

Renato Machniewsz

**OTIMIZAÇÃO DE FORMA EMPREGANDO O MÉTODO
DAS DIREÇÕES VIÁVEIS DE ZOUTENDIJK**

Curitiba – PR

Outubro / 2003

Renato Machniewscz

**OTIMIZAÇÃO DE FORMA EMPREGANDO O MÉTODO
DAS DIREÇÕES VIÁVEIS DE ZOUTENDIJK**

Dissertação apresentada como parte dos requisitos à obtenção do título de Mestre em Engenharia Mecânica, pelo Programa de Pós-Graduação em Engenharia Mecânica, pela Pontifícia Universidade Católica do Paraná.

Orientador: Prof. Dr. Renato Barbieri

Co-Orientador: Prof. Dr. Nilson Barbieri

Curitiba – PR

Outubro / 2003

Dissertação de Mestrado sob o título “*Otimização de Forma Empregando o Método das Direções Viáveis de Zoutendijk*”, defendida por Renato Machnievscz e aprovada em 30 de outubro de 2003, em Curitiba, Paraná, pela banca examinadora constituída pelos doutores:

Prof. Dr. Renato Barbieri
PPGEM - PUCPR
Orientador

Prof. Dr. Nilson Barbieri
PPGEM - PUCPR

Prof. Dr. Jucélio Tomás Pereira
Departamento Acadêmico de Mecânica - CEFET-PR

Prof. Dr. João Elias Abdalla Filho
PPGEM - PUCPR-PR

*Dedico esta dissertação a
Eliani e Rafaela.*

Agradecimentos

Agradeço especialmente ao meu orientador e amigo professor Renato Barbieri, por sua confiança em minha capacidade, por sua motivação, por seus ensinamentos e pela agradável convivência durante o tempo deste mestrado.

Ao professor Nilson Barbieri, co-orientador.

Aos demais professores, amigos (em especial Key Lima e José Borgert) e aos colegas pelo apoio, pelas brincadeiras e, sobretudo pela paciência.

Não poderia deixar também de lembrar da secretária do mestrado, Sra. Jane, pela ajuda descompromissada e pelos cafezinhos de todas as horas.

E agradeço ao Sr. Antonio Rodrigues (in memoriam), sócio fundador da EMIC Equipamentos e Sistemas de Ensaios Ltda., pelo incentivo, pela amizade e pelo apoio financeiro.

Resumo

O processo de otimização tornou-se indispensável. Os cálculos, muitas vezes tediosos, de escolhas de parâmetros, formas e tamanhos, tornou-se muito mais eficaz e rápido fazendo uso das técnicas de Programação Matemática, PM. Procurou-se, neste trabalho, compreender e aplicar um dos métodos da PM: o método das direções viáveis de Zoutendijk. As principais características deste método são a robustez e a eficiência, e aplica-se, principalmente, para o caso de restrições não-lineares. Foi utilizada a técnica das variáveis naturais, desenvolvida por Belegundu e Rajan, na qual as variáveis de projeto, são as magnitudes das cargas virtuais aplicadas ao corpo. Os deslocamentos causados por estas cargas virtuais, chamados de campos de velocidade, são adicionados à malha inicial para a obtenção de uma nova forma. Através do método dos elementos finitos é estabelecida uma relação entre os pontos nodais e as variáveis de projeto. Foram usados campos de velocidades criados a partir de cargas nodais unitárias e campos de pressão, para problemas de elasticidade plana bi-dimensional. Mostra-se também que é possível, adicionalmente, aplicar restrições geométricas de deslocamento, com a finalidade de impedir que o contorno a ser modificado ultrapasse determinada dimensão. Apesar de não ter sido implementada a aproximação do contorno por meio de *splines*, e também de não se fazer uma nova malha (*remesh*) a cada iteração, os resultados são compatíveis com os encontrados na literatura, embora a distorção dos elementos foi controlada pela definição de valores específicos para a perturbação, nos cálculos dos gradientes.

Abstract

Nowadays, the optimization process is indispensable. With the use of Mathematical Programming (MP) techniques, the choice, almost always tedious, of parameters, shapes and sizes has become faster and more efficient. The aim of this work is to understand and apply an MP method to optimize a bidimensional elasticity problem in engineering. In order to achieve that the Zoutendijk method of feasible directions was chosen. The main characteristics of the Zoutendijk method is robustness and efficiency and it is geared mainly to solving problems with inequality constraints. It was used a natural design variables technique, developed by Belegundu and Rajan, to solve this elasticity problem. In this technique the design variables are the magnitudes of a set of virtual loads applied on the body. The displacements due to these virtual loads, which is named velocity field, are added to the initial grid to obtain a new shape. A relation between the nodal points and the design variables is established through a finite element analysis. Two methods were used to create the velocity field: the virtual load technique and the virtual pressure field technique. The element distortion was controlled by defining specific values to the perturbation function in the gradient evaluation. This technique was used since no remeshing or boundary spline approximation is implemented. Additionally, it is possible to apply geometrical constraints to limit the problem boundaries. In spite of using unusual techniques the results found are close to those reported in the literature which confirms that this approach is efficient when there are limited computational resources such as an integrated mesh generator.

Sumário

1	Introdução	1
1.1	Histórico da Otimização	2
1.2	Motivação e Objetivos.....	4
2	Conceitos Matemáticos Básicos	5
2.1	Otimização com restrições	6
2.2	Condições necessárias para a otimalidade	8
2.2.1	Problema com restrições de igualdade	8
2.2.1.1	Condição necessária de primeira ordem	10
2.2.1.2	Condição necessária de segunda ordem	10
2.2.1.3	Condição suficiente de segunda ordem	11
2.2.2	Problemas com restrições de desigualdade	11
	Direção Descendente:	11
	Direção Viável:	12
	Condições necessárias para a otimalidade (KKT).....	14
2.2.3	Problema com restrições de igualdade e desigualdade	15

2.2.3.1	Condição necessária para a otimalidade (KKT)	15
2.3	Condições suficientes para a otimalidade	15
3	Método das Direções Viáveis	17
3.1	Método de Zoutendijk	18
3.1.1	Restrições não-lineares	19
3.1.2	Problema da busca de direção	21
3.1.3	Determinação do tamanho do intervalo (<i>step size</i>)	22
3.1.4	Parâmetro ϵ de ativação das restrições	22
3.1.5	Algoritmo básico do método de Zoutendijk	23
	Algoritmo	23
4	Otimização de Forma	25
4.1	Método das variáveis naturais	26
4.1.1	Campos de velocidade	26
4.1.2	Escolha das variáveis de projeto	30
4.1.3	Algoritmo básico	31
4.2	Análise de sensibilidade	32
4.2.1	Diferenças finitas	32
4.2.2	Método direto(<i>pseudo-load vector</i>)	33
5	Implementação Numérica	36
5.1	Algoritmo do aplicativo Shape	37
5.1.1	Representação geométrica	37
5.1.2	Análise estrutural	38
5.1.3	Algoritmo de otimização	39
5.1.4	Critérios de parada	40

5.2	Controle de distorção	40
5.2.1	Parâmetro de distorção DP	41
5.3	Implementação do controle de distorção	41
6	Resultados	44
6.1	Pórtico	46
6.2	Alavanca	50
6.3	Placa	53
6.4	Concentração de tensões	62
7	Considerações Finais	66
7.1	Sugestões para futuros trabalhos	67
	Referências	68

Lista de Figuras

Figura 1	Exemplo de restrições ativas e inativas.	7
Figura 2	Plano tangente.	8
Figura 3	A direção \mathbf{d} e o cone descendente.	12
Figura 4	O cone viável e a direção viável.	13
Figura 5	O cone das direções decrescente-viáveis.	19
Figura 6	A direção decrescente-viável e o efeito do fator θ	20
Figura 7	O efeito do parâmetro ϵ na determinação da direção \mathbf{d}	23
Figura 8	Carregamentos nodais aplicados aos nós de controle.	27
Figura 9	Placa submetida a um carregamento bi-axial (estrutura principal).	28
Figura 10	Estrutura auxiliar e as cargas unitárias.	29
Figura 11	Deformações causadas pelas cargas unitárias na estrutura auxiliar.	30
Figura 12	Geração do campo de velocidade interno e externo.	31
Figura 13	Tipos de carregamentos virtuais.	31
Figura 14	Fluxograma geral de otimização de forma.	37
Figura 15	Parâmetro DP e as formas dos elementos.	42
Figura 16	Dimensões do pórtico (dimensões em mm).	46
Figura 17	Geometria e condições de contorno.	47

Figura 18	Campos de velocidades.	48
Figura 19	Forma final para o pórtico.	49
Figura 20	Redução de volume após 21 iterações.	49
Figura 21	Dimensões e condições de contorno (dimensões em mm).	50
Figura 22	Partes que não sofrerão modificação em sua forma.	50
Figura 23	Malha inicial com condições de contorno.	51
Figura 24	Forma final da alavanca comparada com a forma inicial.	51
Figura 25	Redução no volume após 39 iterações (com <i>remesh</i>).	52
Figura 26	Tensões equivalentes de von Mises.	52
Figura 27	Dimensões e condições de contorno para chapa com furo (dimensões em mm).	53
Figura 28	Condições iniciais de tensões.	54
Figura 29	Aspecto da malha após a primeira convergência, e ampliação da região com alto grau de distorção.	55
Figura 30	Geometria final, com distribuição de tensões.	56
Figura 31	Redução no volume para o exemplo da placa com furo.	57
Figura 32	Geometria final obtida para o limite de tensão $\sigma_{\text{lim}} = 7.0$ MPa.	58
Figura 33	Resultado obtido ao se aplicar somente cargas unitárias.	59
Figura 34	Exemplo do mau uso de campos de velocidades.	59
Figura 35	Campos de velocidades gerados usando campos de pressão.	61
Figura 36	Dimensões, condições de contorno e restrições geométricas (dimensões em mm).	62
Figura 37	Tensões equivalentes de von Mises.	63
Figura 38	Malha e distribuição de tensões após a primeira convergência.	63
Figura 39	Distribuição de tensões no modelo conforme Grodzinski.	64
Figura 40	Distribuição de tensões após otimização.	64
Figura 41	Histórico de iterações para o modelo de concentração de tensões.	65

Figura 42	Forma final ao se retirar as restrições geométricas.	65
-----------	---	----

Lista de Tabelas

Tabela 1	Métodos de Pesquisa Operacional.	2
Tabela 2	Aplicação manual dos deslocamentos nodais.	45
Tabela 3	Campos de velocidades para o pórtico.	48
Tabela 4	Comparação entre os algoritmos.	57

Lista de Símbolos

f	Função objetivo
\mathfrak{R}^n	Conjunto dos números reais
Ω	Conjunto dos pontos viáveis
\mathbf{x}	Vetor das variáveis do problema de otimização
$\mathbf{g}_i(\mathbf{x}) \leq \mathbf{0}$	Restrições de desigualdade
$\mathbf{h}_j(\mathbf{x}) = \mathbf{0}$	Restrições de igualdade
\mathbf{x}_0	Vetor que contém os pontos iniciais
\mathbf{x}^*	Vetor que contém os pontos de ótimo
\mathbf{x}_k	Ponto atual, na k -ésima iteração
\mathbf{d}	Vetor direção
α	Incremento da busca (<i>step size</i>)
S	Superfície
M	Plano tangente
λ	Multiplicador de Lagrange, associado às restrições de igualdade
L	Lagrangeano
∇f	Gradiente da função objetivo
μ	Multiplicador de Lagrange, associado às restrições de desigualdade

ϵ	Parâmetro para ativação das restrições
σ_{lim}	Valor limite das restrições
E	Módulo de Elasticidade
ν	Coefficiente de Poisson

CAPÍTULO 1

Introdução

“Otimizar é o ato de se obter o melhor resultado sob dadas circunstâncias”^[1]. Diariamente o engenheiro de projetos toma várias decisões baseadas exclusivamente em sua experiência, porém com a crescente exigência de redução de custos, diminuição do consumo de energia, redução do ruído causado por máquinas e motores, etc, sua experiência pessoal pode não ser suficiente para alcançar estes objetivos.

Felizmente pode-se contar com uma ferramenta computacional, que torna a decisão da escolha de uma solução muito mais fácil, cabendo ao engenheiro reinterpretar a solução numérica para torná-la executável.

O nome desta excepcional ferramenta chama-se *otimização*.

É importante notar que a otimização pode se referir tanto à maximização quanto à minimização, e que neste texto a palavra otimização estará sempre se referindo a um problema de minimização, salvo quando indicado.

Não existe nenhum método geral que resolva todos os problemas de otimização de maneira eficiente, portanto, uma grande quantidade de métodos foi desenvolvida para os diferentes tipos de otimização. O ramo da matemática que estuda a otimização é chamado Pesquisa Operacional (P.O.), e uma das suas ramificações são as técnicas de Programação Matemática. A tabela 1 lista os vários métodos matemáticos juntamente com outras

áreas da pesquisa operacional. Esta tabela não pretende ser definitiva e, serve apenas de informação. Uma análise mais detalhada dos diversos métodos, teorias e algoritmos, pode ser encontrado em diversos textos especializados.

Tabela 1: Métodos de Pesquisa Operacional.

Técnicas de Programação Matemática	Técnicas de Processos Estocásticos	Métodos Estatísticos
Cálculo Diferencial	Teoria da Decisão Estatística	Análise de Regressão
Cálculo Variacional	Processos de Markov	Cluster Analysis
Programação Não-linear	Queueing Theory	Reconhecimento de Padrão
Programação Geométrica	Renewal Theory	Análise Discriminante
Programação Quadrática	Métodos de Simulação	
Programação Linear	Análise de Confiabilidade	
Programação Dinâmica		
Programação Inteira		
Programação Estocástica		
Programação Separável		
Programação Multi-Objetivo		
Métodos de Rede: Pert-CPM		
Teoria dos Jogos		
Simulated Annealing		
Algoritmos Genéticos		
Redes Neurais		

Fonte: RAO, S. S. *Engineering optimization*. 3 ed. New York: Wiley, 1996.

1.1 Histórico da Otimização

O primeiro trabalho de que se tem notícia foi atribuído a Galileu (1638), que procurava a solução ótima de uma viga, variando sua espessura. Na seqüência vieram os trabalhos de J. Bernoulli (1687), Newton (1687), D. Bernoulli (1733), Lagrange (1770), também sobre otimização estrutural. Cauchy (1847)^[1, 8], desenvolveu o primeiro método que faz uso do gradiente (necessidade do cálculo das derivadas parciais da função). Mas apesar da precocidade da descoberta do método, foi apenas na metade do século passado que a otimização moderna teve sua maior evolução. Courant (1943) com seu trabalho sobre as funções penalidade, Dantzig (1951) com o método simplex para a Programação Linear (P.L.) e Karush, Kuhn e Tucker (1939, 1951) que desenvolveram as condições que levam os seus nomes (KKT), abriram o caminho para a Programação Matemática.

Schmidt (1960) foi o primeiro que acoplou a análise numérica, pelo método dos elementos finitos (MEF), à programação matemática.

Ainda nos anos 60, os trabalhos de Rosen, com o método da projeção do gradiente, e Zoutendijk, com o método das direções viáveis, deram importantes contribuições para os problemas com restrições. O método dos gradientes conjugados, para problemas não restritos, foi desenvolvido por Fletcher e Reeves (1964), e Davidon-Fletcher-Reeves (1959, 1963) criaram os métodos de variável métrica (DFP). Em 1966, Abadie, Carpentier e Hengsen desenvolveram o método do gradiente reduzido generalizado (GRG). Os métodos de busca, usando os números de Fibonacci ou a proporção áurea foram seguidos pelo eficiente método de Brent (1971), que utiliza o método do ajuste polinomial.

Os métodos da programação seqüencial quadrática (SQP) foram desenvolvidos nos anos 70 por Pshenichny na Rússia. Ainda nos anos 60, paralelamente aos métodos baseados nos gradientes, surgiram os métodos “diretos” ou não-gradientes, como o método de Rosenbrock das direções ortogonais (1960), o método de Hooke e Jeeves para reconhecimento de padrões (1961), o método das direções conjugadas de Powell (1964) e mais recentemente os algoritmos genéticos (Holland, 1975 e Goldberg, 1989).

Novos métodos, que exploram peculiaridades dos problemas, foram também desenvolvidos. A programação dinâmica (Bellman, 1952) e a programação geométrica (Duffin, Peterson, Zener, 1967) são alguns exemplos.

A otimização estrutural pode ser classificada em três grandes áreas:

1. Otimização de dimensões ou parâmetros (*sizing optimization*). Também conhecida como otimização da seção transversal (*cross sectional optimization*), trabalha com os parâmetros geométricos, tais como área, raios, espessuras, comprimentos, dos componentes estruturais típicos, como as vigas, cascas e placas. Nesta otimização a forma do elemento estrutural é invariante.
2. Otimização topológica (*topology optimization*). A otimização topológica determina a distribuição ótima de material para um dado domínio, sujeito às condições de contorno do problema. Neste processo, material pode ser retirado, adicionado ou movido e, por este motivo, a forma final é diferente da forma inicial.
3. Otimização de forma (*shape optimization*). A definição da forma ótima de um domínio, bi ou tridimensional, é conseguida modificando seus contornos (interno ou externo). Os contornos são aproximados por curvas splines, polinômios ou funções

naturais e a análise da função objetivo e das funções de restrição é feita por meio do método dos elementos finitos (MEF) ou pelo método dos elementos de contorno (MEC).

1.2 Motivação e Objetivos

A motivação desta dissertação foi compreender todo o processo da otimização, desde o modelamento do objeto de estudo, até a interpretação dos resultados, buscando a melhor solução para o problema de otimização de forma. Uma vez que o processo de otimização é bastante complexo, apresentando vários métodos distintos entre si, e podendo ter tantas soluções distintas quantas forem as variáveis de projeto escolhidas, torna-se importante dominar todos os passos.

Assim, a partir da escolha de um método pretende-se analisar problemas de otimização de forma. Para isto utilizar-se-á o método dos elementos finitos como ferramenta de análise de tensões e deformações.

Deste modo, pretende-se propor novas soluções de projetos, baseados nos critérios da análise pelo método dos elementos finitos e pela otimização de forma.

Conceitos Matemáticos Básicos

Considere o seguinte problema de otimização não-linear:

$$\begin{aligned} & \text{minimize} && f(\mathbf{x}) \\ & \text{sujeita a} && \mathbf{g}_i(\mathbf{x}) \leq 0, i = 1, 2, \dots, m \\ & && \mathbf{h}_j(\mathbf{x}) = 0, j = 1, 2, \dots, l \\ & && \mathbf{x} \in \Omega \end{aligned} \tag{2.1}$$

onde $f, g_1, \dots, g_m, h_1, \dots, h_l$ são funções definidas em \mathfrak{R}^n , Ω é um subconjunto de \mathfrak{R}^n , também chamado de conjunto das restrições^[2], e \mathbf{x} é um vetor com n componentes x_1, \dots, x_n que satisfazem as restrições e minimizam a função f .

A função f é chamada *função objetivo* ou função custo. As restrições $\mathbf{g}_i(\mathbf{x}) \leq \mathbf{0}$ são chamadas de *restrições de desigualdade* e as restrições $\mathbf{h}_j(\mathbf{x}) = \mathbf{0}$ são chamadas de *restrições de igualdade*, ou ainda de *restrições funcionais*^[2]. Os problemas que se apresentam nesta forma são denominados de problemas de *otimização restrita*. Um ponto $\mathbf{x} \in \Omega$ que satisfaça todas as restrições é chamado de *solução viável* do problema. O conjunto de todas as soluções forma a *região viável*.

Assim, a programação não-linear (PNL) procura encontrar um ponto \mathbf{x}^* tal que $f(\mathbf{x}) \geq f(\mathbf{x}^*)$ para todo $\mathbf{x} \in \Omega$. Este ponto é chamado de *solução ótima*, ou simplesmente

de *solução*.

Neste momento é oportuno introduzir a notação que será utilizada ao longo dos capítulos seguintes:

- $\mathbf{x}_0 = \{x_0^1, x_0^2, \dots, x_0^n\}$ é o vetor que contém os pontos iniciais ou as variáveis iniciais do problema;
- \mathbf{x}^* é o vetor que contém os pontos de ótimo;
- \mathbf{x}_k é o ponto atual, na k -ésima iteração;
- \mathbf{d} é o vetor direção;
- α é o incremento da busca (*step size*).

2.1 Otimização com restrições

Normalmente os problemas de otimização na engenharia apresentam-se restringidos por alguns parâmetros importantes, daí o nome de otimização restrita. O exemplo do projeto de uma estrutura com peso mínimo e sujeita a um valor de tensão admissível, deformação, frequência natural ou restrições de fabricação^[9], ilustra bem um dos problemas mais usuais encontrados e para os quais se faz uso das técnicas de otimização.

A estrutura mais comum do problema de otimização sujeita a restrições é:

$$\begin{aligned}
 & \min \quad f(\mathbf{x}) \\
 & \text{s. a.} \quad \mathbf{g}_i(\mathbf{x}) \leq 0, i = 1, 2, \dots, m \\
 & \quad \quad \mathbf{h}_j(\mathbf{x}) = 0, j = 1, 2, \dots, l \\
 & \quad \quad \mathbf{x} \in \Omega
 \end{aligned} \tag{2.2}$$

onde \mathbf{x} é o vetor coluna das variáveis de projeto, f é a função objetivo, g são as restrições de desigualdade e h representa as restrições de igualdade.

Uma abordagem fundamental é a das *restrições ativas*. Uma restrição de desigualdade é chamada ativa, no ponto viável \mathbf{x} , se $\mathbf{g}_i(\mathbf{x}) = \mathbf{0}$ e inativa em \mathbf{x} se $\mathbf{g}_i(\mathbf{x}) < \mathbf{0}$. Por convenção^[2], as restrições de igualdade, $\mathbf{h}_j(\mathbf{x}) = \mathbf{0}$, são ativas para qualquer ponto viável. É importante salientar que as restrições ativas, em um ponto viável \mathbf{x} , restringem o domínio da viabilidade na vizinhança de \mathbf{x} , enquanto que as restrições inativas não têm

nenhuma influência. No estudo das propriedades de um mínimo local, fica evidente que deve ser dada atenção apenas às restrições ativas. Se o conjunto das restrições ativas for conhecido, pode-se encontrar a solução do problema ao ignorar as restrições que estão inativas e tratar todas as restrições ativas como restrições de igualdade. A figura 1 mostra que as propriedades locais, satisfeitas pela solução \mathbf{x}^* não dependem das restrições g_2 e g_3 .

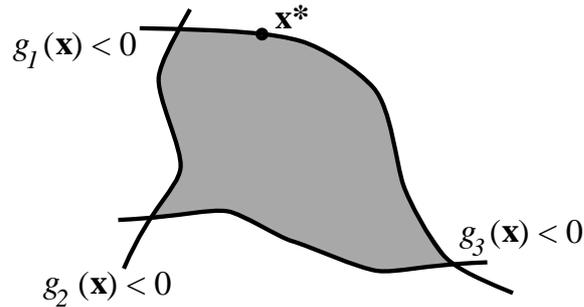


Figura 1: Exemplo de restrições ativas e inativas.

Denominando de Ω a região compreendida por todos os pontos viáveis, onde as restrições são satisfeitas, isto é, o conjunto das restrições ativas:

$$\Omega = \{\mathbf{x} : \mathbf{g}(\mathbf{x}) \leq \mathbf{0}, \mathbf{h}(\mathbf{x}) = \mathbf{0}\} \quad (2.3)$$

a equação (2.2) pode ser reescrita conceitualmente como:

$$\begin{aligned} \min \quad & f(\mathbf{x}) \\ \text{s. a.} \quad & \mathbf{x} \in \Omega \end{aligned} \quad (2.4)$$

Considera-se na equação (2.2) que as restrições são contínuas, o que implica que Ω é um conjunto fechado. A função objetivo $f(\mathbf{x})$ também deve ser contínua para todo $\mathbf{x} \in \mathfrak{R}$ e, preferencialmente, para todo $\mathbf{x} \in \Omega$. Se a região viável é não-vazia e limitada, ou seja, existe $a > 0$ tal que $\|\mathbf{x}\| \leq a \forall \mathbf{x} \in \mathfrak{R}$, então existe uma solução \mathbf{x}^* (teorema de Weierstrass)^[6].

