.$$

A partir daí, observa-se que

$$\lambda \;=\; 1 \rightarrow d_{tri} = 1,$$

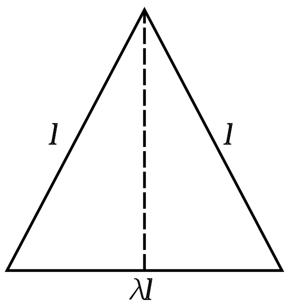


Figura 4.13: Triângulo isósceles parametrizado pela variável λ.

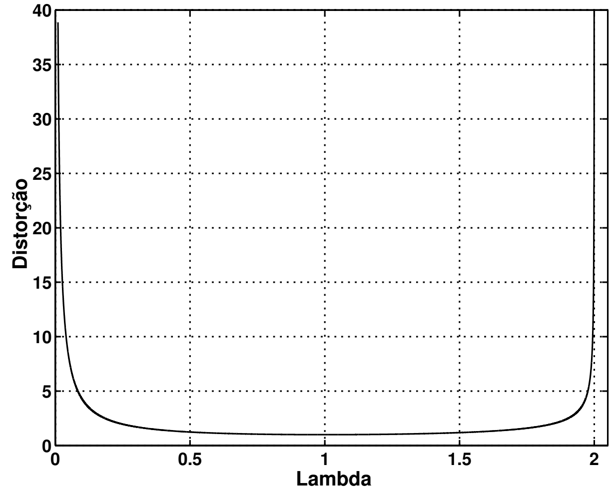


Figura 4.14: Distorção no triângulo isósceles de acordo com a variação no parâmetro λ.

$$\lambda \rightarrow 0, \quad d_{tri} \rightarrow \infty,$$
$$\lambda \rightarrow 2, \quad d_{tri} \rightarrow \infty,$$

de acordo com o comportamento mostrado no gráfico da Figura 4.14.

Tetraedros

Aplica-se a elementos tetraédricos, procedimento análogo ao usado para triângulos. Neste caso, tem-se a relação adimensional

$$\frac{L_e^3}{V_e}, \tag{4.13}$$

sendo L_e a soma dos comprimentos das arestas e V_e o volume do tetraedro. No caso de um tetraedro equilátero, tem-se

$$\frac{L_e^3}{V_e} = 1296\sqrt{2}.$$

Define-se a distorção de um tetraedro como a normalização de (4.13) pelo valor observado em tetraedros equiláteros, ou seja,

$$\boxed{d_{tet} = \frac{L_e^3}{1296\sqrt{2}V_e}.} \tag{4.14}$$

Da mesma forma que o observado em triângulos, $d_{tet} \geq 1$, sendo que o valor mínimo corresponde à geometria não distorcida. A medida de distorção (4.14) também controla ângulos internos e razão de aspecto.

Quadrângulos e Hexaedros

No caso de elementos planos de quatro lados e hexaedros, as medidas de distorção são relacionadas à variação entre os ângulos formados pelas arestas e o ângulo reto.

Para quadrângulos, aplica-se a relação

$$\boxed{d_{quad} = \frac{1}{\min \sin \theta_i}} \tag{4.15}$$

como medida de distorção, sendo $\theta_i,\ i = 1, \ldots, 4$ os ângulos internos do elemento. Observa-se que

$$\begin{aligned}
d_{quad} &\geq 1, \\
d_{quad} &= 1 \Leftrightarrow \theta_i = 90^\circ, \quad i = 1, \ldots, 4, \\
d_{quad} &\to \infty, \quad \theta_i \to 0^\circ \quad \forall i, \\
d_{quad} &\to \infty, \quad \theta_i \to 180^\circ \quad \forall i.
\end{aligned}$$

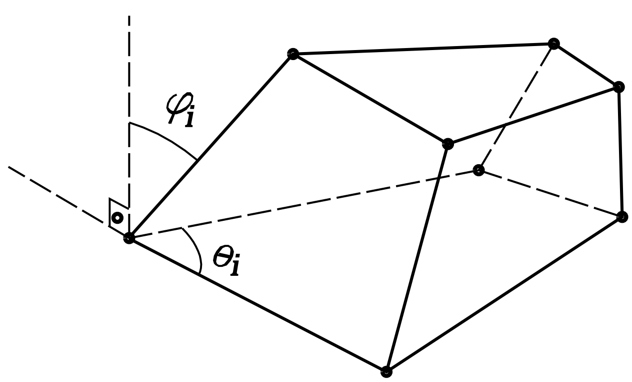


Figura 4.15: Ângulos usados nas medidas de distorção de hexaedros.

Em hexaedros, usa-se

$$\boxed{d_{hexa} = \frac{1}{\min\left(\sin \theta_i \cos \varphi_i\right)}} \tag{4.16}$$

sendo $\sin \theta_i \cos \varphi_i$ calculado em seus oito vértices; θ_i e φ_i são definidos de acordo com a Figura 4.15.

As seguintes propriedades são observadas em (4.16):

$$
\begin{aligned}
d_{hexa} &\geq 1, \\
d_{hexa} &= 1 \Leftrightarrow \theta_i, \varphi_i = 90^\circ, \quad i = 1, \ldots, 8, \\
d_{hexa} &\rightarrow \infty, \quad \theta_i, \varphi_i \rightarrow 0^\circ \quad \forall i, \\
d_{hexa} &\rightarrow \infty, \quad \theta_i, \varphi_i \rightarrow 180^\circ \quad \forall i.
\end{aligned}
$$

Deve-se observar, entretanto que as medidas de distorção (4.15) e (4.16) não consideram a razão de aspecto dos elementos.

Em testes numéricos, as medidas (4.12), (4.14), (4.15) e (4.16) se mostraram comparáveis aos recursos de avaliação de qualidade de malhas de softwares comerciais tais como o Ansys, identificando regiões semelhantes de elementos distorcidos (Ferreira, 2002).

4.3 Algoritmo de Otimização

Apesar de fornecerem a base téorica para a determinação de pontos de mínimo em problemas genéricos, as condições de Karush-Kuhn-Tucker (Bazaraa *et al.*, 1993) não fornecem um procedimento geral aplicável a uma grande variedade de problemas práticos. Nesse sentido, a utilização de ferramentas numéricas de programação matemática se torna necessária.

No início da década de 80, demonstrou-se a necessidade de dispensar atenção especial às características específicas dos problemas de otimização estrutural (Belegundu e Arora, 1985a; Belegundu e Arora, 1985b; Vanderplaats, 1982). De fato, não é possível relacionar a performance de um algoritmo em um problema de programação matemática com o seu comportamento em problemas de otimização de projetos. A razão é que problemas de programação matemática consistem de funções explícitas, de pequena dimensão e pequeno número de mínimos locais. Assim, os problemas de programação matemática são em geral triviais em termos do tempo computacional de análise de resposta e determinação de gradientes. Dessa forma, em tais problemas normalmente se utiliza uma combinação de tempo de CPU, tempo de preparação, facilidade de uso, etc., como aspectos principais de um código. Conseqüentemente um algoritmo pode parecer muito bom mesmo exigindo centenas de avaliações da função objetivo e das restrições para resolver um problema de apenas três ou quatro variáveis. O custo da aplicação de tal código em um problema que envolva análises complexas de resposta e sensibilidade é proibitivo.

Isto sugere que, em problemas práticos de projeto, somente dois critérios são significativos. Em primeiro lugar, se o algoritmo e sua implementação são confiáveis em atingir um mínimo aproximado a partir de um ponto inicial arbitrário. E segundo, se o programa realiza o menor número possível de avaliações das funções do problema e de seus gradientes.

Um algoritmo é classificado como *globalmente convergente* se alcança um mínimo local a partir de um ponto inicial arbitrário. Fica evidente a partir dessa definição que a convergência global é uma indicação de robustez e confiabilidade, mas não de eficiência. Este critério é usado como um ponto básico no estudo dos vários métodos e não deve ser confundido com o problema de determinação de mínimo global (Arora *et al.*, 1995). A convergência global é uma exigência importante porque permite utilizar o recurso da otimização com confiança. Também é necessário que a implementação numérica dos algoritmos exibam propriedades de convergência global.

Para exigir um número mínimo de avaliações de funções e gradientes, o algoritmo de minimização deve fornecer, em cada iteração, uma direção de busca eficiente. Isso significa uma direção que permita máximo descréscimo da função objetivo num passo viável sem saturar prematuramente nenhuma restrição. Associada a esta característica deve estar um algoritmo de busca linear que tire máximo proveito das informações numéricas disponíveis no projeto corrente (valores dos funcionais e suas derivadas) na determinação do projeto ótimo de cada iteração.

Neste trabalho, utiliza-se o Método de Pontos Interiores de Herskovits (Herskovits, 1986), o qual consiste de um algoritmo de pontos interiores para otimização não-linear sujeita a restrições de igualdade e desigualdade.

Devido à sua característica de convergência global com taxa superlinear quando associado à busca linear imprecisa (Herskovits, 1986; Herskovits e Santos, 1997; Panier *et al.*, 1988), e por gerar uma seqüência de pontos viáveis, tem sido aplicado com sucesso em problemas estruturais (Herskovits e Coelho, 1989; Evsukoff, 1992; Fancello, 1993; Herskovits *et al.*, 1998; Dias *et al.*, 1998; Silva, 1997; Silva e Bittencourt, 1997; Silva e Bittencourt, 1998; Silva e Bittencourt, 1999b; Silva e Bittencourt, 1999a; Silva e Bittencourt, 2000).

O método exige apenas a solução de dois sistemas lineares de mesma matriz (e dimensão $n+m+p$)

para determinar a direção de busca. Além disso, como a seqüência dos multiplicadores de Lagrange das restrições de desigualdade é estritamente positiva, as direções de busca são sempre direções de decréscimo da função objetivo (Herskovits, 1986). No caso de haver apenas restrições de desigualdade, não há necessidade de nenhuma função de penalização.

Para obter a direção de busca a cada iteração, este algoritmo de pontos interiores emprega uma técnica iterativa de ponto fixo na solução direta do sistema de equações, nas variáveis primais e duais, obtido das condições necessárias de otimalidade de Karush-Kuhn-Tucker. O algoritmo é tal que as restrições de desigualdade e a exigência de não-negatividade de seus multiplicadores de Lagrange são satisfeitas a cada iteração, garantido a convergência para pontos de mínimo e o respeito estrito às restrições.

A exigência de ser iniciado com projeto viável é uma característica que pode ser criticada, mas que não chega a ser restritiva devido à existência de métodos específicos para a solução de sistemas de equações e inequações não-lineares (Elwakeil e Arora, 1995).

O algoritmo de busca linear imprecisa implementado apresenta características de eficiência que o tornam bastante atraente para aplicações de otimização estrutural (Evsukoff, 1992; Silva, 1997). O algoritmo busca um passo viável que respeite o decréscimo mínimo especificado para função em cada passo. Sua hipótese básica é que, na maioria dos problemas de otimização, a solução se localiza sobre o contorno da região viável Ω, ou seja, uma ou mais restrições estarão saturadas. O passo inicial é então determinado através de interpolação linear das restrições: o maior passo positivo que se sature a aproximação linear do conjunto de restrições é selecionado. Se necessário, esse passo é refinado através de interpolações quadráticas dos funcionais do problema para cumprir as exigências de viabilidade e decréscimo. Os testes numéricos executados com a implementação atual mostraram que a maior parte das buscas lineares são concluídas com 1 a 3 avaliações dos funcionais de performance.

Tabela 4.8: Gradientes do funcional de deslocamento máximo (0.0304010 cm). Derivadas parciais de variáveis de forma são avaliadas com dois tipos de campos de velocidades e comparados com valores obtidos por diferenças finitas. AS: análise de sensibilidade. MDF: diferenças finitas usando o respectivo campo de velocidades. A concordância considera o valor MDF como padrão.

Método da Camada Unitária de Contorno			Método de Deslocamentos Fictícios do Contorno		
AS	**MDF**	**Concordância %**	**AS**	**MDF**	**Concordância %**
-0.0019128591	-0.0019128326	99.998612380646	-0.0019140349	-0.0019140064	99.998510116967
0.0013896411	0.0013896290	99.999125379770	0.0013911530	0.0013911396	99.999032982680
-0.0018746357	-0.0018745966	99.997915589214	-0.0018753953	-0.0018753525	99.997718342300
0.0014995433	0.0014995155	99.998145746558	0.0015002566	0.0015002249	99.997883568543
-0.0009774573	-0.0009774337	99.997589989757	-0.0009784651	-0.0009784415	99.997589744164
0.0007914052	0.0007913868	99.997676001307	0.0007923675	0.0007923481	99.997546748035
0.0009141137	0.0009139925	99.986735277441	0.0009113731	0.0009112448	99.985919166783
-0.0010023199	-0.0010020547	99.973529020722	-0.0010014587	-0.0010011408	99.968247746807
0.0003226029	0.0003226053	99.999244868388	0.0003212260	0.0003212407	99.995424943269
-0.0000529705	-0.0000529443	99.950583835940	-0.0000563482	-0.0000563070	99.926857055047

Tabela 4.9: Gradientes do funcional de deslocamento absoluto máximo na direção x (0.0105026 cm). Derivadas parciais de variáveis de forma são avaliadas com dois tipos de campos de velocidades e comparados com valores obtidos por diferenças finitas. AS: análise de sensibilidade. MDF: diferenças finitas usando o respectivo campo de velocidades. A concordância considera o valor MDF como padrão.

Método da Camada Unitária de Contorno			Método de Deslocamentos Fictícios do Contorno		
AS	**MDF**	**Concordância %**	**AS**	**MDF**	**Concordância %**
-0.0007769657	-0.0007769580	99.999006437624	-0.0007772311	-0.0007772225	99.998899224533
0.0005514398	0.0005514401	99.999934006220	0.0005522640	0.0005522635	99.999894366098
-0.0006371690	-0.0006371675	99.999766271296	-0.0006374518	-0.0006374491	99.999572391359
0.0005092594	0.0005092605	99.999767921587	0.0005095050	0.0005095047	99.999928981709
-0.0002957063	-0.0002957079	99.999455898204	-0.0002961793	-0.0002961809	99.999475419802
0.0002427555	0.0002427562	99.999703491220	0.0002429167	0.0002429169	99.999908492771
-0.0000662496	-0.0000662325	99.974231161462	-0.0000658075	-0.0000657903	99.973917727488
0.0000235586	0.0000235319	99.886294218082	0.0000249678	0.0000249424	99.898297670972
-0.0000532753	-0.0000532691	99.988373567158	-0.0000535175	-0.0000535108	99.987416323745
0.0000052251	0.0000052160	99.825858938539	0.0000055858	0.0000055758	99.819993710752

4.4 Estudo de Casos de Análise de Sensibilidade e Otimização

4.4.1 Performance e Precisão de Análise de Sensibilidade em Problemas Não-Lineares

A geometria e a discretização das Figuras 4.4(a) e 4.4(c) foram utilizadas num problema de deformação plana com modelo de material Mooney-Rivlin incompressível ($A_{10} = 0.055$, $A_{01} = 0.0138$, $\tilde{\kappa} = 10^4$) e carregamento de 3×10^{-4} kN/cm. Considerando apenas as variáveis de forma da Figura 4.4(b), os resultados de sensibilidade de funcionais de deslocamento e energia de deformação são mostrados nas Tabelas 4.8 a 4.11.

Observa-se nas tabelas acima que o nível de precisão atingida na análise de sensibilidade de

Tabela 4.10: Gradientes do funcional de deslocamento absoluto máximo na direção y (0.0285292 cm). Derivadas parciais de variáveis de forma são avaliadas com dois tipos de campos de velocidades e comparados com valores obtidos por diferenças finitas. AS: análise de sensibilidade. MDF: diferenças finitas usando o respectivo campo de velocidades. A concordância considera o valor MDF como padrão.

Método da Camada Unitária de Contorno			Método de Deslocamentos Fictícios do Contorno		
AS	**MDF**	**Concordância %**	**AS**	**MDF**	**Concordância %**
-0.0017523326	-0.0017523071	99.998548052945	-0.0017534878	-0.0017534606	99.998446618064
0.0012778105	0.0012777974	99.998975935058	0.0012791181	0.0012791040	99.998896062829
-0.0017630652	-0.0017630241	99.997669364586	-0.0017637705	-0.0017637260	99.997471659777
0.0014104513	0.0014104212	99.997868426920	0.0014111209	0.0014110872	99.997611686098
-0.0009327278	-0.0009327021	99.997245200413	-0.0009336276	-0.0009336019	99.997246986157
0.0007539620	0.0007539421	99.997365382679	0.0007549281	0.0007549073	99.997245295108
0.0009984764	0.0009983410	99.986429879736	0.0009953933	0.0009952503	99.985627096994
-0.0010767538	-0.0010764613	99.972826879739	-0.0010763548	-0.0010760067	99.967650704526
0.0003633810	0.0003633813	99.999913104656	0.0003620029	0.0003620161	99.996358690078
-0.0000583693	-0.0000583381	99.946478533310	-0.0000621015	-0.0000620539	99.923322217446

Tabela 4.11: Gradientes do funcional de energia de deformação (0.0000228 kNcm). Derivadas parciais de variáveis de forma são avaliadas com dois tipos de campos de velocidades e comparados com valores obtidos por diferenças finitas. AS: análise de sensibilidade. MDF: diferenças finitas usando o respectivo campo de velocidades. A concordância considera o valor MDF como padrão.

Método da Camada Unitária de Contorno			Método de Deslocamentos Fictícios do Contorno		
AS	**MDF**	**Concordância %**	**AS**	**MDF**	**Concordância %**
-0.0000006801	-0.0000006799	99.965342338772	-0.0000006803	-0.0000006803	99.997638088816
0.0000005280	0.0000005280	99.988167924395	0.0000005277	0.0000005286	99.841380411815
-0.0000009385	-0.0000009378	99.931167546422	-0.0000009396	-0.0000009384	99.869308426532
0.0000007510	0.0000007504	99.930857786584	0.0000007521	0.0000007510	99.849821468046
-0.0000005718	-0.0000005714	99.933201460971	-0.0000005716	-0.0000005720	99.929173277493
0.0000004614	0.0000004610	99.914396376248	0.0000004627	0.0000004618	99.791434344582
0.0000008556	0.0000008250	96.291535530547	0.0000008397	0.0000008219	97.840223768537
-0.0000008536	-0.0000008441	98.869357693318	-0.0000008783	-0.0000008449	96.041783345959
0.0000003289	0.0000003351	98.139261653117	0.0000003196	0.0000003335	95.831417006109
-0.0000000508	-0.0000000509	99.911783794551	-0.0000000583	-0.0000000551	94.278084088435

problemas não-lineares é o mesmo de problemas lineares (Tabelas 4.1 a 4.4).

Se o intervalo de execução da análise de resposta for tomado como unidade de referência (ou seja, os tempos de execução serão normalizados pelo intervalo de análise de resposta[8]), o método adjunto de análise de sensibilidade com método de camada unitária de contorno é concluído no tempo normalizado de 0.003571 (4 gradientes e 10 variáveis). Com método de deslocamentos fictícios do contorno, os gradientes são obtidos no tempo normalizado de 0.017857. O cálculo de gradientes pelo método de diferença finita central requer 19.86 vezes o tempo da análise de resposta (2 análises para cada variável). O tempo de avaliação de gradientes por diferenças finitas poderia ser reduzido à metade caso fossem utilizadas técnicas de diferenças à frente ou atrás, porém com o custo adicional da dificuldade de determinação do tamanho da perturbação em cada variável, muito mais crítico nessas duas técnicas.

Independente do algoritmo de solução de sistemas lineares utilizado na solução, a vantagem da análise de sensibilidade em relação a diferenças finitas é evidente. Se os sistemas lineares são resolvidos por eliminação de Gauss, como nos resultados acima, o método adjunto já recebe a última matriz de rigidez tangente triangularizada, sendo o problema adjunto é resolvido por substituição apenas, tornando o algoritmo ainda mais eficiente.

4.4.2 Análise de Resposta e Otimização de Coxins de Suspensão

Modelo Bidimensional

Coxins de suspensão são elementos compostos de borracha e metal, geralmente aço, projetados para funcionar como molas de rigidez baixa e aproximadamente constante dentro do intervalo de trabalho. Esses componentes são utilizados como interfaces entre subconjuntos de sistemas mecânicos para isolar ou absorver vibrações. Um exemplo de aplicação são os coxins do conjunto do trem de força de automóveis (motor, embreagem e câmbio). Esse conjunto é fixado ao chassi ou carroceria do veículo por um sistema de suspensão composto de braços e coxins, elementos de borracha que atuam como molas.

A partir das freqüências naturais e modos de vibração de cada subsistema, determinam-se o posicionamento ótimo e a rigidez necessária nos coxins. O passo seguinte consiste em adotar componentes

[8] Essa normalização é necessária pois as análises foram executadas em sistemas com performance bastante diferente. A normalização fornece uma base de comparação objetiva do custo de solução, independente do sistema utilizado.

que mais se aproximem dessas características de rigidez ou projetar novos coxins que cumpram as especificações de forma ótima. Os recursos de análise de resposta, análise de sensibilidade e otimização desenvolvidos anteriormente serão utilizados para obter tais componentes de forma sistemática.

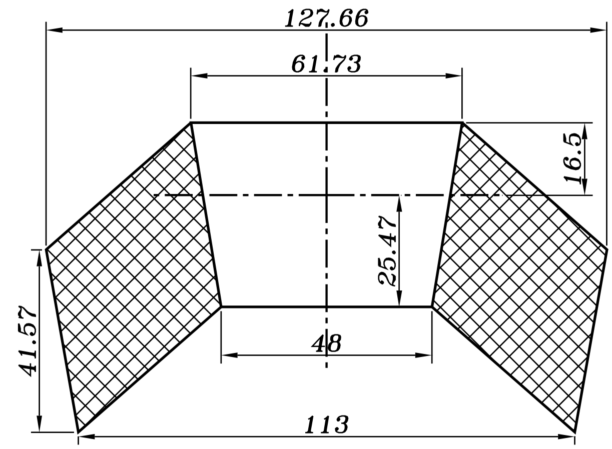


Figura 4.16: Geometria inicial do coxim.

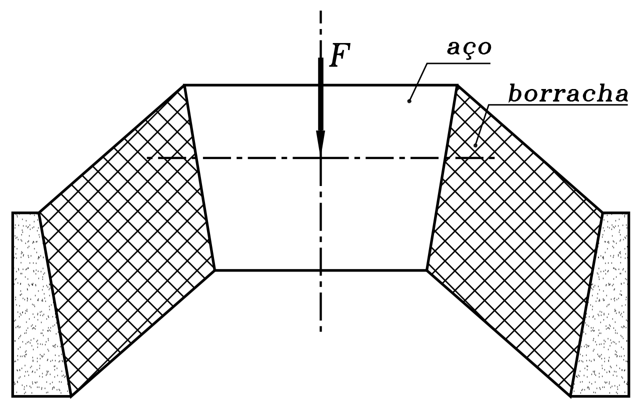


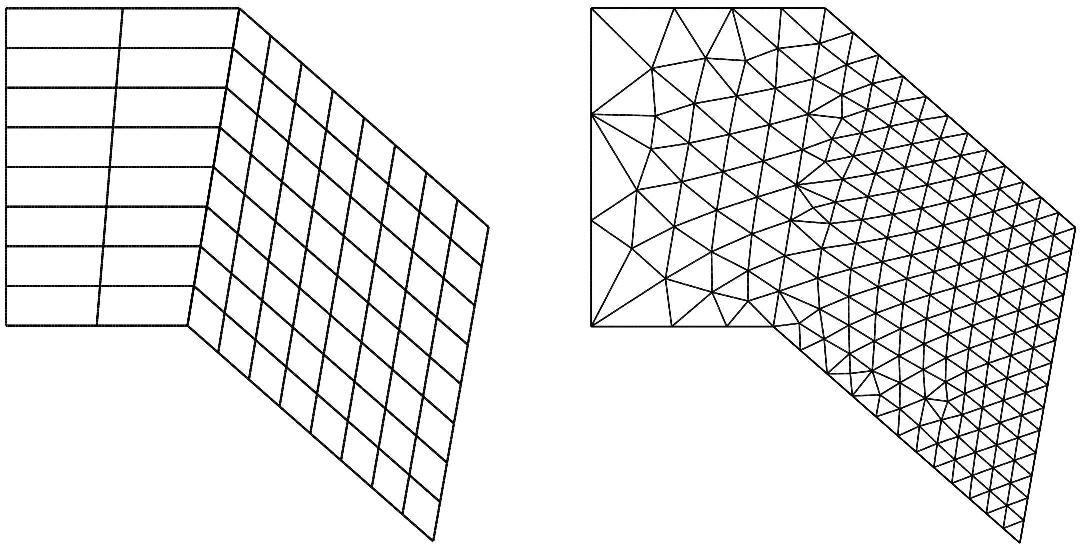
Figura 4.17: Condições de contorno, carregamento e materiais (Borracha: Mooney-Rivlin, $A_{10} = 0.138740$ N/mm^2, $A_{01} = 0.05933$ N/mm^2, $\tilde{\kappa} = 6599.2$ N/mm^2. Aço: $E = 207 \times 10^3$ N/mm^2, $\nu = 0.3$.).

Para um dada aplicação, necessita-se de coxim com valor determinado de rigidez vertical. Inicialmente, serão investigadas as características de um coxim com a geometria mostrada na Figura 4.16 sujeito às condições de contorno, carregamento e propriedades de material da Figura 4.17, num modelo de deformação plana. Em seguida, a geometria desse componente será modificada para cumprir as especificações do projeto de suspensão do trem de força.

Utilizando as duas discretizações mostradas na Figura 4.18, um deslocamento vertical de 9.0 mm foi imposto na face superior do núcleo de aço em nove passos de carregamento iguais. Os resultados de força por deslocamento nos dois casos são comparados na Figura 4.20. Observa-se que as duas discretizações fornecem respostas praticamente idênticas de rigidez.

De acordo com os gráficos da Figura 4.20, a rigidez vertical desse componente em torno do deslocamento de -3.0 mm é de 1.216 N/mm, enquanto que a aplicação exige 0.810 N/mm. Ou seja, o deslocamento de -3.0 mm desse ser atingido quando for aplicada uma força de -1.215 N no modelo considerando a simetria.

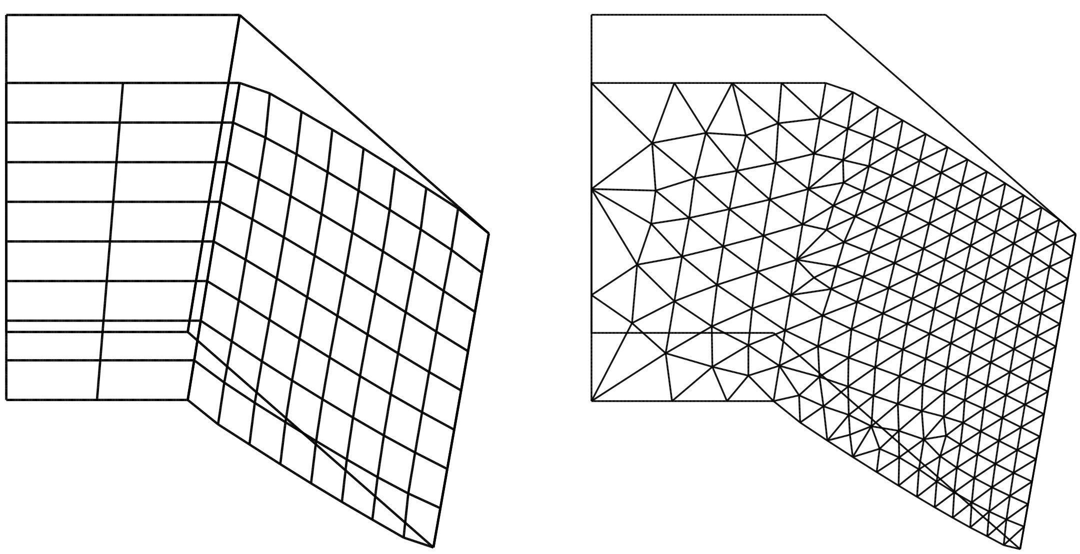
Como a rigidez do coxim deve ser reduzida, isso pode ser atingido diminuindo-se o volume de borracha (o que também permite reduzir o custo do componente). Em termos de um problema de



(a) Malha estruturada de elementos quadrangulares.

(b) Malha não-estruturada de elementos triagulares.

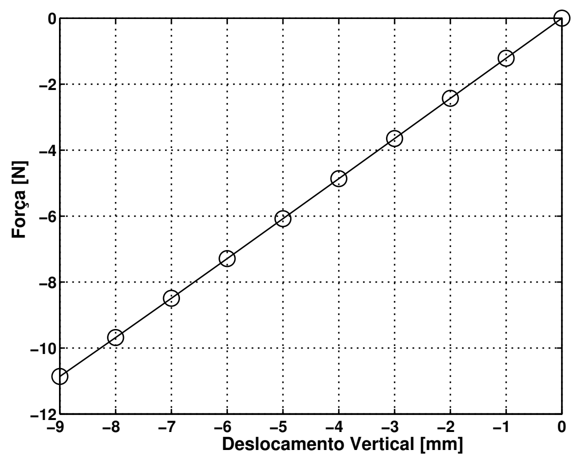
Figura 4.18: Discretizações para solução do problema. Volume de borracha: 1571.14802762 mm^3.



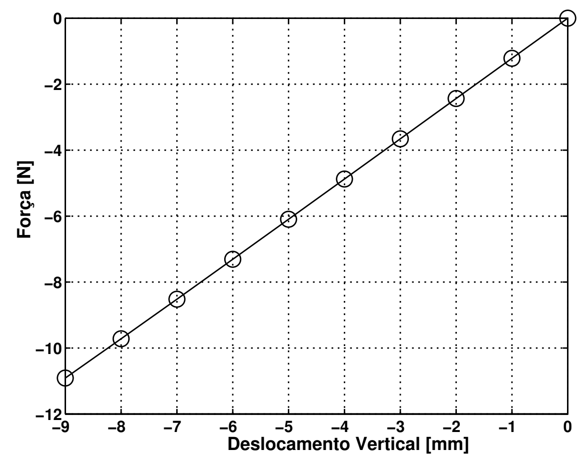
(a) Elementos quadrandulares.

(b) Elementos triangulares.

Figura 4.19: Geometrias deformadas com deslocamento vertical de 9.0 mm (escala 1:1).



(a) Elementos quadrandulares.



(b) Elementos triangulares.

Figura 4.20: Curvas de força por deslocamento vertical com cada discretização.

Tabela 4.12: Valores das variáveis de projeto: inicial, limites máximos e mínimos, projeto ótimo [mm].

Variável	Inicial	Mínimo	Máximo	Ótimo
1	117.45800000	114.00	119.50	115.32006385
2	74.21400000	50.00	80.00	67.74753867
3	127.34750000	120.00	134.00	125.65323697
4	65.53500000	40.00	80.00	56.16166904
5	137.23700000	135.00	140.00	136.87073865
6	56.85600000	45.00	70.00	49.10233489
7	110.50000000	107.00	112.00	110.29317133
8	32.32400000	30.00	40.00	38.72662048
9	120.25000000	113.00	127.00	121.79908887
10	23.76500000	15.00	35.00	32.65529169
11	130.00000000	128.00	133.00	131.71769505
12	15.20600000	12.00	30.00	21.15286329

otimização, o objetivo é minimizar o volume de borracha exigindo-se que o deslocamento do núcleo de borracha seja 3.0 mm para uma força vertical de -1.215 N. A parametrização da geometria é mostrada na Figura 4.21, sendo as variáveis de projeto as coordenadas x e y dos pontos de controle internos de duas curvas B-splines de grau 2 com 5 pontos de controle. Os valores iniciais e os limites máximos e mínimos de cada uma das 12 variáveis de projeto são mostrados na Tabela 4.12. Uma restrição adicional é que a energia de deformação não ultrapasse de 2.5 Nmm. Logo, tem-se um total de 25 restrições de desigualdade (cada limite induz uma inequação de restrição, além da restrição na energia de deformação) e uma restrição de igualdade (deslocamento do núcleo de aço).

Aplicou-se então o algoritmo de pontos interiores de Herskovits combinado com método adjunto de análise de sensibilidade e atualização de malhas através de deslocamentos fictícios do contorno.

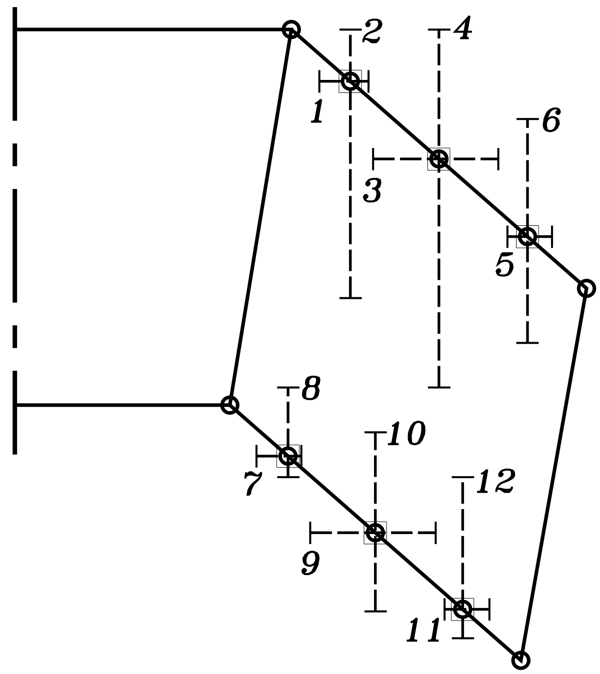


Figura 4.21: Variáveis de projeto e respectivos limites.

Apesar de possibilitar uma solução mais eficiente, a discretização estruturada em quadriláteros não pode ser utilizada, pois as primeiras tentativas de alterar a malha (com campos de velocidade e com gerador de malhas externo) eram seguidamente rejeitadas nos critérios de distorção descritos no Capítulo 4 para esse tipo de elemento, impedindo a progressão do algoritmo de otimização. Com a discretização não-estruturada em triângulos, a solução do problema é atingida em 7 iterações, com um total de 21 análises de resposta. As atualizações da malha de elementos finitos foram feitas apenas com informações dos campos de velocidades em todas as alterações do projeto, não sendo necessário regenerar a discretização com gerador externo. Ou seja, todas as malhas foram obtidas apenas reposicionando os nós. A evolução do volume de borracha na seqüência de iterações é mostrada na Figura 4.22 e o valor da restrição (normalizada) de igualdade de deslocamento é mostrada na Figura 4.23. Essa restrição se inicia violada, sendo satisfeita (dentro da precisão exigida) ao final do processo de otimização.

O projeto ótimo, cuja discretização é mostrada na Figura 4.24, é atingida quando se exige uma precisão de 0.0001 para o critério de convergência do algoritmo de minimização. Nesse ponto, o deslocamento vertical do núcleo de aço é de -2.99994492 mm e o volume de borracha é de 1079.00569837 mm^3, o que representa uma redução de 28.36% em relação ao volume original de 1571.14802762 mm^3. Deve-se observar que a qualidade da discretização final é bastante razoável. O histórico das variáveis de projeto é mostrado na Figura 4.25.

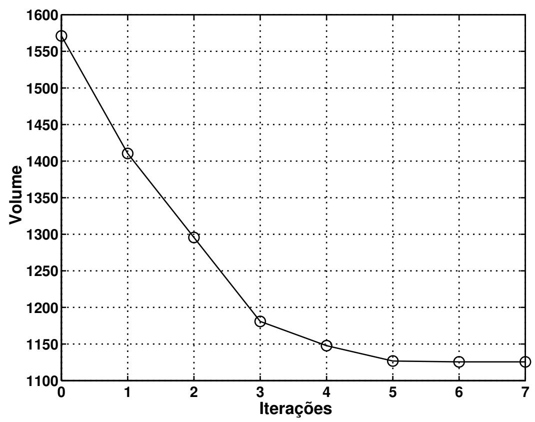


Figura 4.22: Evolução da função objetivo (volume de borracha).

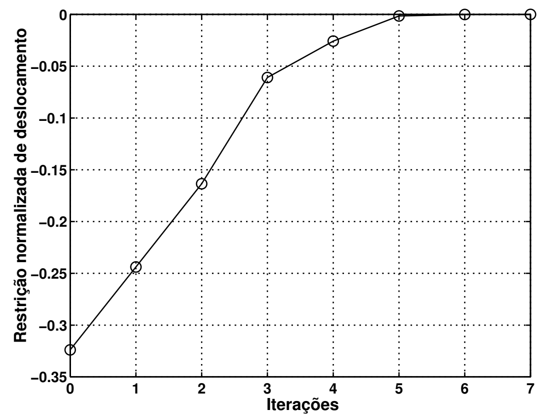
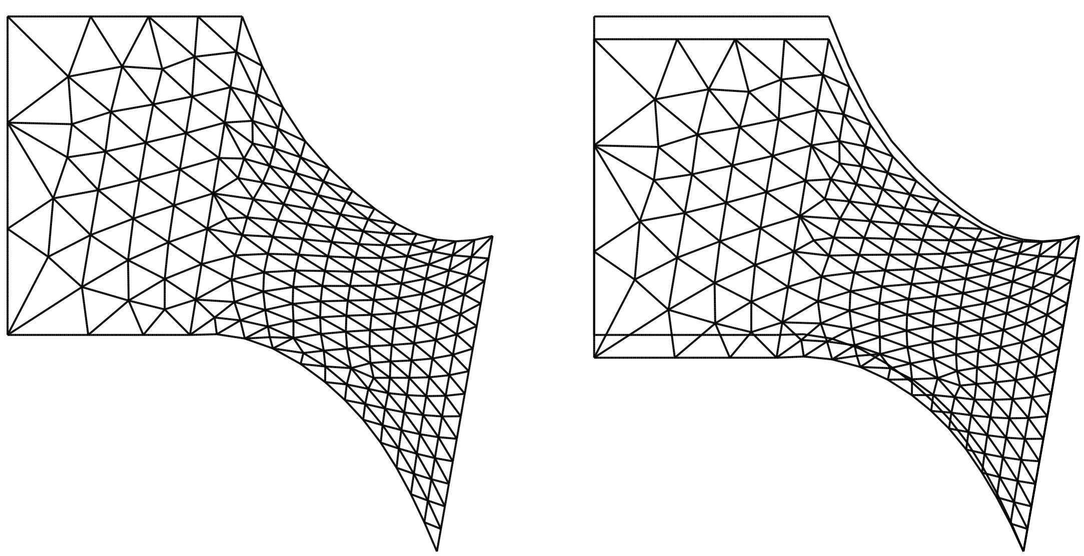


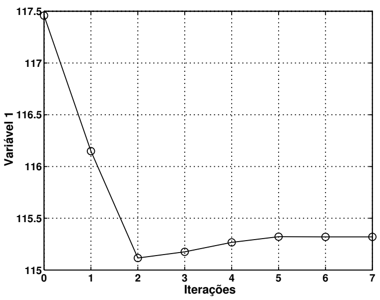
Figura 4.23: Evolução da restrição normalizada de deslocamento.



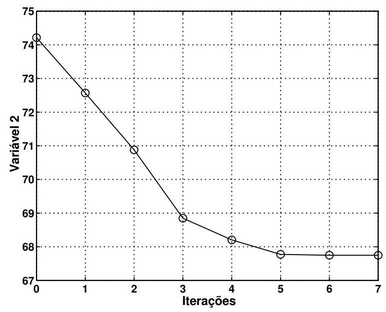
(a) Geometria não-deformada.

(b) Geometria deformada ($u_y = -2.99994492$ mm).

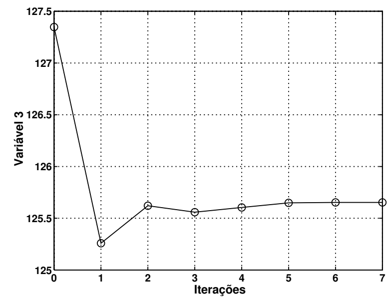
Figura 4.24: Projeto ótimo. Volume de borracha: 1125.55779686 mm^3. Energia de deformação: 1.84717475 Nmm.



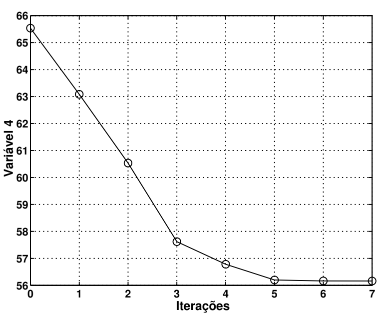
(a) Variável 1.



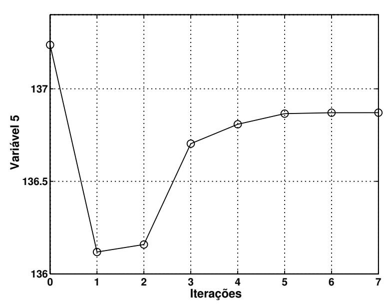
(b) Variável 2.



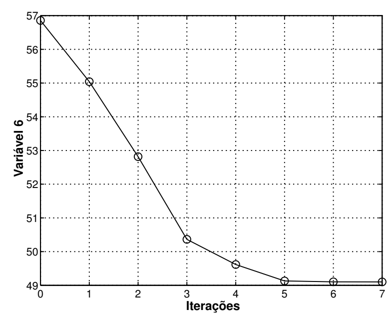
(c) Variável 3.



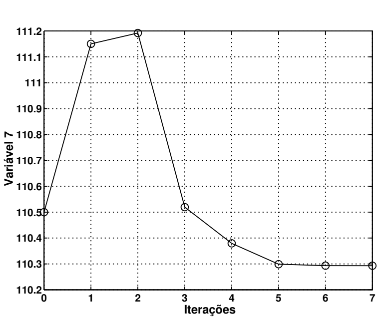
(d) Variável 4.



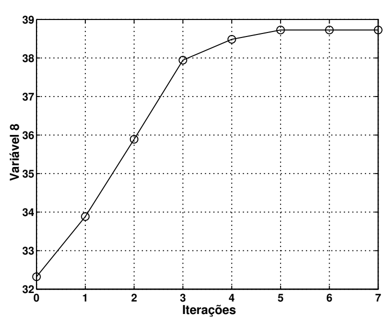
(e) Variável 5.



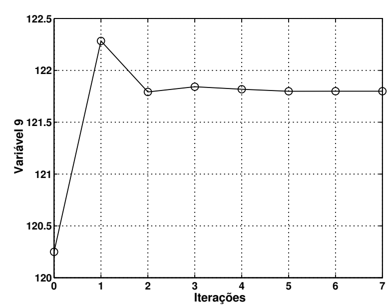
(f) Variável 6.



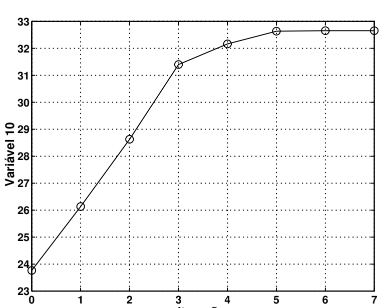
(g) Variável 7.



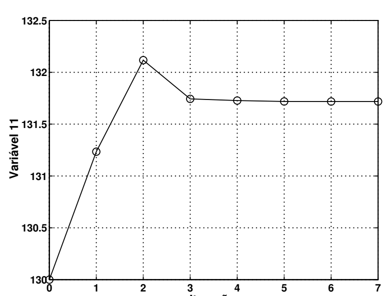
(h) Variável 8.



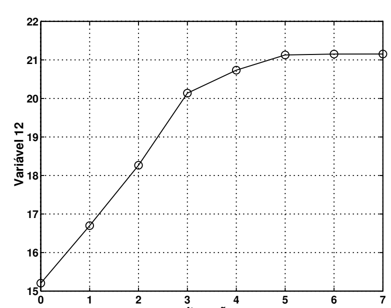
(i) Variável 9.



(j) Variável 10.



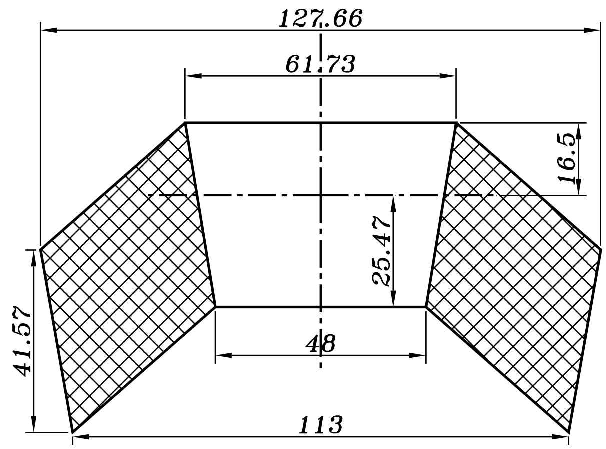
(k) Variável 11.



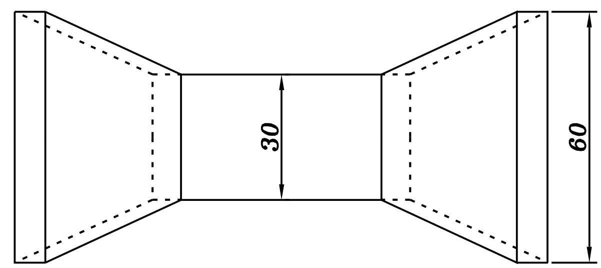
(l) Variável 12.

Figura 4.25: Histórico das variáveis de projeto [mm].

Modelo Tridimensional



(a)

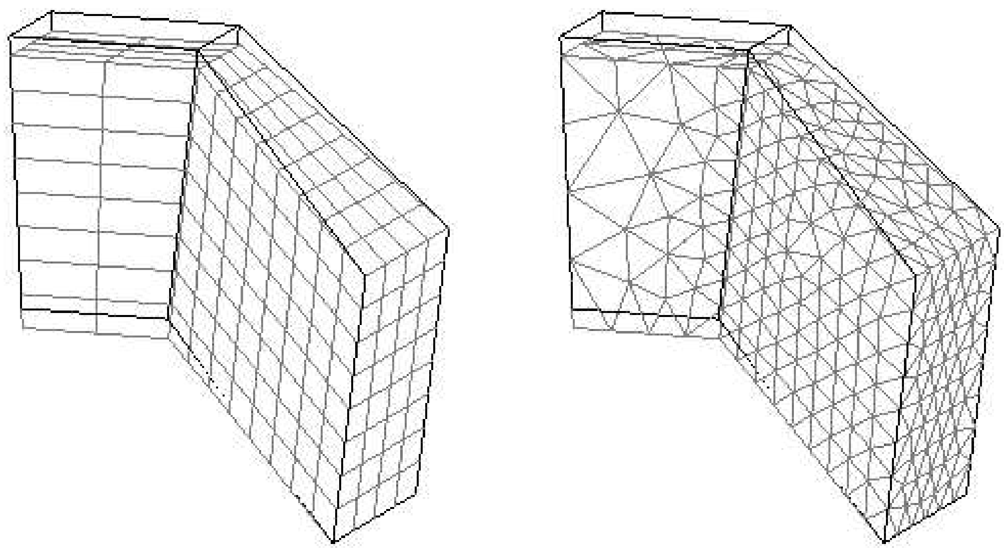


(b)

Figura 4.26: Geometria inicial do coxim.

Um modelo tridimensional de coxim é mostrado na Figura 4.26, sendo as condições de contorno semelhantes às mostradas na Figura 4.17. Utilizando as duas discretizações mostradas na Figura 4.27 (devido às simetrias nas direções x e z, apenas um quarto do componente foi modelado), um deslocamento vertical de 1.0 mm foi imposto na face superior do núcleo de aço em dois passos de carregamento iguais. Os resultados mostram uma rigidez de 40.684 N/mm na vizinhança desse deslocamento. A aplicação desse componente requer uma rigidez de 29.628 N/mm.

Da mesma forma que no exemplo anterior, formula-se um problema de otimização no qual o objetivo é reduzir o volume de borracha do componente, exigindo-se simultaneamente que o deslocamento do núcleo de aço seja de 1.35 mm para uma força vertical de -10.0 N no modelo de elementos finitos considerando a simetria (-40.0 N no modelo completo). A parametrização da geometria é mostrada na Figura 4.28, sendo as variáveis de projeto as coordenadas dos pontos de controle de 4 superfícies NURBS. Os valores iniciais e os limites máximos e mínimos de cada uma das 21 variáveis de projeto são mostrados na Tabela 4.13, juntamente com o significado de cada variável. Uma restrição adicional é que a energia de deformação não ultrapasse de 8.0 Nmm. Tem-se um total de 43 restrições de desigualdade e uma restrição de igualdade relacionada ao deslocamento do núcleo de aço. Por oferecer maior flexibilidade de modificação da geometria, a discretização em tetraedros foi utilizada no ciclo de otimização.



(a) Hexaedros. **(b)** Tetraedros

Figura 4.27: Discretizações da geometria inicial com deformação devido ao deslocamento.

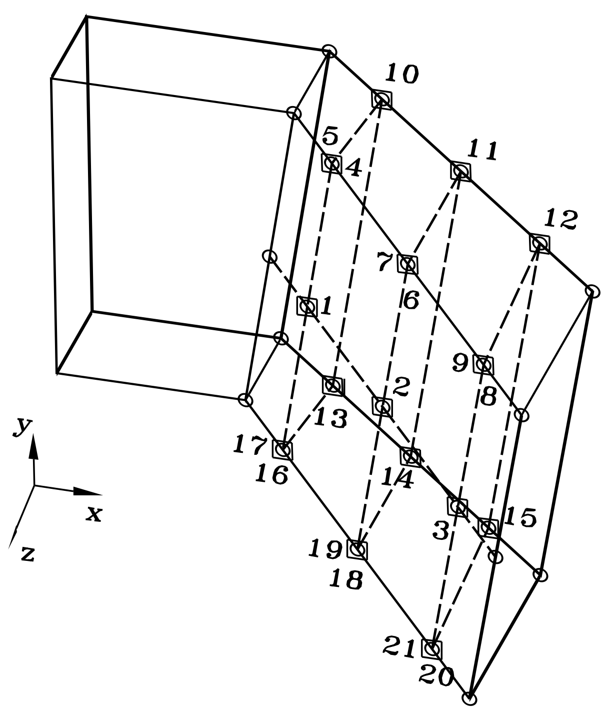


Figura 4.28: Variáveis de projeto e respectivos limites.

Tabela 4.13: Variáveis de projeto: significado, valores iniciais, limites máximos e mínimos, projeto ótimo [mm].

Variável	Significado	Inicial	Mínimo	Máximo	Ótimo
1	z	17.500000	14.0	18.5	14.01396844
2	z	22.500000	18.0	23.5	19.77219330
3	z	27.500000	23.5	28.5	23.51660888
4	y	55.891667	50.0	60.0	53.94129887
5	z	27.500000	23.5	28.5	23.52235259
6	y	65.535000	50.0	70.0	64.66501579
7	z	22.500000	20.0	24.5	24.45383121
8	y	75.178333	65.2	80.0	66.47478445
9	z	17.500000	15.0	18.5	16.57283684
10	y	74.214000	65.2	80.0	65.21299637
11	y	65.535000	50.0	70.0	61.87324420
12	y	56.856000	50.0	60.0	52.56456103
13	y	32.324000	28.0	35.0	34.84412684
14	y	23.765000	22.0	27.0	26.91077027
15	y	15.206000	14.0	18.5	18.47609098
16	y	14.255000	13.0	17.0	15.75345511
17	z	27.500000	23.5	28.5	23.52427786
18	y	23.765000	20.0	26.0	25.30741137
19	z	22.500000	18.0	23.5	18.39308939
20	y	33.275000	30.0	36.0	33.78082667
21	z	17.500000	14.0	18.5	14.95128675

O problema de otimização é resolvido em 15 iterações e 21 análises de resposta do algoritmo de pontos interiores de Herskovits combinado com método adjunto de análise de sensibilidade e atualização de malhas através de deslocamentos fictícios do contorno.

A evolução do volume de borracha na seqüência de iterações é mostrada na Figura 4.29. O projeto ótimo, cuja discretização é mostrada na Figura 4.30, é atingida quando se exige uma precisão de 0.0005 para o critério de convergência do algoritmo de minimização. Nesse ponto, o deslocamento vertical do núcleo de aço é de -1.34999357 mm e o volume de borracha é de 28897.08171555 mm^3, o que representa uma redução de 18.26% em relação ao volume original de 35351.15674344 mm^3. Deve-se observar que a qualidade da discretização final é bastante razoável.

4.5 Conclusões e Análise dos Resultados

Os resultados obtidos mostram que problemas de projeto da forma de componentes mecânicos descritos por modelos de material hiperelástico incompressível não-linear podem ser efetivamente resolvidos com os recursos de análise de resposta e sensibilidade descritos nos Capítulos 2 e 3 em conjunto com as técnicas de geração de campos de velocidades, atualização de malhas e otimização apresentadas neste capítulo.

