

TÉCNICAS DE PONTO INTERIOR PARA SISTEMAS DE EQUAÇÕES E  
OTIMIZAÇÃO NÃO LINEAR

Alfredo Canelas Botta

TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS  
PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE  
FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS  
NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS  
EM ENGENHARIA MECÂNICA.

Aprovada por:

---

Prof. José Herskovits Norman, D. Ing.

---

Prof. Nestor Zouain Pereira, D. Sc.

---

Prof. Alfredo Rocha Faria, D. Sc.

---

Prof. Berardi Sensale, D. Ing.

RIO DE JANEIRO, RJ - BRASIL

MARÇO DE 2005

BOTTA, ALFREDO CANELAS

Técnicas de Ponto Interior para Sistemas  
de Equações e Otimização Não Linear [Rio  
de Janeiro] 2005

IX, 117 p. 29,7 cm (COPPE/UFRJ,  
M.Sc., Engenharia Mecânica, 2005)

Tese - Universidade Federal do Rio de  
Janeiro, COPPE

1. Otimização não linear.

I. COPPE/UFRJ II. Título (série)

# Agradecimentos

Ao professor Herskovits, pela orientação e pelo grande apoio para a realização deste trabalho.

Ao CNPQ pelo apoio financeiro.

Ao corpo docente do Programa de Engenharia Mecânica.

Aos professores do “Instituto de Estructuras y Transporte” de Uruguai pelo incentivo constante.

Ao grande amigo Sandro Mazorche, o qual trabalhou comigo muitas horas na frente do quadro, agradeço a ele pelas valiosas observações e comentários.

Ao apoio dos colegas e amigos do Laboratório Optimize: Paulo, Veranise, Evandro, Passarella, Moisés, Gabriel, Miguel e Marcelo.

Ao pessoal administrativo do Programa de Engenharia Mecânica.

Aos meus queridos pais, Carlos e Marisa, pelo carinho, incentivo e apoio constantes.

Aos grandes amigos Eugênio e Antônio.

Às pessoas que colaboraram de forma direta ou indireta, por meio de incentivo, confiança e troca de experiências.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

## TÉCNICAS DE PONTO INTERIOR PARA SISTEMAS DE EQUAÇÕES E OTIMIZAÇÃO NÃO LINEAR

Alfredo Canelas Botta

Março/2005

Orientador: José Herskovits Norman

Programa: Engenharia Mecânica

Este trabalho desenvolve técnicas de ponto interior baseadas no método de Newton para solução de sistemas de equações não lineares e de problemas de otimização não linear com restrições de igualdade e desigualdade.

Em primeiro lugar é apresentado um algoritmo para resolver sistemas de equações não lineares que tem como característica fundamental gerar uma seqüência de pontos convergente para a solução do problema que, em cada iteração, respeita algumas restrições de desigualdade dadas. Para este algoritmo foram obtidos resultados de convergência global e de velocidade de convergência.

Em segundo lugar são apresentados algoritmos para solução de problemas de otimização não linear baseados no algoritmo descrito anteriormente.

Por último, são apresentados resultados obtidos com os algoritmos descritos para dois exemplos de otimização: O primeiro exemplo é um problema freqüentemente utilizado como problema teste e o segundo exemplo é o conhecido problema de otimização topológica de treliças. Os resultados obtidos são promissores e sugerem temas de pesquisa futuros.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

INTERIOR-POINT TECHNIQUES FOR NONLINEAR SYSTEMS OF  
EQUATIONS AND NONLINEAR OPTIMIZATION

Alfredo Canelas Botta

March/2005

Advisor: José Herskovits Norman

Department: Mechanical Engineering

This work presents interior-point techniques based on Newton's method for the solution of nonlinear systems of equations and of nonlinear optimization problems with general constrains.

We present first an algorithm for the solution of nonlinear systems of equations that has as a fundamental characteristic the generation of a sequence of points that converges to a solution of the problem and satisfies a set of inequality constrains at all iterations. Some results about global convergence and speed of convergence are presented for this algorithm.

Then, there are presented some algorithms for the solution of nonlinear optimization problems based on the algorithm described before.

Some results obtained with the algorithms described are presented. These results address two optimization examples: The first example is often used as a benchmark problem, the second example is on the well known topologic optimization of trusses. The results obtained are encouraging and suggest topics for future research.

# Índice

<b>Introdução</b>	<b>1</b>
<b>1 Métodos de pontos interiores para otimização não linear</b>	<b>4</b>
1.1 Problema de Otimização não linear . . . . .	4
1.2 Métodos de pontos interiores . . . . .	6
1.2.1 Métodos de direções viáveis - Algoritmo FAIPA . . . . .	8
1.2.2 Método Primal-dual . . . . .	12
<b>2 Métodos viáveis para equações não lineares</b>	<b>15</b>
2.1 Método de Newton-Raphson . . . . .	15
2.2 Método de tipo Newton que respeita restrições de desigualdade . . . . .	16
2.2.1 Modificação da região viável . . . . .	24
2.2.2 Escolha do parâmetro $\sigma$ . . . . .	26
2.2.3 Correção de segunda ordem . . . . .	33
<b>3 Métodos viáveis para problemas de otimização</b>	<b>36</b>
3.1 Métodos viáveis para problemas convexos . . . . .	36
3.2 Métodos viáveis para problemas gerais . . . . .	40
3.2.1 Atualização quase-Newton . . . . .	43
3.2.2 Correção de segunda ordem . . . . .	46
3.2.3 Simetrização e redução do sistema linear . . . . .	47
3.2.4 Problemas com restrições de igualdade . . . . .	49
3.3 Métodos viáveis para problemas degenerados . . . . .	50
3.3.1 Função de Fischer-Burmeister . . . . .	52
3.3.2 Algoritmo para problemas degenerados . . . . .	52

<b>4</b>	<b>Resultados numéricos</b>	<b>57</b>
4.1	Problema da mínima superfície com obstáculo . . . . .	57
4.2	Otimização topológica de treliças . . . . .	63
4.2.1	Modelo 1: Minimização da energia de deformação . . . . .	65
4.2.2	Modelo 2: Minimização do volume . . . . .	67
4.2.3	Modelo 3: Formulação implícita . . . . .	71
4.2.4	Resultados obtidos, Modelos 1 e 2 . . . . .	73
<b>5</b>	<b>Conclusões</b>	<b>85</b>
	<b>Referências bibliográficas</b>	<b>87</b>
	<b>Anexo 1</b>	<b>91</b>

# Lista de símbolos

$\mathbb{R}^n$	Conjunto de vetores reais de dimensão $n$
$\mathbb{R}^{m \times n}$	Conjunto de matrizes reais de dimensões $m \times n$
$x$	Vetor de variáveis primais ou variáveis de projeto
$f$	Função objetivo. $f : \mathbb{R}^n \rightarrow \mathbb{R}$
$g$	Função das restrições de desigualdade. $g : \mathbb{R}^n \rightarrow \mathbb{R}^m$
$h$	Função das restrições de igualdade. $h : \mathbb{R}^n \rightarrow \mathbb{R}^p$
$\nabla f$	Gradiente da função $f(x)$ , $\nabla f : \mathbb{R}^n \rightarrow \mathbb{R}^n$
$\nabla g$	Gradiente da função $g(x)$ , $\nabla g : \mathbb{R}^n \rightarrow \mathbb{R}^{m \times n}$
$\nabla h$	Gradiente da função $h(x)$ , $\nabla h : \mathbb{R}^n \rightarrow \mathbb{R}^{p \times n}$
$\Omega$	Região viável, conjunto de pontos que satisfazem as restrições
$\lambda$	Multiplicadores de Lagrange das restrições de desigualdade, $\lambda \in \mathbb{R}^m$
$\mu$	Multiplicadores de Lagrange das restrições de igualdade, $\mu \in \mathbb{R}^p$
$L$	Função lagrangeana. $L : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p \rightarrow \mathbb{R}$
$H$	Matriz hessiana. $H : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p \rightarrow \mathbb{R}^{n \times n}$
$x_k, \lambda_k, \mu_k$	Valores das variáveis na iteração $k$
$f_k, g_k, h_k$	$f(x_k), g(x_k), h(x_k)$ . Valores das funções no ponto $x_k$
$L_k, H_k$	$L(x_k, \lambda_k, \mu_k), H(x_k, \lambda_k, \mu_k)$ . Valores das funções no ponto $(x_k, \lambda_k, \mu_k)$
$t$	Passo, $t \in \mathbb{R}$
$d_x, d_\lambda, d_\mu$	Direções de busca. $d_x \in \mathbb{R}^n, d_\lambda \in \mathbb{R}^m, d_\mu \in \mathbb{R}^p$
$\Lambda_k, G_k$	Matrizes diagonais, $\text{diag}(\lambda_k), \text{diag}(g_k)$
$S_k$	Matriz diagonal, $S_k = \Lambda^{-1}G_k$
$B_k$	Matriz quase-Newton, aproximação da matriz hessiana na iteração $k$



$F$	Função, $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$
$F_k$	Valor da função $F$ no ponto $x_k$ , $F(x_k)$
$\sigma$	Parâmetro de desvio
$\beta, \alpha, \eta, \nu$	Outros parâmetros reais
$\ \cdot\ $	Norma euclideana
$\Phi$	Função potencial, $\Phi : \mathbb{R}^n \rightarrow \mathbb{R}$
$\psi$	Função de Fischer-Burmeister, $\psi : \mathbb{R}^2 \rightarrow \mathbb{R}$

# Introdução

Em diversas áreas da engenharia freqüentemente aparecem problemas de otimização complexos e de grande porte. Isto acontece, por exemplo, quando as variáveis e funções que definem o problema de otimização derivam da discretização de corpos materiais contínuos. Com o objetivo de resolver estes problemas de otimização foram estudadas diversas técnicas baseadas nos métodos de pontos interiores utilizados na otimização não linear.

Diversos estudos [1, 2, 3] mostram que os métodos de pontos interiores constituem poderosas ferramentas para resolução de problemas de otimização não linear de grande escala. Os algoritmos de pontos interiores parecem ser os mais robustos e eficientes como algoritmos de otimização de propósito geral. Na resolução de problemas de grande escala os métodos de pontos interiores apresentam-se como os métodos de melhor escalabilidade [1, 3].

Os métodos de pontos interiores têm sido intensamente estudados a partir da apresentação do algoritmo de Karmarkar [4], o primeiro algoritmo de complexidade polinomial e eficiente na prática capaz de ser um forte competidor do algoritmo SIMPLEX na resolução do problema de programação linear. A partir da apresentação do algoritmo de Karmarkar, um grande número de pesquisadores tem se dedicado ao estudo dos métodos de pontos interiores, produzindo um grande número de trabalhos. Os métodos de pontos interiores têm produzido um grande avanço nas áreas de otimização linear e não linear. Em alguns casos tem se apresentado como as melhores ferramentas para resolver importantes classes de problemas, em outros têm produzido uma forte competitividade que derivou em avanços importantes em outras classes de métodos [5]. Na atualidade os métodos de ponto interior de tipo Primal-dual (métodos onde a teoria de dualidade desempenha um importante papel e é dado um tratamento igual às variáveis primais e duais) são considerados os mais promissores, de modo que grande parte da pesquisa em métodos de pontos interiores está concentrada no estudo destes métodos.

De acordo com o pesquisador Tapia [6] o papel fundamental da metodologia de pontos interiores é produzir, numa forma significativa e efetiva, um sistema de

equações não lineares que representa suficientemente bem o problema de otimização com restrições de desigualdade de forma que a aplicação do método de Newton nesse sistema é efetiva e bem sucedida.

Tendo isto em consideração, um dos assuntos principais deste trabalho é o desenvolvimento de técnicas baseadas no método de Newton para resolução de sistemas de equações não lineares. As mesmas são desenvolvidas com o objetivo principal de aplicá-las no sistema não linear de equações dado pelas igualdades das condições de otimalidade de Karush-Kuhn-Tucker. O outro tema principal é o estudo de técnicas específicas para resolução do problema geral de otimização não linear, baseadas no trabalho realizado sobre resolução de sistemas de equações não lineares.

Este trabalho está organizado em capítulos da seguinte forma:

No Capítulo 1 é apresentada a formulação matemática do problema de otimização não linear, junto com algumas definições habituais e alguns resultados preliminares. Posteriormente são descritos dois métodos de pontos interiores de busca linear, o método de direções viáveis, e o método Primal-dual. Como algoritmo representativo dos métodos de pontos interiores de direções viáveis é descrito o algoritmo FAIPA, apresentado por Herskovits [7].

No Capítulo 2 é descrito o algoritmo de Newton-Raphson, algoritmo para resolver sistemas de equações não lineares no qual são baseados os métodos de pontos interiores. Posteriormente é descrito um método de pontos interiores para resolver sistemas de equações não lineares. A característica principal deste método, que o diferencia do método de Newton, é que a seqüência de pontos gerada satisfaz em cada iteração um conjunto de restrições de desigualdade dadas. Esta característica o faz conveniente para desenvolver métodos de pontos interiores para resolver problemas de otimização não linear. Para este método são apresentados alguns resultados de convergência global e de velocidade de convergência.

No Capítulo 3 são apresentados alguns métodos para problemas de otimização baseados no método descrito no Capítulo 2.

No Capítulo 4 são estudados dois exemplos de otimização conhecidos que permitem testar o desempenho dos algoritmos apresentados no Capítulo 3. Como exemplos foram utilizados o problema da mínima superfície com obstáculo, problema

relativamente fácil de resolver, e o conhecido e difícil problema de otimização topológica de treliças.

Por último, no Capítulo 5 são descritas as conclusões deste trabalho.

# 1 Métodos de pontos interiores para otimização não linear

Na seção 1.1 deste capítulo é descrito o problema de otimização não linear. Na seção 1.2 descrevem-se dois métodos de pontos interiores para resolução destes problemas.

## 1.1 Problema de otimização não linear

Um problema de otimização consiste na minimização ou maximização de uma determinada função (função objetivo) sujeito a restrições nas suas variáveis. Matematicamente o problema geral de otimização não linear pode ser formulado como:

$$\begin{aligned} & \text{minimize} && f(x) \\ & \text{sujeito a:} && g(x) \geq 0 \\ & && h(x) = 0 \end{aligned} \tag{1.1}$$

Na Eq. 1.1  $x \in \mathbb{R}^n$  é o vetor de variáveis do problema ou variáveis de projeto,  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  é a função objetivo,  $g : \mathbb{R}^n \rightarrow \mathbb{R}^m$  é a função que define as restrições de desigualdade e  $h : \mathbb{R}^n \rightarrow \mathbb{R}^p$  é a função das restrições de igualdade. As funções  $f$ ,  $g$  e  $h$  devem ser funções contínuas com derivadas contínuas.

**Definição 1.1** (*Região viável*) A região viável  $\Omega$  é o conjunto de pontos que satisfazem as restrições:  $\Omega = \{x \mid g(x) \geq 0, h(x) = 0\}$

**Definição 1.2** (*Mínimo local*)  $x^*$  é um mínimo local do problema 1.1 se existe uma vizinhança  $N$  de  $x^*$  tal que  $f(x) \geq f(x^*)$  para todo ponto  $x$  pertencente a  $\Omega \cap N$ .

**Definição 1.3** (*Conjunto de restrições ativas*) Dado um ponto  $x \in \Omega$ , o conjunto de restrições de desigualdade ativas é:  $A(x) = \{i \in \{1, \dots, m\} \mid g_i(x) = 0\}$

**Definição 1.4** (*LICQ*) Dado um ponto  $x \in \Omega$ , as restrições satisfazem o requisito de independência linear em  $x$  (*LICQ* por sua sigla em inglês) se os gradientes das restrições de igualdade e de desigualdade ativas são linearmente independentes, ou seja, o conjunto  $\{\nabla g_i(x) \mid i \in A(x), \nabla h_i(x) \mid i \in \{1, \dots, p\}\}$  é linearmente independente.

**Definição 1.5 (KKT)** *Seja  $x \in \mathbb{R}^n$ , se satisfazem as condições de primeira ordem de Karush-Kuhn-Tucker (KKT) em  $x$  se existem vetores  $\lambda \in \mathbb{R}^m$  e  $\mu \in \mathbb{R}^p$  que verificam:*

$$\nabla f(x) - \sum_{i=1}^m \lambda_i \nabla g_i(x) - \sum_{i=1}^p \mu_i \nabla h_i(x) = 0 \quad (1.2)$$

$$g_i(x) \lambda_i = 0, \quad i = 1, \dots, m \quad (1.3)$$

$$h(x) = 0 \quad (1.4)$$

$$g(x) \geq 0 \quad (1.5)$$

$$\lambda \geq 0 \quad (1.6)$$

**Teorema 1.1** *Seja  $x$  um mínimo local do problema 1.1 onde se satisfaz o requisito LICQ, então se satisfazem as condições de primeira ordem de KKT no ponto  $x$  e ademais os vetores  $\lambda$  e  $\mu$  são únicos.*

Demonstração: Ver na referência [8].

**Definição 1.6 (Função lagrangeana)** *A função lagrangeana para o problema 1.1 é a função  $L : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p \rightarrow \mathbb{R}$  definida como:*

$$L(x, \lambda, \mu) = f(x) - \sum_{i=1}^m \lambda_i g_i(x) - \sum_{i=1}^p \mu_i h_i(x) \quad (1.7)$$

Com esta definição a Eq. 1.2 pode ser escrita de forma mais compacta como:

$$\nabla_x L(x, \lambda, \mu) = 0$$

**Definição 1.7 (Hessiana da função lagrangeana)** *A hessiana da função lagrangeana para o problema 1.1 é a função  $H : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p \rightarrow \mathbb{R}^{n \times n}$  definida como:*

$$H(x, \lambda, \mu) = \nabla_{xx}^2 L(x, \lambda, \mu) = \nabla^2 f(x) - \sum_{i=1}^m \lambda_i \nabla^2 g_i(x) - \sum_{i=1}^p \mu_i \nabla^2 h_i(x) \quad (1.8)$$

## 1.2 Métodos de pontos interiores

Em geral, a única forma de resolver o problema de otimização não linear dado pela Eq. 1.1 é por meio de um método iterativo. Os métodos de pontos interiores compreendem uma ampla gama de técnicas iterativas para achar um mínimo local do problema. Os mesmos geram uma seqüência de pontos convergente para um mínimo local e que satisfazem em forma estrita as restrições de desigualdade do problema. O interesse em métodos de pontos interiores cresceu muito na década de 1980 a partir da apresentação do algoritmo de Karmarkar [4], o primeiro algoritmo de complexidade polinomial e eficiente na prática para resolver o problema de otimização linear. As melhores propriedades de complexidade dos métodos de pontos interiores os fazem muito competitivos em problemas de grande escala comparados com o método clássico da programação linear, o método SIMPLEX. Em problemas de programação não linear, diversos estudos [1, 2, 3] indicam que os métodos de pontos interiores são mais eficientes e robustos que outro tipo de métodos para problemas tanto de pequeno como de grande tamanho.

Algumas vantagens importantes dos métodos de pontos interiores são:

- Muito convenientes para problemas esparsos ou com outro tipo de estrutura. É simples adaptá-los para tomar o máximo proveito das características destes problemas.
- Muito convenientes para executar em computadores de arquitetura paralela, pois podem ser implementados de forma a aproveitar ao máximo o potencial destes computadores.

Os métodos de pontos interiores podem ser classificados em dois grandes grupos:

1. Métodos viáveis: Tal qual foi descrito anteriormente estes métodos geram uma seqüência de pontos que satisfazem estritamente as restrições de desigualdade do problema. Algumas das vantagens destes métodos são:
  - A função objetivo pode não ser definida para pontos que não satisfazem algumas das restrições do problema.

- Em aplicações de tempo real pode ser necessário utilizar o próximo ponto que o método gera em vez de esperar a alcançar a otimalidade. Se o método é viável, esse ponto satisfaz as restrições de desigualdade, requisito que pode ser fundamental para a utilização dessa solução não ótima.