Para um ponto \mathbf{x}^* ser considerado um mínimo, todos os pontos na sua vizinhança devem ter um valor de função superior, ou seja:

- $\mathbf{x}^* = [x_1, x_2, \dots, x_n]^T$ é um *mínimo local fraco* se existir um $\delta > 0$ tal que $f(\mathbf{x}^*) \leq f(\mathbf{x})$ para todo \mathbf{x} tal que $\|\mathbf{x} - \mathbf{x}^*\| < \delta$.

- $\mathbf{x}^* = [x_1, x_2, \dots, x_n]^T$ é um *mínimo local forte* se $f(\mathbf{x}^*) \leq f(\mathbf{x})$ para qualquer \mathbf{x} na vizinhança de \mathbf{x}^* .
- $\mathbf{x}^* = [x_1, x_2, \dots, x_n]^T$ é um *mínimo global* se $f(\mathbf{x}^*) \leq f(\mathbf{x})$ para qualquer $\mathbf{x} \in \Omega$.

2.2 Condições necessárias para a otimalidade

É sabido, por meio do cálculo diferencial, que se \mathbf{x}^* é um mínimo irrestrito de uma função $f(\mathbf{x})$, então $\nabla f(\mathbf{x}^*) = \mathbf{0}$ e o $\nabla^2 f(\mathbf{x}^*)$ é definida positiva. Na otimização restrita é necessário separar as restrições, em restrições de igualdade e desigualdade, para derivar as condições *necessárias* para o ponto de ótimo.

2.2.1 Problema com restrições de igualdade

na derivação das condições necessárias e suficientes para que um ponto seja um minimizador local, sujeito às restrições de igualdade, faz-se necessário introduzir o conceito de plano tangente, figura 2.

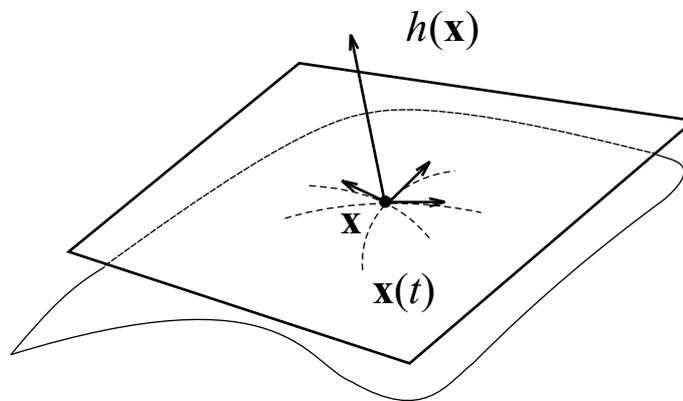


Figura 2: Plano tangente.

Um conjunto de restrições de igualdade no espaço \mathfrak{R}^n

$$\begin{aligned}
 h_1(\mathbf{x}) &= 0 \\
 h_2(\mathbf{x}) &= 0 \\
 &\vdots \\
 h_l(\mathbf{x}) &= 0
 \end{aligned}
 \tag{2.5}$$

define um subconjunto de \mathfrak{R}^n o qual é melhor compreendido como uma hipersuperfície de dimensão $n - l$. As funções h_i , $i = 1, 2, \dots, l$ são contínuas e diferenciáveis, $h_i \in C^1$, e portanto suaves.

Associado a um ponto em uma superfície suave encontra-se o *plano tangente* àquele ponto, o qual tem significado óbvio em duas ou três dimensões. Define-se curva em uma superfície S como a família de pontos $\mathbf{x}(t) \in S$, continuamente parametrizada por t para $a \leq t \leq b$. A curva é diferenciável se $\dot{\mathbf{x}} \equiv (d/dt)\mathbf{x}(t)$ existe e se $\ddot{\mathbf{x}}(t)$ também existe. Uma curva $\mathbf{x}(t)$ passa por um ponto \mathbf{x}^* se $\mathbf{x}^* = \mathbf{x}(t^*)$ para qualquer t^* , $a \leq t^* \leq b$. A derivada da curva em \mathbf{x}^* é definida por $\dot{\mathbf{x}}(t^*)$.

Considera-se agora todas as curvas diferenciáveis em S passando através de um ponto \mathbf{x}^* . O plano tangente em \mathbf{x}^* é definido como a coleção das derivadas em \mathbf{x}^* de todas as curvas diferenciáveis. O plano tangente é um subespaço de \mathfrak{R}^n .

Para as superfícies definidas através do conjunto definido pela equação (2.5) existe uma dificuldade em representar explicitamente o plano tangente. Idealmente poder-se-ia expressar este plano tangente em termos das derivadas das funções h_i que definem a superfície. Pode-se resolver isto introduzindo o subespaço $M = \{\mathbf{y} : \nabla \mathbf{h}(\mathbf{x}^*)\mathbf{y} = \mathbf{0}\}$ e investigando sob quais condições M é igual ao plano tangente em \mathbf{x}^* .

O principal conceito envolvido neste propósito é que \mathbf{x}^* seja um *ponto regular*.

Definição 1. Um ponto \mathbf{x}^* que satisfaça as restrições $\mathbf{h}(\mathbf{x}^*) = \mathbf{0}$ é chamado um ponto *ponto regular* das restrições se os vetores gradiente $\nabla h_1(\mathbf{x}^*), \nabla h_2(\mathbf{x}^*), \dots, \nabla h_l(\mathbf{x}^*)$ são *linearmente independentes* [2].

Assim, em um ponto regular é possível caracterizar o plano tangente em termos dos gradientes das funções de restrição.

Teorema 1. Em um ponto regular \mathbf{x}^* na superfície S definida por $\mathbf{h}(\mathbf{x}) = \mathbf{0}$, o plano tangente é igual a

$$M = \{\mathbf{y} : \nabla \mathbf{h}(\mathbf{x}^*)\mathbf{y} = \mathbf{0}\} \quad (2.6)$$

■

É importante reconhecer que a condição de ser um ponto regular não é uma condição da superfície de restrição, mas sua representação em termos de \mathbf{h} . O plano tangente é definido independentemente de sua representação, enquanto M não o é.

2.2.1.1 Condição necessária de primeira ordem

Lema 1. *Seja \mathbf{x}^* um ponto regular das restrições $\mathbf{h}(\mathbf{x}^*) = \mathbf{0}$ e um ponto extremo local de f sujeita a estas restrições. Então todo $\mathbf{y} \in \mathfrak{R}^n$ que satisfaça*

$$\nabla \mathbf{h}(\mathbf{x}^*) \mathbf{y} = \mathbf{0} \quad (2.7)$$

também deve satisfazer

$$\nabla f(\mathbf{x}^*) \mathbf{y} = 0 \quad (2.8)$$

■

O lema acima indica que $\nabla f(\mathbf{x}^*)$ é ortogonal ao plano tangente, ou seja $\nabla f(\mathbf{x}^*)$ é uma combinação linear de $\nabla \mathbf{h}(\mathbf{x}^*)$, o que leva a introdução dos *multiplicadores de Lagrange*.

Teorema 2. *Seja \mathbf{x}^* um ponto extremo local de f sujeita às restrições $\mathbf{h}(\mathbf{x}) = \mathbf{0}$. Assumindo que \mathbf{x}^* seja um ponto regular destas restrições, então existe um $\lambda \in \mathfrak{R}^n$ tal que:*

$$\nabla f(\mathbf{x}^*) + \lambda^T \nabla \mathbf{h}(\mathbf{x}^*) = \mathbf{0} \quad (2.9)$$

■

A equação (2.9) é chamada de condição necessária de primeira ordem. Juntamente com as restrições, elas formam um conjunto de $n + l$ equações com $n + l$ variáveis englobando \mathbf{x}^* e λ . Assim, estas condições necessárias formam um conjunto completo, desde que, pelo menos localmente, elas determinem uma solução única.

Definindo o Lagrangeano (L) associado ao problema das restrições como:

$$L(\mathbf{x}^*, \lambda) = f(\mathbf{x}) + \lambda^T \mathbf{h}(\mathbf{x}) \quad (2.10)$$

pode-se expressar as condições necessárias de primeira ordem na forma

$$\begin{aligned} \nabla_{\mathbf{x}} L(\mathbf{x}, \lambda) &= \mathbf{0} \\ \nabla_{\lambda} L(\mathbf{x}, \lambda) &= \mathbf{0} \end{aligned} \quad (2.11)$$

2.2.1.2 Condição necessária de segunda ordem

Para o desenvolvimento desta seção é admitido que $f, \mathbf{h} \in C^2$.

Se \mathbf{x}^* é um mínimo local de f sujeita a $\mathbf{h}(\mathbf{x}) = \mathbf{0}$ e que \mathbf{x}^* é um ponto regular destas restrições, então existe um $\lambda \in \Re^l$ tal que $\nabla f(\mathbf{x}^*) + \lambda^T \nabla \mathbf{h}(\mathbf{x}^*) = \mathbf{0}$. Se chamarmos de M o plano tangente $M = \{\mathbf{y} : \nabla \mathbf{h}(\mathbf{x}^*)\mathbf{y} = \mathbf{0}\}$, então a matriz

$$\nabla^2 L(\mathbf{x}^*) = \nabla^2 f(\mathbf{x}^*) + \lambda^T \nabla^2 \mathbf{h}(\mathbf{x}^*) \quad (2.12)$$

é definida positiva em M , ou seja, $\mathbf{y}^T \nabla^2 L(\mathbf{x}^*)\mathbf{y} \geq 0$ para todo $\mathbf{y} \in M$.

2.2.1.3 Condição suficiente de segunda ordem

Se existe um ponto \mathbf{x}^* satisfazendo $\mathbf{h}(\mathbf{x}^*) = \mathbf{0}$ e um $\lambda \in \Re^l$ tal que $\nabla f(\mathbf{x}^*) + \lambda^T \nabla \mathbf{h}(\mathbf{x}^*) = \mathbf{0}$ e a matriz dada pela equação (2.12) é positiva definida no plano tangente M , isto é, para $\mathbf{y} \in M$, $\mathbf{y} \neq \mathbf{0}$, isto assegura que $\mathbf{y}^T \nabla^2 L(\mathbf{x}^*)\mathbf{y} > 0$. Então \mathbf{x}^* é um *mínimo local estrito* de f sujeita a $\mathbf{h}(\mathbf{x}) = \mathbf{0}$.

2.2.2 Problemas com restrições de desigualdade

Para o problema com restrições de desigualdade:

$$\min f(\mathbf{x}) \quad (2.13a)$$

$$\text{s.a. } \mathbf{g}_i(\mathbf{x}) \leq \mathbf{0}, \quad i = 1, \dots, m \quad (2.13b)$$

faz-se necessário apresentar primeiro o conceito de *direção descendente* e *direção viável*.

Direção Descendente: A direção \mathbf{d} é um vetor no espaço das variáveis de projeto, isto é o espaço x_1, x_2, \dots, x_n , ou ainda $\mathbf{x} \in \Re^n$. Admitindo que f seja uma função diferenciável, então o vetor \mathbf{d} é uma direção descendente, no ponto \mathbf{x}_k se:

$$\nabla f^T(\mathbf{x}_k)\mathbf{d} < 0 \quad (2.14)$$

Esta condição assegura que, para um $\alpha > 0$, suficientemente pequeno, a inequação:

$$f(\mathbf{x}_k + \alpha\mathbf{d}) < f(\mathbf{x}_k) \quad (2.15)$$

deve ser mantida.

A interpretação da equação (2.15) é que pode-se reduzir o valor da função percorrendo o espaço das variáveis de projeto, ao longo da direção \mathbf{d} . É conhecido do cálculo

diferencial que o gradiente da função, ∇f , aponta na direção do máximo crescimento da f , assim \mathbf{d} é uma direção descendente se ela indicar o oposto do meio-espaco de ∇f , figura 3.

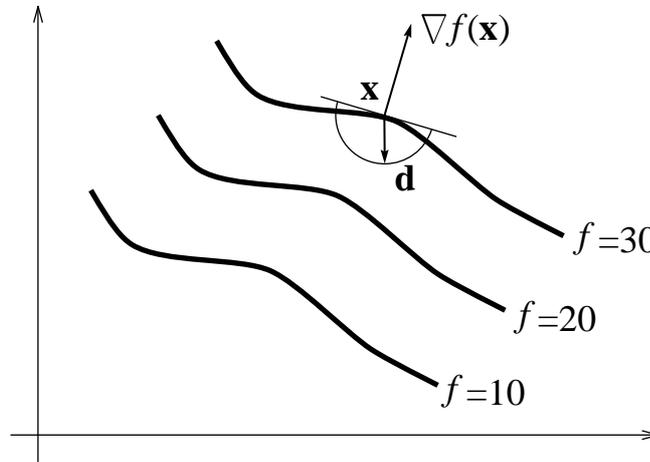


Figura 3: A direção \mathbf{d} e o cone descendente.

Direção Viável: Seja um ponto viável $\mathbf{x}_k \in \Omega$, sendo Ω a região viável definida pelo conjunto de pontos que satisfazem a equação (2.13b). Então \mathbf{d} é uma direção viável se existir um $\bar{\alpha} > 0$ tal que $(\mathbf{x}_k + \alpha \mathbf{d}) \in \Omega$ para todo α , $0 \leq \alpha \leq \bar{\alpha}$.

Sabe-se que $\mathbf{g}(\mathbf{x})$ são funções diferenciáveis, então pode-se definir uma direção viável de outra forma.

Seja um ponto viável \mathbf{x}_k , com uma ou mais restrições *ativas*, ou seja, uma ou mais $g_i = 0$. Chamando o conjunto das restrições ativas por

$$I = \{i : g_i(\mathbf{x}_k) = 0, \quad i = 1, \dots, m\} \quad (2.16)$$

e definindo \mathbf{x}_k como um ponto regular (\mathbf{x}_k é um ponto viável), se o vetor gradiente $\nabla \mathbf{g}_i(\mathbf{x}_k)$, para todo $i \in I$, são linearmente independentes.

Então, um vetor \mathbf{d} é uma direção viável se

$$\nabla \mathbf{g}_i(\mathbf{x}_k)^T \mathbf{d} < 0, \quad \text{para todo } i \in I \quad (2.17)$$

o que garante que, para um $\alpha > 0$, suficientemente pequeno, $(\mathbf{x}_k + \alpha \mathbf{d})$ será viável ou $\mathbf{g}_i(\mathbf{x}_k + \alpha \mathbf{d}) > 0$, $i = 1, \dots, m$. De fato, para cada $i \in I$, a equação (2.17) define um

meio-espaco e a intersecção de todos estes meio-espacos formam um cone viável, dentro do qual se encontrará \mathbf{d} , figura 4.

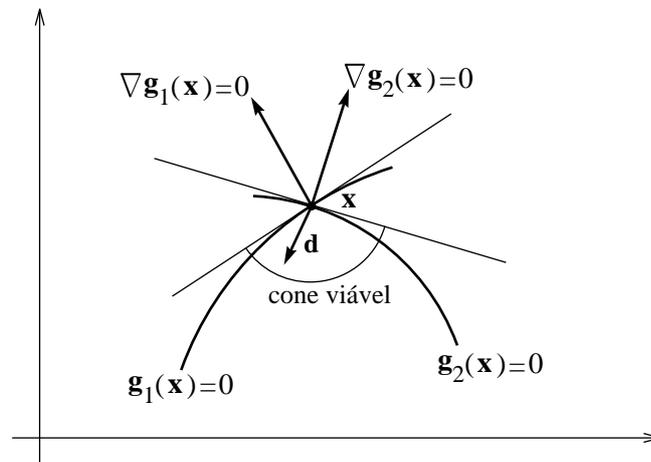


Figura 4: O cone viável e a direção viável.

Tendo sido definidos os conceitos de direção viável e decrescente, pode-se derivar as condições de otimalidade para o problema das restrições de desigualdade, equação (2.13).

Considere-se primeiro o caso quando, no ponto \mathbf{x}_k não existe nenhuma restrição ativa. Neste caso \mathbf{x}_k é um ponto interior com respeito às restrições, e a função objetivo será reduzida ao se tomar como direção $-\nabla f(\mathbf{x}_k)$, que é o método do gradiente. O incremento de busca é então uma minimização unidimensional, que pode ser feita por vários métodos tradicionais (Fibonacci, razão áurea, Brent). O ponto assim obtido terá um valor da função f menor que o anterior. Procede-se assim até que o $\nabla f = 0$ e \mathbf{x}_k é a solução da equação (2.13), sendo este um ponto interior. O processo descrito acima é o mesmo que se usa para a solução de problemas irrestritos.

Na solução de problemas de engenharia algumas restrições estarão ativas: podem ser os limites inferiores ou superiores das variáveis, limites de tensão, deformação.

Para este segundo caso, conforme Belegundu^[8], quando o ponto atual \mathbf{x}_k é definido por I , equação (2.16), é necessário que o vetor \mathbf{d} não seja mutuamente uma direção decrescente e viável.

A proposição anterior pode ser provada com o auxílio do lema de Farkas^[8].

Lema 2 (Lema de Farkas). *Dados dois vetores \mathbf{a}^i , $i = 1, \dots, t$ e \mathbf{g} , não existe um vetor*

\mathbf{d} que satisfaça as condições

$$\mathbf{g}^T \mathbf{d} < 0 \quad (2.18)$$

e

$$\mathbf{a}^{iT} \mathbf{d} \geq 0 \quad (2.19)$$

se, e somente se \mathbf{g} puder ser escrito na forma

$$\mathbf{g} = \sum_{i=1}^t \mu_i \mathbf{a}^i, \quad \mu_i > 0 \quad (2.20)$$

■

Então, associando \mathbf{g} com $\nabla f(\mathbf{x}^*)$ e cada componente de \mathbf{a}^i com o negativo do gradiente de uma restrição ativa, $-\nabla \mathbf{g}_i(\mathbf{x}^*)$, $i \in I$, e sendo t o número de restrições ativas, pode-se ver que o vetor \mathbf{d} , o qual é decrescente e viável não pode existir se

$$-\nabla f(\mathbf{x}^*) = \sum_{i \in I} \mu_i \mathbf{g}_i(\mathbf{x}^*), \quad \mu_i \geq 0 \quad (2.21)$$

O significado geométrico disto é que o vetor \mathbf{d} não pode satisfazer as equações (2.14) e (2.17) ao mesmo tempo, se o $-\nabla f$ encontra-se no cone convexo formado pelos vetores $\nabla \mathbf{g}$, $i \in I$. Para o caso de haver apenas uma restrição ativa, a equação (2.21) torná-se $-\nabla f = \nabla g_i$, e o negativo do gradiente da função custo alinha-se com gradiente da restrição, ou seja, não se pode percorrer esta direção, a direção decrescente, sem que a restrição aumente seu valor, $g > 0$.

Condições necessárias para a otimalidade (KKT). Assumindo que \mathbf{x}^* é um ponto regular, então este ponto é mínimo local da função objetivo f , sujeita às restrições $\mathbf{g}_i \leq 0$, $i = 1, \dots, m$ apenas se

$$\frac{\partial L}{\partial x_p} = \frac{\partial f}{\partial x_p} + \sum_{i=1}^m \mu_i \frac{\partial \mathbf{g}_i}{\partial x_p} = 0, \quad p = 1, \dots, n \quad (2.22a)$$

$$\mu_i \geq 0 \quad (2.22b)$$

$$\mu_i \mathbf{g}_i = 0 \quad (2.22c)$$

$$\mathbf{g}_i(\mathbf{x}) \leq 0 \quad (2.22d)$$

onde L é o Lagrangeano, definido como $L = f + \sum_{i=1}^m \mu_i \mathbf{g}_i$ e μ é o multiplicador de Lagrange, associado às restrições de desigualdade. As condições para a otimalidade apre-

sentadas acima são chamadas de condições de Karusch-Kuhn-Tucker^[8], e um ponto que as satisfaça é chamado de ponto KKT. Para que todas as restrições possam ser incorporadas na condição dada pela equação (2.22a), ou seja, para que as restrições inativas possam ser usadas^[3, 8], a equação (2.22c) define $\mu_i = 0$.

2.2.3 Problema com restrições de igualdade e desigualdade

2.2.3.1 Condição necessária para a otimalidade (KKT)

A partir das equações (2.9), (2.21) e (2.22) pode-se estabelecer as condições para a otimalidade do caso geral, representado pela equação (2.2). Supondo que \mathbf{x}^* seja um ponto regular, então ele será um mínimo local da função objetivo f sujeitas às restrições de igualdade e desigualdade apenas se^[8]

$$\frac{\partial L}{\partial x_p} = \frac{\partial f}{\partial x_p} + \sum_{i=1}^m \mu_i \frac{\partial \mathbf{g}_i}{\partial x_p} + \sum_{j=1}^l \lambda_j \frac{\partial \mathbf{h}_j}{\partial x_p} = 0, \quad p = 1, \dots, n \quad (2.23a)$$

$$\mu_i \geq 0, \quad i = 1, \dots, m \quad (2.23b)$$

$$\mu_i \mathbf{g}_i = 0, \quad i = 1, \dots, m \quad (2.23c)$$

$$\mathbf{g}_i(\mathbf{x}) \leq 0, \quad i = 1, \dots, m \quad \text{e} \quad \mathbf{h}_j(\mathbf{x}) = 0, \quad j = 1, \dots, l \quad (2.23d)$$

Segundo Fritzsche^[3], Bazaraa^[4] e Belegundu^[8] as equações acima são também chamadas por *otimalidade* ou *viabilidade dual*, equação (2.23a); a equação (2.23b) por *não-negatividade*; a equação (2.23c) por *complementaridade*, *folga* ou *complementaridade de relaxação*; e *viabilidade* para a equação (2.23d).

2.3 Condições suficientes para a otimalidade

As condições anteriores definem um ponto, candidato a ponto ótimo, pois são condições necessárias, ou seja, um ponto \mathbf{x}^* que satisfaça as equações (2.23) tanto pode ser um ponto de mínimo como também um ponto de sela.

Para que um ponto que tenha satisfeito as condições de KKT seja um mínimo de fato, é necessário que satisfaça também um conjunto de condições chamadas *suficientes*.

Seja f , \mathbf{g} e $\mathbf{h} \in C^2$. Então, um ponto \mathbf{x}^* é um mínimo da equação (2.2), se existirem multiplicadores de Lagrange μ e λ , tal que^[8]

- As condições necessárias de KKT sejam satisfeitas em \mathbf{x}^* e,
- A matriz Hessiana

$$\nabla^2 L(\mathbf{x}^*) = \nabla^2 f(\mathbf{x}^*) + \sum_{i=1}^m \mu_i \nabla^2 \mathbf{g}_i(\mathbf{x}^*) + \sum_{j=1}^l \lambda_j \nabla^2 \mathbf{h}_j(\mathbf{x}^*) \quad (2.24)$$

é *positiva definida* no subespaço de \mathfrak{R}^n conforme as condições:

$$\begin{aligned} \mathbf{y}^T \nabla^2 L(\mathbf{x}^*) \mathbf{y} &> 0, \quad \mathbf{y} \neq \mathbf{0} \\ \nabla \mathbf{h}_j(\mathbf{x}^*)^T \mathbf{y} &= 0, \quad j = 1, \dots, l \quad \text{e} \quad \nabla \mathbf{g}_i(\mathbf{x}^*)^T \mathbf{y} = 0, \quad i \in I, \quad \mu_i > 0 \end{aligned} \quad (2.25)$$

Assim, as condições suficientes de KKT apenas exigem que a matriz Hessiana seja definida positiva no plano tangente M , definido pela equação (2.6), e que $\nabla \mathbf{g}_i(\mathbf{x}^*)^T \mathbf{y} = 0$ para todo i onde $\mathbf{g}_i(\mathbf{x}^*) = 0$ com $\mu_i > 0$.