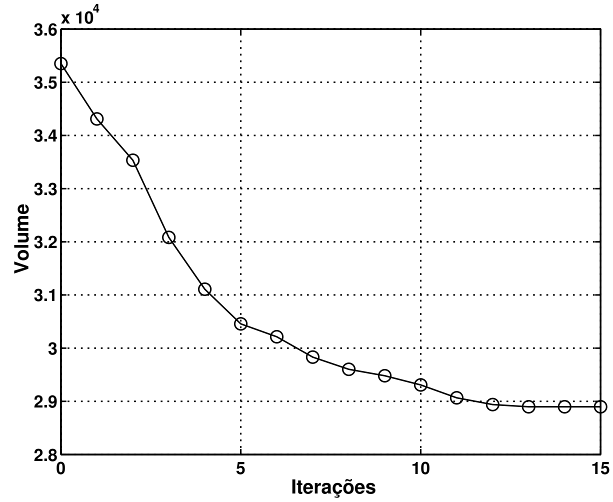
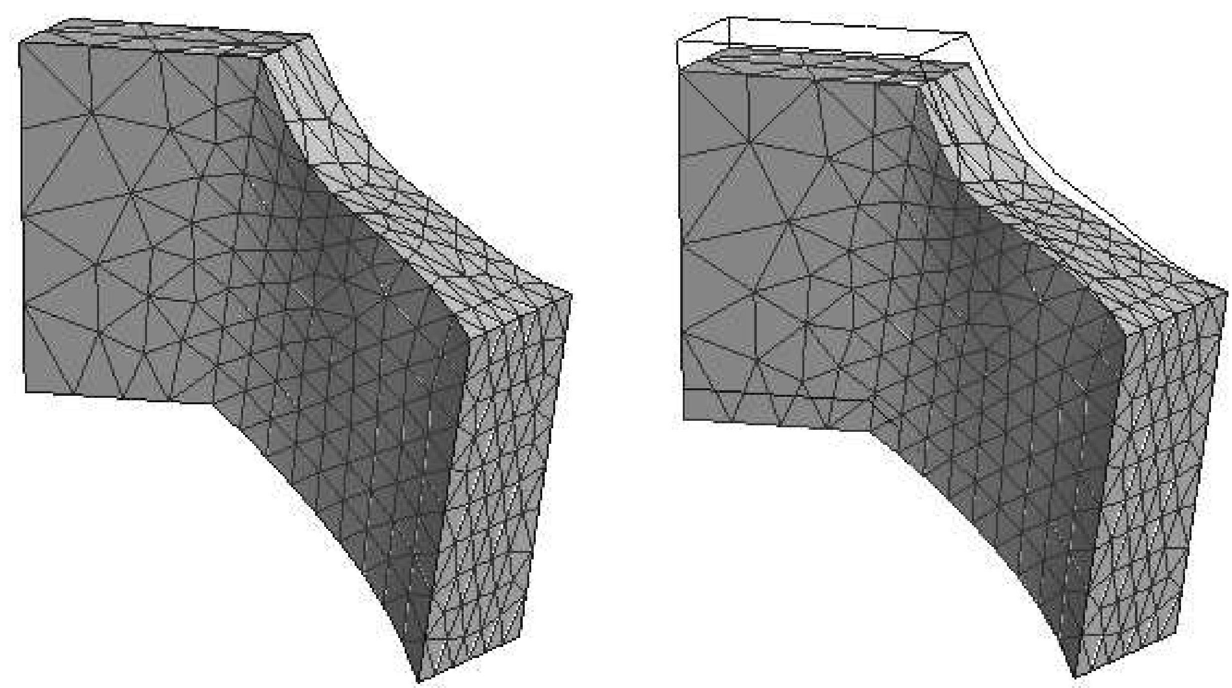


Figura 4.29: Evolução da função objetivo (volume de borracha).



(a) Geometria não-deformada.

(b) Geometria deformada ($u_y = -1.34999357$ mm).

Figura 4.30: Projeto ótimo. Volume de borracha: 28897.08171555 mm^3.

Como observado, o nível de precisão dos gradientes obtidos pela técnica de análise de sensibilidade é bastante satisfatório, sendo semelhante em problemas lineares e não-lineares. Essa característica permite que se tenha uma boa taxa de convergência efetiva do algoritmo de otimização estrutural, refletindo na necessidade de baixo número de iterações e análises de resposta. O recurso de combinar restrições de desigualdade e igualdade permite que, simultaneamente ao objetivo funcional (por exemplo, massa de elastômero – diretamente associada a custo), sejam atingidas características desejáveis tais como propriedades de rigidez. As reduções de massa de elastômero obtidas são expressivas, especialmente na hipótese de produção em escala.

As técnicas de campos de velocidades descritas neste capítulo permitem manter malhas com a mesma topologia no decorrer de várias iterações, especialmente próximo à convergência. Isso permite acelerar a solução de projetos consecutivos quando se aplica solução iterativa de sistemas lineares também na análise de resposta ao se adotar a solução do projeto anterior como estimativa inicial das estimativas de projeto da iteração corrente.

Nos exemplos, nota-se ainda que elementos quadriláteros e hexaédricos, apesar de exigirem menor esforço de solução, são limitados para a aplicação em otimização de forma devido à maior possibilidade de distorção geométrica. Nesse sentido, recursos de análise de resposta e sensibilidade com triângulos e tetraedros se tornam fundamentais.

Capítulo 5

Processo de Desenvolvimento de Software

A evolução da Mecânica Computacional, entendida como o desenvolvimento de modelos matemáticos de sistemas mecânicos e estudo de técnicas numéricas para solucioná-los, foi influenciada profundamente pelo grande aumento da capacidade de processamento e a difusão dos computadores ocorridos nos últimos 25 anos. Por sua vez, a crescente disponibilidade de poder de processamento gera expectativas quanto a possibilidade de aplicar modelagens mais refinadas e precisas na solução de problemas. Com isso, os sistemas de software que implementam as técnicas de solução têm ficado cada vez maiores e mais complexos, uma vez que são necessárias combinações de algoritmos e estruturas de dados também crescentemente mais complexas e especializadas. Num contexto mais geral, também se observa como tendência o aumento do tamanho e da complexidade dos sistemas de software, pois demandas por automação de atividades e manipulação de grandes volumes de dados, aliadas a exigências por interfaces mais intuitivas, compartilhamento de dados e aplicações via Internet, ocorrem em praticamente todos os casos em que sistemas computacionais são empregados.

O aumento da complexidade, por sua vez, torna projeto, especificação e organização global do sistema de software problemas tão grandes – ou mesmo maiores – que o desenvolvimento e implementação de algoritmos (Jacobson *et al.*, 1999). A complexidade aumenta os riscos e custos de desenvolvimento, pois dificulta a compreensão de requisitos e torna implementação e integração de componentes pouco previsíveis e produtivas. Como as demandas cresceram muito, o desenvolvimento dos sistemas de software atuais requer técnicas específicas de gerenciamento da complexidade.

Um corpo organizado de técnicas de desenvolvimento é denominado *processo*. Um processo define *quem* está fazendo *o quê*, *quando* e *como* para atingir um determinado *objetivo* (Jacobson *et al.*, 1999). Em engenharia de software, o objetivo é construir um novo sistema ou melhorar um sistema já existente. Assim, um processo de engenharia de software é um conjunto de atividades que visam transformar um conjunto de requisitos num sistema de software funcional.

No caso do desenvolvimento de software para a solução numérica de problemas de mecânica, é usual concentrar a discussão muito mais nas funcionalidades (ainda que de maneira informal), uma vez que há uma relação mais ou menos direta entre função e algoritmo ou conjunto de algoritmos, e muito pouco na estrutura – usualmente denominada *arquitetura* – do sistema. Contudo, é justamente a arquitetura que fornece a base na qual os diversos algoritmos e tecnologias interagem, definindo quais as possibilidades de evolução do sistema como um todo. Uma boa arquitetura é modular e tem interfaces bem definidas, ou seja, as várias partes do sistema têm contornos destacados, o que facilita o desenvolvimento e detecção de erros, caracterizando um sistema robusto e resiliente. Arquiteturas bem construídas podem crescer, ser atualizadas e agregar novas funcionalidades de forma segura e sustentável. Arquiteturas ruins são frágeis e tendem a se tornar instáveis ao sofrerem modificações sendo, portanto, de evolução lenta e difícil.

Arquiteturas devem ser descritas e desenvolvidas através de modelos. A modelagem de sistemas é uma técnica reconhecida e adotada em todos os campos da engenharia de sistemas físicos. Um modelo consiste de uma simplificação da realidade, focando-se em determinados aspectos e eliminando-se outros, irrelevantes naquele momento. Portanto, modelos são construídos para melhor entender o sistema em desenvolvimento, pois há limites na habilidade humana em analisar sistemas. Quanto mais complexo o sistema, maior a necessidade de modelá-lo com varios níveis de detalhes e de diversas perspectivas.

Porém, por mais resultados que a técnica de modelagem tenha mostrado no campo da engenharia de sistemas físicos, esta ainda não é parte integrante de muitas atividades de desenvolvimento de software. Isso ocorre apesar da quantidade de recursos humanos e financeiros ser geralmente similar ao envolvido na produção de um sistema físico. No campo da Mecânica Computacional, por exemplo, a preocupação com questões de modelagem de sistemas é recente (Silva e Bittencourt, 2000; Sotelino *et al.*, 1998; Besson e Foerch, 1997; Yu e Kumar, 2001).

Ao aplicar orientação por objetos na implementação de software para a solução numérica de problemas, tende-se a dispensar uma atenção maior à relação entre funções e estrutura do sistema, devido às próprias características dessa técnica [1]. De fato, se comparada com a programação orientada por algoritmos, a programação orientada por objetos permite a construção de sistemas mais robustos e ganhos de produtividade pois, sendo uma tecnologia de implementação superior, aumenta o limite de gerenciamento de sistemas software mais complexos mesmo na ausência de técnicas de modelagem ou processo de desenvolvimento. Mas atingido esse limite, são encontrados os mesmos problemas de manutenção e instabilidade de sistemas. Isto ocorre porque sem recursos de modelagem, gerencia-se o código fonte diretamente, comprometendo a obtenção da arquitetura adequada para o sistema, uma vez que se perde o foco do desenvolvimento ao se preocupar com detalhes de implementação antes de planejar com mais cuidado as características globais do sistema. Segundo, sem um processo como suporte corre-se o risco de aplicar modelagem e técnicas de programação sem objetivo ou controle.

Outro fato geralmente neglicenciado é que as expectativas quanto às funcionalidades a serem implementadas mudam durante o desenvolvimento do sistema. De modo geral, tais expectativas são denominadas *requisitos*. Podem ser incluídos novos requisitos ou modificados aqueles já definidos. No caso de um software comercial, isso pode ser motivado por lançamentos de produtos concorrentes e, no ambiente de pesquisa, pelo surgimento de novas técnicas ou por uma melhor compreensão do problema em estudo. Além disso, um fator complicador e recorrente é que os requisitos somente são completamente compreendidos durante o decorrer do trabalho. Em muitos casos, somente após uma implementação do sistema estar concluída.

A mudança dos requisitos não é um fato negativo, principalmente quando significa uma compreensão melhor do problema a ser resolvido. Não estar preparado para isso desde o início da construção do sistema é a verdadeira fonte das dificuldades.

Um processo de desenvolvimento de software lida com a complexidade incluindo procedimentos específicos para a captação e gerenciamento de requisitos, definição de uma arquitetura que cumpra os requisitos identificados e administração de mudanças[2], ditando a execução de atividades, atribuição de tarefas e verificação de resultados.

[1] A técnica de orientação por objetos é baseada da definição entidades abstratas (classes) que agregam dados e funções com visibilidade controlada (encapsulamento de informações). Um sistema orientado por objetos é composto por interações entre instâncias de classes (objetos) através de suas interfaces visíveis (públicas). Sistemas orientados por objetos são projetados e construídos através da identificação de classes e a atribuição de responsabilidades e, portanto, a orientação por objetos induz maior preocupação com a estrutura do sistema e sua interação com as funcionalidades.

[2] Ferramentas CASE que dão suporte à execução do processo mantém relações entre os diversos documentos, diagramas e código-fonte, estabelecendo "trilhas" que permitem propagar as mudanças de forma coerente e organizada.

Organizações voltadas para o desenvolvimento de software, sejam elas empresas ou instituições de pesquisa, que consigam desenvolver os sistemas desejados de maneira estável, produtiva e previsível (ou seja, dentro do tempo e do orçamento) serão sustentáveis. Por isso, é necessário também discutir o nível mais alto de organização e de definição de características do sistema de software, e não apenas algoritmos numéricos isolados. É importante, entretanto, não perder o foco: o objetivo final é produzir um sistema de software que funcione como desejado. Processo, arquitetura, modelagem, gerenciamento de requisitos, bem como os documentos e diagramas são apenas meios que têm se mostrado eficientes para atingir esse fim.

Nesse trabalho, aplicou-se o *Processo Unificado* (Jacobson *et al.*, 1999), a *Linguagem de Modelagem Unificada* UML (Booch *et al.*, 1999; Rumbaugh *et al.*, 1999) e programação orientada por objetos em C++ (Lippman, 1991) como técnicas de desenvolvimento de software. Discutem-se, a seguir, conceitos do processo e linguagem de modelagem citados. Por fim, exemplifica-se o uso desses conceitos na definição da arquitetura básica (*framework*) para implementação dos algoritmos de análise de resposta, análise de sensibilidade e otimização apresentados nos capítulos anteriores.

5.1 Processo Unificado de Desenvolvimento de Software

Como citado acima, um processo consiste num conjunto de procedimentos que define *quem* está fazendo *o quê*, *quando* e *como* para atingir um determinado *objetivo*. Em engenharia de software, o objetivo é construir ou atualizar um sistema de software. A aplicação de um processo formalmente constituído tem a vantagem de trazer a experiência acumulada por seus autores em grande número de projetos, nas mais diversas áreas, de forma estruturada e concentrada. Quanto menor a tradição da organização no uso de processos de desenvolvimento, maior a utilidade de tal experência acumulada.

O Processo Unificado (Jacobson *et al.*, 1999) é um processo de engenharia de software desenvolvido através da fusão e revisão de técnicas de engenharia de software estabelecidas e difundidas (Booch, 1991; Martin e Odell, 1998; Jacobson *et al.*, 1994; Rumbaugh *et al.*, 1991) com técnicas reconhecidas de gerência de projetos (Project Management Institute, 2000; Martins, 2002) e desenvolvimento de produtos (Cantor, 2001). Uma versão proprietária desse processo, denominada *Rational Unified Process* (RUP) (Rational Software Corporation Inc., 2000; Kruchten, 2001c), foi desenvolvida pela Rational Software Corporation a partir do processo original, reescrito em formato de livro eletrônico juntamente com *templates* de documentação e guias de execução de tarefas. O RUP também é suportado por um conjunto de ferramentas CASE (*Computer Aided Software Engineering*), sendo o mesmo projetado e mantido como uma ferramenta de software aplicando seus próprios conceitos.

O Processo Unificado captura práticas atualmente mais recomendadas de desenvolvimento de software tais como:

- Gerenciamento de requisitos;
- Gerenciamento de riscos;
- Desenvolvimento de arquiteturas modulares;
- Modelagem visual de software (aplicando a linguagem de modelagem UML);
- Desenvolvimento iterativo e incremental;
- Verificação contínua da qualidade do software;
- Controle de mudanças.

Dada a sua ênfase em desenvolvimento iterativo e gerenciamento, nesse processo também se recomenda explicitamente a adoção de ferramentas de apoio para automação de tarefas e sincronização de documentos, modelos e código. O Processo Unificado pode ser inclusive considerado um *framework* de processo de desenvolvimento, pois apresenta uma estrutura genérica que pode ser especializada dependendo do tipo de organização, de sistema a ser desenvolvido e do tipo de área de aplicação, mas mantendo o núcleo básico de técnicas e diretrizes (Augustine, 2001; Kruchten, 2002b; Kruchten, 2002a; Probasco, 2000; Evans, 2001). Organizações menores tendem a aplicar processos formais de forma menos burocrática, pois em geral tem-se maior facilidade de comunicação. Por sua vez, organizações de maior porte obtém vantagens com a aplicação rigorosa dos procedimentos.

Independentemente da especialização, o Processo Unificado se caracteriza por ser

- dirigido por *casos de uso* (use cases),
- centrado na construção da arquitetura,
- iterativo e incremental.

Essas características centrais são direcionadas para lidar com as especificidades do desenvolvimento de software e, em torno delas, se estruturam todas as práticas e técnicas estabelecidas por esse processo.

5.1.1 Processo Dirigido por Casos de Uso (*Use Cases*)

Caso de uso é definido como uma seqüência de ações realizadas pelo sistema que fornecem ao usuário um resultado de valor (Jacobson *et al.*, 1999). No presente contexto, um usuário é qualquer agente externo (pessoa ou outro sistema) que irá interagir com o sistema. Basicamente, casos de uso capturam requisitos funcionais do sistema. A inovação da aplicação de casos de uso está em juntar aos tradicionais requisitos funcionais o conceito de usuário, forçando os desenvolvedores a pensar na agregação de valor para o usuário e não apenas nas funções que o sistema deva ter (Bittner, 2000).

Além de capturarem e documentarem os requisitos funcionais do sistema, os casos de uso também dirigem as etapas de análise, projeto e testes.

Nas etapas de análise e projeto, definem-se as entidades (tipos abstratos de dados, classes) que executam (ou "realizam") as ações de um conjunto de casos de uso. Logo, as colaborações entre as entidades devem fornecer as funcionalidades especificadas nos use cases. Tais colaborações dão origem às relações estruturais entre as classes do sistema. A fase de testes, por sua vez, consiste na verificação da execução correta dos use cases.

Freqüentemente, num sistema orientado por objetos tradicional, é difícil compreender como o sistema realiza o que se espera dele. Em outras palavras, as interações entre objetos que executam as funcionalidades do sistema geralmente não são claras. Essa dificuldade se origina da falta de uma conexão visível entre as entidades ao executar suas funções do sistema, pois as responsabilidades ficam divididas entre elas. Ao dirigir o processo, os casos de uso fornecem tal conexão.

Observe que os casos de uso não são selecionados para implementação em determinada iteração de forma independente ou aleatória. Arquitetura e casos de uso evoluem simultaneamente, sendo o desenvolvimento iterativo fundamental para construção e refinamento paralelos de ambos. Devem ser escolhidos para as primeiras iterações aqueles casos de uso de maior risco e que descrevam as funcionalidades principais do sistema, ou seja, aqueles com maior impacto na arquitetura. Por sua vez, as primeiras versões de arquitetura fornecem linhas gerais para a execução de outros casos de uso.

Casos de uso são individualmente descritos na forma de texto e estruturados em conjunto de diagramas. No texto, tem-se o fluxo de eventos envolvido na execução do caso de uso, enquanto que os diagramas descrevem como os use cases se relacionam através de relações de especialização, inclusão (o fluxo de eventos de um caso de uso requer a execução de outros casos de uso) ou extensão.

5.1.2 Processo Centrado na Construção da Arquitetura

A arquitetura é uma visão do sistema com que todos os desenvolvedores devem concordar sobre sua organização, elementos estruturais que o compõem e suas interfaces, composição dos elementos estruturais e comportamentais em subsistemas progressivamente maiores, estilo e padrões a serem seguidos. Em casos nos quais já existe um sistema prévio que executa algumas funções do sistema proposto, compreender seu funcionamento, com pouca ou nenhuma documentação, e qual código deve ser reutilizado fazem parte da definição de uma arquitetura.

O conceito de arquitetura agrupa os aspectos estáticos e dinâmicos mais significativos de um sistema de software. Nasce das necessidades como identificadas pelos usuários e *stakeholders*[3] refletidas nos casos de uso, mas também é influenciada pela plataforma na qual o software funcionará, blocos reutilizáveis existentes, considerações de distribuição do produto a ser desenvolvido, compatibilidade com sistemas anteriores e requisitos não-funcionais.

A arquitetura de um sistema de software é uma visão global do sistema com as características mais importantes em destaque, deixando detalhes de lado. Como o que é importante depende em parte de julgamento – que vem com a experiência – a qualidade da arquitetura produzida vai depender da pessoa ou grupo encarregado de desenvolvê-la. De fato, uma das críticas que pode ser feita à engenharia de software em geral, e ao Processo Unificado em particular, é sua dependência crucial em relação à experiência. Isso porém independe da técnica de desenvolvimento e sim da inexistência de um equivalente às leis físicas para o domínio do desenvolvimento de software. Por isso, os processos estabelecem práticas recomendadas mas não práticas garantidas. Com isso, é impossível avaliar antecipadamente a qualidade de qualquer arquitetura além da mais básica antes de começar a desenvolvê-la.

Como citado anteriormente, casos de uso e arquitetura são intrinsecamente relacionados: a ar-

[3]Stakeholders são indivíduos ou organizações que estão ativamente envolvidos no projeto, ou cujos interesses podem ser positiva ou negativamente afetados pelos resultados do projeto (Project Management Institute, 2000). Mantém-se o termo em inglês pela falta de equivalente em português nesse contexto.

quitetura representa a forma do sistema, enquanto os casos de uso definem sua função (é uma certa simplificação dizer que apenas os casos de uso dirigem o Processo Unificado). Dessa forma, os dois aspectos devem estar equilibrados, ou seja, as execuções de casos de uso devem se encaixar de forma natural na arquitetura e a arquitetura deve permitir a realização dos casos de uso presentes e futuros. Essa última característica, a resiliência à mudanças, pode ser usada para julgar a qualidade da arquitetura. Em outras palavras, uma arquitetura conveniente permite que novos casos de uso sejam realizados pelo sistema de forma natural ou com mudanças localizadas e incrementais. O sistema em questão neste trabalho, por exemplo, tem sido aplicado com sucesso em problemas fora de seu escopo inicial (Driemeier, 2002; Ferreira, 2002; Nogueira Jr., 2002).

A arquitetura deve ser então concebida para permitir a evolução do sistema. Nesse sentido é necessário reconhecer que o sistema de software a ser produzido não pode ser somente um conjunto de códigos executáveis. Todos os documentos e diagramas associados são suas partes integrantes, uma vez que fornecem às pessoas envolvidas no desenvolvimento, subsídios fundamentais para entendê-lo e modificá-lo de versão em versão. Documentos e diagramas de requisitos, interações e colaborações entre classes e subsistemas, relações estruturais e dinâmicas entre classes, especificações de interfaces entre componentes e planos de testes são exemplos de partes integrantes de um sistema de software, ainda que não sejam visíveis ao usuário final. Por esse motivo, a descrição da arquitetura de um sistema consiste numa compilação resumida dos aspectos mais importantes dos diversos documentos e diagramas de forma a dar uma visão geral de suas características. Uma parte integrante do Processo Unificado é a utilização da UML como linguagem de modelagem visual na composição de documentos e diagramas.

Portanto, a arquitetura é necessária para entender o sistema, organizar o seu desenvolvimento, atualizá-lo e induzir reutilização de código, componentes e subsistemas. Um dos objetivos principais do Processo Unificado é definir e validar uma arquitetura já nas primeiras iterações do desenvolvimento (quando pouco foi investido em termos de tempo e recursos) de forma a estabelecer uma base estável para o período seguinte no qual a implementação é mais intensiva.

5.1.3 Processo Iterativo e Incremental

Ao construir um sistema de software, busca-se equilibrar casos de uso (função) e arquitetura (forma): casos de uso não são compreendidos em sua totalidade sem uma arquitetura que oriente sua análise e a própria definição da arquitetura decorre do conjunto de casos de uso. Essa dependência recursiva indica que casos de uso e arquitetura devam ser desenvolvidos e refinados simultaneamente de forma iterativa e incremental até a conclusão do sistema.

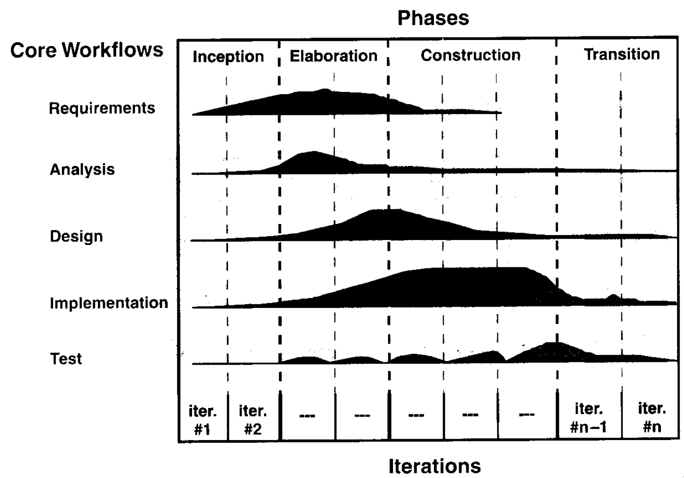


Figura 5.1: Ênfase nas atividades no decorrer das fases do Processo Unificado (Jacobson *et al.*, 1999).

No Processo Unificado, as iterações são agrupadas em fases, cada qual com um objetivo bem definido. Dentro de cada iteração, tem-se de forma reduzida as atividades centrais usuais de desenvolvimento de um software, ou seja, requisitos, análise, projeto, implementação e testes, como mostra a Figura 5.1. Nessa figura, tem-se o aspecto bidimensional da organização do trabalho proposta nesse processo: na horizontal tem-se a evolução do tempo caracterizada pela seqüência de iterações e fases; na vertical tem-se as atividades executadas em cada iteração. O produto das duas direções é a ênfase dispensada a cada atividade no decorrer do tempo. Cada iteração do Processo Unificado é um mini-projeto de desenvolvimento de software no qual os documentos e diagramas são revisados e incrementalmente estendidos.

Cada uma das fases do processo é caracterizada por seus objetivos em relação à evolução do sistema como se segue:

- **Incepção:** o objetivo dessa fase é desenvolver uma visão do produto final, ou seja, deve-se definir *o quê* deve ser desenvolvido.
- **Elaboração:** nesta fase, deve-se estabelecer e validar uma arquitetura inicial que funcione como uma base sobre a qual o produto final será construído. As linhas gerais básicas da arquitetura devem estar definidas ao final dessa etapa e os principais riscos técnicos devem ter sido localizados e equacionados. Os casos de uso dirigem o Processo Unificado durante todo o ciclo de desenvol-

vimento, mas as atividades de projeto – ênfase maior da fase de Elaboração – concentram-se na arquitetura.

- **Construção:** objetivo aqui é finalizar a implementação do sistema com todas as funcionalidades, ainda que sem total confiabilidade. A arquitetura é estável no início dessa fase mas, como durante a implementação os desenvolvedores podem descobrir melhores formas de estruturar o sistema, modificações incrementais na arquitetura podem ocorrer.

- **Transição:** equivale ao período durante o qual o produto vai de sua versão beta até a versão final. O objetivo, portanto, é depurar o sistema, eliminando problemas em seu funcionamento e atingir o grau de estabilidade funcional esperado. Após atingir a versão final, essa etapa pode evoluir para fase de Incepção da versão seguinte do sistema.

Sempre há um grau de inovação na construção ou atualização de um sistema. Logo, sempre está envolvido algum grau de risco. A principal virtude de um desenvolvimento iterativo é incentivar a redução dos riscos de desenvolvimento. Primeiro, como cada fase tem objetivos bem definidos, é possível avaliar a evolução global do projeto. Segundo, especialmente no início da fase de elaboração, cada iteração pode ser planejada para atacar grupos pequenos de casos de uso de maior risco. Ao lidar com conjuntos menores de dificuldades, desenvolvedores ficam menos intimidados e tem-se maior previsão quanto à conclusão do trabalho.

Trabalhar iterativamente também incentiva a produção de arquiteturas robustas, a acomodação de mudanças nos requisitos e a adoção de mudanças táticas. De fato, com a construção gradual do sistema, as pessoas envolvidas sentem-se incentivadas a identificar e reportar problemas, pois a cada fase a quantidade de retrabalho é pequena. Com isso, as primeiras iterações da elaboração são dedicadas à acomodação entre arquitetura e os requisitos de maior importância. Nesse ponto, mudanças estruturais podem ser feitas rapidamente e com baixo custo, pois pouco código fonte foi produzido. Logo, mudanças nos requisitos principais são permitidas e em certo sentido desejáveis.

No decorrer do projeto, nas fases de Construção e Transição, a arquitetura se torna continuamente mais estável e os casos de uso a serem implementados em cada iteração podem ser acomodados sem maiores problemas, representando apenas adições incrementais num sistema crescentemente funcional. Ao produzir um seqüência de sistemas funcionais, testes de integração dos diversos componentes do sistema se tornam freqüentes revelando possíveis incompatilidades em tempo. Também é possível liberar uma versão do sistema sem a totalidade das características inicialmente propostas mas cumprindo os requisitos principais, como resposta tática a eventos externos à organização, tais como o lançamento de software concorrente. Mudanças táticas também podem estar relacionadas à decisão de adquirir certos componentes do sistema de terceiros ao invés de desenvolvê-los internamente.

O desenvolvimento iterativo também traz mais estímulo a todas as pessoas envolvidas no desenvolvimento, pois é possível observar e julgar o resultado de suas atividades freqüentemente e não somente ao final de todo o projeto. Com isso, cada desenvolvedor pode aprimorar o próprio trabalho continuamente e também observar e aprender com o trabalho dos demais, adquirindo uma razoável visão global do processo em poucas iterações. Com isso a própria aplicação do processo pode ser aprimorada a cada iteração.

5.2 Linguagem de Modelagem Unificada (Unified Modeling Language - UML)

A *Linguagem de Modelagem Unificada*, (*Unified Modeling Language* - UML), é uma linguagem de modelagem de sistemas de software orientados por objetos (Booch *et al.*, 1999; Fowler e Scott, 1997; Rumbaugh *et al.*, 1999). Essa linguagem foi obtida através da fusão e revisão de técnicas já estabelecidas de modelagem de software (Booch, 1991; Jacobson *et al.*, 1994; Rumbaugh *et al.*, 1991) e também por isso tem se tornado um padrão. Tem sido ainda aplicada com sucesso nos mais diversos domínios tais como negócios, telecomunicações, pesquisa científica, transporte e serviços baseados na Web. A UML tem como objetivo permitir a visualização, especificação e documentação que apoiem de maneira consistente a construção de sistemas de software.

Utilizando a UML é possível modelar e estruturar requisitos, analisar e projetar os aspectos conceituais e físicos dos sistemas. Por ser orientada por objetos, a UML é baseada na definição de entidades (classes), que possuem características (atributos) e responsabilidades (operações). As relações entre essas entidades são descritas em diagramas para modelagem estática (relações estruturais) e dinâmica (comportamento e funcionamento) da arquitetura do sistema.

As Figuras 5.2 a 5.8 mostram alguns diagramas utilizados no desenvolvimento de um sistema para análise de resposta, análise de sensibilidade e otimização de problemas mecânicos. As Figuras 5.2, 5.3

e 5.8 mostram aspectos estáticos do sistema, enquanto que as Figuras 5.4 a 5.6 mostram características dinâmicas. A Figura 5.7 exemplifica a modelagem de requisitos.

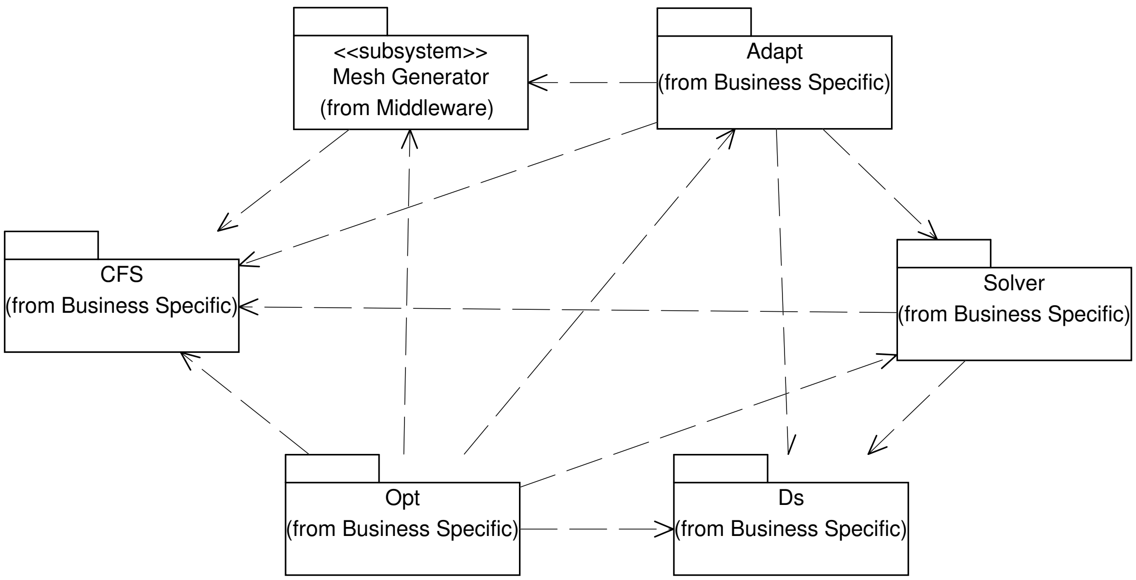


Figura 5.2: Diagrama de pacotes e subsistemas. As setas indicam dependência da origem em relação ao destino.

Na Figura 5.2, tem-se uma parte da arquitetura mostrando a dependência entre bibliotecas de classes num *diagrama de pacotes*. Pacotes são estruturas genéricas de organização na UML, sendo usados para agrupar quaisquer conjuntos de entidades relacionadas. Assim, é possível ter pacotes de casos de uso, pacotes de realizações de casos de uso, pacotes de classes, pacotes de componentes, pacotes de diagramas ou mesmo pacotes de entidades diversas. São aplicados especialmente para estruturar o modelo de projeto, dividindo-o em partes menores. É possível ainda controlar a visibilidade, determinando elementos públicos e privados. Um pacote tem apenas a semântica de conjunto, de forma que seu comportamento é dado pela combinação dos comportamentos de seus elementos.

Em alguns casos, é necessário combinar as semânticas de pacote e classe (comportamento), dando origem a subsistemas. Por convenção, todos os elementos internos de um subsistema são privados e suas funcionalidades são disponíveis apenas através de interfaces formalmente constituídas. Os subsistemas de um mesmo sistema de software podem ser desenvolvidos seguindo ciclos de vida independentes, contanto que as interfaces sejam mantidas. Subsistemas trazem grande resiliência à arquitetura, pois suas interfaces isolam o sistema de modificações internas. Na atividade de projeto, programas executáveis são modelados como subsistemas. Este é o caso do susbsistema Mesh Generator da Figura 5.2, o qual encapsula geradores de malhas adquiridos de terceiros.

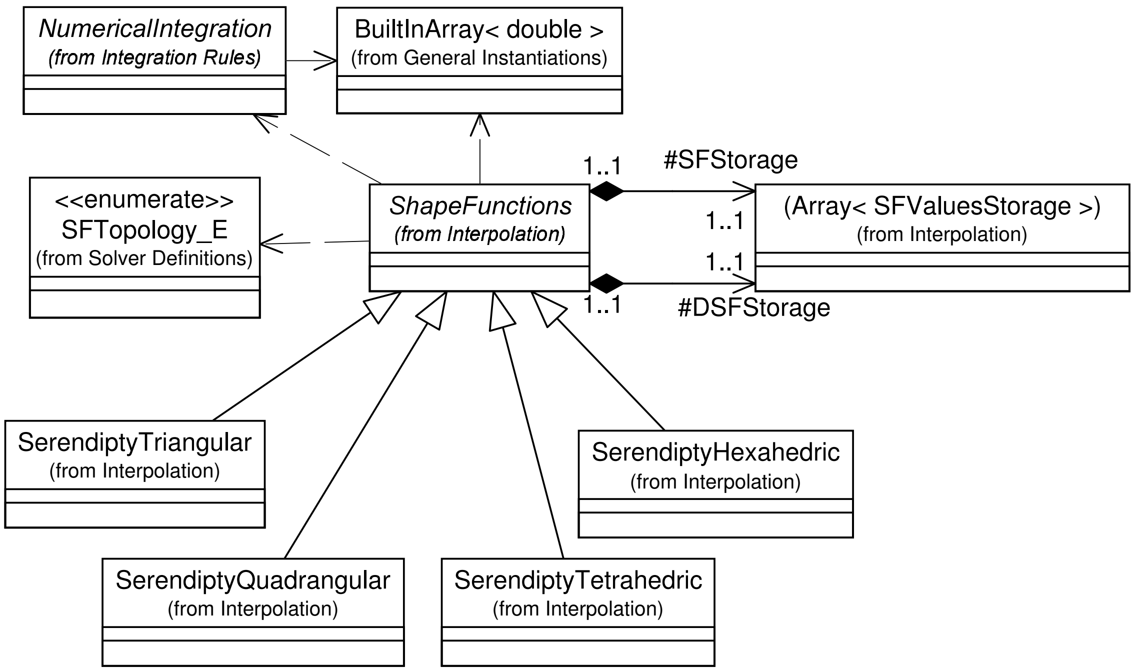


Figura 5.3: Diagrama de classes. Modelo de funções de forma.

Tem-se um diagrama de classes na Figura 5.3. Diagramas de classes, que são os mais utilizados da UML, descrevem dependências, associações e hierarquias entre classes, além de suas características internas. Dependências (linhas tracejadas) indicam que uma classe utiliza objetos de outra como atributos, parâmetros de operações ou variáveis locais. Associações (linhas contínuas opcionalmente com diamante e/ou setas nas extremidades) representam conexões formais de longa duração entre objetos de classes diferentes. Associações são semanticamente muito ricas, podendo especificar a cardinalidade da operação, papéis e visibilidade em cada sentido da associação. Podem também especificar uma relação entre o todo e suas partes, tornando-se uma agregação[4]. Hierarquias (linhas contínuas com setas vazadas) indicam que uma ou mais classes são especializadas por outras. O diagrama da Figura 5.3 mostra as relações estruturais da classe abstrata *ShapeFunctions* (classes abstratas são indicadas em diagramas de classe por fonte itálica), especializada pelas classes SerendiptyTriangular, SerendiptyQuadrangular, SerendiptyTetrahedric e SerendiptyHexahedric. *ShapeFunctions* depende das classes NumericalIntegration e BuiltInArray<double>, do tipo enumerado SFTopology_E e ainda mantém duas relações de agregação com Array<SFValuesStorage> .

[4]Ao gerar código fonte em C++ a partir de um modelo de classes em UML, agregações são mapeadas em atributos de classes. Logo, a partir do código-fonte, não há diferença entre associações e atributos de uma classe. Adotar um ou outro no projeto é uma questão de estilo. É interessante explicitar em associações as relações entre classes com impacto relevante na arquitetura.

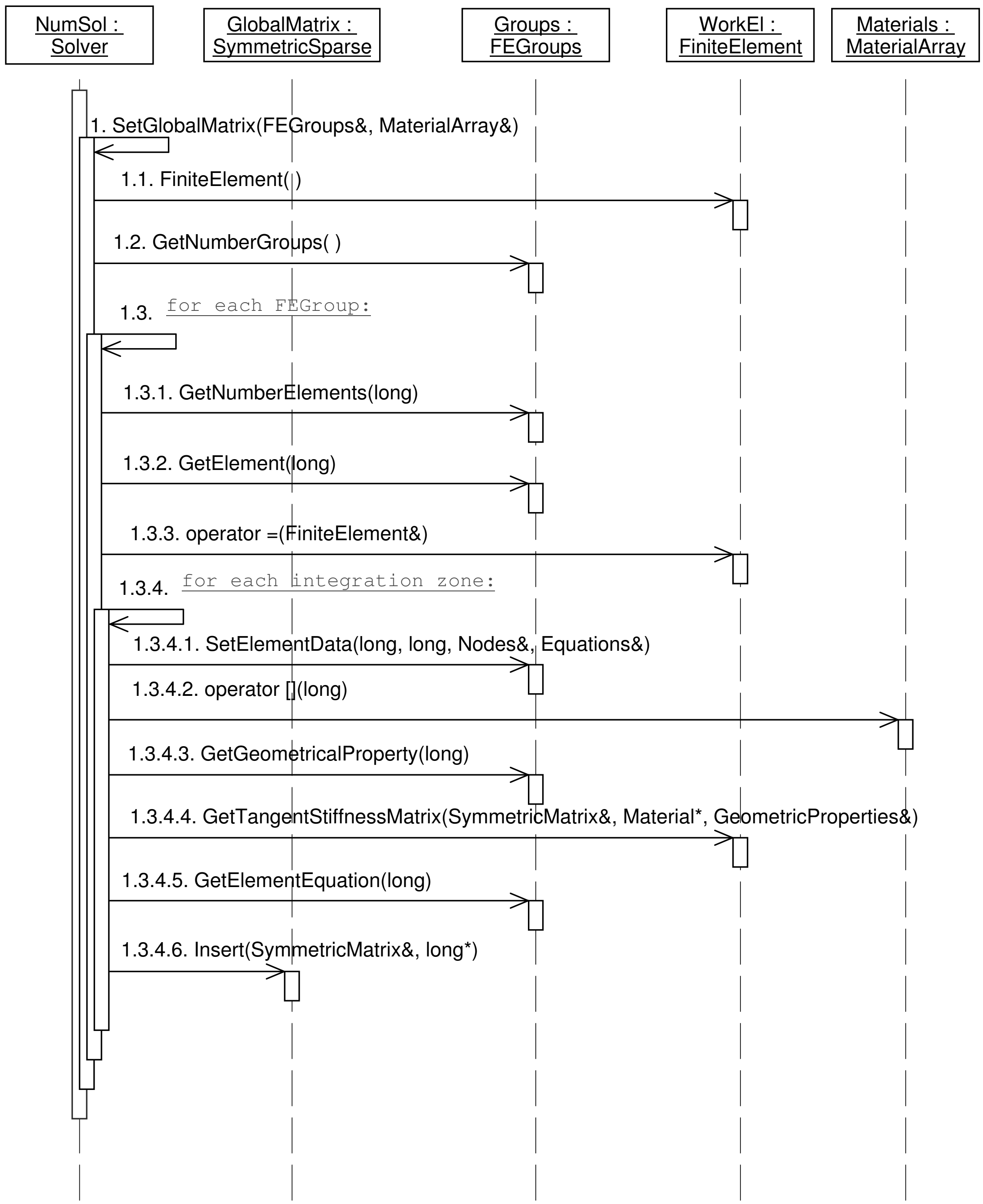


Figura 5.4: Diagrama de seqüência. Construção de matriz de rigidez global.

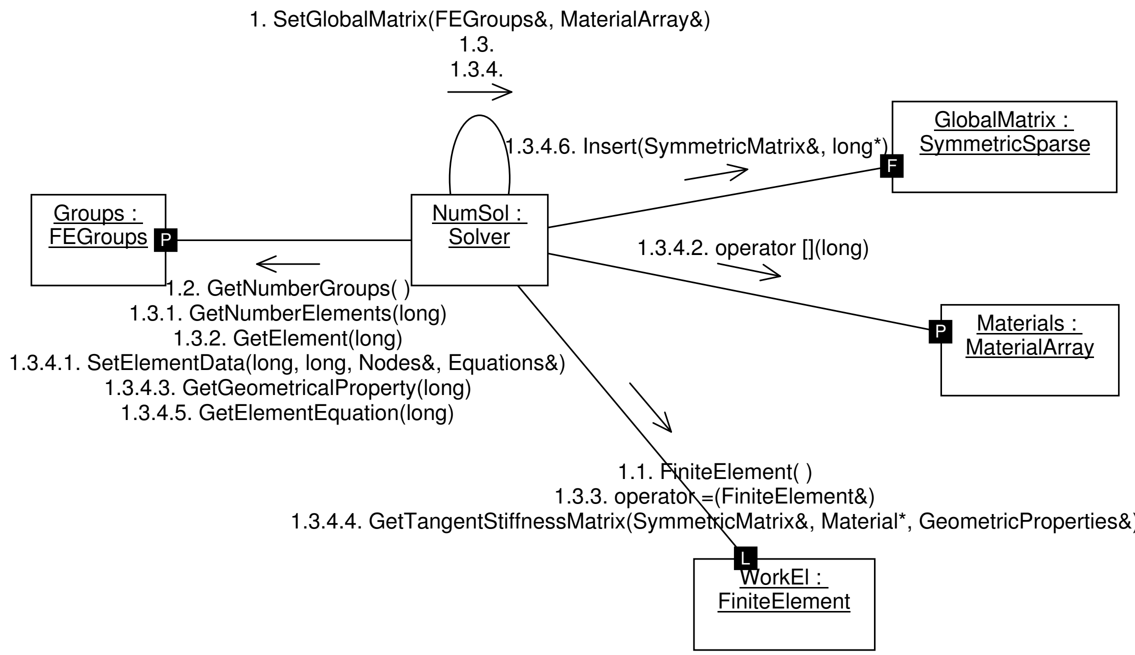


Figura 5.5: Diagrama de colaboração. Construção de matriz de rigidez global.

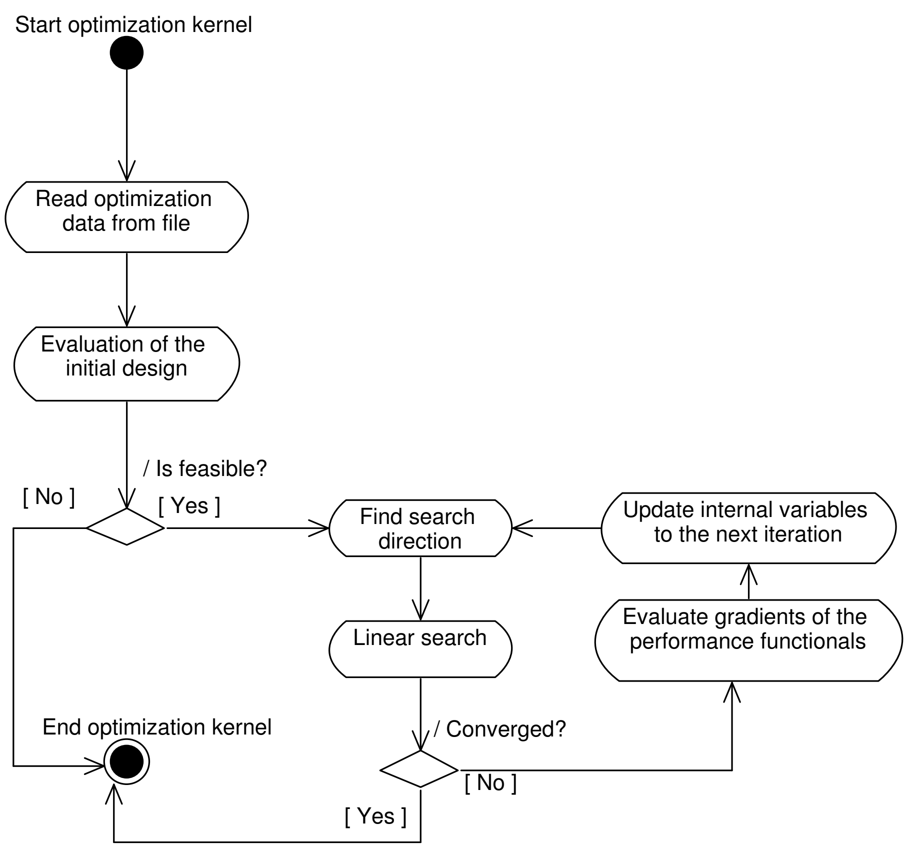


Figura 5.6: Diagrama de atividades. Algoritmo de otimização.

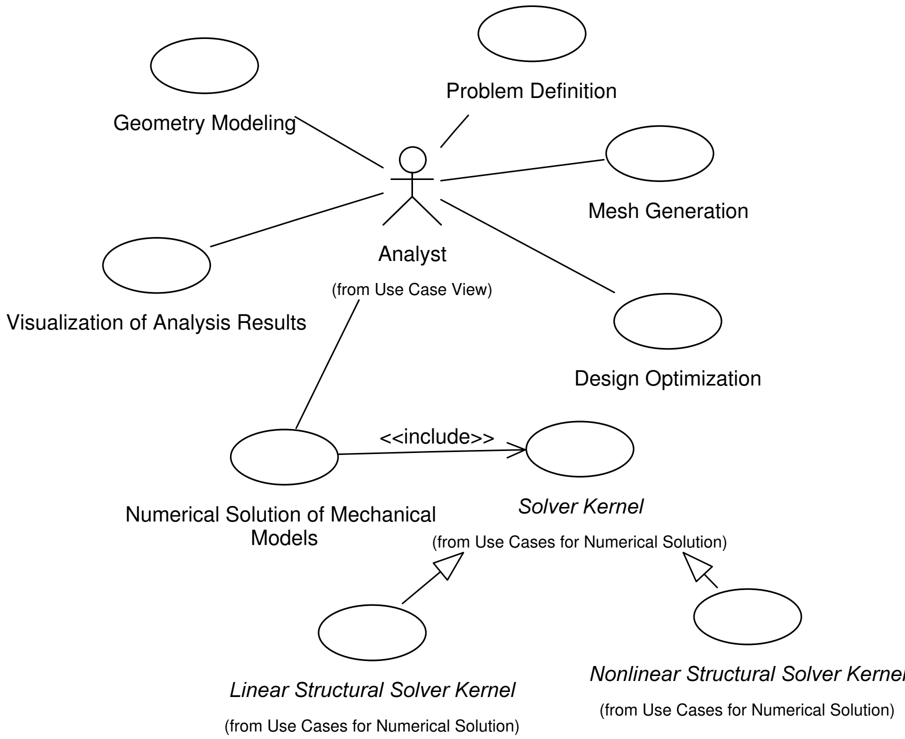


Figura 5.7: Diagrama de casos de uso.

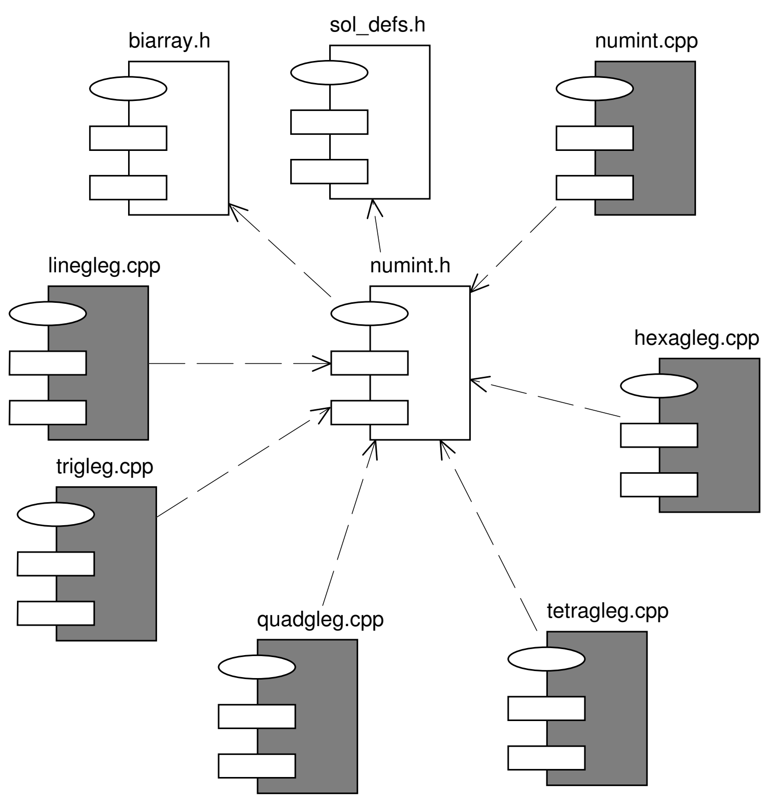


Figura 5.8: Diagrama de componentes.

Diagramas de classes são construídos com diversos níveis de detalhe no decorrer do desenvolvimento. Na fase de análise são puramente conceituais e são aplicados com o objetivo de determinar as entidades do domínio de solução de determinado problema. Durante a fase de projeto podem ser detalhados até o ponto de descreverem exatamente o código a ser gerado, o que é especialmente útil quando são usadas ferramentas de sincronização entre projeto e código.

Na Figura 5.4, tem-se a seqüência de ações envolvidas na construção de uma matriz de rigidez global. Nesse tipo de diagrama, são ordenadas as trocas de mensagens entre os objetos durante a execução de determinada função do sistema. De acordo com a figura, o objeto NumSol (da classe Solver) envia a mensagem SetGlobalMatrix para si mesma, ou seja, faz uma chamada para um método da própria classe. Em seguida, NumSol declara um ponteiro WorkEl para a classe genérica FiniteElement e determina o número de grupos de elementos finitos (em cada grupo, tem-se elementos de mesmo tipo em termos de formato, tipo de interpolação, propriedades de material e propriedades geométricas). Dentro de cada grupo, NumSol determina o respectivo número de elementos enviando a mensagem GetNumberElements para o objeto Groups e acessa o elemento de trabalho do grupo (através das funções GetElement seguida do operador =), o qual é utilizado como interface de acesso a propriedades de cada elemento. Dentro de cada elemento, denominado de zona de integração no diagrama, as características do elemento necessárias para a determinação da matriz de rigidez local (índice de identificação, incidência, numeração de equações, propriedades de material e propriedades geométricas) são transferidas para o elemento indicado por WorkEl pela mensagem SetElementData. A construção da matriz de rigidez é então iniciada pela mensagem GetTangentStiffnessMatrix, enviada para WorEl. Por último, a matriz local é inserida na matriz global (mensagem Insert) utilizando a numeração de equações retornada por GetElementEquation. Observe que as mensagens 1.3 e 1.3.4 indicam laços de iteração.

Na Figura 5.5, tem-se a descrição do mesmo aspecto do sistema de uma perspectiva estrutural, ou seja, quais entidades devem estar ligadas e de que forma. Diagramas de seqüência e colaboração são semanticamente idênticos. A diferença no formato, contudo, auxilia a definição da estrutura estática necessária para suportar o comportamento dinâmico: descrevendo a troca mensagens, primeiro identifica-se onde devem existir relações e, em seguida, quais seus tipos. Na Figura 5.5, NumSol pode enviar mensagens para os objetos Groups e Materials, pois estes são parâmetros (identificador P) de SetGlobalMatrix, operação de NumSol. O objeto WorkEl é uma variável local (identificador L) dessa função enquanto GlobalMatrix é um atributo da classe Solver (identificador F). Assim, todos os objetos estão no mesmo contexto durante a execução da operação e portanto existem conexões para o envio de mensagens. Conexões com parâmetros e variáveis locais são temporárias e dão origem a relações de dependência. Campos significam relações duráveis refletindo-se em associações, agregações ou atributos de classes.

O diagrama de atividades mostrado na Figura 5.6 é usado para descrever comportamentos globais, os quais dariam origem a diagramas de seqüência muito longos e intrincados. Um diagrama de atividade pode ser entendido como um fluxograma no qual cada um dos estados desse diagrama pode ser detalhado por outros diagramas de atividade ou de seqüência. Diagramas de atividade são úteis tanto para descrever casos de uso num formato mais sintético, quanto para modelar e organizar algoritmos complexos. Observando a Figura 5.6, por exemplo, é possível compreender rapidamente o funcionamento global de um algoritmo de otimização, independente de sua complexidade interna.

No Processo Unificado, os três tipos de diagramas anteriores são utilizados para modelar a execução de casos de uso, ou seja, descrevem como a forma e a função do sistema se associam.

O diagrama de casos de uso da Figura 5.7 exemplifica como estruturar e organizar casos de uso. Tais diagramas são úteis para visualizar de forma rápida como os casos de uso estão relacionados entre si e com os usuários, chamados de *atores* na UML. No diagrama, o ator Analyst interage com o sistema através de uma série de casos de uso, entre eles Numerical Solution of Mechanical Models, o qual inclui Solver Kernel. A inclusão funciona como a chamada de um procedimento por outro dentro da programação procedural. Solver Kernel é um use case genérico especializado por Linear Structural Solver Kernel e Nonlinear Structural Solver Kernel. A especialização de casos de uso é semelhante à especialização de classes, ou seja, em todas as situações em que Solver Kernel é válido, qualquer das suas especializações pode ser usada.