2. Métodos inviáveis: A maior diferença com respeito dos métodos viáveis é que neste caso se agregam variáveis de folga definindo o seguinte problema equivalente ao anterior:

$$\begin{aligned}
 & \text{minimize} && f(x) \\
 & \text{sujeito a:} && g(x) - s = 0 \\
 & && h(x) = 0 \\
 & && s \geq 0
 \end{aligned} \tag{1.9}$$

Os pontos da seqüência que geram os métodos inviáveis satisfazem a desigualdade estrita nas variáveis de folga, ou seja,  $s > 0$  em cada iteração, mas não satisfazem em geral as restrições de desigualdade do problema original  $g(x) \geq 0$ .

Algumas das vantagens dos métodos inviáveis são:

- Em alguns problemas não existe um ponto evidente que satisfaça as restrições de desigualdade. Nestes casos é em geral ineficiente gerar uma seqüência que primeiro ache um ponto na região viável e logo procure uma solução [9].
- A região de pontos definida pelas restrições de desigualdade é em geral bem mais simples no problema 1.9. Isto pode facilitar muito o progresso do algoritmo à solução do problema. A implementação da busca linear nos algoritmos inviáveis é em geral muito simples.

Excetuando os casos óbvios em que é evidente que tipo de método empregar, não existe consenso sobre que método é melhor. Um bom trabalho que compara o desempenho de um mesmo algoritmo de pontos interiores nas suas versões viável e inviável encontra-se na referência [10].



Neste capítulo descrevem-se dois métodos de pontos interiores freqüentemente utilizados para resolver o problema de otimização não linear 1.1. Os métodos descritos nas seções 1.2.1 e 1.2.2 são muito similares na sua implementação final, a maior diferença é, talvez, que o primeiro foi concebido como um método de direções viáveis e o outro como um método de barreiras logarítmicas, eles têm a característica em comum de serem baseados na iteração de Newton sobre o sistema de igualdades dado pelas condições de otimalidade de primeira ordem, Eqs. 1.2 a 1.4. A seção 2.1 do próximo capítulo deste trabalho descreve o método de Newton-Raphson no qual estão baseados os métodos descritos nas seções 1.2.1 e 1.2.2.

### 1.2.1 Métodos de direções viáveis - Algoritmo FAIPA

Os métodos de direções viáveis são métodos iterativos que resolvem o problema geral de otimização não linear movendo-se desde um ponto viável (que pertence à região  $\Omega$  de pontos que satisfazem as restrições do problema) a outro ponto viável onde a função objetivo adota um valor menor. Dado um ponto viável  $x_k$ , o método determina uma direção  $d_x$  para a qual existe um valor  $t > 0$  que cumpre com as seguintes propriedades:

- $x_k + td_x \in \Omega$
- $f(x_k + td_x) < f(x_k)$

Logo depois de se determinar a direção  $d_x$ , um problema de otimização de uma única variável é resolvido para determinar o valor de  $t$  (passo). A resolução deste subproblema é conhecida como busca linear.

Para o problema de otimização não linear sem restrições de igualdade, o algoritmo FAIPA (*Feasible Arc Interior Point Algorithm*) proposto por Herskovits [7] é um algoritmo de direções viáveis. Em cada iteração é achada uma direção viável e de descida para a função objetivo. Uma busca linear inexata nessa direção para a função objetivo determina o próximo ponto viável. Para definir a direção de busca o algoritmo FAIPA baseia-se na direção da iteração de Newton na variável  $(x, \lambda)$  para o sistema de equações dado pelas igualdades das condições de otimalidade de primeira ordem Eqs. 1.2 a 1.4.

Seja por exemplo um problema de otimização sem restrições de igualdade. A iteração de Newton (amortecida) é:

$$(x_{k+1}, \lambda_{k+1}) = (x_k, \lambda_k) + t(d_x, d_\lambda)$$

onde a direção  $(d_x, d_\lambda)$  verifica:

$$\begin{pmatrix} H_k & -\nabla g_k^T \\ \Lambda_k \nabla g_k & G_k \end{pmatrix} \begin{pmatrix} d_x \\ d_\lambda \end{pmatrix} = \begin{pmatrix} -\nabla_x L_k^T \\ -\Lambda_k g_k \end{pmatrix} \quad (1.10)$$

Na Eq. 1.10,  $H_k = H(x_k, \lambda_k)$  é a hessiana da função lagrangeana,  $L_k = L(x_k, \lambda_k)$  é a função lagrangeana e  $\Lambda_k$  e  $G_k$  são as matrizes diagonais  $\text{diag}(\lambda_k)$  e  $\text{diag}(g(x_k))$ .

Definindo a variável  $\lambda = \lambda_k + d_\lambda$  a direção  $d_x$  pode-se achar resolvendo o sistema:

$$\begin{pmatrix} H_k & -\nabla g_k^T \\ \Lambda_k \nabla g_k & G_k \end{pmatrix} \begin{pmatrix} d_x \\ \lambda \end{pmatrix} = \begin{pmatrix} -\nabla f_k^T \\ 0 \end{pmatrix} \quad (1.11)$$

Em geral o método de Newton não converge para um mínimo local do problema 1.1 já que em alguns casos a direção  $d_x$  não está bem definida pois a matriz de coeficientes das Eqs. 1.10 e 1.11 é singular. De qualquer forma, mesmo que a matriz seja não singular para todos os pontos da seqüência gerada, esta seqüência pode ser divergente ou convergir para um ponto que não seja mínimo local do problema 1.1. Pode acontecer o caso no qual o algoritmo converge para um máximo local do problema 1.1, já que nesses pontos as equações de igualdade das condições de KKT verificam-se para valores não necessariamente positivos da variável  $\lambda$ . Para garantir convergência global para um ponto de KKT do problema 1.1 o algoritmo FAIPA modifica a forma em que calcula-se a direção de busca  $d_x$  e introduz uma busca linear na função objetivo para achar o passo  $t$ . As principais modificações são:

1. Modificação da matriz do sistema: Esta modificação é feita para que a matriz do sistema linear não seja singular e para garantir que a direção de busca seja de descida para a função objetivo. A matriz  $H_k$  é substituída por uma matriz  $B_k$  definida positiva. A direção achada resolvendo este sistema,  $d_x^a$ , é de descida para a função objetivo. A mesma calcula-se resolvendo o sistema:

$$\begin{pmatrix} B_k & -\nabla g_k^T \\ \Lambda_k \nabla g_k & G_k \end{pmatrix} \begin{pmatrix} d_x^a \\ \lambda_a \end{pmatrix} = \begin{pmatrix} -\nabla f_k^T \\ 0 \end{pmatrix} \quad (1.12)$$

2. Desvio da direção de busca: Ainda que a direção  $d_x^a$  seja de descida para a função objetivo, as restrições do problema podem obrigar o algoritmo a dar passos cada vez menores, levando a convergência da seqüência de pontos para um ponto na fronteira da região viável, não necessariamente mínimo local do problema. Para evitar este comportamento é necessário um desvio na direção de busca de forma que a nova direção aponte para dentro da região viável, permitindo um maior progresso á solução. A direção desviada calcula-se como:

$$d_x = d_x^a + \rho d_x^b$$

onde  $\rho$  é um valor suficientemente pequeno de modo que a direção  $d_x$  seja também de descida para a função objetivo. Calcula-se a direção de desvio  $d_x^b$  resolvendo o seguinte sistema linear:

$$\begin{pmatrix} B_k & -\nabla g_k^T \\ \Lambda_k \nabla g_k & G_k \end{pmatrix} \begin{pmatrix} d_x^b \\ \lambda_b \end{pmatrix} = \begin{pmatrix} 0 \\ -\Lambda_k \omega_k \end{pmatrix} \quad (1.13)$$

onde  $\omega_k \in \mathbb{R}_+^m$ .

3. Correção de segunda ordem: Outra modificação importante é uma correção de segunda ordem sobre a equação de complementaridade 1.3 para melhorar a convergência do algoritmo. Esta modificação é feita para evitar um efeito similar ao descrito por Maratos [11] que afetaria a velocidade de convergência do algoritmo se não fora implementada.

O novo ponto da seqüência se encontrará no arco  $x(t) = x_k + td_x + t^2 d_x^c$ , onde  $d_x^c$  calcula-se a partir do seguinte sistema linear:

$$\begin{pmatrix} B_k & -\nabla g_k^T \\ \Lambda_k \nabla g_k & G_k \end{pmatrix} \begin{pmatrix} d_x^c \\ \lambda_c \end{pmatrix} = \begin{pmatrix} 0 \\ -\Lambda_k \tilde{\omega}_k \end{pmatrix} \quad (1.14)$$

onde o vetor  $\tilde{\omega}_k$  calcula-se com a fórmula:

$$\tilde{\omega}_k = g(x_k + d_x) - g(x_k) - \nabla g(x_k) d_x$$

4. Busca linear: Consiste em achar  $t$  de modo de minimizar aproximadamente a função objetivo no arco de busca.

### Algoritmo FAIPA

Parâmetros:  $\psi \in (0, 1)$ ,  $\eta \in (0, 1)$ ,  $\phi > 0$  e  $\nu \in (0, 1)$ .

Dados:  $x_0 \in \Omega^0$ ,  $\lambda_0$  positivo,  $B_0 \in \mathbb{R}^{n \times n}$  simétrica definida positiva e  $\omega_0 \in \mathbb{R}^m$  positivo.

**Passo 0:** Teste de convergência

**Passo 1:** Cálculo do arco viável

1. Ache  $(d_x^a, \lambda_a)$  resolvendo o sistema linear da Eq. 1.12.

Se  $d_x^a = 0$ , fim.

2. Ache  $(d_x^b, \lambda_b)$  resolvendo o sistema linear da Eq. 1.13.

3. Se  $\nabla f_k d_x^b > 0$ , defina:

$$\rho = \min \left\{ \phi \|d_x^a\|^2; (\psi - 1) \frac{\nabla f_k d_x^a}{\nabla f_k d_x^b} \right\}$$

Se não, defina:

$$\rho = \phi \|d_x^a\|^2$$

4. Ache a direção desviada:

$$d_x = d_x^a + \rho d_x^b$$

também:

$$\bar{\lambda} = \lambda_a + \rho \lambda_b$$

5. Ache a direção  $d_x^c$  resolvendo o sistema linear da Eq. 1.14.

**Passo 2:** Busca linear

Calcule  $t$  como o primeiro número da seqüência  $\{1, \nu, \nu^2, \nu^3, \dots\}$  que satisfaça:

$$f(x_k + td_x + t^2 d_x^c) \leq f(x_k) + t\eta \nabla f_k d_x$$

$$g_i(x_k + td_x + t^2 d_x^c) > 0 \quad \text{se } \bar{\lambda} \geq 0, \text{ ou}$$

$$g_i(x_k + td_x + t^2 d_x^c) > g_i(x_k) \quad \text{outro caso,} \quad i = 1, \dots, m$$

**Passo 3:** Atualização

Defina  $x_{k+1} = x_k + td_x + t^2 d_x^c$

Defina valores para  $\lambda_{k+1} > 0$ ,  $B_{k+1}$  simétrica definida positiva e  $\omega_{k+1} > 0$ .

Retorne ao Passo 0.

### 1.2.2 Método Primal-dual

O método descrito nesta seção está muito relacionado com o método da seção anterior, as maiores diferenças estão na definição da direção de desvio e na função potencial utilizada. Nesta seção descreve-se a versão inviável do método que é a que usualmente é preferida. Para simplificar a notação considera-se o problema 1.9 sem restrições de igualdade. A descrição do método segue fundamentalmente aquela dada nas referências [12, 13].

O problema de barreira logarítmica correspondente ao problema 1.9 sem restrições de igualdade é:

$$\begin{aligned} \text{minimize} \quad & f(x) - \sigma \sum_{i=1}^m \log(s_i) \\ \text{sujeito a:} \quad & g(x) - s = 0 \end{aligned} \tag{1.15}$$

As condições de otimalidade para este problema são:

$$\begin{aligned} \nabla f(x) - \sum_{i=1}^m \mu_i \nabla g_i(x) &= 0 \\ -\frac{\sigma}{s_i} + \mu_i &= 0 \quad i = 1, \dots, m \\ g(x) - s &= 0 \end{aligned}$$

Definindo a variável  $\lambda_i = \sigma/s_i$  as condições de otimalidade podem escrever-se de forma equivalente como:

$$\nabla f(x) - \sum_{i=1}^m \mu_i \nabla g_i(x) = 0 \tag{1.16}$$

$$\mu - \lambda = 0 \tag{1.17}$$

$$g(x) - s = 0 \tag{1.18}$$

$$\lambda_i s_i = \sigma \quad i = 1, \dots, m \tag{1.19}$$

às quais devem-se agregar as condições:

$$s \geq 0$$

$$\lambda \geq 0$$

Em algumas implementações utiliza-se a Eq. 1.17 para eliminar a variável  $\mu$ . Fazendo isto as condições de otimalidade ficam:

$$\nabla f(x) - \sum_{i=1}^m \lambda_i \nabla g_i(x) = 0 \tag{1.20}$$

$$g(x) - s = 0 \quad (1.21)$$

$$\lambda_i s_i = \sigma \quad i = 1, \dots, m \quad (1.22)$$

$$s \geq 0 \quad (1.23)$$

$$\lambda \geq 0 \quad (1.24)$$

Definindo a matriz  $S_k = \text{diag}(s_k)$ , a iteração de Newton para o sistema de equações dado pelas Eqs. 1.20 a 1.22 é:

$$\begin{pmatrix} H_k & 0 & -\nabla g_k^T \\ 0 & \Lambda_k & S_k \\ \nabla g_k & -I & 0 \end{pmatrix} \begin{pmatrix} d_x \\ d_s \\ d_\lambda \end{pmatrix} = \begin{pmatrix} -L_k \\ -\Lambda_k s_k + \sigma \\ -g_k + s \end{pmatrix} \quad (1.25)$$

De forma de ter um sistema linear de equações não singular, e com ele obter uma direção de descida para uma função potencial conveniente, usualmente a matriz do sistema linear é modificada adicionando um múltiplo da matriz identidade,  $\varphi I$ , na matriz hessiana  $H_k$ , ou seja, o sistema linear que é resolvido é:

$$\begin{pmatrix} H_k + \varphi I & 0 & -\nabla g_k^T \\ 0 & \Lambda_k & S_k \\ \nabla g_k & -I & 0 \end{pmatrix} \begin{pmatrix} d_x \\ d_s \\ d_\lambda \end{pmatrix} = \begin{pmatrix} -L_k \\ -\Lambda_k s_k + \sigma \\ -g_k + s \end{pmatrix} \quad (1.26)$$

Na referência [13] prova-se que para  $\varphi$  e  $\beta$  suficientemente grandes a direção  $d_x$  achada é de descida para a função potencial dada pela Eq. 1.27.

$$\phi(x, s) = b_\sigma(x, s) + \frac{1}{2}\beta \|\rho(x, s)\|^2 \quad (1.27)$$

onde:

$$\begin{aligned} b_\sigma(x, s) &= f(x) - \sigma \sum_{i=1}^m \log(s_i) \\ \rho(x, s) &= s - g(x) \end{aligned}$$

A iteração do algoritmo Primal-dual consiste então em escolher os parâmetros  $\sigma$ ,  $\varphi$  e  $\beta$ , resolver o sistema linear para achar a direção  $(d_x, d_s, d_\lambda)$  e minimizar aproximadamente a função potencial nessa direção e dentro da região definida pelas desigualdades 1.23 e 1.24.

De forma mais detalhada o algoritmo Primal-dual é:

### Algoritmo Primal-dual

Parâmetros:  $\eta \in (0, 1)$  e  $\nu \in (0, 1)$ .

Dados:  $x_0, s_0$  positivo e  $\lambda_0$  positivo.

**Passo 0:** Teste de convergência

**Passo 1:** Cálculo da direção de busca

1. Escolha os parâmetros  $\sigma$  e  $\varphi$ .
2. Ache  $(d_x, d_s, d_\lambda)$  resolvendo o sistema linear da Eq. 1.26.

**Passo 2:** Busca linear

1. Escolha  $\beta$ .
2. Calcule  $t$  como o primeiro número da seqüência  $\{1, \nu, \nu^2, \nu^3, \dots\}$  que satisfaça:

$$\begin{aligned}\phi(x_k + td_x, s_k + td_s) &\leq \phi(x_k) + t\eta(\nabla_x \phi_k d_x + \nabla_s \phi_k d_s) \\ s_k + d_s &> 0 \\ \lambda_k + d_\lambda &> 0\end{aligned}$$

**Passo 3:** Atualização

Defina  $x_{k+1} = x_k + td_x$ ,  $s_{k+1} = s_k + td_s$ ,  $\lambda_{k+1} = \lambda_k + td_\lambda$

Retorne ao Passo 0.

## 2 Métodos viáveis para equações não lineares

Neste capítulo é estudado um método para resolução de sistemas de equações não lineares. A característica principal deste método, que o diferencia do método de Newton, é que a seqüência de pontos que gera satisfaz em cada iteração um conjunto de restrições de desigualdade dadas que definem uma região viável para o problema. Esta característica o faz conveniente para desenvolver métodos de pontos interiores para resolver problemas de otimização não linear.

Na seção 2.1 descreve-se o método de Newton, o método mais popular para resolução sistemas de equações não lineares no qual estão baseados a maioria dos outros métodos assim como também os métodos de pontos interiores para resolução de problemas de otimização.

Na seção 2.2 descreve-se o método objeto de estudo deste capítulo. Para este método são apresentados alguns resultados de convergência global e de velocidade de convergência.

### 2.1 Método de Newton-Raphson

O método de Newton-Raphson é um método iterativo que reduz a resolução de um sistema de  $n$  equações não lineares com  $n$  incógnitas à resolução de uma seqüência de sistemas de equações lineares. Seja o seguinte sistema de equações não lineares:

$$F(x) = 0 \quad (2.1)$$

onde a função  $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$  é uma função contínua com derivada contínua. A iteração de Newton para este sistema de equações é:

$$x_{k+1} = x_k + d_x \quad (2.2)$$

onde a direção  $d_x$  verifica:

$$\nabla F(x_k)d_x = -F(x_k) \quad (2.3)$$

Um método que em muitos casos pode ser mais conveniente que o método de Newton, é o método de Newton amortecido, neste caso a iteração é:

$$x_{k+1} = x_k + td_x \quad (2.4)$$

onde a direção  $d_x$  verifica a Eq. 2.3 e o valor  $t \in \mathbb{R}$  verifica  $0 < t \leq 1$ .



## 2.2 Método de tipo Newton que respeita restrições de desigualdade

Seja o seguinte sistema de equações não lineares:

$$F(x) = 0 \tag{2.5}$$

onde a função  $F : D \rightarrow \mathbb{R}^n$  é uma função contínua com derivada contínua definida no domínio  $D \subset \mathbb{R}^n$ . Em geral, a única forma de resolver o problema dado pela Eq. 2.5 é por meio de um método iterativo que gere uma seqüência de pontos que aproxime uma solução. Suponha que se deseja ademais que a seqüência de pontos que gera o método iterativo respeite as seguintes restrições de desigualdade:

$$F_i(x_k) \geq 0 \quad \text{para todo } i \in \{1, \dots, m\} \quad \text{com } m \leq n \tag{2.6}$$

Coerentemente com as desigualdades da Eq. 2.6 define-se a seguinte região viável:

$$\Gamma = \{x \in \mathbb{R}^n / F_i(x) \geq 0 \quad \text{para todo } i \in \{1, \dots, m\}\} \tag{2.7}$$

Um algoritmo que gere uma seqüência de pontos desta forma pode estar motivado pelas seguintes razões:

- As funções  $F_i$ ,  $i \in \{1, \dots, m\}$  podem não estar definidas para o argumento  $x$  fora da região  $\Gamma$  (por exemplo no caso em que  $\Gamma \equiv D$ ).
- O gradiente da função  $F$  pode ser singular para o argumento  $x$  fora da região  $\Gamma$ , tornando inviável a utilização de um método baseado na iteração de Newton.

Ambas as razões aparecem, por exemplo, ao resolver as condições de otimalidade de KKT em problemas de otimização.