Método das Direções Viáveis

Luenberger^[2] define os métodos primais como sendo aqueles métodos que trabalham diretamente no problema original, buscando a solução ótima na região viável. Desta forma, cada ponto, durante o processo, é um ponto viável e a função tem seu valor continuamente diminuído.

Estes métodos possuem três grandes vantagens, as quais os tornam aplicáveis a quase todos os problemas de Programação Não Linear, PNL:

- Cada ponto é viável. Se o processo é terminado antes de alcançar a solução final, o último ponto é viável;
- Apresentam características globais de convergência satisfatórias;
- A maioria dos métodos primais não exige uma estrutura especial do problema, como convexidade, podendo, portanto ser aplicados aos problemas gerais de PNL.

A maior desvantagem destes métodos é se manter na região viável, quando as restrições são não lineares, e alguns destes métodos podem falhar também com as restrições de desigualdade, se não forem tomadas algumas precauções.

Assim estes métodos são muito importantes entre os algoritmos de PNL, devido a sua aplicabilidade e simplicidade, particularmente para restrições lineares. A idéia básica,

que está por trás do método das direções viáveis, é movimentar-se através da região viável da seguinte maneira:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d} \quad (3.1)$$

onde \mathbf{d}_k é a direção do movimento e α_k é um escalar não negativo, que indica a distância do movimento. O valor de α é escolhido de forma que \mathbf{x}_{k+1} permaneça na região viável. A direção de pesquisa \mathbf{d}_k é obtida de tal forma que:

1. Um pequeno movimento nesta direção não viole nenhuma restrição;
2. O valor da função objetivo possa ser reduzido.

Uma direção que satisfaça a propriedade (1) é chamada viável (*feasible direction*) e se satisfizer ambas as propriedades (1) e (2) é dita decrescente^[8] (*usable feasible direction*). Esta é a razão porque o método é conhecido como *método das direções viáveis*. Os métodos que se encaixam nesta definição são^[4]: o método de Rosen da projeção do gradiente, o método de Zoutendijk das direções viáveis, programação quadrática recursiva (SQP), o método de Wolfe do gradiente reduzido, o método de gradiente reduzido generalizado (GRG) e o método convex-simplex de Zangwill. Neste trabalho será abordado apenas o método de Zoutendijk, os demais métodos podem ser encontrados em Bazaraa, Luenberger, Fiacco-McCormick, entre outros.

3.1 Método de Zoutendijk

Este método foi desenvolvido por Zoutendijk em 1960, e continua sendo um método eficiente e robusto de otimização, que tanto pode ser usado para restrições de desigualdade como também para restrições de igualdade, necessitando esta última de cuidados especiais^[4, 8].

De um modo geral, este método pode ser iniciado de duas maneiras: se o ponto inicial é um ponto interior, a direção decrescente-viável é tomada como sendo a direção negativa do gradiente da função; por outro lado, se o ponto inicial localiza-se na fronteira da região viável, então algumas restrições estarão ativas e o procedimento a ser seguido é discutido a seguir.

3.1.1 Restrições não-lineares

Considere o seguinte problema

$$\text{minimize } f(\mathbf{x}) \quad (3.2)$$

$$\text{sujeita a } \mathbf{g}_i(\mathbf{x}) \leq \mathbf{0}, \quad i = 1, \dots, m \quad (3.3)$$

Assumindo que f e \mathbf{g}_i tem derivadas contínuas, que não precisam ser convexas^[8], e lembrando-se da definição de direção descendente, equação (2.14), e direção viável, equação (2.17), e sendo \mathbf{x}_k um ponto viável, então, se for possível determinar uma direção que seja decrescente e viável, uma busca ao longo desta direção levará a um ponto com um valor menor da função, figura 5.

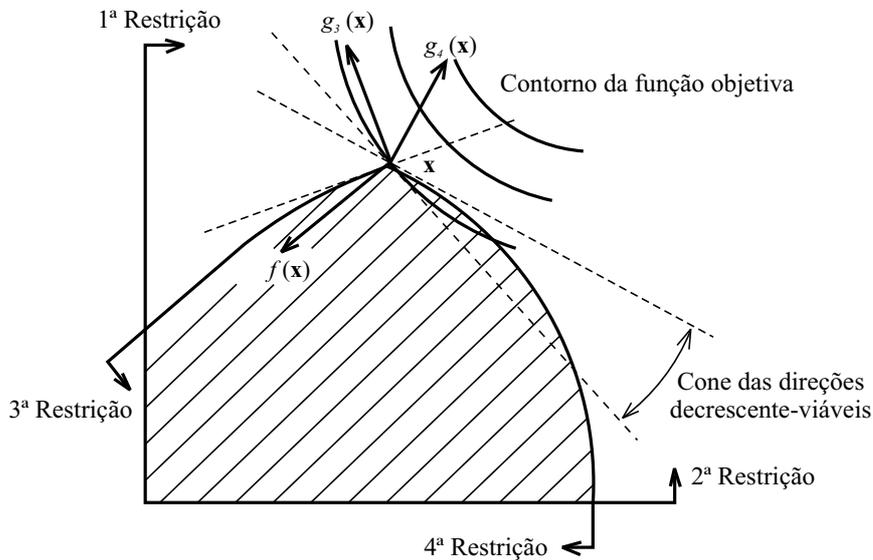


Figura 5: O cone das direções decrescente-viáveis.

É necessário, então, encontrar um vetor direção \mathbf{d} que satisfaça $\nabla f(\mathbf{x}_k)^T \mathbf{d} < 0$ e $\nabla \mathbf{g}_j(\mathbf{x}_k)^T \mathbf{d} < 0$, para $j \in I$, o que é obtido encontrando o mínimo do máximo de $\nabla f(\mathbf{x}_k)^T \mathbf{d}$ e $\nabla \mathbf{g}_{k_j}(\mathbf{x})^T \mathbf{d}$, para $j \in I$. Fazendo $\nabla f = \nabla f(\mathbf{x}_k)$ e $\nabla \mathbf{g}_j = \nabla \mathbf{g}_j(\mathbf{x}_k)$ por simplicidade, e introduzindo uma variável artificial α

$$\alpha = \max\{\nabla f^T \mathbf{d}, \nabla \mathbf{g}_j^T \mathbf{d}, j \in I\} \quad (3.4)$$

pode-se, então, formular um sub-problema que determine a direção viável-decrescente,

figura 6, ou seja

$$\begin{aligned}
 & \min \alpha \\
 & \text{s.a. } \nabla f^T \mathbf{d} \leq \alpha \\
 & \nabla \mathbf{g}_j^T \mathbf{d} \leq \alpha, \quad j \in I \\
 & -1 \leq d_i \leq 1, \quad i = 1, \dots, n
 \end{aligned} \tag{3.5}$$

O problema acima apresenta $n + 1$ variáveis: \mathbf{d} e α , e restringe-se o módulo de d_i ao valor unitário para que a solução seja limitada. O valor negativo de α implica que o produto escalar entre a normal da restrição e o vetor direção também é negativo (no hiperespaço), ou seja, o ângulo entre estes dois vetores é obtuso. Zoutendijk introduziu um fator θ_j (*push-off factor*), para controlar este ângulo: $\nabla \mathbf{g}_j^T \mathbf{d} \leq \theta_j \alpha$, para cada $j \in I$. Uma direção muito próxima à tangente da restrição não é desejada, pois um pequeno avanço nesta direção pode violar a restrição, figura 6. Para as restrições não-lineares a direção deve ser afastada da restrição, por este motivo adota-se $\theta_j > 0$, sendo $\theta_j = 1$ o valor mais citado na literatura^[1, 8].

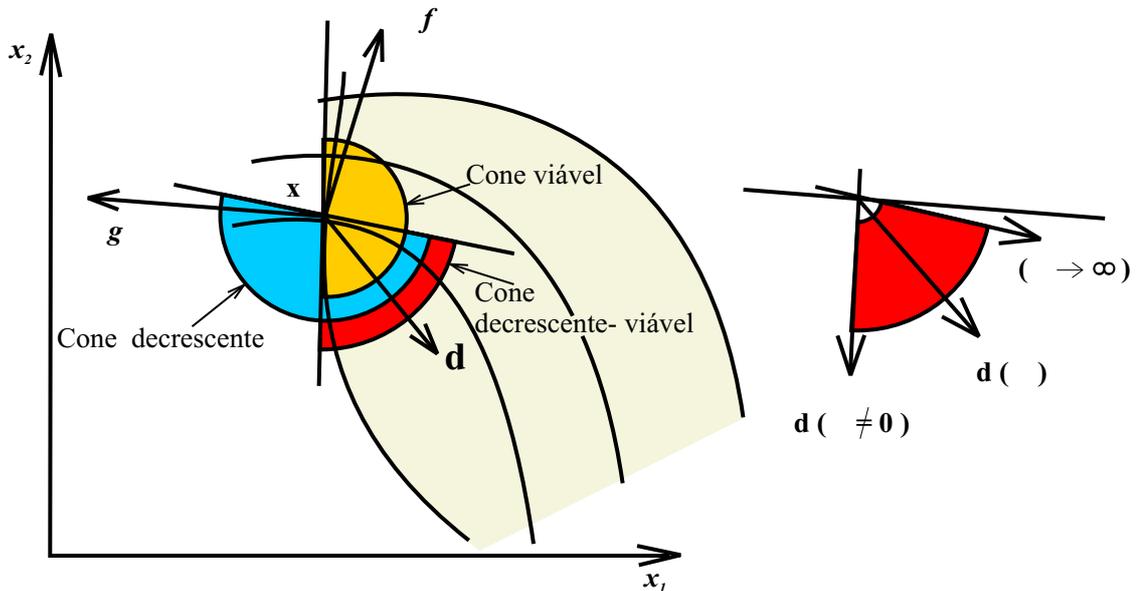


Figura 6: A direção decrescente-viável e o efeito do fator θ .

Sadek^[10] apresenta uma maneira de melhorar a solução da equação (3.5), tornando θ_j uma função do valor de sua restrição ativa correspondente. Ainda, segundo Sadek, quanto mais próximo estiver o ponto, da hipersuperfície de restrição, maior deve ser o fator para afastar (o ponto) daquela superfície. Um valor pequeno, para este fator, resultará

em uma direção que irá diminuir rapidamente a função objetivo, mas, entretanto, poderá encontrar mais adiante, a mesma superfície (da restrição). Valores maiores, por outro lado, reduzem este risco, mas em compensação, não reduzirão tão rapidamente a função objetivo. Para o caso de restrições lineares, $\nabla \mathbf{g}_j^T \mathbf{d}^* = 0$, isto é, quando o vetor direção é normal ao gradiente da restrição, adota-se $\theta_j = 0$.

3.1.2 Problema da busca de direção

Pode-se ver que a equação (3.5) é linear em termos das variáveis $d_1, d_2, \dots, d_n, \alpha$ assim como a normalização do vetor \mathbf{d} . Desta forma este problema pode ser resolvido por meio da programação linear, na qual Zoutendijk^[1] propôs uma modificação, tornando a solução mais eficiente. Assim, explicitamente tem-se:

$$\text{minimize } \alpha$$

sujeito a

$$\begin{aligned} d_1 \frac{\partial f}{\partial x_1} + d_2 \frac{\partial f}{\partial x_2} + \dots + d_n \frac{\partial f}{\partial x_n} - \alpha &\leq 0 \\ d_1 \frac{\partial g_1}{\partial x_1} + d_2 \frac{\partial g_1}{\partial x_2} + \dots + d_n \frac{\partial g_1}{\partial x_n} - \theta_1 \alpha &\leq 0 \\ d_1 \frac{\partial g_2}{\partial x_1} + d_2 \frac{\partial g_2}{\partial x_2} + \dots + d_n \frac{\partial g_2}{\partial x_n} - \theta_2 \alpha &\leq 0 \\ &\vdots \\ d_1 \frac{\partial g_j}{\partial x_1} + d_2 \frac{\partial g_j}{\partial x_2} + \dots + d_n \frac{\partial g_j}{\partial x_n} - \theta_j \alpha &\leq 0 \\ d_1 - 1 &\leq 0 \\ d_2 - 1 &\leq 0 \\ &\vdots \\ d_n - 1 &\leq 0 \\ -1 - d_1 &\leq 0 \\ -1 - d_2 &\leq 0 \\ &\vdots \\ -1 - d_n &\leq 0 \end{aligned} \tag{3.6}$$

onde j é o número de restrições ativas e as derivadas parciais foram avaliadas para o ponto \mathbf{x} . Os componentes da direção, d_i , $i = 1, \dots, n$, admitem qualquer valor entre -1

e 1, podendo, então, ser substituídos por uma nova variável $s_i = d_i + 1$, $i = 1, \dots, n$, tornando-as não-negativas, e fazendo $\beta = -\alpha$. Com estas modificações o problema acima pode ser reescrito como um problema de programação linear convencional:

$$\begin{aligned}
 & \text{minimize } -\beta \\
 & \text{sujeito a } \nabla f^T \mathbf{s} + \beta \leq \sum_{i=1}^n \frac{\partial f}{\partial x_i} \\
 & \nabla \mathbf{g}_j^T \mathbf{s} + \theta_j \beta \leq \sum_{i=1}^n \frac{\partial \mathbf{g}_j}{\partial x_i}, \text{ para cada } j \in I \\
 & s_i \leq 2, \quad i = 1, \dots, n \\
 & s_i \geq 0, \quad \beta \geq 0
 \end{aligned} \tag{3.7}$$

A equação (3.7) é um problema básico de programação linear que pode ser resolvido com o método Simplex, por exemplo. Da solução desta equação, $\beta^* > 0$ ou $\alpha^* < 0$ é equivalente a $\nabla f^T \mathbf{d} < 0$ e $\nabla \mathbf{g}_j^T \mathbf{d} < 0$, para cada restrição ativa, e $\beta^* > 0$ implica que \mathbf{d}^* é uma direção decrescente-viável. Mas se $\alpha^* = 0$ então pode-se provar que \mathbf{x}_k é um ponto KKT^[1, 8].

3.1.3 Determinação do tamanho do intervalo (*step size*)

A determinação do tamanho do intervalo é uma minimização unidimensional restrita e pode ser expressa como:

$$\begin{aligned}
 & \min f(\mathbf{x}_k + \alpha \mathbf{d}) \\
 & \text{s.a. } \mathbf{g}_i(\mathbf{x}_k + \alpha \mathbf{d}) \leq 0, \quad i = 1, \dots, m
 \end{aligned} \tag{3.8}$$

Pode-se resolver a equação (3.8) em duas etapas: primeiro escolher um tamanho de intervalo conveniente, e segundo, corrigi-lo de forma a reduzir a função objetivo tanto quanto possível, sem violar nenhuma restrição. Vários são os métodos para a solução do problema acima, e no Capítulo 5 será explicitado o método implementado.

3.1.4 Parâmetro ϵ de ativação das restrições

Devido a imprecisão numérica de se obter $\nabla g = 0$, causada pelas aproximações utilizadas, torna-se necessário introduzir uma tolerância ϵ , dentro da qual considera-se que a restrição está ativa. Deve-se tomar cuidado ao escolher este valor, pois um valor

inicial muito pequeno poderá não ativar restrições na iteração corrente e, ativá-las na próxima iteração, resultando em oscilações (efeito de *zigzag*). Um valor maior, por outro lado, pode ser usado como primeira escolha para se obter um ponto melhor, e então usá-lo como ponto inicial para uma nova solução, com um ϵ de menor valor. Pode-se ver na figura 7 que ao se relaxar um pouco o parâmetro ϵ a nova direção $\bar{\mathbf{d}}$ aponta para uma solução melhor.

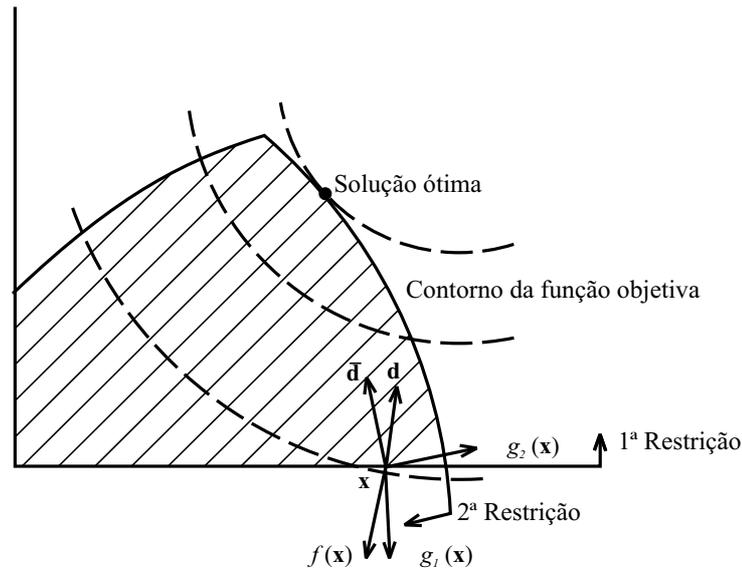


Figura 7: O efeito do parâmetro ϵ na determinação da direção \mathbf{d} .

3.1.5 Algoritmo básico do método de Zoutendijk

O algoritmo básico do método de Zoutendijk das direções viáveis procura determinar uma direção que seja viável e decrescente ao mesmo tempo. Para a situação em que o ponto atual é um ponto interior, toma-se como direção o negativo do gradiente da função: $\mathbf{d} = -\nabla f(\mathbf{x}_k)$, e se alguma restrição encontra-se ativa, procura-se uma direção decrescente-viável que satisfaça as equações (2.14) e (2.17). O algoritmo é apresentado abaixo:

Algoritmo

1. $i = 1$;

Defina um ponto inicial viável: \mathbf{x}_i e parâmetros $\epsilon, \epsilon_1, \epsilon_2$ e ϵ_3 ;

Calcule $f(\mathbf{x}_i)$ e $\mathbf{g}_j(\mathbf{x}_i)$, $j = 1, \dots, m$.

2. Se $\mathbf{g}_j(\mathbf{x}_i) < 0$, $j = 1, \dots, m$, então:

$$\mathbf{d} = -\nabla f(\mathbf{x}_i)$$

Normalize \mathbf{d} , vá para **5**.

Se existir $\mathbf{g}_j(\mathbf{x}_i) = 0$, vá para **3**.

3. Encontre uma direção viável \mathbf{d} , resolvendo o problema da busca de direção:

$$\begin{aligned} & \min -\beta \\ & \text{sujeito a } \nabla f^T \mathbf{s} + \beta \leq \sum_{i=1}^n \frac{\partial f}{\partial x_i} \\ & \nabla \mathbf{g}_j^T \mathbf{s} + \theta_j \beta \leq \sum_{i=1}^n \frac{\partial \mathbf{g}_j}{\partial x_i}, \text{ para cada } j \in I \\ & s_i \leq 2, \quad i = 1, \dots, n \\ & s_i \geq 0, \quad \beta \geq 0 \end{aligned}$$

4. Se $\beta^* \leq \epsilon_1$, $\mathbf{x}^* = \mathbf{x}_i$. Fim.

Se $\beta^* > \epsilon_1$, faça $\mathbf{d} = \mathbf{s}$, vá para **5**.

5. Determine um tamanho de intervalo α , ao longo da direção \mathbf{d} e faça:

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \alpha \mathbf{d}$$

6. Calcule a função objetivo $f(\mathbf{x}_{i+1})$.

7. Verifique a convergência:

$$\text{Se: } \left\| \frac{f(\mathbf{x}_i) - f(\mathbf{x}_{i+1})}{f(\mathbf{x}_i)} \right\| \leq \epsilon_2 \text{ e } \|\mathbf{x}_i - \mathbf{x}_{i+1}\| \leq \epsilon_3, \text{ faça } \mathbf{x}^* = \mathbf{x}_{i+1}. \text{ Fim.}$$

Caso contrário vá para **8**.

8. Faça $i = i + 1$. Vá para **2**.

CAPÍTULO 4

Otimização de Forma

O problema geral da otimização de forma é encontrar a melhor geometria possível, pela redução de uma função objetivo, normalmente o peso, sujeita a restrições de tensão, frequência, deformação ou de manufatura. Os métodos mais comuns empregados escolhem como variáveis de projeto os coeficientes polinomiais, os pontos de controle das curvas *splines*, as coordenadas dos pontos de controle e outros parâmetros geométricos. Nos primeiros trabalhos de otimização de forma, realizados por Zienkiewicz e Campbell^[11], a própria malha de elementos finitos era usada como variável de projeto. Esta idéia foi logo descartada, pois era extremamente difícil manter um contorno suave, e na realidade a forma obtida mais se parecia com um serrilhado.

Apenas recentemente os programas comerciais de análise estrutural começaram a oferecer a otimização com uma opção para os analistas. O que havia até então, eram programas desenvolvidos *in house*, muitos deles na indústria automobilística e aeroespacial.

Este trabalho irá discutir e implementar a otimização de forma baseada em variáveis naturais, assim definido por Belegundu e Rajan^[12], e também desenvolvido por Chargin et al^[13], ou seja, as variáveis de projeto do problema representam as modificações na malha de elementos finitos, causadas por cargas unitárias ou campos de pressão virtuais. Estas variações são adicionadas à malha inicial para gerar uma nova forma. O método

das direções viáveis de Zoutendijk é utilizado para definir um novo conjunto de variáveis que levem à forma ótima. De uma forma simplificada, o método pode ser compreendido como um “martelamento” da superfície do sólido, feito pelas cargas virtuais, que levam a uma forma otimizada, reduzindo o valor da função custo e obedecendo às restrições.

Dado um corpo sólido (contínuo) que ocupa uma região V no espaço xyz , limitado por uma superfície S e com cargas, condições de contorno e distribuição de material, deve-se otimizar a forma deste sólido definindo a seguinte expressão:

$$\begin{aligned} \min f(\mathbf{x}, \mathbf{u}(\mathbf{x})) \\ \text{s.a. } \mathbf{g}_i(\mathbf{x}, \mathbf{u}(\mathbf{x})) \leq 0, \quad i = 1, \dots, m \end{aligned} \quad (4.1)$$

onde \mathbf{x} é o vetor das variáveis de projeto, \mathbf{u} é o vetor dos deslocamentos nodais determinado pela equação de equilíbrio do método dos elementos finitos:

$$K(\mathbf{x})\mathbf{u} = \mathbf{F} \quad (4.2)$$

onde $K(\mathbf{x})$ é a matriz rigidez e \mathbf{F} é o vetor força. As funções f e g_i são funções implícitas das variáveis de projeto \mathbf{x} , ou seja, dependem do vetor de deslocamentos $\mathbf{u}(\mathbf{x})$.

Na equação (4.1) a função objetivo f é o volume da estrutura e a restrição g_i é o valor limite da tensão ou deslocamento. É importante expressá-la na forma normalizada, assegurando que o valor da restrição fique limitado ao intervalo $[0, 1]$. Assim, ao ser estabelecido um valor limite para a tensão, por exemplo, 75MPa, a restrição assume a forma: $g_i = \frac{\sigma_i}{75} - 1 \leq 0, i = 1, \dots, ncon$, onde $ncon$ é o número de restrições.

A formulação do problema na equação (4.1) pode ser generalizada para manipular restrições de igualdade ou restrições na frequência natural do sistema^[10, 14]

4.1 Método das variáveis naturais

4.1.1 Campos de velocidade

O método das variáveis naturais utiliza como variável de projeto cargas fictícias, em pontos de controle (nós), figura 8, selecionados adequadamente. As deformações nodais, devidas a estes carregamentos, são somadas à malha atual para a obtenção de uma nova malha.