Um componente é uma entidade física de um sistema de software. Diagramas de componentes possibilitam modelar a parte física do sistema ao contrário de todos os diagramas anteriores, usados para modelagem conceitual. Em outras palavras, tais diagramas indicam como se organizam códigos-fonte, executáveis, bibliotecas, tabelas de bancos de dados, páginas para WWW, etc. Ferramentas de geração automática de código usam diagramas de componentes para conectar classes e código-fonte e organizar a estrutura de diretórios na qual o código deve ser gerado, bem como converter relações entre classes em inclusões de arquivos de código-fonte. Na Figura 5.8, o arquivo `numint.h`, que depende dos arquivos `biarray.h` e `sol_defs.h`, contém as declarações de classes e funções implementadas em `numint.cpp`, `trigleg.cpp`, `quadleg.cpp`, `tetraleg.cpp` e `hexaleg.cpp`.

Há ainda recursos para modelagem da implantação física do sistema, ou seja, para a especificação das unidades físicas (hardware) nas quais os componentes de software serão executados. Tais modelos são especialmente importantes em sistemas distribuídos e arquiteturas cliente-servidor.

A UML, porém, é apenas uma linguagem de modelagem. Assim, a UML fornece um conjunto de ferramentas, mas não informa como e com que finalidade tais ferramentas devem ser aplicadas. É necessário escolher quais e quando utilizar os diversos diagramas disponíveis e qual o nível de detalhe a ser adotado nos modelos, manter a conexão e a sincronização dos modelos com o código produzido e modelar o sistema a partir de várias perspectivas para compreendê-lo em sua totalidade. A adoção de um processo de desenvolvimento de software permite escolher os diagramas e o momento de aplicá-los, bem como as perspectivas nas quais os mesmos devem ser produzidos.

5.3 Considerações sobre as Técnicas de Engenharia de Software

A aplicação sistemática das técnicas discutidas acima, juntamente com ferramentas adequadas de gerenciamento e sincronização de documentos, diagramas e código, tem certamente contribuído para trazer mais controle, produtividade e qualidade ao desenvolvimento de software. Por outro lado, mesmo organizações experientes no desenvolvimento de sistemas extensos têm encontrado sérias dificuldades em suas atividades, freqüentemente estourando prazos e orçamentos, e também cancelando projetos[5].

A conclusão óbvia é que tais procedimentos não são garantias infalíveis de sucesso, sendo necessário considerar que a implantação e o aprendizado das técnicas podem ser fontes de algum risco, custo e consumo de tempo. Sua correta aplicação, porém, torna a imprevisibilidade razoavelmente administrável, possibilitando lidar de forma mais racional com as dificuldades inerentes à definição e organização dessa classe de sistemas. Outra contribuição importante para a melhoria dessa atividade seria entender a origem de tais dificuldades.

Se comparada ao projeto e construção de sistemas físicos, observa-se que a engenharia de software é fortemente baseada em experiência, em tentativa e erro (Kruchten, 2001a; Buhrer, 2000), enquanto a evolução das engenharias de sistemas físicos é caracterizada pela aplicação cada vez mais sistemática de princípios físicos (Timoshenko, 1953). Com recursos de análise de sensibilidade e otimização, por

[5]Apenas para citar dois exemplos, Windows 2000 e Mac OS X foram lançados com mais de um ano de atraso e com menos funcionalidades do que inicialmente proposto.

exemplo, busca-se reduzir a necessidade de experiência prévia em várias situações de projeto de sistemas físicos.

Princípios físicos agem como restrições e objetivos a serem atingidos durante o projeto. Em última análise permitem a definição de regras de projeto que são válidas em quaisquer casos, respeitando as premissas dessas regras. Em contrapartida, não há princípios físicos envolvidos na especificação, organização e construção de software. Os procedimentos discutidos nas seções anteriores são as melhores técnicas extraídas da experiência prática através da observação do fracasso e sucesso de grande número de projetos. Por esse motivo, em qualquer processo de engenharia de software é crucial alocar desenvolvedores experientes para atividades críticas como, por exemplo, a definição de arquiteturas. Julgar *a priori* se uma arquitetura de software está modular ou resiliente o suficiente para suportar os requisitos envolve bem mais que habilidade técnica: requer experiência, talento e sensibilidade. Uma resposta a isso seria o desenvolvimento iterativo, no qual propostas de arquitetura são produzidas e testadas[6]. Julgar se o modelo de um sistema físico cumpre seus requisistos requer principalmente habilidade técnica na aplicação dos princípios físicos e solução das equações resultantes, e adicionalmente graus variáveis de experiência e talento conforme o caso.

Outra fonte de dificuldades são as expectativas por vezes errôneas surgidas da aplicação imprecisa dos termos "projeto" e "construção" no contexto da engenharia de software (Buhrer, 2000). No caso de sistemas físicos, a conclusão do projeto caracteriza-se por um desenho final de produto que descreve em detalhes materiais, subconjuntos, uso de componentes de terceiros, partes a serem produzidas, tolerâncias, seqüências de montagem, etc. A partir dessas informações é possível determinar com razoável precisão custos e tempos de construção (produção). Logo, a etapa de projeto fornece informação suficiente para se obter o sistema final dentro de prazos e custos conhecidos.

No caso de software, finalizado o "projeto", tem-se apenas um conjunto estruturado de conceitos, ou seja, entidades, interfaces e relações que compõem o sistema. Seguindo tais informações não é possível obter o produto final, ou seja, um sistema executável, pois o código fonte ainda não foi totalmente escrito. Isso se dá somente após a fase de "construção".

No sentido estrito de se obter um produto final seguindo instruções de um plano detalhado, é o compilador que de fato realiza a construção do sistema de software, a custo e tempo desprezíveis nesse caso. No sentido de fornecer um plano detalhado para a construção do sistema, apenas ao final da

[6] Mas também aí são necessárias habilidades não-técnicas para selecionar os requisitos de maior influência arquitetônica.

fase de "construção" o verdadeiro projeto do sistema de software está concluído. Em resumo, o que se convencionou chamar de projeto de software corresponde a um esboço de projeto de um sistema físico.

Em qualquer disciplina, o projeto de um novo produto envolve imprevisibilidade, pois nessa etapa é necessário considerar muito mais que seqüências de execução e montagem, e a interação entre elementos e subconjuntos adjacentes. É necessário especificar, selecionar e combinar componentes, analisar interações entre eles (o número de interações que cresce não-linearmente com o número de componentes), verificar incompatibilidades e performance.

Desconsiderar as diferenças entre sistemas físicos e software pode levar a conclusões erradas sobre técnicas de engenharia de software. Inicialmente, por exemplo, acreditava-se que a programação orientada por objetos, devido às características de abstração e encapsulamento, traria para o desenvolvimento de software a mesma produtividade e evolução rápida que tem caracterizado a indústria de componentes eletrônicos (Cox, 1986), o que não se confirmou após a difusão dessa técnica de programação. Além disso, o desenvolvimento de sistemas de software envolve várias outras etapas além da implementação, cada qual com impacto na produtividade e na qualidade.

A sistematização da reutilização de soluções de projeto e arquitetura de sistemas de software é uma estratégia adicional às já apresentadas para aliviar as incertezas próprias da fase de projeto e a falta de leis gerais de desenvolvimento. Soluções comprovadas de projeto são encontradas na forma de gabaritos genéricos denominados *padrões* (patterns). Vários padrões de projeto, análise e arquitetura são encontrados na forma de catálogos apresentando motivação, aplicabilidade e diagramas da estrutura de cada solução, juntamente com exemplos de códigos fonte que os implementam (Gamma *et al.*, 1995; Larman, 1998; Larman, 2001). Construir arquiteturas baseadas em padrões conhecidos e comprovados é uma forma razoável de reduzir a dependência em termos de experiência e ao mesmo tempo adquiri-la e difundi-la entre os desenvolvedores. Padrões também surgem das soluções encontradas pelos desenvolvedores para problemas recorrentes, fazendo parte da definição de uma arquitetura o reconhecimento de padrões dentro da base de software da organização.

À primeira vista, a aplicação das técnicas de desenvolvimento de software descritas acima parece desviar parte considerável do tempo anteriormente dedicado à codificação dos algoritmos para outras atividades. Apesar de ser verdadeira a afirmação de que a utilização sistemática de um processo de desenvolvimento torna a etapa de codificação mais objetiva e eficiente é inegável a introdução de novas fases de trabalho. Nesse sentido é recomendada a adoção de ferramentas de apoio ao desenvolvimento de software para modelagem gráfica, gerenciamento de requisitos, geração automática de código e sincronização, bem como sistema para controle de versões de arquivos.

5.4 Arquitetura para Análise de Resposta, Análise de Sensibilidade e Otimização

A descrição de uma arquitetura de um sistema de software em certo momento do processo de desenvolvimento é composta por versões dos documentos e diagramas produzidos até ali mostrando os aspectos mais significativos do sistema. Busca-se aqui um ambiente para modelagem geométrica, geração de malhas, análise de resposta, análise de sensibilidade, otimização de problemas de mecânica do contínuo e visualização de resultados. Inicialmente, a ênfase do sistema será o tratamento de problemas estruturais lineares e não-lineares.

A descrição da arquitetura é feita em quatro partes: modelo de casos de uso, organização geral do sistema de software, modelo de projeto e modelo de componentes. O primeiro objetivo desta descrição é mostrar a organização geral do sistema, suas divisões principais e dependências. São mostradas em detalhe as estruturas que dão suporte aos procedimentos numéricos de análise de resposta, análise de sensibilidade e otimização.

O código produzido incorpora os principais conceitos descritos na literatura de implementação de elementos finitos com técnicas de orientação por objetos: estrutura genérica de funções de forma, regras de integração numérica, elementos (sub, super e isoparamétricos), materiais; núcleo de solução independente da dimensão do domínio; programação de baixo nível com uso intensivo de ponteiros; armazenamento esparso; métodos diretos e iterativos de solução de sistemas lineares; análise de sensibilidade integrada à análise de resposta de problemas lineares e não-lineares.

5.4.1 Visão Arquitetônica do Modelo de Use Cases

O diagrama da Figura 5.9 mostra os principais casos de uso de um ambiente de simulação, com os quais interage um ator denominado Analyst. Esses casos de uso são usuais nesse tipo de sistema, não sendo necessário aqui detalhar a seqüência de ações de cada um deles. De forma resumida, cada use case está relacionado aos seguintes requisitos:

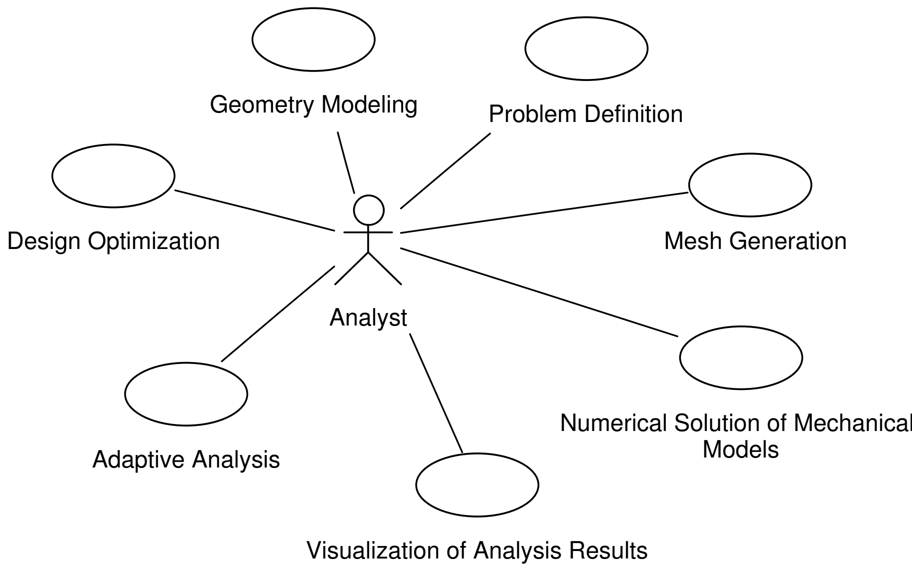


Figura 5.9: Use cases principais de ambiente de simulação de problemas mecânicos.

- Geometry Modeling: definição de geometrias bi e tridimensionais, importação e correção de geometrias em formatos IGES, SAT, CATIA, PRO-ENGINEER e STEP.

- Mesh Generation: geração de malhas lineares e quadráticas de triângulos, quadrados (em regiões bidimensionais e superfícies), tetraedros e hexaedros.

- Problem Definition: especificação de condições de contorno, carregamentos concentrados e distribuídos, forças de volume, materiais, cinemática de deformação, dependência do tempo, etc.

- Numerical Solution of Mechanical Models: solução de problemas estruturais lineares e não-lineares planos e tridimensionais envolvendo elasticidade linear, hiperelasticidade não-linear, plasticidade, contato, efeitos transientes, dano, placas e cascas; formulações h e p de elementos finitos; métodos diretos, iterativos e multigrid para a solução de sistemas de equações.

- Adaptive Analysis: estimação de erros e cálculo de parâmetros de refinamento de malha em problemas lineares elípticos.

- Design Optimization: cálculo de gradientes por análise de sensibilidade, otimização de forma, parâmetros distribuídos e topologia.

- Visualization of Analysis Results: visualização gráfica de campos de resposta de problema, tais como deslocamentos e tensões.

A consideração de todos esses casos de uso é fundamental para a organização geral do sistema. Este trabalho, entretanto, concentra-se do desenvolvimento do código correspondente aos casos de uso Numerical Solution of Mechanical Models e Design Optimization, os quais devem ser detalhados para permitir o projeto do núcleo de solução numérica.

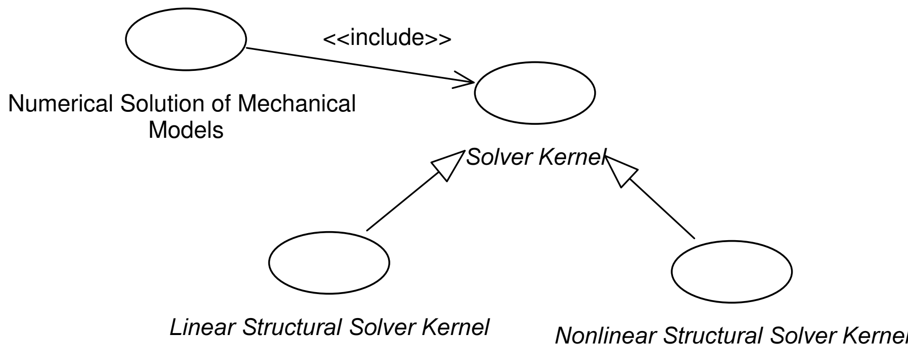


Figura 5.10: Casos de uso gerais do núcleo numérico de análise de resposta.

A Figura 5.10 mostra a relação entre Numerical Solution of Mechanical Models e seu núcleo numérico *Solver Kernel*, que é um use case genérico especializado por *Linear Structural Solver Kernel* e *Nonlinear Structural Solver Kernel*.

Solver Kernel consiste dos seguintes passos:

1. Leitura de arquivo de dados definindo o problema e diretivas de solução (tais como métodos de solução de sistemas lineares e não-lineares, precisões e número máximo de iterações).
2. Construção e solução de sistemas lineares de acordo com o tipo de problema e diretivas de solução.
3. Cálculo de campos de resposta a partir das soluções dos sistemas lineares.
4. Gravação de campos de resposta em banco de dados binário para posterior visualização.

As especializações *Linear Structural Solver Kernel* e *Nonlinear Structural Solver Kernel* desse caso de uso detalham justamente o passo 2.

Os casos de uso do núcleo numérico de análise de sensibilidade e otimização, e sua relação com os casos de uso de análise de resposta, são mostrados na Figura 5.11. Nesse diagrama, observa-se que *Generic Optimizer* é especializado por algoritmos de minimização específicos como é o caso de *HIPM*, que representa o fluxo de ações do método de pontos interiores de Herskovits (Herskovits, 1986; Panier *et al.*, 1988; Herskovits e Coelho, 1989; Evsukoff, 1992). Já o caso de uso *Velocity Field* é especializado por *Boundary Layer* e *Boundary Displacement*, os quais descrevem, respectivamente, fluxos de eventos de cálculo de campos de velocidade em camada de contorno e por deslocamento fictício de nós do contorno. Observa-se também que, enquanto *Design Analysis* depende do tipo de problema e portanto de *Solver Kernel*, o fluxo de eventos de análise de sensibilidade depende apenas de *Linear Structural Solver Kernel*, pois a análise de sensibilidade consiste num procedimento linear, independentemente do tipo de problema.

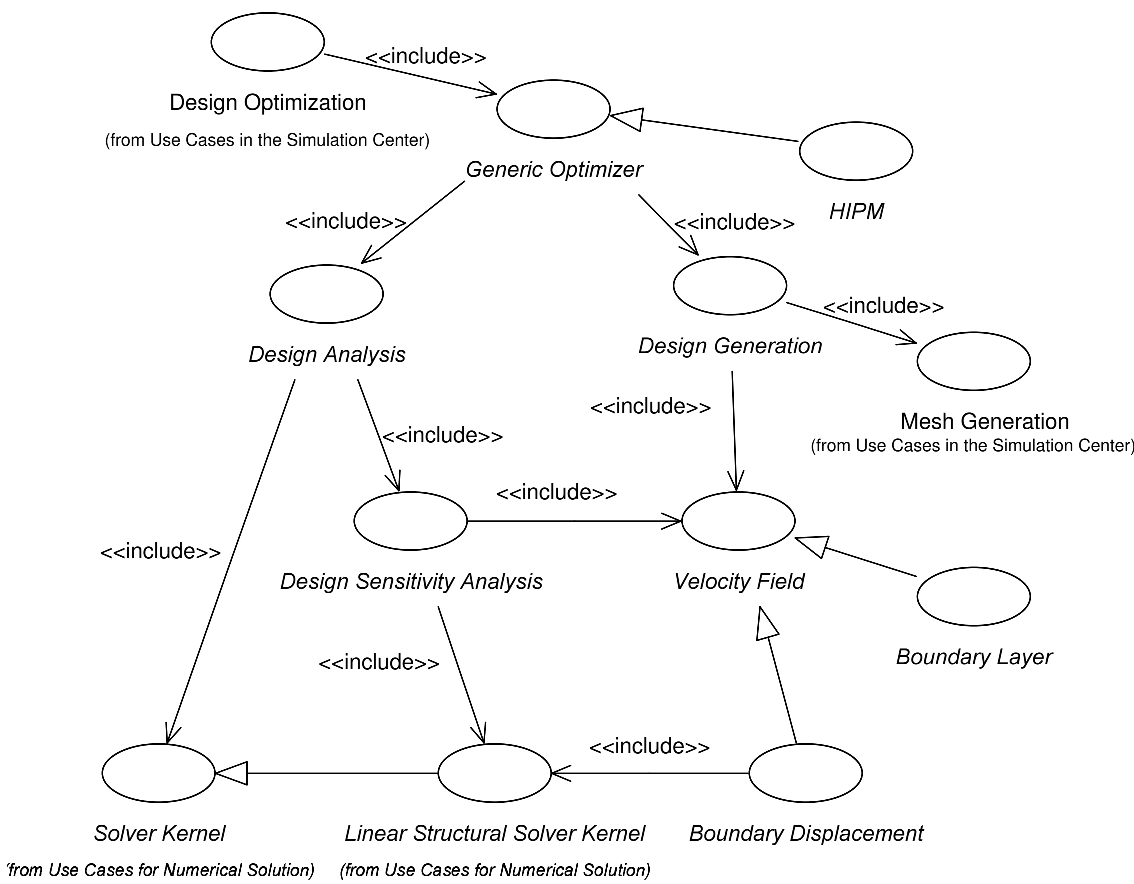


Figura 5.11: Casos de uso do núcleo numérico de análise de sensibilidade e otimização.

Pode-se usar diagramas de atividades para descrever casos de uso de modo esquemático. A Figura 5.6, por exemplo, descreve o fluxo de eventos do use case *Generic Optimizer* (na Figura 5.12, detalha-se a atividade Linear Search desse diagrama). Na Figura 5.13, tem-se o fluxo de eventos do use case *Design Analysis*.

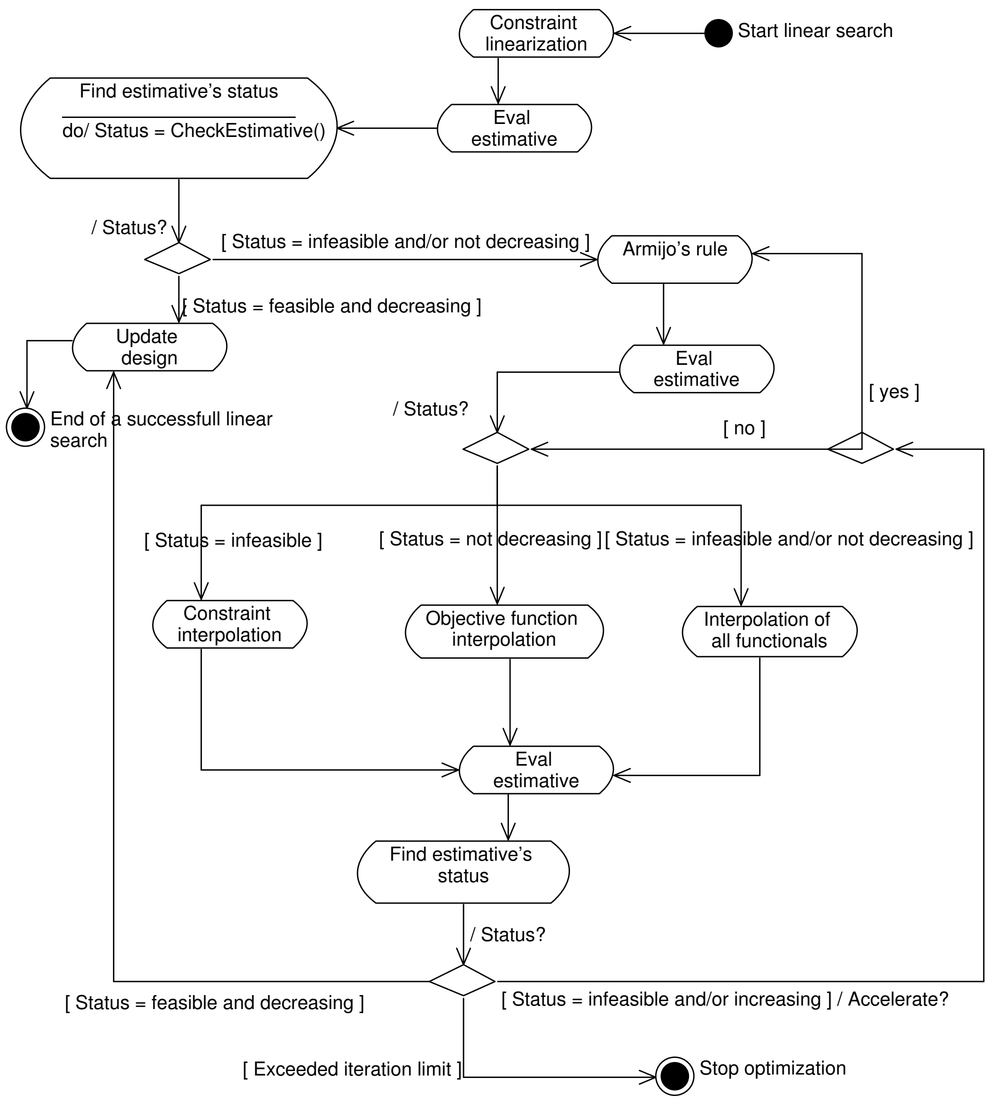


Figura 5.12: Diagrama de atividades de busca linear.

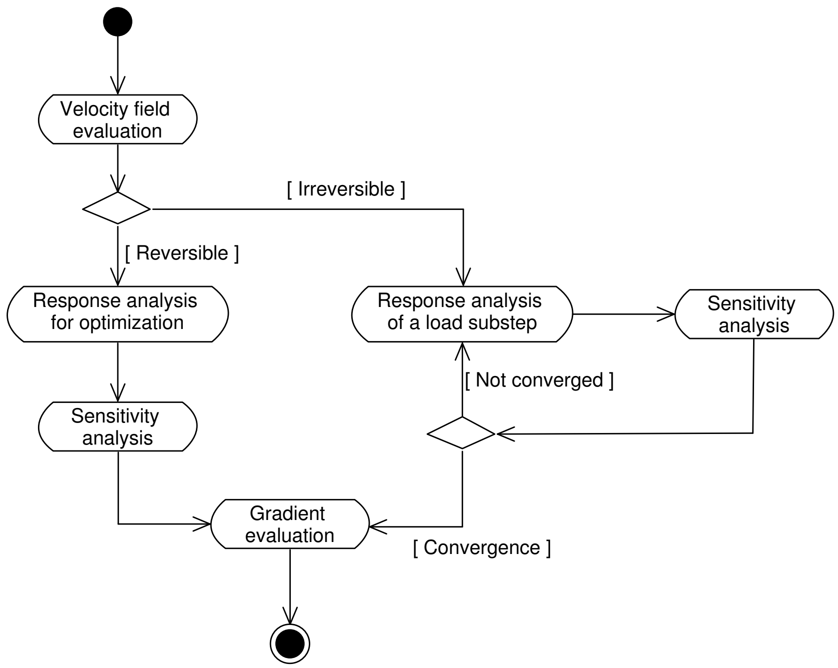


Figura 5.13: Fluxo de eventos do use case *Design Analysis*.

5.4.2 Visão Arquitetônica da Organização Geral do Sistema

O primeiro passo para a definição da estrutura do sistema consiste na sua organização geral em termos de pacotes e subsistemas. Para isso é necessário considerar os casos de uso mais significativos do sistema e outros requisitos não funcionais, tais como a base de código já existente e os sistemas operacionais sobre os quais o sistema funcionará.

Os casos de uso usados para a definição do sistema em consideração foram descritos na seção anterior. A base de software inicial (Bittencourt e Feijóo, 1996; Fancello *et al.*, 1991; Fancello e Feijóo, 1992; Guimarães e Feijóo, 1989; Silva e Bittencourt, 1997) apresenta a seguinte composição:

- **ACDP:** biblioteca de funcionalidades básicas, tais como tratamento e notificação de erros e exceções, leitura de dados de arquivos ASCII, manipulação de bancos de dados binários e rotinas matemáticas utilizadas por todos os programas.
- **DS:** biblioteca de estruturas de dados básicas (matrizes, vetor, arrays, string) e procedimentos numéricos de manipulação associados, tais como produto interno, soma, multiplicação por escalar e métodos de solução de sistemas lineares.
- **SOLVER:** conjunto de classes para a versão h do método de elementos finitos aplicados a pro-

blemas estruturais lineares bi e tridimensionais. Implementação baseada na generalização de elementos, funções de forma, regras de integração e materiais. Desse modo, novos tipos específicos podem ser incluídos desde que respeitem a estrutura original estabelecida para cada uma das entidades genéricas.

- **OPT:** conjunto de classes para análise de sensibilidade e otimização de problemas estruturais lineares. Sensibilidade a variação de parâmetros discretos e de forma do domínio. Implementação baseada na generalização de variáveis de projeto, funcionais de performance e campos de velocidades.

- **SAT 2D:** aplicativos para geração de geometrias baseado em NURBS, importação de arquivos em formato DXF, geração de malhas, definição de parâmetros do problema e visualização de resultados em domínios bidimensionais.

- **ADAPT:** estimador de erros Zienkiewicz-Zhu (Zienkiewicz e Zhu, 1987) e refinamento de malhas em problemas estruturais lineares bidimensionais em conjunto com o gerador de malhas FRONTAL.

- **Geradores de malha** (adquiridos de terceiros)**:** FRONTAL (malhas bidimensionais de triângulos e refinamento adaptável) e GID (malhas bi e tridimensionais de triângulos, quadrados tetraedros e hexaedros).

As bibliotecas e programas desenvolvidos foram implementados com técnicas de programação orientada por objetos em C++. Essa base de software já existente foi usada como ponto de partida para o desenvolvimento do novo sistema, sendo os recursos do Processo Unificado usados para orientar essa transição (Kruchten, 2001b). Técnica de engenharia reversa, por exemplo, foi utilizada para mapear o código já existente em diagramas em UML.

No novo sistema, as bibliotecas ACDP e DS foram mantidas inalteradas, sendo reproduzidas nos pacotes ACDP95 e Ds respectivamente. Em termos de organização, as bibliotecas SOLVER, OPT e ADAPT também foram reproduzidas, porém com conteúdo reprojetado para cumprir os novos requisitos. Devido aos novos requisitos introduzidos pelos casos de uso da Figura 5.9, o SAT 2D foi substituído por um conjunto de aplicativos baseados no núcleo gráfico ACIS (Spatial Corp., 2001; Corney, 1997), que já incorpora ferramentas de importação/exportação de arquivos em vários formatos e procedimentos de verificação e correção de modelos de CAD (Driemeier, 2002). O novo sistema deve ainda rodar sobre plataformas Windows e GNU/Linux.

Para organização global do sistema foi utilizado o padrão arquitetônico de camadas[7] baseado em reutilização (Eeles, 2001; Jacobson *et al.*, 1999). Nesse padrão, os pacotes e subsistemas são localizados em camadas de acordo com sua generalidade, sendo as camadas inferiores reservadas à partes menos específicas do sistema e, portanto, com interfaces mais estáveis. O padrão arquitetônico de camadas é indicado para sistemas muito complexos para serem entendidos em sua totalidade, exigindo decomposição. Dois padrões arquitetônicos usuais são a decomposição baseada em responsabilidade e a decomposição baseada em reutilização.

Na decomposição em camadas baseada em responsabilidade, as classes são organizadas de acordo com sua função na lógica do sistema, facilitando o desenvolvimento do sistema e sua manutenção. Essa estratégia é bastante usada no projeto de sistemas distribuídos, tais como aplicações para o ambiente cliente-servidor e para web. Um exemplo seria a divisão em três camadas: *Apresentação*, *Domínio de Atividade* e *Acesso a Dados*. Na primeira camada, tem-se os componentes das interfaces de comunicação com o usuário; na segunda, tem-se os componentes específicos do domínio do problema; na última camada, se localizam os elementos de acesso às informações persistentes do sistema.

A arquitetura de camadas baseada em reutilização é aplicada nas situações nas quais a reutilização de componentes é um dos objetivos principais dos desenvolvedores. Os componentes são então agrupados de acordo com sua generalidade, evidenciando seu potencial de reutilização. A intenção desse padrão é que os níveis superiores, os quais mudam mais freqüentemente, sejam construídos sobre uma base estável formada pelos níveis inferiores. Para que isso ocorra, conclui-se que as dependências devem ocorrer apenas no mesmo nível e em relação aos níveis inferiores.

Um aspecto bastante positivo desse padrão arquitetônico é que os diversos tipos de decomposição em camadas não são excludentes e podem ser aplicados simultaneamente ao mesmo sistema. Nesse caso, cada tipo de decomposição é perpendicular às demais.

O sistema desenvolvido foi dividido em quatro camadas:

- System Software: sistemas operacionais e interfaces com equipamentos.
- Middleware: pacotes e subsistemas prontos adquiridos de terceiros para aplicações genéricas tais

[7] Patterns layer.

como sistemas de bancos de dados e linguagens de programação.

- Business Specific: pacotes e subsistemas reutilizáveis específicos do domínio de atuação da organização. São desenvolvidos pela própria organização e são utilizados por vários subsistemas do próximo nível.

- Application Subsystems: conjunto de subsistemas dedicados que compõem um aplicativo ou aplicativos visíveis pelo usuário final.

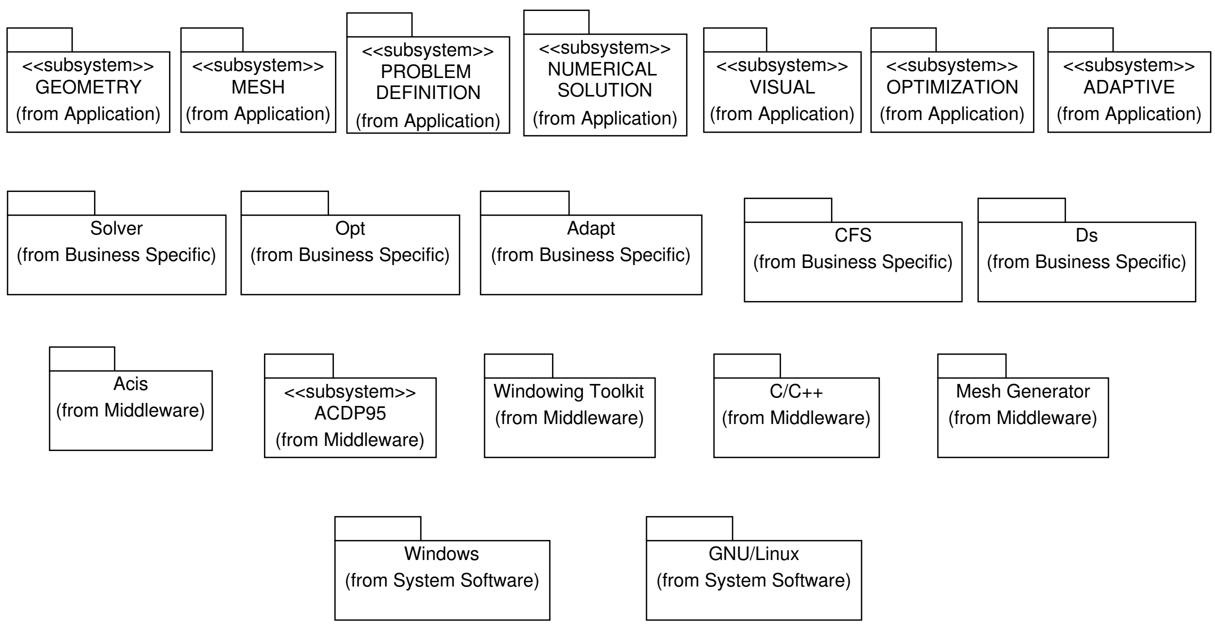


Figura 5.14: Organização geral do sistema em pacotes, subsistemas e padrão de camadas.

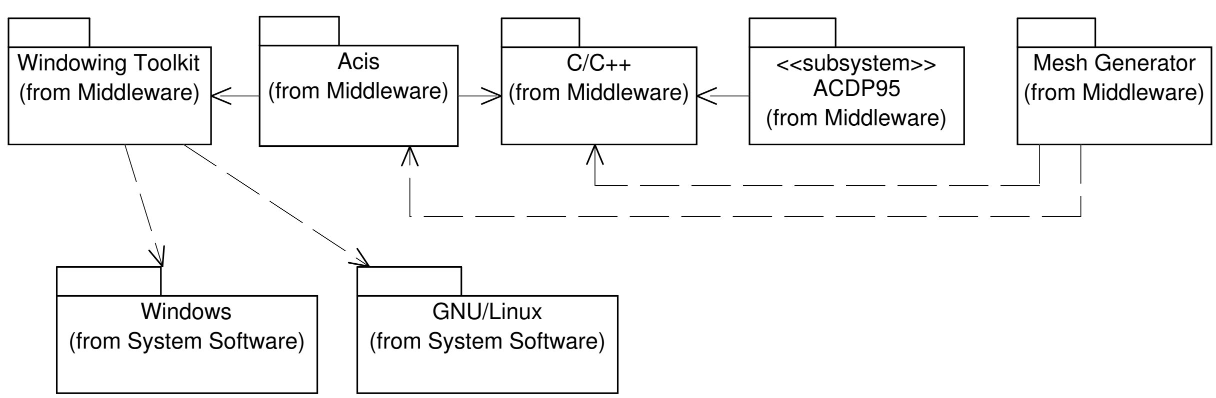


Figura 5.15: Camadas Middleware e System Software.

A organização geral do sistema é mostrado na Figura 5.14. Como o nível mais alto é composto apenas por subsistemas, pode-se ter ciclos de desenvolvimento independentes para cada um deles.

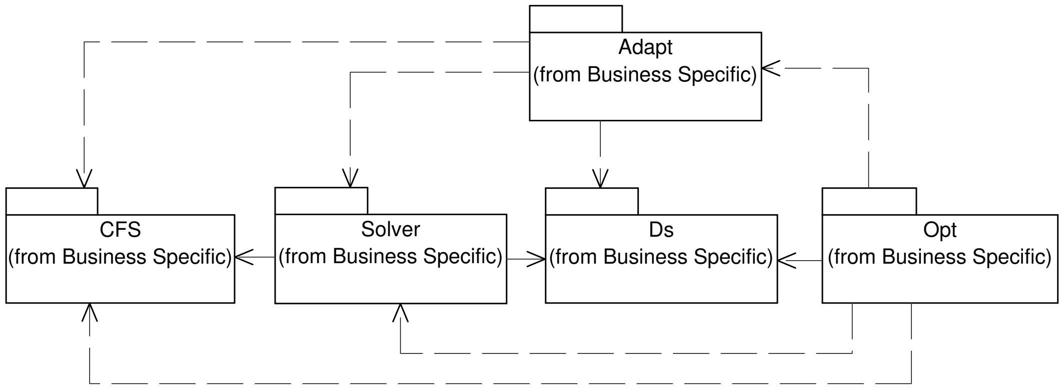


Figura 5.16: Camada Business Specific.

Na Figura 5.15, têm-se as dependências entre pacotes e subsistemas dos dois níveis mais baixos. Observa-se que apenas Windowing Toolkit depende do sistema operacional, pois são usados esquemas diferentes de construção de janelas em cada um. Por exemplo, os sistemas de janelas do Windows, tais como o MFC, não são compatíveis com o XWindow do GNU/Linux. Os outros pacotes têm versões com as mesmas interfaces nos dois sistemas, tornando as dependências de sistema razoavelmente irrelevantes. Uma outra opção seria a substituição do Windowing Toolkit por um sistema gráfico com interfaces de programação comuns a vários sistemas, como o Qt (Troll Tech AS., 2001; Dalheimer, 1999; Ferreira, 2002), tornando a arquitetura independente do sistema operacional.

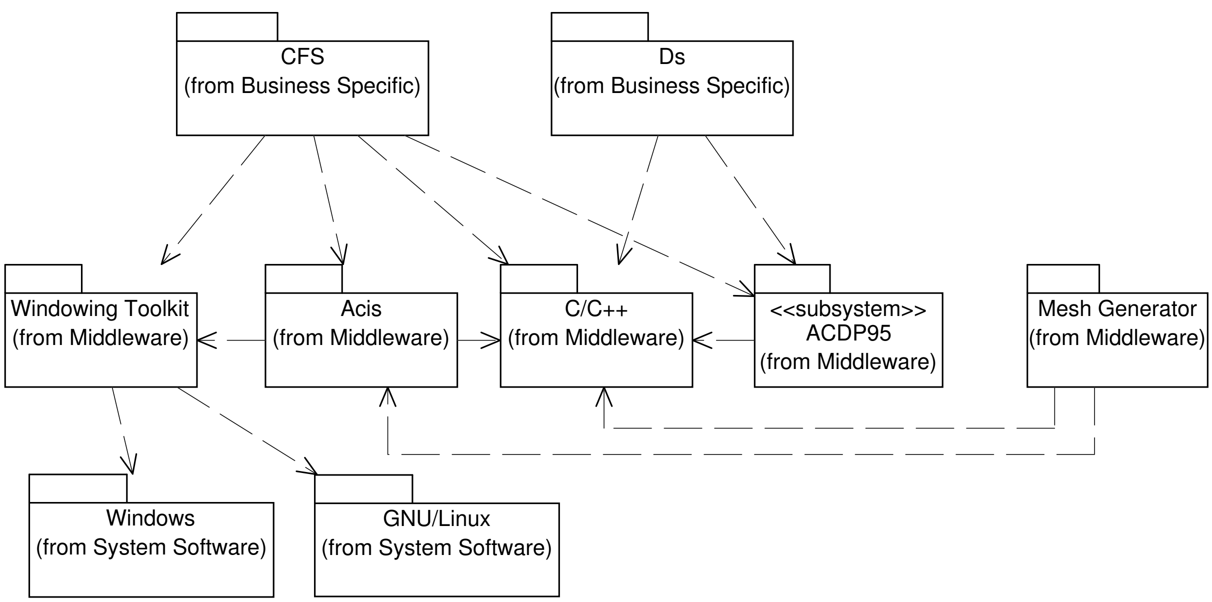


Figura 5.17: Dependências entre Business Specific e Middleware.

Na Figura 5.16, tem-se as dependências internas da camada Business Specific. O pacote CFS contém interfaces para acesso a modelos de simulação armazenados em banco de dados ACDP. O pacote

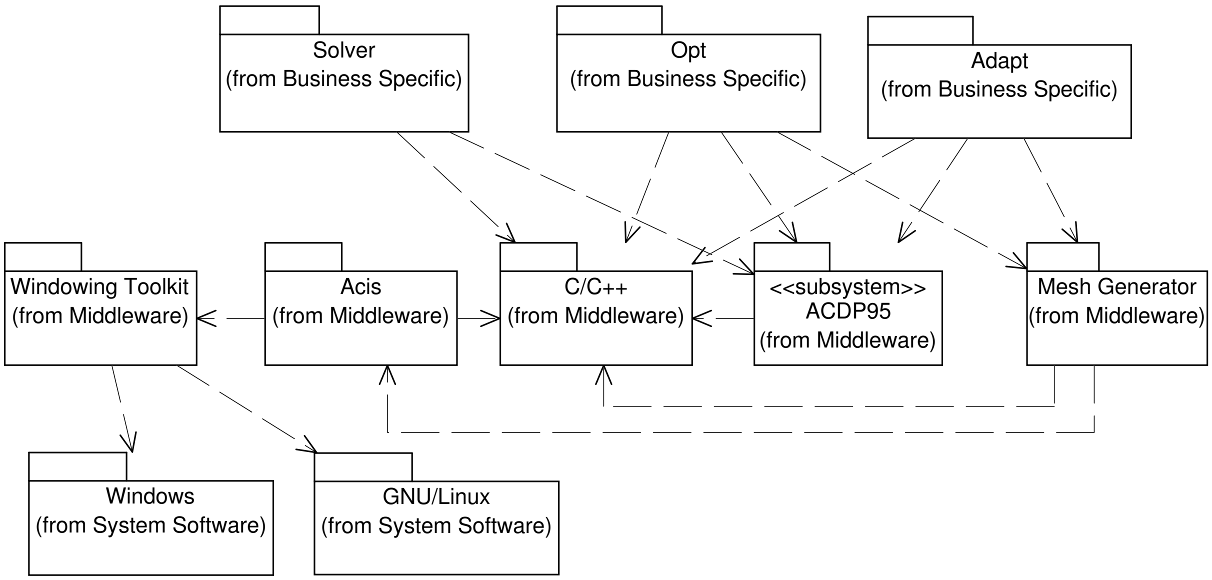


Figura 5.18: Dependências entre Business Specific e Middleware.

Ds é usado como base para construção dos demais pacotes dessa camada, centralizando a alocação e desalocação de memória de vários tipos de estruturas de dados. Isso facilita a manutenção e detecção de erros, pois o gerenciamento de memória é fonte usual de problemas quando se usa C/C++. Os pacotes Ds, Solver, Opt e Adapt formam o núcleo numérico do ambiente, não incorporando interfaces gráficas, como se observa nas dependências dessa camada em relação a Middleware (Figuras 5.17 e 5.18).

As dependências da camada Application Subsystems em relação a Middleware e Business Specific é mostrada nas Figuras 5.19 a 5.22.

5.4.3 Visão Arquitetônica do Modelo de Projeto

O passo seguinte à organização global do sistema no padrão de camadas é o projeto interno dos pacotes e subsistemas de Business Specific e Application Subsystems. Tem-se a seguir a visão arquitetônica do projeto dos pacotes Ds, Solver e Opt.

Pacote Ds

O pacote Ds reproduz a biblioteca **Ds** na modelagem do novo sistema. É interessante observar no diagrama da Figura 5.23 a ausência de associações entre as classes desse pacote. As classes ou estão isoladas ou mantém apenas dependências, que são relacionamentos de curta duração nos quais

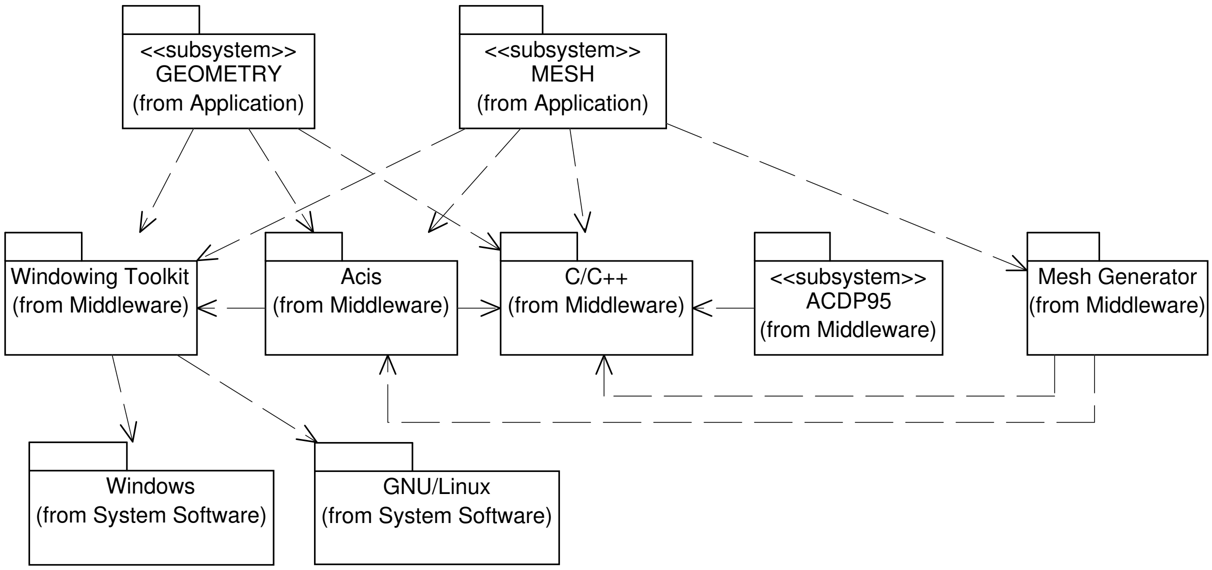


Figura 5.19: Dependências entre Application Subsystems e Middleware.

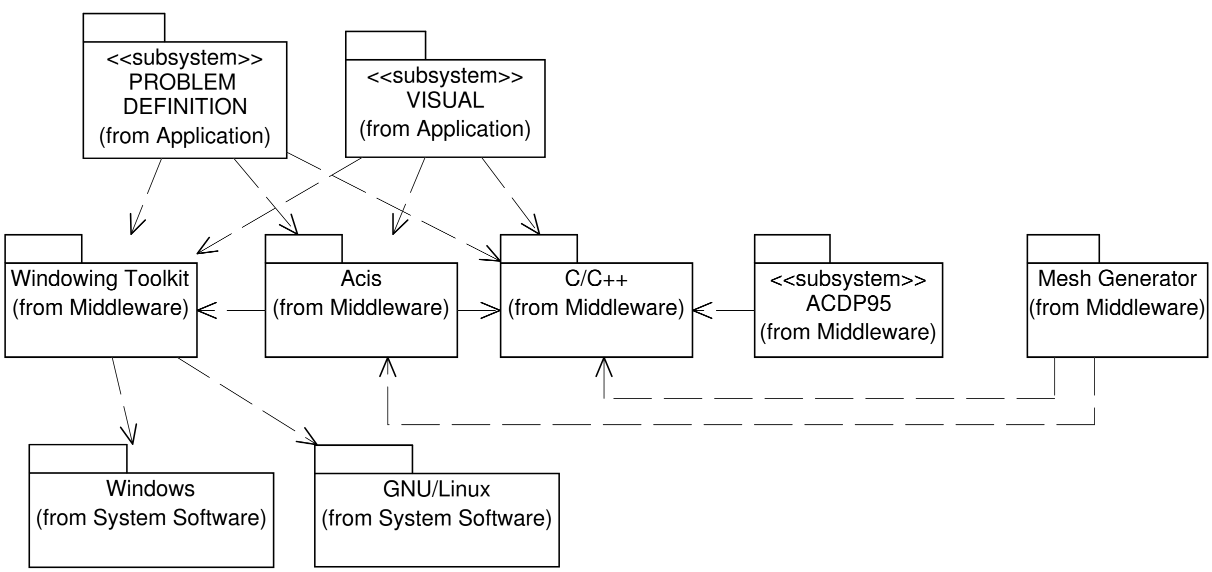


Figura 5.20: Dependências entre Application Subsystems e Middleware.

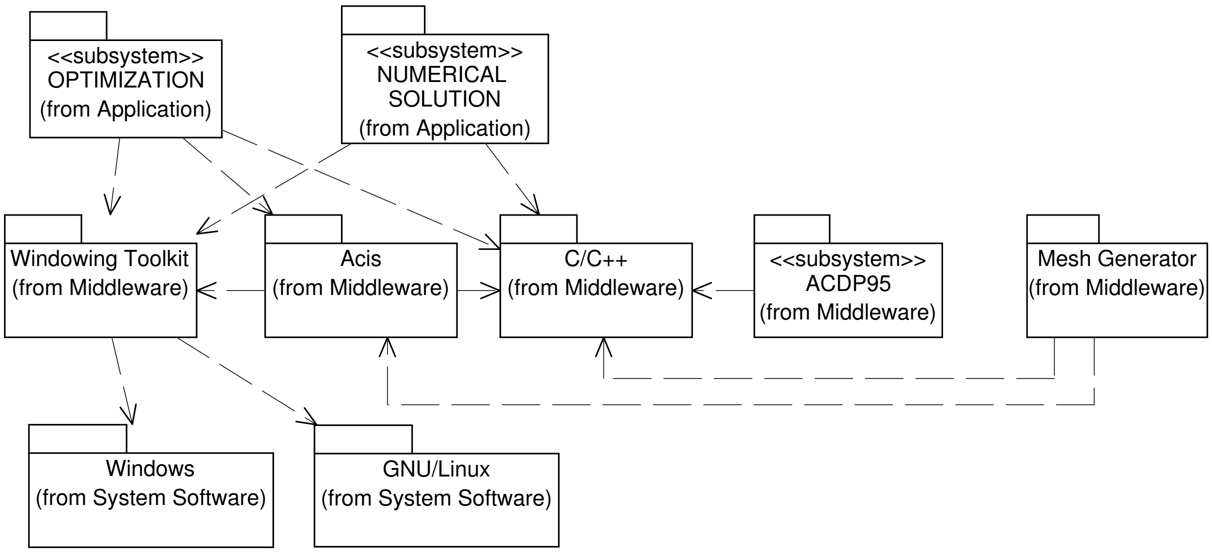


Figura 5.21: Dependências entre Application Subsystems e Middleware.

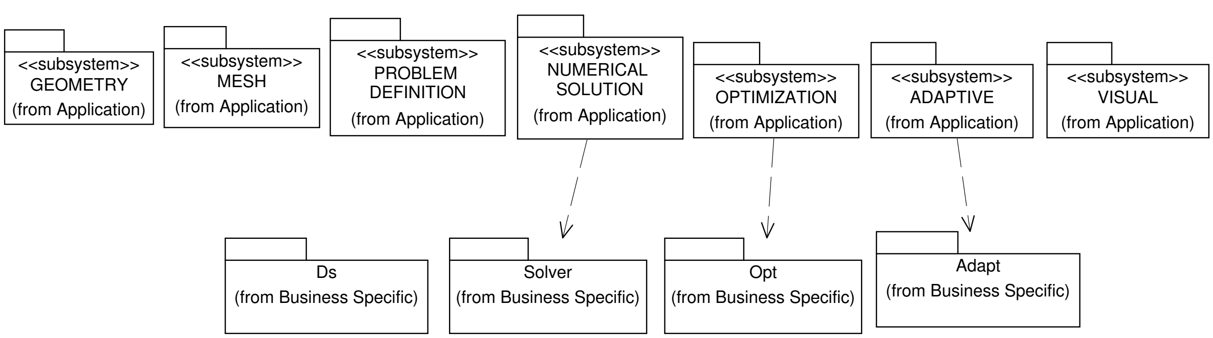


Figura 5.22: Dependências entre Application Subsystems e Business Specific.

uma classe apenas enxerga a outra através de parâmetros de funções ou variáveis locais. Com isso, as modificações ficam mais localizadas e com menor possibilidade de se propagar para outras partes do sistema em consideração. Por outro lado, os pacotes Solver e Opt utilizam intensivamente todas as classes desse pacote (Figura 5.16) e, portanto, suas interfaces devem ser mantidas estáveis o tanto quanto possível.

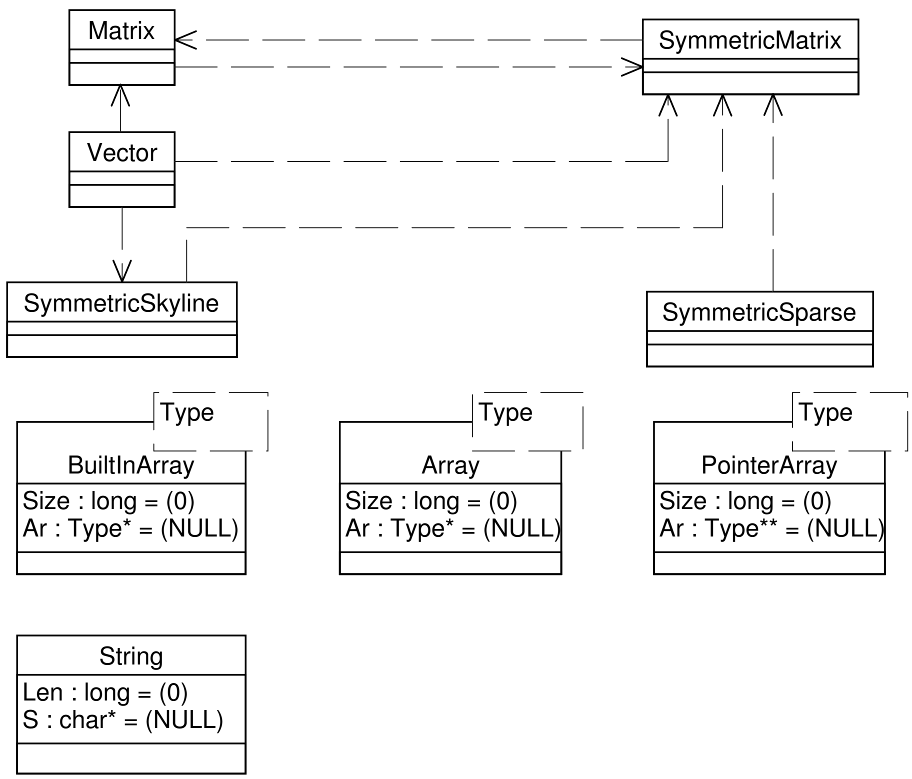


Figura 5.23: Classes do pacote Ds.