A aplicação do método de Newton amortecido para problemas deste tipo não vai dar bons resultados em geral, pois a direção de Newton é tangente ao domínio  $\Gamma$  quando o ponto da iteração atual é um ponto na fronteira de  $\Gamma$ . Inclusive partindo de um ponto inicial afastado da fronteira de  $\Gamma$  a seqüência pode convergir a um ponto na fronteira que não seja solução do sistema de equações.

Uma possível forma de resolver o problema é modificar a forma em que é achada a direção de busca do método. Uma direção de busca mais conveniente acha-se resolvendo o seguinte sistema de equações lineares:

$$\nabla F(x)d_x = -F(x) + \sigma E \quad (2.8)$$

onde  $\sigma$  é um valor real positivo e  $E$  é um vetor com componentes:

$$E_i = 1, \text{ para } i \in \{1, \dots, m\} \quad (2.9)$$

$$E_i = 0, \text{ para } i \in \{m + 1, \dots, n\} \quad (2.10)$$

Claramente para  $\sigma$  positivo a direção de busca do algoritmo é viável para todo ponto  $x \in \Gamma$  já que se  $F_i(x) = 0$  verifica-se:  $\nabla F_i(x)d_x = \sigma > 0$ . Portanto existe para todo ponto  $x \in \Gamma$  um valor real positivo  $\delta_v(x)$  que verifica:

$$x + td_x \in \Gamma, \text{ para todo } t \text{ tal que } 0 \leq t \leq \delta_v(x)$$

ou seja que verifica:

$$F_i(x + td_x) \geq 0, \text{ para todo } t \text{ tal que } 0 \leq t \leq \delta_v(x) \text{ e para todo } i \in \{1, \dots, m\}$$

Sejam as seguintes suposições:

**Suposição 2.1** *A matriz  $\nabla F(x)$  não é singular em  $\Gamma$  e  $\|\nabla F(x)^{-1}\| \leq M$  para todo  $x \in \Gamma$ .*

**Suposição 2.2** *Os gradientes  $\nabla F_i$ ,  $i \in \{1, \dots, n\}$  são funções lipschitzianas com constante de Lipschitz  $k_i$ .*

**Suposição 2.3** *Existem valores  $\alpha_1 > 0$  e  $\alpha_2 > 0$  independentes de  $x$  tais que  $\sigma \in [\alpha_1 \|F(x)\|^2, \alpha_2 \|F(x)\|]$ .*

**Lema 2.1** *Se as suposições 2.1 e 2.3 verificam-se, então existe um valor  $K$  independente de  $x$  tal que para  $d_x$  calculado com a fórmula da Eq. 2.8 se verifica:*

$$\|d_x\| \leq K \|F(x)\| \quad (2.11)$$

Demonstração:

A direção de busca  $d_x$  é:

$$d_x = \nabla F(x)^{-1}(-F(x) + \sigma E)$$

portanto tem-se:

$$\begin{aligned} \|d_x\| &= \|\nabla F(x)^{-1}(-F(x) + \sigma E)\| \\ &\leq \|\nabla F(x)^{-1}\| \|(-F(x) + \sigma E)\| \\ &\leq M(\|F(x)\| + \sigma\|E\|) \\ &\leq M(1 + \alpha_2\sqrt{m})\|F(x)\| \end{aligned}$$

Chamando  $K = M(1 + \alpha_2\sqrt{m})$  tem-se a Eq. 2.11.

■

**Teorema 2.1** *Se as suposições 2.1 a 2.3 verificam-se, então para todo ponto  $x$  tal que  $\|F(x)\| > 0$  existe um valor real positivo  $\delta_v$  independente de  $x$  que verifica:*

$$(1) \quad x + td_x \in \Gamma, \text{ para todo } t \text{ tal que } 0 \leq t \leq \delta_v \quad (2.12)$$

$$(2) \quad x + td_x \in \Gamma^0, \text{ para todo } t \text{ tal que } 0 < t < \delta_v \quad (2.13)$$

Demonstração:

Para  $i \in \{1, \dots, m\}$ , por ser  $\nabla F_i$  função lipschitziana de constante  $k_i$  verifica-se:

$$F_i(x + td_x) \geq F_i(x) + t\nabla F_i(x)d_x - t^2k_i\|d_x\|^2$$

Substituindo  $\nabla F_i(x)d_x$  usando a Eq. 2.8 tem-se:

$$F_i(x + td_x) \geq (1 - t)F_i(x) + t(\sigma - tk_i\|d_x\|^2) \quad (2.14)$$

considerando a suposição 2.3 e a Eq. 2.11 tem-se:

$$F_i(x + td_x) \geq (1 - t)F_i(x) + t\|F(x)\|^2(\alpha_1 - tk_iK^2) \quad (2.15)$$

Portanto para ter  $F_i(x + td_x) \geq 0$  é suficiente que  $t$  verifique as condições:

$$t \geq 0, \quad t \leq 1 \quad \text{e} \quad t \leq \frac{\alpha_1}{k_iK^2}$$

Para ter  $F_i(x + td_x) > 0$  é suficiente que  $t$  verifique as condições:

$$t > 0, \quad t < 1 \quad \text{e} \quad t < \frac{\alpha_1}{k_i K^2}$$

Chamando  $k = \max\{k_i, 1 \leq i \leq m\}$  tem-se que um valor de  $\delta_v$  que verifica as Eqs. 2.12 e 2.13 é:

$$\delta_v = \min \left\{ 1, \frac{\alpha_1}{kK^2} \right\} \quad (2.16)$$

■

### Observações:

1) O campo de direções  $d_x$  é uniformemente viável. É assim que se denomina um campo de direções para o qual existe um passo  $\delta_v$  independente de  $x$  para o qual qualquer passo menor dá sempre um ponto na região viável.

2) O valor de  $\delta_v$  depende inversamente da constante  $k$  definida como a maior das constantes de Lipschitz das funções  $\nabla F_i$  com  $i \in \{1, \dots, m\}$ . Se essas funções  $F_i$  foram lineares,  $k$  seria zero e, portanto  $\delta_v$  seria um. Seja por exemplo  $F$  linear, então tem-se:

$$\begin{aligned} F(x + d_x) &= F(x) + \nabla F(x)d_x \\ &= F(x) - F(x) + \sigma E \\ &= \sigma E \in \Gamma \end{aligned} \quad (2.17)$$

A Eq. 2.17 mostra também que a direção  $d_x$  definida na Eq. 2.8 pode ser pensada como a direção da iteração de Newton para achar  $x$  que verifique  $F(x) = \sigma E$ .

■

Seja agora a função potencial definida como:

$$\Phi(x) = \frac{1}{2} \|F(x)\|^2 \quad (2.18)$$

A derivada da função  $\Phi$  na direção  $d_x$  é:

$$\begin{aligned} \nabla \Phi(x)d_x &= F(x)^T \nabla F_i(x)d_x \\ &= F(x)^T (-F(x) + \sigma E) \\ &= -\|F(x)\|^2 + \sigma F(x)^T E \leq -\|F(x)\|^2 + \sigma \sqrt{m} \|F(x)\| \end{aligned} \quad (2.19)$$

Então a direção  $d_x$  é de descida para o potencial  $\Phi$  se verifica-se:

$$\sigma < \frac{1}{\sqrt{m}} \|F(x)\| \quad (2.20)$$

A Eq. 2.20 motiva a seguinte suposição:

**Suposição 2.4** *Existe um valor  $\alpha_3 \in (0, 1)$  independente de  $x$  tal que se verifica:*

$$\sigma \leq \frac{\alpha_3}{\sqrt{m}} \|F(x)\| \quad (2.21)$$

**Teorema 2.2** *Se as suposições 2.1 a 2.4 verificam-se, então  $\nabla\Phi(x)d_x < 0$  e para todo  $x$  tal que  $\|F(x)\| > 0$  e para todo  $\eta \in (0, 1)$  existe  $\delta_d$  independente de  $x$  tal que verifica-se:*

$$\Phi(x + td_x) \leq \Phi(x) + t\eta\nabla\Phi(x)d_x \quad \text{para todo } t \text{ tal que } 0 \leq t \leq \delta_d \quad (2.22)$$

Demonstração:

Pela suposição 2.4 se verifica a Eq. 2.20 e portanto  $\nabla\Phi(x)d_x < 0$ . Das Eqs. 2.19 e 2.21 tem-se:

$$\nabla\Phi(x)d_x \leq -(1 - \alpha_3)\|F(x)\|^2 \quad (2.23)$$

Pela definição da função potencial  $\Phi$ , Eq. 2.18, tem-se que o gradiente  $\nabla\Phi$ , é também uma função lipschitziana em  $\mathbb{R}^n$ . Se  $k_\Phi$  é sua constante de Lipschitz tem-se:

$$\begin{aligned} \Phi(x + td_x) &\leq \Phi(x) + t\nabla\Phi(x)d_x + t^2k_\Phi\|d_x\|^2 \\ &\leq \Phi(x) + t\nabla\Phi(x)d_x + t^2k_\Phi K\|F(x)\|^2 \end{aligned} \quad (2.24)$$

Portanto, para que se verifique a Eq. 2.22 é suficiente impor:

$$\Phi(x) + t\nabla\Phi(x)d_x + t^2k_\Phi K\|F(x)\|^2 \leq \Phi(x) + t\eta\nabla\Phi(x)d_x$$

Ou seja:

$$\nabla\Phi(x)d_x \leq -\frac{tk_\Phi K}{(1 - \eta)}\|F(x)\|^2 \quad (2.25)$$

Pela Eq. 2.23 tem-se que a Eq. 2.25 se verifica se  $t$  verifica:

$$t \geq 0, \quad t \leq \frac{(1 - \alpha_3)(1 - \eta)}{k_\Phi K}$$

Portanto um valor para  $\delta_d$  é:

$$\delta_d = \frac{(1 - \alpha_3)(1 - \eta)}{k_\Phi K} \quad (2.26)$$

■

Pode-se definir agora um algoritmo para resolver o sistema de equações dado pela Eq. 2.5 e que respeita em cada iteração as desigualdades da Eq. 2.6:

### Algoritmo 2.1

Parâmetros:  $\eta, \nu \in (0, 1)$ .

Dados:  $k = 0, x_0 \in \Gamma$ .

**Passo 0:** Teste de convergência

**Passo 1:** Cálculo da direção viável

1. Escolha  $\sigma$ .
2. Ache  $d_x$  resolvendo o sistema linear da Eq. 2.8.

Se  $d_x = 0$ , fim.

**Passo 2:** Busca linear

Calcule  $t$  como o primeiro número da seqüência  $\{1, \nu, \nu^2, \nu^3, \dots\}$  que satisfaça:

$$\begin{aligned}\Phi(x_k + td_x) &\leq \Phi(x_k) + t\eta\nabla\Phi_k d_x \\ x_k + td_x &\in \Gamma\end{aligned}$$

**Passo 3:** Atualização

Defina  $x_{k+1} = x_k + td_x$

Retorne ao Passo 0.

■

A seguinte suposição é necessária para estabelecer a convergência global do Algoritmo 2.1.

**Suposição 2.5** *O conjunto  $\Gamma_0 = \{x \in \Gamma \mid \Phi(x) \leq \Phi(x_0)\}$  é um conjunto compacto em  $\mathbb{R}^n$ .*

**Teorema 2.3** *Se as suposições 2.1 a 2.5 verificam-se, então o Algoritmo 2.1 tem convergência global a pontos que satisfazem a Eq. 2.5.*

A demonstração do Teorema 2.3 resulta da aplicação direta do Teorema 7.2.3 da referência [14]. Em primeiro lugar se demonstra que existe uma aplicação ponto-conjunto fechada que define os pontos da seqüência. A mesma resulta da

composição das seguintes aplicações:

$$\begin{aligned}
A_1 : \quad x &\rightarrow (x, \sigma) && \text{Aplicação ponto-conjunto} \\
A_2 : \quad (x, \sigma) &\rightarrow (x, d_x) && \text{Aplicação ponto-ponto} \\
A_3 : \quad (x, d_x) &\rightarrow (x, d_x, t) && \text{Aplicação ponto-conjunto} \\
A_4 : \quad (x, d_x, t) &\rightarrow x + td_x && \text{Aplicação ponto-ponto}
\end{aligned}$$

A aplicação  $A_1$  define-se como:

$$A_1(x) = \left\{ (x, \sigma) \mid \sigma \in [\alpha_1 \|F(x)\|^2, \alpha_2 \|F(x)\|], \text{ e } \sigma \leq \frac{\alpha_3}{\sqrt{m}} \|F(x)\| \right\}$$

A aplicação  $A_1$  é fechada pela continuidade das funções  $F$  e  $\|\cdot\|$ .

A aplicação  $A_2$  define-se como:

$$A_2(x, \sigma) = (x, d_x) \text{ tal que } d_x = \nabla F(x)^{-1}(-F(x) + \sigma E)$$

A aplicação  $A_2$  é uma aplicação contínua pela continuidade das funções  $F$  e  $\nabla F$ , e pela continuidade da inversão (a matriz  $\nabla F(x)$  é não singular), portanto é uma aplicação fechada.

A aplicação  $A_3$  define-se como:

$$A_3(x, d_x) = \{(x, d_x, t) \mid t \geq \nu\delta, \Phi(x + td) \leq \Phi(x) + t\eta\nabla\Phi_k d_x \text{ e } x + td_x \in \Gamma\}$$

onde  $\delta = \min\{\delta_v, \delta_d\}$ . A aplicação  $A_3$  é uma aplicação fechada pela continuidade das funções  $\Phi$ ,  $\nabla\Phi$  e das funções  $F_i$  que definem o conjunto  $\Gamma$ . Claramente a busca de Armijo utilizada no Algoritmo 2.1 acha um valor para  $t$  de forma que  $(x, d_x, t) \in A_3(x, d_x)$ .

A aplicação  $A_4$  é contínua e portanto fechada.

A função de descida do algoritmo é a função  $\Phi$ , ela adota um valor menor no ponto  $x_{k+1}$  sempre que o ponto  $x_k$  não é solução. Também por isso toda a seqüência  $x_k$  está contida no conjunto compacto  $\Gamma_0$ .

Do Teorema 7.2.3 da referência [14] conclui-se:

1) Cada subsequência convergente da seqüência  $x_k$  é convergente a uma solução do problema.

2)  $\Phi(x_k) \rightarrow 0$ .

■

### Observações:

1) Dado que a seqüência decresce monotonamente, na suposição 2.1 o conjunto  $\Gamma$  pode ser substituído pelo conjunto  $\Gamma_0$  sem que isso afete o Teorema 2.3.

2) É necessário que a matriz  $\nabla F(x)$  não seja singular para  $x$  em  $\Gamma_0$  mas a condição de limitação da norma de  $\|\nabla F(x)^{-1}\|$  não é necessária se de alguma forma diferente que como foi mostrado consegue-se provar a desigualdade do Lema 2.1, esta sim desigualdade fundamental para a demonstração do Teorema 2.3. Eliminar a condição  $\|\nabla F(x)^{-1}\| \leq M$  é importante para provar a convergência do algoritmo em problemas onde a matriz  $\nabla F(x)$  é singular no conjunto solução. ■

A viabilidade uniforme da direção de busca achada pelo algoritmo permite provar a sua convergência de uma forma diferente àquela mostrada no Teorema 2.3. Pode-se provar que em cada iteração tem-se uma redução da função potencial que assegura que a mesma converge para zero com uma taxa de convergência pelo menos linear. Isto é mostrado no teorema seguinte:

**Teorema 2.4** *Se as suposições 2.1 a 2.4 verificam-se, então o Algoritmo 2.1 converge para um ponto que satisfaz a Eq. 2.5 com velocidade de convergência pelo menos linear tanto na seqüência  $\Phi_k = \Phi(x_k)$  como na seqüência  $x_k$ .*

Demonstração:

O passo  $t$  achado pelo algoritmo verifica sempre a condição de Armijo:

$$\Phi(x_{k+1}) = \Phi(x_k + td_x) \leq \Phi(x_k) + t\eta\nabla\Phi(x_k)d_x$$

Como a busca linear acha sempre um passo  $t$  que verifica  $t \geq \nu\delta$  e também  $\nabla\Phi(x_k)d_x \leq -(1 - \alpha_3)\|F(x_k)\|^2 = -\frac{1}{2}(1 - \alpha_3)\Phi(x_k)$  tem-se:

$$\Phi(x_{k+1}) \leq \left(1 - \frac{1}{2}\nu\delta\eta(1 - \alpha_3)\right) \Phi(x_k) \quad (2.27)$$

A Eq. 2.27 implica na convergência pelo menos linear da seqüência  $\Phi_k$ . Ela implica também:

$$\Phi(x_{k+1}) \leq \left(1 - \frac{1}{2}\nu\delta\eta(1 - \alpha_3)\right)^k \Phi(x_0)$$

que implica:

$$\|F(x_{k+1})\|^2 \leq \left(1 - \frac{1}{2}\nu\delta\eta(1 - \alpha_3)\right)^k \|F(x_0)\|^2$$



ou seja:

$$\|F(x_{k+1})\| \leq \left( \sqrt{1 - \frac{1}{2}\nu\delta\eta(1 - \alpha_3)} \right)^k \|F(x_0)\|$$

Pela desigualdade da Eq. 2.11 na iteração  $k$  tem-se:

$$\|d_x\| \leq \left( \sqrt{1 - \frac{1}{2}\nu\delta\eta(1 - \alpha_3)} \right)^k M\|F(x_0)\|$$

Como  $\|x_{k+1} - x_k\| = \|td_x\|$  e  $0 \leq t \leq 1$  tem-se:

$$\|x_{k+1} - x_k\| \leq \left( \sqrt{1 - \frac{1}{2}\nu\delta\eta(1 - \alpha_3)} \right)^k M\|F(x_0)\|$$

Por comparação com a seqüência geométrica é fácil mostrar que a seqüência  $x_k$  é convergente com velocidade de convergência pelo menos linear [15].

■

### Observações:

1) As observações feitas para o Teorema 2.3 são válidas também para o Teorema 2.4.

2) O Teorema 2.3 somente prova a convergência para uma solução do problema de uma subsequência da seqüência  $x_k$ , não prova a convergência da seqüência  $x_k$ . Isto sim é provado pelo Teorema 2.4 que ademais agrega um resultado de velocidade de convergência. A importância do Teorema 2.3 está em que pode ser utilizado para mostrar a convergência do Algoritmo 2.1 para uma classe mais ampla de problemas para a qual não se consiga provar a viabilidade uniforme do campo de direções  $d_x$ .

3) O Teorema 2.4 prova a convergência linear do algoritmo para a solução do problema, mas por ser baseado no método de Newton o algoritmo pode, em geral, apresentar convergência superlinear. Isto vai depender da escolha utilizada do parâmetro  $\sigma$ , ver seção 2.2.2.

■

### 2.2.1 Modificação da região viável

O Algoritmo 2.1 pode resolver eficientemente alguns problemas, porém para ser a base de um método para resolver o problema de otimização 1.1 é necessário fazer uma importante modificação. A dificuldade surge ao aplicar o algoritmo às equações

das condições de otimalidade de Karush-Kunh-Tucker. As funções  $F_i$  definidas como  $F_i(x, \lambda) = \lambda_i g_i(x)$  têm gradiente nulo nos casos em que  $\lambda_i = g_i(x) = 0$  e portanto não se verifica a suposição 2.1 nesses pontos pois a matriz  $\nabla F$  fica com uma linha nula e portanto é singular.

Uma forma de resolver o problema é modificar a região onde o algoritmo vai gerar os pontos da seqüência.

Seja a função  $Q$  definida da seguinte forma:

$$\begin{aligned} Q_i(x) &= F_i(x) - s(F(x)) \quad \text{para } i \in \{1, \dots, m\} \\ Q_i(x) &= F_i(x) \quad \text{outro caso} \end{aligned}$$

onde  $s$  é uma função que adota valores positivos para todo argumento não nulo. Se o algoritmo mantém o  $Q_i(x)$  não negativo, o valor  $F_i(x)$  será sempre positivo exceto para os pontos  $x$  soluções do problema.