Com isto é possível estabelecer uma relação entre os nós da malha e as variáveis

de projeto:

$$\mathbf{G}(\mathbf{x}) = \mathbf{G}^{old} + \sum_{i=1}^n x_i \mathbf{q}_i \quad (4.3)$$

onde \mathbf{G} é o vetor ($n \times 1$) das coordenadas nodais x , y e z atual, \mathbf{q}^i é o vetor ($n \times 1$) que representa as mudanças nas posições nodais, causadas por uma força unitária aplicada no i -ésimo grau de liberdade. Pode-se ver que a matriz de rigidez é uma função das variáveis de projeto: $\mathbf{K}(\mathbf{G}(\mathbf{x}))$. O nome de campos de velocidade vem da semelhança que a equação (4.3) tem com a equação da mecânica do contínuo, que descreve o movimento de um meio, com x_i representando o parâmetro tempo e \mathbf{q}^i como o vetor velocidade. Assim, temos:

$$\left. \frac{\partial \mathbf{G}}{\partial x_i} \right|_{x_i=0} = \mathbf{q}^i \quad (4.4)$$

A equação (4.3) também pode ser escrita como:

$$\mathbf{G}(\mathbf{x}) = \mathbf{G}^{old} + [Q]\mathbf{q}_i \quad (4.5)$$

sendo que as colunas da matriz Q são formadas pelos vetores \mathbf{q}^i . Cada vetor \mathbf{q}^i , por sua vez, é obtido pela equação:

$$K_a \mathbf{q}^i = \mathbf{1}^i \quad (4.6)$$

onde $\mathbf{1}^i$ é o vetor carregamento com seus elementos iguais a zero, com exceção da i -ésima posição, cujo valor é a unidade. Isto é exatamente o carregamento virtual, representado por uma força de valor unitário (há a opção de usar um deslocamento nodal unitário). A matriz de rigidez da equação anterior, K_a é uma matriz rigidez diferente da utilizada na avaliação da função objetivo e das restrições. Esta matriz é obtida com condições de contorno diferentes daquelas do problema original.

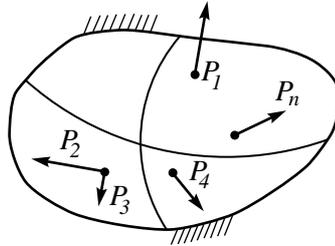


Figura 8: Carregamentos nodais aplicados aos nós de controle.

O método de se obter o campo de velocidades, a partir de deformações causadas na malha de elementos finitos, foi primeiro desenvolvido por Belegundu e Rajan^[12] e por Chargin^[13], de forma independente. Este método consiste, basicamente, em gerar um vetor de deslocamento, ou seja, a função natural de forma, a partir de uma solução estática, para um dado número de cargas aplicadas, ou deslocamentos prescritos. Este conceito é melhor ilustrado aplicando-o no exemplo de uma placa, bi-axialmente carregada, com um furo no centro. Nesta placa o interesse está em modificar a região interna (furo), sem alterar o contorno externo. A figura 9 mostra as condições de contorno originais deste problema, aqui representado em apenas um quarto, devido a sua geometria ser simétrica.

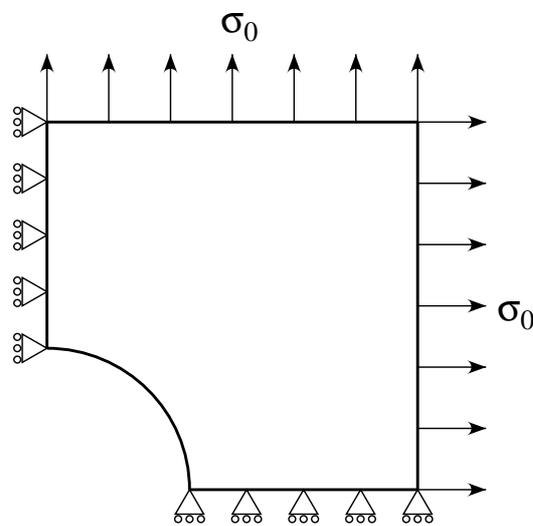


Figura 9: Placa submetida a um carregamento bi-axial (estrutura principal).

A estrutura inicial, que será usada para o cálculo da função objetivo e da restrição, recebe o nome de *estrutura principal*, pois ela define o domínio da estrutura e as suas condições de contorno. Chama-se de *estrutura auxiliar* a estrutura que será utilizada unicamente para aplicação das cargas ou dos deslocamentos virtuais prescritos. A figura 10 mostra tal estrutura. Aplica-se neste corpo as cargas virtuais unitárias, ou os deslocamentos prescritos, nos nós que formam a região na qual existe o interesse em modificar, neste caso, o furo central.

A deformação, causada por cada uma das cargas fictícias, é obtida por meio de uma análise de elementos finitos, e o campo resultante de deformações irá compor o vetor \mathbf{q}^i . Deve-se observar que o contorno externo, figura 11, não sofre deformação alguma, como se deseja.

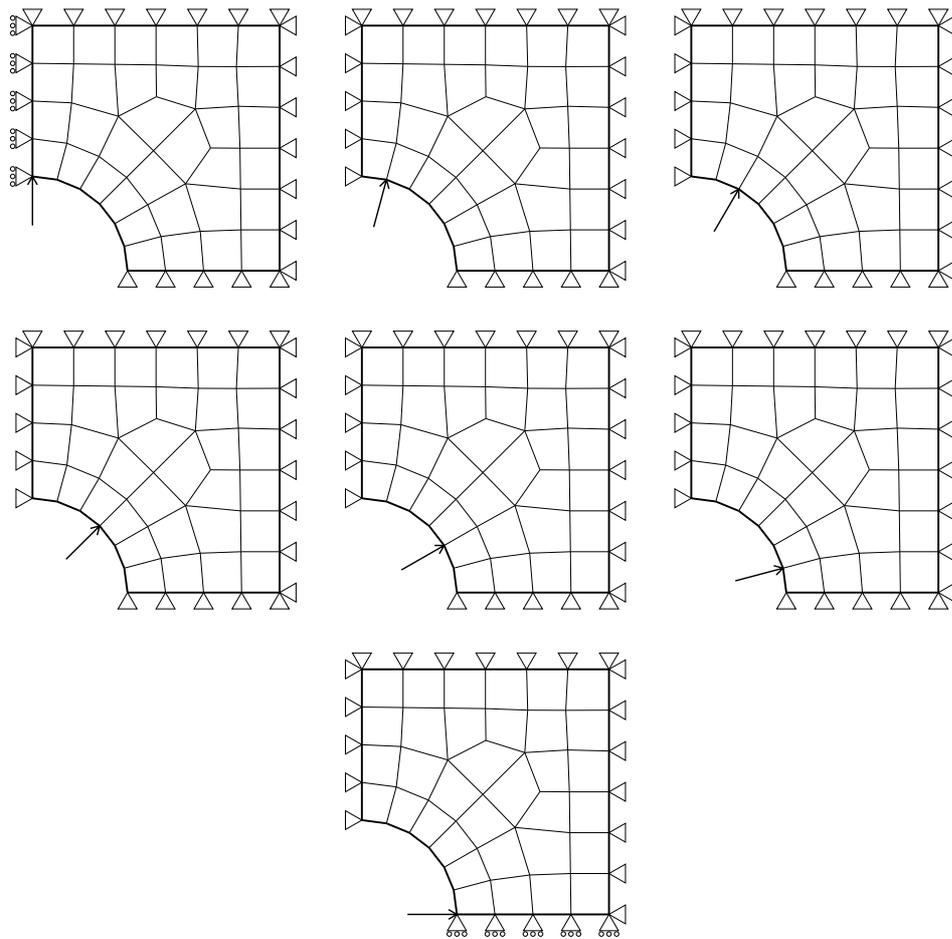


Figura 10: Estrutura auxiliar e as cargas unitárias.

Uma análise da estrutura deformada, pelo conjunto de cargas virtuais, deve ser feita, pois a forma final é uma combinação dos vários campos de deformação obtidos. Isto é facilmente verificado plotando o resultado de cada uma das análises por meio da seguinte relação:

$$\mathbf{G}(\mathbf{x}) = \mathbf{G}^{old} + s\mathbf{q}^i \quad (4.7)$$

onde s é um fator de escala adequado.

Para os casos onde se deseja gerar um campo de velocidades interno e externo, deve-se proceder da seguinte maneira: escolher uma superfície, a externa, por exemplo, e travá-la. Aplicar nos nós que compõem a superfície interna as cargas virtuais. Feito isto, inverte-se o procedimento, travando a superfície interna, e aplicam-se as cargas virtuais na superfície externa para gerar os campos de deformação, figura 12.

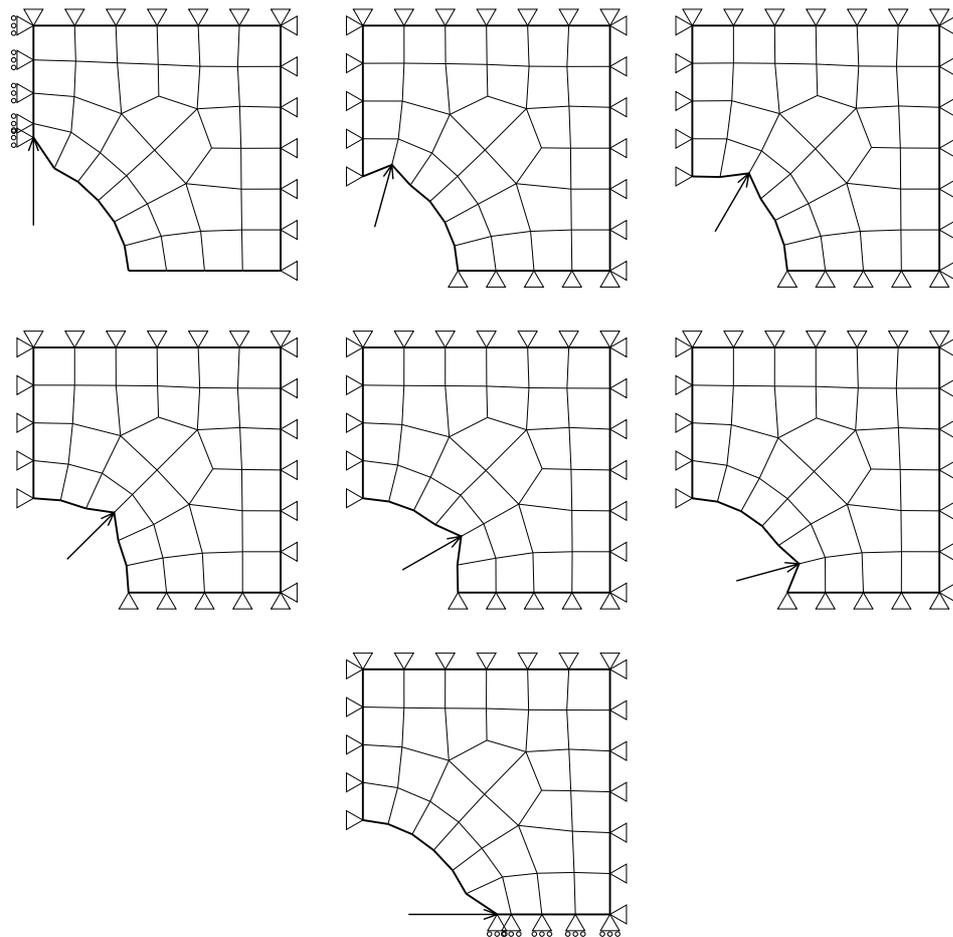


Figura 11: Deformações causadas pelas cargas unitárias na estrutura auxiliar.

4.1.2 Escolha das variáveis de projeto

No método descrito na seção anterior são usadas cargas fictícias como variáveis de projeto. Pode-se, então, escolher o tipo de carregamento virtual que mais se adapte ao tipo de problema. De um modo geral, existem três tipos de carregamento, que são mostrados na figura 13.

Na figura 13(a) a carga é aplicada na direção das coordenadas do problema; na figura 13(b), normal à superfície; e na figura 13(c) o carregamento é representado por um campo de pressão, ao longo de uma superfície. A principal vantagem do uso de uma carga normal ou do campo de pressão é poder trabalhar com um menor número de variáveis de projeto. Se o analista achar melhor, pode também trabalhar com deslocamentos prescritos. Comentários adicionais serão feitos no Capítulo 6. A figura 4.7 evidencia o campo de deformação causado pela aplicação de um campo de pressão.

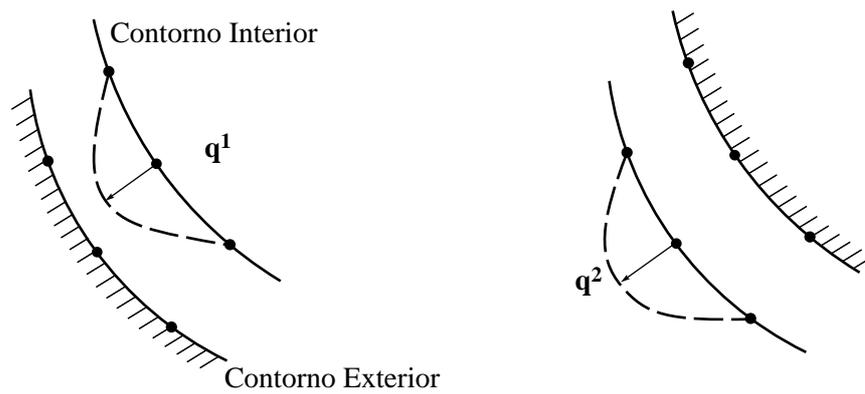


Figura 12: Geração do campo de velocidade interno e externo.

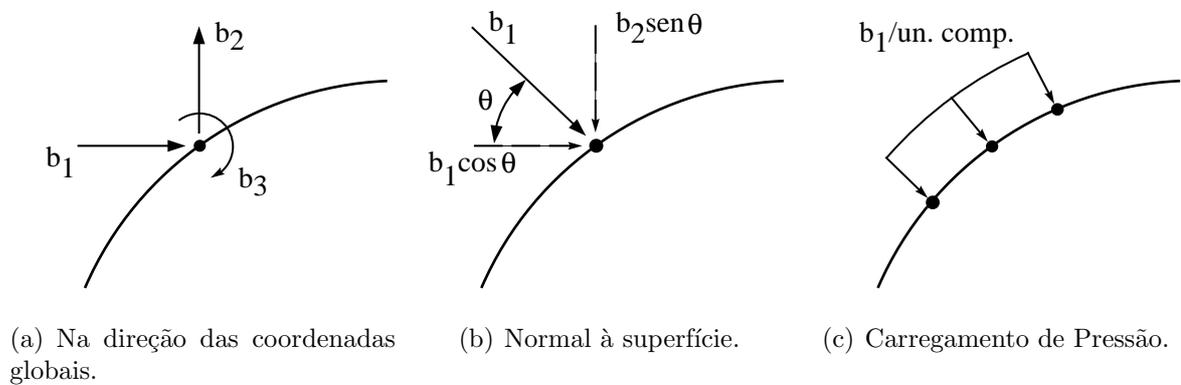


Figura 13: Tipos de carregamentos virtuais.

4.1.3 Algoritmo básico

A partir do exposto anteriormente pode-se estabelecer um algoritmo básico que descreve cada uma das etapas necessárias à obtenção de uma nova e viável forma. Seja k a iteração corrente, então:

1. Defina $\mathbf{G}^{(k)}$ como a forma atual (malha do MEF). Faça $\mathbf{x}^{(k)} = 0$.
2. Aplique as cargas unitárias, nos “pontos de controle”, uma por vez, e resolva a equação (4.6) para cada \mathbf{q}^i . Observar que este procedimento vincula as coordenadas nodais às variáveis de projeto.
3. Determine a partir da equação 4.1 os gradientes $\nabla f(\mathbf{x})$ e $\nabla \mathbf{g}_i(\mathbf{x})$.
4. Com os gradientes calculados em 3, resolva o problema de PNL para obter um novo conjunto solução $\mathbf{x}^{(k)}$.

5. Defina a nova forma fazendo:

$$\mathbf{G}^{(k+1)} = \mathbf{G}^{(k)} + [\mathbf{Q}]\mathbf{x}^{(k)} \quad (4.8)$$

4.2 Análise de sensibilidade

Análise de sensibilidade é a maneira pela qual pode-se obter informações do comportamento da função objetivo ao se modificarem as suas variáveis. Historicamente, a aplicação da análise de sensibilidade começou em meados dos anos 60^[15], onde dois métodos predominaram: o método direto (*pseudo-load*) e o método da carga virtual (*virtual load*). Nos anos 70 o método adjunto (*adjoint variable method*), adaptado da teoria do controle ótimo, foi aplicado à otimização (Haug e Arora). A referência 15 faz uma comparação entre estes três métodos citados.

Para o problema clássico de PNL, equação (2.1), um algoritmo de otimização baseado nos gradientes normalmente irá requerer:

- A sensibilidade da função objetivo: $\partial f / \partial x_i$ ($n \times 1$).
- A sensibilidade de todas as restrições ativas no ponto atual: $\partial g_j / \partial x_k$ ($m \times n$).

4.2.1 Diferenças finitas

Diferença finita é a mais comum das formas de se calcular a sensibilidade. Apesar de sua aproximação não ser particularmente precisa e eficiente, é o método de mais fácil implementação^[16]. Todas as fórmulas das diferenças finitas (*forward difference*, *backward difference* e *central difference*) podem ser calculadas a partir da expansão das séries de Taylor em torno de um ponto x :

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2!}f''(x) + \frac{h^3}{3!}f'''(x) + \dots \quad (4.9)$$

resolvendo para f'

$$f'(x) = \frac{f(x+h) - f(x)}{h} + \mathcal{O}(h) \quad (4.10)$$

onde h é o chamado intervalo da diferença finita e $\mathcal{O}(h)$ é o erro de truncamento, definido assim pela equação (4.10) como a aproximação de primeira ordem.

Para uma aproximação de segunda ordem basta expandir para $f(x - h)$:

$$f(x - h) = f(x) - hf'(x) + \frac{h^2}{2!}f''(x) - \frac{h^3}{3!}f'''(x) + \dots \quad (4.11)$$

subtraindo a equação (4.11) da equação (4.9) e resolvendo para f' se obtém a equação da diferença central:

$$f'(x) = \frac{f(x + h) - f(x - h)}{2h} + \mathcal{O}(h^2) \quad (4.12)$$

Deve-se tomar cuidado quando usar estas equações, devido ao dilema do tamanho do incremento (*step size dilemma*), isto é, deseja-se escolher um valor pequeno para minimizar os erros de truncamento, e que, ao mesmo tempo, não seja pequeno o suficiente para tornar os erros dos termos subtraídos dominante.

4.2.2 Método direto(*pseudo-load vector*)

Neste método aplica-se a diferenciação direta da função. Para a função das restrições $g_i(\mathbf{x}, u)$, por exemplo, os coeficientes de sensibilidade são dados por:

$$\frac{dg_i}{d\mathbf{x}} = \frac{\partial g_i}{\partial \mathbf{x}} + \frac{\partial g_i}{\partial u} \frac{du}{d\mathbf{x}} \quad (4.13)$$

onde $\partial g_i/\partial \mathbf{x}$ e $\partial g_i/\partial u$ são prontamente calculadas, e $du/d\mathbf{x}$ é obtida diferenciando a equação de equilíbrio, equação (4.2):

$$\frac{\partial u}{\partial \mathbf{x}} K + u \frac{\partial K}{\partial \mathbf{x}} = \frac{\partial F}{\partial \mathbf{x}} \quad (4.14)$$

e rearranjando os termos:

$$K \frac{du}{d\mathbf{x}} = - \frac{\partial}{\partial \mathbf{x}} (K\hat{u} - F) \quad (4.15)$$

onde \hat{u} é o vetor deslocamento obtido da equação (4.2), e o termo entre parênteses é o que dá nome ao método. O valor da derivada da matriz rigidez, $\partial K/\partial x_j$, pode ser exatamente calculada fazendo

$$\frac{\partial K}{\partial x_j} = \sum_{l=1}^n \frac{\partial K}{\partial G_l} \frac{\partial G_l}{\partial x_j} \quad (4.16)$$

e da equação (4.5)

$$\frac{\partial K}{\partial x_j} = \sum_{l=1}^n \frac{\partial K}{\partial G_l} Q_{lj} \quad (4.17)$$

onde $\partial K/\partial x_j$, $j = 1, \dots, n$, é a derivada da matriz rigidez com relação às coordenadas nodais. Contudo este cálculo é bastante trabalhoso, e não se traduz em precisão e economia de tempo, quando comparado às diferenças finitas ^[12]. Assim, pode-se efetuar esta

derivada numericamente:

$$\frac{\partial K}{\partial x_j} \approx \frac{K(x_1, \dots, x_j + \epsilon, \dots, x_k) - K(x_1, \dots, x_j)}{\epsilon} \quad (4.18)$$

onde ϵ é o tamanho do incremento. Segundo Belegundu, mesmo que para calcular a equação (4.18) seja necessário montar k vezes a matriz rigidez, ainda é mais rápido do que decompor K para resolver a equação (4.2). Por este motivo, a equação (4.18) é usada em vez da equação (4.17) para calcular $\partial K/\partial x_j$. O uso da equação (4.18) na equação (4.15) é conhecido como método semi-analítico para o cálculo da sensibilidade.

O vetor das coordenadas nodais, para um j -ésimo elemento, sofrendo um perturbação ϵ , pode ser escrita de acordo com a equação (4.3):

$$\mathbf{G}_\epsilon = \mathbf{G}^0 + \epsilon \mathbf{q}^j \quad (4.19)$$

onde \mathbf{G}_ϵ corresponde à forma atual. Desta forma a equação (4.18) pode ser definida como:

$$\frac{\partial K}{\partial x_j} \approx \frac{K(\mathbf{G}_\epsilon) - K(\mathbf{G}^0)}{\epsilon} \quad (4.20)$$

Cabe aqui uma advertência quanto a escolha do parâmetro ϵ . No método baseado em variáveis naturais, este parâmetro é uma carga aplicada à estrutura (ou um deslocamento prescrito), e por este motivo deve ser escolhido grande o suficiente para afetar a matriz $K(\mathbf{G}_\epsilon)$. Belegundu simplificou este problema fazendo $\|\mathbf{G}_\epsilon - \mathbf{G}^0\| = \beta$, e tomando $\beta = 5\%$. Assim, a partir da equação (4.19), o parâmetro ϵ pode ser calculado por

$$\epsilon = \frac{\beta}{\|\mathbf{q}^j\|} \quad (4.21)$$

onde $\|\mathbf{q}^j\|$ é o campo de velocidades normalizado.

Para o cálculo da sensibilidade da função objetivo pode-se empregar o mesmo procedimento demonstrado para a equação (4.20). Assim, para uma função ψ qualquer pode-se escrever:

$$\frac{\partial \psi}{\partial x_i} = \frac{[\psi(\mathbf{G}_\epsilon) - \psi(\mathbf{G}^{old})]}{\epsilon}, \quad i = 1, \dots, n \quad (4.22)$$

O tópico "Análise de Sensibilidade" é de extrema importância no campo da otimização, sendo por este motivo, tema de pesquisas. Os problemas de otimização de forma diferem dos problemas gerais de otimização uma vez que as restrições, sejam elas tensões, frequências naturais ou deformações, não podem ser expressas explicitamente em termos das variáveis de projeto utilizadas. Estas restrições são calculadas neste tra-

balho com o método dos elementos finitos, que é a técnica numérica mais empregada em otimização (ou também o método dos elementos de contorno).