Pacote Solver

As funcionalidades desse pacote estão centralizadas nas classes DiscreteModel e Solver. A primeira armazena o modelo numérico em termos usuais de nós, elementos, materiais, condições de contorno nodais e propriedades geométricas em listas centralizadas. Na segunda, tem-se as entidades para armazenamento do sistema global, procedimentos para a sua construção e métodos de solução de modelos numéricos lineares e não-lineares. A classe auxiliar SolutionManager é responsável pelo gerenciamento dos arquivos de entrada e saída, acesso a banco de dados para salvamento de resultados e controle da seqüência de ações de solução de problemas através do envio de mensagens para objetos de DiscreteModel e Solver. O diagrama da Figura 5.24 mostra a relação entre essas classes e os atributos e operações

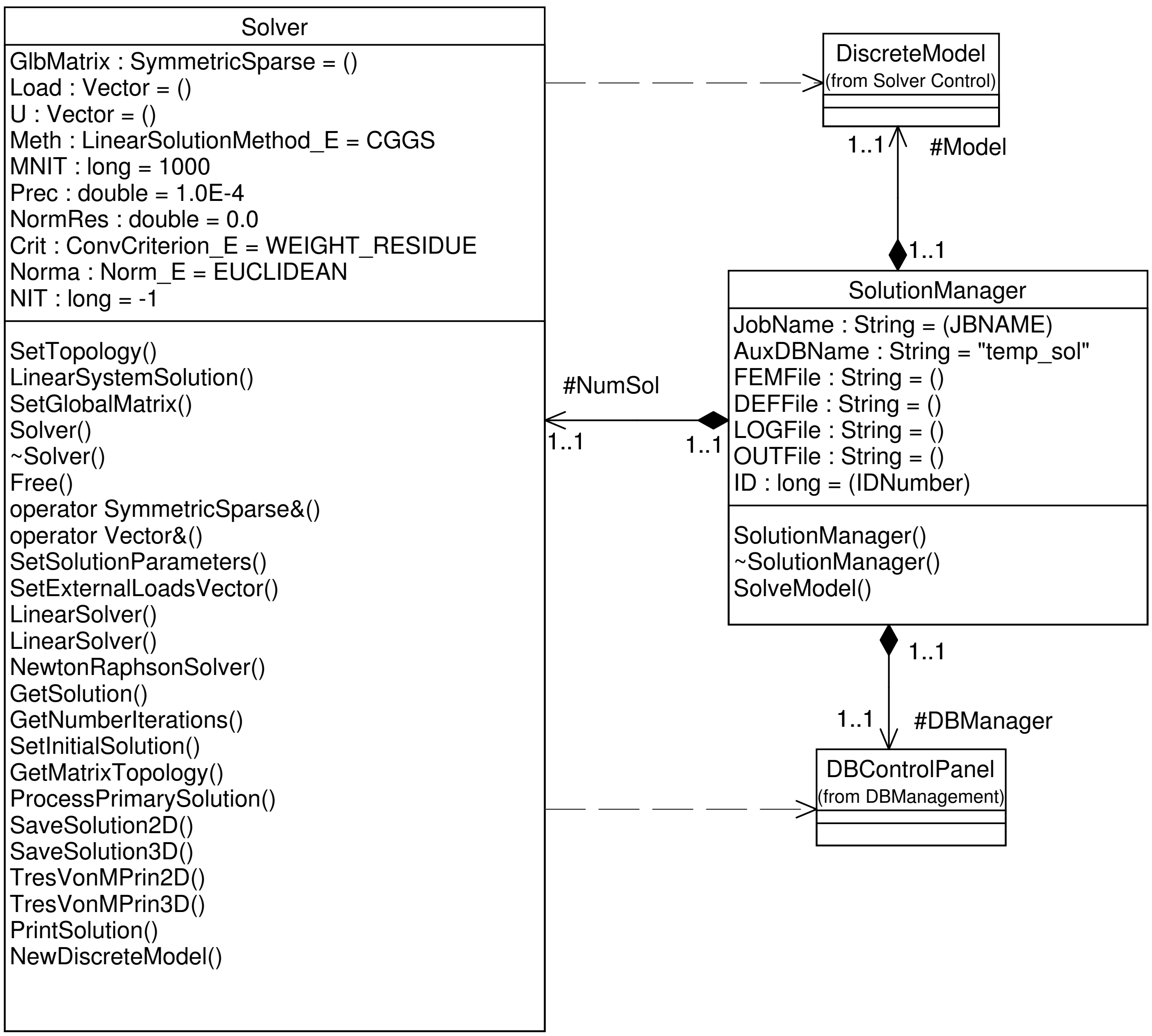


Figura 5.24: Classes responsáveis pela descrição e solução de modelos discretos.

das classes Solver e SolutionManager. A classe DBControlPanel pertence ao pacote CFS e controla o acesso ao banco de dados para salvamento de resultados.

As agregações que formam a classe DiscreteModel são mostradas na Figura 5.25: Coords, Groups, DOFs, DOFBCs, LSets, Materials, NoNo e NoEl são, respectivamente, instâncias de classes de nós (Nodes), grupos de elementos finitos (FEGroups), equações (Equations), condições de contorno de Dirichlet (DOF-BoundaryConditions), carregamentos (LoadSets), materiais (MaterialArray) e topologias de malha (Node-NodeNeighborhood e NodeElementNeighborhood). Tem-se ainda a classe ProblemDescription, que armazena informações genéricas usadas para decidir quais algoritmos devem ser usados durante a solução.

A maior parte das entidades desse pacote é construída usando classes template do pacote Ds (Bittencourt, 2000). Esse é o caso da classe MaterialArray, que descreve o conjunto de materiais de um modelo e consiste de um array de objetos MatInfo, como mostrado na Figura 5.26. Observa-se que a classe MatInfo está associada por referência à classe abstrata *Material*, da qual derivam os tipos específicos de modelos tais como elástico isotrópico linear, Mooney-Rivlin ou elastoplástico J_2. Assim, um modelo discreto pode conter qualquer número de materiais de tipos e propriedades diferentes, sendo esses materiais conhecidos apenas no momento de execução do sistema. Além disso, novos tipos de materiais podem ser incluídos, fazendo apenas novas derivações, uma vez que os algoritmos gerais são implementados em função da interface das classes MaterialArray ou *Material* e não das classes que implementam tipos específicos de materiais. No momento da execução, os algoritmos acessam as funcionalidades internas de cada modelo de material através da interface da classe genérica *Material*.

A classe abstrata *FiniteElement* implementa um tipo genérico de elemento finito. Especializações dessa classe definem tipos específicos como, por exemplo, o elemento estrutural tetraédrico para deformação infinitesimal. O esquema de interpolação da classe *FiniteElement* é encapsulado pela classe Interpolation (Figura 5.27), a qual mantém duas associações com *ShapeFunctions*: uma para interpolação geométrica (GeoInterpolation) e outra para interpolações dos campos de resposta (FieldInterpolation), principais e mistos. Esse esquema permite ter elementos sub, iso e superparamétricos de forma transparente para a classe *FiniteElement*, pois os cálculos de jacobianos e derivadas globais são responsabilidades de Interpolation.

Os algoritmos são implementados usando a interface pública de *FiniteElement* e *ShapeFunctions*. Logo, novos tipos de elementos e funções de forma podem ser incluídos apenas definindo-se especializações adicionais de cada uma dessas classes (ver Figura 5.3). Seguindo o mesmo estilo, as regras de

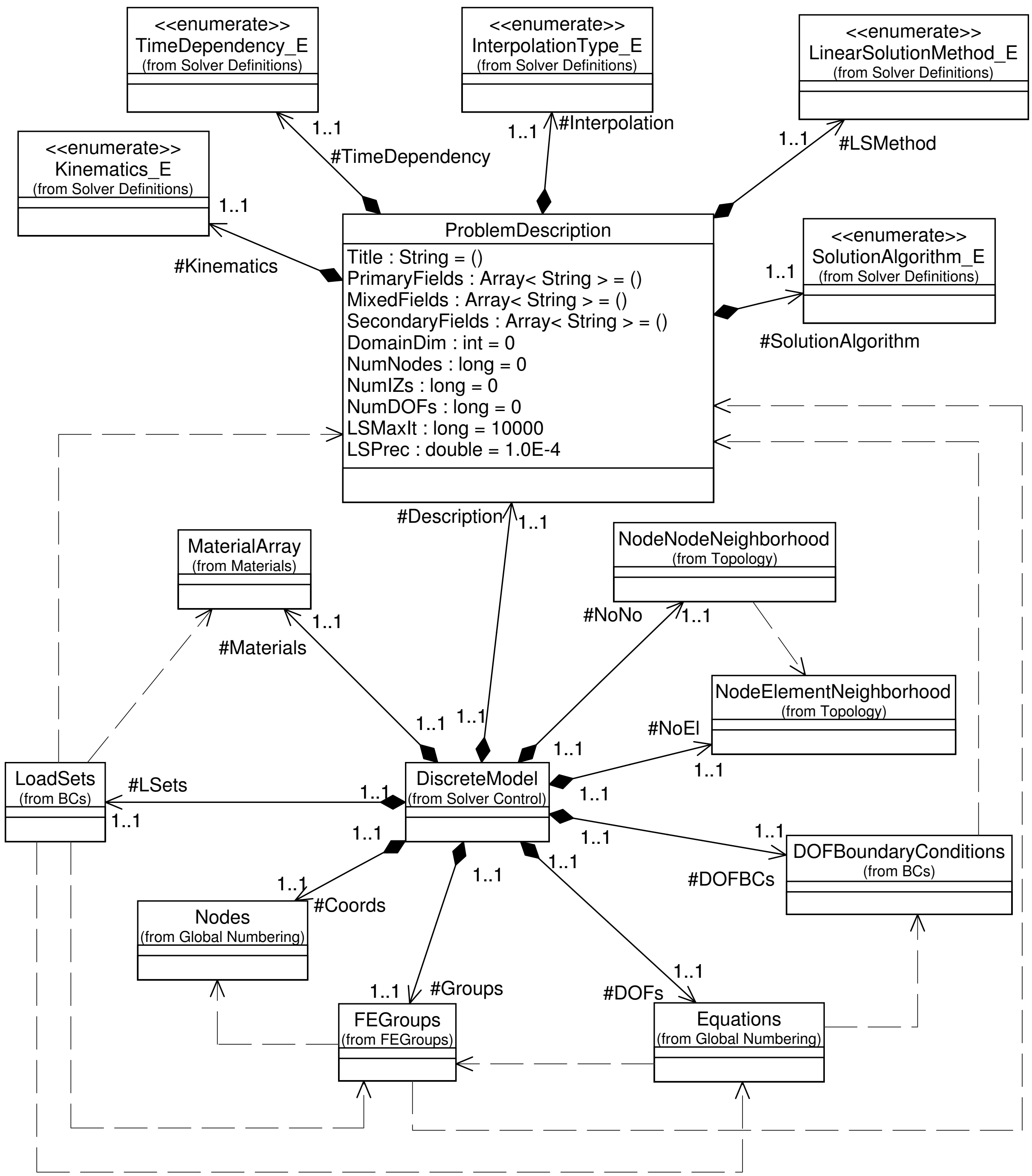


Figura 5.25: Composição da classe DiscreteModel.

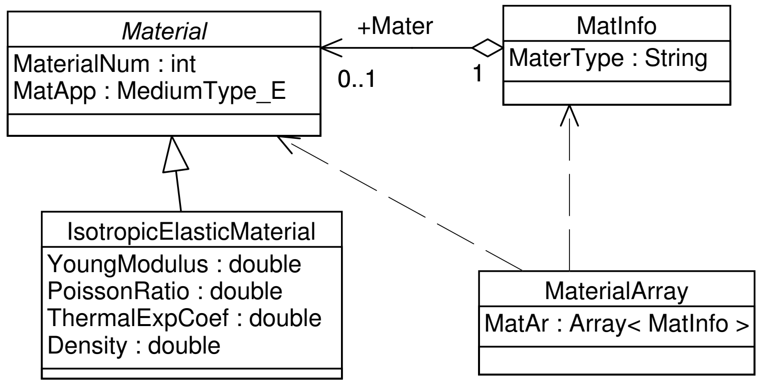


Figura 5.26: Classes para descrição de materiais e armazenamento de propriedades.

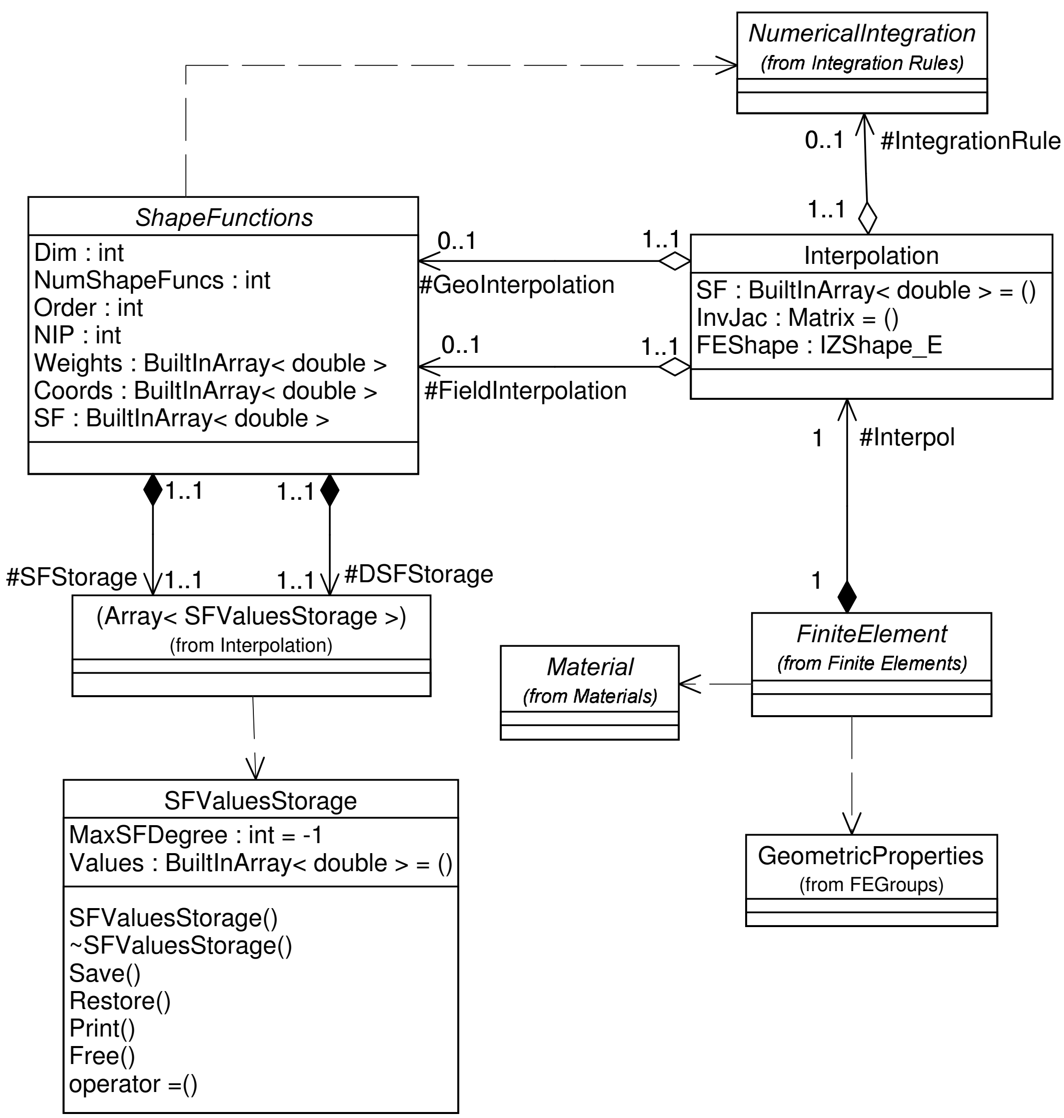


Figura 5.27: Relações da classe *FiniteElement* e as classes de interpolação.

integração são especializações da classe *NumericalIntegration*.

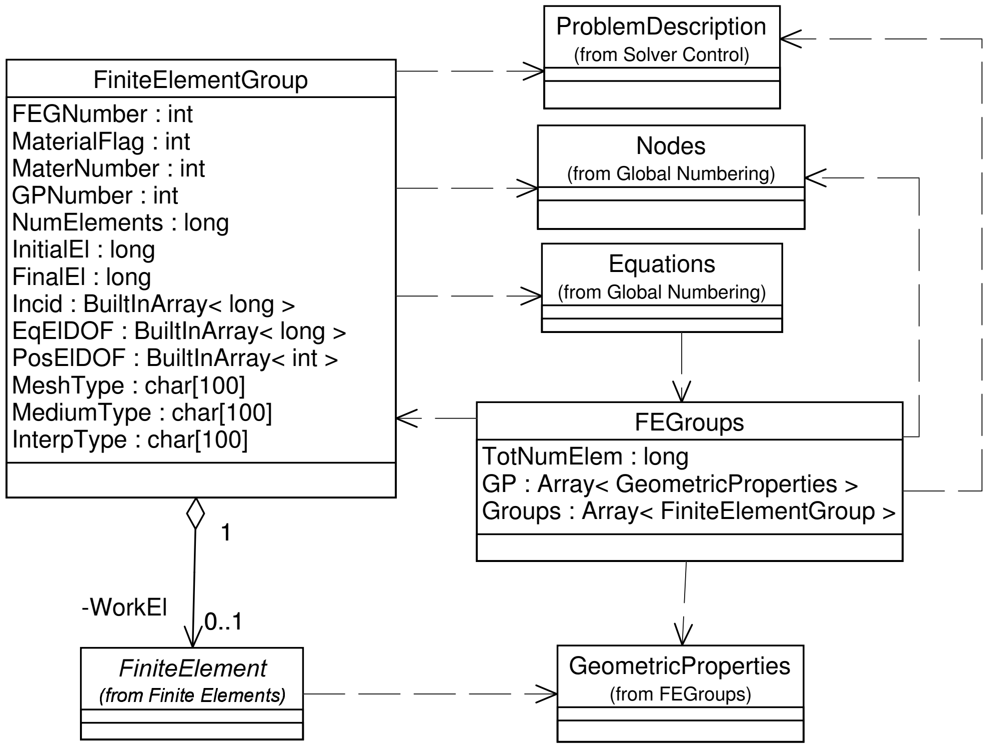


Figura 5.28: Classes para representação e manipulação do conjunto de elementos do modelo discreto.

A classe FEGroups representa o conjunto de todos os elementos do modelo discreto (Figura 5.28), sendo seu principal atributo um array de objetos FiniteElementGroup (Array<FiniteElementGroup>). O atributo GP : Array<GeometricProperties> armazena as propriedades geométricas de todo o modelo. Cada objeto FiniteElementGroup define um grupo de elementos, conjunto de elementos de mesmo tipo (formato e esquemas de interpolação e integração), material e propriedades geométricas. Seus principais atributos são a incidência do grupo (armazenada seqüencialmente no atributo Incid : BuiltInArray<long>) e um apontador WorkEl para *FiniteElement*, o qual pode ser inicializado como qualquer uma de suas especializações.

Quando é necessário fazer algum cálculo a nível elementar, por exemplo, a construção da matriz de rigidez local de determinado elemento, WorkEl recebe todos os seus dados – incidência, coordenadas nodais, numeração das equações de seus graus de liberdade, número de identificação de material e propriedades geométricas – e assume sua identidade. As dependências mostradas nas Figuras 5.27 e 5.28 indicam as entidades que *FiniteElement* precisa acessar em situações desse tipo.

Foi implementada estratégia de armazenamento dos valores das funções de forma e suas derivadas

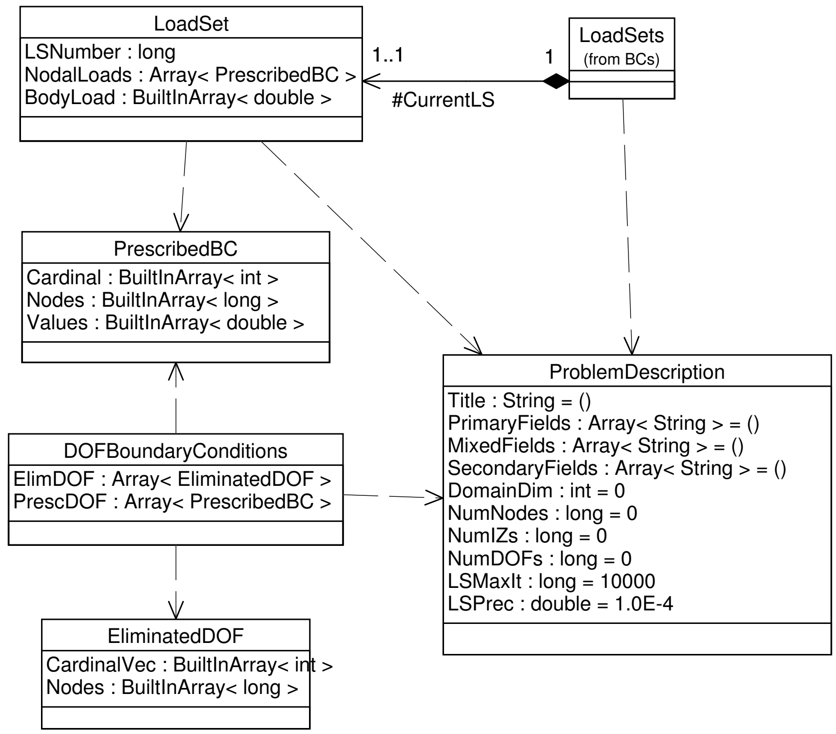


Figura 5.29: Classes para representação e manipulação de condições de contorno de Dirichlet e Newman.

nos pontos de integração do elemento de trabalho WorkEl. A classe *ShapeFunctions* é responsável por esse esquema e por isso depende da classe *NumericalIntegration*. Os valores são armazenados nessa classe quando requisitados pela primeira vez em qualquer ponto do programa, sendo os valores relativos a cada ordem de integração armazenados independentemente das demais. Esse esquema possibilita ganhos de eficiência na solução de problemas não-lineares com integração consistente e em determinadas formulações p de elementos finitos, nas quais as funções de forma e suas derivadas não são definidas explicitamente mas através de algoritmos numéricos (Karniadakis e Sherwin, 1999).

As condições de contorno e carregamentos são descritas pelas classes EliminatedDOF, PrecribedBC, DOFBoundaryConditions, LoadSet e LoadSets (Figura 5.29). As dependências em relação a ProblemDescription são necessárias para acessar os nomes dos graus de liberdade. Condições de Dirichlet nulas são representadas por EliminatedDOF, condições de Dirichlet não-nulas e cargas nodais são representadas por PrecribedBC. DOFBoundaryConditions define o conjunto de condições de Dirichlet do modelo e LoadSet os carregamentos (cargas nodais e as forças de corpo). Essas duas classes consistem basicamente de arrays de objetos EliminatedDOF e PrecribedBC.

Um objeto LoadSet representa um caso de carregamento em problemas lineares e um passo de carregamento em problemas não-lineares. LoadSets representa o conjunto total de carregamentos do modelo.

Pacote Opt

No pacote Opt, a classe DesignModel (Figura 5.30) funciona como entidade central. Essa classe agrega todas as características de um modelo parametrizado: variáveis de projeto, funcionais de performance, campos de velocidades de projeto, modelo discreto, solver, descrição geométrica e procedimentos de atualização. Com o método Eval(), avaliam-se estimativas de projeto, ou seja, verifica-se o valor dos funcionais de performance para um conjunto de valores de variáveis. Com os métodos AdjointDerivative() e DirectDerivative(), avaliam-se os gradientes desses funcionais. Portanto, juntamente com suas agregações, essa classe tem a responsabilidade de executar todos os procedimentos de análise (análise de resposta e análise de sensibilidade) e atualização de projetos (alteração de geometrias e malhas).

Os algoritmos de minimização são implementados como especializações de DesignModel. Tem-se, por exemplo, a classe HIPMethod, que implementa o método de pontos interiores de Herskovits (Herskovits, 1986; Panier *et al.*, 1988). A responsabilidade das classes de algoritmos de minimização é de atingir o objetivo do problema de otimização de modo sistemático, usando apenas os métodos DesignModel::Eval(), DesignModel::AdjointDerivative() ou DesignModel::DirectDerivative(). Por isso, todas as agregações de DesignModel são invisíveis para suas especializações (isso é indicado pelo sinal '-' em frente aos nomes das agregações). Em outras palavras, o algoritmo de minimização não precisa conhecer o modelo, mas apenas valores de funcionais e gradientes.

Usando a classe DBControlPanel, DesignModel armazena em banco de dados as diversas versões de geometria, malha e resultados obtidos no decorrer do processo de otimização. DesignUpdateMethods é responsável pela atualização de geometria e malha no decorrer do processo de otimização.

Como a classe DesignModel não é abstrata, é possível utilizá-la isoladamente (sem algoritmos de minimização associados) para efetuar modificações interativamente, avaliando performance e sensibilidade de cada projeto.

Na Figura 5.31, tem-se a estrutura de variáveis de projeto do sistema. A classe DesignModelVariables apresenta a agregação Variables, a qual consiste num array de ponteiros para objetos *DesignVariable*. Cada ponteiro corresponde à definição de uma variável de projeto e é inicializado em tempo de execução

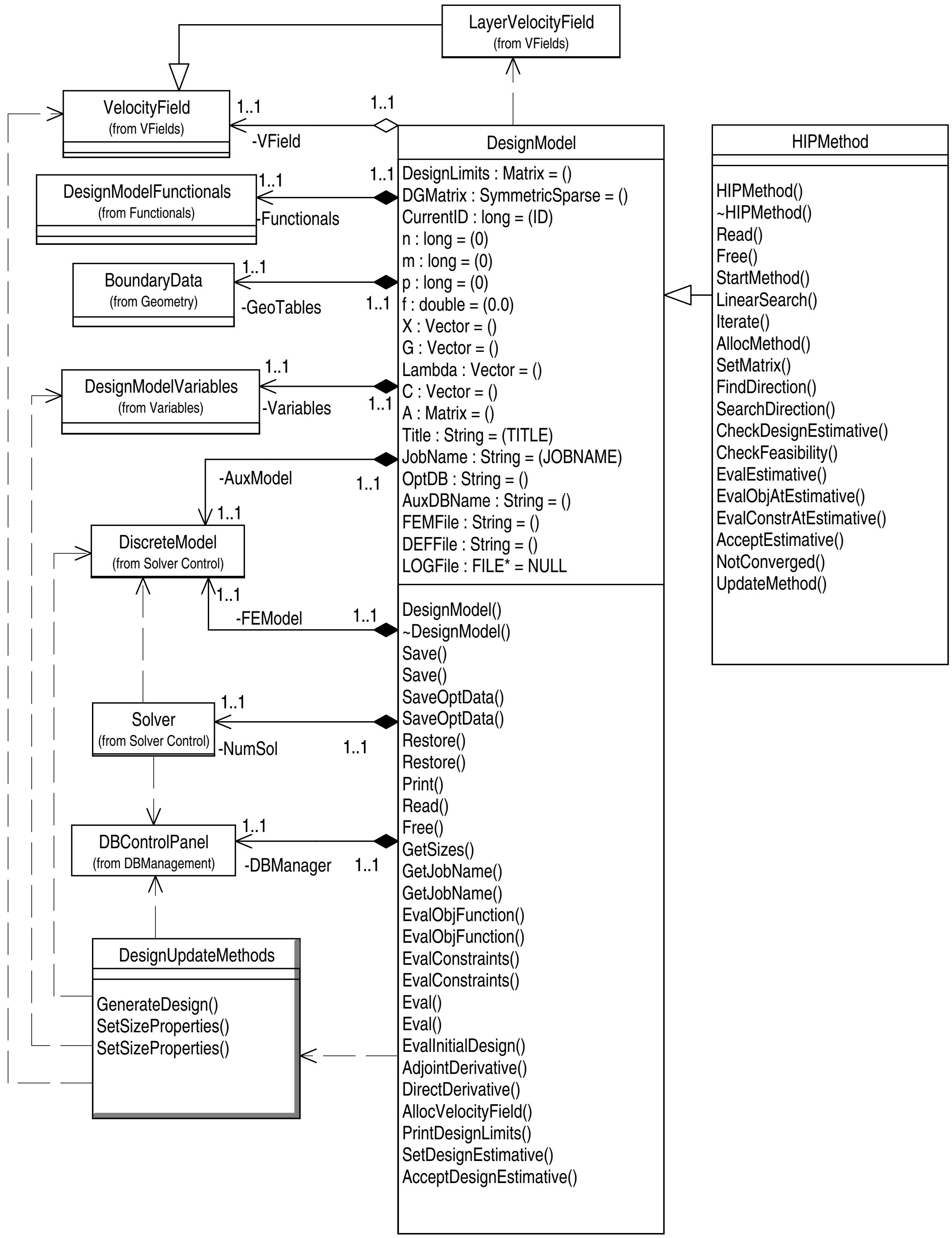


Figura 5.30: Classes de modelo parametrizado e algoritmo de otimização.

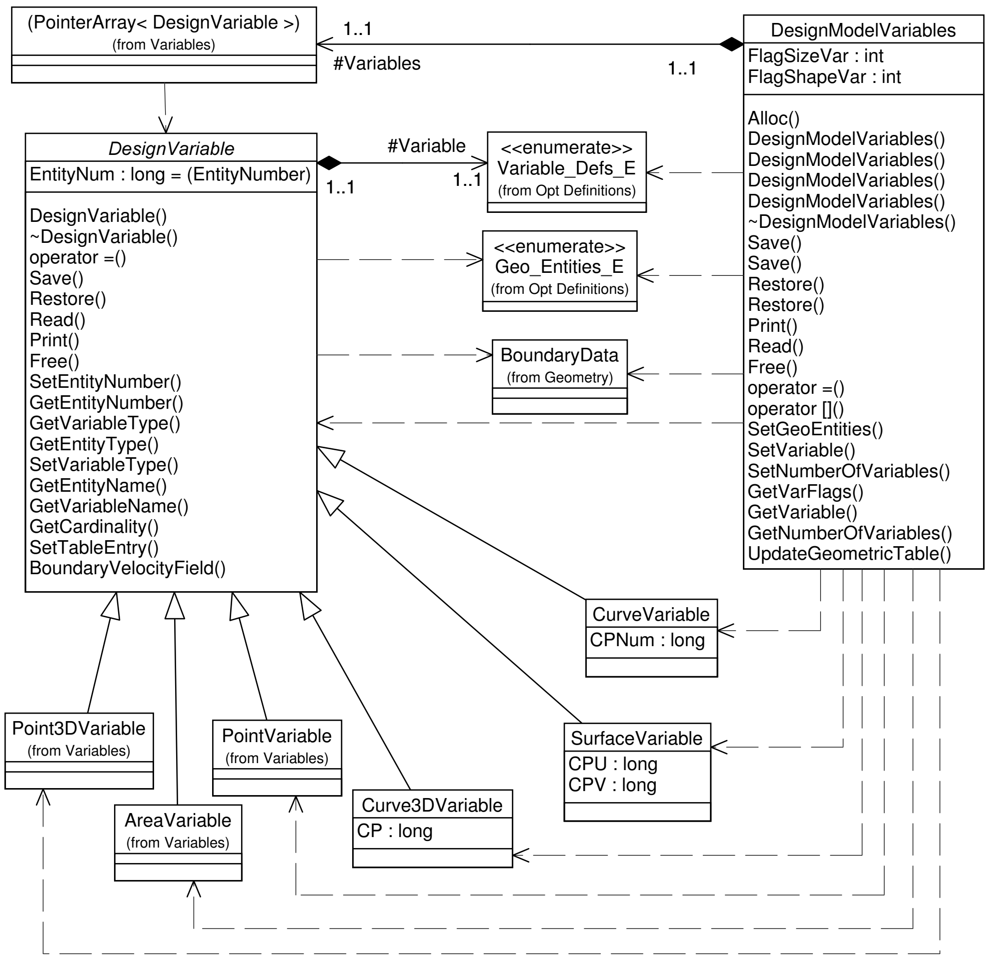


Figura 5.31: Estrutura de variáveis para parametrização de projetos.

com alguma das especializações de *DesignVariable*.

Uma maneira natural de especificar variáveis de projeto é associá-las a entidades geométricas (Silva e Bittencourt, 2000). Tem-se, por exemplo, SurfaceVariable e CurveVariable como especializações de *DesignVariable*. A primeira para variáveis associadas a superfícies NURBS (coordenadas ou pesos de pontos de controle) e a segunda para variáveis de curvas NURBS localizadas no plano. Novos tipos de variáveis podem ser incluídos com facilidade, pois somente DesignModelVariables depende das classes especializadas.

Os funcionais de performance seguem praticamente o mesmo padrão usado para as variáveis de projeto, como se observa na semelhança entre os diagramas de classes das Figuras 5.31 e 5.32. A única diferença significativa é a existência da associação adicional ObjFunction para a representação da função objetivo. Segundo as propriedades do padrão adotado, as dependências em relação às especializações de *PerformanceFunctional* estão encapsuladas por DesignModelFunctionals.

Outro ponto do sistema que requer flexibilidade é o esquema de geração de campos de velocidade de projeto. Campos de velocidade no contorno dependem apenas da forma matemática usada para descrever a geometria. No entanto, sua extensão para o interior do domínio pode ser feita de diversas formas, desde que sejam seguidos os aspectos discutidos no Capítulo 4. A classe abstrata VelocityField (Figura 5.33) é responsável pelo cálculo dos campos de velocidade no contorno e suas especializações pela extensão dos campos para o interior do domínio.

LayerVelocityField constrói um campo de velocidades não-nulo apenas na camada de elementos adjacentes ao contorno parametrizado. O procedimento consiste apenas em identificar a camada de elementos e usar funções de forma lineares para determinar as velocidades dos nós internos à camada, interpolando-se os valores no contorno externo e os valores nulos da outra extremidade da camada, em caso de eelementos quadráticos. Para isso, LayerVelocityField precisa acessar NodeElementNeighborhood e *ShapeFunctions*. A agregação NoEl é feita por referência para acessar a topologia nó-elemento do modelo de elementos finitos. SF também é acessado por referência, pois é necessário inicializar a função de forma correspondente à geometria dos elementos em uso.

BDVelocityField resolve um problema linear fictício no qual o campo de velocidades no contorno é interpretado como um campo de deslocamentos impostos. As relações com MaterialArray, DOFBoundaryConditions e Equations permitem a construção e solução dos problemas auxiliares correspondentes

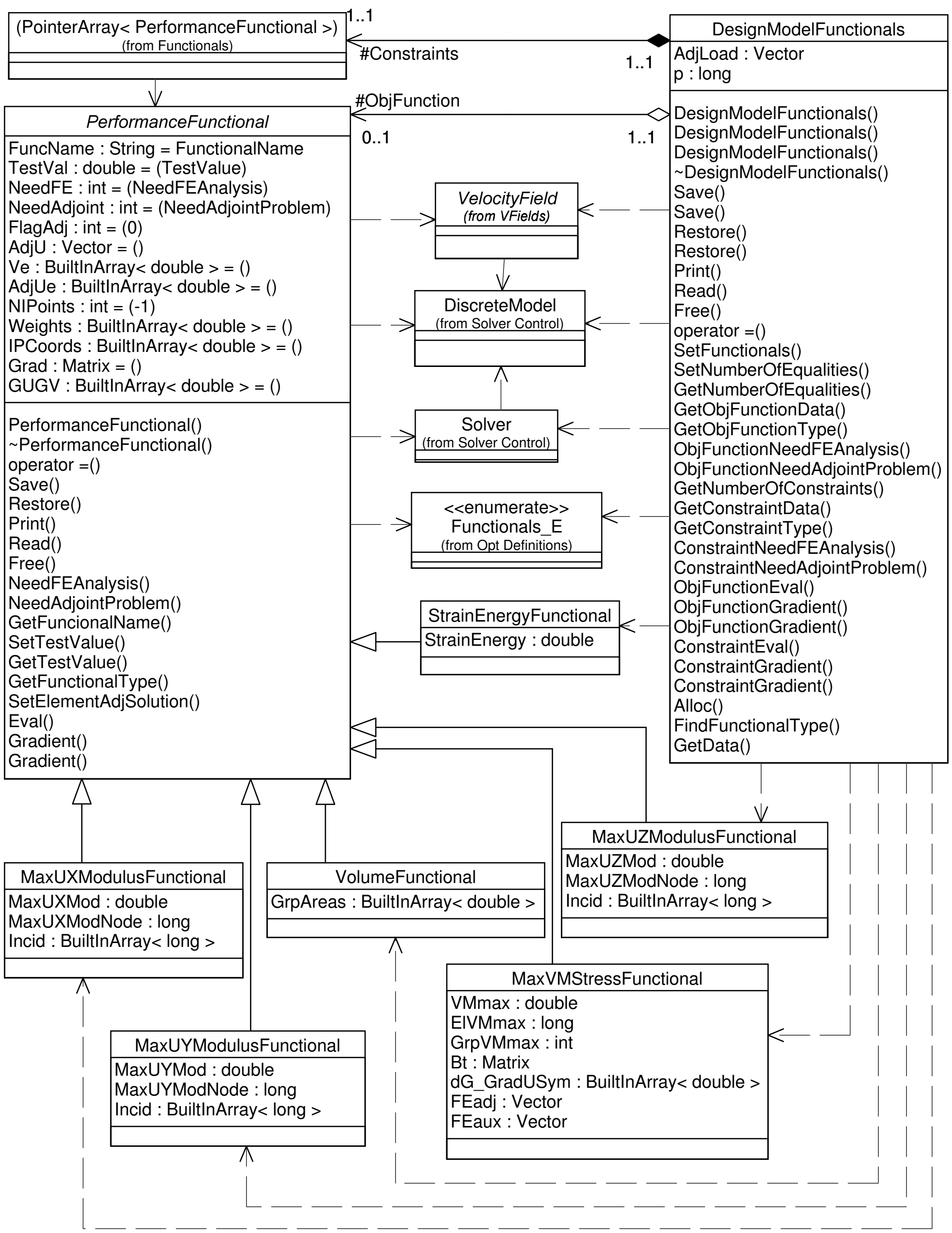


Figura 5.32: Classes para definição e manipulação de funcionais de performance.

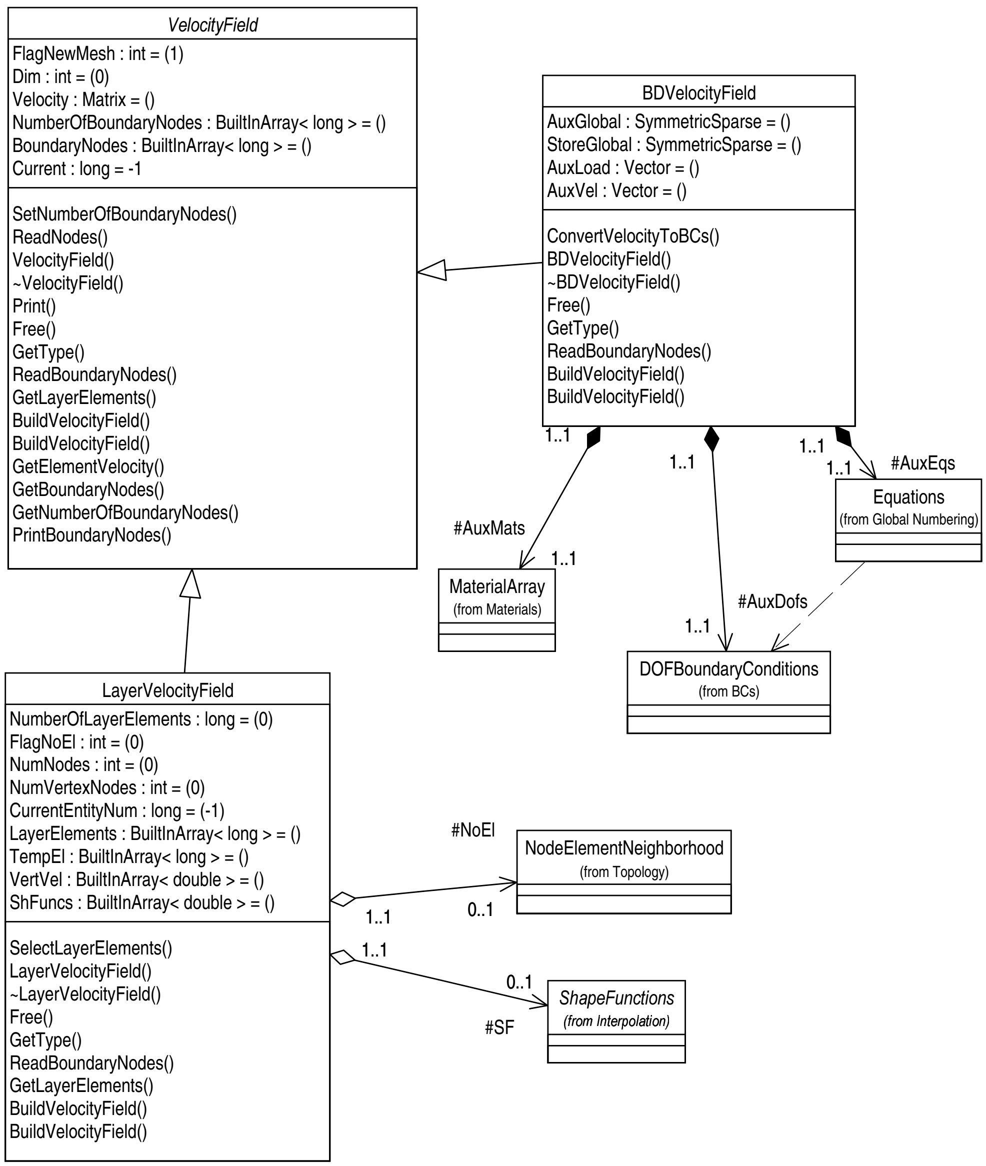


Figura 5.33: Classes de campos de velocidade de projeto.

a cada variável de forma.

5.4.4 Visão Arquitetônica do Modelo de Componentes

A organização dos arquivos do código-fonte é feita de forma eficiente pelas ferramentas de compilação, não sendo necessário tratá-la do ponto de vista da arquitetuta. É mais interessante visualizar as relações entre os executáveis do sistema e os arquivos de entrada, saída e bancos de dados. Relações de dependência têm os seguintes significados:

- um arquivo depende de um executável quando é gerado por ele;
- um banco de dados depende de um executável quando dados devem ser incluídos ou atualizados por esse executável;
- um executável depende de um arquivo (ou banco de dados) quando precisar ler seus dados durante a execução.

A Figura 5.34 mostra as dependências entre arquivos durante a solução do projeto JobName. Os dados de geometria são armazenados em arquivos `JobName_n.sat` e malhas, definições de problema e resultados são armazenados em banco de dados ACPD nos arquivos `JobName.dir` e `JobName.bdg`. Os arquivos ascii `JobName.fem` (malha) e `JobName.def` (definições de problema) são a entrada de dados para os procedimentos de análise de resposta. Os executáveis `Mesh.exe` e `Definiton.exe` também armazenam essas informações no banco de dados. `JobName.opt` é o arquivo de entrada para os procedimentos numéricos de otimização cujos resultados (geometrias, malhas, definições e resultados), ao final de cada iteração, são armazenados em arquivos SAT e no banco de dados. As mensagens do sistema são retornadas por todos os executáveis no arquivo `JobName.log`.

5.4.5 Fases de Desenvolvimento

Ao final da fase de Elaboração, a arquitetura obtida foi validada para problemas lineares bi e tridimensionais na versão h de elementos finitos, reproduzindo os mesmos resultados de análise de resposta obtidos com a versão anterior do código. As inovações nos cálculos de campos de velocidade e atualização de malhas originaram melhores resultados de análise de sensibilidade e otimização.

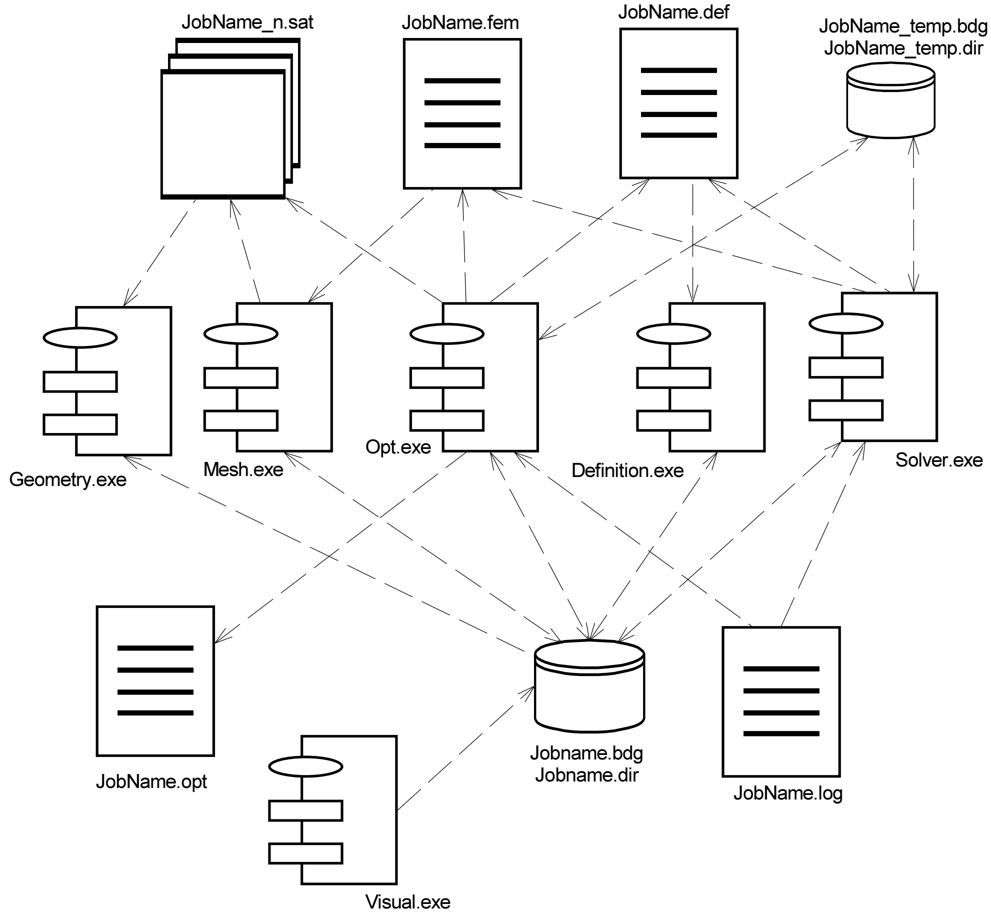


Figura 5.34: Dependências entre arquivos de entrada, saída, bancos de dados e executáveis do sistema.

Na fase de Construção, a arquitetura implementada serviu de base para a implementação de análise de resposta e sensibilidade para não-linearidades geométricas, modelo de material hiperelástico e incompressibilidade. Foram inicialmente introduzidos elementos cinemática finita e incluídos os métodos de passos de carregamento e Newton-Raphson na classe Solver. Método para cálculo de força interna foi introduzido na classe FiniteElement. Para o tratamento de modelo de material hiperelástico incompressível foram criadas uma nova classe de material e elementos com cinemática finita incompressível.

Paralelamente, foram implementadas otimização topológica (Driemeier, 2002), modelo de dano e geração de malhas (Ferreira, 2002), e versão p de elementos finitos (Nogueira Jr., 2002).

Capítulo 6

Conclusões e Perspectivas Futuras

Diferentes tópicos dos campos de análise de resposta e análise de sensibilidade em hiperleasticidade não-linear, bem como de técnicas de atualização de malhas, otimização e desenvolvimento de programas orientados por objetos foram discutidos neste trabalho. Tais tópicos se integram no contexto da implementação do núcleo central de uma biblioteca de programas para análise e projeto de estruturas auxiliados por recursos de Mecânica Computacional. As conclusões do presente trabalho e perspectivas de continuidade são apresentadas nas seções a seguir.

6.1 Análise de Resposta

- O método de projeção de pressão fornece comportamento equivalente à formulação mista correspondente em termos da ordem de interpolação da pressão, ao mesmo tempo que simplifica a inclusão de modelos de materiais incompressíveis em códigos de análise estrutural baseados em deslocamentos.

- A interpolação constante da pressão é satisfatória para o domínio de validade do modelo de material de Mooney-Rivlin quasi-incompressível quando associada à interpolação quadrática do campo de deslocamentos com elementos Serendipty (ou seja, quadriláteros 8/1 para problemas planos e hexaedros 20/1 para problemas tridimensionais).

  Modelos de material utilizados para analisar altas taxas de deformação, entretanto, exigem a aplicação de outros tipos de interpolações. Na versão h de elementos finitos, boas relações custo/benefício são observadas na interpolação do campo de deslocamentos com elementos quadráticos de Lagrange associados à interpolação linear da pressão hidrostática (*e.g.*, quadriláteros 9/3 e o elemento hexaédrico 27/4), vide (Bathe, 1996). Uma perspectiva promissora é passar a utilizar interpolações p de ordem mais alta para o campo de deslocamentos (Nogueira Jr., 2002; Chilton e Suri, 2001).

- Na simulação de coxins de aplicações automobilísticas, tais como sustentação de motores, fixação de amortecedores, etc., os elastômeros são razoavelmente rígidos e apresentam deformação finita moderada em utilização normal. Portanto, a formulação empregada é bastante conveniente.

- Não é usual observar na literatura a utilização de elementos triangulares e tetraédricos. Nos exemplos apresentados no Capítulo 2, exemplificou-se a aplicação efetiva desses tipos de elementos, os quais são fundamentais numa ferramenta de otimização de forma de propósito geral.

- Todos os cálculos foram efetuados com integração consistente, a menos dos termos da matriz de elasticidade tangente do modelo quasi-incompressível que não podem ser descritos exatamente por polinômios. Na literatura, é recorrente a utilização de integração numérica aproximada.

  Deve-se observar que o conceito de integração reduzida é bastante diferente da aplicação de integração aproximada. A integração reduzida consiste numa técnica fundamentada para o tratamento de incompressibilidade, enquanto que integração aproximada significa apenas utilizar números menores de pontos de integração do que o necessário para a integração exata do polinômio em questão.

- A solução da análise de resposta requer a inclusão de busca linear (foi implementada minimização da energia de deformação) para acelerar a convergência. Técnicas adicionais, tais como a divisão automática de passos de tempo, são recomendáveis.

- O modelo de incompressibilidade dá origem a matrizes de rigidez tangente mal-condicionadas, uma vez que a rigidez é muito maior em direções de compressão e tração em comparação com direções de cisalhamento.

  Pré-condicionadores eficientes em problemas de elasticidade linear não mostraram bons resultados quando aplicados em problemas de hiperelasticidade não-linear incompressível, dificultando a aplicação de métodos iterativos de solução de sistemas lineares. Contudo, a aplicação de tais métodos seria interessante uma vez que possibilita a utilização de respostas de modelos semelhantes como estimativa inicial para o algoritmo de Newton-Raphson (o que seria muito conveniente no contexto de otimização, no qual se conhece um conjunto de soluções semelhantes anteriores). Além disso, como o método de Newton-Raphson é direcionado pela queda do módulo do incremento de deslocamento, espera-se que haja aceleração da seqüência de soluções ao se utilizar sempre um vetor de componentes nulas como estimativa inicial de cada solução de sistema linear. Alguns trabalhos têm mostrado bons resultados com métodos multigrid em não-linearidades estruturais de hiperleasticidade não-linear incompressível (Adams, 2000a).

- Num contexto de otimização de estruturas não-lineares, grande parte do custo está relacionado diretamente à análise de resposta. Mesmo empregando algoritmos de minimização de maior taxa de convergência, ganhos de eficiência e estabilidade na etapa de análise de resposta são fundamentais para a adoção da otimização no projeto de estruturas envolvendo modelos não-lineares.

- Aproveitando a modularidade do código, passos seguintes seriam incorporar outros modelos de material à análise de resposta, contato e análise transiente, assim como a integração de interpolações de alta ordem.

- O custo computacional de otimização de problemas não-lineares (mesmo de pequena escala) é extremamente alto. O uso de computação de alta performance é essencial neste contetxo.

6.2 Análise de Sensibilidade

- As vantagens em precisão e eficiência na determinação de grandientes de funcionais de performance da análise de sensibilidade são mais evidentes em problemas não-lineares que nos lineares.

  Funcionais de performace estrutural definidos sobre campos de resposta de problemas lineares já apresentam comportamento não-linear, o qual se torna mais forte quando os campos de resposta resultam de análises de respostas estruturais não-lineares.

  Em aproximações por diferenças finitas, isso dificulta a escolha de tamanhos de perturbação convenientes. Além disso, a determinação da resposta de domínios perturbados, ainda que a estimativa inicial seja a resposta conhecida do problema não-perturbado (em torno do qual se deseja a derivada) envolve iterações do método de Newton-Raphson, resultando num custo muito maior que o da análise de sensibilidade.

- A informação de sensibilidade se torna ainda mais importante quando a análise de resposta é não-linear devido à maior dificuldade em se extrapolar o comportamento do sistema a partir de variações de seus parâmetros.

- Por apresentar operador simétrico e estrutura conservativa, foi aplicado método adjunto de análise de sensibilidade devido à sua eficiênica. Em problemas envolvendo irreversibilidades, o método de diferenciação direta é mais eficiente e em problemas com operador não-simétrico, apenas o método de diferenciação direta pode ser aplicado. Portanto, ainda que a filosofia básica seja única, cada tipo de análise de resposta apresenta especificidades de implementação (Kim, 1999).

- A condensação da pressão hidrostática em cada célula simplifica bastante o procedimento de análise de sensibilidade. A princípio, a extensão para interpolações de maior ordem da pressão seguem o procedimento padrão de projeção da pressão (Chen e Pan, 1996; Chen *et al.*, 1996). Porém, essa extensão ainda não foi apresentada em nenhum trabalho.

- A extensão do núcleo do solver para executar cálculos de sensibilidade é simples e localizada, de acordo com os Apêndices C.1.3 e C.2.3. A avaliação da sensibilidade dos operadores (por exemplo, dos operadores de rigidez), foram implementadas dentro de cada elemento. Os funcionais de performance foram implementados em estruturas independentes, de forma que a inclusão de novos funcionais e suas respectivas expressões de sensibilidades são apenas clientes das sensibilidades dos operandos. Dessa forma, novos funcionais ou novos tipos de elementos podem ser incluídos independentemente.

- Desenvolvimentos seguintes envolvem a inclusão de recursos de análise de sensibilidade em elementos finitos na versão p e nos novos tipos de análise de resposta a serem considerados na biblioteca de programas.

6.3 Algoritmos de Campos de Velocidades, Atualização de Malhas e Otimização

- Os campos de velocidade são definidos univocamente apenas sobre o contorno, decorrendo das expressões paramétricas usadas na sua definição.