Seja por exemplo a função  $s$  definida como  $s(y) = \alpha \|y\|$ . A nova região viável é a região  $\Gamma_1$  definida como:

$$\Gamma_1 = \{x \mid Q_i(x) = F_i(x) - \alpha \|F(x)\| \geq 0 \text{ para todo } i \in \{1, \dots, m\}\} \quad (2.28)$$

Os gradientes das funções  $Q_i$  são:

$$\nabla Q_i(x) = \nabla F_i(x) - \alpha \frac{F(x)^T \nabla F(x)}{\|F(x)\|}$$

As funções  $\nabla Q_i$  são também lipschitzianas em  $\mathbb{R}^n$ , sejam  $k_i$  as constantes de Lipschitz das mesmas. Então se verifica:

$$Q_i(x + td_x) \geq Q_i(x) + t \nabla Q_i(x) d_x - t^2 k_1 \|d_x\|^2$$

Substituindo na mesma forma que no Teorema 2.1 tem-se:

$$\begin{aligned} Q_i(x + td_x) &\geq (1-t)Q_i(x) + t \left( \sigma \left( 1 - \alpha \frac{F(x)^T E}{\|F(x)\|} \right) - tk_1 \|d_x\|^2 \right) \\ &\geq (1-t)Q_i(x) + t \left( \sigma(1 - \alpha \sqrt{m}) - tk_1 \|d_x\|^2 \right) \end{aligned} \quad (2.29)$$

$$\geq (1-t)Q_i(x) + t \|F(x)\|^2 \left( \alpha_1(1 - \alpha \sqrt{m}) - tk_1 K^2 \right) \quad (2.30)$$

De novo, de acordo com a Eq. 2.30, para  $\alpha < \frac{1}{\sqrt{m}}$  existe um  $\delta_v$  que verifica as Eqs. 2.12 e 2.13 (esta vez em relação à região  $\Gamma_1$ ), o mesmo é:

$$\delta_v = \min \left\{ 1, \frac{\alpha_1(1 - \alpha \sqrt{m})}{k_1 K^2} \right\}$$

O algoritmo que define uma seqüência na região  $\Gamma_1$  é:

## Algoritmo 2.2

Parâmetros:  $\alpha \in (0, \frac{1}{\sqrt{m}})$ ,  $\eta, \nu \in (0, 1)$ .

Dados:  $k = 0$ ,  $x_0 \in \Gamma_1$ .

**Passo 0:** Teste de convergência

**Passo 1:** Cálculo da direção viável

1. Escolha  $\sigma$ .
2. Ache  $d_x$  resolvendo o sistema linear da Eq. 2.8.

Se  $d_x = 0$ , fim.

**Passo 2:** Busca linear

Calcule  $t$  como o primeiro número da seqüência  $\{1, \nu, \nu^2, \nu^3, \dots\}$  que satisfaça:

$$\begin{aligned}\Phi(x_k + td_x) &\leq \Phi(x_k) + t\eta\nabla\Phi_k d_x \\ x_k + td_x &\in \Gamma_1\end{aligned}$$

**Passo 3:** Atualização

Defina  $x_{k+1} = x_k + td_x$

Retorne ao Passo 0.

■

### 2.2.2 Escolha do parâmetro $\sigma$

Fica ainda definir uma regra prática para a escolha de  $\sigma$  de forma que existam os parâmetros  $\alpha_1$ ,  $\alpha_2$  e  $\alpha_3$  que verifiquem as suposições 2.3 e 2.4. A escolha do parâmetro  $\sigma$  vai ser fundamental para obter um algoritmo eficiente e de convergência superlinear para a solução do problema.

Da Eq. 2.14 tem-se:

$$F_i(x + td_x) \geq (1 - t)F_i(x) + t(\sigma - tk_i K^2 \|F(x)\|^2)$$

Portanto, um valor de  $\sigma$  suficientemente alto daria para  $t = 1$  um ponto na região viável. Valores menores de  $\sigma$  poderiam obrigar ao algoritmo a dar passos menores. O desafio é escolher  $\sigma$  menor em cada iteração para obter uma direção o mais

parecida possível com a direção de Newton, mas, grande o suficiente para obter passos unitários.

A escolha de  $\sigma$  apresentada nesta seção está baseada na aplicação do método de Newton num sistema de equações não linear equivalente ao sistema de equações dado pela Eq. 2.5. Com isto espera-se obter um algoritmo que tenha convergência superlinear para problemas que verifiquem as hipóteses habituais do método de Newton.

Seja o seguinte sistema não linear de equações:

$$W(F(x)) = 0 \quad (2.31)$$

onde a função  $W : \mathbb{R}^n \rightarrow \mathbb{R}^n$ .

O sistema de equações 2.31 é equivalente ao sistema de equações 2.5 se a função  $W$  cumpre:

$$W(y) = 0 \leftrightarrow y = 0$$

A direção da iteração de Newton para o sistema de equações dado pela Eq. 2.31 resulta de resolver o sistema linear:

$$\nabla W(F(x)) \nabla F(x) d_x = -W(F(x)) \quad (2.32)$$

Seja por exemplo uma função  $W$  da forma:

$$W(y) = y + w(y)E \quad (2.33)$$

onde a função  $w : \mathbb{R}^n \rightarrow \mathbb{R}$  e o vetor  $E \in \mathbb{R}^n$  é o vetor definido pelas Eq. 2.9 e 2.10. O gradiente de  $W$  é:

$$\nabla W(y) = I + E \nabla w(y)$$

onde a matriz  $I$  é a matriz identidade. A matriz inversa de  $\nabla W(y)$  pode-se achar facilmente aplicando a regra de Sherman-Morrison [8]:

$$\nabla W(y)^{-1} = I - \frac{E \nabla w(y)}{1 + \nabla w(y)E} \quad (2.34)$$

Substituindo 2.33 e 2.34 em 2.32 tem-se:

$$\nabla F(x) d_x = -F(x) + \frac{-w(F(x)) + \nabla w(F(x))F(x)}{1 + \nabla w(F(x))E} E \quad (2.35)$$

Seja por exemplo  $w(y) = \beta\|y\|^2$  com  $\beta$  constante. A Eq. 2.35 fica:

$$\nabla F(x)d_x = -F(x) + \frac{\beta\|F(x)\|^2}{1 + 2\beta E^T F(x)} E \quad (2.36)$$

A Eq. 2.36 tem a mesma forma da Eq. 2.8. O valor de  $\sigma$  resulta por este método:

$$\sigma = \frac{\beta}{1 + 2\beta E^T F(x)} \|F(x)\|^2 \quad (2.37)$$

**Teorema 2.5** *Seja  $\beta$  um parâmetro fixo no intervalo  $(0, 1/(\sqrt{m}\|F(x_0)\|))$ . Para a escolha de  $\sigma$  dado pela Eq. 2.37 se verificam as suposições 2.3 e 2.4.*

Demonstração:

O valor de  $\sigma$  dado pela Eq. 2.37 verifica:

$$\begin{aligned} \sigma &= \frac{\beta}{1 + 2\beta E^T F(x)} \|F(x)\|^2 \\ &\geq \frac{\beta}{1 + 2\beta\sqrt{m}\|F(x)\|} \|F(x)\|^2 \\ &\geq \frac{\beta}{1 + 2\beta\sqrt{m}\|F(x_0)\|} \|F(x)\|^2 \end{aligned}$$

e também verifica:

$$\begin{aligned} \sigma &= \frac{\beta}{1 + 2\beta E^T F(x)} \|F(x)\|^2 \\ &\leq \beta\|F(x)\|^2 \\ &\leq \beta\|F(x_0)\|\|F(x)\| \end{aligned}$$

O que significa que se satisfazem as suposições 2.3 e 2.4 para valores de  $\alpha_1$ ,  $\alpha_2$  e  $\alpha_3$  iguais a:

$$\begin{aligned} \alpha_1 &= \frac{\beta}{1 + 2\beta\sqrt{m}\|F(x_0)\|} \\ \alpha_2 &= \beta\|F(x_0)\| \\ \alpha_3 &= \sqrt{m}\beta\|F(x_0)\| \in (0, 1) \end{aligned}$$

■

A restrição de escolher  $\beta$  menor de  $1/(\sqrt{m}\|F(x_0)\|)$  é para garantir a existência de  $\alpha_3$  no intervalo  $(0, 1)$  e com isso ter uma direção  $d_x$  de descida para a função potencial  $\Phi$ . Outro potencial daria uma restrição diferente ou nenhuma restrição no valor de  $\beta$ .

O valor de  $\beta$  influi de forma fundamental no desempenho do algoritmo. Valores baixos de  $\beta$  podem produzir valores do passo pequenos e assim prejudicar a convergência do algoritmo. Valores altos podem aumentar o valor do passo mas produzir uma convergência lenta para a solução devido a que a direção não fica próxima da direção de Newton.

Neste trabalho optou-se por utilizar um valor de  $\beta$  variável escolhido da seguinte forma: Primeiro escolhe-se um valor para  $\alpha_3 \in (0, 1)$  e logo se define  $\beta = \alpha_3/(\sqrt{m}\|F(x_0)\|)$ . Isso impede que o valor de  $\beta$  seja grande demais. Logo, cada vez que o passo  $t = 1$  dá um ponto fora da região viável,  $\beta$  é redefinido na mesma forma utilizando os valores atuais, ou seja, define-se:  $\beta = \alpha_3/(\sqrt{m}\|F(x_k)\|)$ .

A motivação para essa escolha é que se foi obtido um valor do passo menor a um então seria conveniente aumentar  $\beta$  para aumentar o valor de  $\sigma$ .

Essa escolha de  $\beta$ , que é redefinido cada vez que o passo unitário não dá um ponto na região viável, traz a questão da validade das suposições 2.3 e 2.4. Dado que  $\|F(x)\|$  decresce em cada iteração se obtém uma seqüência crescente de  $\beta$ 's, mas que sempre cumpre  $\beta\|F(x)\| \leq \alpha_3/\sqrt{m}$ . Definindo  $\beta_0 = \alpha_3/(\sqrt{m}\|F(x_0)\|)$  tem-se:

$$\begin{aligned} \sigma &= \frac{\beta}{1 + 2\beta E^T F(x)} \|F(x)\|^2 \\ &\geq \frac{\beta}{1 + 2\beta\sqrt{m}\|F(x)\|} \|F(x)\|^2 \\ &\geq \frac{\beta}{1 + 2\alpha_3} \|F(x)\|^2 \end{aligned} \tag{2.38}$$

$$\geq \frac{\beta_0}{1 + 2\alpha_3} \|F(x)\|^2 \tag{2.39}$$

Também tanto para as iterações em que  $\beta$  é redefinido ou para as outras tem-se que  $\beta\|F(x)\| \leq \alpha_3/\sqrt{m}$  e, portanto tem-se:

$$\begin{aligned} \sigma &= \frac{\beta}{1 + 2\beta E^T F(x)} \|F(x)\|^2 \\ &\leq \beta\|F(x)\|^2 \\ &\leq \frac{\alpha_3}{\sqrt{m}} \|F(x)\| \end{aligned} \tag{2.40}$$

As desigualdades obtidas acima, Eqs. 2.39 e 2.40, permitem concluir que o algoritmo com  $\beta$  variável cumpre com as suposições 2.3 e 2.4, portanto pelo Teorema 2.3 o Algoritmo 2.2 converge a uma solução do problema. O Teorema seguinte mostra que na verdade  $\beta$  será redefinido somente um número finito de vezes.

**Teorema 2.6** *Para o Algoritmo 2.2 com  $\beta$  variável,  $\beta$  somente será redefinido um número finito de vezes.*

Demonstração:

Suponha-se que a tese do teorema não é verdadeira. Então o valor do  $\beta$  vai ser redefinido infinitas vezes com a fórmula  $\beta = \alpha_3/(\sqrt{m}\|F(x)\|)$ , mas como o algoritmo converge a uma solução o valor de  $\beta$  tende a infinito. A Eq. 2.29 era:

$$Q_i(x + td_x) \geq (1 - t)Q_i(x) + t \left( \sigma(1 - \alpha\sqrt{m}) - tk_1\|d_x\|^2 \right) \quad (2.41)$$

Substituindo  $\|d_x\|$  e o valor de  $\sigma$  pela Eq. 2.38 tem-se:

$$Q_i(x + td_x) \geq (1 - t)Q_i(x) + t\|F(x)\|^2 \left( \frac{\beta(1 - \alpha\sqrt{m})}{1 + 2\alpha_3} - tk_1K^2 \right) \quad (2.42)$$

Portanto o ponto vai estar na região viável se o  $0 \leq t \leq \delta_v$  onde  $\delta_v$  é:

$$\delta_v = \min \left\{ 1, \frac{\beta(1 - \alpha\sqrt{m})}{k_1K^2(1 + 2\alpha_3)} \right\}$$

Como o segundo termo tende a infinito,  $\delta_v$  não pode ser menor que 1 em infinitas iterações, portanto a tese do teorema não pode ser falsa. ■

### Observação:

1) Logo que o algoritmo redefine  $\beta$  pela última vez ele se reduz ao algoritmo de Newton com busca na função potencial para o sistema de equações dado pela Eq. 2.31. Nos casos nos que na solução a matriz  $\nabla F(x)$  não seja singular, o algoritmo convergirá para a solução com velocidade de convergência quadrática. ■

O seguinte exemplo mostra o funcionamento do Algoritmo 2.1 com  $\beta$  variável. O ponto inicial é  $x = (2, 6)$  e a função  $F$  é definida como:

$$\begin{aligned} F_1(x_1, x_2) &= 2x_1^2 - 1.12 - x_2 \\ F_2(x_1, x_2) &= x_2 - (x_1 - 0.4)^2 + 0.8 \end{aligned}$$

O Algoritmo 2.1 com  $\beta$  variável para  $\alpha_3 = 0.2$  e  $\nu = 0.9$  produz a saída mostrada na Tabela 2.1. Para  $\alpha_3 = 0.02$  a saída do Algoritmo 2.1 produz a saída mostrada na Tabela 2.2.

Tabela 2.1: Saída do Algoritmo 2.1,  $\alpha_3 = 0.2$ .

Iter	IterBL	$t$	$\sigma/\ F(x)\ $	$\ F(x_k)\ $
0	-	-	-	4.3303e+000
1	2	0.900	2.000e-001	1.7534e+000
2	2	0.900	2.000e-001	7.5349e-001
3	1	1.000	2.000e-001	2.1029e-001
4	4	0.729	6.315e-002	7.5113e-002
5	1	1.000	2.000e-001	1.2025e-002
6	1	1.000	2.725e-002	4.4164e-004
7	1	1.000	1.177e-003	5.9588e-007
8	1	1.000	1.538e-006	1.1061e-012

Tabela 2.2: Saída do Algoritmo 2.1,  $\alpha_3 = 0.02$ .

Iter	IterBL	$t$	$\sigma/\ F(x)\ $	$\ F(x_k)\ $
0	-	-	-	4.3303e+000
1	3	0.810	2.000e-002	1.6476e+000
2	8	0.478	2.000e-002	1.0084e+000
3	17	0.185	2.000e-002	8.3372e-001
4	22	0.109	2.000e-002	7.4612e-001
...	...	...	...	...
20	13	0.282	2.000e-002	7.5436e-002
21	10	0.387	2.000e-002	4.7076e-002
22	6	0.590	2.000e-002	2.0097e-002
23	1	1.000	2.000e-002	4.9840e-004
24	1	1.000	4.591e-004	3.6165e-007
25	1	1.000	3.593e-007	1.7380e-013

Nas Tabelas 2.1 e 2.2 “Iter” indica a iteração e “IterBL” indica o número de iterações de Armijo nessa iteração.

A tolerância utilizada no critério de parada foi de  $10^{-10}$  na função potencial. No primeiro caso o algoritmo faz 8 iterações,  $\beta$  é redefinido pela última vez na iteração



4, a partir da iteração 5 o algoritmo faz passos unitários com redução importante no parâmetro  $\sigma$  e da função potencial.

No segundo caso o valor  $\alpha_3 = 0.02$  escolhido mostra ser muito pequeno para as características do problema, o algoritmo faz 25 iterações,  $\beta$  é redefinido pela última vez na iteração 22 e somente a partir da iteração 23 o algoritmo faz passos unitários com redução importante do parâmetro  $\sigma$  e da função potencial.

As Figuras 2.1 e 2.2 mostram o percurso do algoritmo para os dois casos.

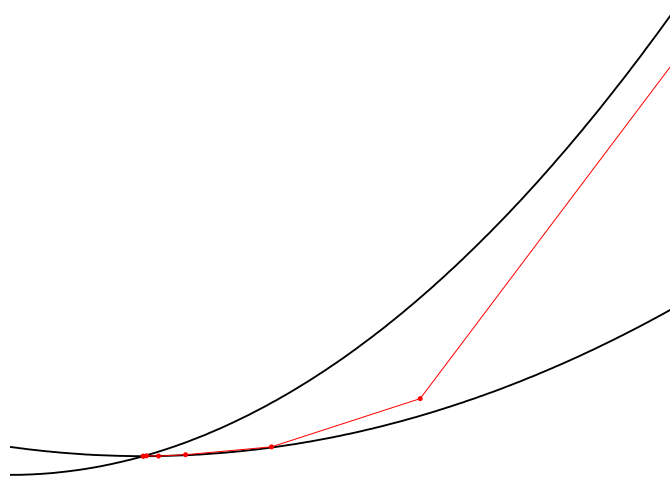


Figura 2.1: Percurso do Algoritmo 2.1,  $\alpha_3 = 0.2$ .

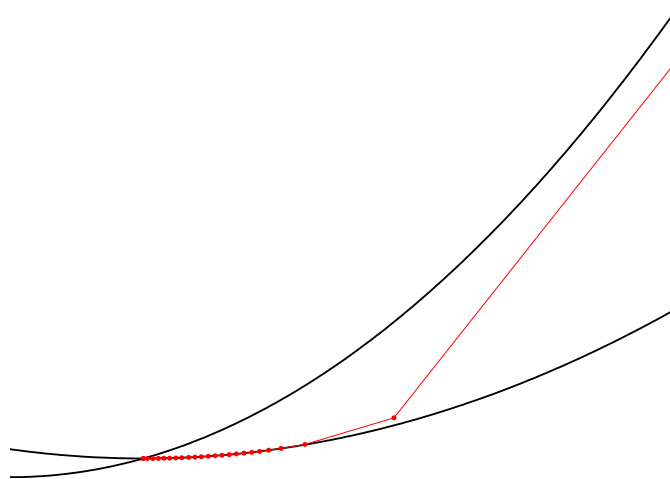


Figura 2.2: Percurso do Algoritmo 2.1,  $\alpha_3 = 0.02$ .

### 2.2.3 Correção de segunda ordem

A direção de busca do Algoritmo 2.1 é baseada na direção da iteração de Newton para o sistema não linear de equações dado pela Eq. 2.5. A mesma resulta de utilizar a aproximação linear da função  $F$  no ponto da iteração atual. É possível melhorar a direção da iteração de Newton tendo em conta alguma informação disponível da diferença entre essa aproximação linear e a função  $F$ .

Seja  $F(x) = F(x_k) + \nabla F(x_k)(x - x_k) + o(x - x_k)$ . Aqui  $o(x - x_k)$  representa a diferença entre a função  $F$  e sua aproximação linear no ponto  $x_k$ . Chamando  $d_x = x - x_k$  tem-se:

$$F(x_k + d_x) = F_k + \nabla F_k d_x + o(d_x) \quad (2.43)$$

Resolver o problema dado pela Eq. 2.5 é equivalente a resolver o problema não linear dado pela seguinte equação na variável  $d_x$ :

$$F_k + \nabla F_k d_x + o(d_x) = 0 \quad (2.44)$$

O problema anterior é equivalente ao problema de encontrar  $d_x$  que verifique  $d_x = -\nabla F_k^{-1}(F_k + o(d_x))$ . Para resolver este problema pode-se aplicar o seguinte processo iterativo:

$$d_x^{m+1} = -\nabla F_k^{-1}(F_k + o(d_x^m)) \quad (2.45)$$

Por último, substituindo  $o(d_x^m)$  utilizando a Eq. 2.43 tem-se:

$$d_x^{m+1} = d_x^m - \nabla F_k^{-1}(F(x_k + d_x^m)) \quad (2.46)$$

#### Observações:

1) No caso em que a seqüência converge, tem-se que  $F(x + d_x^\infty) = 0$  onde aqui chamou-se  $d_x^\infty$  ao limite da seqüência  $d_x^m$ .