Como a maioria dos algoritmos de otimização é baseado nos gradientes da função e das restrições, a análise de sensibilidade de primeira ordem é bastante utilizada. Apesar do mérito deste ou daquele método ser bastante discutido, quanto ao grau de precisão, eficiência ou custo computacional, o fato é que a obtenção das derivadas, principalmente para um grande número de variáveis, faz com que este tema esteja sempre em evidência^[10–12]. Mais recentemente, alguns pesquisadores se concentraram em algoritmos de ordem zero, isto é, algoritmos que não precisam calcular as derivadas, tornando este campo de pesquisa ainda mais interessante. Para um maior aprofundamento neste assunto recomenda-se consultar as referências^[11, 14, 17].

Implementação Numérica

O programa de otimização de forma foi implementado em Fortran 95, tendo duas sub-rotinas principais: o método das direções viáveis de Zoutendijk e a análise numérica com o método dos elementos finitos para um elemento quadrangular linear. Neste programa, o método dos elementos finitos é utilizado para avaliar as restrições, ou seja, as tensões, elemento a elemento. A avaliação da função, isto é, o cálculo do volume ou do peso da estrutura, não exige que seja executada a análise do MEF, como veremos mais adiante. Fica clara a intensa troca de informações entre estes dois módulos, motivo de preocupação quando se trata de sistemas com grande número de elementos.

Para estes sistemas, vários métodos foram propostos, para diminuir o custo computacional. Pode-se citar, por exemplo, os *solvers* para matrizes esparsas, o uso do método do gradiente reduzido generalizado (GRG), que evita a montagem da matriz de rigidez global, e o método proposto por Vanderplaats, usado por Naquib^[18], onde os gradientes são calculados apenas se houver uma sensível alteração no seu valor, caso isto não ocorra, utilizam-se os gradientes anteriores.

Nesta implementação usou-se os métodos tradicionais para a montagem da matriz rigidez e para o cálculo do gradiente das restrições.

Nas próximas sessões será detalhada esta implementação. O algoritmo básico do

método das direções viáveis também será detalhado, uma vez que é um pouco diferente daqueles apresentados para o cálculo de funções analíticas.

5.1 Algoritmo do aplicativo Shape

Todo processo de otimização de forma pode ser definido em três etapas: representação geométrica, análise estrutural e otimização. A figura 14 mostra o processo de otimização.

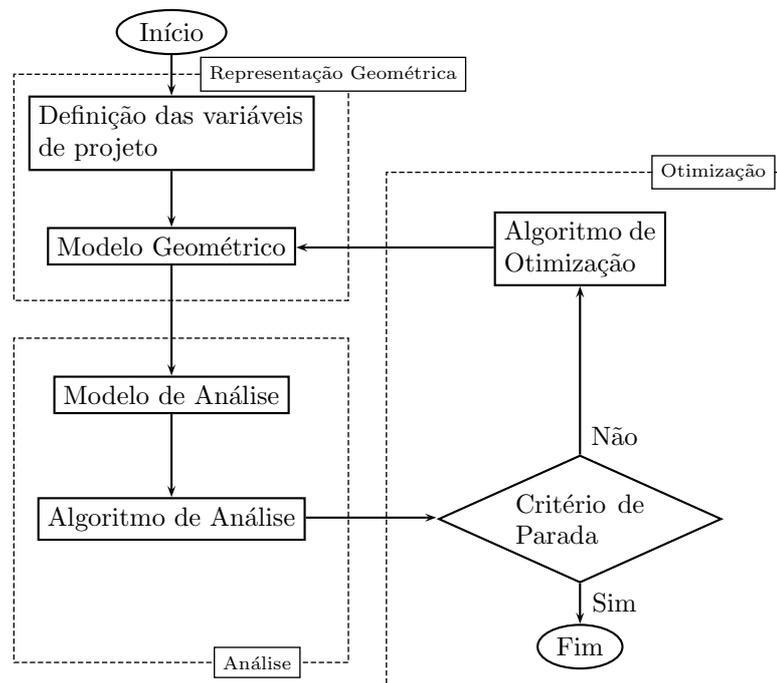


Figura 14: Fluxograma geral de otimização de forma.

5.1.1 Representação geométrica

O módulo de representação geométrica tem por finalidade definir o domínio da estrutura que será otimizada, bem como definir as variáveis de projeto. Normalmente esta representação é feita por meio de uma malha de elementos finitos, que irá definir a estrutura e as suas condições de contorno, os quais, posteriormente, irão alimentar o módulo de análise. Como já visto no capítulo anterior, a definição das variáveis é uma parte muito importante dentro do módulo de representação geométrica. A escolha é feita de tal forma que a mudança na geometria, determinada pelo módulo de otimização,

mantenha a suavidade na malha, ponto a ponto. É importante também que a escolha não aumente demasiadamente o número de variáveis, o que não traria nenhuma vantagem em termos de precisão, e apenas aumentaria o custo computacional. Duas das metodologias mais aceitas para a determinação das variáveis de projeto são as curvas *splines* e as variáveis naturais.

A malha de elementos finitos para problemas bi-dimensionais não constitui um problema, podendo ser escolhido elementos triangulares ou quadrangulares. Quanto à ordem do elemento, não há na literatura nenhuma preferência específica. Nota-se apenas que empregando elementos de ordem superior pode-se diminuir a quantidade dos mesmos, principalmente em geometrias mais complexas.

5.1.2 Análise estrutural

O método dos elementos finitos é a ferramenta de análise estrutural mais utilizada. Para isto, tanto se podem utilizar programas comerciais, quanto programas desenvolvidos *in house*. A escolha entre um e outro depende da maior facilidade de interfaceamento entre os módulos de análise e otimização.

Outra metodologia é o uso de elementos de contorno, que vem se tornando uma boa escolha devido ao fato de reduzir o número de coordenadas nodais ao extremo, mantendo o foco apenas no que mais importa: o contorno.

Na otimização de forma, as tensões são as restrições do problema de PNL. Assim toda vez que o módulo de otimização as solicita, é necessário proceder a uma análise estrutural para que se possam avaliar as restrições e seus gradientes, ficando claro que apenas as restrições que estão ativas são avaliadas. As tensões foram calculadas nos pontos de integração (quatro) e o gradiente das restrições foi calculado por meio de diferenças finitas, equação (4.22).

A função objetivo, que define o volume da estrutura, não necessita da análise dos elementos finitos para ser calculada. Para o seu cálculo basta somente a geometria da malha:

$$f(\mathbf{x}) = \sum_{i=1}^{nel} \det J \times \text{espessura} \quad (5.1)$$

onde $\det J$ é o determinante do Jacobiano, e nel é o número total de elementos. Para a

avaliação do gradiente da função também se utiliza a equação (4.22). Assim, toda vez que o módulo de otimização solicitar o cálculo das restrições e seus gradientes, o MEF será executado.

5.1.3 Algoritmo de otimização

Neste módulo se faz a aplicação de algum algoritmo de programação não-linear, que trocará dados com o módulo de análise, permitindo a solução do problema de otimização.

Os principais algoritmos usados em otimização classificam-se em métodos primais e métodos lagrangeanos ou duais. O método dual trabalha no espaço m -dimensional dos multiplicadores de Lagrange, resolvendo uma seqüência de sub-problemas lineares. Uma variação simplificada deste método é chamada programação linear seqüencial (SLP), um dos métodos mais utilizados em otimização linear. Uma derivação deste método é chamada programação quadrática seqüencial (SQP), onde a função objetivo é quadrática e as restrições são lineares.

Os métodos de busca direta ou primais, por outro lado, trabalham diretamente na região viável do problema, como mostrado no Capítulo 3. No aplicativo SHAPE foi implementado o método das direções viáveis de Zoutendijk. A seguir é mostrada uma iteração típica neste aplicativo:

Primeira chamada: $k = 0$

1. $\mathbf{x}^0 = \mathbf{0}$
2. Defina os limites inferiores e superiores: x_i^I, x_i^S ;
3. Leia o arquivo contendo os campos de velocidade, $\mathbf{q}^i, i = 1, \dots, n$;
4. Normalize os campos de velocidades, tal que o maior componente de $\mathbf{Q} = \|\mathbf{q}^i\|$ seja, por exemplo, 2% da dimensão estrutural;
5. Leia no arquivo do MEF, a malha atual \mathbf{G}^{old}

Avaliação da função (\mathbf{x}^k)

1. Atualize a malha: $\mathbf{G}(\mathbf{x}^k) = \mathbf{G}^{old} + \sum x_i^k \mathbf{q}^i$;
2. Forme a matriz rigidez \mathbf{K} e resolva a equação $K(\mathbf{x})\mathbf{u} = \mathbf{F}$. Avalie a função objetivo e as restrições.

Avaliação dos gradientes (\mathbf{x}^k)

1. Atualize a malha: $\mathbf{G}(\mathbf{x}^k) = \mathbf{G}^{old} + \sum x_i^k \mathbf{q}^i$;
2. Faça $\mathbf{G}^{old} = \mathbf{G}(\mathbf{x}^k)$;
3. Faça $\mathbf{x}^k = \mathbf{0}$;
4. Calcule o gradiente, segundo a equação (4.22).

5.1.4 Critérios de parada

Os critérios de parada são monitorados em duas etapas distintas, uma dentro da sub-rotina da busca da direção e outra pela comparação da variação da função objetivo.

No problema da busca da direção, a convergência é alcançada quando $\beta^* = 0$. Isto implica que o ponto corrente é um ponto KKT. A implementação segue a seguinte relação: se $|\beta^*| < \text{tolerância}$, pare. Se $\beta^* > 0$, a direção é decrescente-viável e o tamanho do incremento (α) deve ser calculado.

Pela comparação dos valores atual e passado da função custo pode-se determinar a parada. No aplicativo foi determinado que esta hipótese deve ser conferida por três vezes consecutivas, para só então terminar o programa.

5.2 Controle de distorção

À medida que a malha vai sendo alterada, ela pode se tornar muito distorcida, o que torna o controle da distorção de suma importância.

Zhang e Belegundu^[19] descrevem uma maneira de controlar a distorção a partir do Jacobiano. Este método avalia um parâmetro de deformação DP e o compara com um valor pré-estabelecido, para evitar a degeneração do elemento quadrangular. Se a distorção de algum elemento atingir o limite, as mudanças na forma são paradas. O processo somente avança se esta restrição for removida, com relocação dos nós, aumento da ordem das funções de forma ou acréscimo no número de elementos.

5.2.1 Parâmetro de distorção DP

Na referência [19] é definido um indicador da qualidade da malha, a partir do Jacobiano para um elemento quadrangular:

$$DP = \frac{4 \det[J]_{\min}}{A} \quad (5.2)$$

onde $\det[J]_{\min}$ refere-se ao menor valor do determinante do Jacobiano avaliado no intervalo $(-1 \leq \xi \leq 1, -1 \leq \eta \leq 1)$ e A é a área do elemento. Para o elemento quadrangular o parâmetro de distorção pode ser escrito como:

$$DP = \frac{4 \min(DJ_i)}{A}, \quad i = 1, \dots, 4 \quad (5.3)$$

onde

$$\begin{aligned} DJ_1 &= \frac{1}{4}(x_{21}y_{41} - x_{41}y_{21}) \\ DJ_2 &= \frac{1}{4}(x_{32}y_{12} - x_{12}y_{32}) \\ DJ_3 &= \frac{1}{4}(x_{43}y_{23} - x_{23}y_{43}) \\ DJ_4 &= \frac{1}{4}(x_{14}y_{34} - x_{34}y_{14}) \end{aligned} \quad (5.4)$$

O parâmetro DP traz embutido algumas propriedades interessantes:

1. É invariante com respeito à rotação, à translação e ao fator de escala do sistema de coordenadas.
2. $DP > 0$ se, e somente se, o elemento quadrangular for *convexo*, figura 15(a).
3. $DP = 0$ se, e somente se, o elemento for degenerado (*triângulo*), figura 15(b).
4. $DP < 0$ se, e somente se, o elemento quadrangular for *côncavo*, figura 15(c).
5. $DP = 1$ quando o elemento é um *paralelogramo*, figura 15(d).

5.3 Implementação do controle de distorção

Já foi demonstrado em [19] que o momento mais apropriado para se controlar a qualidade da malha é durante a determinação da direção, no módulo de otimização, não sendo o controle de distorção tratado como uma restrição na PNL. Nesta etapa, as

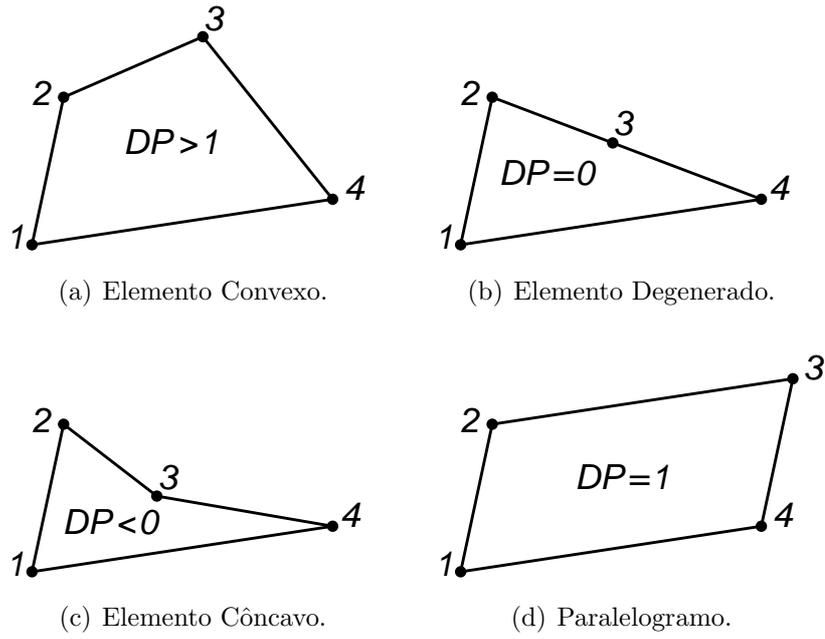


Figura 15: Parâmetro DP e as formas dos elementos.

variáveis são controladas pelo parâmetro de tamanho de incremento α ao longo de uma direção conhecida \mathbf{d} . Assim:

$$\mathbf{x} = \alpha \mathbf{d} \quad (5.5)$$

A necessidade de que $\mathbf{x} = \mathbf{0}$, no início de cada iteração, deve-se ao fato de que $\mathbf{G} = \mathbf{G}^{old}$. As equações (4.3) e (5.2) definem a seguinte relação:

$$DP[\mathbf{G}(\mathbf{x})] = DP[\mathbf{G}(\alpha \mathbf{d})] \equiv \widetilde{DP}(\alpha) \quad (5.6)$$

e fazendo DP_l o limite inferior de \widetilde{DP} , então:

$$\widetilde{DP}(\alpha) = DP_l \quad (5.7)$$

o que fornece o máximo tamanho do incremento do intervalo para o elemento em questão, α_{DP}^e . A matriz do Jacobiano pode ser escrita como:

$$J = \begin{bmatrix} J_{11}^y + \alpha J_{11}^q & J_{12}^y + \alpha J_{12}^q \\ J_{21}^y + \alpha J_{21}^q & J_{22}^y + \alpha J_{22}^q \end{bmatrix} \quad (5.8)$$

onde

$$J_{11}^y = \sum_{i=1}^4 \frac{\partial N_j}{\partial \xi} y_j^{old}, \quad J_{11}^q = \sum_{i=1}^{np} \sum_{IP}^4 \mathbf{d}_i \frac{\partial N_j}{\partial \xi} q_{yj}^i \quad (5.9)$$

com N_j , y_j^{old} e q_{yj}^i sendo, respectivamente, as funções de forma, a coordenada nodal atual

e o i -ésimo campo de velocidade. Das equações (5.7) e (5.8), e da imposição que o determinante seja nulo, pode-se escrever:

$$a\alpha^2 + b\alpha + c = 0 \quad (5.10)$$

onde a , b e c são constantes conhecidas. A solução da equação anterior fornece α_{DP}^e .

Então, pode-se definir:

$$\alpha_{DP} = \min_e \alpha_{DP}^e \quad (5.11)$$

que é o limite no tamanho do incremento, relacionado à distorção permissível. Com isto, durante o processo de otimização, o melhor tamanho de incremento é obtido a partir de:

$$\alpha = \min_e[\alpha_{DP}, \alpha_{PNL}] \quad (5.12)$$

onde α_{PNL} é o tamanho do incremento definido durante a PNL.

CAPÍTULO 6

Resultados

Após a implementação numérica, foram modelados alguns exemplos clássicos de otimização, com a finalidade de comparar os resultados obtidos através do programa SHAPE. Todos estes modelos são encontrados na literatura de otimização de forma, tornando-se assim um *benchmark* para avaliação de um novo programa. Cada modelo estudado tem suas particularidades, o que torna a tarefa de otimizar uma estrutura ou peça, algo bastante desafiador.

Como, a princípio, não existem respostas prontas, o analista deve estudar bem o problema, definir o tamanho da malha inicial, definir quais serão as variáveis naturais utilizadas (a geração dos campos de velocidade), definir as tolerâncias dos critérios de parada e ativação das restrições e executar o aplicativo pela primeira vez, observando a resposta. A escolha das variáveis naturais é de suma importância, pois afetarão de forma significativa os resultados.

Feita esta primeira análise, deve-se decidir pela manutenção das variáveis de projeto ou pela escolha de um novo padrão. É neste ponto que ficará claro a influência que cada escolha terá nos resultados. De uma maneira geral, pode-se dizer que os campos de velocidades que são gerados por campos de pressão, além de terem seu tamanho reduzido, permitem uma suavidade maior do contorno, quando comparado à aplicação de cargas unitárias nodais. Outra característica importante é o fato que o lado do elemento que

foi submetido à pressão sofre um “deslocamento de corpo rígido”, o que se presta àqueles casos onde se espera uma grande deformação ou modificação da estrutura.

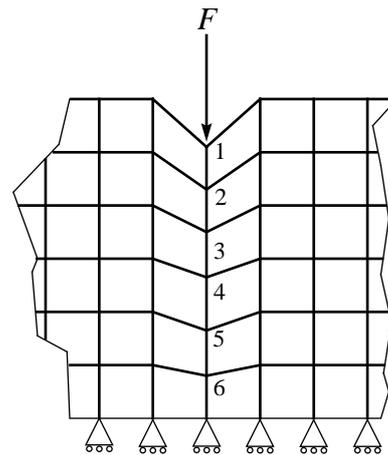
Outra observação pertinente é o fato de que se deve procurar que a carga aplicada ao nó, ou a pressão exercida, cause uma deformação que possa extrapolar pelo menos quatro ou cinco nós abaixo daquele que foi submetido ao carregamento. Isto significa que ao gerar o campo de velocidades, seja possível experimentar relações constitutivas diferentes daquelas da peça original, com a única finalidade de se obter um campo de deslocamento nodal, abaixo do nó ou do elemento carregado, que perturbe de fato a malha.

Isto é facilmente verificado ao fazer manualmente um campo de velocidade, onde por exemplo, os deslocamentos dos seis nós, que estejam na mesma linha de ação da força, tenham a seguinte distribuição, Tabela 2

Tabela 2: Aplicação manual dos deslocamentos nodais.

Número do Nó	Deslocamento
1	$6/6 = 1$
2	$5/6$
3	$4/6 = 2/3$
4	$3/6 = 1/2$
5	$2/6 = 1/3$
6	$1/6$

Fonte: Autor.



O modelo da placa com furo, figura 9, página 28, foi analisado desta forma. O fato é que, para aplicações em geometrias complexas ou para um grande número de pontos de aplicação do carregamento, isto gera um trabalho razoável. Este exemplo foi feito desta forma para que se pudesse avaliar o comportamento da otimização quando um campo de velocidades “ideal” é obtido. Os comentários sobre isto serão feitos junto à descrição do modelo, veja seção 6.3. Na maioria das vezes, porém, as variáveis naturais são obtidas de forma automática, com ajuda de algum software, neste caso o Algor®. Deve ser lembrado que os campos de velocidades são normalizados com relação à dimensão da estrutura.

Para os exemplos, onde é mostrada a deformação causada pelos campos de velocidade, foram retiradas as escalas numéricas. Uma vez que os campos são normalizados,

não faz sentido apresentar os valores absolutos de deformação, contudo a variação das cores indica o grau de deformação, sendo a cor vermelha indicativo de maior deformação e o azul escuro de deformação nula.

6.1 Pórtico

Neste exemplo, citado em Belegundu e Chandrupatla^[8], é realizada a otimização de forma de um pórtico, tendo como restrição a tensão máxima $\sigma_{lim} = 80\text{N/mm}^2$ (tensão plana). É um exemplo bastante simples, que foi reproduzido aqui por ter sido o primeiro resolvido durante a execução deste trabalho. Esta estrutura se beneficia da simetria para reduzir o número de elementos. A figura 16 mostra as dimensões e o carregamento ao qual o pórtico está sujeito.

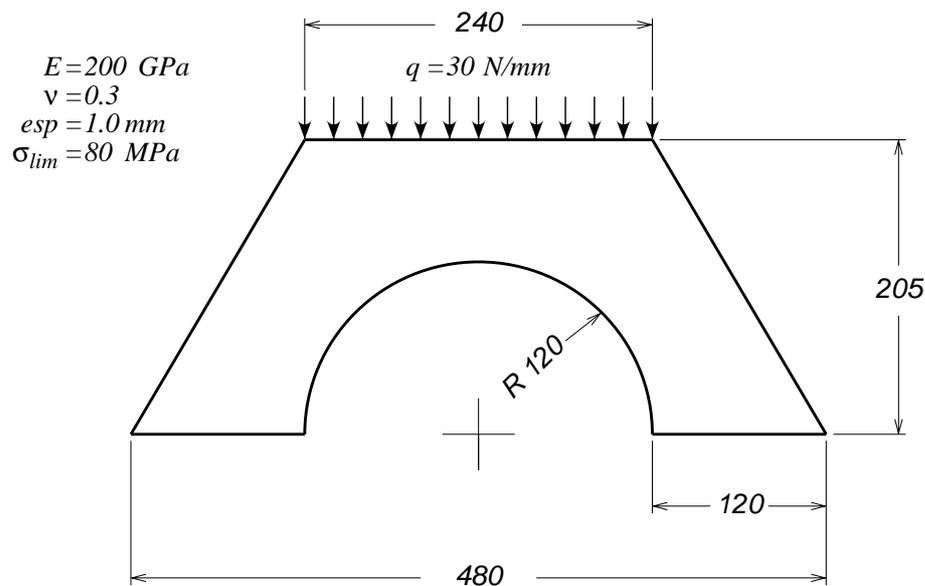


Figura 16: Dimensões do pórtico (dimensões em mm).

O objetivo desta otimização é minimizar o peso da estrutura mantendo o contorno externo inalterado, sendo que apenas ao semicírculo é permitido variar. A malha grosseira, com a qual foi iniciado o processo, deve-se ao fato dela permitir uma grande deformação dos nós de controle, e por conseqüência, também dos nós vizinhos, sem causar distorção excessiva nos elementos, o que impediria o programa de prosseguir. Os diversos modelos que foram otimizados seguem o mesmo padrão. Apenas quando já houve uma grande deformação, isto é quando a convergência para aquela malha é atingida, ou ocorre a distorção, é que se faz um *remesh*, possibilitando um melhor ajuste do contorno. A

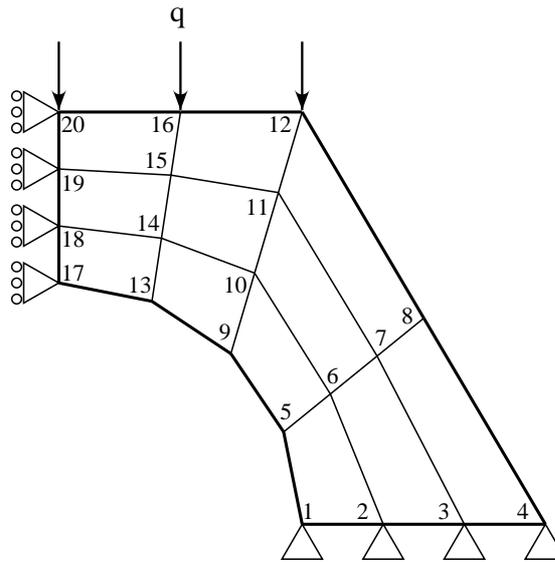


Figura 17: Geometria e condições de contorno.

figura 17 mostra a geometria utilizada no problema de otimização.