  Sua extrapolação para o interior do domínio é orientada apenas por requisitos teóricos (mesma regularidade do campo de deslocamentos e linearidade em relação às variáveis de projeto) e práticos (fornecer uma regra consistente de atualização da discretização e ser baseada num algoritmo genérico e reutilizável).

Se as exigências teóricas são respeitadas, soluções analíticas com qualquer campo levariam aos mesmos resultados de sensibilidade. Numericamente, se os campos respeitam tais condições, os resultados serão precisos no sentido padrão de verificação de resultados de análise de sensibilidade (Haug *et al.*, 1986; Santos e Choi, 1988; Kim, 1999; Tseng e Arora, 1989), que consiste em compará-los com derivadas obtidas por diferença finita central na qual as malhas das perturbações do domínio são obtidas com os mesmos campos de velocidades da análise de sensibilidade.

- Os campos de velocidades são entidades centrais num algoritmo de otimização de forma, sendo decisivos no próprio algoritmo de minimização, além de na avaliação de expressões de sensibilidade quanto em atualização de malhas. Isto porque os algoritmos de minimização baseados em gradientes assumem suavidade local dos funcionais de performance, a qual somente é atingida em funcionais com dependência implícita das variáveis de projeto quando o campo de velocidades é utilizado na atualização da discretização.

- A parametrização do contorno com entidades NURBS e a adoção de coordenadas de pontos de controle na parametrização da geometria compõem um algoritmo de geração de campos de velocidades sobre o contorno que respeita todos os requisitos práticos e teóricos.

  A único exigência dessa técnica é que as coordenadas paramétricas dos nós localizados sobre o contorno sejam conhecidas com precisão. Com esse objetivo, foi apresentado algoritmo de recuperação de coordenadas paramétricas que utiliza apenas os coordenadas globais dos nós e a expressão descrevendo cada superfície. Esse algoritmo consiste na solução de problema não-linear de baixo custo em cada nó do contorno por Newton-Raphson.

- Para a extrapolação dos campos de velocidades para o interior do domínio foram apresentadas duas técnicas respeitando os requisitos teóricos e práticos: método de camada unitária de contorno e método de deslocamentos fictícios de contorno.

  O primeiro tem custo bastante baixo, pois nenhum cálculo está envolvido, apenas a interpolação linear dos valores do contorno na camada de elementos adjacentes ao contorno. Porém, por gerar perturbações apenas numa única camada de elementos, há maior possibilidade de degradação da qualidade dos elementos durante atualizações da discretização.

Por sua vez, o método de deslocamentos fictícios do contorno tem maior custo, mas fornece campos de velocidades de melhor qualidade para atualização da geometria.

- O formato padrão do método de deslocamentos fictícios do contorno tem alto custo uma vez que os problemas elásticos auxiliares (um por variável de forma) são resolvidos exatamente. Adotando-se um método iterativo de solução de sistemas lineares, foi possível observar que reduzindo-se a precisão exigida para a solução, reduz-se bastante o custo associado ao método, sem impacto significativo nos resultados de análise de sensibilidade e otimização.

- Os resultados de sensibilidade obtidos com ambos os campos de velocidade são idênticos (a menos de erros de truncamento numérico). A escolha do método a ser utilizado depende da disponibilidade da aplicação do recurso de geração automática de malhas.

- Os elementos indicados pelas medidas de distorção propostas coincidem com aqueles também indicados pelos algoritmos de geradores de malhas comerciais. No caso de triângulos e tetraedros, uma única medida permite identificar razões de aspecto e ângulos obtusos.

- Os resultados de otimização obtidos no Capítulo 4 demonstram que os algoritmos desenvolvidos neste trabalho podem ser satisfatoriamente aplicados no projeto de componentes mecânicos compostos por materiais hiperelásticos incompressíveis. Especificamente, foi possível controlar com precisão a rigidez de coxins em modelos bi e tridimensionais.

- Os conceitos de atualização de malhas propostos podem ser aplicados em contextos além da otimização de forma. Especialmente, cita-se a utilização da descrição geométrica do contorno para a determinação de campos de velocidade sobre o contorno, os quais definem precisamente as condições de contorno para a autalização da malha no interior do domínio. O maior desafio dessa técnica é sua integração efetiva com sistemas de CAD comerciais.

6.4 Desenvolvimento de Programas

- Uma ferramenta de otimização de estruturas deve combinar de forma eficiente algoritmos e técnicas diversas. O nível de complexidade dos programas requer não somente técnicas de modelagem, mas uma abordagem mais global de gerenciamento de requisitos, casos de uso, iterações de desenvolvimento e fases.

- A adoção de um processo de desenvolvimento formal, tal como o Processo Unificado, favorece a compreensão global dos requisitos do sistema a ser desenvolvimento, dos fluxos de atividades e das responsabilidades dos subsistemas. Além disso, existe hoje grande disponibilidade de bibliotecas e ferramentas, aumentando a importância das atividades de planejamento e integração.

- Apesar de haver um paralelo entre as fases do Processo Unificado (abertura, elaboração, construção e transição) e a seqüência de desenvolvimento de programas científicos, o custo inicial de adoção desse processo é alto. A melhor estratégia é a implantação gradual a partir do núcleo básico que define o processo (Kruchten, 2002a; Evans, 2001; Probasco, 2000).

- As três bases do processo são: arquitetura, casos de uso e desenvolvimento iterativo. A arquitetura consiste na definição formal da abordagem e da filosofia de solução dos problemas (casos de uso) do domínio do sistema. Portanto, a arquitetura estabelece a linha geral a ser seguida pela equipe de desenvolvimento. Portanto, o Processo Unificado facilita a coordenação de esforços de grupos de trabalho e integração mais rápida e eficiente de novos integrantes.

- O design adotado segue a abordagem tradicional de implementação de programas científicos em C++: organização de estruturas de dados e abstrações em níveis mais altos com recursos de orientação por objetos e implementação efetiva dos dados e algoritmos em C. Com isso, busca-se aliar flexibilidade e eficiência.

  Considerando que a maior parte do processo de solução é gasto na solução de sistemas lineares – os quais são implementados em C puro – a relação custo/benefício é satisfatória.

  Ganhos incrementais de eficiência podem ainda ser obtidos com ajustes localizados em alguns conjuntos de classes. Porém, ganhos significativos de eficiência somente podem ser alcançados com a adoção de outras abordagens arquitetônicas, sendo a que se apresenta mais promissora atualmente aquela baseada em *expression templates* (Haney e Crotinger, 1999; Oldham, 2002). Essa técnica representa hoje o estado da arte em aplicações numéricas de orientação por objetos, sendo ainda tópico de pesquisa. Além disso, a aplicação intensiva de *expression templates* é menos flexível que a arquitetura adotada, exigindo maior investimento em tempo de projeto e uma compreensão mais profunda de todos os casos de uso do sistema.

- A arquitetura adotada é modular e flexível em termos de interpolação e integração, o que permitiu a extensão para a versão p de elementos finitos (Nogueira Jr., 2002). Além disso, o núcleo de solução não-linear pode ser aplicado em problemas fora de seu escopo original, tais como modelos de dano (Ferreira, 2002).

- A incorporação de um núcleo básico de manipulação de dados geométricos baseado em NURBS ao sistema tornou o sistema independente do programa de CAD utilizado na modelagem e do gerador de malhas aplicado na discretização.

Referências Bibliográficas

Acklam, E., Langtangen, H. (1999). Parallelization of explicit finite difference schemes via domain decomposition. Oslo Scientific Computing Archive 1999-2. URL: http://www.math.uio.no/OSCA; ISSN 1500-6050.

Adams, M. (2000a). Evaluation of three unstructured multigrid methods on 3D finite element problems in solid mechanics. *Submitted to: International Journal for Numerical Methods in Engineering*. http://www.cs.berkeley.edu/˜madams/amg.ps.

Adams, M. (2000b). Parallel multigrid solvers for 3d unstructured finite element problems in large deformation elasticity and plasticity. *International Journal for Numerical Methods in Engineering*, v. 48, n. 8, p. 1241–1262.

Adams, M., Demmel, J. (1999). Parallel multigrid solver algorithms and implementations for 3D unstructured finite element problem. In: *Proceedings of SC99: High Performance Networking and Computing*, Portland, Oregon.

Ansys Inc. (1995a). *Ansys Release 5.1 – Procedures Manual*.

Ansys Inc. (1995b). *Ansys Release 5.1 – Theory Manual*.

Archer, G., Fenves, G. , Thewalt, C. (1999). A new object-oriented finite element analysis program architecture. *Computers & Structures*, v. 70, n. 1, p. 63–75.

Arora, J. (1989). *Introduction to Optimum Design*. Mechanical Engineering Series. McGraw-Hill, New York.

Arora, J., Cardoso, J. (1992). Variational principle for shape design sensitivity analysis. *AIAA Journal*, v. 30, n. 2, p. 538–547.

Arora, J., Elwakeil, O. , Chahande, A. (1995). Global optimization methods for engineering applications - a review. *Structural Optimization*, v. 9, n. 3/4, p. 137–159.

Arora, J., Haug, E. (1979). *Applied Optimal Design - Mechanical and Structural Systems*. John Wiley & Sons, New York.

Augustine, L. (2001). Using the rational unified process (RUP) successfully for small development projects. *The Rational Edge*. http://www.therationaledge.com.

Balagangadhar, D., Tortorelli, D. (2000a). Design of large-deformation steady elastoplastic manufacturing processes. Part I: Displacement-based reference frame formulation. *International Journal for Numerical Methods in Engineering*, v. 49, p. 899–932.

Balagangadhar, D., Tortorelli, D. (2000b). Design of large-deformation steady elastoplastic manufacturing processes. Part II: Sensitivity analysis and optimization. *International Journal for Numerical Methods in Engineering*, v. 49, p. 933–950.

Balay, S., Gropp, W., McInnes, L. , Smith, B. (2001). PETSc users manual. Technical Report ANL-95/11 - Revision 2.1.1, Argonne National Laboratory, Argonne, IL.

Bathe, K. (1996). *Finite Element Procedures*. Prentice Hall, Englewood Cliffs.

Bazaraa, M., Sherali, H. , Shetty, C. (1993). *Nonlinear Programming – Theory and Applications*. John Wiley & Sons, New York, 2nd ed.

Becker, J., Bloebaum, C. , Hulme, K. (1997). Distributed computing for multidisciplinary design optimization using java. *Structural Optimization*, v. 14, p. 203–218.

Belegundu, A., Arora, J. (1985a). A study of mathematical programming methods for structural optimization. Part I: Theory. *International Journal for Numerical Methods in Engineering*, v. 21, p. 1583–1599.

Belegundu, A., Arora, J. (1985b). A study of mathematical programming methods for structural optimization. Part II: Numerical results. *International Journal for Numerical Methods in Engineering*, v. 21, p. 1601–1623.

Bennett, J., Botkin, M. (1985). Structural shape optimization with geometric description and adaptive mesh refinement. *AIAA Journal*, v. 23, n. 3, p. 458–464.

Benson, S., McInnes, L. , Moré, J. (2002). TAO users manual. Technical Report ANL/MCS-TM-242-Revision 1.3, Argonne National Laboratory, Argonne, IL.

Besson, J., Foerch, R. (1997). Large scale object-oriented finite element code design. *Computer Methods in Applied Mechanics and Engineering*, v. 142, n. 1-2, p. 165–187.

Bittencourt, M. (1990). Análise estática e dinânica por subestruturação e programação orientada por objetos. Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas. Dissertação (Mestrado).

Bittencourt, M. (1996). *Métodos multigrid e iterativos adaptáveis aplicados em malhas não-estruturadas*. Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas, Tese (Doutorado).

Bittencourt, M. (2000). Using C++ templates to develop finite element classes. *Engineering Computations*, v. 17, n. 7, p. 775–788.

Bittencourt, M., Feijóo, R. (1996). Análise comparativa de métodos diretos e iterativos para a solução de sistemas de equações. Relatório de Pesquisa e Desenvolvimento 01/96, LNCC, Rio de Janeiro.

Bittencourt, M., Feijóo, R. (1997). Análise comparativa de métodos diretos e iterativos para a solução de sistemas de equações. *Revista Internacional de Métodos Numéricos y Diseño em Ingenieria*, v. 13, n. 2, p. 123–148.

Bittencourt, M., Feijóo, R. (1998). Object-oriented non-nested multigrid methods. In: I. Idelsohn, E. O., Dvorkin, E., (Eds.), *Computational Mechanics – New Trends and Applications*, p. 1–15, Barcelona. CIMNE.

Bittner, K. (2000). Why use cases are not functions. *The Rational Edge*. http://www.therationaledge.com.

Bonet, J., Wood, R. (1997). *Nonlinear Continuum Mechanics for Finite Element Analisys*. Cambridge University Press, Cambridge.

Booch, G. (1991). *Object Oriented Design with Applications*. The Benjamin/Cummings Publishing Company, Inc., USA.

Booch, G., Rumbaugh, J. , Jacobson, I. (1999). *The Unified Modeling Language User Guide*. Addison Wesley, Reading (MA).

Boogaard, A., Vliet, N. , Huétink, J. (1998). Object oriented design of a thermo-mechanical FEM code. In: S. Idelsohn, E. Oñate, E. D., (Editor), *Computational Mechanics - New Trends and Applications*, p. 1–17, Barcelona. CIMNE.

Botkin, M. (1992). Three-dimensional shape optimization using fully automatic mesh generation. *AIAA Journal*, v. 30, n. 7, p. 1932–1934.

Botkin, M., Bennett, J. (1986). The application of adaptive mesh refinement to shape optimization of plate structures. In: Babuška, I., Zienkiewicz, O. , J. Gago, E. O., (Eds.), *Accuracy Estimates and Adaptive Refinements in Finite Element Computations*, Cap. 13, p. 227–246. John Wiley & Sons.

Brenner, S., Scott, L. (1994). *The Mathematical Theory of Finite Element Methods*. Springer-Verlag, New York.

Brezzi, F., Fortin, M. (1991). *Mixed and Hybrid Finite Element Methods*. Springer Series in Computational Mathematics. Springer-Verlag, New York.

Bugeda, G., Oliver, J. (1993). A general methodology for structural shape optimization problems using automatic adaptive remeshing. *International Journal for Numerical Methods in Engineering*, v. 36, n. 18, p. 3161–3185.

Buhrer, K. (2000). From craft to science: Searching for first principles of software development. *The Rational Edge*. http://www.therationaledge.com.

Buscaglia, G., Durán, R., Fancello, E., Feijóo, R. A. , Padra, C. (1998). An error estimator for adaptive frictionless contact finite element analysis. Relatórios de Pesquisa e Desenvolvimento 11/98, LNCC, Rio de Janeiro.

Buscaglia, G., Feijóo, R. , Padra, C. (1995). A posteriori error estimation in sensitivity analysis. *Structural Optimization*, v. 9, n. 3/4, p. 194–199.

Canales, J., Tárrago, J. , Hernández, A. (1993). An adaptive mesh refinement procedure for shape optimal design. *Advances in Engineering Software*, v. 18, p. 131–145.

Cantor, M. (2001). *Software Leadership: A Guide to Sucessful Software Development*. Addison Wesley.

Cardoso, J., Arora, J. (1992). Design sensitivity analysis of nonlinear dynamic response of structural and mechanical systems. *Structural Optimization*, v. 4, p. 37–46.

Cardoso, J., Santos, J. (1993). Object oriented design sensitivity analysis and optimization. In: Herskovits, J., (Editor), *Proceedings of Structural Optimization 93 – The World Congress on Optimal Design of Structural Systems*, p. 423–430, Rio de Janeiro. ABCM and COPPE/UFRJ.

Cea, J. (1981). Numerical methods of shape optimal design. In: Haug, E., Cea, J., (Eds.), *Optimization of Distributed Parameter Structures*, p. 1049–1087. Sijthoff & Noordhoff, Alphen aan den Rijn.

Chang, K., Choi, K. (1992). An error analysis and mesh adaptation method for shape design of structural components. *Computers & Structures*, v. 44, n. 6, p. 1275–1289.

Chen, J., Han, W., Wu, C. , Duan, W. (1997a). On the perturbed lagrangian formulation for nearly incompressible and incompressible hyperelasticity. *Computer Methods in Applied Mechanics and Engineering*, v. 142, p. 335–351.

Chen, J., Pan, C. (1996). A pressure projection method for nearly incompressible rubber hyperelasticity, Part I: Theory. *Journal of Applied Mechanics*, v. 63, p. 862–868.

Chen, J., Pan, C. , Wu, C. (1997b). Large deformation analysis of rubber based on a reproducing kernel particle method. *Computational Mechanics*, v. 19, p. 211–227.

Chen, J., Wu, C. , Pan, C. (1996). A pressure projection method for nearly incompressible rubber hyperelasticity, part II: Applications. *Journal of Applied Mechanics*, v. 63, p. 869–876.

Chen, J., Yoon, S., Wang, H. , Liu, W. K. (2000). An improved reproducing kernel particle method for nearly incompressible finite elasticity. *Computer Methods in Applied Mechanics and Engineering*, v. 181, p. 117–145.

Chen, S., Tortorelli, D. A. (1997). Three-dimensional shape optimization with variational geometry. *Structural Optimization*, v. 13, p. 81–94.

Chilton, L., Suri, M. (2001). Locking free mixed *hp* finite element methods for curvilinear domains. *Computer Methods Applied to Mechanics and Engineering*, v. 190, p. 3427–3442.

Cho, S., Choi, K. (2000a). Design sensitivity analysis and optimization of non-linear transient dynamics. Part I – sizing design. *International Journal for Numerical Methods in Engineering*, v. 48, p. 351–373.

Cho, S., Choi, K. (2000b). Design sensitivity analysis and optimization of non-linear transient dynamics. Part II – configuration design. *International Journal for Numerical Methods in Engineering*, v. 48, p. 375–399.

Choi, K., Chang, K. (1994). A study of design velocity field computation for shape optimal design. *Finite Elements in Analysis and Design*, v. 15, n. 4, p. 317–341.

Choi, K., Duan, W. (2000). Design sensitivity analysis and shape optimization of structural components with hyperelastic material. *Computer Methods in Applied Mechanics and Engineering*, v. 187, n. 1-2, p. 219–243.

Choi, K., Haug, E. (1983). Shape design sensitivity analysis of elastic structures. *Journal of Structural Mechanics*, v. 11, p. 231–269.

Choi, K., Santos, J. (1987). Design sensitivity analisys for nonlinear structural systems part I: Theory. *International Journal for Numerical Methods in Engineering*, v. 24, n. 1, p. 2039–2055.

Choi, K., Seong, H. (1986). A domain method for shape design sensitivity analysis of built-up structures. *Computer Methods in Applied Mechanics and Engineering*, v. 57, n. 1, p. 1–15.

Choi, K., Twu, S. (1989). Equivalence of continuum and discrete methods of shape design sensitivity analysis. *AIAA Journal*, v. 27, n. 10, p. 1419–1424.

Corney, J. (1997). *3D Modeling with ACIS Kernel and Toolkit*. John Wiley and Sons, New York.

Cox, B. (1986). *Object-Oriented Programming – An Evolutionary Approach*. Addison-Wesley.

Dalheimer, M. (1999). *Programming with Qt*. OReilly & Associates, New York.

Devloo, P. (1997). PZ: An object oriented environment for scientific programming. *Computer Methods in Applied Mechanics and Engineering*, v. 150, p. 133–153.

Devloo, P., Longhin, G. (2002). Object oriented design philosophy for scientific computing. *Mathematical Modelling and Numerical Analysis*, v. 36, n. 5, p. 793–807.

Dias, G., Herskovits, J. , Rochinha, F. (1998). Simultaneous shape optimization and nonlinear analysis of elastic solids. In: I. Idelsohn, E. O., Dvorkin, E., (Eds.), *Computational Mechanics - New Trends and Applications*, p. 1–13, Barcelona. CIMNE.

Ding, Y. (1986). Shape optimization of structures: A literature survey. *Computers & Structures*, v. 24, n. 6, p. 985–1004.

Driemeier, L. (2002). Aplicação do conceito de derivada topológica na otimização estrutural de problemas de elasticidade. DPM/FEM/UNICAMP. Dissertação (Mestrado).

Dubois-Pèlerin, Y., Pegon, P. (1998). Object-oriented programming in nonlinear finite element analysis. *Computers & Structures*, v. 67, n. 4, p. 225–241.

Dubois-Pèlerin, Y., Zimmermann, T. (1993). Object-oriented finite element programming: III. An efficient implementation in C++. *Computer Methods in Applied Mechanics and Engineering*, v. 108, n. 1-2, p. 165–183.

Dufeu, E., Gómez, J., Cugnon, F. , Beckers, P. (1997). Control del error en la optimización de forma de estructuras bidimensionales. *Revista Internacional de Métodos Numéricos para Cálculo y Diseño y Ingeniería*, v. 13, n. 2, p. 165–183.

Eeles, P. (2001). Layering strategies. *The Rational Edge*. http://www.therationaledge.com.

Elwakeil, O., Arora, J. (1995). Methods for finding feasible points in constrained optimization. *AIAA Journal*, v. 33, n. 9, p. 1715–1719.

Evans, G. (2001). A simplified approach to RUP. *The Rational Edge*. http://www.therationaledge.com.

Evsukoff, A. (1992). Sobre a introdução de um algoritmo de ponto interior no ambiente de projeto de engenharia. COPPE/UFRJ, Rio de Janeiro. Dissertação (Mestrado).

Fancello, E. (1993). *Análise de sensibilidade, geração automática de malhas e o método dos elementos finitos na otimização de forma em problemas de contato e mecânica da fratura*. COPPE/UFRJ, Tese (Doutorado).

Fancello, E., Feijóo, R. (1992). ADAPTE: estimador de erro para problemas planos em elasticidade linear. Relatório de Pesquisa e Desenvolvimento 38/92, LNCC, Rio de Janeiro.

Fancello, E., Feijóo, R. (1994). Shape optimization in frictionless contact problems. *International Journal for Numerical Methods in Engineering*, v. 37, p. 2311–2335.

Fancello, E., Guimarães, A., Feijóo, R. , Venere, M. (1991). Geração automática de malhas 2d em programação orientada a objetos. In: *Anais do XI Congresso Brasileiro de Engenharia Mecânica*, p. 635–638, São Paulo. ABCM.

Fancello, E., Haslinger, J. , Feijóo, R. (1995). Numerical comparison between two cost functions in contact shape optimization. *Structural Optimization*, v. 9, n. 1, p. 57–68.

Feijóo, R., Guimarães, A. , Fancello, E. (1991). Algumas experiencias en la programación orientada por objetos y su aplicación en el método de los elementos finitos. Relatório de Pesquisa e Desenvolvimento 15/91, LNCC, Rio de Janeiro.

Fenves, G. (1990). Object-oriented programming for engineering software development. *Engineering with Computers*, v. 6, n. 1, p. 1–15.

Ferreira, W. (2002). Desenvolvimento de ferramentas computacionais para análise estrutural em fadiga e geração de malhas de elementos finitos. DPM/FEM/UNICAMP. Dissertação (Mestrado).

Filho, J., Devloo, P. (1991). Object oriented programming in scientific computations: the beginning of a new era. *Engineering Computations*, v. 8, p. 81–8.

Fleury, C. (1989). CONLIN: An efficient dual optimizer based on convex approximation concepts. *Structural Optimization*, v. 1, p. 81–89.

Forde, B., Foschi, R. , Stiemer, S. (1990). Object-oriented finite element analysis. *Computers & Structures*, v. 34, p. 355–374.

Fourment, L., Chenot, J. (1996a). Optimal design for non-steady-state metal forming processes - I: Shape optimization method. *International Journal for Numerical Methods in Engineering*, v. 39, p. 33–50.

Fourment, L., Chenot, J. (1996b). Optimal design for non-steady-state metal forming processes - II: Application of shape optimization in forging. *International Journal for Numerical Methods in Engineering*, v. 39, p. 51–65.

Fowler, M., Scott, K. (1997). *UML Distilled, Applying the Standard Object Modeling Language.* Addison-Wesley, Massachusetts.

Fuenmayor, F., Oliver, J. , Ródenas, J. (1997). Extension of the zienkiewicz-zhu error estimator to shape sensitivity analysis. *International Journal for Numerical Methods in Engineering*, v. 40, p. 1413–1433.

Gallimard, L., Ladevèze, P. , Pelle, J. (1996). Error estimation and adaptivity in elastoplasticity. *International Journal for Numerical Methods in Engineering*, v. 39, p. 189–217.

Gamma, E., Helm, R., Johnson, R. , Vlissides, J. (1995). *Design Patterns: Elements of Reusable Object-Oriented Software.* Addison-Wesley.

Grindeanu, I., Chang, K., Choi, K. , Chen, J. (1998). Design sensitivity analysis of hyperelastic structures using a meshless method. *AIAA Journal*, v. 36, n. 4, p. 618–627.

Guimarães, A., Feijóo, R. (1989). O sistema ACDP. Relatório de Pesquisa e Desenvolvimento 027/89, LNCC, Rio de Janeiro.

Gurtin, M. (1981). *An Introduction to Continuum Mechanics*, v. 158 de *Mathematics in Science and Engineering.* Academic Press.

Haftka, R., Adelman, H. (1989). Recent developments in structural sensitivity analysis. *Structural Optimization*, v. 1, n. 3, p. 137–151.

Haftka, R., Barthelemy, B. (1989). On the accuracy of shape sensitivity. In: Brebbia, C., Hernandez, S., (Eds.), *Computer Aided Optimum Design of Structures: Recent Advances*, p. 327–336, Southampton. Computational Mechanics Publications.

Haftka, R., Grandhi, R. (1986). Structural shape optimization — a survey. *Computer Methods in Applied Mechanics and Engineering*, v. 57, n. 1, p. 91–106.

Haney, S., Crotinger, J. (1999). How templates enable high-performance scientific computing in C++. *IEEE Computing in Science and Engineering*, v. 1, n. 4, p. 66–72.

Haslinger, J., Jedelský, D. (1996). Genetic algorithms and fictitious domain based approaches in shape optimization. *Structural Optimization*, v. 12, n. 4, p. 257–264.

Haug, E., Choi, K. , Komkov, V. (1986). *Design sensitivity analysis of structural systems*, v. 177 de *Mathematics in Science and Engineering*. Academic Press, Orlando.

Herskovits, J. (1986). A two-stage feasible directions algorithm for nonlinear constrained optimization. *Mathematical Programming*, v. 36, p. 19–38.

Herskovits, J., Coelho, C. (1989). An interior points algorithm for structural optimization problems. In: Brebbia, C., Hernandez, S., (Eds.), *Computer Aided Optimum Design of Structures: Recent Advances - Proceedings of the First International Conference, Southampton, June, 1989*, p. 231–241, Southampton. Computational Mechanics Publications - Springer Verlag.

Herskovits, J., Leontiev, A., Dias, G. , Santos, G. (1998). An interior point algorithm for optimal design of unilateral constrained mechanical systems. In: I. Idelsohn, E. O., Dvorkin, E., (Eds.), *Computational Mechanics - New Trends and Applications*, p. 1–17, Barcelona. CIMNE.

Herskovits, J., Santos, G. (1997). On the computer implementation of feasible direction interior point algorithms for nonlinear optimization. *Structural Optimization*, v. 14, p. 165–172.

Huang, M., Arora, J. (1997). Optimal design with discrete variables: some numerical experiments. *International Journal for Numerical Methods in Engineering*, v. 40, p. 165–188.

Hughes, T. J. R. (1980). Generalization of selective integration procedures to anisotropic and nonlinear media. *International Journal for Numerical Methods in Engineering*, v. 15, p. 1413–1418.

Humphrey, W. (2000). The personal software process (PSP). Technical Report CMU/SEI-2000-TR-022, Carnegie Mellon University, Pittsburgh, PA.

Hwang, H., Choi, K. , Chang, K. (1997). Second-order shape design sensitivity using p-version finite element analysis. *Structural Optimization*, v. 14, p. 91–99.

Jacobson, I., Booch, G. , Rumbaugh, J. (1999). *The Unified Software Development Process*. Addison Wesley, Reading (MA).

Jacobson, I., Christerson, M. , Constantine, L. (1994). The OOSE method: Use-case-driven approach. In: Carmichael, A., (Editor), *Object Development Methods*, p. 247–270. Sig Books, New York.

Karniadakis, G. E., Sherwin, S. J. (1999). *Spectral/hp Element Methods for CFD*. Oxford University Press, Oxford.

Kikuchi, N., Chung, K., Torigaki, T. , Taylor, J. (1986). Adaptive finite element methods for shape optimization of linearly elastic structures. *Computer Methods in Applied Mechanics and Engineering*, v. 57, n. 1, p. 67–89.

Kim, N., Choi, K. , Chen, J. (2001a). Die shape design optimization of sheet metal stamping process using meshfree method. *International Journal for Numerical Methods in Engineering*, v. 51, p. 1385–1405.

Kim, N., Park, Y. , Choi, K. (2001b). Optimization of a hyper-elastic structure with multibody contact using continuum-based shape design sensitivity analysis. *Structural and Multidisciplinaty Optimization*, v. 21, n. 3, p. 196–208.

Kim, N. H. (1999). *Shape Design Sensitivity Analysis and Optimization of Nonlinear Static/Dynamic Structures with Contact/Impact*. The University of Iowa, Tese (Doutorado).

Kleiber, M. (1993). Shape and non-shape structural sensitivity analisys for problems with material and kinematic non-linearity. *Computer Methods in Applied Mechanics and Engineering*, v. 108, n. 1, p. 73–97.

Kleiber, M., Kowalczyk, P. (1996). Sensitivity analysis in plane stress elasto-plasticity and elasto-viscoplasticity. *Computer Methods in Applied Mechanics and Engineering*, v. 137, p. 395–409.

Klein, B. (1955). Direct use of extremal principles in solving certain optimization problems involving inequalities. *Journal of the Operations Research Society of America*, v. 3, p. 168–175.

Kreyszig, E. (1989). *Introductory Functional Analysis with Applications*. John Wiley & Sons, New York.

Kruchten, P. (2001a). The nature of software: What's so special about software engineering. *The Rational Edge*. http://www.therationaledge.com.

Kruchten, P. (2001b). Using the RUP to evolve a legacy system. *The Rational Edge*. http://www.therationaledge.com.

Kruchten, P. (2001c). What is the Rational Unified Process? *The Rational Edge*. http://www.therationaledge.com.

Kruchten, P. (2002a). Agility with the RUP. *The Rational Edge*. http://www.therationaledge.com.

Kruchten, P. (2002b). A software development process for a team of one. *The Rational Edge*. http://www.therationaledge.com.

Lai, W., Rubin, D. , Krempl, E. (1993). *Introduction to Continuum Mechanics*. Butterworth-Heinemann, 3rd ed.

Langtangen, H. (1999). *Computational Partial Differential Equations: Numerical Methods and Diffpack Programming*. Springer, New York.

Larman, C. (1998). *Applying UML and Patterns*. Prentice Hall.

Larman, C. (2001). *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design*. Prentice Hall.

Lavi, A., Vogl, T., (Eds.) (1966). *Recent Advances in Optimization Techniques - Proceedings of the Symposium*, New York. IEEE and Optical Society of America, John Wiley & Sons.

Lee, H., Arora, J. (1991). Object-oriented programming for engineering applications. *Engineering with Computers*, v. 7, n. 4, p. 225–235.

Lee, T., Arora, J. (1995). A computational method for design sensitivity analysis of elastoplastic structures. *Computer Methods in Applied Mechanics and Engineering*, v. 122, p. 27–50.

Leigh, D. C. (1968). *Nonlinear Continuum Mechanics*. McGraw-Hill, New York.

Leinen, P. (1995). Data structures and concepts for adaptive finite element methods. *Computing*, v. 55, p. 325–354.

Lemaitre, J., Chaboche, J. (1990). *Mechanics of Solid Materials*. Cambridge University Press, Cambridge.

Levy, R., Lev, O. (1987). Recent developments in structural optimization. *Journal of Structural Engineering*, v. 113, n. 9, p. 1939–1962.

Liefooghe, D., Fleury, C. (1989). Interactive capability for shape optimal design. *Finite Elements in Analysis and Design*, v. 5, n. 1, p. 39–55.

Lim, O., Arora, J. (1986). An active set RQP algorithm for engineering design optimization. *Computer Methods in Applied Mechanics and Engineering*, v. 57, n. 1, p. 51–65.

Lindby, T., Santos, J. (1997). 2-d and 3-d shape optimization using mesh velocities to integrate analytical sensitivities with associative CAD. *Structural Optimization*, v. 13, p. 213–222.

Lippman, S. (1991). *C++ primer*. Addison-Wesley, Reading, 2nd ed.

Luenberger, D. (1989). *Linear and Nonlinear Programming*. Addison-Wesley, Massachusetts, 2nd ed.

Lund, E. (1994). *Finite element based design sensitivity analysis and optimization*. Aalborg University, Tese (Doutorado).

Mackerle, J. (2000). Object-oriented techniques in FEM and BEM: A bibliography (1996-1999). *Finite Elements in Analysis and Design*, v. 36, p. 189–196.

Mackie, R. (1992). Object oriented programming of the finite element method. *International Journal for Numerical Methods in Engineering*, v. 35, n. 2, p. 425–436.

Mackie, R. (1997). Using objects to handle complexity in finite element software. *Engineering with Computers*, v. 13, n. 2, p. 99–111.

Mackie, R. (2000). An object-oriented approach to calculation control in finite element programs. *Computers & Structures*, v. 77, p. 461–474.

Maniatty, A., Chen, M. (1996). Shape sensitivity analysis for steady metal-forming processes. *International Journal for Numerical Methods in Engineering*, v. 39, p. 1199–1217.

Marsden, J., Tromba, A. (1988). *Vector Calculus*. W.H. Freeman, New York.

Martin, J., Odell, J. (1998). *Object-Oriented Methods: A Foundation, UML Edition*. Prentice Hall, New Jersey.

Martins, J. C. C. (2002). *Gestão de Projetos de Desenvolvimento de Software (PMI - UML)*. Brasport, Rio de Janeiro.

Masmoudi, M., Broudiscou, C. , Guillaume, P. (1993). Automatic differentiation and shape optimization. In: Herskovits, J., (Editor), *Proceedings of Structural Optimization 93 – The World Congress on Optimal Design of Structural Systems*, p. 189–196, Rio de Janeiro. ABCM and COPPE/UFRJ.

Menétrey, P., Zimmermann, T. (1993). Object-oriented nonlinear finite element analysis: application to J2 plasticity. *Computers & Structures*, v. 49, n. 5, p. 767–777.

Michell, A. (1904). The limits of economy of material in frame structures. *Philosophical Magazine*, v. 8, n. 47, p. 589–595.

Miki, M. (1995). Object-oriented optimization of discrete structures. *AIAA Journal*, v. 33, n. 10, p. 1940–1945.

Miller, G. (1991). An object-oriented approach to structural analysis and design. *Computers & Structures*, v. 40, n. 1, p. 75–82.

Nogueira Jr., A. (2002). *Formulação* p *Do Método de Elementos Finitos Em Problemas de Elasticidade Linear e Não-Linear Com Malhas 3D Não-Estruturadas e Em Métodos Multigrid Algébricos*. Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas, Tese (Doutorado).

Oden, J. T. (1972). *Finite Elements of Nonlinear Continua*. Advanced Engineering Series. McGraw-Hill, New York.

Ogden, R. (1984). *Non-Linear Elastic Deformations*. Ellis Horwood Limited, Chichester.

Oldham, J. (2002). *POOMA: A C++ Toolkit for High-Performance Parallel Scientific Computing*. CodeSourcery. http://www.codesourcery.com/.

Oñate, E., Idelsohn, I. , Dvorkin, E., (Eds.) (1998). *Computational Mechanics – New Trends and Applications*, Barcelona. CIMNE.

Özakça, M., Hinton, E. , Rao, N. (1993). Shape optimization of axisymmetric structures with adaptive finite element procedures. *Structural Optimization*, v. 5, p. 256–264.

Panier, E., Tits, A. , Herskovits, J. (1988). A qp-free, globally convergent, locally superlinear convergent algorithm for inequality constrained optimization. *SIAM Journal of Control and Optimization*, v. 26, n. 4, p. 788–810.

Pantuso, D., Bathe, K. (1997). On the stability of mixed finite elements in large strain analysis of incompressible solids. *Finite Elements in Analysis and Design*, v. 28, n. 2, p. 83–104.

Park, Y., Choi, K. (1996a). Configuration design sensitivity analysis of nonlinear structural systems with elastic material. *Mechanics in Structures and Machines*, v. 24, n. 2, p. 217–255.

Park, Y., Choi, K. (1996b). Design sensitivity analysis of truss structures with elastoplastic material. *Mechanics in Structures and Machines*, v. 24, n. 2, p. 189–216.

Phelan, D., Vidal, C. , Haber, R. (1989). Explicit sensitivity analysis of nonlinear elastic systems. In: Brebbia, C., Hernandez, S., (Eds.), *Computer Aided Optimum Design of Structures: Recent Advances*, p. 357–366, Southampton. Computational Mechanics Publications.

Pironneau, O. (1984). *Optimal Shape Design for Elliptic Systems*. Springer Series in Computational Physics. Springer-Verlag, New York.

Probasco, L. (2000). The ten essentials of RUP. *The Rational Edge*. http://www.therationaledge.com.

Project Management Institute (2000). *A Guide to the Project Management Body of Knowledge, A PMBOK Guide*. Project Management Institute. www.pmi.org.

Rajan, S., Belegundu, A. (1989). Shape optimal design using fictitious loads. *AIAA Journal*, v. 27, n. 1, p. 102–107.

Rational Software Corporation Inc. (2000). *Rational Unified Proccess – Online Documentation (http://www.rational.com/products/rup/resource_center/index.jsp)*.

Reddy, B. (1986). *Functional Analysis and Boundary-Value Problems: An Introductory Treatment*. Longman Scientific & Technical, Essex.

Rogers, D., Adams, J. (1990). *Mathematical Elements for Computer Graphics*. McGraw-Hill, 2nd ed.

Rucki, M., Miller, G. (1996). An algorithmic framework for flexible finite element-based structural modeling. *Computer Methods in Applied Mechanics and Engineering*, v. 136, n. 3-4, p. 363–384.

Rucki, M., Miller, G. (1998). An adaptable finite element modelling kernel. *Computers & Structures*, v. 69, n. 3, p. 399–409.

Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F. , Lorensen, W. (1991). *Object-Oriented Modeling and Design*. Prentice Hall, New Jersey.

Rumbaugh, J., Jacobson, I. , Booch, G. (1999). *The Unified Modeling Language Reference Manual*. Addison Wesley, Reading (MA).

Ryu, Y., Haririan, M., Wu, C. , Arora, J. (1985). Structural design sensitivity analysis of nonlinear response. *Computers & Structures*, v. 21, n. 1-2, p. 245–255.

Sampath, R., Zabaras, N. (2000). An object-oriented framework for the implementation of adjoint techniques in the design and control of complex continuum systems. *International Journal for Numerical Methods in Engineering*, v. 48, p. 239–266.

Santos, J., Choi, K. (1988). Sizing design sensitivity analysis of nonlinear structural systems part II: numerical method. *International Journal for Numerical Methods in Engineering*, v. 26, p. 2097–2114.

Santos, J., Choi, K. (1992). Shape design sensitivity analysis of nonlinear structural systems. *Structural Optimization*, v. 4, n. 1, p. 23–35.

Schmit, L. (1960). Structural design by systematic synthesis. In: *Proceedings of de 2nd Conference on Eletronic Computation*, p. 105–122, New York. ASCE.

Schmit, L. (1981). Structural synthesis - its genesis and development. *AIAA Journal*, v. 19, n. 10, p. 1249–1263.

Scholz, S. (1992). Elements of an object-oriented fem++ program in C++. *Computers & Structures*, v. 43, n. 3, p. 517–529.

Seong, H., Choi, K. (1987). Boundary-layer approach to shape design sensitivity analysis. *Mechanics of Structures and Machines*, v. 15, n. 2, p. 241–263.

Silva, C. (1997). Otimização estrutural e análise de sensibilidade orientadas por objetos. DPM/FEM/UNICAMP, Campinas. Dissertação (Mestrado).

Silva, C., Bittencourt, M. (1997). Análise de sensibilidade e otimização estrutural em problemas elastostáticos lineares. In: *Proceedings of 14th Brazilian Congress of Mechanical Engineering*, Bauru. ABCM.

Silva, C., Bittencourt, M. (1998). An interactive object-oriented tool for structural optimization. In: I. Idelsohn, E. O., Dvorkin, E., (Eds.), *Computational Mechanics - New Trends and Applications*, p. 1–16, Barcelona. CIMNE.

Silva, C., Bittencourt, M. (1999a). Aspects of three-dimensional structural shape optimization. In: *Proceedings of 15th Brazilian Congress of Mechanical Engineering*, p. 1–10, Águas de Lindóia. ABCM.

Silva, C., Bittencourt, M. (1999b). Expressões de análise de sensibilidade em problemas elásticos. *Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería*, v. 5, n. 2, p. 243–268.

Silva, C., Bittencourt, M. (2000). An object-oriented structural optimization program. *Structural and Multidisciplinary Optimization*, v. 20, n. 2, p. 154–166.

Simo, J., Hughes, T. (1998). *Computational Inelasticity*. Springer, New York.

Smith, D., Tortorelli, D. , Tucker, C. (1998a). Analysis and sensitivity analysis for polymer injection and compression molding. *Computer Methods in Applied Mechanics and Engineering*, v. 167, p. 325–344.

Smith, D., Tortorelli, D. , Tucker, C. (1998b). Optimal design for polymer extrusion. Part I: Sensitivity analysis for nonlinear steady-state systems. *Computer Methods in Applied Mechanics and Engineering*, v. 167, p. 283–302.

Smith, D., Tortorelli, D. , Tucker, C. (1998c). Optimal design for polymer extrusion. Part II: Sensitivity analysis for weakly-coupled nonlinear steady-state systems. *Computer Methods in Applied Mechanics and Engineering*, v. 167, p. 303–323.

Sotelino, E., Chen, W. , White, D. (1998). Future challenges for simulation in structural engineering. In: I. Idelsohn, E. O., Dvorkin, E., (Eds.), *Computational Mechanics - New Trends and Applications*, p. 1–18, Barcelona. CIMNE.

Spatial Corp. (2001). *Online Help for ACIS, Version 6.3.*

Sussman, T., Bathe, K. (1987). A finite element formulation for nonlinear incompressible elastic and inelastic analysis. *Computers & Structures*, v. 26, n. 1/2, p. 357–409.

Taroco, E. (2000). Shape sensitivity analysis in linear elastic fracture mechanics. *Computer Methods in Applied Mechanics and Engineering*, v. 188, n. 1, p. 697–712.

Taroco, E., Buscaglia, G. , Feijóo, R. (1998). Second-order shape sensitivity analysis for nonlinear problems. *Structural Optimization*, v. 15, n. 2, p. 101–113.

Tiller, M., Dantzig, J. (1996). Implementation of design sensitivity analysis and numerical optimization in engineering analysis. *Applied Mathematical Modelling*, v. 20, p. 792–799.

Timoshenko, S. (1953). *History of Strength of Materials*. McGraw-Hill, New York.

Tortorelli, D. (1992). Sensitivity analysis for nonlinear constrained elastostatic systems. *International Journal for Numerical Methods in Engineering*, v. 33, p. 1643–1660.

Troll Tech AS. (2001). *Qt Online Reference Documentation, Version 2.3.0*.

Tsay, J., Arora, J. (1989). Optimum design of nonlinear structures with path dependent response. *Structural Optimization*, v. 1, p. 203–213.

Tsay, J., Arora, J. (1990). Nonlinear structural design sensitivity analysis for path dependent problems. part 1: General theory. *Computer Methods in Applied Mechanics and Engineering*, v. 81, n. 2, p. 183–208.

Tsay, J., Cardoso, J. , Arora, J. (1990). Nonlinear structural design sensitivity analysis for path dependent problems. part 2: Analytical examples. *Computer Methods in Applied Mechanics and Engineering*, v. 81, n. 2, p. 209–228.

Tseng, C., Arora, J. (1989). Numerical verification of design sensitivity analysis. *AIAA Journal*, v. 27, n. 1, p. 117–119.

Twu, S., Choi, K. (1992). Configuration design sensitivity analysis of built-up structures part I: Theory. *International Journal for Numerical Methods in Engineering*, v. 35, n. 5, p. 1127–1150.

Vanderplaats, G. (1982). Structural optimization - past, present and future. *AIAA Journal*, v. 20, n. 7, p. 992–1000.

Vanderplaats, G. (1993). Thirty years of modern structural optimization. *Advances in Engineering Software*, v. 16, n. 2, p. 81–88.

Vanderplaats, G., Sugimoto, H. (1986). A general-purpose optimization program for engineering design. *Computers & Structures*, v. 24, n. 1, p. 13–21.

Veldhuizen, T. (2000). Techniques for scientific C++. Computer Science Technical Report 542, Indiana University.

Vidal, C., Haber, R. (1993). Design sensitivity analysis of rate-independent elastoplasticity. *Computer Methods in Applied Mechanics and Engineering*, v. 107, p. 393–431.

Wang, S., Sun, Y. , Gallagher, R. (1985). Sensitivity analysis in shape optimization of continuum structures. *Computers & Structures*, v. 20, n. 5, p. 855–867.

Yang, R., Botkin, M. (1987). Accuracy of the domain material derivative approach to shape design sensitivities. *AIAA Journal*, v. 25, n. 12, p. 1606–1610.

Yang, R., Choi, K. (1985). Accuracy of finite element based shape design sensitivity analysis. *Journal of Structural Mechanics*, v. 13, n. 2, p. 223–239.

Yao, T., Choi, K. (1989). 3D shape optimal design and automatic finite element regridding. *International Journal for Numerical Methods in Engineering*, v. 28, n. 2, p. 369–384.

Yatheendhar, M., Belegundu, A. (1993). Analytical shape sensitivity by implicit differentiation for general velocity fields. *Computers & Structures*, v. 46, n. 4, p. 617–623.

Younsi, R., Knof-Lenoir, C. , Selman, A. (1996). Multi-mesh and adaptivity in 3D shape optmization. *Computers & Structures*, v. 61, n. 6, p. 1125–1133.

Yu, G., Adeli, H. (1993). Object-oriented finite element analysis using EER model. *Journal of Structural Engineering*, v. 119, n. 9, p. 2763–2781.

Yu, L., Kumar, A. (2001). An object-oriented modular framework for implementing the finite element method. *Computers & Structures*, v. 79, p. 919–928.

Zeglinski, G., Han, R. , Aitchison, P. (1994). Object oriented matrix classes for use in a finite element code using C++. *International Journal for Numerical Methods in Engineering*, v. 37, n. 22, p. 3921–3937.

Zhang, W., Beckers, P. (1989). Comparison of different sensitivity analysis approaches for structural shape optimization. In: Brebbia, C. A., Hernandez, S., (Eds.), *Computer Aided Optimum Design of Structures: Recent Advances*, p. 347–356, Southampton. Computational Mechanics Publications.

Zhao, G., Wright, E. , Grandhi, R. V. (1997). Preform die shape design in metal forming using an optimization method. *International Journal for Numerical Methods in Engineering*, v. 40, p. 1213–1230.

Zhao, Z. (1991). *Shape Design Sensitivity Analysis and Optimization Using the Boundary Element Method*, v. 62 de *Lecture Notes in Engineering*. Springer-Verlag, Berlin.

Zienkiewicz, O., Campbell, J. (1973). Shape optimization and sequential linear programming. In: Gallagher, R., Zienkiewicz, O., (Eds.), *Optimum Structural Design - Theory and Applications*, p. 109–125, New York. John Wiley.

Zienkiewicz, O., Zhu, J. (1987). A simple error estimator and adaptive procedure for practical engineering analysis. *International Journal for Numerical Methods in Engineering*, v. 24, p. 337–359.

Zimmermann, T., Dubois-Pèlerin, Y. , Bomme, P. (1992). Object-oriented finite element programming: I. govening principles. *Computer Methods in Applied Mechanics and Engineering*, v. 98, n. 2, p. 291–303.

Zolésio, J. (1981). The material derivative (or speed) method for shape optimization. In: Haug, E., Cea, J., (Eds.), *Optimization of Distributed Parameter Structures*. Sijthoff & Noordhoff, Alphen aan den Rijn.

Apêndice A

Formulação Básica de Corpos Deformáveis

Este apêndice apresenta os conceitos básicos de corpos deformáveis e está baseado em (Bonet e Wood, 1997; Gurtin, 1981; Lai *et al.*, 1993; Leigh, 1968; Lemaitre e Chaboche, 1990; Ogden, 1984).

A.1 Definições Iniciais

Na Mecânica dos Meios Contínuos, supõe-se que o espaço no qual ocorrem os fenômenos físicos seja o espaço Euclidiano pontual tridimensional $\mathcal{E}$ e que os instantes em que esses fenômenos são observados sejam medidos na reta real $\Re$. Para um observador[1] O, um evento está associado a pares $(\mathbf{x}, t) \in \mathcal{E} \times \Re$. Quando o observador O arbitra uma origem $\mathbf{o}$, os pontos podem ser interpretados através de vetores posição[2] do espaço $\Re^3$.

Formalmente, um corpo é identificado por uma região $\mathcal{B}$ regular de $\mathcal{E}$. Apesar de não haver nenhuma relação intrínseca entre o corpo e a região $\mathcal{B}$, pois os corpos ocupam diferentes regiões no decorrer do tempo, é conveniente adotar um estado de referência inicial para a descrição dos fenômenos que o corpo pode estar sujeito. Dessa forma, a região $\mathcal{B}$ é denominada *configuração de referência* ou

[1] Para descrever quantitativamente e qualitativamente os fenômenos físicos é necessário definir a existência das capacidades e meios de medição de grandezas físicas, particularmente, posições relativas de pontos no espaço e o progresso do tempo. A noção de *observador* representa esse conjunto de capacidades e meios.

[2] Logo, dado um ponto $\mathbf{X} \in \mathcal{E}$, $\mathbf{X} - \mathbf{o}$ define um vetor que também pode, num conveniente abuso de notação, ser denotado por $\mathbf{X}$.

material. Os pontos $\mathbf{X} \in \mathcal{B}$ são chamados *pontos materiais*, pois indicam a posição das partículas compondo o corpo material. Subregiões regulares limitadas de $\mathcal{B}$ são denominadas *partes* e denotadas por $\mathcal{P}$. Uma vez compreendida essa associação geométrica arbitrária entre o corpo material e sua posição no espaço, pode-se denominar o corpo simplesmente por $\mathcal{B}$.

A diferença entre pontos $\mathbf{X}$ e $\mathbf{Y}$ quaisquer do espaço Euclidiano $\mathcal{E}$ define um vetor $\mathbf{v}$, ou seja,

$$\mathbf{v} = \mathbf{X} - \mathbf{Y}.$$

O conjunto de todos os vetores define o espaço vetorial $\mathcal{F}$ associado a $\mathcal{E}$.

A.2 Tensores

Um tensor $\mathbf{A}$ é definido como uma transformação linear de $\mathcal{F}$ em $\mathcal{F}$, ou seja,

$$\begin{aligned} \mathbf{A}: \;\; & \mathcal{F} \rightarrow \mathcal{F} \\ & \mathbf{u} \rightarrow \mathbf{v} \quad = \mathbf{A}\mathbf{u}, \end{aligned}$$

satisfazendo

$$\mathbf{A}\left(\alpha \mathbf{u} + \beta \mathbf{v}\right) = \alpha \left(\mathbf{A}\mathbf{u}\right) + \beta \left(\mathbf{A}\mathbf{v}\right), \qquad \forall \mathbf{u}, \mathbf{v} \in \mathcal{F}, \quad \alpha, \beta \in \Re.$$

A definição anterior está relacionada a tensores de segunda ordem. Escalares são tensores de ordem zero, enquanto vetores são tensores de primeira ordem. De forma análoga, definem-se tensores de ordem mais alta. Por exemplo, tensores de terceira ordem são transformações lineares que mapeiam vetores em tensores de segunda ordem ou tensores de segunda ordem em vetores[3].

As equações constitutivas de material são, em geral, representadas por tensores de quarta ordem. Por sua vez, esses são definidos como mapeamentos lineares de vetores em tensores de terceira ordem, tensores de segunda ordem em tensores de segunda ordem ou tensores de terceira ordem em vetores. Logo, para um tensor C de quarta ordem tem-se

$$\mathsf{C}\mathbf{u} \;\; = \;\; A$$

[3] Podem-se denominar *escalares* como *tensores de ordem zero*, *vetores* como *tensores de primeira ordem* e os *tensores usuais* como *tensores de segunda ordem* devido ao número de índices que cada entidade possui. Esse tipo de denominação pode ser generalizada para tensores de dimensões maiores. No decorrer desse texto será utilizada a denominação mais comum (escalares, vetores e tensores), aplicando-se a denominação alternativa apenas quando for indispensável enfatizar a ordem da entidade.

$$
\begin{aligned}
\mathsf{D}\mathbf{T} &= \mathbf{S} \\
\mathsf{E}B &= \mathbf{v}
\end{aligned}
$$

sendo $\mathbf{u}, \mathbf{v}$ vetores, $\mathbf{T}, \mathbf{S}$ tensores de segunda ordem, A, B tensores de terceira ordem.