2) A iteração da Eq. 2.46 pode ser vista como a iteração de um método baseado na iteração de Newton porém que utiliza sempre a matriz de derivadas no ponto  $x_k$ . Chamando  $x_k^m = x_k + d_x^m$  e somando  $x_k$  em ambos os lados da Eq. 2.46 tem-se:

$$x_k^{m+1} = x_k^m - \nabla F_k^{-1} F(x_k^m)$$

3) Mesmo que a seqüência  $x_k^m$  converja para a solução, pode-se preferir parar com  $m$  pequeno e utilizar  $d_x^m$  como direção de busca do algoritmo principal para definir  $x_{k+1}$ .

4) O custo da iteração 2.46 pode ser pequeno se previamente são achados os fatores da decomposição LU da matriz  $\nabla F_k$ . Se a fatorização da matriz é muito custosa pode ser conveniente fazer mais iterações em  $d_x^m$  pois a melhora na direção de busca pode reduzir o número de iterações do programa principal e com isso reduzir o número de fatorizações da matriz de derivadas.

■

Por exemplo, partindo com  $d_x^0 = 0$  e fazendo duas iterações na direção de busca tem-se as direções:

$$\begin{aligned} d_x^1 &= -\nabla F_k^{-1} F(x_k) \\ d_x^2 &= d_x^1 - \nabla F_k^{-1} (F(x_k + d_x^1)) \end{aligned}$$

Com isso pode-se fazer a busca linear na curva  $x(t) = x_k + td_x^2$ . Outro procedimento possível é fazer a busca linear no arco  $x(t) = x_k + td_x^1 + t^2(d_x^2 - d_x^1)$ . A busca em arco tem a vantagem de ter  $x'(0) = d_x^1$  e, portanto mantém a boa propriedade de partir com a mesma direção da iteração de Newton.

Voltando agora ao problema original, uma forma que permite aplicar a correção de segunda ordem é pensando em encontrar um ponto que verifique  $F(x) = \sigma E$ . Neste caso a iteração na direção de busca fica:

$$d_x^{m+1} = d_x^m - \nabla F_k^{-1} (F(x_k + d_x^m) - \sigma E) \quad (2.47)$$

Por exemplo, partindo com  $d_x^0 = 0$  tem-se as direções:

$$d_x^1 = -\nabla F_k^{-1} (F(x_k) - \sigma E) \quad (2.48)$$

$$d_x^2 = d_x^1 - \nabla F_k^{-1} (F(x_k + d_x^1) - \sigma E) \quad (2.49)$$

Neste caso  $d_x^1$  coincide com a direção  $d_x$  definida na Eq. 2.8. De novo pode-se fazer a busca linear na curva  $x(t) = x_k + td_x^2$  ou fazer a busca linear no arco  $x(t) = x_k + td_x^1 + t^2(d_x^2 - d_x^1)$ . A busca em arco tem a propriedade de ter  $x'(0) = d_x^1$  e, portanto o arco vai ser ao mesmo tempo de descida para a função potencial e viável.

Para o exemplo mostrado anteriormente o algoritmo com busca em arco converge em seis e cinco iterações para  $\alpha_3 = 0.2$  e  $\alpha_3 = 0.02$  respectivamente. A Figura 2.3 mostra o percurso do Algoritmo 2.1 com busca em arco para  $\alpha_3 = 0.02$ .

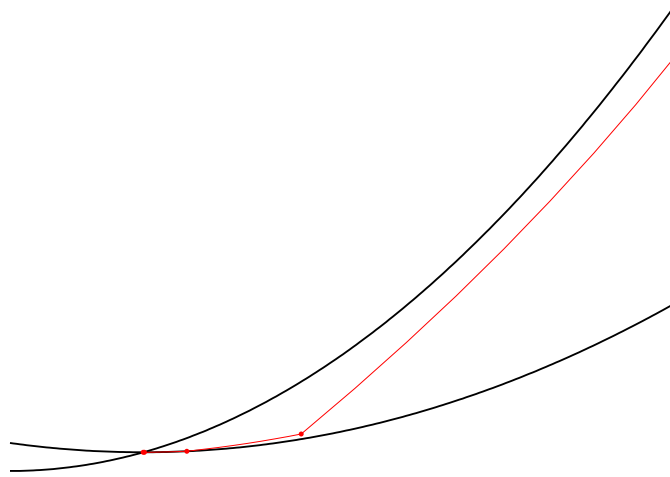


Figura 2.3: Percurso do Algoritmo 2.1 com busca em arco,  $\alpha_3 = 0.02$ .

Como se pode ver na Figura 2.3 a busca em arco ajuda o algoritmo a evitar as restrições. Neste caso o algoritmo converge em cinco iterações em vez das 25 que o algoritmo fazia sem a correção de segunda ordem.

### 3 Métodos viáveis para problemas de otimização

Neste capítulo descrevem-se alguns métodos de otimização baseados no Algoritmo 2.2 descrito no capítulo anterior com a escolha de  $\sigma$  dado pela Eq. 2.37 com  $\beta$  variável.

A aplicação direta do Algoritmo 2.2 no sistema não linear de equações dado pelas igualdades das condições de otimalidade de KKT é adequada para resolver problemas de otimização convexos. Este método é descrito na seção 3.1. Na seção 3.2 é descrito um método para problemas de otimização gerais. Nessa seção são descritas também algumas técnicas adequadas para melhorar o desempenho do algoritmo, como a correção de segunda ordem e simetrização e redução dos sistemas lineares. Na seção 3.3 descreve-se um método proposto para resolver problemas de otimização degenerados.

#### 3.1 Métodos viáveis para problemas convexos

Seja por exemplo o problema de otimização não linear 1.1 sem restrições de igualdade onde a função objetivo  $f$  é convexa da mesma forma que as opostas das funções que definem as restrições, ou seja,  $-g_i$  é convexa com  $1 \leq i \leq m$ .

Definindo a variável  $z = (x, \lambda)$  pode-se aplicar em forma direta o Algoritmo 2.2 no problema de achar um ponto que satisfaça as condições de otimalidade de KKT.

A função  $F : \mathbb{R}^{n+m} \rightarrow \mathbb{R}^{n+m}$  define-se neste caso como:

$$F(z) = (\nabla L(x, \lambda), \Lambda g(x))$$

A região  $\Gamma_1$  deve ser definida neste caso como:

$$\Gamma_1 = \{z = (x, \lambda) \mid x \in \Omega, \lambda \geq 0, \text{ e } \Lambda g(x) - \alpha \|\nabla L(x, \lambda), \Lambda g(x)\| \geq 0\} \quad (3.1)$$

A função potencial é neste caso:

$$\Phi(z) = \frac{1}{2} \|\nabla L(x, \lambda), \Lambda g(x)\|^2$$

Isto leva ao seguinte algoritmo:

### Algoritmo 3.1

Parâmetros:  $\alpha \in (0, \frac{1}{\sqrt{m}})$ ,  $\eta, \nu \in (0, 1)$ .

Dados:  $k = 0$ ,  $z_0 = (x_0, \lambda_0)$  onde  $z_0 \in \Gamma_1$ .

**Passo 0:** Teste de convergência

**Passo 1:** Cálculo da direção viável

1. Escolha  $\sigma$ .
2. Ache  $d_z = (d_x, d_\lambda)$  resolvendo o sistema linear da Eq. 3.2 onde neste caso  $E$  é um vetor de componentes unitárias.

$$\begin{pmatrix} H_k & -\nabla g_k^T \\ \Lambda_k \nabla g_k & G_k \end{pmatrix} \begin{pmatrix} d_x \\ d_\lambda \end{pmatrix} = \begin{pmatrix} -\nabla L_k^T \\ -\Lambda_k g_k + \sigma E \end{pmatrix} \quad (3.2)$$

Se  $d_x = 0$ , fim.

**Passo 2:** Busca linear

Calcule  $t$  como o primeiro número da seqüência  $\{1, \nu, \nu^2, \nu^3, \dots\}$  que satisfaça:

$$\begin{aligned} \Phi(z_k + tdz) &\leq \Phi(z_k) + t\eta \nabla \Phi_k d_z \\ z_k + tdz &\in \Gamma_1 \end{aligned}$$

**Passo 3:** Atualização

Defina  $z_{k+1} = z_k + tdz$

Retorne ao Passo 0.

■

A única diferença do Algoritmo 3.1 com o Algoritmo 2.2 é a definição da região viável. A definição dada pela Eq. 2.28 não coincide exatamente com a dada pela Eq. 3.1 pois à segunda se agregaram as restrições  $x \in \Omega$  e  $\lambda \geq 0$ . Porém, é fácil provar que o valor de  $\delta_v$  para a região viável do Algoritmo 3.1 coincide com o valor obtido anteriormente. De fato chamando  $x(t) = x + td_x$  e  $\lambda(t) = \lambda + td_\lambda$  pela Eq. 2.13 do Teorema 2.1 tem-se:

$$0 < t < \delta_v \rightarrow (\lambda_i(t))g_i(x(t)) - \alpha \|(\nabla L(x(t), \lambda(t)), (\Lambda(t)g(x(t))))\| > 0$$

Se por exemplo existir  $t_1$ , com  $0 < t_1 < \delta_v$ , tal que  $g_i(x(t_1)) \leq 0$  por continuidade da função  $g_i(x(t))$  existiria um valor  $t_2$ , com  $0 < t_2 \leq t_1$  tal que  $g_i(x(t_2)) = 0$  e portanto se deveria cumprir  $-\alpha\|(\nabla L(x(t_2), \lambda(t_2)), \Lambda(t_2)g(x(t_2)))\| > 0$  o qual não é possível. Pelo mesmo argumento prova-se que  $\lambda(t) > 0$  para  $0 \leq t < \delta_v$ .

O algoritmo 3.1 pode ser muito eficiente na prática para resolver muitos problemas de otimização mas em muitos problemas a matriz do sistema linear 3.2 pode ser singular.

Sejam as seguintes suposições:

**Suposição 3.1** *O ponto  $x_k$  satisfaz a condição de regularidade LICQ.*

**Suposição 3.2** *A matriz  $H(x_k, \lambda_k)$  é definida positiva quando restringida ao espaço tangente à região viável  $\Omega$  no ponto  $x_k$ .*

**Suposição 3.3** *Para  $i \in \{1, \dots, p\}$  tem-se  $g_i(x_k) + \lambda_i > 0$ , ou seja, a restrição  $g_i(x)$  e o multiplicador de Lagrange correspondente não se anulam simultaneamente.*

**Observação:**

1) Devido ao problema ser convexo a matriz  $H(x_k, \lambda_k)$  com  $\lambda_k$  não negativo é pelo menos semidefinida positiva, já que as matrizes  $\nabla^2 f(x)$  e  $-\nabla^2 g_i(x)$ ,  $1 \leq i \leq m$  assim são. Nos casos em que a função  $f$  seja fortemente convexa a matriz  $\nabla^2 f(x)$  será definida positiva e, portanto verifica-se a suposição 3.2.

■

**Teorema 3.1** *Se as suposições 3.1 a 3.3 verificam-se, então a matriz do sistema linear 3.2 é não singular.*

Demonstração:

Seja o seguinte sistema linear de equações:

$$\begin{pmatrix} H_k & -\nabla g_k^T \\ \Lambda_k \nabla g_k & G_k \end{pmatrix} \begin{pmatrix} d_x \\ d_\lambda \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad (3.3)$$

A matriz do sistema é não singular se e somente se o sistema da Eq. 3.3 tem como solução única os vetores  $d_x = 0$  e  $d_\lambda = 0$ . Sem perda de generalidade suponha-se que  $g_k = (g_k^a, g_k^b, g_k^c)$  e  $\lambda_k = (\lambda_k^a, \lambda_k^b, \lambda_k^c)$ , onde o sub-índice  $a$  indica que tanto  $g$  como

$\lambda$  são positivos, o sub-índice  $b$  indica que somente  $\lambda$  é positivo e o sub-índice  $c$  indica que somente  $g$  é positivo. Definindo  $d_\lambda = (d_\lambda^a, d_\lambda^b, d_\lambda^c)$  o sistema da Eq. 3.3 é:

$$\begin{pmatrix} H_k & -\nabla g_k^{aT} & -\nabla g_k^{bT} & -\nabla g_k^{cT} \\ \Lambda_k^a \nabla g_k^a & G_k^a & 0 & 0 \\ \Lambda_k^b \nabla g_k^b & 0 & 0 & 0 \\ 0 & 0 & 0 & G_k^c \end{pmatrix} \begin{pmatrix} d_x \\ d_\lambda^a \\ d_\lambda^b \\ d_\lambda^c \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad (3.4)$$

Como  $g_k^c > 0$ , da quarta linha da Eq. 3.4 se deduz que  $d_\lambda^c = 0$ . Como  $\lambda_k^b > 0$  da terceira linha se deduz que  $\nabla g_k^b d_x = 0$  e, portanto  $d_x$  pertence ao espaço tangente à região viável  $\Omega$ . Definindo  $S_k^a = (\Lambda_k^a)^{-1} G_k^a$  tem-se que as três primeiras linhas do sistema 3.4 são equivalentes ao sistema:

$$\begin{pmatrix} H_k & -\nabla g_k^{aT} & -\nabla g_k^{bT} \\ \nabla g_k^a & S_k^a & 0 \\ \nabla g_k^b & 0 & 0 \end{pmatrix} \begin{pmatrix} d_x \\ d_\lambda^a \\ d_\lambda^b \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \quad (3.5)$$

Multiplicando pela esquerda pelo vetor  $(d_x^T, d_\lambda^{aT}, d_\lambda^{bT})$  tem-se:

$$d_x^T H_k d_x + d_\lambda^{aT} S_k^a d_\lambda^a = 0$$

Como no espaço tangente a matriz  $H_k$  é definida positiva e a matriz  $S_k^a$  também é definida positiva tem-se que  $d_x = 0$  e  $d_\lambda^a = 0$ . Por último da primeira linha do sistema 3.5 tem-se que  $\nabla g_k^{bT} d_\lambda^b = 0$  e devido à independência linear dos gradientes das restrições ativas se deduz que  $d_\lambda^b = 0$ . ■

Para o algoritmo 2.2 não é totalmente necessário que a matriz  $H_k$  seja definida positiva no espaço tangente. Pela definição do conjunto  $\Gamma_1$  tanto  $g(x)$  como  $\lambda$  são positivos fora do conjunto solução e portanto chamando  $S_k = \Lambda_k^{-1} G_k$  o sistema da Eq. 3.3 é equivalente ao sistema:

$$\begin{pmatrix} H_k & -\nabla g_k^T \\ \nabla g_k & S_k \end{pmatrix} \begin{pmatrix} d_x \\ d_\lambda \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad (3.6)$$

Da segunda linha se deduz que  $d_\lambda = -S_k^{-1} \nabla g_k d_x$  e substituindo na primeira linha tem-se:

$$(H_k + \nabla g_k^T S_k^{-1} \nabla g_k) d_x = 0 \quad (3.7)$$



Portanto para obter  $d_x = 0$  e com isso obter  $d_\lambda = 0$  é suficiente que a matriz  $H_k + \nabla g_k^T S_k^{-1} \nabla g_k$  seja não singular. Como o Algoritmo 3.1 mantém positiva a diagonal da matriz  $S_k$ , é suficiente que a matriz  $\nabla g_k$  tenha posto maior ou igual a  $m$  para que a matriz do sistema linear da Eq. 3.7 seja definida positiva.

### 3.2 Métodos viáveis para problemas gerais

Nos problemas de otimização convexos a matriz  $H_k$  é definida positiva na região viável do problema e, portanto a matriz dos sistemas lineares do Algoritmo 3.1 é não singular. Em problemas mais gerais o Algoritmo 3.1 pode não funcionar. Uma solução para este problema é a de modificar a matriz  $H_k$  do sistema 3.2 substituindo-a por uma matriz definida positiva  $B_k$ . Neste caso a direção de busca em vez de ser calculada resolvendo o sistema 3.2 é achada resolvendo o seguinte sistema:

$$\begin{pmatrix} B_k & -\nabla g_k^T \\ \Lambda_k \nabla g_k & G_k \end{pmatrix} \begin{pmatrix} d_x \\ d_\lambda \end{pmatrix} = \begin{pmatrix} -\nabla L_k^T \\ -\Lambda_k g_k + \sigma E \end{pmatrix} \quad (3.8)$$

Na mesma forma que no algoritmo FAIPA, a partir do sistema 3.8 é possível definir um algoritmo de direções viáveis. A direção  $d_x$  é a soma da direção da iteração de Newton mais uma componente de desvio, ou seja,  $d_z = d_z^a + \sigma d_z^b$  onde  $d_z^a = (d_x^a, d_\lambda^a)$  e  $d_z^b = (d_x^b, d_\lambda^b)$  calculam-se a partir do sistema:

$$\begin{pmatrix} B_k & -\nabla g_k^T \\ \Lambda_k \nabla g_k & G_k \end{pmatrix} \begin{pmatrix} d_x^a & d_x^b \\ d_\lambda^a & d_\lambda^b \end{pmatrix} = \begin{pmatrix} -\nabla L_k^T & 0 \\ -\Lambda_k g_k & E \end{pmatrix} \quad (3.9)$$

É fácil ver que a direção  $d_x^a$  é de descida para a função objetivo  $f$ . Da segunda linha do sistema linear 3.9 pode-se expressar  $d_\lambda^a$  em função de  $d_x^a$ , e substituir na primeira linha para obter:

$$\left( B_k + \nabla g_k^T S_k^{-1} \nabla g_k \right) d_x^a = -\nabla f_k^T \quad (3.10)$$

Chamando  $M$  à matriz definida positiva  $B_k + \nabla g_k^T S_k^{-1} \nabla g_k$  tem-se:

$$\begin{aligned} d_x^a &= -M^{-1} \nabla f_k^T \\ \nabla f_k d_x^a &= -\nabla f_k M^{-1} \nabla f_k^T < 0 \end{aligned}$$

A última desigualdade é devido a que a matriz  $M^{-1}$  também é definida positiva.

Logo para que a direção  $d_x$  seja de descida para a função  $f$  é suficiente pedir que se verifique:

$$\nabla f_k d_x \leq \gamma \nabla f_k d_x^a \quad (3.11)$$

onde  $\gamma$  é uma constante no intervalo  $(0, 1)$ . A desigualdade dada pela Eq. 3.11 acrescenta uma outra restrição na escolha de  $\sigma$ . De fato tem-se:

$$\begin{aligned} \nabla f_k (d_x^a + \sigma d_x^b) &\leq \gamma \nabla f_k d_x^a \\ (\nabla f_k d_x^b) \sigma &\leq (\gamma - 1) \nabla f_k d_x^a \end{aligned}$$

Quando  $\nabla f_k d_x^b$  for positivo  $\sigma$  deve ser escolhido de forma que se verifique:

$$\sigma \leq \frac{(\gamma - 1) \nabla f_k d_x^a}{\nabla f_k d_x^b} \quad (3.12)$$

O algoritmo que utiliza a matriz definida positiva  $B_k$  e como função potencial à função  $f$  é:

### Algoritmo 3.2

Parâmetros:  $\alpha \in (0, \frac{1}{\sqrt{m}})$ ,  $\eta, \nu \in (0, 1)$ .

Dados:  $k = 0$ ,  $z_0 = (x_0, \lambda_0)$  onde  $z_0 \in \Gamma_1$ .

**Passo 0:** Teste de convergência

**Passo 1:** Cálculo da direção viável

1. Ache  $d_z^a = (d_x^a, d_\lambda^a)$  e  $d_z^b = (d_x^b, d_\lambda^b)$  resolvendo os sistemas lineares da Eq. 3.13.

$$\begin{pmatrix} B_k & -\nabla g_k^T \\ \Lambda_k \nabla g_k & G_k \end{pmatrix} \begin{pmatrix} d_x^a & d_\lambda^a \\ d_x^b & d_\lambda^b \end{pmatrix} = \begin{pmatrix} -\nabla L_k^T & 0 \\ -\Lambda_k g_k & E \end{pmatrix} \quad (3.13)$$

Se  $d_x = 0$ , fim.