A seguir será apresentada a geração dos campos de velocidade. Os cinco nós internos (1, 5, 9, 13, 17) foram definidos como nós de controle, ou variáveis de projeto ($ndv = 5$). Foi aplicado, para cada um dos nós, um deslocamento unitário responsável por gerar o campo de deformações q^i , $i = 1, \dots, ndv$. É importante notar que o deslocamento unitário é dado ao elemento mais externo. Os demais são perturbados seguindo uma dada razão, como a apresentada na Tabela 2.

Os valores numéricos para os campos de velocidade podem ser vistos na tabela 3. A figura 18 mostra como fica a malha perturbada pelos campos de velocidade mostrados na tabela 3.

A forma final otimizada para o pórtico pode ser vista na figura 19. As linhas tracejadas definem a geometria inicial.

O gráfico da figura 20 mostra a redução de volume após 21 iterações. A redução foi de aproximadamente 20% a partir da malha inicial. Este valor pode sofrer variações dependendo do limite de tensão estabelecido (σ_{lim}) como restrição, do campo de velocidades gerado e de outros parâmetros do aplicativo.

Tabela 3: Campos de velocidades para o pórtico.

Campos de Velocidade										
Cpo	1		2		3		4		5	
Nó	x	y	x	y	x	y	x	y	x	y
1	1.000	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0
5	0	0	0.725	0.688	0	0	0	0	0	0
6	0	0	0.483	0.459	0	0	0	0	0	0
7	0	0	0.242	0.230	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0.259	0.966	0	0	0	0
10	0	0	0	0	0.173	0.644	0	0	0	0
11	0	0	0	0	0.086	0.322	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0.134	0.991	0	0
14	0	0	0	0	0	0	0.089	0.661	0	0
15	0	0	0	0	0	0	0.045	0.330	0	0
16	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	1.000
18	0	0	0	0	0	0	0	0	0	0.667
19	0	0	0	0	0	0	0	0	0	0.333
20	0	0	0	0	0	0	0	0	0	0

Fonte: Autor.

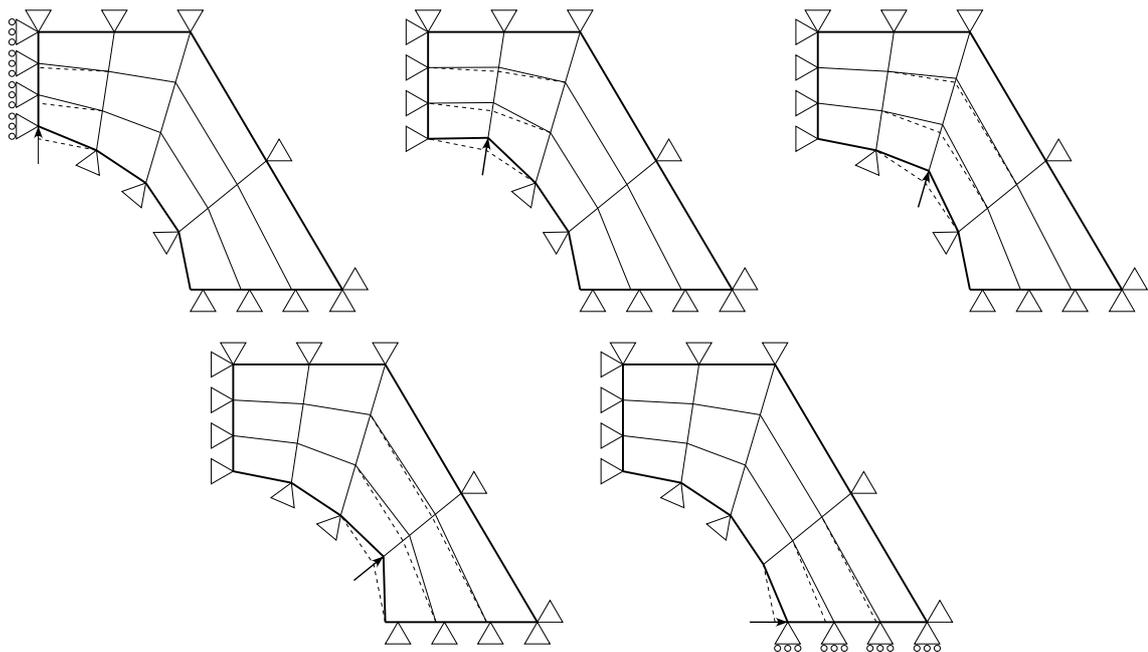


Figura 18: Campos de velocidades.

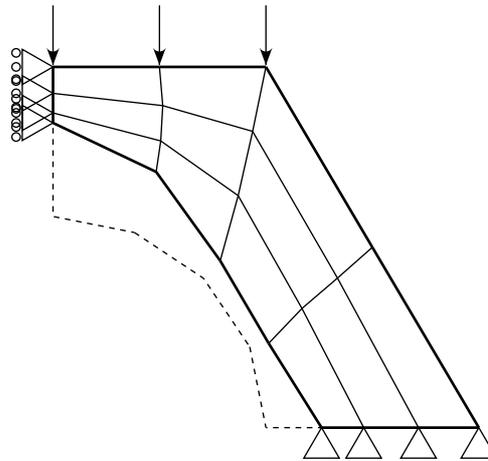


Figura 19: Forma final para o pórtico.

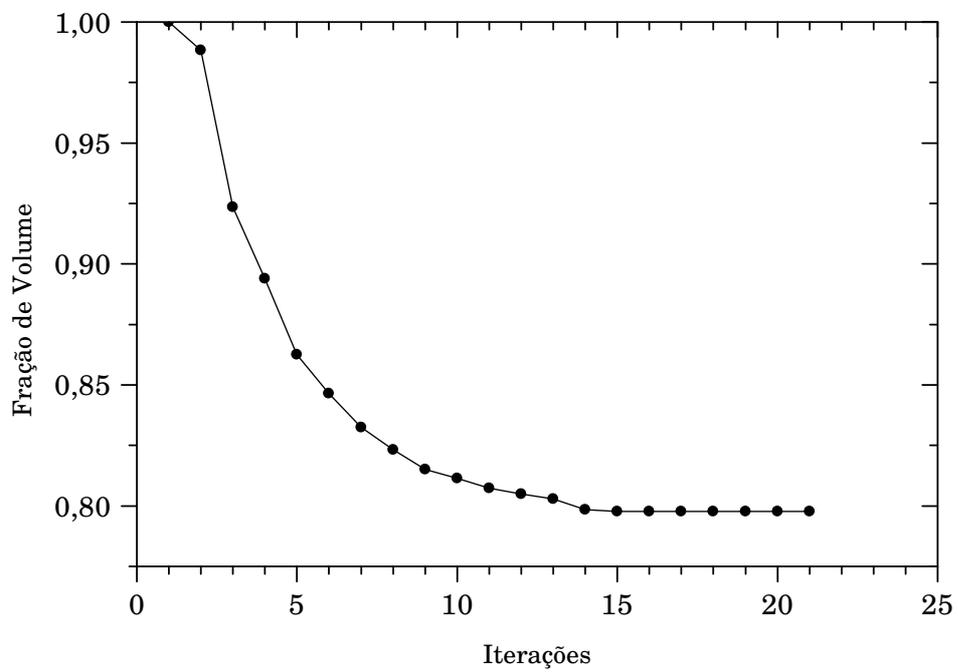


Figura 20: Redução de volume após 21 iterações.

6.2 Alavanca

Este exemplo pode ser encontrado nas referências 11, 12, 17 e em outros trabalhos, com algumas pequenas modificações. Esta alavanca é solicitada por duas forças, uma orientada ao longo do eixo x , no valor de 300N, e outra de 500N, aplicada na direção y , que provocarão tensões de flexão e tração, figura 21.

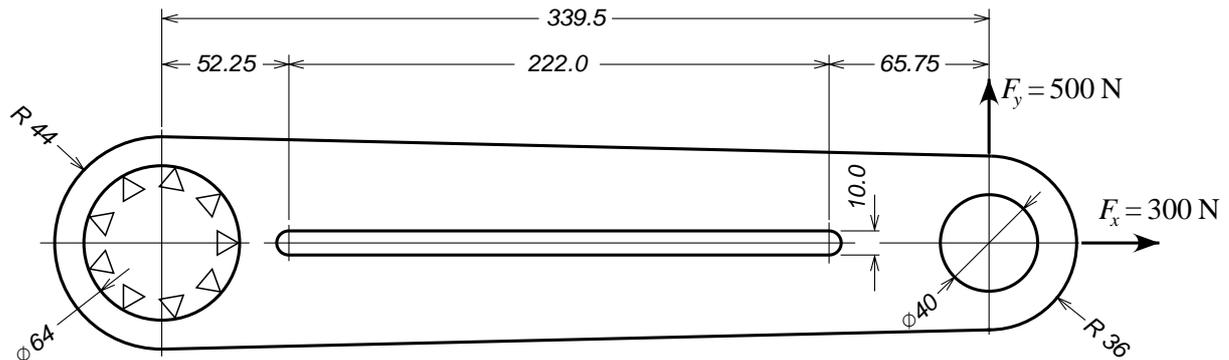


Figura 21: Dimensões e condições de contorno (dimensões em mm).

Foi imposto que a região interna aos furos e os arcos externos concêntricos aos furos, assinalados com linhas mais escuras na figura 22, não serão modificados em sua forma. O módulo de elasticidade E usado foi 210GPa e o coeficiente de Poisson ν igual a 0,3, indicando propriedades de um aço comum, com espessura unitária. O limite de tensão imposto foi de $\sigma_{\text{lim}} = 600\text{N/mm}^2$. A figura 23 mostra a malha usada inicialmente.

Como solução há duas possibilidades: fazer com que o lado externo da peça seja deformado, ou permitir uma variação da forma a partir do lado interno, juntamente com a parte externa. Esta última opção é a de mais difícil obtenção. Isto se deve ao fato que há uma grande alteração na forma, enquanto que o volume sofre uma pequena diminuição.

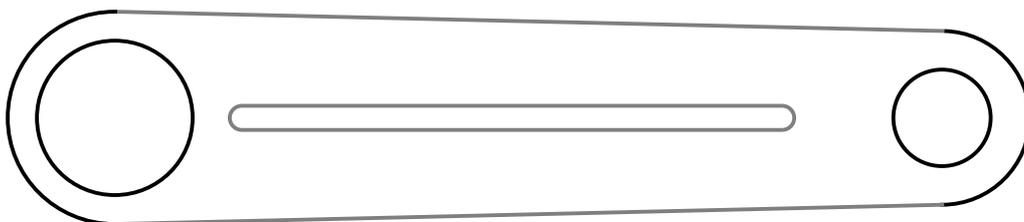


Figura 22: Partes que não sofrerão modificação em sua forma.

Para a solução deste problema foi criado um campo de velocidades interno e outro externo e, em vez da aplicação de cargas unitárias, optou-se pela aplicação de campos de pressão. O problema convergiu para a forma intermediária e foi necessário fazer um *remesh*, devido à qualidade da malha final. Neste *remesh* optou-se por aumentar o número de elementos, com a finalidade de se conseguir uma forma final tão suave quanto possível. Adicionalmente, também foram aplicados externamente campos de pressão.

O resultado obtido pode ser visto na figura 24. No total foram 24 iterações, sendo nove com a malha inicial e 15 após *remesh*. Com isto foi obtida uma redução de 50% no volume de material.

A falta de simetria entre os dois braços deve-se ao fato de que as cargas aplicadas causam tensões de flexão e de tração. Pode-se observar, também, as regiões que foram impedidas de se modificarem (os furos e os arcos externos).

Na figura 25 observa-se que após o *remesh* a remoção de material foi acentuada apenas na primeira iteração, convergindo depois lentamente. O salto observado no gráfico deve-se à nova fronteira interna redesenhada com curvas *spline*, o que acabou acrescentando material, e ao aumento na tensão limite, que fez com que em apenas uma iteração quase se atingisse a forma final. Uma vez que o limite de tensão inicial gerou uma deter-

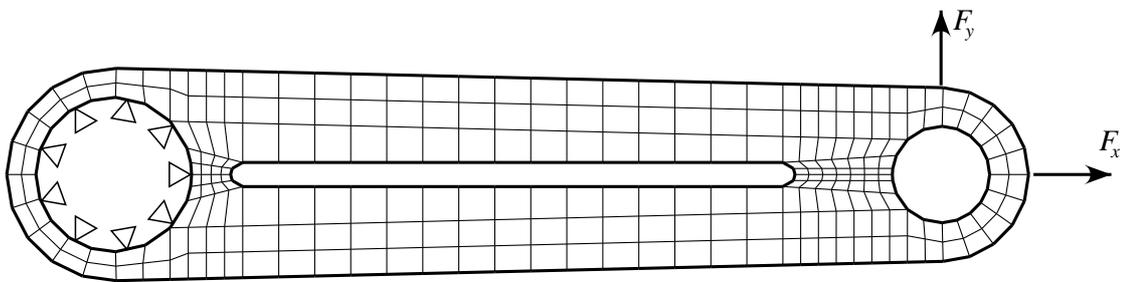


Figura 23: Malha inicial com condições de contorno.

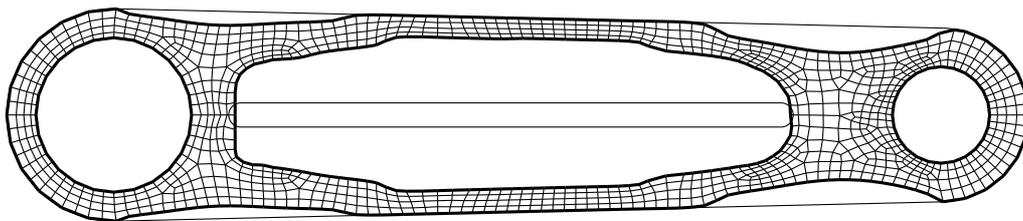


Figura 24: Forma final da alavanca comparada com a forma inicial.

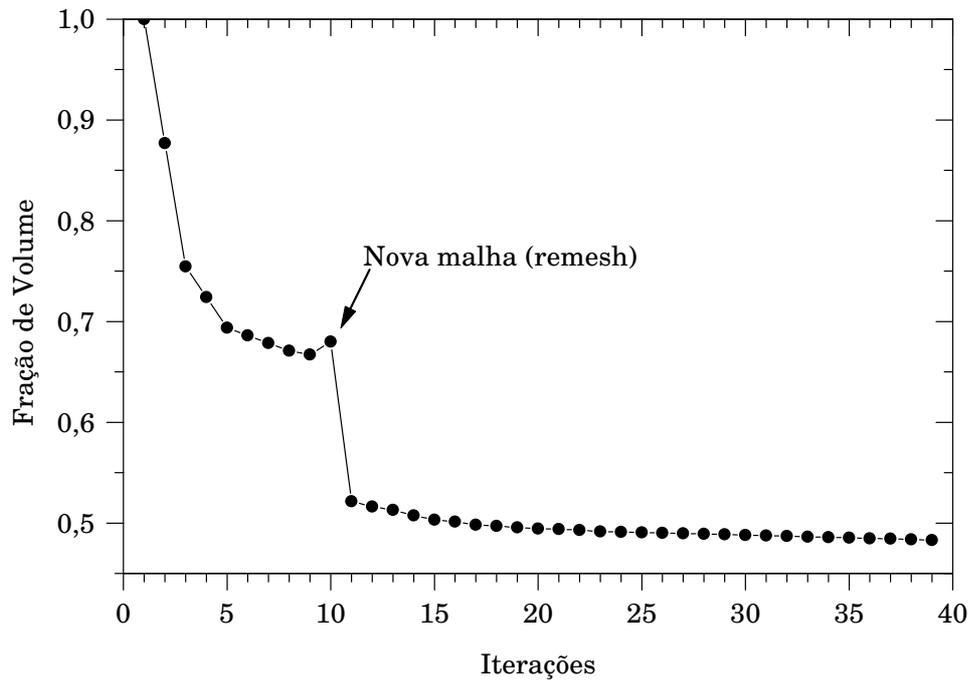


Figura 25: Redução no volume após 39 iterações (com *remesh*).

minada geometria, no passo seguinte (*remesh*) é necessário aumentar este valor, para que a mudança de forma seja mais acentuada. Foi usado um valor de $\sigma_{\text{lim}} = 800\text{N/mm}^2$. A figura 26 mostra os valores de tensão na peça otimizada.

Faz-se necessário esclarecer que o resultado anterior, foi obtido com uma tensão limite de $\sigma_{\text{lim}} = 800\text{N/mm}^2$, e a máxima tensão que o gráfico indica é menor devido ao cálculo da tensão ser distinto nos dois programas. O programa de análise de elementos finitos Algor[®] interpola o resultado nos pontos nodais, enquanto que no aplicativo o

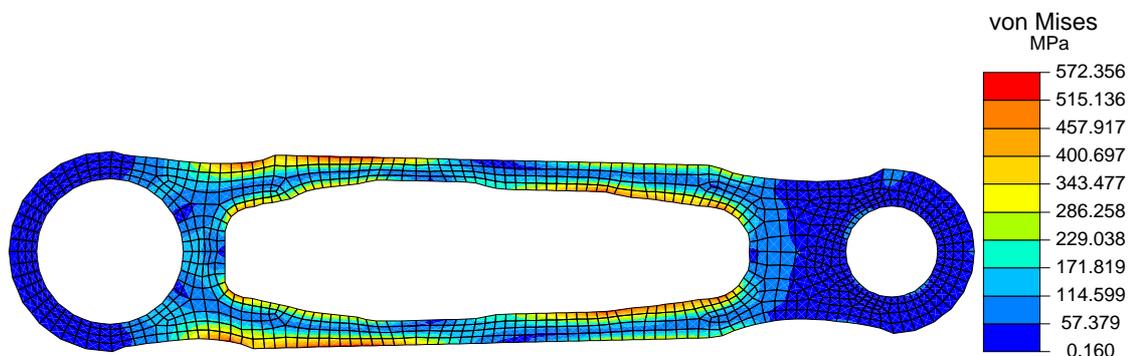


Figura 26: Tensões equivalentes de von Mises.

valor da tensão usada é o valor calculado para os pontos de integração de Gauss. O motivo desta opção deve-se ao fato de que as tensões calculadas nos pontos de integração apresentam um valor mais preciso, enquanto que a extrapolação para os pontos nodais resulta na perda de informação, somando-se ainda o fato que as tensões nos contornos podem apresentar um valor bastante distinto, neste caso em torno de 200MPa.

6.3 Placa

Outro problema clássico é uma chapa com furo, figura 27, carregada bi-axialmente, onde se deseja que apenas a geometria interna (furo) sofra modificação. Devido à simetria da peça, é necessário modelar apenas um quarto da chapa. O carregamento de pressão aplicado ao modelo foi $p_1 = 1\text{MPa}$ na face vertical e de $p_2 = 0,65\text{MPa}$ na face horizontal, a espessura do modelo é $0,1\text{mm}$, o módulo de elasticidade $E = 200\text{GPa}$ e $\nu = 0,3$. O valor limite para a tensão foi definido inicialmente como $\sigma_{\text{lim}} = 3\text{N/mm}^2$. A figura 28 mostra a distribuição de tensões na peça, para a geometria inicial do problema.

Nesta primeira etapa os campos de velocidades foram criados manualmente, com cargas unitárias, sempre normais à superfície e perturbando seis nós na direção radial, a partir do nó de aplicação da carga. Foram necessários 17 campos de velocidades, um para

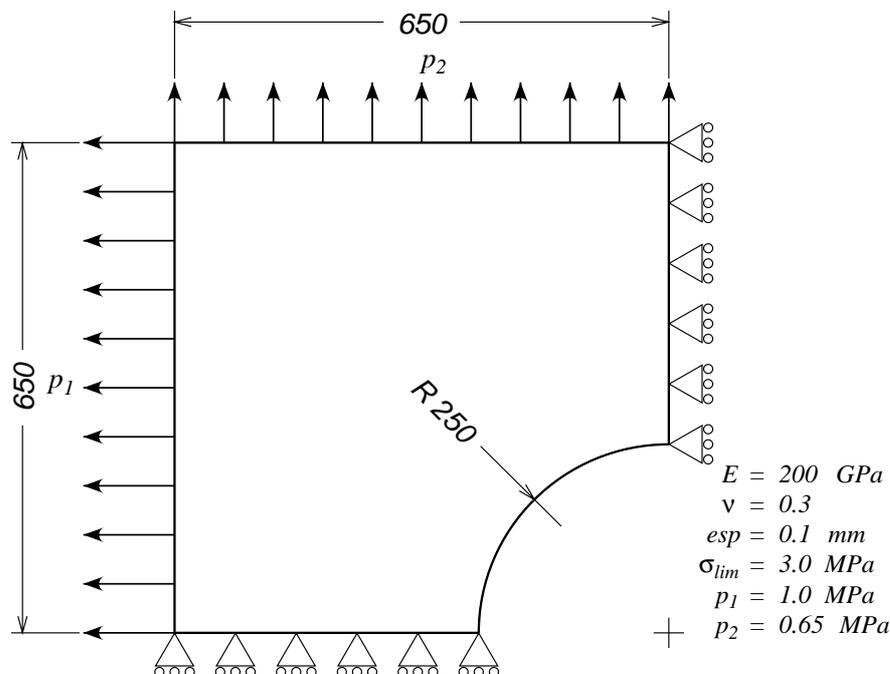


Figura 27: Dimensões e condições de contorno para chapa com furo (dimensões em mm).

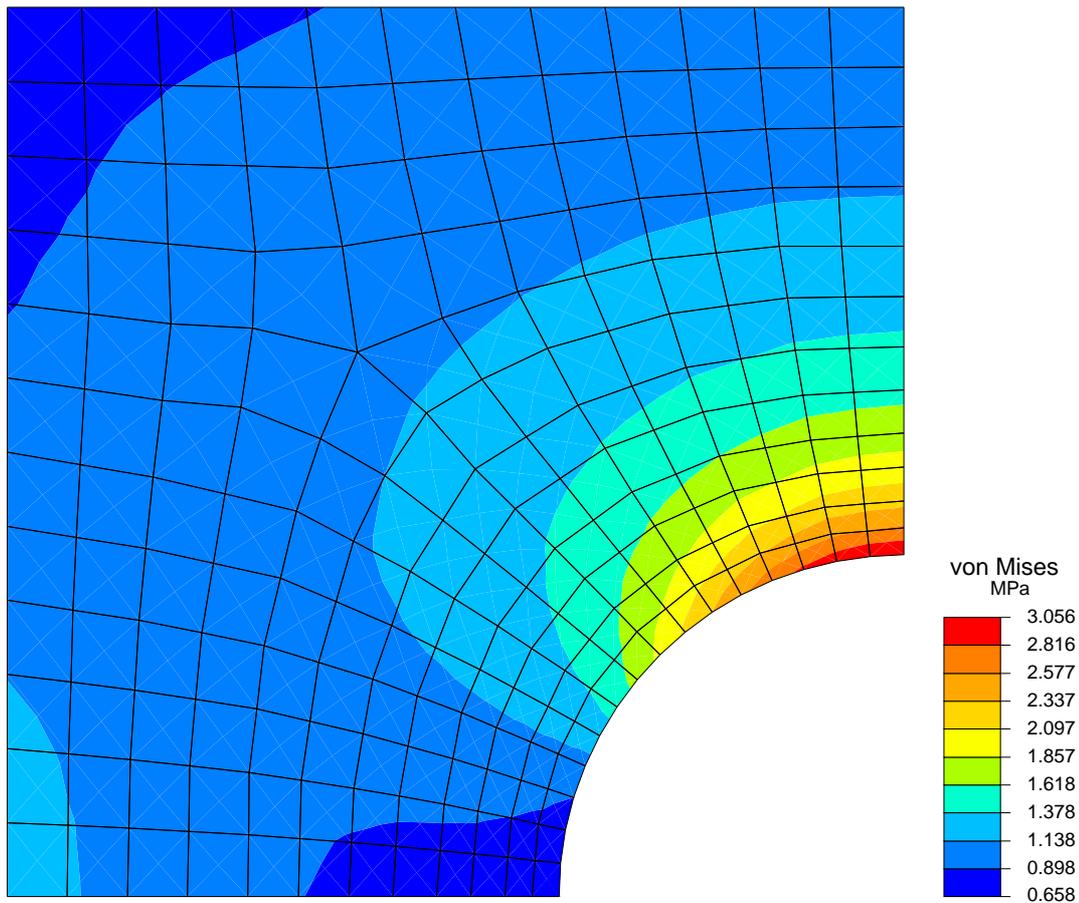


Figura 28: Condições iniciais de tensões.

cada nó interno. Apesar do trabalho manual envolvido nesta operação, e particularmente para esta geometria (a malha também foi criada manualmente), o campo permanece atualizado ao longo das iterações, pois cada carga aplicada somente perturba os nós que estão na direção radial. Isto assegura que, mesmo durante a otimização, o campo de velocidades permaneça orientado.