A.3 Produtos Tensorial e Interno

- **Produto tensorial entre vetores.** O produto tensorial $\mathbf{a} \otimes \mathbf{b}$ entre dois vetores $\mathbf{a}$ e $\mathbf{b}$ é o *tensor* que associa a cada vetor $\mathbf{v}$ o vetor $(\mathbf{b} \cdot \mathbf{v})\,\mathbf{a}$ (Gurtin, 1981), ou seja,

$$(\mathbf{a} \otimes \mathbf{b})\,\mathbf{v} = (\mathbf{b} \cdot \mathbf{v})\,\mathbf{a}.$$

De outra forma,

$$(\mathbf{a} \otimes \mathbf{b})\,\mathbf{v} = (\mathbf{b} \cdot \mathbf{v})\,\mathbf{a} = (b_j v_j)\,a_i = (a_i b_j)\,v_j = \left(\mathbf{a}\mathbf{b}^T\right)\mathbf{v}.$$

Portanto,

$$(\mathbf{a} \otimes \mathbf{b})_{ij} = a_i b_j \qquad \text{ou} \qquad \mathbf{a} \otimes \mathbf{b} = \mathbf{a}\mathbf{b}^T.$$

- **Produto interno entre tensores de quarta ordem e segunda ordem.** O produto $\mathsf{A} : \mathbf{B}$ entre o tensor de quarta ordem A e o tensor de segunda ordem $\mathbf{B}$ é o tensor de segunda ordem $\mathbf{C}$,

$$\mathbf{C} = \mathsf{A} : \mathbf{B}, \tag{A.1}$$

definido pela relação

$$C_{ij} = A_{ijkl} B_{kl}.$$

- **Produto tensorial entre tensores de segunda ordem.** O produto tensorial $\mathbf{A} \otimes \mathbf{B}$ entre dois tensores de segunda ordem $\mathbf{A}$ e $\mathbf{B}$ é o *tensor de quarta ordem* que associa a cada tensor de segunda ordem $\mathbf{V}$ o tensor $(\mathbf{B} \cdot \mathbf{V})\,\mathbf{A}$, ou seja,

$$(\mathbf{A} \otimes \mathbf{B})\,\mathbf{V} = (\mathbf{B} \cdot \mathbf{V})\,\mathbf{A}.$$

De outra forma,

$$(\mathbf{A} \otimes \mathbf{B})\,\mathbf{V} = (\mathbf{B} \cdot \mathbf{V})\,\mathbf{A} = \operatorname{tr}\left(\mathbf{V}^T\mathbf{B}\right)\mathbf{A} = (V_{kl} B_{kl})\,A_{ij} = (A_{ij} B_{kl})\,V_{kl}.$$

Portanto,

$$(\mathbf{A} \otimes \mathbf{B})_{ijkl} = A_{ij} B_{kl}.$$

Utilizando as operações anteriores, pode-se definir o produto $\mathbf{I} \otimes \mathbf{I}$ como

$$\mathbf{I} \otimes \mathbf{I} = (\delta_{ij}\delta_{kl})\, \mathbf{e}_i \otimes \mathbf{e}_j \otimes \mathbf{e}_k \otimes \mathbf{e}_l, \tag{A.2}$$

sendo

$$\begin{aligned} (\mathbf{I} \otimes \mathbf{I})_{ijkl} &= 1 \qquad \text{se} \qquad i = j \text{ e } k = l, \\ (\mathbf{I} \otimes \mathbf{I})_{ijkl} &= 0 \qquad \text{caso contrário.} \end{aligned}$$

Assim, para um tensor $\mathbf{A}$, tem-se que

$$(\mathbf{I} \otimes \mathbf{I}) : \mathbf{A} = \operatorname{tr}(\mathbf{A})\, \mathbf{I}. \tag{A.3}$$

A partir de (A.2), tem-se que

$$(\mathbf{I} \otimes \mathbf{I})^T = (\delta_{il}\delta_{jk})\, \mathbf{e}_i \otimes \mathbf{e}_j \otimes \mathbf{e}_k \otimes \mathbf{e}_l, \tag{A.4}$$

Define-se o *tensor identidade simétrico de quarta ordem* I como

$$\mathsf{I} = \frac{1}{2}\left[(\mathbf{I} \otimes \mathbf{I}) + (\mathbf{I} \otimes \mathbf{I})^T\right] = \frac{1}{2}(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk})\, \mathbf{e}_i \otimes \mathbf{e}_j \otimes \mathbf{e}_k \otimes \mathbf{e}_l. \tag{A.5}$$

Para um tensor de segunda ordem $\mathbf{A}$, tem-se que

$$\mathsf{I} : \mathbf{A} = \mathbf{A}. \tag{A.6}$$

A.4 Mudança de Configuração Espacial. Deformação

Mudanças na configuração espacial ou deformação dos corpos, como exemplificado na Figura A.1, são descritas matematicamente por uma transformação $\mathbf{f}$ um-a-um (para que o corpo não penetre em si mesmo) atuando nos pontos materiais, ou seja,

$$\boxed{\mathbf{x} = \mathbf{f}(\mathbf{X}).} \tag{A.7}$$

Denomina-se $\mathbf{x}$ como *ponto espacial*, pois o mesmo indica a posição no espaço ocupado pelo ponto material $\mathbf{X}$ após a mudança de configuração.

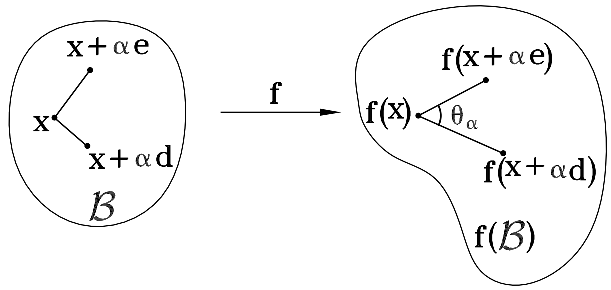


Figura A.1: Mudança de configuração espacial $\mathbf{f}$ aplicada ao corpo $\mathcal{B}$.

Verifica-se que $\det \nabla \mathbf{f}$ representa localmente o volume depois da alteração da configuração espacial por unidade de volume original. Portanto, é razoável assumir que $\det \nabla \mathbf{f} > 0$.

Em resumo, uma *mudança de configuração espacial* de $\mathcal{B}$ é uma transformação $\mathbf{f}$ um-a-um que leva $\mathcal{B}$ a uma região fechada de $\mathcal{E}$ com $\det \nabla \mathbf{f} > 0$.

O vetor

$$\boxed{\mathbf{u}(\mathbf{X}) = \mathbf{f}(\mathbf{X}) - \mathbf{X}} \tag{A.8}$$

representa o *deslocamento* do ponto material $\mathbf{X}$.

O tensor

$$\mathbf{F}(\mathbf{X}) = \nabla \mathbf{f}(\mathbf{X}) \tag{A.9}$$

é o gradiente da mudança de configuração em relação a $\mathbf{X}$ e pertence a Lin^+, conjunto dos tensores com determinante positivo. Como $\mathbf{f}(\mathbf{X}) = \mathbf{X} + \mathbf{u}(\mathbf{X})$, esse tensor também pode ser escrito na forma

$$\mathbf{F}(\mathbf{X}) = \mathbf{I} + \nabla \mathbf{u}(\mathbf{X}), \tag{A.10}$$

sendo $\nabla \mathbf{u}$ o tensor gradiente do campo de deslocamentos.

Se $\mathbf{F}$ é constante, $\mathbf{f}$ é uma mudança de configuração *homogênea*. Se $\mathbf{f}$ é homogênea, sua expansão em série de Taylor, em torno de um ponto arbitrário $\mathbf{Y}$, descreve toda transformação, como se observa na expressão

$$\mathbf{f}(\mathbf{X}) = \mathbf{f}(\mathbf{Y}) + \mathbf{F}(\mathbf{X} - \mathbf{Y}). \tag{A.11}$$

Observa-se que, para $\mathbf{f}$ e $\mathbf{F}$ é simétrico positivo definido, tem-se um alongamento a partir do ponto $\mathbf{Y}$; se $\mathbf{F}$ é ortogonal, tem-se uma rotação em torno de $\mathbf{Y}$.

Observa-se que na vizinhança de um ponto arbitrário $\mathbf{Y}$ e dentro de um erro $o(\mathbf{X}-\mathbf{Y})$, qualquer mudança de configuração se comporta como uma mudança de configuração homogênea, ou seja,

$$\mathbf{f}(\mathbf{X}) = \mathbf{f}(\mathbf{Y}) + \mathbf{F}(\mathbf{Y})(\mathbf{X}-\mathbf{Y}) + o(\mathbf{X}-\mathbf{Y}). \tag{A.12}$$

Logo, para $\mathbf{f}$ genérica, $\mathbf{F}$ contém toda a informação de primeira ordem acerca da mudança de configuração do corpo em uma vizinhança de $\mathbf{Y}$. No limite, descreve como o próprio ponto $\mathbf{Y}$ está se comportando.

A.5 Medidas de Deformação

A.5.1 Tensores de Deformação de Cauchy-Green

Teorema A.1 ***Decomposição Polar*** (Gurtin, 1981).

Seja $\mathbf{F} \in \mathrm{Lin}^{+}$. *Portanto, existem tensores simétricos positivo-definidos* $\mathbf{U}$, $\mathbf{V}$ *e uma rotação* $\mathbf{R}$ *tais que*

$$\boxed{\mathbf{F} = \mathbf{R}\mathbf{U} = \mathbf{V}\mathbf{R}.} \tag{A.13}$$

Tais decomposições são únicas. De fato,

$$\mathbf{U} = \sqrt{\mathbf{F}^T\mathbf{F}}, \qquad \mathbf{V} = \sqrt{\mathbf{F}\mathbf{F}^T}. \tag{A.14}$$

A representação $\mathbf{F} = \mathbf{R}\mathbf{U}$ *é chamada de representação polar direita de* $\mathbf{F}$ *e* $\mathbf{F} = \mathbf{V}\mathbf{R}$ *a representação polar esquerda.*

A expressão (A.12) e o teorema anterior sugerem a decomposição polar pontual

$$\mathbf{F}(\mathbf{X}) = \mathbf{R}(\mathbf{X})\,\mathbf{U}(\mathbf{X}) = \mathbf{V}(\mathbf{X})\,\mathbf{R}(\mathbf{X}),$$

obtendo-se campos tensoriais definidos em $\mathcal{B}$ com a seguinte interpretação: $\mathbf{R}(\mathbf{X})$ mede a rotação rígida local dos pontos em torno de $\mathbf{X}$; $\mathbf{U}(\mathbf{X})$ e $\mathbf{V}(\mathbf{X})$ medem o alongamento local a partir de $\mathbf{X}$, sendo chamados de tensores de alongamento direito e esquerdo.

Como o cálculo de $\mathbf{U}$ e $\mathbf{V}$ é difícil, pois envolvem a raiz quadrada de tensores, introduzem-se os *tensores* (e os respectivos campos tensoriais simétricos) *de deformação direito e esquerdo de Cauchy-*

Green **C** e **B**,

$$\mathbf{C} = \mathbf{U}^2 = \mathbf{F}^T\mathbf{F}, \qquad \mathbf{B} = \mathbf{V}^2 = \mathbf{F}\mathbf{F}^T. \tag{A.15}$$

Denotados em notação indicial, tem-se

$$C_{ij} = \frac{\partial f_m}{\partial X_i}\frac{\partial f_m}{\partial X_j}, \qquad B_{ij} = \frac{\partial f_i}{\partial X_m}\frac{\partial f_j}{\partial X_m}.$$

Observa-se também que

$$\mathbf{V} = \mathbf{R}\mathbf{U}\mathbf{R}^T, \qquad \mathbf{B} = \mathbf{R}\mathbf{C}\mathbf{R}^T.$$

A partir de (A.10), os campos **C** e **B** podem ser definidos em termos do campo de deslocamentos, ou seja,

$$\begin{aligned} \mathbf{C} &= \mathbf{I} + \nabla\mathbf{u} + \nabla\mathbf{u}^T + \nabla\mathbf{u}^T\nabla\mathbf{u}, \\ \mathbf{B} &= \mathbf{I} + \nabla\mathbf{u} + \nabla\mathbf{u}^T + \nabla\mathbf{u}\nabla\mathbf{u}^T. \end{aligned} \tag{A.16}$$

Em notação indicial, tem-se

$$\begin{aligned} C_{ij} &= \delta_{ij} + \left(\frac{\partial u_i}{\partial X_j} + \frac{\partial u_j}{\partial X_i}\right) + \frac{\partial u_m}{\partial X_i}\frac{\partial u_m}{\partial X_j}, \\ B_{ij} &= \delta_{ij} + \left(\frac{\partial u_i}{\partial X_j} + \frac{\partial u_j}{\partial X_i}\right) + \frac{\partial u_i}{\partial X_m}\frac{\partial u_j}{\partial X_m}. \end{aligned}$$

Para alterações de configuração rígidas (translações e/ou rotações) $\mathbf{U} = \mathbf{V} = \mathbf{C} = \mathbf{B} = \mathbf{I}$, pois não ocorre deformação. Nesse caso,

$$\nabla\mathbf{u} + \nabla\mathbf{u}^T + \nabla\mathbf{u}\nabla\mathbf{u}^T = \mathbf{0}. \tag{A.17}$$

A.5.2 Tensor Lagrangeano de Deformação Finita (ou de Green)

O campo tensorial

$$\boxed{\mathbf{E} = \frac{1}{2}\left(\mathbf{C} - \mathbf{I}\right)} \tag{A.18}$$

é chamado de campo Lagrangeano de deformação finita. Quando avaliado em um ponto **X**, determina o tensor Lagrangeano de deformação finita $\mathbf{E}(\mathbf{X})$. Se não há deformação, $\mathbf{C} = \mathbf{I} \rightarrow \mathbf{E} = \mathbf{0}$.

Em termos do campo de deslocamentos $\mathbf{u}$, o campo Lagrangeano de deformação assume a forma

$$\mathbf{E} = \frac{1}{2}\left(\nabla\mathbf{u} + \nabla\mathbf{u}^T\right) + \frac{1}{2}\nabla\mathbf{u}^T\nabla\mathbf{u}, \tag{A.19}$$

a qual é escrita em notação indicial como

$$E_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial X_j} + \frac{\partial u_j}{\partial X_i}\right) + \frac{1}{2}\frac{\partial u_m}{\partial X_i}\frac{\partial u_m}{\partial X_j}.$$

Esse tensor é qualificado de Lagrangeano, pois traduz as mudanças de comprimento e ângulo em relação ao comprimento e ângulos iniciais (Gurtin, 1981; Lai *et al.*, 1993).

A.5.3 Tensor de Deformação Infinitesimal

O campo tensorial

$$\boxed{\boldsymbol{\epsilon} = \frac{1}{2}\left(\nabla\mathbf{u} + \nabla\mathbf{u}^T\right)} \tag{A.20}$$

é chamado de campo de deformação infinitesimal. Em notação indicial, escreve-se

$$\epsilon_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial X_j} + \frac{\partial u_j}{\partial X_i}\right).$$

A partir de (A.16) observa–se que,

$$\begin{aligned}
\mathbf{C} &= \mathbf{I} + 2\boldsymbol{\epsilon} + \nabla\mathbf{u}^T\nabla\mathbf{u}, \\
\mathbf{B} &= \mathbf{I} + 2\boldsymbol{\epsilon} + \nabla\mathbf{u}\nabla\mathbf{u}^T.
\end{aligned}$$

O conceito de deformação infinitesimal pode ser compreendido a partir de uma família de alterações de configuração espaciais $\mathbf{f}_\varepsilon$ $(0 < \varepsilon < \varepsilon_0)$ com $\|\nabla\mathbf{u}_\varepsilon\| = \varepsilon$. Fazendo-se o limite $\varepsilon \to 0$, ou em outras palavras, considerando campos de deslocamentos com gradientes de módulo sucessivamente menores, observa-se que,

$$\boldsymbol{\epsilon}_\varepsilon = \frac{1}{2}\left(\mathbf{C}_\varepsilon - \mathbf{I}_\varepsilon\right) + o\left(\varepsilon\right) = \frac{1}{2}\left(\mathbf{B}_\varepsilon - \mathbf{I}_\varepsilon\right) + o\left(\varepsilon\right).$$

Em resumo, dentro de um erro $o\left(\|\nabla\mathbf{u}\|\right)$, o campo de deformação finita $\mathbf{E}$ se reduz ao campo infinitesimal $\boldsymbol{\epsilon}$ e os campos $\mathbf{C}$ e $\mathbf{B}$ coincidem.

A.6 Movimentos

Seja $\mathcal{B}$ um corpo. Um *movimento* de $\mathcal{B}$ é uma função $\mathsf{x} : \mathcal{B} \times \Re \rightarrow \mathcal{E}$ de classe C^3, sendo $\mathsf{x}(\cdot, t)$, para um t fixo, uma alteração da configuração espacial de $\mathcal{B}$. Dessa forma, um movimento é uma família de alterações de configuração suaves e parametrizadas pelo escalar t, geralmente (mas não obrigatoriamente) associado ao *tempo*. Assim,

$$\mathbf{x} = \mathsf{x}(\mathbf{X}, t) \tag{A.21}$$

é a *posição* ocupada pelo ponto material $\mathbf{X}$ no tempo t. Da mesma forma, denota-se

$$\mathcal{B}_t = \mathsf{x}(\mathcal{B}, t) \tag{A.22}$$

a região do espaço ocupado pelo corpo $\mathcal{B}$ em t.

Em alguns casos é mais conveniente trabalhar com posições espaciais e tempos ao invés de pontos materiais e tempos. Por essa razão, introduz-se o conjunto *trajetória*

$$\mathcal{T} = \{(\mathbf{x}, t) \,|\mathbf{x} \in \mathcal{B}_t, t \in \Re\}. \tag{A.23}$$

A cada t, $\mathsf{x}(\mathbf{X}, t)$ é uma transformação suave um-a-um de $\mathcal{B}$ sobre $\mathcal{B}_t$. Dessa maneira, existe a transformação inversa $\mathsf{X}(\cdot, t) : \mathcal{B}_t \rightarrow \mathcal{B}$ definida por

$$\mathbf{X} = \mathsf{X}(\mathbf{x}, t) \qquad \text{tal que} \qquad \mathsf{x}(\mathsf{X}(\mathbf{x}, t), t) = \mathbf{x}, \qquad \mathsf{X}(\mathsf{x}(\mathbf{X}, t), t) = \mathbf{X}.$$

Dado $(\mathbf{x}, t) \in \mathcal{T}$,

$$\mathbf{X} = \mathsf{X}(\mathbf{x}, t) \tag{A.24}$$

é o ponto material que ocupa a posição $\mathbf{x}$ no tempo t. A transformação

$$\mathsf{X} : \mathcal{T} \rightarrow \mathcal{B}$$

é chamada de *transformação de referência* do movimento.

Pode-se também utilizar o conceito de vetor deslocamento (A.8) para descrever um movimento. Em cada instante

$$\mathbf{x} = \mathbf{X} + \mathbf{u}.$$

Logo,

$$\boxed{\mathsf{x}(\mathbf{X},t) = \mathbf{X} + \mathbf{u}(\mathbf{X},t)\,.} \tag{A.25}$$

De forma geral, qualquer campo associado ao movimento pode ser expresso em função do domínio $\mathcal{B} \times \Re$ (campos materiais) ou de $\mathcal{T}$ (campos espaciais). Uma descrição em campos materiais é uma descrição do tipo *Lagrangeana*: acompanha-se a evolução dos pontos materiais no tempo, tendo-se como base a geometria inicial $\mathcal{B}$; ao se fixar um determinado ponto material $\mathbf{X}$, tem-se sua posição espacial $\mathbf{x}$ a cada instante t. Uma descrição em campos espaciais é uma descrição do tipo *Euleriana*: acompanha-se o comportamento de posições no espaço, seguindo as geometrias modificadas $\mathcal{B}_t$ durante o movimento; ao se fixar uma posição $\mathbf{x}$, a cada t observa-se um ponto material diferente.

As derivadas no tempo do movimento fornecem a velocidade e a aceleração instantâneas dadas, respectivamente, por

$$\dot{\mathsf{x}}(\mathbf{X},t) = \frac{\partial}{\partial t}\mathsf{x}(\mathbf{X},t)\,, \qquad \ddot{\mathsf{x}}(\mathbf{X},t) = \frac{\partial^2}{\partial t^2}\mathsf{x}(\mathbf{X},t)\,.$$

Usando a transformação de referência (A.24), define-se a descrição espacial da velocidade como

$$\mathbf{v}(\mathbf{x},t) = \dot{\mathsf{x}}(\mathsf{X}(\mathbf{x},t)\,,t)\,. \tag{A.26}$$

O vetor $\mathbf{v}(\mathbf{x},t)$ é a velocidade do ponto material que ocupa a posição espacial $\mathbf{x}$ no tempo t.

O campo

$$\mathbf{F} = \nabla\mathsf{x} \tag{A.27}$$

é chamado *gradiente da mudança de configuração espacial* no movimento x. Como a transformação $\mathbf{X} \mapsto \mathsf{x}(\mathbf{X},t)$ é uma deformação de $\mathcal{B}$, então $\det\mathbf{F} > 0$.

Considere agora um campo de velocidades genérico $\mathbf{v}$ e sua expansão em série de Taylor na vizinhança de uma posição espacial $\mathbf{y}$

$$\mathbf{v}(\mathbf{x},t) = \mathbf{v}(\mathbf{y},t) + \mathbf{L}(\mathbf{y},t)(\mathbf{x}-\mathbf{y}) + o(\mathbf{x}-\mathbf{y}) \tag{A.28}$$

sendo $\mathbf{L} = \operatorname{grad}\mathbf{v}$ o tensor gradiente do campo de velocidades. Sejam então $\mathbf{D}$ e $\mathbf{W}$, respectivamente, as partes simétrica e antissimétrica de $\mathbf{L}$, ou seja,

$$\mathbf{D} = \frac{1}{2}\left(\mathbf{L} + \mathbf{L}^T\right) = \frac{1}{2}\left(\operatorname{grad}\mathbf{v} + \operatorname{grad}\mathbf{v}^T\right),$$

$$\mathbf{W} \quad = \quad \frac{1}{2}\left(\mathbf{L} - \mathbf{L}^T\right) = \frac{1}{2}\left(\text{grad}\,\mathbf{v} - \text{grad}\,\mathbf{v}^T\right). \tag{A.29}$$

Logo, $\mathbf{L} = \mathbf{D} + \mathbf{W}$ e $\mathbf{v}(\mathbf{x}, t) = \mathbf{v}(\mathbf{y},t) + \mathbf{W}(\mathbf{y},t)(\mathbf{x} - \mathbf{y}) + \mathbf{D}(\mathbf{y},t)(\mathbf{x} - \mathbf{y}) + o(\mathbf{x} - \mathbf{y})$.

Assim, em uma vizinhança de um dado ponto $\mathbf{y}$ e dentro de um erro $o(\mathbf{x} - \mathbf{y})$, um campo de velocidades genérico é a soma de um campo de velocidades rígido

$$\mathbf{x} \mapsto \mathbf{v}(\mathbf{y},t) + \mathbf{W}(\mathbf{y},t)(\mathbf{x} - \mathbf{y}), \tag{A.30}$$

e um campo de velocidades de deformação

$$\mathbf{x} \mapsto \mathbf{D}(\mathbf{y},t)(\mathbf{x} - \mathbf{y}). \tag{A.31}$$

Em (A.30), $\mathbf{v}(\mathbf{y},t)$ é a velocidade de translação e $\mathbf{W}(\mathbf{y},t)$ é o tensor antissimétrico associado a rotação. Por sua vez, (A.31) descreve um campo de velocidades variando linearmente a partir do ponto $\mathbf{y}$ (com velocidade nula) em três direções mutuamente ortogonais (teorema espectral (Gurtin, 1981)), ou seja, descreve três campos de alongamentos instantâneos a partir de $\mathbf{y}$. Por isso, $\mathbf{W}(\mathbf{y},t)$ e $\mathbf{D}(\mathbf{y},t)$ são chamados, respectivamente, de tensores de *spin* e de *taxa de deformação*.

A.7 Aplicação de Esforços Externos

Seja um movimento do corpo $\mathcal{B}$. Durante um movimento mecânico, as interações entre as diversas partes do corpo entre si e com o meio externo são descritas por *forças*, as quais podem ser dividas em dois grupos: *forcas de contato* e *forças de volume*. As forças de contato são exercidas sobre a superfície das partes pelo meio externo ou pelas outras partes do corpo. As forças de volume são exercidas pelo meio externo nos pontos interiores do corpo, tais como a gravidade ou a força eletromagnética.

De acordo com a hipótese de Cauchy para as forças de contato, tem-se uma densidade de força de superfície $\mathbf{s}(\mathbf{n}, \mathbf{x}, t)$ definida para cada vetor unitário $\mathbf{n}$ em cada ponto $(\mathbf{x}, t)$ da trajetória $\mathcal{T}$ do movimento. Dessa forma, seja Γ_t um superfície orientada de $\mathcal{B}_t$ com normal unitária $\mathbf{n}$ em $\mathbf{x}$. Logo, $\mathbf{s}(\mathbf{n}, \mathbf{x}, t)$ é a força, por unidade de área, exercida através de Γ_t sobre o material do lado negativo de Γ_t pelo material do lado positivo. Em outras palavras, a força de contato depende apenas da normal e do ponto, não dependendo intrínsecamente da superfície.

A partir dessa hipótese, a força total de contato exercida em uma parte $\mathcal{P}$ de $\mathcal{B}$ no tempo t é

$$\int_{\partial \mathcal{P}t} \mathbf{s}(\mathbf{n})\ dA \equiv \int_{\partial \mathcal{P}t} \mathbf{s}(\mathbf{n}, \mathbf{x}, t)\ dA. \tag{A.32}$$

As forças de volume aplicadas pelo meio são definidas por um campo vetorial espacial $\mathbf{b}(\mathbf{x}, t)$ que fornece a força por unidade de volume em cada ponto da trajetória. Logo, a força de volume em $\mathcal{P}$ é dada pela integral

$$\int_{\mathcal{P}t} \mathbf{b}\ dV \equiv \int_{\mathcal{P}t} \mathbf{b}(\mathbf{x}, t)\ dV. \tag{A.33}$$

Os axiomas básicos associando movimento e forças são as leis de balanço de momentos lineares e angulares, calculadas no domínio espacial, dadas, respectivamente, por

$$\begin{aligned} \int_{\partial \mathcal{P}t} \mathbf{s}(\mathbf{n})\ dA + \int_{\mathcal{P}t} \mathbf{b}\ dV &= \dot{\mathbf{m}}_L(\mathcal{P}, t), \\ \int_{\partial \mathcal{P}t} \mathbf{r} \times \mathbf{s}(\mathbf{n})\ dA + \int_{\mathcal{P}t} \mathbf{r} \times \mathbf{b}\ dV &= \dot{\mathbf{m}}_A(\mathcal{P}, t), \end{aligned} \tag{A.34}$$

sendo $\mathbf{n}$ a normal unitária a $\partial \mathcal{P}_t$ e $\mathbf{r}$ o vetor posição. Os momentos linear $\mathbf{m}_L(\mathcal{P}, t)$ e angular $\mathbf{m}_A(\mathcal{P}, t)$ (em torno da origem $\mathbf{o}$) de $\mathcal{P}$ no instante t são dados, respectivamente, por

$$\begin{aligned} \mathbf{m}_L(\mathcal{P}, t) &= \int_{\mathcal{P}t} \mathbf{v}\rho\ dV, \\ \mathbf{m}_A(\mathcal{P}, t) &= \int_{\mathcal{P}t} \mathbf{r} \times \mathbf{v}\rho\ dV, \end{aligned} \tag{A.35}$$

para o campo escalar de densidade ρ e vetor posição $\mathbf{r}(\mathbf{x}) = \mathbf{x} - \mathbf{o}$.

O teorema seguinte é um dos resultados principais da mecânica do contínuo. Sua afirmação mais importante é que $\mathbf{s}(\mathbf{n})$ é linear em $\mathbf{n}$.

Teorema A.2 ***Teorema de Cauchy*** (Gurtin, 1981).
Seja $(\mathbf{s}, \mathbf{b})$ *um sistema de forças atuando no corpo* $\mathcal{B}$ *durante um movimento. Portanto, uma condição necessária e suficiente para que as equações de balanço de momento sejam satisfeitas é que exista um campo tensorial espacial* $\boldsymbol{\sigma}$ *(chamado de tensão de Cauchy) tal que:*

1. *para cada vetor unitário* $\mathbf{n}$

$$\mathbf{s}(\mathbf{n}) = \boldsymbol{\sigma}\mathbf{n}, \tag{A.36}$$

2. $\boldsymbol{\sigma}$ *é simétrico,*

3. $\boldsymbol{\sigma}$ *satisfaz a equação de movimento*

$$\operatorname{div}\ \boldsymbol{\sigma} + \mathbf{b} = \rho\dot{\mathbf{v}}. \tag{A.37}$$

Pelo teorema anterior, tem-se que um sistema de forças $(\mathbf{s}, \mathbf{b})$ é completamente determinado pela tensão $\boldsymbol{\sigma}$ e pelo movimento x. O par $(\mathsf{x}, \boldsymbol{\sigma})$ define um **processo dinâmico**, sendo

a) x um movimento,

b) $\boldsymbol{\sigma}$ um campo tensorial simétrico na trajetória $\mathcal{T}$ de x,

c) $\boldsymbol{\sigma}(\mathbf{x}, t)$ uma função suave de $\mathbf{x}$ em $\mathcal{B}_t$.

O balanço de energia envolvida em um processo dinâmico, para toda parte $\mathcal{P}$ e tempo t, é dado pela expressão (Gurtin, 1981)

$$\boxed{\int_{\partial\mathcal{P}t} \boldsymbol{\sigma}\mathbf{n}\cdot\mathbf{v}\ dA + \int_{\mathcal{P}t} \mathbf{b}\cdot\mathbf{v}\ dV = \int_{\mathcal{P}t} \boldsymbol{\sigma}\cdot\mathbf{D}\ dV + \frac{d}{dt}\int_{\mathcal{P}t} \frac{1}{2}\rho\mathbf{v}\cdot\mathbf{v}\ dV.} \tag{A.38}$$

O lado esquerdo de (A.38),

$$P_{ext} = \int_{\partial\mathcal{P}t} \boldsymbol{\sigma}\mathbf{n}\cdot\mathbf{v}\ dA + \int_{\mathcal{P}t} \mathbf{b}\cdot\mathbf{v}\ dV, \tag{A.39}$$

representa a potência de todas as ações mecânicas externas atuando sobre a parte $\mathcal{P}$ do corpo durante o movimento. Por isso, pode ser chamada de *potência mecânica externa* da parte $\mathcal{P}$. Essa quantidade traduz toda energia mecânica fornecida a $\mathcal{P}$ pelo meio externo. Por sua vez, o termo

$$P_{int} + P_{cin} = -\left(\int_{\mathcal{P}t} \boldsymbol{\sigma}\cdot\mathbf{D}\ dV + \frac{d}{dt}\int_{\mathcal{P}t} \frac{1}{2}\rho\mathbf{v}\cdot\mathbf{v}\ dV\right)$$

é a *potência mecânica armazenada* na parte $\mathcal{P}$, sendo

$$P_{int} = \int_{\mathcal{P}t} \boldsymbol{\sigma}\cdot\mathbf{D}\ dV \tag{A.40}$$

a *potência mecânica interna* (energia armazenada como deformação instantânea) e

$$P_{cin} = \frac{d}{dt}\int_{\mathcal{P}t} \frac{1}{2}\rho\mathbf{v}\cdot\mathbf{v}\ dV \tag{A.41}$$

a *potência cinética* (variação instantânea da energia cinética).

A.8 Tensores de Tensão de Piola-Kirchhoff

O tensor de tensão de Cauchy mede a força de contato por unidade de área na configuração deformada. Entretanto, como na maioria dos casos a configuração deformada não é conhecida *a priori*[4], introduz-se um tensor de tensões que fornece a força por unidade de área na configuração de referência.

Seja $(\mathsf{x}, \boldsymbol{\sigma})$ um processo dinâmico. Aplica-se a mudança de domínio de integração (Gurtin, 1981)

$$\int_{\partial \mathcal{P}t} \boldsymbol{\sigma} \mathbf{m} \, dA = \int_{\partial \mathcal{P}} (\det \mathbf{F}) \, \boldsymbol{\sigma}_m \mathbf{F}^{-T} \mathbf{n} \, dA, \tag{A.42}$$

sendo $\mathbf{m}$ e $\mathbf{n}$, respectivamente, os campos de normais unitárias em $\partial \mathcal{P}_t$ e $\partial \mathcal{P}$, enquanto que $\boldsymbol{\sigma}_m$ é a descrição material do tensor de tensões de Cauchy $\boldsymbol{\sigma}$.

Define-se o campo tensorial $\mathbf{T} : \mathcal{B} \times \Re \rightarrow \mathrm{Lin}$, chamado de *primeiro tensor de tensão de Piola-Kirchhoff* como

$$\boxed{\mathbf{T} = (\det \mathbf{F}) \, \boldsymbol{\sigma}_m \mathbf{F}^{-T}.} \tag{A.43}$$

Portanto,

$$\int_{\partial \mathcal{P}t} \boldsymbol{\sigma} \mathbf{m} \, dA = \int_{\partial \mathcal{P}} \mathbf{T} \mathbf{n} \, dA. \tag{A.44}$$

De acordo com (A.44), $\mathbf{Tn}$ é a força de superfície medida por unidade de área na configuração de referência.

Da mesma forma, integrando a força de volume no domínio de referência,

$$\int_{\mathcal{P}t} \mathbf{b} \, dV = \int_{\mathcal{P}} \mathbf{b}_m (\det \mathbf{F}) \; dV = \int_{\mathcal{P}} \mathbf{b}_0 \, dV,$$

define-se a força de volume de referência $\mathbf{b}_0$, que fornece a força de volume por unidade de volume da configuração de referência.

A partir dessas definições, pode-se reescrever as equações de balanço (A.34) no domínio de referência da seguinte forma

$$\int_{\partial \mathcal{P}} \mathbf{T} \mathbf{n} \, dA + \int_{\mathcal{P}} \mathbf{b}_0 \, dV \quad = \quad \int_{\mathcal{P}} \ddot{\mathsf{x}} \rho_0 \, dV$$

[4]Este é o caso de problemas envolvendo sólidos, nos quais a própria configuração deformada é uma das incógnitas do problema.

$$\int_{\partial\mathcal{P}} (\mathsf{x} - \mathbf{o}) \times \mathbf{Tn}\; dA + \int_{\mathcal{P}} (\mathsf{x} - \mathbf{o}) \times \mathbf{b}_0\; dV \quad = \quad \int_{\mathcal{P}} (\mathsf{x} - \mathbf{o}) \times \ddot{\mathsf{x}}\rho_0\; dV \tag{A.45}$$

sendo $\mathbf{o}$ a origem adotada pelo observador O e ρ_0 o campo de densidade na configuração de referência.

Considerando (A.45), o teorema da divergência e a simetria de $\boldsymbol{\sigma}$ conclui-se que $\mathbf{T}$ satisfaz as seguintes equações de campo

$$\begin{aligned} \operatorname{Div}\mathbf{T} + \mathbf{b}_0 &= \rho_0\ddot{\mathsf{x}}, \\ \mathbf{TF}^T &= \mathbf{FT}^T. \end{aligned} \tag{A.46}$$

Pode-se também reescrever a expressão (A.38) no domínio de referência, ou seja,

$$\boxed{\int_{\partial\mathcal{P}} \mathbf{Tn}\cdot\dot{\mathsf{x}}\; dA + \int_{\mathcal{P}} \mathbf{b}_0\cdot\dot{\mathsf{x}}\; dV = \int_{\mathcal{P}} \mathbf{T}\cdot\dot{\mathbf{F}}\; dV + \frac{d}{dt}\int_{\mathcal{P}} \frac{\dot{\mathsf{x}}^2}{2}\rho_0\; dV.} \tag{A.47}$$

Observa-se que $\mathbf{T}$ é geralmente não-simétrico. Além disso, $\mathbf{T}$ é um tensor *misto*[5], ou seja, apesar de estar definido no domínio dos pontos materiais origina uma transformação cujo resultado é uma grandeza espacial. De fato, $\mathbf{T}\mathbf{n}dA$ fornece uma força na configuração *deformada* relativa a um elemento de área $\mathbf{n}dA$ na configuração de *referência*. Em resumo, $\mathbf{T}$ não está completamente relacionado à configuração de referência. Por esse motivo, a partir de (A.43), define-se o *segundo tensor de tensão de Piola-Kirchhoff* como

$$\boxed{\mathbf{S} = \mathbf{F}^{-1}\mathbf{T},} \tag{A.48}$$

ou seja,

$$\mathbf{S} = \mathbf{F}^{-1}(\det\mathbf{F})\,\boldsymbol{\sigma}_m\mathbf{F}^{-T} \quad \Rightarrow \quad \boldsymbol{\sigma}_m = (\det\mathbf{F})^{-1}\,\mathbf{FSF}^T. \tag{A.49}$$

Como $\mathbf{F}^{-1}$ age localmente como uma transformação $\mathbf{x} \to \mathbf{X}$, então $\mathbf{S}\mathbf{n}dA$ fornece uma força *na configuração de referência* (equivalente à força na configuração deformada $\mathbf{T}\mathbf{n}dA$). Portanto, $\mathbf{S}$ está completamente associado à configuração material. É simples observar que $\mathbf{S}$ é um tensor simétrico.

Como mencionado, (A.47) é obtido de (A.38) através da mudança de domínio de integração. Refazendo este cálculo para o primeiro termo do lado direito de (A.38), tem-se

$$\int_{\mathcal{P}t} \boldsymbol{\sigma}\cdot\mathbf{D}\; dV = \int_{\mathcal{P}} \boldsymbol{\sigma}_m\cdot\mathbf{L}_m \det\mathbf{F}\; dV = \int_{\mathcal{P}} \boldsymbol{\sigma}_m\cdot\dot{\mathbf{F}}\mathbf{F}^{-1} \det\mathbf{F}\; dV = \int_{\mathcal{P}} \mathbf{T}\cdot\dot{\mathbf{F}}\; dV.$$

[5]O mesmo ocorre com o tensor $\mathbf{F}$ que representa uma transformação local dos pontos do corpo de sua descrição material para a espacial.

Partindo da mesma integral, pode-se também obter a seguinte relação de energia

$$\int_{\mathcal{P}t} \boldsymbol{\sigma} \cdot \mathbf{D}\ dV = \int_{\mathcal{P}} (\det \mathbf{F})\, \boldsymbol{\sigma}_m \cdot \mathbf{D}_m\ dV = \int_{\mathcal{P}} \mathbf{FSF}^T \cdot \mathbf{F}^{-T}\dot{\mathbf{E}}\mathbf{F}^{-1}\ dV = \int_{\mathcal{P}} \mathbf{S} \cdot \dot{\mathbf{E}}\ dV.$$

Aplica-se a definição do segundo tensor de Piola-Kirchhoff e

$$\dot{\mathbf{E}} = \frac{1}{2}\left(\dot{\mathbf{F}}^T\mathbf{F} + \mathbf{F}^T\dot{\mathbf{F}}\right) = \frac{1}{2}\left(\mathbf{F}^T\mathbf{L}_m^T\mathbf{F} + \mathbf{F}^T\mathbf{L}_m\mathbf{F}\right) = \mathbf{F}^T\mathbf{D}_m\mathbf{F}. \tag{A.50}$$

Portanto, a potência interna de um corpo poder ser descrita de forma equivalente por diversos pares tensão/deformação. Porém cada aplicação pode induzir escolhas mais convenientes. Por exemplo, na formulação numérica e implementação de problemas estruturais com não-linearidades geométricas é mais simples adotar $\int_{\mathcal{P}} \mathbf{S} \cdot \dot{\mathbf{E}}\ dV$, devido à sua simetria, ao invés de $\int_{\mathcal{P}} \mathbf{T} \cdot \dot{\mathbf{F}}\ dV$.

A.9 Elasticidade Linear e Pequenas Deformações

A elasticidade linear compreende as situações nas quais as componentes do tensor gradiente do campo de deslocamentos $\nabla\mathbf{u}$ são pequenas e a equação constitutiva em termos do tensor de Piola-Kirchhoff, $\mathbf{T} = \boldsymbol{\Upsilon}(\mathbf{F})$, pode ser linearizada.

Supondo que a tensão residual é nula ($\boldsymbol{\Upsilon}(\mathbf{I}) = \mathbf{0}$) e linearizando a equação constitutiva vem que

$$\begin{aligned}
\boldsymbol{\Upsilon}(\mathbf{F}) &= \boldsymbol{\Upsilon}(\mathbf{I} + \nabla\mathbf{u}) = \boldsymbol{\Upsilon}(\mathbf{I}) + D\boldsymbol{\Upsilon}(\mathbf{I})[\nabla\mathbf{u}] + o(\nabla\mathbf{u}) \\
&= \mathsf{C}[\boldsymbol{\epsilon}] + o(\nabla\mathbf{u}),
\end{aligned}$$

sendo $\mathsf{C} = D\boldsymbol{\Upsilon}(\mathbf{I})$ o tensor de elasticidade.

Portanto,

$$\boldsymbol{\sigma} = \mathbf{T} = \boldsymbol{\Upsilon}(\mathbf{F}) = \mathsf{C}[\boldsymbol{\epsilon}] + o(\nabla\mathbf{u}).$$

Logo, se a tensão residual na configuração de referência é nula, dentro de um erro $o(\nabla\mathbf{u})$ quando $\nabla\mathbf{u} \to \mathbf{0}$, a tensão de Piola-Kirchhoff $\mathbf{T}$ é uma função linear da deformação infinitesimal $\boldsymbol{\epsilon}$. Além disso, como C é simétrico, dentro do mesmo erro, $\mathbf{T}$ é um tensor simétrico, coincidindo com $\boldsymbol{\sigma}$.

A teoria linear da elasticidade é então baseada nas seguintes relações:

$$\begin{aligned}
&\boldsymbol{\sigma} = \mathsf{C}\left[\boldsymbol{\epsilon}\right] && \rightarrow && \text{relação tensão-deformação linear,}\\
&\boldsymbol{\epsilon} = \tfrac{1}{2}\left(\nabla\mathbf{u} + \nabla\mathbf{u}^T\right) && \rightarrow && \text{relação deformação-deslocamento,}\\
&\operatorname{Div}\boldsymbol{\sigma} + \mathbf{b}_0 = \rho_0\ddot{\mathbf{u}} && \rightarrow && \text{equação de movimento,}
\end{aligned} \tag{A.51}$$

sendo $\mathbf{u}\left(\mathbf{p}, t\right) = \mathsf{x}\left(\mathbf{X}, t\right) - \mathbf{X}$. As hipóteses são

1. tensão residual nula na configuração de referência
2. o gradiente do campo de deslocamentos é pequeno.

Quando o material do corpo $\mathcal{B}$ é isotrópico, tem-se

$$\boldsymbol{\sigma} = \mathsf{C}\left[\boldsymbol{\epsilon}\right] = 2\tilde{\mu}\boldsymbol{\epsilon} + \tilde{\lambda}\left(\operatorname{tr}\boldsymbol{\epsilon}\right)\mathbf{I}. \tag{A.52}$$

Diz-se que o corpo é homogêneo quando a densidade ρ_0 e os coeficientes de Lamé $\tilde{\mu}$ e $\tilde{\lambda}$ são constantes para todo $\mathbf{X} \in \mathcal{B}$.

Nesse contexto, a função de densidade de energia de deformação é dada por

$$W = \tilde{\mu}\boldsymbol{\epsilon}\cdot\boldsymbol{\epsilon} + \frac{1}{2}\tilde{\lambda}\left(\operatorname{tr}\boldsymbol{\epsilon}\right)^2. \tag{A.53}$$

Por sua vez, o tensor de elasticidade, de acordo com a Seção A.3, é dado por

$$\mathsf{C} = 2\tilde{\mu}\mathsf{I} + \tilde{\lambda}\left(\mathbf{I}\otimes\mathbf{I}\right). \tag{A.54}$$

Portanto, reescrevem-se as expressões (A.52) e (A.53) como

$$\boldsymbol{\sigma} = \mathsf{C}\left[\boldsymbol{\epsilon}\right] = \mathsf{C} : \boldsymbol{\epsilon} = 2\tilde{\mu}\mathsf{I} : \boldsymbol{\epsilon} + \tilde{\lambda}\left(\mathbf{I}\otimes\mathbf{I}\right) : \boldsymbol{\epsilon}, \tag{A.55}$$

$$W = \frac{1}{2}\boldsymbol{\epsilon}\cdot\mathsf{C}\left[\boldsymbol{\epsilon}\right] = \frac{1}{2}\boldsymbol{\epsilon}\cdot\mathsf{C} : \boldsymbol{\epsilon}. \tag{A.56}$$

O tensor de deformação infinitesimal $\boldsymbol{\epsilon}$ pode ser decomposto na soma de uma parcela relacionada somente à deformação volumétrica $\tilde{\boldsymbol{\epsilon}}$ e outra parcela relacionada à deformação de distorção $\bar{\boldsymbol{\epsilon}}$, a qual preserva o volume ($\operatorname{tr}\bar{\boldsymbol{\epsilon}} = 0$), ou seja,

$$\boldsymbol{\epsilon} = \tilde{\boldsymbol{\epsilon}} + \bar{\boldsymbol{\epsilon}} \tag{A.57}$$

com

$$\tilde{\boldsymbol{\epsilon}} = \frac{1}{3}\operatorname{tr}\left[\boldsymbol{\epsilon}\right]\mathbf{I} \qquad \text{e} \qquad \bar{\boldsymbol{\epsilon}} = \boldsymbol{\epsilon} - \frac{1}{3}\operatorname{tr}\left[\boldsymbol{\epsilon}\right]\mathbf{I}. \tag{A.58}$$

Pode-se denotar os tensores anteriores como

$$\tilde{\boldsymbol{\epsilon}} = \frac{1}{3}(\mathbf{I} \otimes \mathbf{I}) : \boldsymbol{\epsilon} \qquad \text{e} \qquad \bar{\boldsymbol{\epsilon}} = \left[\mathsf{I} - \frac{1}{3}\mathbf{I} \otimes \mathbf{I}\right] : \boldsymbol{\epsilon} = \mathsf{I}_{\text{dev}} : \boldsymbol{\epsilon}, \tag{A.59}$$

com

$$\mathsf{I}_{\text{dev}} = \mathsf{I} - \frac{1}{3}\mathbf{I} \otimes \mathbf{I}, \tag{A.60}$$

$$\mathsf{I}_{\text{dev}} : \mathbf{A} = \mathbf{A} - \frac{1}{3}\operatorname{tr}[\mathbf{A}]\,\mathbf{I}. \tag{A.61}$$

Substituindo (A.57) e (A.58) em (A.55) e (A.56), tem-se que

$$\boldsymbol{\sigma} = 2\tilde{\mu}\bar{\boldsymbol{\epsilon}} + \tilde{\kappa}\,(\operatorname{tr}\boldsymbol{\epsilon})\,\mathbf{I}, \tag{A.62}$$

$$W = \tilde{\mu}\bar{\boldsymbol{\epsilon}} \cdot \bar{\boldsymbol{\epsilon}} + \frac{1}{2}\tilde{\kappa}\,(\operatorname{tr}\boldsymbol{\epsilon})^2 = \bar{W} + \tilde{W}, \tag{A.63}$$

sendo

$$\tilde{\kappa} = \tilde{\lambda} + \frac{2}{3}\tilde{\mu} \tag{A.64}$$

o módulo volumétrico (*bulk modulus*) e $\tilde{\mu}$ o módulo de cisalhamento. Por razões físicas, impõe-se que $\tilde{\mu} > 0$ e $\tilde{\kappa} > 0$ (Gurtin, 1981). O tensor de elasticidade passa a ser expresso por,

$$\mathsf{C} = 2\tilde{\mu}\mathsf{I}_{\text{dev}} + \tilde{\kappa}\mathbf{I} \otimes \mathbf{I}. \tag{A.65}$$

Pode-se reescrever (A.62) como

$$\boldsymbol{\sigma} = 2\tilde{\mu}\bar{\boldsymbol{\epsilon}} + q\mathbf{I}, \tag{A.66}$$

sendo $q \equiv q(\mathbf{u}) = \tilde{\kappa}\,(\operatorname{tr}\boldsymbol{\epsilon}) = \tilde{\kappa}\operatorname{div}\mathbf{u}$ no caso de material compressível. Se o material é incompressível, tem-se $\operatorname{tr}[\boldsymbol{\epsilon}] = 0$, $\boldsymbol{\epsilon} = \bar{\boldsymbol{\epsilon}}$ e q é um variável independente pois a condição de incompressibilidade implica $\tilde{\kappa} \to \infty$ e $\operatorname{div}\mathbf{u} \to 0$. Nesse caso,

$$W = \tilde{\mu}\boldsymbol{\epsilon} \cdot \boldsymbol{\epsilon} = \tilde{\mu}\bar{\boldsymbol{\epsilon}} \cdot \bar{\boldsymbol{\epsilon}}. \tag{A.67}$$

Estas equações são das mesmas usadas no escoamento de fluidos viscosos incompressíveis.

No caso de sólidos, ao invés dos módulos de Lamé, geralmente as propriedades do material são expressas em termos das propriedades $\tilde{E}$ e $\tilde{\nu}$, obtidas de ensaios de tração, denominadas respectivamente de *módulo de Young* e *coeficiente de Poisson*[6]:

$$\tilde{E} = \frac{\tilde{\mu}\left(2\tilde{\mu} + 3\tilde{\lambda}\right)}{\tilde{\mu} + \tilde{\lambda}}, \qquad \tilde{\nu} = \frac{\tilde{\lambda}}{2\left(\tilde{\mu} + \tilde{\lambda}\right)} \tag{A.68}$$

Observa-se que no modelo de elasticidade aplicado aqui, C possui as simetrias

$$\mathsf{C}_{ijkl} = \mathsf{C}_{klij} = \mathsf{C}_{ijlk} = \mathsf{C}_{jilk}, \tag{A.69}$$

ou seja, para quaisquer $\mathbf{A}$ e $\mathbf{B}$,

$$\mathbf{A} : \mathsf{C} : \mathbf{B} = \mathbf{B} : \mathsf{C} \!: \mathbf{A}.$$

A.10 Mudança de Observador

Na mecânica clássica, um observador é definido como um corpo rígido com um relógio. Na teoria de Mecânica do Contínuo, um observador constitui um *quadro*. Uma mudança de observador significa a transformação entre o par $\{\mathbf{X}, t\}$ num quadro para o par $\{\mathbf{X}^*, t^*\}$ de um quadro diferente, sendo $\mathbf{X}$ e $\mathbf{X}^*$ os vetores posição de um ponto material observados nos dois quadros e t e t^* os respectivos tempos.

Considere dois observadores O e O^* presenciando o mesmo evento. Assume-se, por hipótese, que ambos concordam sobre intervalos de tempo e distância[7]. Assim, uma mudança de observador consiste em uma transformação de $\mathcal{E} \times \Re$ em si mesmo preservando distância entre pontos, intervalo de tempo e ordem dos eventos (Ogden, 1984).

Dois observadores presenciando um corpo sofrendo mudanças em sua configuração espacial irão geralmente acompanhar movimentos diferentes do mesmo corpo devido a diferentes percepções de velocidade e aceleração. Como os quadros são corpos rígidos, essa diferença corresponderá a um movimento de corpo rígido de um observador em relação ao outro. Mas apesar das quantidades físicas dependerem do observador, o fenômeno físico é independente e a formulação matemática das leis físicas deve refletir isso. Por exemplo, a constante elástica de uma mola é a mesma para todos os observadores. Os observadores têm a liberdade de escolher diferentes configurações de referência, mas é conveniente adotar uma configuração independente do observador para que todos localizem as partículas materiais $\mathbf{X} \in \mathcal{B} \subset \mathcal{E}$.

[6]A condição $\operatorname{tr} \boldsymbol{\epsilon} = 0$ implica $\tilde{\nu} \to 0.5$ mas como $\tilde{\mu} > 0$ então $\tilde{\lambda} \to \infty$ ($\tilde{\kappa} \to \infty$).
[7]Desprezam-se efeitos relativísticos no presente contexto.

A expressão

$$\mathbf{f}(\cdot, t) = \mathbf{q}(t) + \mathbf{Q}(t)(\cdot - \mathbf{o})$$

define um movimento de corpo rígido, sendo $\mathbf{q}(t)$ uma translação e $\mathbf{Q}(t) \in \text{Orth}^+$ uma rotação. Assim,

$$\begin{cases} \mathbf{X}^* = \mathbf{q}(t) + \mathbf{Q}(t)(\mathbf{X} - \mathbf{o}) \\ t^* = t - a \end{cases} \qquad \forall \mathbf{X}, \forall t.$$

sendo a uma constante.

Sejam x e x^* movimentos de $\mathcal{B}$. Estes dois movimentos serão relacionados por uma mudança de observador (ou mudança de sistema de referência) se

$$\mathsf{x}^*(\mathbf{X}, t) = \mathbf{q}(t) + \mathbf{Q}(t)[\mathsf{x}(\mathbf{X}, t) - \mathbf{o}] \quad \forall \mathbf{X}, \forall t. \tag{A.70}$$

Portanto, a cada instante t, $\mathsf{x}^*(\cdot, t)$ é simplestemente a alteração de configuração $\mathsf{x}(\cdot, t)$ seguida da alteração de configuração rígida $\mathbf{f}(\cdot, t)$, ou seja,

$$\mathsf{x}^*(\cdot, t) = \mathbf{f}(\cdot, t) \circ \mathsf{x}(\cdot, t). \tag{A.71}$$

Derivando (A.70) em relação a $\mathbf{X}$ e adotando a notação $\mathbf{F} = \nabla \mathsf{x}$ e $\mathbf{F}^* = \nabla \mathsf{x}^*$, obtem-se

$$\mathbf{F}^*(\mathbf{X}, t) = \mathbf{Q}(t)\mathbf{F}(\mathbf{X}, t), \tag{A.72}$$

sendo $\det \mathbf{Q} = 1$. Portanto, $\det \mathbf{F}^* = \det \mathbf{F}$.

A partir do teorema da decomposição polar tem-se que

$$\mathbf{F} = \mathbf{R}\mathbf{U} = \mathbf{V}\mathbf{R}, \qquad \mathbf{F}^* = \mathbf{R}^*\mathbf{U}^* = \mathbf{V}^*\mathbf{R}^*.$$

Logo, (A.72) implica que $\mathbf{F}^* = \mathbf{R}^*\mathbf{U}^* = \mathbf{Q}\mathbf{R}\mathbf{U}$.

Como $\mathbf{QR}$ é um rotação, a partir do mesmo teorema de decomposição polar conclui-se que

$$\mathbf{R}^* = \mathbf{Q}\mathbf{R}, \qquad \mathbf{U}^* = \mathbf{U} \tag{A.73}$$

e

$$\mathbf{V}^* = \mathbf{R}^*\mathbf{U}^*\mathbf{R}^{*T} = \mathbf{Q}\mathbf{R}\mathbf{U}\mathbf{R}^T\mathbf{Q}^T, \qquad \mathbf{V}^* = \mathbf{Q}\mathbf{V}\mathbf{Q}^T. \tag{A.74}$$

Aplicando mesmas relações aos tensores de deformação de Cauchy-Green,

$$\mathbf{C}^* = \mathbf{U}^{*2} \Rightarrow \mathbf{C}^* = \mathbf{C} \qquad \mathbf{B}^* = \mathbf{V}^{*2} \Rightarrow \mathbf{B}^* = \mathbf{QBQ}^T, \tag{A.75}$$

obtem-se

$$\mathbf{E}^* = \mathbf{E} \tag{A.76}$$

como relação para o tensor Lagrangeano de deformação finita.