2. Escolha  $\sigma$ .

3. Defina:

$$\begin{aligned} d_x &= d_x^a + \sigma d_x^b \\ d_\lambda &= d_\lambda^a + \sigma d_\lambda^b \end{aligned}$$

**Passo 2:** Busca linear

Calcule  $t$  como o primeiro número da seqüência  $\{1, \nu, \nu^2, \nu^3, \dots\}$  que satisfaça:

$$\begin{aligned} f(x_k + tdx) &\leq f(x_k) + t\eta\nabla f_k dx \\ z_k + tdz &\in \Gamma_1 \end{aligned}$$

**Passo 3:** Atualização

Defina  $z_{k+1} = z_k + tdz$

Retorne ao Passo 0.

■

**Observações:**

1) O Algoritmo 3.2 tem a vantagem de fazer a busca linear na função objetivo. Desta forma é bem mais provável obter convergência para um ponto de mínimo relativo e não a um ponto de máximo relativo ou ponto de sela do problema.

2) Em diversas situações foi observado que o Algoritmo 3.2 não converge para uma solução do problema. Algumas das causas foram:

- A restrição na escolha de  $\sigma$  dada pela Eq. 3.12 em alguns casos obrigou o algoritmo a definir um valor de  $\sigma$  muito pequeno. Nesses problemas obteve-se convergência do algoritmo para pontos na fronteira da região viável que não são solução do problema. Os exemplos que apresentaram esse problema não verificavam as condições de regularidade LICQ, provavelmente para problemas que verificam as condições de regularidade o Algoritmo 3.2 não falhe devido à restrição na escolha de  $\sigma$  dada pela Eq. 3.12, mas isto ainda não foi provado.
- Em alguns problemas obteve-se divergência da matriz dos sistemas lineares. Em algumas situações pela divergência da matriz  $B_k$  (que foi definida de acordo a regra de atualização BFGS, ver seção 3.2.1) e em outros problemas pela divergência da seqüência de multiplicadores de Lagrange,  $\lambda_k$ . Dado que a função  $f$  depende somente da variável primal,  $x$ , o algoritmo não foi capaz de evitar a divergência a infinito de alguns dos multiplicadores de Lagrange. Uma solução para este problema pode ser a de impor um limite máximo no valor

dos multiplicadores de Lagrange como feito na referência [7]. Na referência [16] é mostrado como modificar a regra de atualização dos multiplicadores de Lagrange de forma de melhorar a estabilidade de um algoritmo similar ao Algoritmo 3.2.

■

### 3.2.1 Atualização quase-Newton

Os métodos quase-Newton compõem uma classe de métodos que utilizam somente informação de primeira ordem (valores das funções e de suas derivadas) para definir uma regra de atualização da matriz  $B_k$ . Tendo em conta as variações dos gradientes das funções, eles constroem um modelo quadrático do problema que usualmente é bom o suficiente para produzir convergência superlinear para a solução [8].

O método quase-Newton mais popular é o conhecido com o nome BFGS. Para problemas sem restrições a atualização BFGS da matriz  $B_k$  é [17]:

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k} \quad (3.14)$$

onde  $s_k = x_{k+1} - x_k$  e  $y_k = \nabla f_{k+1} - \nabla f_k$ .

Para problemas com restrições a fórmula dada pela Eq. 3.14 pode ser utilizada definindo  $y_k = \nabla L(x_{k+1}, \lambda_{k+1}, \mu_{k+1}) - \nabla L(x_k, \lambda_{k+1}, \mu_{k+1})$ .

É possível mostrar que a fórmula dada pela Eq. 3.14 produz uma matriz  $B_{k+1}$  definida positiva sempre que seja definida positiva a matriz  $B_k$  e se verifique também a condição:

$$s_k^T y_k > 0 \quad (3.15)$$

Para problemas sem restrições a Eq. 3.15 se verifica se a função objetivo é fortemente convexa ou na busca linear são requeridas as condições de parada de Wolfe [8]. Quando o problema tem restrições ou ainda sem restrições a função objetivo não é convexa e é feita uma busca linear que não impõe as condições de Wolfe, uma possível solução é a de utilizar a atualização BFGS amortecida, a mesma define o vetor  $r_k = \theta_k y_k + (1 - \theta_k) B_k s_k$ , onde o escalar  $\theta_k$  é definido como:

$$\theta_k = \begin{cases} 1 & \text{se } s_k^T y_k \geq 0.2 s_k^T B_k s_k \\ 0.8 (s_k^T B_k s_k) / (s_k^T B_k s_k - s_k^T y_k) & \text{se } s_k^T y_k < 0.2 s_k^T B_k s_k \end{cases}$$

Logo a atualização de  $B_k$  é feita pela fórmula:

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{r_k r_k^T}{r_k^T s_k} \quad (3.16)$$

Os métodos de otimização que utilizam a atualização quase-Newton têm a grande vantagem de não requerer informação de segunda ordem (derivadas segundas) das funções que definem o problema. Em alguns casos, como por exemplo, em problemas de engenharia onde o cômputo das funções e suas derivadas já pode ter um custo computacional muito importante, estes métodos podem ser significativamente mais eficientes que os métodos que utilizam informação de segunda ordem.

Uma desvantagem dos métodos quase-Newton é que as matrizes  $B_k$  geradas são em geral densas, isto acontece também em problemas onde a matriz hessiana seja esparsa. A necessidade de memória para armazenar a matriz e o custo computacional alto da resolução de sistemas lineares com matrizes de coeficientes densas fazem que os métodos quase-Newton sejam convenientes somente para problemas de otimização de tamanho pequeno [5, 8].

Em problemas com e sem restrições o custo computacional da resolução do sistema linear pode ser significativamente reduzido atualizando em vez da matriz quase-Newton os seus fatores de Cholesky. Para reduzir o custo de armazenamento, uma das técnicas mais utilizadas é a técnica BFGS de memória limitada. Esta técnica permite fazer algumas operações que envolvem a matriz quase-Newton armazenando somente algumas matrizes de pequeno tamanho. A técnica BFGS de memória limitada permite também reduzir o custo computacional da resolução dos sistemas lineares em problemas com restrições. Para a resolução eficiente dos sistemas lineares por meio de métodos diretos uma técnica que utiliza a técnica BFGS de memória limitada é apresentada na referência [18]. Para a resolução por meio de métodos iterativos uma técnica eficiente é apresentada na referência [19].

Outros métodos quase-Newton diferentes ao método BFGS podem ser mais convenientes em alguns problemas. Nocedal [8] menciona que em problemas com restrições onde a matriz hessiana seja indefinida a técnica BFGS amortecida pode apresentar dificuldades, sendo talvez melhor para estes problemas o método de atualização quase-Newton de posto 1 ou método SR1. A dificuldade da aplicação

deste método é devido a que em geral não gera matrizes definidas positivas pelo qual deve ser combinado com alguma técnica de modificação da matriz para ser utilizada por métodos de pontos interiores.

Com o objetivo de utilizar o potencial dos resolvidores para sistemas lineares esparsos, vêm sendo estudadas técnicas de atualização esparsa da matriz  $B_k$ . Lamentavelmente é conhecido que os requisitos de esparsidade e positividade entram em conflito com a estabilidade [20], por isso estas técnicas vêm sendo pouco estudadas. Alguns resultados preliminares são apresentados na referência [21].

Uma outra desvantagem dos métodos quase-Newton é que em geral necessitam de mais iterações para atingir uma mesma tolerância que os métodos que utilizam informação de segunda ordem. Por exemplo, num problema de otimização sem restrições com a função objetivo quadrática, o desempenho do método é na teoria similar que o método do gradiente conjugado. Se o problema é suficientemente mal condicionado o método do gradiente conjugado requer um número de iterações da ordem do número de variáveis do problema. Isto é inviável quando o número de variáveis é grande.

Em problemas com restrições de desigualdade aparece um outro problema que pode fazer aumentar muito o número de iterações para obter a solução. O mesmo parece ser causado por uma tendência da seqüência de pontos que o método gera de ficar próximo da fronteira da região viável do problema. Já nos métodos que utilizam informação de segunda ordem é conhecido que é conveniente que as iterações mantenham-se afastadas da fronteira da região viável, pois, para pontos que estão perto da fronteira a direção da iteração de Newton não produz em geral um progresso significativo em direção da solução do problema. A razão dessa tendência pode ser explicada pela utilização da matriz quase-Newton. Seja por exemplo um problema sem restrições de igualdade e as funções  $e_1(x, \lambda) = \frac{1}{2} \|\nabla L(x, \lambda)\|^2$  e  $e_2(x, \lambda) = \frac{1}{2} \|\Lambda g(x)\|^2$ . As derivadas das funções  $e_1(x, \lambda)$  e  $e_2(x, \lambda)$  na direção  $d_z^1 = (d_x^1, d_\lambda^1)$  do Algoritmo 2.1 são:

$$\begin{aligned} \frac{\partial e_1}{\partial d_z^1}(x_k, \lambda_k) &= \nabla_x e_1(x_k, \lambda_k) d_x^1 + \nabla_\lambda e_1(x_k, \lambda_k) d_\lambda^1 \\ &= \nabla L_k(H_k d_x^1 + \nabla g_k^T d_\lambda^1) \\ &= \nabla L_k(H_k - B_k) d_x^1 + \nabla L_k(B_k d_x^1 + \nabla g_k^T d_\lambda^1) \end{aligned}$$

$$= \nabla L_k(H_k - B_k)d_x^1 - \|\nabla L_k\|^2 \quad (3.17)$$

$$\begin{aligned} \frac{\partial e_2}{\partial d_z^1}(x_k, \lambda_k) &= \nabla_x e_2(x_k, \lambda_k)d_x^1 + \nabla_\lambda e_2(x_k, \lambda_k)d_\lambda^1 \\ &= g_k \Lambda_k (\Lambda_k \nabla g_k d_x^1 + G_k d_\lambda^1) \\ &= -\|\Lambda_k g_x\|^2 \end{aligned} \quad (3.18)$$

Da Eq. 3.18 observa-se que a função  $e_2$  tende a diminuir na direção  $d_z^1$ , pelo contrario a função  $e_1$  pode crescer nessa direção devido ao término  $\nabla L_k(H_k - B_k)d_x^1$  que pode ser positivo.

Nos exemplos resolvidos e apresentados no Capítulo 4, freqüentemente foi observado que a função  $e_2$  é reduzida rapidamente enquanto a função  $e_1$  permanece sem redução importante até as últimas iterações. Esse fato indica que a seqüência de pontos gerada seguiu um caminho próximo da fronteira da região viável.

Este problema que os métodos quase-Newton apresentam parece ser o mais difícil de resolver. Sem dúvida é um importante tema de estudo já que freqüentemente é apontado que o sucesso dos métodos de ponto interior foi devido à definição de uma iteração não linear que se mantém sempre afastada da fronteira da região viável [5].

### 3.2.2 Correção de segunda ordem

De acordo com a seção 2.2.3 ao Algoritmo 3.2 pode ser adicionada uma correção de segunda ordem para melhorar o desempenho do mesmo. Logo de achar  $\sigma$ ,  $d_x$  e  $d_\lambda$  no passo 1 do Algoritmo 3.2, de acordo com a Eq. 2.49 a direção para definir o arco de busca é achada resolvendo o seguinte sistema linear:

$$\begin{pmatrix} B_k & -\nabla g_k^T \\ \Lambda_k \nabla g_k & G_k \end{pmatrix} \begin{pmatrix} d_x^c \\ d_\lambda^c \end{pmatrix} = \begin{pmatrix} -\nabla L(x_k + d_x, \lambda_k + d_\lambda)^T \\ -(\Lambda_k + d_\Lambda)g(x_k + d_x) \end{pmatrix} \quad (3.19)$$

Em geral a direção para a busca em arco dada pela Eq. 3.19 não deu bons resultados, nos problemas resolvidos obteve-se passos pequenos nas primeiras iterações do algoritmo. Uma direção mais conveniente é achada resolvendo o seguinte sistema linear:

$$\begin{pmatrix} B_k & -\nabla g_k^T \\ \Lambda_k \nabla g_k & G_k \end{pmatrix} \begin{pmatrix} d_x^c \\ d_\lambda^c \end{pmatrix} = \begin{pmatrix} 0 \\ -(\Lambda_k + d_\Lambda)g(x_k + d_x) \end{pmatrix} \quad (3.20)$$

A direção dada pela Eq. 3.20 mostrou na maioria dos casos uma redução do número de iterações do algoritmo. A mesma pode ser interpretada como a aplicação da correção de segunda ordem somente nas funções que definem as restrições de desigualdade. Nesse aspecto a direção definida pela Eq. 3.20 cumpre com o objetivo de facilitar ao algoritmo o progresso para a solução permitindo passos maiores sem antes atingir a fronteira da região viável.

Com a direção dada pela Eq. 3.20 a busca linear é feita no arco:

$$\begin{aligned}x(t) &= x_k + td_x + t^2 d_x^c \\ \lambda(t) &= \lambda_k + td_\lambda + t^2 d_\lambda^c\end{aligned}$$

### 3.2.3 Simetrização e redução do sistema linear

Seja por exemplo o sistema linear do Algoritmo 3.2 com algumas restrições de tipo faixa, ou seja, restrições do tipo  $x_i^{min} \leq x_i \leq x_i^{max}$ , e algumas restrições do tipo  $g_i^{min} \leq g_i(x) \leq g_i^{max}$ . Para simplificar, sem perda de generalidade será considerado um problema com restrições de desigualdade dadas pelas funções  $g_1(x) = g(x) - g_{min}$ ,  $g_2(x) = g_{max} - g(x)$  e  $g_3(x) = x$ . A análise não é diferente quando as restrições de faixa são da forma mais geral, envolvem somente algumas das variáveis ou existem também outras restrições.

Os sistemas lineares que devem ser resolvidos pelo algoritmo são do tipo:

$$\begin{pmatrix} B & -\nabla g_1^T & -\nabla g_2^T & -\nabla g_3^T \\ \Lambda_1 \nabla g_1 & G_1 & 0 & 0 \\ \Lambda_2 \nabla g_2 & 0 & G_2 & 0 \\ \Lambda_3 \nabla g_3 & 0 & 0 & G_3 \end{pmatrix} \begin{pmatrix} d_x \\ d_\lambda^1 \\ d_\lambda^2 \\ d_\lambda^3 \end{pmatrix} = \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} \quad (3.21)$$

onde  $\nabla g_1 = -\nabla g_2$  e  $\nabla g_3 = I$ .

Definindo  $S_i = \Lambda_i^{-1} G_i$  e negando a primeira linha tem-se:

$$\begin{pmatrix} -B & \nabla g_1^T & \nabla g_2^T & \nabla g_3^T \\ \nabla g_1 & S_1 & 0 & 0 \\ \nabla g_2 & 0 & S_2 & 0 \\ \nabla g_3 & 0 & 0 & S_3 \end{pmatrix} \begin{pmatrix} d_x \\ d_\lambda^1 \\ d_\lambda^2 \\ d_\lambda^3 \end{pmatrix} = \begin{pmatrix} -a \\ b' \\ c' \\ d' \end{pmatrix} \quad (3.22)$$

onde  $b' = \Lambda_1^{-1} b$ ,  $c' = \Lambda_2^{-1} c$  e  $d' = \Lambda_3^{-1} d$ . O sistema da Eq. 3.22 tem a vantagem de ter a matriz de coeficientes simétrica.



Da quarta linha tem-se:

$$d_\lambda^3 = S_3^{-1}d' - S_3^{-1}\nabla g_3 d_x = G_3^{-1}d - S_3^{-1}d_x \quad (3.23)$$

Substituindo nas outras linhas tem-se:

$$\begin{pmatrix} -B' & \nabla g_1^T & \nabla g_2^T \\ \nabla g_1 & S_1 & 0 \\ \nabla g_2 & 0 & S_2 \end{pmatrix} \begin{pmatrix} d_x \\ d_\lambda^1 \\ d_\lambda^2 \end{pmatrix} = \begin{pmatrix} -a' \\ b' \\ c' \end{pmatrix} \quad (3.24)$$

onde  $B' = B + \nabla g_3^T S_3^{-1} \nabla g_3 = B + S_3^{-1}$  e  $a' = a + \nabla g_3^T S_3^{-1} d' = a + G_3^{-1} d$ .

Dado que a matriz de gradientes  $\nabla g_2$  é a oposta da matriz  $\nabla g_1$  é possível fazer uma outra redução no sistema linear dado pela Eq. 3.24. Definindo as incógnitas  $d_\lambda^+ = d_\lambda^1 + d_\lambda^2$  e  $d_\lambda^- = d_\lambda^1 - d_\lambda^2$  tem-se:

$$d_\lambda^1 = \frac{1}{2}(d_\lambda^+ + d_\lambda^-) \quad (3.25)$$

$$d_\lambda^2 = \frac{1}{2}(d_\lambda^+ - d_\lambda^-) \quad (3.26)$$

Substituindo  $d_\lambda^1$  e  $d_\lambda^2$  na Eq. 3.24 pelas expressões anteriores tem-se:

$$\begin{pmatrix} -B' & \nabla g_1^T & 0 \\ \nabla g_1 & \frac{1}{2}S_1 & \frac{1}{2}S_1 \\ \nabla g_2 & -\frac{1}{2}S_2 & \frac{1}{2}S_2 \end{pmatrix} \begin{pmatrix} d_x \\ d_\lambda^- \\ d_\lambda^+ \end{pmatrix} = \begin{pmatrix} -a' \\ b' \\ c' \end{pmatrix} \quad (3.27)$$

Somando a segunda linha á terceira tem-se:

$$\begin{pmatrix} -B' & \nabla g_1^T & 0 \\ \nabla g_1 & \frac{1}{2}S_1 & \frac{1}{2}S_1 \\ 0 & \frac{1}{2}(S_1 - S_2) & \frac{1}{2}(S_1 + S_2) \end{pmatrix} \begin{pmatrix} d_x \\ d_\lambda^- \\ d_\lambda^+ \end{pmatrix} = \begin{pmatrix} -a' \\ b' \\ c' + b' \end{pmatrix} \quad (3.28)$$

Da terceira linha tem-se:

$$d_\lambda^+ = 2(S_1 + S_2)^{-1}(c' + b') - (S_1 + S_2)^{-1}(S_1 - S_2)d_\lambda^- \quad (3.29)$$

Substituindo nas outras linhas tem-se:

$$\begin{pmatrix} -B' & \nabla g_1^T \\ \nabla g_1 & S_1' \end{pmatrix} \begin{pmatrix} d_x \\ d_\lambda^- \end{pmatrix} = \begin{pmatrix} -a' \\ b'' \end{pmatrix} \quad (3.30)$$

onde  $S_1' = (S_1 + S_2)^{-1}S_1S_2$  e  $b'' = (S_1 + S_2)^{-1}(S_2b' - S_1c')$ . O sistema dado pela Eq. 3.30 é bem menor que o sistema original dado pela Eq. 3.22. Além disso, ele também tem matriz de coeficientes simétrica.

Logo de resolver o sistema linear da Eq. 3.30 para achar as direções  $d_x$  e  $d_\lambda^-$ ;  $d\lambda^+$ ,  $d_\lambda^1$ ,  $d_\lambda^2$  e  $d_\lambda^3$  são achados pelas fórmulas das Eq. 3.29, 3.25, 3.26 e 3.23 respectivamente.

O procedimento de resolver o sistema linear dado pela Eq. 3.21 resolvendo primeiro o sistema linear dado pela Eq. 3.30 mostrou-se eficaz e eficiente. Em nenhum problema testado foram percebidos problemas de instabilidade numérica do procedimento descrito.

### 3.2.4 Problemas com restrições de igualdade

Quando o problema de otimização é convexo e tem restrições de igualdade lineares, o Algoritmo 3.1 pode ser utilizado quase sem modificações. Na matriz dos sistemas lineares devem ser agregadas as linhas correspondentes as igualdades ( $\nabla h_k d_x = -h_k$ ) e na função potencial é agregado o termo  $\frac{1}{2}\|h(x)\|^2$ . Quando o problema de otimização não é convexo ou as funções que definem as restrições de igualdade são não lineares é talvez melhor utilizar o Algoritmo 3.2 e agregar uma penalidade na função potencial.