A figura 29 mostra a deformação nos elementos próximos à extremidade da chapa. Apesar da grande distorção localizada, o problema convergiu naturalmente, parando apenas após terem sido alcançados os critérios estabelecidos. Isto acontece porque a área é diferente de zero e geometricamente o elemento quadrangular não sofreu degeneração em sua forma, figura 29(b). Durante esta primeira etapa da otimização, isto é, a primeira convergência, foram necessárias 68 iterações. O volume inicial, que era 149365mm^3 , reduziu-se para 127653mm^3 , ou seja 14,5%.

Para a geometria obtida na primeira convergência, figura 29(a), foi construída

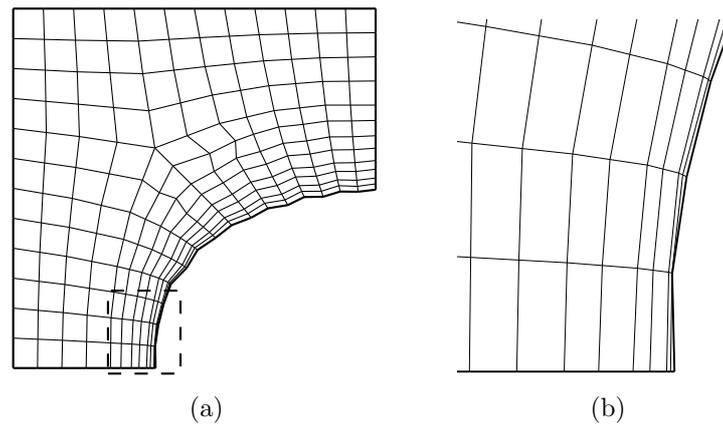


Figura 29: Aspecto da malha após a primeira convergência, e ampliação da região com alto grau de distorção.

uma nova malha, a partir de uma aproximação com *spline*, para amenizar o aspecto serrilhado. Também foi aumentado o número de elementos e construído um novo campo de velocidades aplicando cargas unitárias aos nós que pertencem ao contorno interno. Estes campos de velocidade mantiveram a mesma direção da primeira etapa: foram radialmente orientados. O resultado final, tanto da geometria como das tensões, pode ser visto na figura 30.

Como o valor de tensão ainda não foi atingido, optou-se por outra estratégia: campos de pressão.

Os campos de pressão foram aplicados a cada três elementos, percorrendo todo o perímetro do furo, e mais um campo foi criado considerando todo o contorno interno. A metodologia para gerar estes campos de velocidades foi a seguinte:

- Partindo de uma das extremidades do furo, o primeiro campo definido causou a deformação a partir da pressão aplicada apenas nestes três elementos, com restrição de deslocamento em apenas um eixo, figura 35(a).
- A partir deste, os próximos campos foram obtidos sempre mantendo o anterior, causando uma deformação mais acentuada no interior da malha e permitindo que entre um campo e outro houvesse continuidade entre as regiões deformadas, figuras 35(b) até 35(j).
- Para a última série de campos, repete-se o mesmo procedimento usado na geração do primeiro, figura 35(k).

- E por último, é criado um campo de pressão em todo o contorno, o qual permite um deslocamento total de todos os nós, sem causar deformação localizada, figura 35(1).

O volume, que inicialmente era 128922mm^3 , reduziu-se para 98149mm^3 , um valor bastante abaixo dos encontrados na literatura.

A redução total do volume foi de aproximadamente 35%, figura 31, para o limite de tensão imposto, figura 32.

Parente Jr.^[20] obteve resultados semelhantes e os comparou com outros autores. Além disso, também comparou o desempenho entre dois tipos de algoritmos: programação seqüencial recursiva (SQP) e o método dos pontos interiores (PI). A Tabela 4 relaciona os valores encontrados por ele e os obtidos neste trabalho.

A principal motivação para se explorar a aplicação de campos de pressão foi o aspecto serrilhado da malha após a otimização ao se utilizar campos unitários. Além

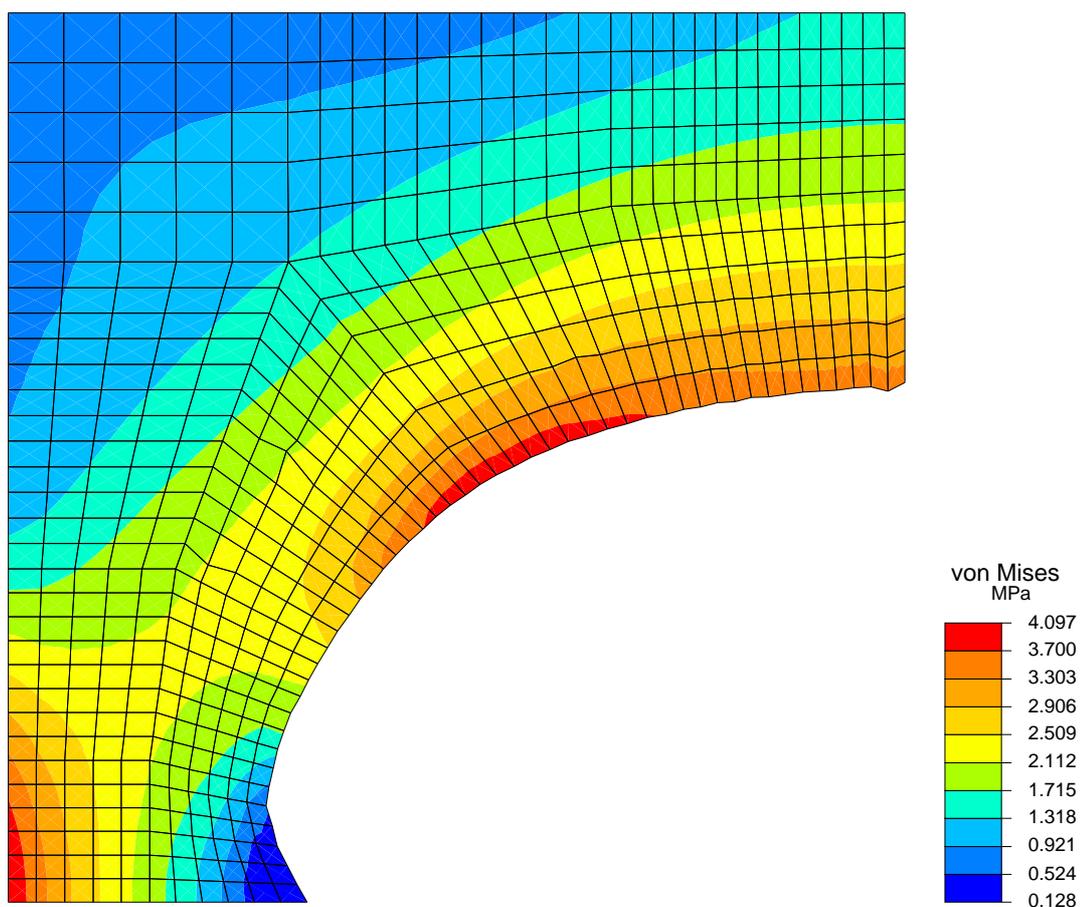


Figura 30: Geometria final, com distribuição de tensões.

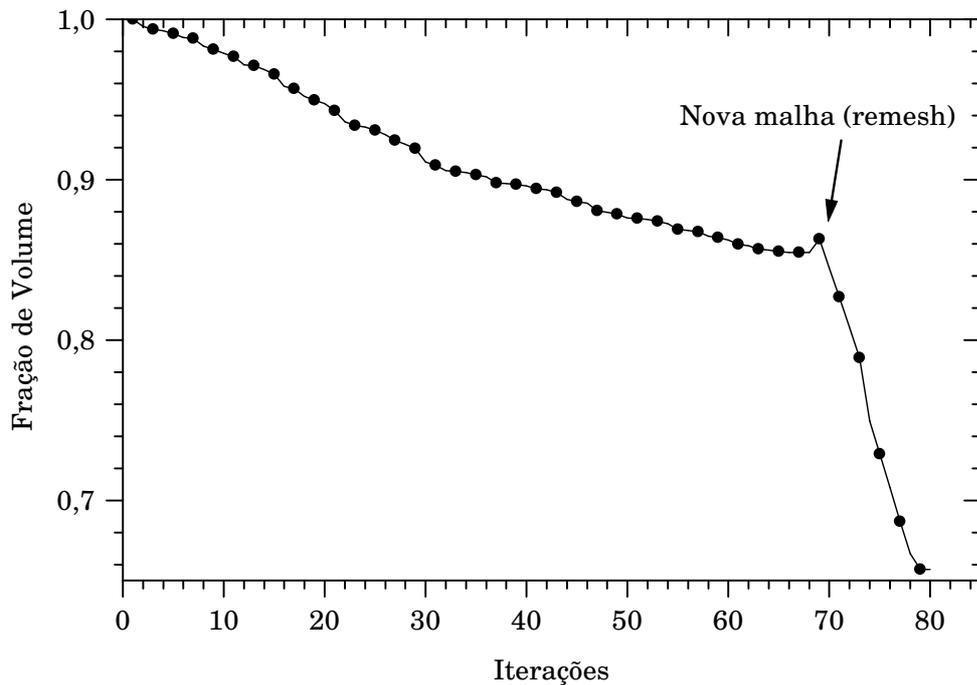


Figura 31: Redução no volume para o exemplo da placa com furo.

disso, alguns nós não se deslocaram da mesma forma que os demais, o que fez com que o aplicativo se encerrasse (ou por ter convergido ou por ter alcançado o limite de iterações). A figura 33 mostra este efeito.

Por outro lado, aplicar indistintamente campos de pressão pode também não resolver o problema. Se não houver continuidade entre os campos gerados, ou seja, cada campo perturba apenas a região influenciada pela pressão, tem-se um resultado como o obtido na figura 34.

Quando o campo de pressão é criado, seguindo as instruções dadas anteriormente,

Tabela 4: Comparação entre os algoritmos.

Autor	Algoritmo	Iterações	Vol. (mm ³)	σ_{lim} (MPa)
Parente Jr. (2000)	SQP	8	111797	7.000
Parente Jr. (2000)	Ptos. Interiores	16	111754	6.998
Sienz (1994)	SQP	15	112132	6.970
Özakça	n.d.	n.d.	114000	n.d.
Machnievszc (2003)	Zoutendijk	77	98149	6.904

Fonte: Autor.

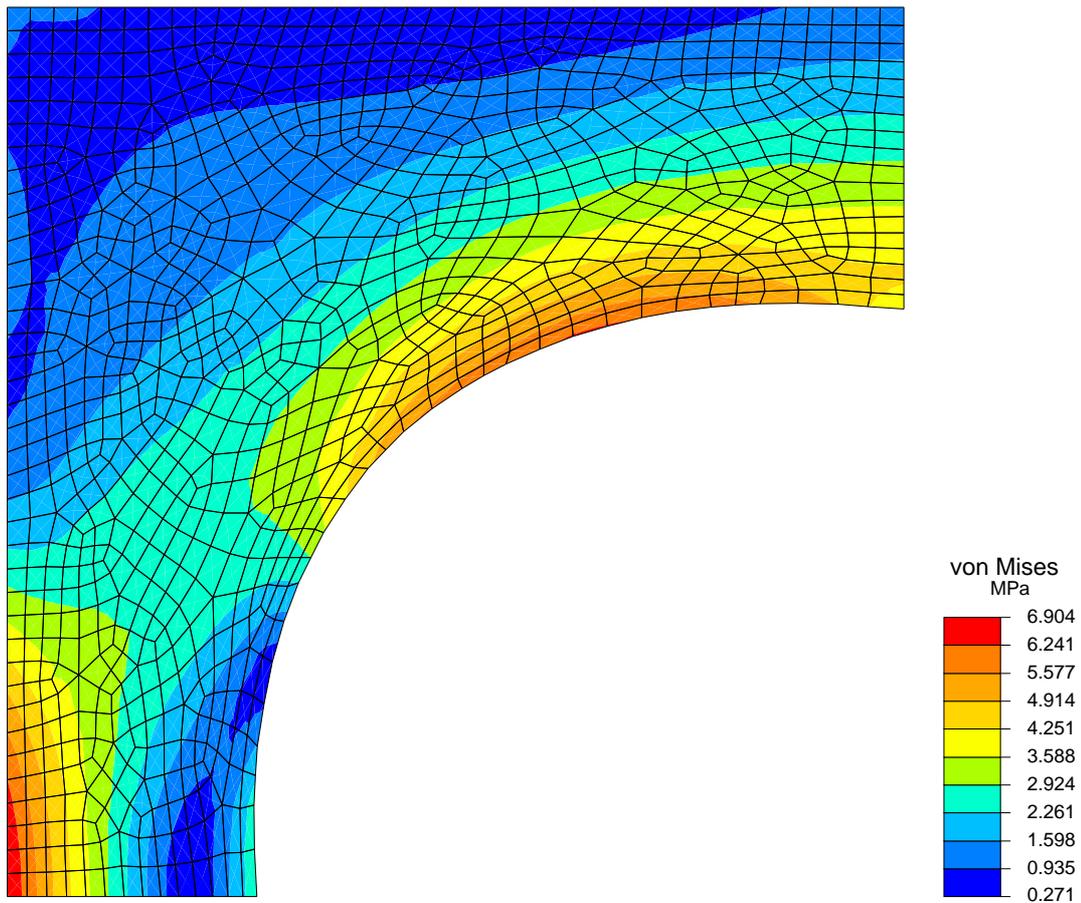


Figura 32: Geometria final obtida para o limite de tensão $\sigma_{\text{lim}} = 7.0$ MPa.

o resultado é similar ao do uso de *spline*, ou seja, existe continuidade entre as deformações causadas pelos campos adjacentes. Isto tem como resultado uma deformação da malha mais controlada e livre de discontinuidades, as quais indicam erroneamente que a tensão limite foi alcançada. Na realidade, segundo o método de Zoutendijk, isto informa ao programa que aquela restrição está ativa, o que irá gerar uma direção sabidamente errada. No caso de haver continuidade, a restrição se ativará conforme a forma for se alterando, e não devida a uma concentração de tensão causada por uma malha com singularidades.

Para uma melhor compreensão deste procedimento será ilustrado o aspecto de todos os campos de pressão gerados para este exemplo, figura 35.

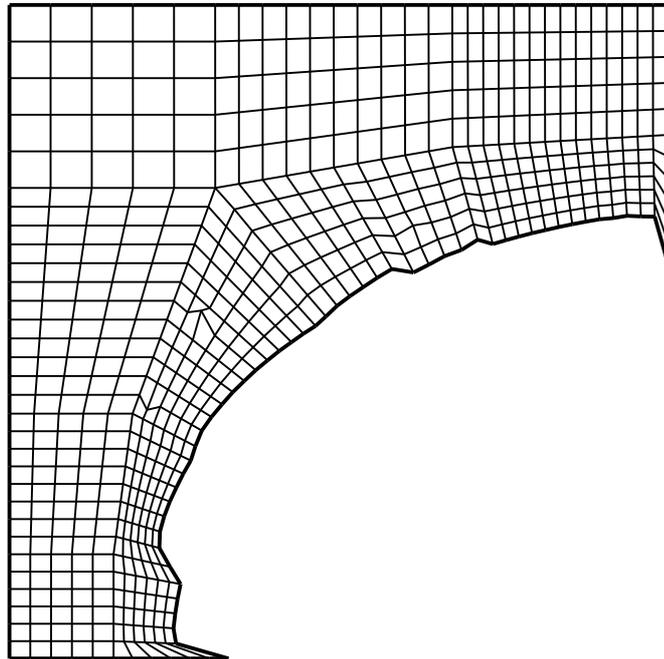


Figura 33: Resultado obtido ao se aplicar somente cargas unitárias.

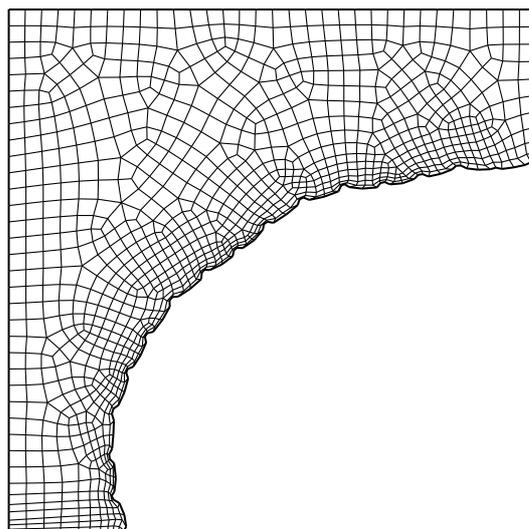
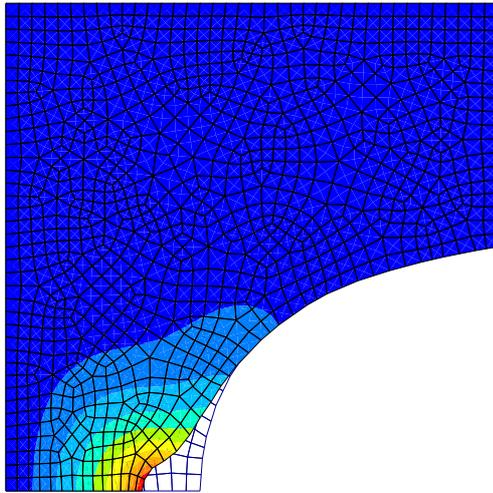
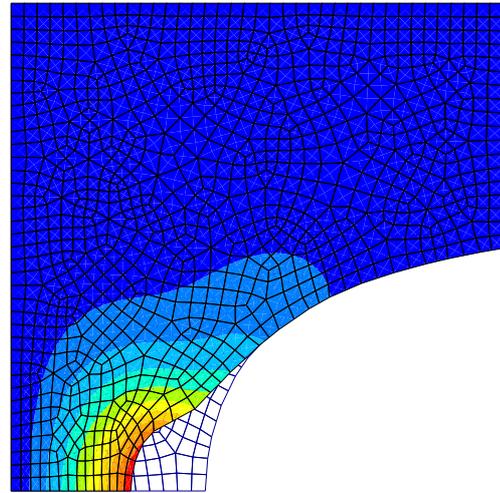


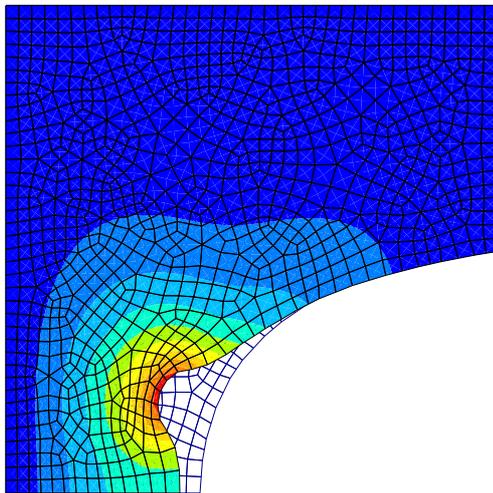
Figura 34: Exemplo do mau uso de campos de velocidades.