Derivando (A.70) em relação ao tempo, obtem-se

$$\dot{\mathsf{x}}^* (\mathbf{X}, t) = \dot{\mathbf{q}} (t) + \mathbf{Q} (t) \dot{\mathsf{x}} (\mathbf{X}, t) + \dot{\mathbf{Q}} (t) [\mathsf{x} (\mathbf{X}, t) - \mathbf{o}] .$$

Tomando $\mathbf{X} = \mathsf{X}^* (\mathbf{x}^*, t)$ do lado esquerdo e $\mathbf{X} = \mathsf{X} (\mathbf{x}, t)$ do lado direito conclui-se que

$$\mathbf{v}^* (\mathbf{x}^*, t) = \dot{\mathbf{q}} (t) + \mathbf{Q} (t) \mathbf{v} (\mathbf{x}, t) + \dot{\mathbf{Q}} (t) [\mathbf{x} - \mathbf{o}] , \tag{A.77}$$

que define a expressão da lei de transformação da velocidade.

Considerando $\mathbf{x}^* = \mathbf{f} (\mathbf{x}, t)$ em (A.77), diferenciando com respeito a $\mathbf{x}$, usando a regra da cadeia e $\operatorname{grad} \mathbf{f} = \mathbf{Q}$, chega-se à relação

$$\mathbf{L}^* (\mathbf{x}^*, t) \mathbf{Q} (t) = \mathbf{Q} (t) \mathbf{L} (\mathbf{x}, t) + \dot{\mathbf{Q}} (t) .$$

Portanto,

$$\mathbf{L}^* (\mathbf{x}^*, t) = \mathbf{Q} (t) \mathbf{L} (\mathbf{x}, t) \mathbf{Q} (t)^T + \dot{\mathbf{Q}} (t) \mathbf{Q} (t)^T , \tag{A.78}$$

que é a lei de transformação do gradiente do campo de velocidades.

Como $\mathbf{QQ}^T = \mathbf{I}$,

$$\dot{\mathbf{Q}}\mathbf{Q}^T + \mathbf{Q}\dot{\mathbf{Q}}^T = \mathbf{0} \Rightarrow \dot{\mathbf{Q}}\mathbf{Q}^T = - \left(\dot{\mathbf{Q}}\mathbf{Q}^T \right)^T ,$$

ou seja, $\dot{\mathbf{Q}}\mathbf{Q}^T$ é antissimétrico. Então, a parte a parte simétrica de (A.78) dá origem a

$$\mathbf{D}^* (\mathbf{x}^*, t) = \mathbf{Q} (t) \mathbf{D} (\mathbf{x}, t) \mathbf{Q} (t)^T , \tag{A.79}$$

lei de transformação da taxa de deformação. A parte antissimétrica determina a lei de transformação do spin:

$$\mathbf{W}^* (\mathbf{x}^*, t) = \mathbf{Q} (t) \mathbf{W} (\mathbf{x}, t) \mathbf{Q} (t)^T + \dot{\mathbf{Q}} (t) \mathbf{Q} (t)^T .$$

Claramente, o spin, como medida pontual de rotação é influenciado pela rotação dos observadores através do spin relativo $\dot{\mathbf{Q}}(t)\,\mathbf{Q}(t)^T$.

Os tensores de deformação e taxa de deformação (A.76) e (A.79) não dependem de $\dot{\mathbf{Q}}$ e podem portanto serem considerados intrínsecos ao material pois incorporam medidas de alteração de comprimento e ângulo pontuais. Todo observador associa os mesmos valores a cada uma de tais entidades a esta *indeferença* é refletida nas regras de transformação tensoriais (A.75), (A.76) e (A.79).

É importante ressaltar a distinção entre regras de transformação de tensores Eulerianos como $\mathbf{B}$ e $\mathbf{D}$, por exemplo, de tensores Lagrangeanos como $\mathbf{E}$ e $\mathbf{C}$, ou de tensores mistos como $\mathbf{F}$. Portanto, são denominados *objetivos* os tensores independentes da cinemática de mudança de observador que se transformam de acordo com (A.72), (A.75), (A.76) e (A.79), qualificados de acordo com seu tipo (Euleriano, Lagrangeano ou misto).

Portanto, sejam os campos $\varphi(\mathbf{x},t)$ escalar, $\mathbf{m}(\mathbf{x},t)$ vetorial e $\mathbf{M}(\mathbf{x},t)$ tensorial Eulerianos. São chamados de campos escalar, vetorial e tensorial *indiferentes* ou *objetivos* se, respectivamente,

$$\boxed{\begin{aligned} \varphi^*(\mathbf{x}^*,t^*) &= \varphi(\mathbf{x},t) \\ \mathbf{m}^*(\mathbf{x}^*,t^*) &= \mathbf{Q}(t)\,\mathbf{m}(\mathbf{x},t) \\ \mathbf{M}^*(\mathbf{x}^*,t^*) &= \mathbf{Q}(t)\,\mathbf{M}(\mathbf{x},t)\,\mathbf{Q}(t)^T \end{aligned}} \tag{A.80}$$

Com a mudança de observador, campos vetoriais e tensoriais Eulerianos têm o valor de suas componentes alterado. A lei de transformação compensa essa alteração, mantendo as características do vetor ou tensor. Isso pode ser entendido observando graficamente a transformação de um vetor entre dois sistemas de coordenadas relacionados por uma rotação: apesar do valor das componentes ser diferente para cada sistema de coordenadas, o vetor permanece inalterado, ou seja, matém seu significado.

Os campos Lagrangeanos objetivos $\phi(\mathbf{X},t)$, $\mathbf{n}(\mathbf{X},t)$ e $\mathbf{N}(\mathbf{X},t)$, respectivamente escalar, vetorial e tensorial objetivos, têm a seguinte propriedade:

$$\boxed{\begin{aligned} \phi^*(\mathbf{X},t) &= \phi(\mathbf{X},t) \\ \mathbf{n}^*(\mathbf{X},t) &= \mathbf{n}(\mathbf{X},t) \\ \mathbf{N}^*(\mathbf{X},t) &= \mathbf{N}(\mathbf{X},t) \end{aligned}}. \tag{A.81}$$

Logo, não são afetados por mudanças de observador. Isso faz sentido pois, para ambos observadores a configuração de referência é a mesma e portanto campos avaliados sobre ela devem retornar os mesmos valores, independente do observador.

Um dos principais axiomas da Mecânica é a exigência que a resposta de um material seja independente da mudança de observador. Sendo $\boldsymbol{\sigma}$ e $\boldsymbol{\sigma}^*$ os tensores de tensão de Cauchy num ponto do corpo relacionados por uma mudança de observador, os respectivos vetores tensão são

$$\mathbf{s}\left(\mathbf{n}\right) = \boldsymbol{\sigma}\mathbf{n} \qquad \text{e} \qquad \mathbf{s}^*\left(\mathbf{n}^*\right) = \boldsymbol{\sigma}^*\mathbf{n}^*.$$

sendo $\mathbf{n}^* = \mathbf{Q}\mathbf{n}$. Então espera-se que

$$\mathbf{s}^*\left(\mathbf{n}^*\right) = \mathbf{Q}\mathbf{s}\left(\mathbf{n}\right).$$

Logo,

$$\boldsymbol{\sigma}^*\mathbf{n}^* = \mathbf{Q}\boldsymbol{\sigma}\mathbf{n} \Rightarrow \boldsymbol{\sigma}^*\mathbf{Q}\mathbf{n} = \mathbf{Q}\boldsymbol{\sigma}\mathbf{n} \Rightarrow \boldsymbol{\sigma}^* = \mathbf{Q}\boldsymbol{\sigma}\mathbf{Q}^T.$$

Portanto, o tensor de tensões de Cauchy $\boldsymbol{\sigma}$ é objetivo e a resposta do material é independente do observador. Da mesma forma, $\mathbf{S}^* = \mathbf{S}$, para o segundo tensor de Piola-Kirchhoff. Deve-se sempre trabalhar com relações constitutivas tais que a condição de objetividade seja satisfeita.

A.11 Funções Isotrópicas

Seja Orth o conjunto de todos os tensores ortogonais e $\mathcal{A} \subset \text{Lin}$ um conjunto de tensores. Uma função escalar $\varphi : \mathcal{A} \to \Re$ é isotrópica se (Gurtin, 1981)

$$\varphi\left(\mathbf{A}\right) = \varphi\left(\mathbf{Q}\mathbf{A}\mathbf{Q}^T\right), \qquad \forall \mathbf{A} \in \mathcal{A} \text{ e } \forall \mathbf{Q} \in \text{Orth}.$$

Por sua vez, uma função tensorial $\mathbf{G} : \mathcal{A} \to \text{Lin}$ é isotrópica se

$$\mathbf{Q}\mathbf{G}\left(\mathbf{A}\right)\mathbf{Q}^T = \mathbf{G}\left(\mathbf{Q}\mathbf{A}\mathbf{Q}^T\right), \qquad \forall \mathbf{A} \in \mathcal{A} \text{ e } \forall \mathbf{Q} \in \text{Orth}.$$

Apêndice B

Enunciados Variacionais

B.1 Variação de Gateaux e Derivada no Sentido de Fréchet

Seja $\mathcal{V}$ uma espaço funcional apropriado, tipicamente uma espaço de Banach com dual $\mathcal{V}^*$, e emparceiramento de dualidade $\langle \cdot, \cdot \rangle_{\mathcal{V}} : \mathcal{V}^* \times \mathcal{V} \to \Re$. Dado um funcional $\Pi : \mathcal{V} \to \Re$ e um ponto $\mathbf{u}_0 \in \mathcal{V}$, define-se a *variação de Gateaux* em $\mathbf{u}_0 \in \mathcal{V}$ na direção $\boldsymbol{\xi} \in \mathcal{V}$ como o seguinte limite (quando existir) (Simo e Hughes, 1998)

$$\delta\Pi(\mathbf{u}_0, \boldsymbol{\xi}) := \lim_{\alpha \to 0} \frac{\Pi(\mathbf{u}_0 + \alpha\boldsymbol{\xi}) - \Pi(\mathbf{u}_0)}{\alpha} = \frac{d}{d\alpha}\,\Pi(\mathbf{u}_0 + \alpha\boldsymbol{\xi})\Big|_{\alpha=0} \tag{B.1}$$

Na literatura clássica de cálculo de variações, a função $\mathbf{u}_0 + \alpha\boldsymbol{\xi}$, para $\alpha > 0$, é denominada uma *variação* de $\mathbf{u}_0$.

Um funcional $\Pi : \mathcal{V} \to \Re$ é *diferenciável no sentido de Fréchet* em $\mathbf{u}_0 \in \mathcal{V}$ se existir uma transformação linear e contínua $D\Pi(\mathbf{u}_0)$, chamada de *derivada de Fréchet*, tal que

$$\|\boldsymbol{\xi}\|_{\mathcal{V}} \to 0 \quad \Rightarrow \quad \frac{\Pi(\mathbf{u}_0 + \boldsymbol{\xi}) - \Pi(\mathbf{u}_0) - D\Pi(\mathbf{u}_0) \cdot \boldsymbol{\xi}}{\|\boldsymbol{\xi}\|_{\mathcal{V}}} \to 0 \tag{B.2}$$

O termo $D\Pi(\mathbf{u}_0) \cdot \boldsymbol{\xi}$ é denominado derivada de Fréchet de Π em $\mathbf{u}_0$ na direção $\boldsymbol{\xi}$ e, no sentido geométrico, $D\Pi(\mathbf{u}_0)$ é *a melhor aproximação linear de* Π *em torno de* $\mathbf{u}_0$, pois

$$\Pi(\mathbf{u}_0 + \boldsymbol{\xi}) = \Pi(\mathbf{u}_0) + D\Pi(\mathbf{u}_0) \cdot \boldsymbol{\xi} + o(\boldsymbol{\xi})$$

A variação de Gateaux consiste na generalização da noção de derivada direcional de funções do espaço Euclidiano. Entretanto, essa definição não implica a diferenciabilidade forte da derivada no sentido de Fréchet. De fato, $\delta\Pi(\mathbf{u}_0, \boldsymbol{\xi})$ e $D\Pi(\mathbf{u}_0)\cdot\boldsymbol{\xi}$ somente coincidirão se (Simo e Hughes, 1998):

1. $\delta\Pi(\mathbf{u}_0, \boldsymbol{\xi})$ for linear e contínuo em $\boldsymbol{\xi} \in \mathcal{V}$.
2. $|\delta\Pi(\mathbf{u}, \boldsymbol{\xi}) - \delta\Pi(\mathbf{u}_0, \boldsymbol{\xi})| \to 0$ quando $\mathbf{u} \to \mathbf{u}_0$, uniformemente para $\mathbf{u}$ na bola unitária de centro $\mathbf{u}_0$.

Apesar da condição mais forte de derivada de Fréchet ser sempre assumida no desenvolvimento de enunciados variacionais, a expressão que define a variação de Gateaux se constitui na ferramentra mais conveniente para avaliar a derivada de Fréchet em cálculos práticos.

B.1.1 Equações de Euler-Lagrange

Dado um funcional $\Pi : \mathcal{V} \to \Re$, o *funcional derivada* de Π em $\mathbf{u} \in \mathcal{V}$ é um elemento de $\mathcal{V}^*$, denotado por $\frac{\delta\Pi}{\delta\mathbf{u}}(\mathbf{u}) \in \mathcal{V}^*$, que satisfaz a relação

$$\delta\Pi(\mathbf{u}, \boldsymbol{\xi}) =: \left\langle \frac{\delta\Pi}{\delta\mathbf{u}}(\mathbf{u}), \boldsymbol{\xi} \right\rangle_{\mathcal{V}}. \tag{B.3}$$

A aplicação do conceito de funcional derivada de um funcional dado está relacionado com a extensão da condição de pontos extremos de funções do cálculo padrão para o cálculo de variações descrito pela proposição seguinte.

Proposição B.1 *A condição necessária para um funcional* $\Pi : \mathcal{V} \to \Re$ *possuir extremo local (máximo, mínimo ou sela) em um ponto* $\mathbf{u}_0 \in \mathcal{V}$ *é que*

$$\boxed{\frac{\delta\Pi}{\delta\mathbf{u}}(\mathbf{u}_0) = 0} \tag{B.4}$$

O conjunto de equações definido pela expressão (B.4) *é denominado de equações de Euler-Lagrange.*

Como grande número de problemas físicos podem ser formulados e resolvidos através da determinação de pontos estacionários de funcionais, observa-se o aspecto central da Proposição B.1. Entretanto, deve-se observar que a equação (B.4) não representa a forma mais conveniente para implementação numérica. Nesse caso, é mais conveniente resolver

$$\left\langle \frac{\delta\Pi}{\delta\mathbf{u}}(\mathbf{u}), \boldsymbol{\xi} \right\rangle_{\mathcal{V}} = 0 \qquad \forall\boldsymbol{\xi} \in \mathcal{V}$$

que implica justamente $\frac{\delta\Pi}{\delta\mathbf{u}}(\mathbf{u}) = 0$.

B.2 Formulação Variacional da Hiperelasticidade

Considere um problema hiperelástico sem restrições internas e em equilíbrio estático dado por

$$\operatorname{Div}\mathbf{T} + \mathbf{b}_0 = \mathbf{0} \tag{B.5}$$

$$\mathbf{T} = \frac{\partial W}{\partial \mathbf{F}} \qquad \mathbf{F} = \nabla \mathsf{x}, \tag{B.6}$$

$$\mathsf{x} = \bar{\mathsf{x}} \qquad \text{em } \Gamma^1, \tag{B.7}$$

$$\mathbf{T}\mathbf{n} = \bar{\mathbf{t}} \qquad \text{em } \Gamma^2. \tag{B.8}$$

$\mathbf{b} \equiv \mathbf{b}(\mathbf{x})$ é um campo vetorial espacial, sendo $\mathbf{b}_0(\mathbf{X}) = \mathbf{b}(\mathsf{x}(\mathbf{X})) = \mathbf{b}_m$ a descrição material de $\mathbf{b}$; $\bar{\mathsf{x}}$ é uma função conhecida, enquanto $\bar{\mathbf{t}}$ depende geralmente da alteração de configuração x (de modo conhecido).

Um campo $\mathsf{x} \subset C^2$ definido em $\mathcal{B}$ e satisfazendo (B.7) é uma *alteração de configuração cinematicamente admissível* e não necessariamente precisa satisfazer (B.5) ou (B.8).

Um campo tensorial $\mathbf{T}$ definido sobre $\mathcal{B}$ e satisfazendo (B.5) e a condição de contorno (B.8) com $\mathbf{b}$ e $\bar{\mathbf{t}}$ definidos para qualquer alteração de configuração cinematicamente admissível x é chamado *tensão estaticamente admissível*. Uma tensão estaticamente admissível precisa satisfazer apenas a parcela relacionada à tensão, ou seja, (B.5) e (B.8), e não precisa estar relacionada ao deslocamento admissível que define $\mathbf{b}$ e $\bar{\mathbf{t}}$ através das relações de tensão-deformação.

Suponha que x^* e $\breve{\mathsf{x}}$ sejam duas alterações de configuração cinematicamente admissíveis e $\breve{\mathbf{T}}$ uma tensão estaticamente admissível correspondente a $\breve{\mathsf{x}}$, com força de corpo $\breve{\mathbf{b}}_0 = \mathbf{b}(\breve{\mathsf{x}}(\mathbf{X}))$ e tração de superfície $\bar{\mathbf{t}} = \bar{\mathbf{t}}\left(\mathbf{X}, \breve{\mathsf{x}}, \breve{\mathbf{F}}\right)$. Portanto,

$$\operatorname{Div}\breve{\mathbf{T}} + \breve{\mathbf{b}}_0 = \mathbf{0}, \qquad \breve{\mathbf{T}}\mathbf{n} = \breve{\mathbf{t}} \qquad \text{em } \Gamma^2.$$

Aplicando as definições e o teorema da divergência, tem-se

$$\int_{\mathcal{B}} \operatorname{Div}\left(\breve{\mathbf{T}}^T \mathsf{x}^*\right)\, dV = \int_{\partial\mathcal{B}} \mathbf{n} \cdot \left(\breve{\mathbf{T}}^T \mathsf{x}^*\right)\, dA,$$

sendo

$$\operatorname{Div}\left(\breve{\mathbf{T}}^T \mathsf{x}^*\right) = \breve{\mathbf{T}} \cdot \nabla \mathsf{x}^* + \mathsf{x}^* \cdot \operatorname{Div}\breve{\mathbf{T}} = \breve{\mathbf{T}} \cdot \nabla \mathsf{x}^* + \mathsf{x}^* \cdot \left(-\breve{\mathbf{b}}_0\right),$$

tem-se

$$\int_{\partial\mathcal{B}} \mathbf{n} \cdot \left(\breve{\mathbf{T}}^T \mathsf{x}^*\right) \, dA = \int_{\partial\mathcal{B}} \breve{\mathbf{T}}\mathbf{n} \cdot \mathsf{x}^* \, dA = \int_{\Gamma 1} \breve{\mathbf{T}}\mathbf{n} \cdot \bar{\mathsf{x}} \, dA + \int_{\Gamma 2} \breve{\mathbf{t}} \cdot \mathsf{x}^* \, dA.$$

Portanto,

$$\int_{\mathcal{B}} \breve{\mathbf{T}} \cdot \mathbf{F}^* \, dV - \int_{\mathcal{B}} \breve{\mathbf{b}}_0 \cdot \mathsf{x}^* \, dV = \int_{\Gamma 1} \breve{\mathbf{T}}\mathbf{n} \cdot \bar{\mathsf{x}} \, dA + \int_{\Gamma 2} \breve{\mathbf{t}} \cdot \mathsf{x}^* \, dA. \tag{B.9}$$

Fazendo $\mathsf{x}^* = \breve{\mathsf{x}} \to \mathsf{x}$ e $\breve{\mathbf{T}} \to \mathbf{T}$, tem-se

$$\int_{\mathcal{B}} \mathbf{T} \cdot \mathbf{F} \, dV - \int_{\mathcal{B}} \mathbf{b}_0 \cdot \mathsf{x} \, dV = \int_{\Gamma 1} \mathbf{T}\mathbf{n} \cdot \bar{\mathsf{x}} \, dA + \int_{\Gamma 2} \bar{\mathbf{t}} \cdot \mathsf{x} \, dA. \tag{B.10}$$

Do mesmo modo, mas apenas $\breve{\mathbf{T}} \to \mathbf{T}$:

$$\int_{\mathcal{B}} \mathbf{T} \cdot \mathbf{F}^* \, dV - \int_{\mathcal{B}} \mathbf{b}_0 \cdot \mathsf{x}^* \, dV = \int_{\Gamma 1} \mathbf{T}\mathbf{n} \cdot \bar{\mathsf{x}} \, dA + \int_{\Gamma 2} \bar{\mathbf{t}} \cdot \mathsf{x}^* \, dA. \tag{B.11}$$

Subtraindo (B.10) de (B.11) e usando as notações definidas, obtem-se

$$\int_{\mathcal{B}} \mathbf{T} \cdot (\mathbf{F}^* - \mathbf{F}) \, dV - \int_{\Gamma 2} \bar{\mathbf{t}} \cdot (\mathsf{x}^* - \mathsf{x}) \, dA - \int_{\mathcal{B}} \mathbf{b}_0 \cdot (\mathsf{x}^* - \mathsf{x}) \, dV = 0. \tag{B.12}$$

Observa-se a semelhança entre (B.12) e (A.47) para sistemas em equilíbrio. De fato, introduzindo a notação

$$\delta\mathsf{x} = \mathsf{x}^* - \mathsf{x},$$

$$\delta\mathbf{F} = \mathbf{F}^* - \mathbf{F} = \nabla\delta\mathsf{x},$$

reescreve-se (B.12) como,

$$\boxed{\int_{\mathcal{B}} \mathbf{T} \cdot \delta\mathbf{F} \, dV - \int_{\Gamma 2} \bar{\mathbf{t}} \cdot \delta\mathsf{x} \, dA - \int_{\mathcal{B}} \mathbf{b}_0 \cdot \delta\mathsf{x} \, dV = 0,} \tag{B.13}$$

sendo

$$\delta\mathsf{x} = \mathbf{0} \qquad \text{em } \Gamma^1, \tag{B.14}$$

pois x^* e x são cinematicamente admissíveis.

O enunciado (B.13) é geralmente denominado *Princípio dos Trabalhos Virtuais* e estabelece que o trabalho das forças externas em um *deslocamento imaginário ou virtual* $\delta\mathsf{x}$ a partir da configuração corrente, assumindo que as forças permaneçam inalteradas, é igual ao aumento virtual na energia armazenada no corpo devido à deformação[1].

É interessante observar que, no caso de sistemas em equilíbrio, (B.13) pode ser obtido multiplicando-se (A.47) por um passo de tempo Δt (virtual), considerando a notação como uma variação em torno da configuração corrente, ou seja,

$$\delta \mathsf{x} = \mathsf{x}^* - \mathsf{x} = \dot{\mathsf{x}} \Delta t.$$

No mesmo sentido de variação linear em torno da configuração corrente, tem-se

$$\delta W = \dot{W} \Delta t = \left(\frac{\partial W}{\partial \mathbf{F}} \cdot \dot{\mathbf{F}} \right) \Delta t = \frac{\partial W}{\partial \mathbf{F}} \cdot \dot{\mathbf{F}} \Delta t = \frac{\partial W}{\partial \mathbf{F}} \cdot \delta \mathbf{F}. \tag{B.15}$$

A variação total, entretanto, inclui ainda os termos de maior ordem, ou seja,

$$\Delta W = \frac{\partial W}{\partial \mathbf{F}} \cdot \delta \mathbf{F} + o\left(\Delta t\right),$$

mas, para intervalos Δt pequenos o suficiente para se desprezar $o\left(\Delta t\right)$, a variação linear aproxima bem a variação total, ou seja,

$$\delta W \simeq \Delta W$$

Nesse caso, os deslocamentos $\delta \mathsf{x}$ são denominados *infinitesimais* ou *incrementais*, que significa apenas variações em torno da posição corrente nos quais a hipótese de linearização é válida. Por esse motivo, a manipulação das variações infinitesimais é idêntica à manipulação de derivadas de primeira ordem em relação a um parâmetro escalar como o tempo t. Portanto, $\delta \mathsf{x}$ é 'pequeno' em cada $\mathbf{X} \in \mathcal{B}$ se os termos de ordem $\|\delta \mathsf{x}\|^2$ são desprezíveis em comparação com termos de ordem $\|\delta \mathsf{x}\|$.

Considerando a propriedade

$$\mathbf{T} \cdot \delta \mathbf{F} = \mathrm{Div}\left(\mathbf{T}^T \delta \mathsf{x} \right) - \left(\mathrm{Div}\, \mathbf{T} \right) \cdot \delta \mathsf{x},$$

a expressão (B.13) pode ser reescrita como

$$\int_{\Gamma 2} \left(\mathbf{T}\mathbf{n} - \bar{\mathbf{t}} \right) \cdot \delta \mathsf{x} \, dA - \int_{\mathcal{B}} \left(\mathrm{Div}\, \mathbf{T} + \mathbf{b}_0 \right) \cdot \delta \mathsf{x} \, dV = 0. \tag{B.16}$$

A expressão anterior é verdadeira se $\mathbf{T}$ satisfaz (B.5) e (B.8). Da mesma forma, se (B.16) é verdadeira para um deslocamento virtual cinematicamente admissível $\delta \mathsf{x}$ arbitrário respeitando (B.14), então (B.5) e (B.8) decorrem da solução do enunciado. Essa identidade ocorre pois (B.5) representa as

[1] Assumir que as forças permaneçam inalteradas significa assumir $\delta \mathsf{x}$ de tamanho conveniente para que a hipótese seja válida.

equações de Euler-Lagrange de (B.13) considerando as condições de contorno (B.7) e (B.8).

Entretanto, para que um verdadeiro princípio variacional seja estabelecido ainda é necessário mostrar que o lado esquerdo de (B.16) pode ser expresso como a variação de um funcional de x, o qual é então feito estacionário para todos os deslocamentos virtuais admissíveis $\delta\mathsf{x}$ por qualquer x respeitando (B.5) a (B.8). Com isso mostra-se que através de um enunciado variacional *a solução do conjunto de equações* (B.5) a (B.8) *passa a ser um ponto estacionário de um funcional de* x *em relação a qualquer deslocamento virtual admissível*.

Substituindo (B.15) em (B.13), tem-se

$$\int_{\mathcal{B}} \delta W \; dV - \int_{\Gamma 2} \bar{\mathbf{t}} \cdot \delta\mathsf{x} \; dA - \int_{\mathcal{B}} \mathbf{b}_0 \cdot \delta\mathsf{x} \; dV = 0.$$

Se $\mathbf{b}$ for uma força conservativa, i.e.,

$$\mathbf{b}_0 = -\nabla_{\mathsf{x}}\phi \quad \Rightarrow \quad -\delta\phi = \mathbf{b}_0 \cdot \delta\mathsf{x},$$

sendo ϕ uma força escalar de x. Então,

$$\delta \int_{\mathcal{B}} (W + \phi) \; dV - \int_{\Gamma 2} \bar{\mathbf{t}} \cdot \delta\mathsf{x} \; dA = 0.$$

Se $\bar{\mathbf{t}}$ é independente de $\nabla\mathsf{x}$ e depende do movimento apenas através de x, então tem-se o mesmo caso anterior,

$$\bar{\mathbf{t}} = -\nabla_{\mathsf{x}}\psi \quad \Rightarrow \quad -\delta\psi = \bar{\mathbf{t}} \cdot \delta\mathsf{x},$$

ou seja,

$$\delta \left\{ \int_{\mathcal{B}} (W + \phi) \; dV + \int_{\Gamma 2} \psi \; dA \right\} = 0. \tag{B.17}$$

Se o carregamento é independente da deformação, dependendo apenas do ponto material $\mathbf{X}$ no qual está especificado, então $\psi = -\bar{\mathbf{t}} \cdot \mathsf{x}$. Portanto,

$$\delta \left\{ \int_{\mathcal{B}} (W + \phi) \; dV + \int_{\Gamma 2} (-\bar{\mathbf{t}} \cdot \mathsf{x}) \; dA \right\} = 0 \tag{B.18}$$

Se o carregamento é dependente da deformação através de uma relação conhecida com a orientação do contorno, então $\bar{\mathbf{t}} = \mathbf{A}\left(\mathbf{X}, \mathsf{x}, \nabla\mathsf{x}\right)\mathbf{n}$. Através do teorema da divergência

$$\int_{\Gamma 2} \bar{\mathbf{t}} \cdot \delta\mathsf{x} \; dA = \int_{\mathcal{B}} \left\{ (\mathrm{Div}\, \mathbf{A}) \cdot \delta\mathsf{x} + \mathbf{A} \cdot \delta\mathbf{F} \right\} \; dV,$$

mostrando a existência de um funcional $\varphi(\mathbf{X}, \mathsf{x}, \nabla\mathsf{x})$ tal que

$$\delta\varphi = (\mathrm{Div}\,\mathbf{A}) \cdot \delta\mathsf{x} + \mathbf{A} \cdot \delta\mathbf{F}$$

Por fim,

$$\delta\left\{\int_{\mathcal{B}} (W + \phi - \varphi)\; dV\right\} = 0. \tag{B.19}$$

Conclui-se então que

$$\boxed{\delta\Pi(\mathsf{x}, \delta\mathsf{x}) = 0,} \tag{B.20}$$

resume os enunciados (B.17), (B.18) e (B.19), considerando

$$\boxed{\Pi(\mathsf{x}) = \int_{\mathcal{B}} W(\nabla\mathsf{x}(\mathbf{X}))\; dV + \int_{\mathcal{B}} \phi(\mathsf{x}(\mathbf{X}))\; dV + \int_{\Gamma 2} \psi(\mathbf{X}, \mathsf{x}(\mathbf{X}), \nabla\mathsf{x}(\mathbf{X}))\; dA,} \tag{B.21}$$

que é denominado *funcional de energia potencial*

A seguinte notação será utilizada para a ação das forças externas:

$$W_{ext}(\mathbf{X}, \mathsf{x}(\mathbf{X}), \nabla\mathsf{x}(\mathbf{X})) = -\int_{\mathcal{B}} \phi(\mathsf{x}(\mathbf{X}))\; dV - \int_{\Gamma 2} \psi(\mathbf{X}, \mathsf{x}(\mathbf{X}), \nabla\mathsf{x}(\mathbf{X}))\; dA, \tag{B.22}$$

de forma que

$$\Pi(\mathsf{x}) = \int_{\mathcal{B}} W(\nabla\mathsf{x}(\mathbf{X}))\; dV - W_{ext}(\mathbf{X}, \mathsf{x}(\mathbf{X}), \nabla\mathsf{x}(\mathbf{X}))$$

Escrevendo a variação de forma rigorosa, tem-se

$$\delta\Pi(\mathsf{x}, \delta\mathsf{x}) = D\Pi(\mathsf{x})[\delta\mathsf{x}] = 0 \qquad \mathsf{x} = \bar{\mathsf{x}}, \quad \forall\delta\mathsf{x} = \mathbf{0} \text{ em } \Gamma^1, \tag{B.23}$$

que define claramente x como um ponto estacionário de Π para toda variação admissível $\delta\mathsf{x}$. Esse princípio variacional é denominado *princípio da energia potencial estacionária.*

Para as hipóteses assumidas de problema hiperelástico sem restrições internas, a solução x de $\delta\Pi = 0$ é um ponto estacionário específico correspondendo ao *ponto de mínimo* de Π. Em outras palavras, x é a solução de

$$\boxed{\Pi(\mathsf{x}) = \inf_{\chi\in\mathcal{X}} \Pi(\chi)} \tag{B.24}$$

sendo $\mathcal{X}$ um espaço funcional (de Hilbert) conveniente, com $\mathbf{x} = \bar{\mathsf{x}}$ em Γ^1. O espaço $\mathcal{X}$ é denominado genericamente de *espaço de alterações de configuração admissíveis.* Num caso prático de grande interesse (que será a situação usual das aplicações nesse texto), esse espaço é definifo como

$$\mathcal{X} = \left\{ \mathsf{x} : \mathcal{B} \rightarrow \Re^{\dim} | \ \mathsf{x} \in \left[\mathfrak{H}^1 \left(\mathcal{B} \right) \right]^{\dim}; \mathsf{x}|_{\Gamma 1} = \bar{\mathsf{x}} \right\} \tag{B.25}$$

Da mesma forma, o *espaço de variações cinematicamente admissíveis* $\mathcal{V}$ é definido como

$$\mathcal{V} = \left\{ \delta\mathsf{x} : \mathcal{B} \rightarrow \Re^{\dim} | \ \delta\mathsf{x} \in \left[\mathfrak{H}^1 \left(\mathcal{B} \right) \right]^{\dim}; \delta\mathsf{x}|_{\Gamma 1} = \mathbf{0} \right\} \tag{B.26}$$

sendo $\dim \leq 3$ a dimensão do domínio do problema e $\mathfrak{H}^1 \left(\mathcal{B} \right)$ o espaço de Sobolev de funções com derivadas de primeira ordem quadrado integráveis; Γ^1 é a região de $\partial\mathcal{B}$ na qual são aplicadas condições de contorno essenciais.

Observa-se, entretanto, que $\forall \mathsf{x} \in \mathcal{X}$ pode ser escrito como

$$\mathsf{x} = \mathsf{x}^C + \mathsf{x}^H, \tag{B.27}$$

sendo $\mathsf{x}^C \in \mathcal{X}$ um elemento arbitrário fixo e $\mathsf{x}^H \in \mathcal{V}$. Portanto, todo espaço $\mathcal{X}$ pode ser gerado a partir de elementos de $\mathcal{V}$ usando (B.27). Assim, a definição (B.25) do espaço $\mathcal{X}$ é apenas formal pois o domínio do problema (B.23) é, na verdade, apenas $\mathcal{V}$ pois

$$\delta\Pi \left(\mathsf{x}, \delta\mathsf{x} \right) = \delta\Pi \left(\mathsf{x}^C + \mathsf{x}^H, \delta\mathsf{x} \right) = 0 \qquad \mathsf{x}^H \in \mathcal{V}, \quad \forall \delta\mathsf{x} \in \mathcal{V},$$

sendo $\mathsf{x}^C \in \mathcal{X}$ uma função conhecida.

$\mathcal{V}$ é um espaço de Hilbert reflexivo, ou seja, $\mathcal{V} = \mathcal{V}^*$ e portanto, o emparceiramento de dualidade para o cálculo de derivadas é simplesmente o produto interno do espaço funcional $\mathfrak{L}_2$ das funções quadrado integráveis, isto é, $\langle \boldsymbol{\xi}_1, \boldsymbol{\xi}_2 \rangle = \int_{\mathcal{B}} \boldsymbol{\xi}_1 \cdot \boldsymbol{\xi}_2 \ dV \leq \infty$.

Modificações do Princípio Variacional Padrão

Procedimentos genéricos e complexos de transformação de problemas variacionais são fornecidos pela análise convexa e teoria de dualidade (Brezzi e Fortin, 1991). Entretanto para o tratamento da classe de problemas em questão serão necessários apenas alguns resultados básicos e conceitos gerais de dualidade.

A idéia fundamental da teoria de dualidade é que uma função convexa pode ser representada por sua família de funções tangentes afins. Como resultado da análise convexa, dada uma função contínua e convexa $f \left(\mathbf{a} \right)$ com domínio no espaço $\mathcal{A}$, pode-se definir a função conjugada $f^* \left(\mathbf{a}^* \right)$ no espaço dual

$\mathcal{A}'$ de $\mathcal{A}$ como

$$f^*(\mathbf{a}^*) = \sup_{\mathbf{a}\in\mathcal{A}} \Big[\langle \mathbf{a}, \mathbf{a}^* \rangle_{\mathcal{A}\times\mathcal{A}'} - f(\mathbf{a}) \Big], \tag{B.28}$$

sendo $f^*(\mathbf{a}^*)$ também convexa e contínua.

A importância desse resultado está no fato de que se pode então definir $f(\mathbf{a})$ a partir de $f^*(\mathbf{a}^*)$ aplicando a seguinte fórmula simétrica a (B.28)

$$f(\mathbf{a}) = \sup_{\mathbf{a}^*\in\mathcal{A}} \Big[\langle \mathbf{a}^*, \mathbf{a} \rangle_{\mathcal{A}'\times\mathcal{A}} - f^*(\mathbf{a}^*) \Big].$$

Considere a existência de um restrição do tipo

$$h(\nabla\chi) = 0 \qquad h : \mathrm{Lin}^+ \rightarrow \Re$$

para o problema (B.5)–(B.8).

Como as componentes de $\nabla \mathsf{x}$ não são mais independentes, a equação da energia deve ser alterada de forma a considerar a restrição na solução do problema estrutural. Para escrever o problema no mesmo estilo da seção anterior, o problema restrito é reescrito como não-restrito introduzindo-se a função característica ϱ definida como,

$$\varrho(v|\{0\}) = \begin{cases} 0 & \text{se} \quad v = 0 \\ +\infty & \text{nos outros casos} \end{cases} \tag{B.29}$$

Então,

$$\Pi(\chi) = \int_{\mathcal{B}} \bar{W}(\nabla\chi)\ dV + \int_{\mathcal{B}} \varrho(h(\nabla\chi)\,|\,\{0\})\ dV - W_{ext}(\mathbf{X}, \chi(\mathbf{X}), \nabla\chi(\mathbf{X})).$$

Claramente,

$$\int_{\mathcal{B}} \varrho(h(\nabla\chi)\,|\,\{0\})\ dV = \sup_{q\in\mathcal{Q}} \int_{\mathcal{B}} qh(\nabla\chi)\ dV \tag{B.30}$$

que define a exigência de respeito à restrição, sendo $\mathcal{Q} = \mathfrak{L}_2(\mathcal{B})$.

Logo, impondo-se *simultaneamente* a condição de mínima energia potencial e a restrição, observa-se que o problema original de minimização é convertido em um problema de sela com condição inf-sup (Brenner e Scott, 1994; Brezzi e Fortin, 1991), ou seja,

$$\Pi(\mathsf{x}, p) = \inf_{\chi\in\mathcal{V}} \sup_{q\in\mathcal{Q}} \int_{\mathcal{B}} \bar{W}(\nabla\chi)\ dV + \int_{\mathcal{B}} qh(\nabla\chi)\ dV - W_{ext}(\mathbf{X}, \chi(\mathbf{X}), \nabla\chi(\mathbf{X})), \tag{B.31}$$

sendo

$$\Pi(\chi) = \sup_{q \in \mathcal{Q}} \int_{\mathcal{B}} \left[\bar{W}(\nabla\chi) + qh(\nabla\chi)\right] \; dV - W_{ext}(\mathbf{X}, \chi(\mathbf{X}), \nabla\chi(\mathbf{X})) \tag{B.32}$$

Se o conjunto de alterações de configuração admissíveis for extendido, não estando mais restrito a espaços funcionais respeitando as condições essenciais, então o termo

$$-\int_{\Gamma 1} (\chi - \bar{\mathsf{x}}) \cdot \mathbf{Tn} \; dA$$

é incluído em (B.21):

$$\begin{aligned}\Pi(\chi) \quad = \quad & \sup_{q \in \mathcal{Q}} \int_{\mathcal{B}} \left[\bar{W}(\nabla\chi) + qh(\nabla\chi)\right] \; dV + \int_{\mathcal{B}} \phi(\chi) \; dV + \int_{\Gamma 2} \psi(\mathbf{X}, \chi, \nabla\chi) \; dA \\ & - \int_{\Gamma 1} (\chi - \bar{\mathsf{x}}) \cdot \mathbf{Tn} \; dA\end{aligned}$$

O termo $\mathbf{Tn}$, por ser uma incógnita do problema, pode ser tratado como uma variável adicional $\boldsymbol{\lambda}$ (multiplicador de Lagrange) da seguinte forma:

$$\begin{aligned}\Pi(\chi) \quad = \quad & \sup_{q \in \mathcal{Q}} \int_{\mathcal{B}} \left[\bar{W}(\nabla\chi) + qh(\nabla\chi)\right] \; dV + \int_{\mathcal{B}} \phi(\chi) \; dV + \int_{\Gamma 2} \psi(\mathbf{X}, \chi, \nabla\chi) \; dA \\ & - \int_{\Gamma 1} (\chi - \bar{\mathsf{x}}) \cdot \boldsymbol{\lambda} \; dA\end{aligned} \tag{B.33}$$

O enunciado (B.33) é utilizado, por exemplo, quando se formula o problema através do método meshless RKPM (Kim, 1999), cujas funções de forma padrão não respeitam as condições de contorno essenciais. Entretanto, a implementação numérica de (B.33) pode originar matrizes globais mal-condicionadas e por esse motivo foi introduzido um artifício de transformação que gera um novo espaço funcional respeitando as condições de contorno a partir do espaço original de funções meshless (Chen *et al.*, 1997b).

Utilizando elementos finitos geralmente é possível definir espaços funcionais incorporando as condições essenciais (Brenner e Scott, 1994; Reddy, 1986) e portanto os enunciados (B.31) e (B.32) são os pontos de partida para a formulação de problemas hiperelásticos incompressíveis e aproximadamente incompressíveis nessa classe de métodos.

Observa-se que essa abordagem é similar à solução de problemas de otimização sujeitos a restrições pelo método de penalidades (Arora, 1989; Bonet e Wood, 1997; Luenberger, 1989), onde o problema restrito é convertido convertido numa seqüência de subproblemas não-restritos modificados.

Apêndice C

Formas Discretas de Expressões de Análise de Resposta e de Sensibilidade

Nesse apêndice, apresentam-se as expressões discretas das formas contínuas das análises de resposta e sensibilidade de hiperelasticidade discutidas nos Capítulos 2 e 3.

C.1 Não-Linearidades Geométricas sem Restrições Internas

C.1.1 Discretização da Análise de Resposta

Da forma da energia interna,

$$a(\mathsf{x}, \delta\mathsf{x}) = \int_{\mathcal{B}} \mathbf{S} \cdot \delta\mathbf{E}\ dV = \int_{\mathcal{B}} \mathbf{S}(\mathsf{x}) \cdot D\mathbf{E}[\delta\mathsf{x}]\ dV, \tag{C.1}$$

obtém-se a sua derivada como

$$\begin{aligned} Da(\mathsf{x}, \delta\mathsf{x})[\Delta\mathbf{u}] &= \delta a(\mathsf{x};\ \Delta\mathbf{u}, \delta\mathsf{x}) = \int_{\mathcal{B}} D(\mathbf{S} \cdot \delta\mathbf{E})[\Delta\mathbf{u}]\ dV \\ &= \int_{\mathcal{B}} \mathsf{C}(\mathsf{x}) : D\mathbf{E}[\Delta\mathbf{u}] \cdot \delta\mathbf{E}\ dV + \int_{\mathcal{B}} \mathbf{S} \cdot D\delta\mathbf{E}[\Delta\mathbf{u}]\ dV, \end{aligned} \tag{C.2}$$

sendo

$$D\mathbf{E}[\Delta\mathbf{u}] = \Delta\mathbf{E} = \frac{1}{2}\left(\mathbf{F}^T \nabla(\Delta\mathbf{u}) + \nabla(\Delta\mathbf{u})^T \mathbf{F}\right),$$

$$D\delta\mathbf{E}\left[\Delta\mathbf{u}\right]=\frac{1}{2}\left(D\mathbf{F}^{T}\left[\Delta\mathbf{u}\right]\delta\mathbf{F}+\delta\mathbf{F}^{T}D\mathbf{F}\left[\Delta\mathbf{u}\right]\right)=\frac{1}{2}\left(\nabla(\Delta\mathbf{u})^{T}\delta\mathbf{F}+\delta\mathbf{F}^{T}\nabla(\Delta\mathbf{u})\right).$$

Considerando um único elemento finito e, escreve-se a variação do tensor de deformação finita como

$$[\Delta\mathbf{E}]=\mathbf{B}_{N}\Delta\hat{\mathbf{u}}_{e},\qquad\qquad \mathbf{B}_{N}\equiv\mathbf{B}_{N}\left(\mathbf{X},\mathbf{u}\right).$$

Com isso, discretiza-se o integrando do primeiro termo de (C.2) da seguinte forma:

$$\mathsf{C}\left(\mathsf{x}\right):D\mathbf{E}\left[\Delta\mathbf{u}\right]\cdot D\mathbf{E}\left[\delta\mathsf{x}\right]=\left[\mathsf{C}\left(\mathsf{x}\right)\right]\mathbf{B}_{N}\Delta\hat{\mathbf{u}}_{e}\cdot\mathbf{B}_{N}\delta\hat{\mathsf{x}}_{e}=\mathbf{B}_{N}^{T}\left[\mathsf{C}\left(\mathsf{x}\right)\right]\mathbf{B}_{N}\Delta\hat{\mathbf{u}}_{e}\cdot\delta\hat{\mathsf{x}}_{e},$$

sendo $[\mathsf{C}(\mathsf{x})]$ uma matriz 6×6 que corresponde à forma algorítmica de $\mathsf{C}(\mathsf{x})$, tensor de elasticidade tangente.

Usando a simetria do segundo tensor de Piola-Kirchhoff, tem-se que

$$\begin{aligned}\mathbf{S}\left(\mathsf{x}\right)\cdot\left[\frac{1}{2}\left(\nabla\Delta\mathbf{u}^{T}\nabla\delta\mathsf{x}+\nabla\delta\mathsf{x}^{T}\nabla\Delta\mathbf{u}\right)\right] &= \mathbf{S}\left(\mathsf{x}\right)\cdot\left(\nabla(\Delta\mathbf{u})^{T}\nabla\delta\mathsf{x}\right)^{S}=\mathbf{S}\left(\mathsf{x}\right)\cdot\left(\nabla(\Delta\mathbf{u})^{T}\nabla\delta\mathsf{x}\right)\\ &= \left(\nabla\Delta\mathbf{u}\right)\mathbf{S}\left(\mathsf{x}\right)\cdot\nabla\delta\mathsf{x},\end{aligned}$$

de onde se obtém o formato para implementação

$$\nabla(\Delta\mathbf{u})\mathsf{S}\left(\mathsf{x}\right)\cdot\nabla\delta\mathsf{x}=\boldsymbol{\Xi}\left(\mathsf{x}\right)\mathbf{B}_{G}\Delta\hat{\mathbf{u}}_{e}\cdot\mathbf{B}_{G}\delta\hat{\mathsf{x}}_{e}=\mathbf{B}_{G}^{T}\boldsymbol{\Xi}\left(\mathsf{x}\right)\mathbf{B}_{G}\Delta\hat{\mathbf{u}}_{e}\cdot\delta\hat{\mathsf{x}}_{e},\tag{C.3}$$

sendo $\boldsymbol{\Xi}(\mathsf{x})$ uma matriz simétrica formada a partir das componentes de $\mathbf{S}(\mathsf{x})$.

Por fim, falta discretizar (C.1) e assim obter a expressão das forças internas. Seguindo o mesmo procedimento anterior, tem-se que

$$\mathbf{S}\cdot\delta\mathbf{E}=\{\mathbf{S}\}\cdot\mathbf{B}_{N}\delta\hat{\mathsf{x}}_{e}=\mathbf{B}_{N}^{T}\{\mathbf{S}\}\cdot\delta\hat{\mathsf{x}}_{e}$$

sendo $\{\mathbf{S}\}$ vetor com a parte simétrica do tensor $\mathbf{S}$. As matrizes $\mathbf{B}_N$ e $\mathbf{B}_G$ são mostradas na próxima seção.

O sistema global de uma iteração de Newton-Raphson

$$\boxed{{}^{(k)}\left[\mathbf{K}\right]_{n+1}\ {}^{(k+1)}\left[\Delta\hat{\mathbf{u}}\right]=\left[\hat{\mathbf{f}}_{ext}\right]_{n+1}-{}^{(k)}\left[\hat{\mathbf{f}}_{int}\right]_{n+1},}\tag{C.4}$$

é obtido com o processo usual de elementos finitos de montagem a partir das matrizes e vetores elementares, sendo

$$\begin{aligned} {}^{(k)}\left[\mathbf{K}\right]_{n+1}\ {}^{(k+1)}\left[\Delta\hat{\mathbf{u}}\right] &= {}^{(k)}\left[\mathsf{A}_{e=1}^{Ne}\mathbf{K}_e\right]_{n+1}\ {}^{(k+1)}\left[\Delta\hat{\mathbf{u}}\right] \\ &= \mathsf{A}_{e=1}^{Ne}\left\{\int_{\mathcal{B}_e} {}^{(k)}\left[\mathbf{B}_N^T\left[\mathsf{C}\left(\mathsf{x}\right)\right]\mathbf{B}_N\right]_{n+1} dV + \int_{\mathcal{B}_e} {}^{(k)}\left[\mathbf{B}_G^T\mathbf{\Xi}\left(\mathsf{x}\right)\mathbf{B}_G\right]_{n+1} dV\right\}\ {}^{(k+1)}\left[\Delta\hat{\mathbf{u}}\right] \end{aligned} \tag{C.5}$$

a matriz de rigidez tangente e

$${}^{(k)}\left[\hat{\mathbf{f}}_{int}\right]_{n+1} = \mathsf{A}_{e=1}^{Ne}\left\{\int_{\mathcal{B}_e} {}^{(k)}\left[\mathbf{B}_N^T\left\{\mathbf{S}\right\}\right]_{n+1} dV\right\} \tag{C.6}$$

o vetor de forças internas. $\left[\ \hat{\mathbf{f}}_{ext}\right]_{n+1}$ é o vetor obtido dos carregamentos externos no subpasso $n+1$:

$$\left[\ \hat{\mathbf{f}}_{ext}\right]_{n+1} = \left[\mathsf{A}_{e=1}^{Ne}\hat{\mathbf{f}}_{ext_e}\right]_{n+1} = \mathsf{A}_{e=1}^{Ne}\left\{\int_{\mathcal{B}_e}\mathbf{N}^T\left[\mathbf{b}_{n+1}\right]_e\ dV + \int_{\Gamma 2_e}\mathbf{N}^T\left[\bar{\mathbf{t}}_{n+1}\right]_e\ dA\right\} \tag{C.7}$$

C.1.2 Formas Matriciais

Elementos Sólidos Tridimensionais

Considere uma discretização em elementos finitos. Para um elemento e de Nn nós com coordenadas $\mathbf{X}_e^i$ e funções de forma N^i, $i,\ldots,Nn$, tem-se que em seu domínio (e numa aproximação paramétrica) as seguintes aproximações em termos de interpolações valores nodais por funções de forma são válidas:

$$\mathbf{X}_e\left(\zeta,\eta,\varsigma\right) = \sum_{i=1}^{Nn}\mathbf{X}_e^i N^i\left(\zeta,\eta,\varsigma\right), \tag{C.8}$$

$$\mathbf{u}_e = \sum_{i=1}^{Nn}\mathbf{u}_e^i N^i, \tag{C.9}$$

$$\mathbf{x}_e = \mathsf{x}\left(\mathbf{X}_e\right) = \mathbf{X}_e + \mathbf{u}_e = \sum_{i=1}^{Nn}\left(\mathbf{X}_e^i + \mathbf{u}_e^i\right)N^i = \sum_{i=1}^{Nn}\mathbf{x}_e^i N^i, \tag{C.10}$$

sendo ζ,η,ς as coordenadas do ponto material em termos das coordenadas naturais do elemento de referência $\mathcal{B}_{ref}$. Como comentado anteriormente, a triangularização $\mathcal{T}_h$ fornece $\mathcal{B}_e \rightarrow \mathcal{B}_{ref}$ inversível, sendo possível relacionar univocamente as coordenadas materiais $\mathbf{X}$ e as coordenadas naturais (e suas derivadas) dentro de cada elemento. Em outras palavras não é necessário fazer referência às coordenadas naturais no desenvolvimento a seguir.

Do ponto de vista da implementação é interessante expressar as mesmas interpolações de grandezas através de operações matriciais. No caso do deslocamento tem-se,

$$\mathbf{u}_e = \mathbf{N}\hat{\mathbf{u}}_e \tag{C.11}$$

$$\mathbf{N} = \begin{bmatrix} N^1 & 0 & 0 & N^2 & 0 & 0 & \cdots & N^{Nn} & 0 & 0 \\ 0 & N^1 & 0 & 0 & N^2 & 0 & \cdots & 0 & N^{Nn} & 0 \\ 0 & 0 & N^1 & 0 & 0 & N^2 & \cdots & 0 & 0 & N^{Nn} \end{bmatrix}, \tag{C.12}$$

$$\hat{\mathbf{u}}_e^T = \begin{bmatrix} u_1^1 & u_2^1 & u_3^1 & u_1^2 & u_2^2 & u_3^2 & \cdots & u_1^{Nn} & u_2^{Nn} & u_3^{Nn} \end{bmatrix}, \tag{C.13}$$

sendo $\mathbf{u}_e$ o deslocamento de um ponto material do elemento e, e $\hat{\mathbf{u}}_e$ o vetor de deslocamentos dos nós do elemento.

Faz-se o mesmo para o incremento de deslocamento, ou seja,

$$\Delta\mathbf{u}_e = \mathbf{N}\Delta\hat{\mathbf{u}}_e, \tag{C.14}$$

$$\Delta\hat{\mathbf{u}}_e^T = \begin{bmatrix} \Delta u_1^1 & \Delta u_2^1 & \Delta u_3^1 & \Delta u_1^2 & \Delta u_2^2 & \Delta u_3^2 & \cdots & \Delta u_1^{Nn} & \Delta u_2^{Nn} & \Delta u_3^{Nn} \end{bmatrix}. \tag{C.15}$$

Dado o gradiente material do campo de deslocamentos

$$\nabla\mathbf{u} = \frac{\partial u_i}{\partial X_j} = u_{i,j},$$

define-se sua forma discreta através do produto matricial

$$[\nabla\mathbf{u}]_e = \mathbf{B}_G\hat{\mathbf{u}}_e, \qquad \mathbf{B}_G \equiv \mathbf{B}_G(\mathbf{X}), \tag{C.16}$$

sendo

$$\mathbf{B}_G = \begin{bmatrix} N_{,1}^1 & 0 & 0 & N_{,1}^2 & 0 & 0 & \cdots & N_{,1}^{Nn} & 0 & 0 \\ N_{,2}^1 & 0 & 0 & N_{,2}^2 & 0 & 0 & \cdots & N_{,2}^{Nn} & 0 & 0 \\ N_{,3}^1 & 0 & 0 & N_{,3}^2 & 0 & 0 & \cdots & N_{,3}^{Nn} & 0 & 0 \\ 0 & N_{,1}^1 & 0 & 0 & N_{,1}^2 & 0 & \cdots & 0 & N_{,1}^{Nn} & 0 \\ 0 & N_{,2}^1 & 0 & 0 & N_{,2}^2 & 0 & \cdots & 0 & N_{,2}^{Nn} & 0 \\ 0 & N_{,3}^1 & 0 & 0 & N_{,3}^2 & 0 & \cdots & 0 & N_{,3}^{Nn} & 0 \\ 0 & 0 & N_{,1}^1 & 0 & 0 & N_{,1}^2 & \cdots & 0 & 0 & N_{,1}^{Nn} \\ 0 & 0 & N_{,2}^1 & 0 & 0 & N_{,2}^2 & \cdots & 0 & 0 & N_{,2}^{Nn} \\ 0 & 0 & N_{,3}^1 & 0 & 0 & N_{,3}^2 & \cdots & 0 & 0 & N_{,3}^{Nn} \end{bmatrix} \tag{C.17}$$

e

$$[\nabla \mathbf{u}]_e^T = \begin{bmatrix} u_{1,1} & u_{1,2} & u_{1,3} & u_{2,1} & u_{2,2} & u_{2,3} & u_{3,1} & u_{3,2} & u_{3,3} \end{bmatrix}. \tag{C.18}$$

$[\nabla \mathbf{u}]_e^T$ é a forma matricial para implemetação de $\nabla \mathbf{u}$ em um ponto material do elemento e.