O sistema linear que deve ser resolvido neste caso é:

$$\begin{pmatrix} B_k & -\nabla g_k^T & -\nabla h_k^T \\ \Lambda_k \nabla g_k & G_k & 0 \\ \nabla h_k & 0 & 0 \end{pmatrix} \begin{pmatrix} d_x^a & d_x^b \\ d_\lambda^a & d_\lambda^b \\ d_\mu^a & d_\mu^b \end{pmatrix} = \begin{pmatrix} -\nabla L_k^T & 0 \\ -\Lambda_k g_k & E \\ -h_k & 0 \end{pmatrix} \quad (3.31)$$

A função potencial utilizada neste trabalho é a função  $\Phi : \mathbb{R}^n \rightarrow \mathbb{R}$  definida como:

$$\Phi(x) = f(x) + \frac{1}{2}\varphi\|h(x)\|^2 \quad (3.32)$$

Para poder utilizar o Algoritmo 3.2 a direção  $d_x^a$  deveria ser de descida para a função potencial  $\Phi$ , isto é mostrado no seguinte teorema:

**Teorema 3.2** *Para  $\varphi$  suficientemente grande a direção  $d_x^a$  é de descida para a função potencial  $\Phi$ .*

Demonstração:

Utilizando a segunda linha do sistema para eliminar  $d_\lambda^a$  e substituindo na primeira linha tem-se:

$$\left( B_k + \nabla g_k^T S_k^{-1} \nabla g_k \right) d_x^a - \nabla h_k^T d_\mu = -\nabla f_k^T + \nabla h_k^T \mu$$

Chamando  $M$  à matriz definida positiva  $B_k + \nabla g_k^T S_k^{-1} \nabla g_k$  e multiplicando pela esquerda pelo vetor  $d_x^a$  tem-se:

$$d_x^{aT} M d_x^a - d_x^{aT} \nabla h_k^T d_\mu = -\nabla f_k d_x^a + d_x^{aT} \nabla h_k^T \mu \quad (3.33)$$

Da terceira linha do sistema linear tem-se:

$$\nabla h_k d_x^a = -h_k \quad (3.34)$$

A derivada da função  $\Phi$  na direção  $d_x^a$  é:

$$\nabla \Phi_k d_x^a = \nabla f_k d_x^a + \varphi h_k^T \nabla h_k d_x^a$$

Utilizando a Eq. 3.34 tem-se

$$\nabla \Phi_k d_x^a = \nabla f_k d_x^a + \varphi \|h(x)\|^2 \quad (3.35)$$

A demonstração divide-se em dois casos:

1.  $\|h_k\| = 0$ , portanto  $h_k = 0$ , da Eq. 3.34 tem-se  $\nabla h_k d_x^a = 0$  e portanto da Eq. 3.33 tem-se:

$$\begin{aligned} d_x^{aT} M d_x^a &= -\nabla f_k d_x^a \\ &= -\nabla \Phi_k d_x^a \end{aligned}$$

e portanto tem-se:

$$\nabla \Phi_k d_x^a = -d_x^{aT} M d_x^a < 0$$

2.  $\|h_k\| > 0$ , então, pela Eq. 3.35 a direção  $d_x^a$  é de descida para  $\varphi$  maior a  $\frac{-\nabla f_k d_x^a}{\|h\|^2}$ .

■

### 3.3 Método viável para problemas degenerados

Neste trabalho entende-se por problema degenerado aquele problema de otimização que na solução tem alguma restrição de desigualdade ativa com o multiplicador de Lagrange correspondente nulo, ou seja, se  $x^*$  é solução do problema e  $(\lambda^*, \mu^*)$  são os multiplicadores de Lagrange que verificam as condições de otimalidade de KKT, existe  $i \in \{1, \dots, m\}$  tal que  $i \in A(x^*)$  e  $\lambda_i^* = 0$ .

Em problemas degenerados a matriz do sistema 3.2 é singular na solução do problema (uma ou mais linhas se anulam) e além disso não é possível obter desigualdades como mostrada na Eq. 2.11, tornando difícil a aplicação de um algoritmo que gere pontos dentro da região viável  $\Gamma_1$  e consiga ao mesmo tempo uma velocidade de convergência superlinear.

A maioria dos métodos propostos que apresentam velocidade de convergência superlinear neste tipo de problemas são baseados na identificação das restrições que na solução são ativas e apresentam multiplicador de Lagrange nulo [22, 23]. Mesmo que teoricamente apresentam convergência superlinear é apontado que a identificação é certa somente quando o ponto da iteração atual está muito próximo da solução do problema o que limita a importância prática deste tipo de algoritmos.

Nesta seção descreve-se um método baseado na aplicação da função de complementaridade de Fischer-Burmeister nas equações de complementaridade das condições de KKT. A aplicação da função de complementaridade de Fischer-Burmeister pode ser útil para estabelecer um método que não necessariamente tenha que resolver em cada iteração sistemas lineares com matrizes cada vez pior condicionadas, pois existem casos onde a matriz do sistema não fica singular na solução do problema. Essa propriedade das matrizes do método seria desde já uma vantagem na hora de resolver os sistemas lineares, pois não teriam as dificuldades numéricas com as quais se deparam os sistemas lineares com número de condição alto. Em alguns exemplos não foi necessário utilizar resolvidores específicos para o sistema linear de equações, inclusive num dos exemplos do Capítulo 4 foi possível utilizar com sucesso um resolvidor iterativo.

Além do potencial ganho computacional de resolver sistemas lineares melhor condicionados existe um aspecto que é mais importante para estabelecer um método de pontos interiores e que está relacionado a desigualdade 2.11. De fato, é maior a classe de problemas onde uma desigualdade desse tipo é válida.

A aplicação da função de complementaridade de Fischer-Burmeister permite também a formulação de um método que não tenha que respeitar a região viável  $\Gamma_1$  e que provavelmente em alguns problemas apresente melhores propriedades de convergência local que os métodos de pontos interiores. Este método pode ser

utilizado para incrementar a eficiência de outros métodos já que pode ser adicionado na fase final, por exemplo, de um método de pontos interiores para conseguir uma maior velocidade de convergência e assim reduzir o número total de iterações.

### 3.3.1 Função de Fischer-Burmeister

A função de Fischer-Burmeister é a função  $\phi : \mathbb{R}^2 \rightarrow \mathbb{R}$  definida como [24]:

$$\psi(a, b) = a + b - \sqrt{a^2 + b^2} \quad (3.36)$$

As derivadas parciais de  $\psi$  são:

$$\frac{\partial \psi}{\partial a}(a, b) = 1 - \frac{a}{\sqrt{a^2 + b^2}} \quad (3.37)$$

$$\frac{\partial \psi}{\partial b}(a, b) = 1 - \frac{b}{\sqrt{a^2 + b^2}} \quad (3.38)$$

No ponto  $(a, b) = (0, 0)$  a função de Fischer-Burmeister não é diferenciável, porém nesse ponto existe o sub-diferencial que é o conjunto:

$$\partial \psi = \{(v, w) \mid (v - 1)^2 + (w - 1)^2 \leq 1\}$$

### 3.3.2 Método viável para problemas de otimização degenerados

A equação de complementaridade das condições de KKT, Eq. 1.3, não aparece diretamente como propriedade fundamental dos pontos de mínimo relativo do problema de otimização 1.1. De fato a definição dos multiplicadores de Lagrange para as restrições não ativas é somente necessária para uma formulação matemática conveniente. Para esse fim a Eq. 1.3 cumpre com a necessidade de expressar numa fórmula matemática que a restrição  $g_i(x)$  é ativa nesse ponto ou seu multiplicador de Lagrange correspondente é nulo.

Outra possível forma de expressar a complementaridade entre a restrição e seu multiplicador de Lagrange, e que, além disso, inclui as restrições de positividade, é por meio da função de complementaridade de Fischer-Burmeister. A função de Fischer-Burmeister tem a seguinte e importante propriedade:

$$\psi(a, b) = 0 \leftrightarrow \begin{array}{l} a = 0 \text{ e } b \geq 0 \text{ ou} \\ a \geq 0 \text{ e } b = 0 \end{array} \quad (3.39)$$

Essa propriedade da função de Fischer-Burmeister permite escrever as condições de otimalidade de KKT na seguinte forma equivalente:

$$L(x, \lambda, \mu) = 0 \quad (3.40)$$

$$\Psi(g(x), \lambda) = 0 \quad (3.41)$$

$$h(x) = 0 \quad (3.42)$$

onde  $\Psi : \mathbb{R}^{2n} \rightarrow \mathbb{R}^n$  é tal que  $\Psi_i(g(x), \lambda) = \psi(g_i(x), \lambda_i)$ .

O método que se descreve nesta seção é baseado na iteração de Newton aplicada ao sistema de equações não lineares dado pelas Eqs. 3.40 a 3.42.

Seja por exemplo o problema de otimização 1.1 sem restrições de igualdade. A direção da iteração de Newton para as equações 1.2 a 1.4 neste caso é:

$$\begin{pmatrix} H_k & -\nabla g_k^T \\ \Lambda_k^\Psi \nabla g_k & G_k^\Psi \end{pmatrix} \begin{pmatrix} d_x \\ d_\lambda \end{pmatrix} = \begin{pmatrix} -\nabla L_k^T \\ -\Psi_k \end{pmatrix} \quad (3.43)$$

onde  $\Lambda_k^\Psi$  e  $G_k^\Psi$  são matrizes diagonais com componentes:

$$\begin{aligned} \Lambda^\Psi(x, \lambda)(i, i) &= \lambda_i^\Psi = 1 - \frac{g_i(x)}{\sqrt{g_i(x)^2 + \lambda_i^2}} \\ G^\Psi(x, \lambda)(i, i) &= g_i^\Psi = 1 - \frac{\lambda_i}{\sqrt{g_i(x)^2 + \lambda_i^2}} \end{aligned}$$

A diferença deste sistema linear com os dos outros algoritmos descritos é que as matrizes  $\Lambda^\Psi$  e  $G^\Psi$  não tem linhas que se anulem simultaneamente pois  $(\lambda_i^\Psi - 1)^2 + (g_i^\Psi - 1)^2 = 1$ .

A matriz do sistema linear dado pela Eq. 3.43 é não singular se a matriz  $H_k$  é definida positiva (se não fosse sempre existe a possibilidade de substituir ou modificar a matriz  $H_k$  para torná-la definida positiva). Esta propriedade sugere o estudo de um método baseado neste sistema linear que não tenha que respeitar em cada iteração as restrições que definem a região viável. Isto pode ser conveniente para problemas onde é singular a matriz dos sistemas lineares do Algoritmo 3.2. Na referência [25], descreve-se um algoritmo similar ao Algoritmo 3.2 mas que modifica a matriz dos sistemas lineares para não ter que respeitar a região viável em cada iteração. É indicado pelos autores que tal procedimento é conveniente para problemas onde na solução a matriz é singular. A matriz do sistema dado pela Eq. 3.43 consegue o

objetivo de ser não singular e ao mesmo tempo deriva da aplicação do método de Newton a um sistema não linear de equações.

O algoritmo que utiliza a função de Fischer-Burmeister nas equações de complementaridade é:

**Algoritmo 3.3**

Parâmetros:  $\alpha \in (0, \frac{1}{\sqrt{m}})$ ,  $\eta, \nu \in (0, 1)$ .

Dados:  $k = 0$ ,  $z_0 = (x_0, \lambda_0)$  onde  $z_0 \in \Gamma_1$ .

**Passo 0:** Teste de convergência

**Passo 1:** Cálculo da direção viável

1. Escolha  $\sigma$ .
2. Ache  $d_z = (d_x, d_\lambda)$  resolvendo o sistema linear da Eq. 3.44 onde neste caso  $E$  é um vetor de componentes unitárias.

$$\begin{pmatrix} H_k & -\nabla g_k^T \\ \Lambda_k^\Psi \nabla g_k & G_k^\Psi \end{pmatrix} \begin{pmatrix} d_x \\ d_\lambda \end{pmatrix} = \begin{pmatrix} -\nabla L_k^T \\ -\Psi_k + \sigma E \end{pmatrix} \quad (3.44)$$

Se  $d_x = 0$ , fim.

**Passo 2:** Busca linear

Calcule  $t$  como o primeiro número da seqüência  $\{1, \nu, \nu^2, \nu^3, \dots\}$  que satisfaça:

$$\begin{aligned} \Phi(z_k + tdz) &\leq \Phi(z_k) + t\eta \nabla \Phi_k d_z \\ z_k + tdz &\in \Gamma_1 \end{aligned}$$

**Passo 3:** Atualização

Defina  $z_{k+1} = z_k + tdz$

Retorne ao Passo 0.

■

Seja por exemplo o problema:

$$\begin{aligned} \text{minimize} \quad & f(x_1, x_2) = \frac{1}{2} ((x_1 + 1)^2 + x_2^2) \\ \text{sujeito a:} \quad & x_1 \geq 0 \\ & x_2 \geq 0 \end{aligned} \quad (3.45)$$

O exemplo tem a solução  $(x_1, x_2) = (0, 0)$  (ambas as restrições são ativas), com multiplicadores  $(\lambda_1, \lambda_2) = (1, 0)$ .

Para uma tolerância de  $10^{-9}$  nas magnitudes  $\|L_k\|$  e  $\|\Lambda_k g_k\|$  os Algoritmos 3.2 e 3.3 fazem 18 e 12 iterações respectivamente. O primeiro chega ao ponto  $(x_1, x_2) = (4.79 \times 10^{-18}, 1.91 \times 10^{-5})$  em quanto o segundo chega ao ponto  $(x_1, x_2) = (6.20 \times 10^{-10}, 1.06 \times 10^{-9})$ .

Na figura 3.1 mostram-se os percursos dos Algoritmos 3.2 e 3.3, na figura 3.2 mostra-se o número de condição das matrizes dos sistemas lineares em função do número de iteração.

Além de achar uma solução mais precisa e em um número de iterações menor, na figura 3.2 pode-se ver que os sistemas lineares do Algoritmo 3.3 tem matrizes de coeficientes com número de condição abaixo de 300, em quanto para o Algoritmo 3.2 o número de condição cresce em cada iteração (na solução a matriz é singular).

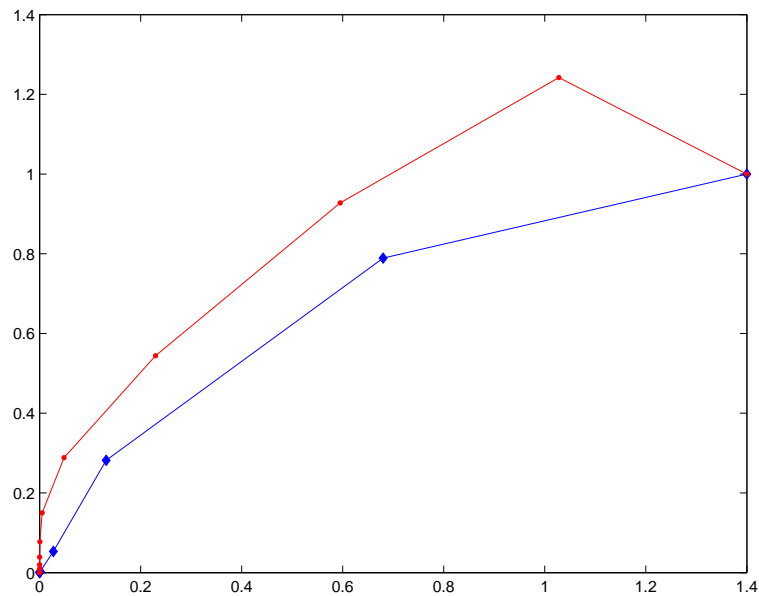


Figura 3.1: Percurso dos Algoritmo 3.2 e 3.3. Nós circulares: Algoritmo 3.2, nós em forma de diamante: Algoritmo 3.3.



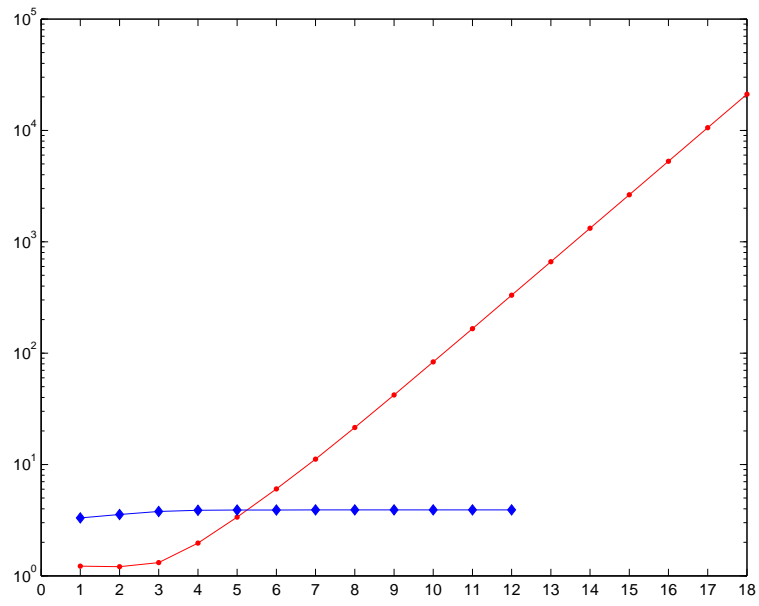


Figura 3.2: Número de condição das matrizes dos Algoritmos 3.2 e 3.3. Nós circulares: Algoritmo 3.2, nós em forma de diamante: Algoritmo 3.3.

## 4 Resultados Numéricos

Neste capítulo são utilizados dois exemplos de otimização conhecidos que permitem testar o desempenho dos métodos apresentados no Capítulo 3. Como exemplos foram utilizados o problema da mínima superfície com obstáculo, problema relativamente fácil de resolver e comumente utilizado para testar algoritmos de otimização, e o conhecido e difícil problema de otimização topológica de treliças, um dos problemas mais estudados da otimização estrutural.

Todos os cálculos foram feitos num computador tipo Intel Pentium III, 805 MHz, 1 GB RAM. Os códigos dos programas foram escritos em linguagem MATLAB exceto pelo código do programa de elementos finitos do exemplo de otimização topológica de treliças que foi escrito em linguagem FORTRAN e anexado como rotina ao programa principal escrito em linguagem MATLAB.

### 4.1 Problema da mínima superfície com obstáculo

O problema da mínima superfície com obstáculo (problema de Plateau [2]) consiste em minimizar a superfície de uma membrana para uma dada condição de contorno e com a restrição de estar por cima de um obstáculo de geometria definida previamente.

Assumindo que a função pode ser representada na forma não paramétrica pela função  $v : \mathbb{R}^2 \rightarrow \mathbb{R}$ , a solução do problema do obstáculo minimiza a função  $f : K \rightarrow \mathbb{R}$  onde:

$$f(v) = \int_D (1 + \|\nabla v(x)\|^2)^{1/2} dx$$

e o conjunto convexo  $K$  é:

$$K = \{v \in H^1(D) : v(x) = v_D(x) \text{ para } x \in \partial D, \text{ e } v(x) \geq v_L(x) \text{ para } x \in D\}$$

$H^1(D)$  é o espaço das funções com gradientes em  $L^2(D)$ , a função  $v_D : \partial D \rightarrow \mathbb{R}$  define a condição de contorno e a função  $v_L : D \rightarrow \mathbb{R}$  define a superfície do obstáculo. Uma aproximação de elementos finitos do problema do obstáculo é obtida discretizando  $D$  numa malha de triângulos e minimizando  $f$  no espaço de funções lineares por partes com valores  $v_{i,j}$  nos nós da malha.