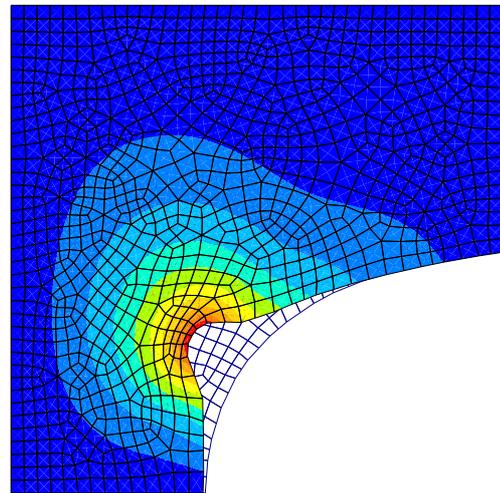
(a) Campo de velocidade n°1



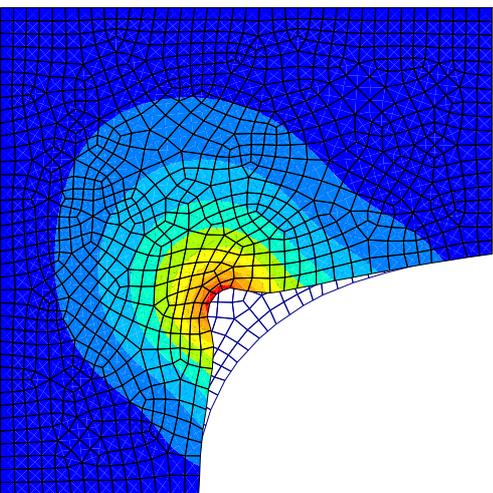
(b) Campo de velocidade n°2



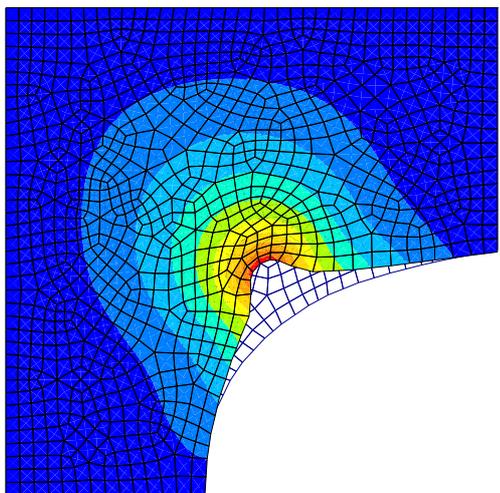
(c) Campo de velocidade n°3



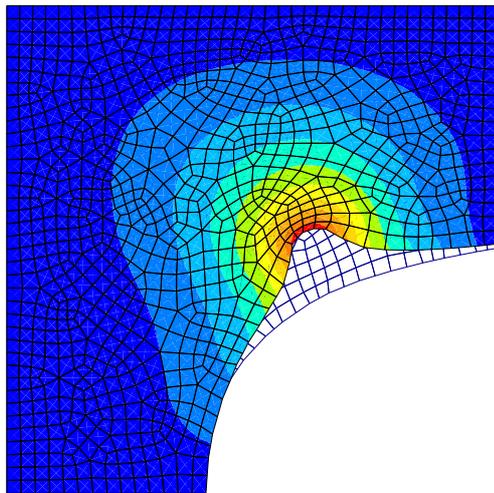
(d) Campo de velocidade n°4



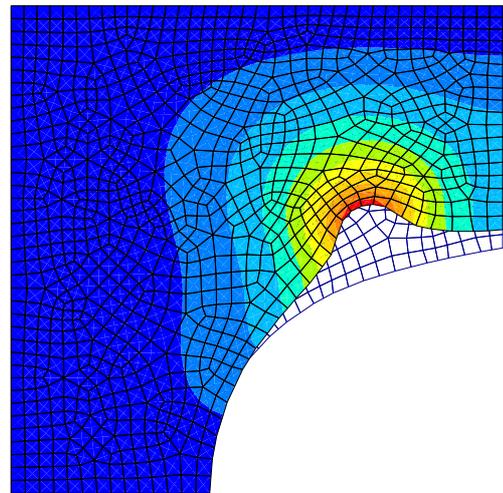
(e) Campo de velocidade n°5



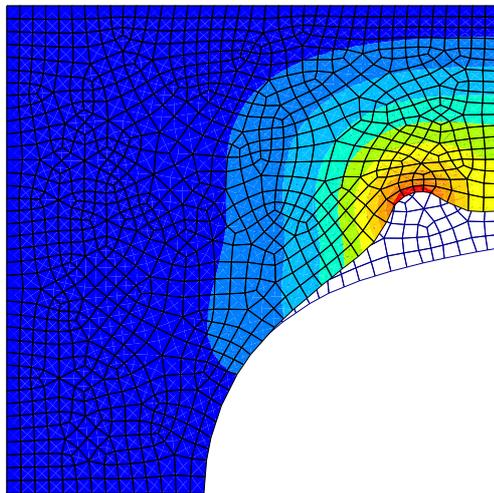
(f) Campo de velocidade n°6



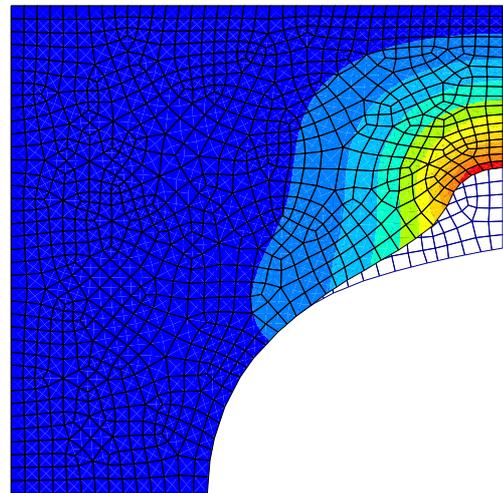
(g) Campo de velocidade n°7



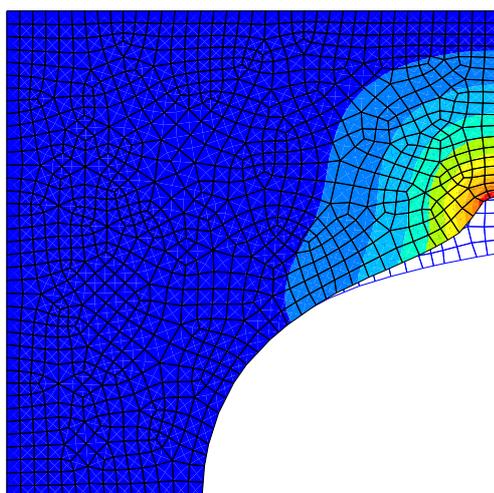
(h) Campo de velocidade n°8



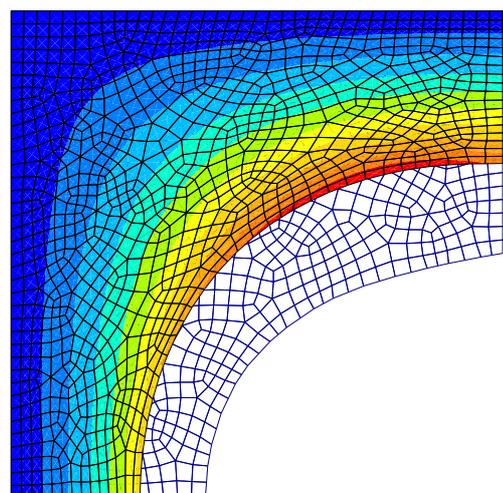
(i) Campo de velocidade n°9



(j) Campo de velocidade n°10



(k) Campo de velocidade n°11



(l) Campo de velocidade n°12

Figura 35: Campos de velocidades gerados usando campos de pressão.

6.4 Concentração de tensões

O exemplo de uma placa carregada axialmente^[11], com um fator de concentração de tensão K_t , foi resolvido, figura 36. Partindo de um valor determinado, o programa de otimização revela a nova forma, onde houve melhora significativa no valor de K_t .

A linha tracejada, na figura 36, mostra o contorno com transição abrupta entre as seções, que provocará uma forte concentração de tensão neste ponto. Conforme os manuais de projeto de elementos de máquinas, bastaria um pequeno raio de concordância para aliviar a concentração de tensão. De acordo com Peterson^[21], existe um método ainda melhor, desenvolvido por Grodzinski, que transforma a transição em um perfil parabólico, praticamente isento de concentração de tensão. Ocorre que a seção, definida por este método, tem a sua área aumentada, o que nem sempre pode ser aceito, devido às próprias restrições do projeto. A otimização desta peça visou tanto à redução de área (ou volume), quanto à redução nos valores de tensão.

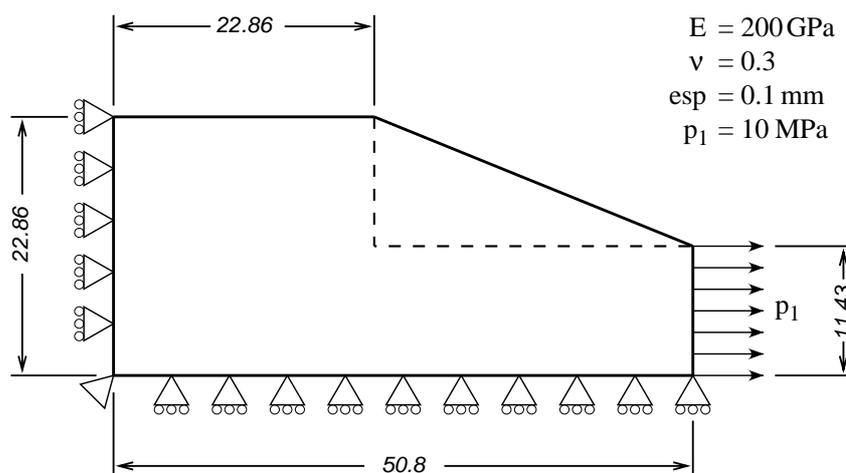


Figura 36: Dimensões, condições de contorno e restrições geométricas (dimensões em mm).

A geometria inicial foi, devido à simetria, modelada em sua quarta parte e foi criada uma transição linear entre as seções. É necessário modelar desta forma para que as tensões não atinjam valores muito elevados, o que faria com que as restrições fossem violadas já no início do programa. Modelando de acordo com a figura 36, as tensões no modelo são próximas da pressão aplicada e pode-se iniciar com um valor de tensão limite mais baixo. A figura 37 mostra a concentração de tensão para a situação inicial, conforme citado em 11.

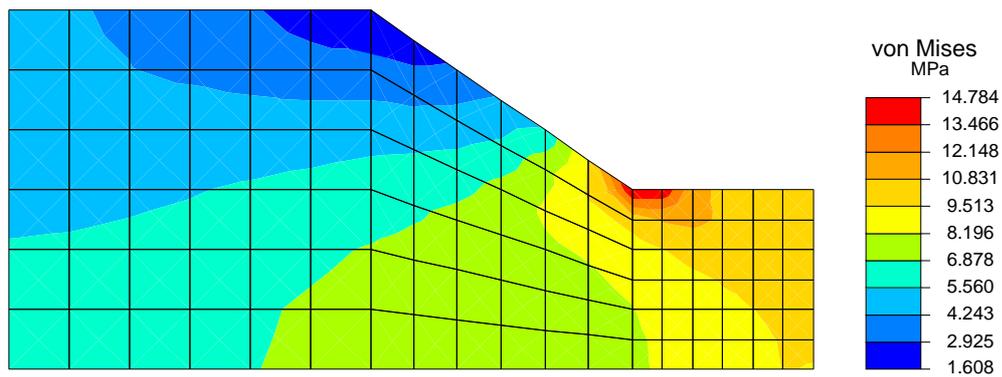


Figura 37: Tensões equivalentes de von Mises.

O primeiro modelo convergiu após alcançar o limite de 100 iterações. O critério de parada foi deixado bastante pequeno (na ordem de 10^{-2}), para que se pudesse verificar se haveria distorção na malha, o que não ocorreu.

A distribuição de tensões, figura 38, após a primeira convergência apresenta um resultado melhor do que o encontrado no modelo construído baseado no método de Grodzinski, figura 39. Apesar da malha grosseira, necessária para permitir maiores deformações durante a otimização, o resultado no fator de concentração foi de 1,075 contra 1,085.

Após *remesh*, onde foi aumentado o número de elementos para melhorar o mapeamento da tensão, foi encontrada a solução mostrada na figura 40, após 71 iterações. Pode-se observar que a tensão caiu de 14,8MPa para 12,1MPa e da mesma forma o K_t passou de 1,48 para 1,21.

O resultado obtido pela otimização reduz de fato a concentração de tensão, não

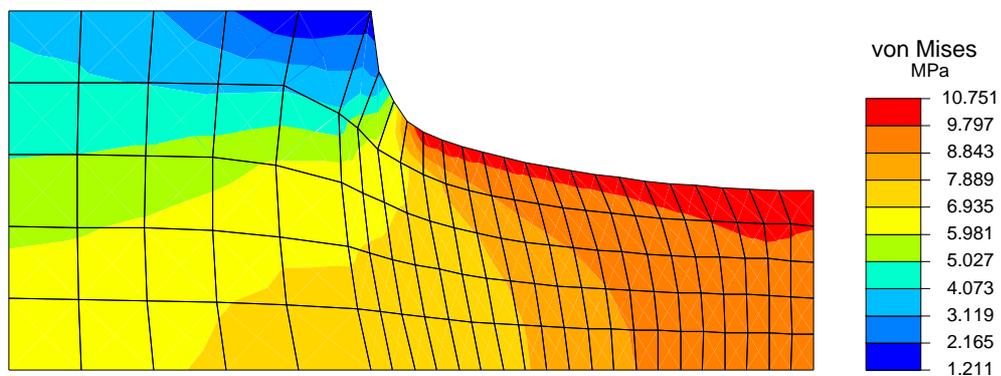


Figura 38: Malha e distribuição de tensões após a primeira convergência.

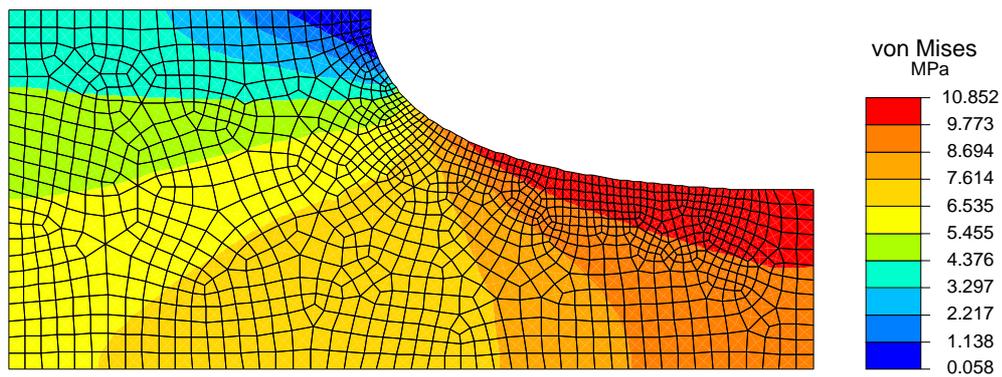


Figura 39: Distribuição de tensões no modelo conforme Grodzinski.

para um valor comparável aos citados anteriormente, mas além de baixar o fator de concentração de tensão (quando comparado aos valores iniciais), ainda reduz o volume do modelo em aproximadamente 38%. A referência 11 afirma ter obtido o valor de $K_t = 1,05$.

O histórico, mostrado na figura 41, demonstra a necessidade de fazer um *remesh*, praticamente a cada iteração. Com isto a malha se livra de quaisquer distorções. Pode-se, ainda, usar elementos de ordem mais elevada, assim o resultado viria em um menor número de iterações. As malhas iniciais usadas aqui apresentam um baixo grau de refinamento, justamente por não haver *remesh* automático, porém, como foi visto para algumas geometrias, permitem um grande número de iterações sem que ocorra a deformação de nenhum elemento.

Para este exemplo foi gerado campos de velocidade com aplicação de cargas virtuais unitárias.

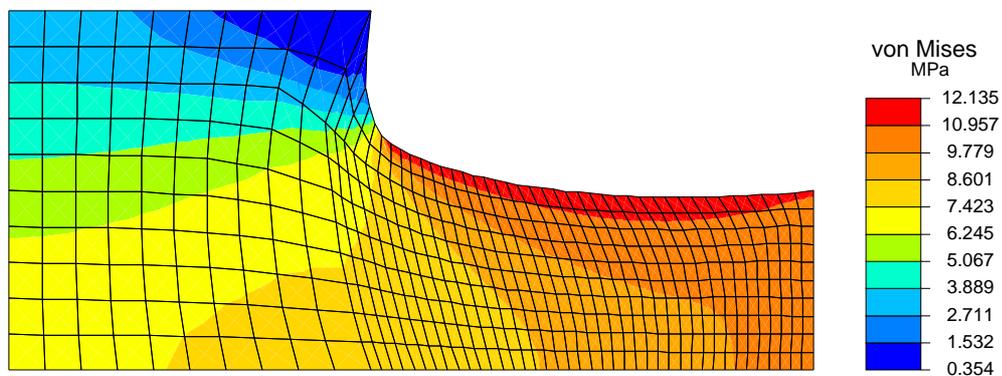


Figura 40: Distribuição de tensões após otimização.

Ao se retirar as restrições geométricas, linhas tracejadas na figura 36, observa-se uma considerável remoção de material, dentro dos limites de tensão aplicados, figura 42.

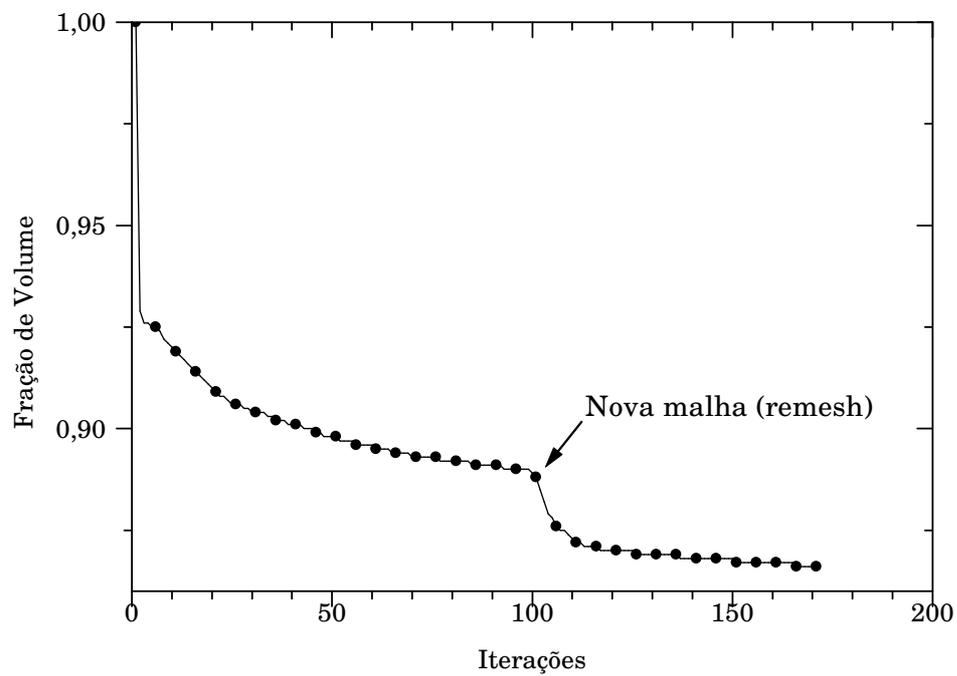


Figura 41: Histórico de iterações para o modelo de concentração de tensões.

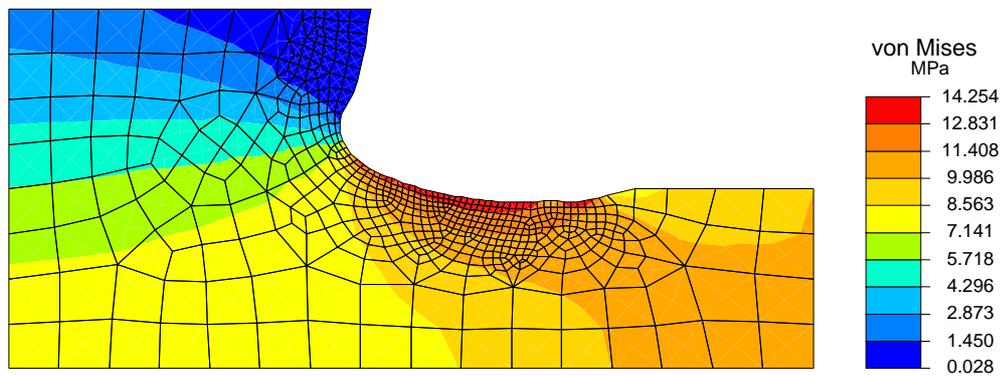


Figura 42: Forma final ao se retirar as restrições geométricas.

CAPÍTULO 7

Considerações Finais

Neste trabalho foi apresentada a otimização de forma empregando o método das direções viáveis de Zoutendijk e o método das variáveis naturais como variáveis de projeto.

Foram apresentados e discutidos exemplos de problemas de tensão plana bi-dimensional e comparados com os exemplos da literatura.

Os resultados apresentados são bastante similares quando se comparam os valores da função objetivo e restrição. Por conta do aplicativo desenvolvido, o qual ainda não contempla geração automática de malha a cada iteração (*remesh*), os valores para o número de iterações e para o número de chamadas da função, são os que apresentam as maiores discrepâncias.

Foi também investigada a influência dos campos de velocidade na obtenção dos resultados. Apesar de não ser possível determinar a escolha diretamente, ou seja, cargas unitárias ou campos de pressão, pode-se inferir que para perímetros extensos ou onde se espera grande retirada de material os campos de pressão são mais eficientes, pois apresentam uma continuidade melhor na vizinhança de aplicação da pressão, o que em parte simula o comportamento de uma *spline*. Outra vantagem dos campos de pressão é o fato de se poder trabalhar com malhas não estruturadas, geradas diretamente de algum aplicativo comercial, sem que se criem distorções. Isto é conseguido escolhendo valores

adequados para a matriz característica. Baixos valores para o módulo de elasticidade permitem que a deformação da malha alcance muitos nós abaixo da aplicação da carga, fator este que diminui sensivelmente a distorção.

Por outro lado, se o interesse recai em otimizar uma pequena região, a aplicação de cargas unitárias levará a melhores resultados, mas em compensação exigirá que a malha seja estruturada e que seus elementos sejam de grande tamanho, caso contrário o elemento será rapidamente distorcido, exigindo a confecção de uma nova malha e uma nova otimização.

O exemplo da placa com furo central, o qual apresentou volume final inferior aos apresentados na literatura, foi iniciado com cargas unitárias, e após a região a ser otimizada estar definida, foram aplicados campos de pressão, que apresentaram uma solução rápida (13 iterações) e um contorno suave ao longo do perímetro interno.

7.1 Sugestões para futuros trabalhos

Apresentam-se a seguir sugestões para o desenvolvimento de futuros trabalhos:

- Implantar e testar o desempenho do aplicativo com o uso de elementos triangulares (controle de distorção);
- Verificar a influência do uso de elementos de ordem superior;
- Acoplar ao aplicativo um gerador automático de malhas e permitir aproximação do contorno por *splines*;
- Comparar o desempenho dos diversos algoritmos com o método aqui apresentado;
- Construir uma interface gráfica mais amigável que permita a geração automática de campos de velocidade;
- Acoplar o processo de otimização com o método dos elementos de contorno e verificar desempenho.

Referências

- 1 RAO, S. S. *Engineering Optimization*. 3. ed. New York: Wiley, 1996.
- 2 LUENBERGER, D. G. *Linear and Nonlinear Programming*. 2. ed. Reading: Addison-Wesley, 1989.
- 3 FRITZSCHE, H. *Programação não Linear*. São Paulo: Blücher, 1978.
- 4 BAZARAA, M. S.; SHERALI, H. D.; SHETTY, C. M. *Nonlinear Programming*. 2. ed. New York: Wiley, 1993.
- 5 MANGASARIAN, L. O. *Nonlinear Programming*. Philadelphia: SIAM, 1994.
- 6 FLETCHER, R. *Practical Methods of Optimization*. Chichester: Wiley, 2001.
- 7 FIACCO, A. V.; McCORMICK, G. P. *Nonlinear Programming*. Philadelphia: SIAM, 1990.
- 8 BELEGUNDU, A. D.; CHANDRUPATLA, T. R. *Optimization Concepts and Applications in Engineering*. Upper Saddle River: Prentice-Hall, 1999.
- 9 CHANG, K. H.; TANG, P. S. Integration of design and manufacturing for structural shape optimization. *Advances in Engineering Software*, n. 32, p. 555–567, 2001.
- 10 SADEK, E. A. Minimum weight design of structures under frequency and frequency response constrains. *Computers & Structures*, v. 60, n. 1, p. 73–77, 1996.
- 11 HSU, Y. A review of structural shape optimization. *Computers in Industry*, v. 26, p. 3–13, 1994.
- 12 BELEGUNDU, A. D.; RAJAN, S. D. A shape optimization approach based on natural design variables and shape functions. *Computer Methods in Applied Mechanics and Engineering*, n. 66, p. 87–106, 1988.

- 13 CHARGIN, M. K. General shape optimization capability. *Finite Elements in Analysis and Design*, v. 7, p. 343–354, 1991.
- 14 SALAJEGHEH, E. Optimum design of structures with high quality approximation of frequency constrains. *Advances in Engineering Software*, v. 31, p. 381–384, 2000.
- 15 ATREK, E. Virtual loads in response sensitivity analysis. In: SYMPOSIUM ON MULTIDISCIPLINARY ANALYSIS AND OPTIMIZATION. Bellevue: AIAA/NASA/ISSMO, 1996.
- 16 MARTINS, J. R. R. A. Sensitivity analysis. 2003. Disponível em: <<http://mdolab.utias.utoronto.ca/documents/aa222sa.pdf>>. Acesso em: 11 jul. 2003.
- 17 RICHARDS, R. *Zeroth-order shape optimization utilizing a learning classifier*. [s.n.], 1995. Disponível em: <<http://stanford.edu/~buc/SPHINcsX/book.html>>. Acesso em: 01 jul. 2003.
- 18 NAQUIB, R. A.; ZUREICK, A.; WILL, K. M. Practical considerations in two-dimensional shape optimization of elastic continuum. *Computers & Structures*, v. 61, n. 2, p. 315–330, 1996.
- 19 ZHANG, S.; BELEGUNDU, A. D. Mesh distortion control in shape optimization. *AIAA Journal*, v. 31, n. 7, p. 1360–1362, 1992.
- 20 PARENTE JUNIOR, E. *Análise de sensibilidade e otimização de forma de estruturas geometricamente não-lineares*. Tese (Doutorado em Ciências) — Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2000.
- 21 PETERSON, R. E. *Stress Concetration Factors*. New York: Wiley, 1974.