O tensor de deformação infinitesimal

$$\boldsymbol{\epsilon} = \nabla \mathbf{u}^S = \frac{1}{2}\left(\frac{\partial u_i}{\partial X_j} + \frac{\partial u_j}{\partial X_i}\right) = \frac{1}{2}\left(u_{i,j} + u_{j,i}\right)$$

é usualmente expresso em implementações como[1]

$$[\boldsymbol{\epsilon}]_e = \mathbf{B}_L \hat{\mathbf{u}}_e, \qquad \mathbf{B}_L \equiv \mathbf{B}_L\left(\mathbf{X}\right), \tag{C.19}$$

$$\mathbf{B}_L = \begin{bmatrix} N_{,1}^1 & 0 & 0 & N_{,1}^2 & 0 & 0 & \cdots & N_{,1}^{Nn} & 0 & 0 \\ 0 & N_{,2}^1 & 0 & 0 & N_{,2}^2 & 0 & \cdots & 0 & N_{,2}^{Nn} & 0 \\ 0 & 0 & N_{,3}^1 & 0 & 0 & N_{,3}^2 & \cdots & 0 & 0 & N_{,3}^{Nn} \\ N_{,2}^1 & N_{,1}^1 & 0 & N_{,2}^2 & N_{,1}^2 & 0 & \cdots & N_{,2}^{Nn} & N_{,1}^{Nn} & 0 \\ N_{,3}^1 & 0 & N_{,1}^1 & N_{,3}^2 & 0 & N_{,1}^2 & \cdots & N_{,3}^{Nn} & 0 & N_{,1}^{Nn} \\ 0 & N_{,3}^1 & N_{,2}^1 & 0 & N_{,3}^2 & N_{,2}^2 & \cdots & 0 & N_{,3}^{Nn} & N_{,2}^{Nn} \end{bmatrix}, \tag{C.20}$$

$$\begin{aligned} [\boldsymbol{\epsilon}]_e^T &= \begin{bmatrix} \epsilon_{11} & \epsilon_{22} & \epsilon_{33} & 2\epsilon_{12} & 2\epsilon_{13} & 2\epsilon_{23} \end{bmatrix} \\ &= \begin{bmatrix} u_{1,1} & u_{2,2} & u_{3,3} & (u_{1,2}+u_{2,1}) & (u_{1,3}+u_{3,1}) & (u_{2,3}+u_{3,2}) \end{bmatrix}. \end{aligned} \tag{C.21}$$

Seguindo a mesma idéia para a variação do tensor de deformação finita tem-se que

$$D\mathbf{E}\left[\Delta \mathbf{u}\right] = \Delta \mathbf{E} = \frac{1}{2}\left(\mathbf{F}^T \nabla \Delta \mathbf{u} + \nabla \Delta \mathbf{u}^T \mathbf{F}\right),$$

$$\mathbf{F}^T \nabla \Delta \mathbf{u} = \left(\mathbf{I} + \nabla \mathbf{u}\right)^T \nabla \Delta \mathbf{u} = \left(\delta_{ni} + u_{n,i}\right)\left(\Delta u_{n,j}\right),$$

$$[\Delta \mathbf{E}]^T = \begin{bmatrix} \Delta E_{11} & \Delta E_{22} & \Delta E_{33} & 2\Delta E_{12} & 2\Delta E_{13} & 2\Delta E_{23} \end{bmatrix}, \tag{C.22}$$

$$\begin{aligned} \Delta E_{11} &= \Delta u_{1,1} + u_{n,1}\Delta u_{n,1}, \\ \Delta E_{22} &= \Delta u_{2,2} + u_{n,2}\Delta u_{n,2}, \end{aligned}$$

[1]O subscrito L em $\mathbf{B}_L$ indica matriz de deformação *linear*

$$
\begin{aligned}
\Delta E_{33} &= \Delta u_{3,3} + u_{n,3}\Delta u_{n,3}, \\
2\Delta E_{12} &= \Delta u_{1,2} + \Delta u_{2,1} + u_{n,1}\Delta u_{n,2} + u_{n,2}\Delta u_{n,1}, \\
2\Delta E_{13} &= \Delta u_{1,3} + \Delta u_{3,1} + u_{n,1}\Delta u_{n,3} + u_{n,3}\Delta u_{n,1}, \\
2\Delta E_{23} &= \Delta u_{2,3} + \Delta u_{3,2} + u_{n,2}\Delta u_{n,3} + u_{n,3}\Delta u_{n,2},
\end{aligned}
$$

$$
[\Delta \mathbf{E}]_e = \mathbf{B}_F \left[\nabla \Delta \mathbf{u}\right]_e, \qquad \mathbf{B}_F \equiv \mathbf{B}_F\left(\mathbf{X}, \mathbf{u}\right), \tag{C.23}
$$

$$
\mathbf{B}_F = \begin{bmatrix}
1+u_{1,1} & 0 & 0 & u_{2,1} & 0 & 0 & u_{3,1} & 0 & 0 \\
0 & u_{1,2} & 0 & 0 & 1+u_{2,2} & 0 & 0 & u_{3,2} & 0 \\
0 & 0 & u_{1,3} & 0 & 0 & u_{2,3} & 0 & 0 & 1+u_{3,3} \\
u_{1,2} & 1+u_{1,1} & 0 & 1+u_{2,2} & u_{2,1} & 0 & u_{3,2} & u_{3,1} & 0 \\
u_{1,3} & 0 & 1+u_{1,1} & u_{2,3} & 0 & u_{2,1} & 1+u_{3,3} & 0 & u_{3,1} \\
0 & u_{1,3} & u_{1,2} & 0 & u_{2,3} & 1+u_{2,2} & 0 & 1+u_{3,3} & u_{3,2}
\end{bmatrix},
$$

$$
[\nabla \Delta \mathbf{u}]_e^T = \begin{bmatrix} \Delta u_{1,1} & \Delta u_{1,2} & \Delta u_{1,3} & \Delta u_{2,1} & \Delta u_{2,2} & \Delta u_{2,3} & \Delta u_{3,1} & \Delta u_{3,2} & \Delta u_{3,3} \end{bmatrix},
$$

$$
[\nabla \Delta \mathbf{u}]_e = \mathbf{B}_G \Delta \hat{\mathbf{u}}_e,
$$

$$
[\Delta \mathbf{E}] = \mathbf{B}_F \left[\nabla \Delta \mathbf{u}\right]_e = \mathbf{B}_F \mathbf{B}_G \Delta \hat{\mathbf{u}}_e = \mathbf{B}_N \Delta \hat{\mathbf{u}}_e, \qquad \mathbf{B}_N \equiv \mathbf{B}_N\left(\mathbf{X}, \mathbf{u}\right). \tag{C.24}
$$

Na prática, entretanto, é mais conveniente calcular $\mathbf{B}_N$ através a soma

$$
\mathbf{B}_N = \mathbf{B}_L + \mathbf{B}_U
$$

sendo (Bathe, 1996)

$$
\mathbf{B}_U = \begin{bmatrix}
u_{1,1}N^1_{,1} & u_{2,1}N^1_{,1} & u_{3,1}N^1_{,1} & u_{1,1}N^2_{,1} & \cdots & u_{3,1}N^{Nn}_{,1} \\
u_{1,2}N^1_{,2} & u_{2,2}N^1_{,2} & u_{3,2}N^1_{,2} & u_{1,2}N^2_{,2} & \cdots & u_{3,2}N^{Nn}_{,2} \\
u_{1,3}N^1_{,3} & u_{2,3}N^1_{,3} & u_{3,3}N^1_{,3} & u_{1,3}N^2_{,3} & \cdots & u_{3,3}N^{Nn}_{,3} \\
u_{1,1}N^1_{,2}+u_{1,2}N^1_{,1} & u_{2,1}N^1_{,2}+u_{2,2}N^1_{,1} & u_{3,1}N^1_{,2}+u_{3,2}N^1_{,1} & u_{1,1}N^2_{,2}+u_{1,2}N^2_{,1} & \cdots & u_{3,1}N^{Nn}_{,2}+u_{3,2}N^{Nn}_{,1} \\
u_{1,1}N^1_{,3}+u_{1,3}N^1_{,1} & u_{2,1}N^1_{,3}+u_{2,3}N^1_{,1} & u_{3,1}N^1_{,3}+u_{3,3}N^1_{,1} & u_{1,1}N^2_{,3}+u_{1,3}N^2_{,1} & \cdots & u_{3,1}N^{Nn}_{,3}+u_{3,3}N^{Nn}_{,1} \\
u_{1,2}N^1_{,3}+u_{1,3}N^1_{,2} & u_{2,2}N^1_{,3}+u_{2,3}N^1_{,2} & u_{3,2}N^1_{,3}+u_{3,3}N^1_{,2} & u_{1,2}N^2_{,3}+u_{1,3}N^2_{,2} & \cdots & u_{3,2}N^{Nn}_{,3}+u_{3,3}N^{Nn}_{,2}
\end{bmatrix}
$$

Diferente da matriz $\mathbf{B}_L$, que depende somente da derivadas das funções de forma em relação à geometria não-deformada (expressando sua linearidade), a matriz $\mathbf{B}_N$ depende também do estado corrente de deformação através da matriz $\mathbf{B}_F$ (ou $\mathbf{B}_U$). Além disso, observa-se que quando $\hat{\mathbf{u}}_e = \mathbf{0}$, $\mathbf{B}_N = \mathbf{B}_L$.

Tem-se ainda os seguintes termos das expressões (C.5) e (C.6):

$$\{\mathbf{S}\}^T = \begin{bmatrix} S_{11} & S_{22} & S_{33} & S_{12} & S_{13} & S_{23} \end{bmatrix} \tag{C.25}$$

e

$$\boldsymbol{\Xi}(\mathsf{x}) = \begin{bmatrix} \mathbf{S}(\mathsf{x}) & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{S}(\mathsf{x}) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{S}(\mathsf{x}) \end{bmatrix} \tag{C.26}$$

As seguintes expressões também são usadas na implementação:

$$\begin{aligned} \det \mathbf{F} &= \det(\mathbf{I} + \nabla \mathbf{u}) = \\ &= \begin{vmatrix} (1+u_{1,1}) & u_{1,2} & u_{1,3} \\ u_{2,1} & (1+u_{2,2}) & u_{2,3} \\ u_{3,1} & u_{3,2} & (1+u_{3,3}) \end{vmatrix} = \begin{array}{l} (1+u_{1,1})(1+u_{2,2})(1+u_{3,3}) + u_{1,2}u_{2,3}u_{3,1} \\ +u_{1,3}u_{2,1}u_{3,2} - u_{1,3}(1+u_{2,2})u_{3,1} \\ -u_{1,2}u_{2,1}(1+u_{3,3}) - (1+u_{1,1})u_{2,3}u_{3,2} \end{array} \end{aligned}$$

$$\begin{aligned} E_{ij} &= \frac{1}{2}(u_{i,j} + u_{j,i}) + \frac{1}{2}(u_{m,i}u_{m,j}) \\ E_{11} &= u_{1,1} + \frac{1}{2}(u_{1,1}u_{1,1} + u_{2,1}u_{2,1} + u_{3,1}u_{3,1}) \\ E_{22} &= u_{2,2} + \frac{1}{2}(u_{1,2}u_{1,2} + u_{2,2}u_{2,2} + u_{3,2}u_{3,2}) \\ E_{33} &= u_{3,3} + \frac{1}{2}(u_{1,3}u_{1,3} + u_{2,3}u_{2,3} + u_{3,3}u_{3,3}) \\ E_{12} &= \frac{1}{2}(u_{1,2} + u_{2,1}) + \frac{1}{2}(u_{1,1}u_{1,2} + u_{2,1}u_{2,2} + u_{3,1}u_{3,2}) \\ E_{13} &= \frac{1}{2}(u_{1,3} + u_{3,1}) + \frac{1}{2}(u_{1,1}u_{1,3} + u_{2,1}u_{2,3} + u_{3,1}u_{3,3}) \\ E_{23} &= \frac{1}{2}(u_{2,3} + u_{3,2}) + \frac{1}{2}(u_{1,2}u_{1,3} + u_{2,2}u_{2,3} + u_{3,2}u_{3,3}) \end{aligned}$$

Elementos Bidimensionais: Tensão e Deformação Planas e Sólido Axissimétrico

Redefinindo as matrizes da seção anterior para o caso de elementos bidimensionais[2] tem-se:

1. Interpolação dos campos de resposta

$$\mathbf{u}_e = \mathbf{N}\hat{\mathbf{u}}_e \tag{C.27}$$

[2]A última linha nas matrizes $\mathbf{B}_G$, $\mathbf{B}_L$ e $\mathbf{B}_U$ é exclusiva para sólido axissimétrico. Nesse caso, $\bar{X}_1$ é a coordenada radial do ponto material.

$$\Delta \mathbf{u}_e = \mathbf{N} \Delta \hat{\mathbf{u}}_e, \tag{C.28}$$

$$\mathbf{N} = \begin{bmatrix} N^1 & 0 & 0 & N^2 & 0 & 0 & \cdots & N^{Nn} & 0 & 0 \\ 0 & N^1 & 0 & 0 & N^2 & 0 & \cdots & 0 & N^{Nn} & 0 \end{bmatrix}, \tag{C.29}$$

$$\hat{\mathbf{u}}_e^T = \begin{bmatrix} u_1^1 & u_2^1 & u_1^2 & u_2^2 & \cdots & u_1^{Nn} & u_2^{Nn} \end{bmatrix}, \tag{C.30}$$

$$\Delta \hat{\mathbf{u}}_e^T = \begin{bmatrix} \Delta u_1^1 & \Delta u_2^1 & \Delta u_1^2 & \Delta u_2^2 & \cdots & \Delta u_1^{Nn} & \Delta u_2^{Nn} \end{bmatrix}. \tag{C.31}$$

2. Gradiente

$$[\nabla \mathbf{u}]_e = \mathbf{B}_G \hat{\mathbf{u}}_e, \qquad \mathbf{B}_G \equiv \mathbf{B}_G (\mathbf{X}), \tag{C.32}$$

$$\mathbf{B}_G = \begin{bmatrix} N_{,1}^1 & 0 & N_{,1}^2 & 0 & \cdots & N_{,1}^{Nn} & 0 \\ N_{,2}^1 & 0 & N_{,2}^2 & 0 & \cdots & N_{,2}^{Nn} & 0 \\ 0 & N_{,1}^1 & 0 & N_{,1}^2 & \cdots & 0 & N_{,1}^{Nn} \\ 0 & N_{,2}^1 & 0 & N_{,2}^2 & \cdots & 0 & N_{,2}^{Nn} \\ \frac{N^1}{X_1} & 0 & \frac{N^2}{X_1} & 0 & \cdots & \frac{N^{Nn}}{X_1} & 0 \end{bmatrix}, \tag{C.33}$$

$$[\nabla \mathbf{u}]_e^T = \begin{bmatrix} u_{1,1} & u_{1,2} & u_{2,1} & u_{2,2} & u_{3,3} \end{bmatrix}. \tag{C.34}$$

3. Deformação Linear

$$[\boldsymbol{\epsilon}]_e = \mathbf{B}_L \hat{\mathbf{u}}_e, \qquad \mathbf{B}_L \equiv \mathbf{B}_L (\mathbf{X}), \tag{C.35}$$

$$\mathbf{B}_L = \begin{bmatrix} N_{,1}^1 & 0 & N_{,1}^2 & 0 & \cdots & N_{,1}^{Nn} & 0 \\ 0 & N_{,2}^1 & 0 & N_{,2}^2 & \cdots & 0 & N_{,2}^{Nn} \\ N_{,2}^1 & N_{,1}^1 & N_{,2}^2 & N_{,1}^2 & \cdots & N_{,2}^{Nn} & N_{,1}^{Nn} \\ \frac{N^1}{X_1} & 0 & \frac{N^2}{X_1} & 0 & \cdots & \frac{N^{Nn}}{X_1} & 0 \end{bmatrix}, \tag{C.36}$$

$$[\boldsymbol{\epsilon}]_e^T = \begin{bmatrix} \epsilon_{11} & \epsilon_{22} & 2\epsilon_{12} & \epsilon_{33} \end{bmatrix} = \begin{bmatrix} u_{1,1} & u_{2,2} & (u_{1,2} + u_{2,1}) & u_{3,3} \end{bmatrix}. \tag{C.37}$$

4. Deformação Finita

$$[\Delta \mathbf{E}]^T = \begin{bmatrix} \Delta E_{11} & \Delta E_{22} & 2\Delta E_{12} & \Delta E_{33} \end{bmatrix}, \tag{C.38}$$

$$[\Delta \mathbf{E}] = \mathbf{B}_N \Delta \hat{\mathbf{u}}_e, \qquad \mathbf{B}_N \equiv \mathbf{B}_N (\mathbf{X}, \mathbf{u}). \tag{C.39}$$

$$\mathbf{B}_N = \mathbf{B}_L + \mathbf{B}_U,$$

$$\mathbf{B}_U = \begin{bmatrix} u_{1,1} N^1_{,1} & u_{2,1} N^1_{,1} & \cdots & u_{1,1} N^{Nn}_{,1} & u_{2,1} N^{Nn}_{,1} \\ u_{1,2} N^1_{,2} & u_{2,2} N^1_{,2} & \cdots & u_{1,2} N^{Nn}_{,2} & u_{2,2} N^{Nn}_{,2} \\ u_{1,1} N^1_{,2} + u_{1,2} N^1_{,1} & u_{2,1} N^1_{,2} + u_{2,2} N^1_{,1} & \cdots & u_{1,1} N^{Nn}_{,2} + u_{1,2} N^{Nn}_{,1} & u_{2,1} N^{Nn}_{,2} + u_{2,2} N^{Nn}_{,1} \\ u_{3,3} \frac{N^1}{X_1} & 0 & \cdots & u_{3,3} \frac{N^{Nn}}{X_1} & 0 \end{bmatrix}.$$

5. Formas algorítmicas do Tensor de Piola-Kirchhoff

$$\{\mathbf{S}\}^T = \begin{bmatrix} S_{11} & S_{22} & S_{12} & S_{33} \end{bmatrix},$$

$$\boldsymbol{\Xi}(\mathsf{x}) = \begin{bmatrix} S_{11} & S_{12} & 0 & 0 & 0 \\ S_{21} & S_{22} & 0 & 0 & 0 \\ 0 & 0 & S_{11} & S_{12} & 0 \\ 0 & 0 & S_{21} & S_{22} & 0 \\ 0 & 0 & 0 & 0 & S_{33} \end{bmatrix}. \tag{C.40}$$

C.1.3 Discretização das Expressões de Análise de Sensibilidade

As expressões de análise de sensibilidade são discretizadas seguindo o mesmo procedimento aplicado nas expressões de análise de resposta.

Sensibilidade à Forma

$$\boxed{\delta a\left(\mathsf{x}_{n+1}; \dot{\mathbf{u}}_{n+1}, \delta \mathsf{x}\right) = l'_{Vn+1}(\delta \mathsf{x}) - a'_V\left(\mathsf{x}_{n+1}, \delta \mathsf{x}\right).}$$

$$a'_V(\mathsf{x}, \delta \mathsf{x}) = \int_{\mathcal{B}} \mathsf{C}(\mathsf{x}) : \mathbf{E}_{\mathbf{V}}(\mathbf{u}) \cdot \delta \mathbf{E} + \mathbf{S} \cdot \delta \mathbf{E}_{\mathbf{V}}(\mathbf{u}) + \mathbf{S} \cdot \delta \mathbf{E} \, \mathrm{Div}\, \mathbf{V} \; dV,$$

$$\mathbf{E}_{\mathbf{V}}(\mathbf{u}) = -\frac{1}{2}\left[(\nabla \mathbf{u} \nabla \mathbf{V})^T \mathbf{F} + \mathbf{F}^T (\nabla \mathbf{u} \nabla \mathbf{V})\right],$$

$$\begin{aligned} \delta \mathbf{E}_{\mathbf{V}}(\mathbf{u}) &= \frac{1}{2}\left[\delta \mathbf{F}^T (-\nabla \mathbf{u} \nabla \mathbf{V}) + (-\nabla \mathbf{u} \nabla \mathbf{V})^T \delta \mathbf{F}\right] + \frac{1}{2}\left[(-\nabla \delta \mathbf{u} \nabla \mathbf{V})^T \mathbf{F} + \mathbf{F}^T (-\nabla \delta \mathbf{u} \nabla \mathbf{V})\right] \\ &= \left[(-\nabla \mathbf{u} \nabla \mathbf{V})^T \delta \mathbf{F}\right]^s + \left[\mathbf{F}^T (-\nabla \delta \mathbf{u} \nabla \mathbf{V})\right]^s. \end{aligned}$$

No domínio de um elemento e:

$$\begin{aligned}\mathsf{C}(\mathsf{x}):\mathbf{E}_{\mathbf{V}}(\mathbf{u})\cdot\delta\mathbf{E} &= -\left[\mathsf{C}(\mathsf{x})\right]\mathbf{B}_F\mathbf{B}_{GV}\hat{\mathbf{u}}_e\cdot\mathbf{B}_N\delta\hat{\mathsf{x}}_e = \\ &= -\left[\mathsf{C}(\mathsf{x})\right]\mathbf{B}_{NV}\hat{\mathbf{u}}_e\cdot\mathbf{B}_N\delta\hat{\mathsf{x}}_e = -\mathbf{B}_N^T\left[\mathsf{C}(\mathsf{x})\right]\mathbf{B}_{NV}\hat{\mathbf{u}}_e\cdot\delta\hat{\mathsf{x}}_e,\end{aligned} \tag{C.41}$$

$$\mathbf{S}\cdot\delta\mathbf{E}_{\mathbf{V}}(\mathbf{u}) = \mathbf{S}\cdot\left[\left(-\nabla\mathbf{u}\nabla\mathbf{V}\right)^T\delta\mathbf{F}\right]^s + \mathbf{S}\cdot\left[\mathbf{F}^T\left(-\nabla\delta\mathbf{u}\nabla\mathbf{V}\right)\right]^s, \tag{C.42}$$

$$\begin{aligned}\mathbf{S}\cdot\left[\left(-\nabla\mathbf{u}\nabla\mathbf{V}\right)^T\delta\mathbf{F}\right]^s &= \mathbf{S}\cdot\left[\left(-\nabla\mathbf{u}\nabla\mathbf{V}\right)^T\delta\mathbf{F}\right] = \left(-\nabla\mathbf{u}\nabla\mathbf{V}\right)\mathbf{S}\cdot\delta\mathbf{F} = \\ &= -\boldsymbol{\Xi}(\mathsf{x})\mathbf{B}_{GV}\hat{\mathbf{u}}_e\cdot\mathbf{B}_G\delta\hat{\mathsf{x}}_e = -\mathbf{B}_G^T\boldsymbol{\Xi}(\mathsf{x})\mathbf{B}_{GV}\hat{\mathbf{u}}_e\cdot\delta\hat{\mathsf{x}}_e,\end{aligned} \tag{C.43}$$

$$\mathbf{S}\cdot\left[\mathbf{F}^T\left(-\nabla\delta\mathbf{u}\nabla\mathbf{V}\right)\right]^s = -\{\mathbf{S}\}\cdot\mathbf{B}_{NV}\delta\mathbf{u} = -\mathbf{B}_{NV}^T\{\mathbf{S}\}\cdot\delta\mathbf{u}. \tag{C.44}$$

$$\mathbf{S}\cdot\delta\mathbf{E}\,\mathrm{Div}\,\mathbf{V} = \mathbf{S}\cdot\mathbf{B}_N\delta\hat{\mathsf{x}}_e\,\mathrm{Div}\,\mathbf{V}_e = \left(\mathrm{Div}\,\mathbf{V}_e\right)\mathbf{B}_N^T\{\mathbf{S}\}\cdot\delta\hat{\mathsf{x}}_e. \tag{C.45}$$

Sensibilidade à Forma em Elasticidade Linear

$$a_V'(\mathbf{u},\mathbf{v}) = \int_{\mathcal{B}}\mathsf{C}\left[\nabla^s\mathbf{u}\right]\cdot\nabla^s\mathbf{v}\,\mathrm{Div}\,\mathbf{V}\;dV - \int_{\mathcal{B}}\mathsf{C}\left[\left(\nabla\mathbf{u}\nabla\mathbf{V}\right)^s\right]\cdot\nabla^s\mathbf{v} + \mathsf{C}\left[\nabla^s\mathbf{u}\right]\cdot\left(\nabla\mathbf{v}\nabla\mathbf{V}\right)^s\;dV.$$

No domínio de um elemento e:

$$\begin{aligned}\mathsf{C}\left[\nabla^s\mathbf{u}\right]\cdot\nabla^s\mathbf{v}\,\mathrm{Div}\,\mathbf{V} &= [\mathsf{C}]\,\mathbf{B}_L\hat{\mathbf{u}}_e\left(\mathrm{Div}\,\mathbf{V}_e\right)\cdot\mathbf{B}_L\delta\mathbf{v}_e \\ &= \mathbf{B}_L^T[\mathsf{C}]\,\mathbf{B}_L\hat{\mathbf{u}}_e\left(\mathrm{Div}\,\mathbf{V}_e\right)\cdot\delta\mathbf{v}_e.\end{aligned}$$

$$\begin{aligned}\mathsf{C}\left[\left(\nabla\mathbf{u}\nabla\mathbf{V}\right)^s\right]\cdot\nabla^s\mathbf{v} + \mathsf{C}\left[\nabla^s\mathbf{u}\right]\cdot\left(\nabla\mathbf{v}\nabla\mathbf{V}\right)^s &= [\mathsf{C}]\,\mathbf{B}_{LV}\hat{\mathbf{u}}_e\cdot\mathbf{B}_L\delta\mathbf{v}_e + [\mathsf{C}]\,\mathbf{B}_L\hat{\mathbf{u}}_e\cdot\mathbf{B}_{LV}\delta\mathbf{v}_e \\ &= \mathbf{B}_L^T[\mathsf{C}]\,\mathbf{B}_{LV}\hat{\mathbf{u}}_e\cdot\delta\mathbf{v}_e + \mathbf{B}_{LV}^T[\mathsf{C}]\,\mathbf{B}_L\hat{\mathbf{u}}_e\cdot\delta\mathbf{v}_e.\end{aligned}$$

C.1.4 Formas Matriciais

Elementos Bidimensionais: Tensão e Deformação Planas

Sendo

$$VN_{,1}^i = V_{1,1}N_{,1}^i + V_{2,1}N_{,2}^i,$$

$$VN^i_{,2} \;=\; V_{1,2}N^i_{,1} + V_{2,2}N^i_{,2}.$$

as matrizes citadas anteriormente têm as seguintes formas:

$$\mathbf{B}_{GV} = \begin{bmatrix} VN^1_{,1} & 0 & \ldots & VN^{Nn}_{,1} & 0 \\ VN^1_{,2} & 0 & \ldots & VN^{Nn}_{,2} & 0 \\ 0 & VN^1_{,1} & \ldots & 0 & VN^{Nn}_{,1} \\ 0 & VN^1_{,2} & \ldots & 0 & VN^{Nn}_{,2} \end{bmatrix},$$

$$\mathbf{B}_{LV} = \begin{bmatrix} VN^1_{,1} & 0 & \ldots & VN^{Nn}_{,1} & 0 \\ 0 & VN^1_{,2} & \ldots & 0 & VN^{Nn}_{,2} \\ VN^1_{,2} & VN^1_{,1} & \ldots & VN^{Nn}_{,2} & VN^{Nn}_{,1} \end{bmatrix}.$$

$$\mathbf{B}_{NV} = \mathbf{B}_{LV} + \mathbf{B}_{UV},$$

$$\mathbf{B}_{UV} = \begin{bmatrix} u_{1,1}VN^1_{,1} & u_{2,1}VN^1_{,1} & \cdots & u_{1,1}VN^{Nn}_{,1} & u_{2,1}VN^{Nn}_{,1} \\ u_{1,2}VN^1_{,2} & u_{2,2}VN^1_{,2} & \cdots & u_{1,2}VN^{Nn}_{,2} & u_{2,2}VN^{Nn}_{,2} \\ u_{1,1}VN^1_{,2} + u_{1,2}VN^1_{,1} & u_{2,1}VN^1_{,2} + u_{2,2}VN^1_{,1} & \cdots & u_{1,1}VN^{Nn}_{,2} + u_{1,2}VN^{Nn}_{,1} & u_{2,1}VN^{Nn}_{,2} + u_{2,2}VN^{Nn}_{,1} \end{bmatrix}.$$

Elementos Sólidos Tridimensionais

Sendo

$$\begin{aligned} VN^i_{,1} &= V_{1,1}N^i_{,1} + V_{2,1}N^i_{,2} + V_{3,1}N^i_{,3}, \\ VN^i_{,2} &= V_{1,2}N^i_{,1} + V_{2,2}N^i_{,2} + V_{3,2}N^i_{,3}, \\ VN^i_{,3} &= V_{1,3}N^i_{,1} + V_{2,3}N^i_{,2} + V_{3,3}N^i_{,3}. \end{aligned}$$

as matrizes citadas anteriormente têm as seguintes formas:

$$\mathbf{B}_{GV} = \begin{bmatrix} VN^1_{,1} & 0 & 0 & \ldots & VN^{Nn}_{,1} & 0 & 0 \\ VN^1_{,2} & 0 & 0 & \ldots & VN^{Nn}_{,2} & 0 & 0 \\ VN^1_{,3} & 0 & 0 & \ldots & VN^{Nn}_{,3} & 0 & 0 \\ 0 & VN^1_{,1} & 0 & \ldots & 0 & VN^{Nn}_{,1} & 0 \\ 0 & VN^1_{,2} & 0 & \ldots & 0 & VN^{Nn}_{,2} & 0 \\ 0 & VN^1_{,3} & 0 & \ldots & 0 & VN^{Nn}_{,3} & 0 \\ 0 & 0 & VN^1_{,1} & \ldots & 0 & 0 & VN^{Nn}_{,1} \\ 0 & 0 & VN^1_{,2} & \ldots & 0 & 0 & VN^{Nn}_{,2} \\ 0 & 0 & VN^1_{,3} & \ldots & 0 & 0 & VN^{Nn}_{,3} \end{bmatrix},$$

$$\mathbf{B}_{LV} = \begin{bmatrix} VN_{,1}^{1} & 0 & 0 & \dots & VN_{,1}^{Nn} & 0 & 0 \\ 0 & VN_{,2}^{1} & 0 & \dots & 0 & VN_{,2}^{Nn} & 0 \\ 0 & 0 & VN_{,3}^{1} & \dots & 0 & 0 & VN_{,3}^{Nn} \\ VN_{,2}^{1} & VN_{,1}^{1} & 0 & \dots & VN_{,2}^{Nn} & VN_{,1}^{Nn} & 0 \\ VN_{,3}^{1} & 0 & VN_{,1}^{1} & \dots & VN_{,3}^{Nn} & 0 & VN_{,1}^{Nn} \\ 0 & VN_{,3}^{1} & VN_{,2}^{1} & \dots & 0 & VN_{,3}^{Nn} & VN_{,2}^{Nn} \end{bmatrix}.$$

$$\mathbf{B}_{NV} = \mathbf{B}_{LV} + \mathbf{B}_{UV},$$

$$\mathbf{B}_{UV} = \begin{bmatrix} u_{1,1}VN_{,1}^{1} & u_{2,1}VN_{,1}^{1} & u_{3,1}VN_{,1}^{1} & \cdots & u_{3,1}VN_{,1}^{Nn} \\ u_{1,2}VN_{,2}^{1} & u_{2,2}VN_{,2}^{1} & u_{3,2}VN_{,2}^{1} & \cdots & u_{3,2}VN_{,2}^{Nn} \\ u_{1,3}VN_{,3}^{1} & u_{2,3}VN_{,3}^{1} & u_{3,3}VN_{,3}^{1} & \cdots & u_{3,3}VN_{,3}^{Nn} \\ u_{1,1}VN_{,2}^{1}+u_{1,2}VN_{,1}^{1} & u_{2,1}VN_{,2}^{1}+u_{2,2}VN_{,1}^{1} & u_{3,1}VN_{,2}^{1}+u_{3,2}VN_{,1}^{1} & \cdots & u_{3,1}VN_{,2}^{Nn}+u_{3,2}VN_{,1}^{Nn} \\ u_{1,1}VN_{,3}^{1}+u_{1,3}VN_{,1}^{1} & u_{2,1}VN_{,3}^{1}+u_{2,3}VN_{,1}^{1} & u_{3,1}VN_{,3}^{1}+u_{3,3}VN_{,1}^{1} & \cdots & u_{3,1}VN_{,3}^{Nn}+u_{3,3}VN_{,1}^{Nn} \\ u_{1,2}VN_{,3}^{1}+u_{1,3}VN_{,2}^{1} & u_{2,2}VN_{,3}^{1}+u_{2,3}VN_{,2}^{1} & u_{3,2}VN_{,3}^{1}+u_{3,3}VN_{,2}^{1} & \cdots & u_{3,2}VN_{,3}^{Nn}+u_{3,3}VN_{,2}^{Nn} \end{bmatrix}.$$

C.2 Hiperelasticidade Quasi-Incompressível

C.2.1 Segundo Tensor de Piola-Kirchhoff e Tensor Constitutivo Tangente

O segundo tensor de Piola-Kirchhoff e o tensor constitutivo tangente de materiais quasi-incompressíveis expressos em termos da série de Mooney dependem das derivadas de primeira e segunda ordem desses invariantes em relação ao tensor de deformação finita $\mathbf{E}$.

As derivadas de primeira ordem dos invariantes reduzidos $\bar{I}_1, \bar{I}_2, J$ com respeito ao tensor de deformação de Lagrange são

$$\bar{I}_{1,\mathbf{E}} = I_{1,\mathbf{E}}\,(I_3)^{-1/3} - \frac{1}{3} I_1\,(I_3)^{-4/3}\, I_{3,\mathbf{E}}, \tag{C.46}$$

$$\bar{I}_{2,\mathbf{E}} = I_{2,\mathbf{E}}\,(I_3)^{-2/3} - \frac{2}{3} I_2\,(I_3)^{-5/3}\, I_{3,\mathbf{E}}, \tag{C.47}$$

$$J_{,\mathbf{E}} = \frac{1}{2}\,(I_3)^{-1/2}\, I_{3,\mathbf{E}}, \tag{C.48}$$

sendo

$$I_1 = \operatorname{tr}\mathbf{C},$$

$$I_2 = \frac{1}{2}\left[(\operatorname{tr}\mathbf{C})^2 - \operatorname{tr}\left(\mathbf{C}^2\right)\right],$$

$$I_3 = \det \mathbf{C} = (\det \mathbf{F})^2 = J^2,$$

$$I_{1,\mathbf{E}} = 2\mathbf{I}, \tag{C.49}$$

$$I_{2,\mathbf{E}} = 4\mathbf{I} + 4\,(\mathrm{tr}\,\mathbf{E})\,\mathbf{I} - 4\mathbf{E}, \tag{C.50}$$

$$I_{3,\mathbf{E}} = 2\mathbf{I} + 4\,(\mathrm{tr}\,\mathbf{E})\,\mathbf{I} - 4\mathbf{E} + 4e_{imn}e_{jrs}E_{mr}E_{ns}, \tag{C.51}$$

$$e_{imn}e_{jrs}E_{mr}E_{ns} = 2\begin{bmatrix} (E_{22}E_{33} - E_{23}E_{23}) & (E_{23}E_{13} - E_{12}E_{33}) & (E_{12}E_{23} - E_{22}E_{13}) \\ (E_{23}E_{13} - E_{33}E_{12}) & (E_{33}E_{11} - E_{13}E_{13}) & (E_{13}E_{12} - E_{23}E_{11}) \\ (E_{12}E_{23} - E_{13}E_{22}) & (E_{13}E_{12} - E_{11}E_{23}) & (E_{11}E_{22} - E_{12}E_{12}) \end{bmatrix}. \tag{C.52}$$

As derivadas de segunda ordem são mostradas em (Sussman e Bathe, 1987).

Segundo (2.88), a segundo tensor de Piola-Kirchhoff para o material de Mooney-Rivlin quasi-incompressível é definido pela seguinte expressão

$$\mathbf{S} = W_{,\mathbf{E}} = A_{10}\frac{\partial \bar{I}_1}{\partial \mathbf{E}} + A_{01}\frac{\partial \bar{I}_2}{\partial \mathbf{E}} + p\frac{\partial J}{\partial \mathbf{E}}, \tag{C.53}$$

e portanto, usando as expressões (C.46) a (C.48), tem-se

$$\begin{aligned} \mathbf{S} \;=\; & \left(2A_{10}J^{-2/3}\right)\mathbf{I} + \left(4A_{01}J^{-4/3}\right)\left[\mathbf{I} + (\mathrm{tr}\,\mathbf{E})\,\mathbf{I} - \mathbf{E}\right] \\ & + \left[-A_{10}\frac{I_1}{3}J^{-8/3} - A_{01}\frac{2I_2}{3}J^{-10/3} + \frac{p}{2}J^{-1}\right] I_{3,\mathbf{E}}. \end{aligned}$$

E da mesma forma obtem-se o tensor constitutivo tangente na forma tensorial

$$\mathsf{C} = W_{,\mathbf{E},\mathbf{E}} = A_{10}\frac{\partial^2 \bar{I}_1}{\partial \mathbf{E}^2} + A_{01}\frac{\partial^2 \bar{I}_2}{\partial \mathbf{E}^2} + p\frac{\partial^2 J}{\partial \mathbf{E}^2}. \tag{C.54}$$

Da expressão da variação incremental do tensor de Piola-Kirchhoff

$$\Delta\mathbf{S} = \mathsf{C} : \Delta\mathbf{E},$$

e de (A.1) tem-se que

$$\Delta S_{ij} = \mathsf{C}_{ijkl} \cdot \Delta\mathbf{E} = \frac{\partial S_{ij}}{\partial \mathbf{E}} \cdot \Delta\mathbf{E}.$$

Na versão algorítmica

$$[\Delta\mathbf{S}] = [\mathsf{C}]\,[\Delta\mathbf{E}]\,,$$

$[\Delta\mathbf{E}]$ é o vetor (C.22), $[\Delta\mathbf{S}]$ é a variação de (C.25) e $[\mathsf{C}]$ é uma matriz simétrica com a seguinte estrutura para sólidos tridimensionais

$$[\mathsf{C}] = \begin{bmatrix} \frac{\partial S_{11}}{\partial E_{11}} & \frac{\partial S_{11}}{\partial E_{22}} & \frac{\partial S_{11}}{\partial E_{33}} & \frac{1}{2}\frac{\partial S_{11}}{\partial E_{12}} & \frac{1}{2}\frac{\partial S_{11}}{\partial E_{13}} & \frac{1}{2}\frac{\partial S_{11}}{\partial E_{23}} \\ & \frac{\partial S_{22}}{\partial E_{22}} & \frac{\partial S_{22}}{\partial E_{33}} & \frac{1}{2}\frac{\partial S_{22}}{\partial E_{12}} & \frac{1}{2}\frac{\partial S_{22}}{\partial E_{13}} & \frac{1}{2}\frac{\partial S_{22}}{\partial E_{23}} \\ & & \frac{\partial S_{33}}{\partial E_{33}} & \frac{1}{2}\frac{\partial S_{33}}{\partial E_{12}} & \frac{1}{2}\frac{\partial S_{33}}{\partial E_{13}} & \frac{1}{2}\frac{\partial S_{33}}{\partial E_{23}} \\ & & & \frac{1}{2}\frac{\partial S_{12}}{\partial E_{12}} & \frac{1}{2}\frac{\partial S_{12}}{\partial E_{13}} & \frac{1}{2}\frac{\partial S_{12}}{\partial E_{23}} \\ & & & & \frac{1}{2}\frac{\partial S_{13}}{\partial E_{13}} & \frac{1}{2}\frac{\partial S_{13}}{\partial E_{23}} \\ & & & & & \frac{1}{2}\frac{\partial S_{23}}{\partial E_{23}} \end{bmatrix},$$

e

$$[\mathsf{C}] = \begin{bmatrix} \frac{\partial S_{11}}{\partial E_{11}} & \frac{\partial S_{11}}{\partial E_{22}} & \frac{1}{2}\frac{\partial S_{11}}{\partial E_{12}} & \frac{\partial S_{11}}{\partial E_{33}} \\ & \frac{\partial S_{22}}{\partial E_{22}} & \frac{1}{2}\frac{\partial S_{22}}{\partial E_{12}} & \frac{\partial S_{22}}{\partial E_{33}} \\ & & \frac{1}{2}\frac{\partial S_{12}}{\partial E_{12}} & \frac{1}{2}\frac{\partial S_{12}}{\partial E_{33}} \\ & & & \frac{\partial S_{33}}{\partial E_{33}} \end{bmatrix}$$

para problemas de tensão plana e sólidos axissimétricos.

C.2.2 Dicretização da Análise de Resposta

A discretização das grandezas relacionadas ao campo de deslocamentos é idêntica à desenvolvida na seção C.1, restando apenas os termos associados à manipulação da restrição de quasi-incompressibilidade, da qual a incompressibilidade é um caso particular.

Considera-se aqui a condensação da pressão em cada elemento através do método de projeção da pressão com interpolação constante. Nesse método, a matriz de rigidez tangente e uma parcela força interna são obtidas da expressão

$$\begin{aligned} \delta a\left(\mathsf{s};\Delta\mathsf{x},\delta\mathsf{x}\right)+\delta b\left(\mathsf{s};\delta\mathsf{x},\Delta p\right) &= \int_{\mathcal{B}}\mathsf{C}\left(\mathsf{x}\right):D\mathbf{E}\left[\Delta\mathbf{u}\right]\cdot\delta\mathbf{E}\;dV+\int_{\mathcal{B}}\mathbf{S}\cdot D\delta\mathbf{E}\left[\Delta\mathbf{u}\right]\;dV \\ &+\int_{\mathcal{B}}J_{,\mathbf{E}}\;\Delta p\cdot\delta\mathbf{E}\;dV, \end{aligned} \tag{C.55}$$

na qual as duas primeiras integrais já foram desenvolvidas e a variação da pressão hidrostática Δp é aproximada por

$$\Delta\hat{p}_e=\frac{\tilde{\kappa}}{\int_{\mathcal{B}_e}dV}\int_{\mathcal{B}_e}\left\{J_{,\mathbf{E}}\cdot D\mathbf{E}\left[\Delta\mathbf{u}\right]+J-1-\frac{\hat{p}_e}{\tilde{\kappa}}\right\}\;dV, \tag{C.56}$$

sendo $\hat{p}_e$ dado pelo valor constante

$$\hat{p}_e = \frac{\tilde{\kappa}}{\int_{\mathcal{B}_e} dV} \int_{\mathcal{B}_e} [J - 1] \; dV \tag{C.57}$$

num elemento e.

Observando que $J_{,\mathbf{E}}$ é um tensor simétrico, discretiza-se o primeiro termo do integrando de (C.56):

$$J_{,\mathbf{E}} \cdot D\mathbf{E}\,[\Delta\mathbf{u}] = [J_{,\mathbf{E}}] \cdot \mathbf{B}_N \Delta\hat{\mathbf{u}}_e = \mathbf{B}_N^T [J_{,\mathbf{E}}] \cdot \Delta\hat{\mathbf{u}}_e = \mathbf{G} \cdot \Delta\hat{\mathbf{u}}_e = \mathbf{G}^T \Delta\hat{\mathbf{u}}_e, \qquad \mathbf{G} = \mathbf{B}_N^T [J_{,\mathbf{E}}], \tag{C.58}$$

sendo

$$[J_{,\mathbf{E}}]^T = \left[\begin{array}{cccccc} (J_{,\mathbf{E}})_{11} & (J_{,\mathbf{E}})_{22} & (J_{,\mathbf{E}})_{33} & (J_{,\mathbf{E}})_{12} & (J_{,\mathbf{E}})_{13} & (J_{,\mathbf{E}})_{23} \end{array}\right] \tag{C.59}$$

para sólidos tridimensionais e

$$[J_{,\mathbf{E}}]^T = \left[\begin{array}{cccc} (J_{,\mathbf{E}})_{11} & (J_{,\mathbf{E}})_{22} & (J_{,\mathbf{E}})_{12} & (J_{,\mathbf{E}})_{33} \end{array}\right]$$

para modelos bidimensionais (deformação plana e sólido axissimétrico).

A partir de (C.56) e (C.58), tem-se a forma algorítmica do integrando do último termo de (C.55):

$$J_{,\mathbf{E}}\; \Delta p \cdot \delta\mathbf{E} \simeq J_{,\mathbf{E}} \Delta\hat{p}_e \cdot \delta\mathbf{E} = J_{,\mathbf{E}} \Delta\hat{p}_e \cdot D\mathbf{E}\,[\delta\mathsf{x}] =$$

$$\begin{aligned} &= J_{,\mathbf{E}} \frac{\tilde{\kappa}}{\int_{\mathcal{B}_e} dV} \int_{\mathcal{B}_e} \left[\mathbf{G}^T \Delta\hat{\mathbf{u}}_e + J - 1 - \frac{\hat{p}_e}{\tilde{\kappa}}\right] \; dV \cdot D\mathbf{E}\,[\delta\mathsf{x}] = \\ &= \mathbf{G} \frac{\tilde{\kappa}}{\int_{\mathcal{B}_e} dV} \int_{\mathcal{B}_e} \left[\mathbf{G}^T \Delta\hat{\mathbf{u}}_e + J - 1 - \frac{\hat{p}_e}{\tilde{\kappa}}\right] \; dV \cdot \delta\hat{\mathsf{x}}_e = \\ &= \left[\mathbf{G} \frac{\tilde{\kappa}}{\int_{\mathcal{B}_e} dV} \left(\int_{\mathcal{B}_e} \mathbf{G}^T \; dV\right) \Delta\hat{\mathbf{u}}_e + \mathbf{G} \frac{\tilde{\kappa}}{\int_{\mathcal{B}_e} dV} \left(\int_{\mathcal{B}_e} J - 1 - \frac{\hat{p}_e}{\tilde{\kappa}} \; dV\right)\right] \cdot \delta\hat{\mathsf{x}}_e. \end{aligned} \tag{C.60}$$

Uma vez construído o sistema para um iteração de Newton-Raphson, a primeira parcela de (C.60), relacionada à incógnita $\Delta\hat{\mathbf{u}}_e$, ajuda a compor a matriz de rigidez tangente do elemento e, enquanto que a segunda parcela contribui para a força interna

$$\begin{aligned} a(\mathsf{x}, \delta\mathsf{x}) + b_1(\delta\mathsf{x}, p) &= \int_{\mathcal{B}} \left(\bar{\mathbf{S}} + \tilde{\mathbf{S}}\right) \cdot \delta\mathbf{E} \; dV = \int_{\mathcal{B}} \mathbf{S} \cdot \delta\mathbf{E} \; dV = \\ &= \mathsf{A}_{e=1}^{Ne} \left\{ \int_{\mathcal{B}_e} \left[\mathbf{B}_N^T \{\mathbf{S}\}\right] \; dV \right\} \cdot \delta\hat{\mathsf{x}}. \end{aligned} \tag{C.61}$$

Portanto, as formas algorítmicas finais de matriz de rigidez tangente e força interna são

$$\begin{aligned} {}^{(k)}\left[\mathbf{K}\right]_{n+1} &= \mathsf{A}_{e=1}^{Ne}\left\{\int_{\mathcal{B}_e} {}^{(k)}\left[\mathbf{B}_N^T\left[\mathsf{C}\left(\mathsf{s}\right)\right]\mathbf{B}_N+\mathbf{B}_G^T\mathbf{\Xi}\left(\mathsf{s}\right)\mathbf{B}_G\right]_{n+1} dV\right\} \\ &+\mathsf{A}_{e=1}^{Ne}\left\{\frac{\tilde{\kappa}}{\int_{\mathcal{B}_e} dV}\int_{\mathcal{B}_e} {}^{(k)}\left[\mathbf{G}\right]_{n+1}\ dV\int_{\mathcal{B}_e} {}^{(k)}\left[\mathbf{G}\right]_{n+1}^T\ dV\right\}, \end{aligned} \tag{C.62}$$

e

$$\begin{aligned} {}^{(k)}\left[\hat{\mathbf{f}}_{int}\right]_{n+1} &= \mathsf{A}_{e=1}^{Ne}\left\{\int_{\mathcal{B}_e} {}^{(k)}\left[\mathbf{B}_N^T\left\{\mathbf{S}\right\}\right]_{n+1}\ dV\right\} \\ &+\mathsf{A}_{e=1}^{Ne}\left\{\frac{\tilde{\kappa}}{\int_{\mathcal{B}_e} dV}\int_{\mathcal{B}_e} {}^{(k)}\left[\mathbf{G}\right]_{n+1}\ dV\int_{\mathcal{B}_e} {}^{(k)}\left[J-1-\frac{\hat{p}_e}{\tilde{\kappa}}\right]_{n+1}\ dV\right\}. \end{aligned} \tag{C.63}$$

C.2.3 Discretização das Expressões de Análise de Sensibilidade

A primeira equação do sistema (3.68) fornece

$$\begin{aligned} \delta a\left(\mathsf{s};\dot{\mathbf{u}},\delta\mathsf{x}\right)+\delta b\left(\mathsf{s};\delta\mathsf{x},\dot{p}\right) &= \int_{\mathcal{B}}\mathsf{C}\left(\mathsf{x}\right):D\mathbf{E}\left[\dot{\mathbf{u}}\right]\cdot\delta\mathbf{E}\ dV+\int_{\mathcal{B}}\mathbf{S}\cdot D\delta\mathbf{E}\left[\dot{\mathbf{u}}\right]\ dV \\ &+\int_{\mathcal{B}} J_{,\mathbf{E}}\ \dot{p}\cdot\delta\mathbf{E}\ dV, \end{aligned} \tag{C.64}$$

à qual será aplicada o método de projeção através da expressão[3]

$$\dot{\hat{p}}_e=\frac{\tilde{\kappa}}{\int_{\mathcal{B}_e} dV}\int_{\mathcal{B}_e}\left[J_{,\mathbf{E}}\cdot D\mathbf{E}\left[\dot{\mathbf{u}}\right]+J_{,\mathbf{E}}\cdot\mathbf{E}_{\mathbf{V}}\left(\mathsf{x}\right)+\left(J-1-G_{n+1}^{*\prime}\right)\operatorname{Div}\mathbf{V}\right]\ dV. \tag{C.65}$$

O segundo termo do integrando acima é discretizado da mesma forma que (C.58), ou seja,

$$J_{,\mathbf{E}}\cdot\mathbf{E}_{\mathbf{V}}\left(\mathsf{x}\right)=-\left[J_{,\mathbf{E}}\right]\cdot\mathbf{B}_{NV}\hat{\mathbf{u}}_e=-\mathbf{B}_{NV}^T\left[J_{,\mathbf{E}}\right]\cdot\hat{\mathbf{u}}_e=-\mathbf{G}_V\cdot\hat{\mathbf{u}}_e=-\mathbf{G}_V^T\hat{\mathbf{u}}_e,\quad \mathbf{G}_V=\mathbf{B}_{NV}^T\left[J_{,\mathbf{E}}\right]. \tag{C.66}$$

Portanto,

$$J_{,\mathbf{E}}\ \dot{p}\cdot\delta\mathbf{E}\simeq J_{,\mathbf{E}}\ \dot{\hat{p}}_e\cdot\delta\mathbf{E}=J_{,\mathbf{E}}\ \dot{\hat{p}}_e\cdot D\mathbf{E}\left[\delta\mathsf{x}\right]=$$

$$=\ J_{,\mathbf{E}}\frac{\tilde{\kappa}}{\int_{\mathcal{B}_e} dV}\int_{\mathcal{B}_e}\left[\mathbf{G}^T\dot{\hat{\mathbf{u}}}_e-\mathbf{G}_V^T\hat{\mathbf{u}}_e+\left(J-1-\frac{\hat{p}_e}{\tilde{\kappa}}\right)\operatorname{Div}\mathbf{V}_e\right]\ dV\cdot D\mathbf{E}\left[\delta\mathsf{x}\right]=$$

[3]Repetição da expressão (3.69).

$$
\begin{aligned}
&= \mathbf{G}\frac{\tilde{\kappa}}{\int_{\mathcal{B}_e} dV}\int_{\mathcal{B}_e}\left[\mathbf{G}^T\dot{\hat{\mathbf{u}}}_e - \mathbf{G}_V^T\hat{\mathbf{u}}_e + \left(J - 1 - \frac{\hat{p}_e}{\tilde{\kappa}}\right)\operatorname{Div}\mathbf{V}_e\right]\, dV \cdot \delta\hat{\mathrm{x}}_e = \\
&= \mathbf{G}\frac{\tilde{\kappa}}{\int_{\mathcal{B}_e} dV}\left[\int_{\mathcal{B}_e}\mathbf{G}^T\, dV\right]\dot{\hat{\mathbf{u}}}_e \cdot \delta\hat{\mathrm{x}}_e + \qquad \text{(C.67)}\\
&\quad + \mathbf{G}\frac{\tilde{\kappa}}{\int_{\mathcal{B}_e} dV}\left\{\int_{\mathcal{B}_e}\left[-\mathbf{G}_V^T\hat{\mathbf{u}}_e + \left(J - 1 - \frac{\hat{p}_e}{\tilde{\kappa}}\right)\operatorname{Div}\mathbf{V}_e\right]\, dV\right\}\cdot\delta\hat{\mathrm{x}}_e. \qquad \text{(C.68)}
\end{aligned}
$$

O termo (C.67), quando somado aos dois primeiros termos de (C.64) compõe a matriz de rigidez tangente da última configuração do sistema de acordo com (C.62), enquanto que (C.68) se soma à derivada da equação diferencial de deslocamentos em relação ao campo de velocidades.