Neste trabalho será considerado um domínio  $D = [0, 1] \times [0, 1]$ , uma malha regular de  $n_1$  por  $n_2$  nós (os retângulos de base  $1/n_1$  e de altura  $1/n_2$  divididos por uma diagonal para dar uma malha de triângulos). Seja  $(x_1, x_2)$  um ponto em  $D$ , a condição de contorno considerada é:

$$v_D(x_1, x_2) = \begin{cases} 1 - (2x_1 - 1)^2, & x_2 = 0, 1. \\ 0, & \text{outro caso.} \end{cases}$$

O obstáculo:

$$v_L(x_1, x_2) = \begin{cases} 1, & \text{se } |x_1 - \frac{1}{2}| \leq \frac{1}{4} \text{ e } |x_2 - \frac{1}{2}| \leq \frac{1}{4} \\ 0, & \text{outro caso.} \end{cases}$$

Como ponto inicial viável foi considerado o dado pela fórmula  $1 - (2x_1 - 1)^2 + 8x_1(x_1 - 1)x_2(x_2 - 1)$  avaliada nos nós da malha.

O desenho da Figura 4.1 é a solução encontrada para uma malha de 52x52 nós.

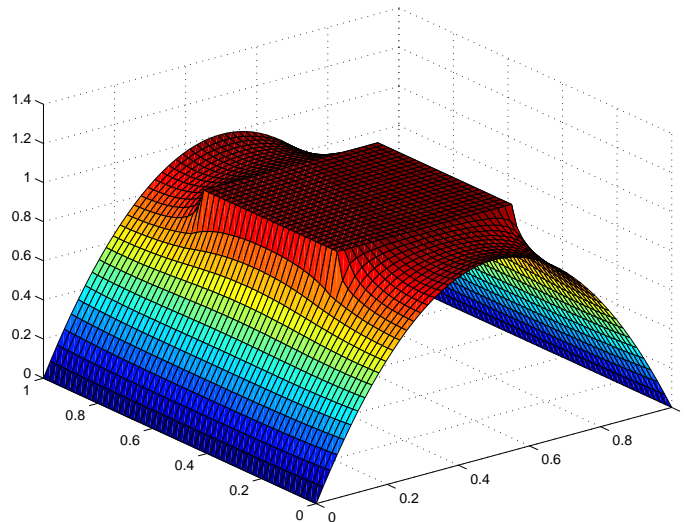


Figura 4.1: Solução do problema da mínima superfície com obstáculo.

O problema do obstáculo foi resolvido utilizando o Algoritmo 3.2, o Algoritmo 3.3 e o algoritmo FAIPA para diferentes tamanhos de malha, 22x22 nós, 52x52 nós, 102x102 nós e 202x202 nós.

As tabelas 4.1 e 4.2 mostram o número de iterações e o tempo em segundos requerido para obter a solução.

Tabela 4.1: Número de iterações requerido.

Problema	22x22	52x52	102x102	202x202
nvar	400	2500	10000	40000
pvar	400	2500	10000	40000
HNP	20	24	27	32
HNPA	16	18	24	26
HNF	24	35	54	(1)
HNFA	(2)	(2)	(2)	(2)
QNP	101	(3)	(4)	(4)
QNPA	93	(3)	(4)	(4)
QNF	201	(3)	(4)	(4)
QNFA	293	(3)	(4)	(4)
FAIPA	232	(3)	(4)	(4)

(1) Erro na resolução do sistema linear.

(2) Máximo número de iterações sem chegar à solução.

(3) Superou o limite de tempo de execução.

(4) Superou o limite de memória do computador.

Tabela 4.2: Tempo requerido em segundos.

Problema	22x22	52x52	102x102	202x202
nvar	400	2500	10000	40000
pvar	400	2500	10000	40000
HNP	5.7	42.8	257.5	2393.7
HNPA	4.9	34.8	243.8	2060.6
HNF	7.1	61.5	499.5	-
HNFA	-	-	-	-
QNP	161.9	-	-	-
QNPA	155.2	-	-	-
QNF	363.0	-	-	-
QNFA	547.3	-	-	-
FAIPA	1563.4	-	-	-

Nas Tabelas 4.1 e 4.2 “nvar” representa o número de variáveis do problema e “pvar” o número de restrições de desigualdade. Nas variantes do algoritmo 3.1, “QN” indica que foi utilizada a versão quase-Newton BFGS, “HN” pelo contrario indica que foi utilizada a matriz hessiana do problema (a matriz hessiana deste problema é definida positiva em todo ponto do domínio). A letra “F” indica que foi utilizada a função de complementaridade de Fischer-Burmeister (Algoritmo 3.3) em quanto a letra “P” indica que foi deixado o produto entre a variável  $\lambda_i$  e a função  $g_i$  (Algoritmo 3.2). A letra “A” indica que foi utilizada uma busca em arco de acordo à correção de segunda ordem.

A tolerância utilizada no critério de parada do algoritmo foi de  $10^{-9}$  para as magnitudes  $\|L_k\|$  e  $\|\Lambda_k g_k\|$  nas variantes que utilizam a matriz hessiana. Nos algoritmos que utilizam a matriz Quase-Newton e no algoritmo FAIPA, a tolerância na magnitude  $\|L_k\|$  foi de  $10^{-5}$ . Mesmo maior, essa tolerância produz valores similares da função objetivo aos obtidos com os algoritmos que utilizam a matriz hessiana.

Uma análise das Tabelas 4.1 e 4.2 indica:

1. Existe uma grande diferença nos resultados obtidos neste exemplo entre os algoritmos que utilizam a matriz hessiana do problema e os que utilizam a matriz quase-Newton BFGS. As causas desta diferença são:
  - A matriz hessiana do problema é muito esparsa (ver Figura 4.2). Por esta razão seu cômputo não tem um custo computacional alto, e, o que é mais importante ainda, é significativamente reduzida a necessidade de memória para armazenamento e número de operações requerido para sua fatorização.
  - Neste problema aparecem várias das dificuldades de trabalhar com a matriz quase-Newton: por ser uma matriz cheia os problemas maiores não podem ser resolvidos por insuficiência de memória do computador. A resolução do sistema linear (sobretudo a factorização da matriz) é consideravelmente mais custoso que quando é utilizada a matriz hessiana. Os pontos da seqüência obtida aproximam-se prematuramente

da fronteira da região viável (isto inferido da evolução das magnitudes  $\|\nabla L_k\|$  e  $\|Ag_k\|$  ver tabelas 4.3 e 4.4) o que produz a necessidade de um maior número de iterações para atingir uma dada tolerância. Em parte este problema pode ser diminuído utilizando a busca em arco e utilizando valores de  $\sigma$  grandes mas não pode ser resolvido totalmente.

2. Os tempos requeridos para resolver os problemas pelos algoritmos que utilizam a matriz hessiana são ainda muito altos, eles são consideravelmente maiores que outros algoritmos conhecidos que também a utilizam. Isto é devido ao cômputo da matriz hessiana. A implementação em linguagem Matlab utilizada é sumamente ineficiente. Por exemplo, para o problema 102x102 a iteração media do algoritmo é de 9.7 segundos, dos quais 6.8 segundos são utilizados no cômputo da matriz hessiana. Pelo contrario, o número de iterações requerido é aceitável.
3. Para os algoritmos que utilizam o produto a busca em arco reduz o número de iterações necessário para obter a solução. Tanto para o algoritmo HNP, para o qual o cômputo da matriz hessiana é custoso, como para o algoritmo QNP, para o qual a fatoração da matriz é custosa, a busca em arco reduz o tempo de execução. Pelo contrário, a busca em arco prejudica o funcionamento dos algoritmos que utilizam a função de Fischer-Burmeister.
4. Os algoritmos que utilizam o produto se apresentam melhores que os correspondentes que utilizam a função de Fischer-Burmeister, isto parece ser devido a passos pequenos dados por estes últimos nas primeiras iterações.

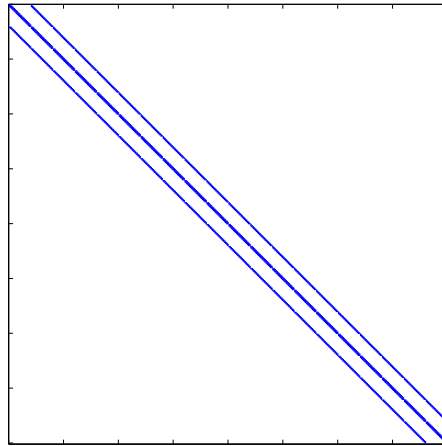


Figura 4.2: Padrão esparsa da matriz hessiana do problema 22x22.

Tabela 4.3: Saída do algoritmo HNP, problema 21x21.

Iter	IterBL	$t$	$f(x_k)$	$\ \nabla L_k\ $	$\ (\Lambda_k g_k, h_k)\ $
0	0	0.000	3.1155e+000	3.829e+000	3.083e+000
1	1	0.953	3.1105e+000	1.801e-001	2.578e-001
2	2	0.700	3.1065e+000	7.189e-002	1.418e-001
3	1	1.000	2.9074e+000	1.819e-002	6.905e-002
4	1	0.887	2.5821e+000	1.472e-002	2.867e-002
...	...	...	...	...	...
15	1	1.000	2.4295e+000	5.177e-007	9.497e-007
16	1	1.000	2.4295e+000	7.943e-008	2.305e-007
17	1	1.000	2.4295e+000	1.063e-008	5.768e-008
18	1	1.000	2.4295e+000	1.356e-009	1.444e-008
19	1	1.000	2.4295e+000	1.705e-010	3.613e-009
20	1	1.000	2.4295e+000	2.134e-011	9.034e-010

Tabela 4.4: Saída do algoritmo QNP, problema 21x21.

Iter	IterBL	$t$	$f(x_k)$	$\ \nabla L_k\ $	$\ (\Lambda_k g_k, h_k)\ $
0	0	0.000	3.1155e+000	3.829e+000	3.083e+000
1	1	0.980	3.1144e+000	2.086e-001	1.894e-001
2	8	0.082	3.1143e+000	1.988e-001	1.838e-001
3	10	0.040	3.1143e+000	1.911e-001	1.805e-001
4	10	0.040	3.1142e+000	1.833e-001	1.776e-001
...	...	...	...	...	...
96	1	0.945	2.4295e+000	5.139e-005	6.107e-011
97	1	0.794	2.4295e+000	1.920e-005	5.078e-011
98	2	0.604	2.4295e+000	2.274e-005	4.427e-011
99	1	1.000	2.4295e+000	1.258e-005	3.524e-011
100	1	1.000	2.4295e+000	1.584e-005	1.687e-011
101	1	1.000	2.4295e+000	9.991e-006	9.481e-012

## 4.2 Otimização topológica de treliças

Como segundo exemplo aplicam-se algumas variantes do Algoritmo 3.2 no conhecido exemplo de otimização topológica de treliças.

Serão considerados dois modelos: O primeiro consiste em minimizar a máxima energia de deformação produzida pelos diferentes estados de carregamento. O segundo consiste em minimizar o volume total de material sujeito a restrições nas tensões nas barras para os diversos estados de carregamento.

Em ambos os modelos é dado um conjunto de  $n$  nós, ou seja, pontos no plano ou no espaço tridimensional. São dadas também  $b$  arestas, ou seja, conexões entre nós que poderão ser barras da estrutura, estas arestas poderão ser todas as possíveis entre os nós ou um subconjunto delas. São dados também os apoios, ou seja, as restrições ao deslocamento para cada um dos nós e um ou vários estados de carregamento que a estrutura deverá suportar. As variáveis fundamentais do problema são os volumes de material que serão atribuídos a cada aresta.

O objetivo é encontrar a distribuição ótima de material, ou seja, a distribuição



de volumes que corresponde à melhor estrutura de acordo num determinado critério. Na solução do problema algumas das barras poderão resultar com um volume de material nulo, por isto se diz que os modelos utilizados são de otimização topológica.

A Figura 4.3 é um exemplo de uma treliça composta por 6 nós e 10 arestas representadas em traço cinza. A força sobre o nó 2 é a única força presente.

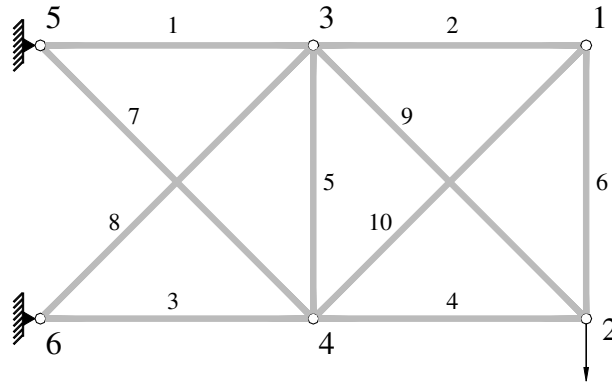


Figura 4.3: Exemplo de 6 nós e 10 arestas.

Uma possível solução ótima mostra-se na Figura 4.4, formada pelos nós 2, 3, 4, 5 e 6 e as barras 1, 3, 4, 8 e 9. O nó 1 e as barras 2, 5, 6, 7 e 10 não formam parte da solução.

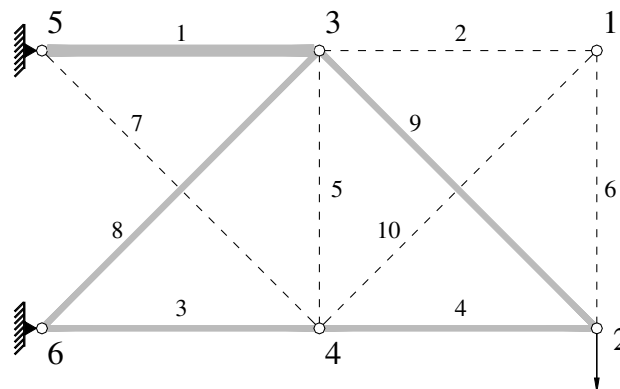


Figura 4.4: Solução para o exemplo da Figura 4.3.

### 4.2.1 Modelo 1: Minimização da energia de deformação

O problema de minimização da energia de deformação para um único estado de carregamento pode ser formulado como:

$$\text{minimize} \quad p^T u \quad (4.1)$$

$$\text{sujeito a:} \quad \sum_{i=1}^b t_i K_i u = p \quad (4.2)$$

$$\sum_{i=1}^b t_i \leq V \quad (4.3)$$

$$t_i \geq 0 \quad i = 1, \dots, b \quad (4.4)$$

Seja  $l$  o número de graus de liberdade da treliça. No modelo  $p \in \mathbb{R}^l$  representa o vetor global de forças nodais atuando sobre a estrutura,  $u \in \mathbb{R}^l$  é o vetor global de deslocamentos nodais,  $K_i \in \mathbb{R}^{l \times l}$  é a contribuição na matriz de rigidez da estrutura da barra número  $i$  calculada para um volume unitário,  $t_i$  é o volume da barra  $i$ ,  $V$  é o volume total a repartir entre todas as barras e  $b$  é o número total de arestas escolhidas que poderão ser barras da estrutura.

As variáveis do modelo são os deslocamentos dados pelo vetor  $u$  e os volumes das barras dados pelo vetor  $t$ . A Eq. 4.1 expressa que a distribuição de volumes das barras dada por  $t$ , quando é solução do problema, minimiza a energia de deformação da estrutura produzida pelas forças nodais  $p$ . De outra forma pode-se dizer que  $t$  é a distribuição de volumes que maximiza a rigidez da estrutura em relação às forças nodais dadas por  $p$ . A Eq. 4.2 é a equação de equilíbrio do problema já que a matriz de rigidez da estrutura é:

$$K(t) = \sum_{i=1}^b t_i K_i \quad (4.5)$$

A Eq. 4.3 é uma restrição para a distribuição de volumes que expressa que a soma dos volumes é menor ou igual à quantidade de volume de material disponível para repartir entre as barras.

A função objetivo do problema de otimização dado pelas Eq. 4.1 a 4.4 é linear, também são lineares as funções que definem as restrições de desigualdade. A única função não linear do problema é a que define a equação de equilíbrio do problema, Eq. 4.2. Seja  $h(t, u) = K(t)u - p$ . O gradiente da função  $h$  é:

$$\nabla h(t, u) = (\nabla_t h, \nabla_u h)$$

onde:

$$\nabla_t h = (K_1 u, \dots, K_b u) \quad (4.6)$$

$$\nabla_u h = K(t) \quad (4.7)$$

Como  $h$  é a única função não linear (na verdade é uma função quadrática), a matriz hessiana do problema é  $H(t, u, \lambda, \mu) = \sum_{i=1}^l \nabla^2 h_i(t, u) \mu_i$ . Substituindo as matrizes de derivadas segundas tem-se:

$$H(t, u, \lambda, \mu) = \begin{pmatrix} H_{tt} & H_{tu} \\ H_{ut} & H_{uu} \end{pmatrix}$$

onde:

$$H_{tt} = 0 \quad (4.8)$$

$$H_{ut} = H_{tu}^T = (K_1 \mu, \dots, K_b \mu) \quad (4.9)$$

$$H_{uu} = 0 \quad (4.10)$$

As Eqs. 4.8 a 4.10 mostram que para este problema o cômputo da matriz hessiana não tem um custo alto, sendo na verdade menor ao cômputo da matriz  $\nabla h$  (é igual ao cômputo da matriz  $\nabla_t h$ ). Elas também mostram que a matriz hessiana não é definida positiva em nenhum ponto, na verdade ela somente pode ser semidefinida positiva se for a matriz nula. Seja  $v = (v_1, v_2)$  um autovetor de autovalor positivo  $\gamma$ . Então, tem-se:

$$\begin{pmatrix} 0 & H_{tu} \\ H_{ut} & 0 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \gamma \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}$$

Portanto  $H_{tu} v_2 = \gamma v_1$  e  $H_{ut} v_1 = \gamma v_2$ . Como também se verifica  $H_{tu}(-v_2) = -\gamma v_1$  e  $H_{ut} v_1 = -\gamma(-v_2)$ , se verifica:

$$\begin{pmatrix} 0 & H_{tu} \\ H_{ut} & 0 \end{pmatrix} \begin{pmatrix} v_1 \\ -v_2 \end{pmatrix} = -\gamma \begin{pmatrix} v_1 \\ -v_2 \end{pmatrix}$$

Portanto  $-\gamma$  é também autovalor da matriz hessiana.

Quando devem ser considerados vários estados de carregamento, uma das possíveis formas de formular o problema consiste em minimizar o valor máximo entre as energias de deformação produzidas pelos mesmos. De forma equivalente,

para formular um problema diferenciável, pode-se adicionar a variável auxiliar  $z$  e fazer:

$$\begin{aligned}
 &\text{minimize} && z \\
 &\text{sujeito a:} && z \geq p^j T u^j && j = 1, \dots, s \\
 & && K(t)u^j = p^j && j = 1, \dots, s \\
 & && \sum_{i=1}^b t_i \leq V \\
 & && t_i \geq 0 && i = 1, \dots, b
 \end{aligned}$$

Onde  $s$  é o número de estados de carregamento considerados.

#### 4.2.2 Modelo 2: Minimização do volume

Neste caso o problema de otimização é:

$$\begin{aligned}
 &\text{minimize} && \sum_{i=1}^b t_i \\
 &\text{sujeito a:} && K(t)u^j = p^j && j = 1, \dots, s \\
 & && t_i \geq 0 && i = 1, \dots, b \\
 & && \sigma_{min} \leq \sigma_i(u^j) \leq \sigma_{max} && i = 1, \dots, b; \quad j = 1, \dots, s
 \end{aligned}$$

Onde  $\sigma_i(u^j)$  representa a tensão na barra  $i$  para o deslocamento  $u^j$ .  $\sigma_i(u^j)$  é uma função linear em  $u^j$ . De fato, como mostra a Figura 4.5, a tensão na barra representada calcula-se pela fórmula:

$$\sigma = E_m \frac{\Delta L}{L} = E_m \frac{(u_3 - u_1) \cos(\theta) + (u_4 - u_2) \sin(\theta)}{L}$$

Onde  $L$  é o comprimento da barra e  $E_m$  é o módulo de elasticidade do material.

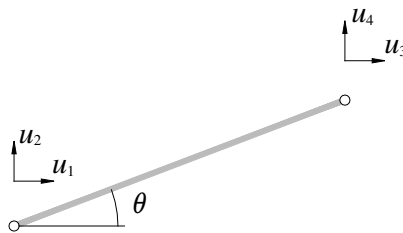


Figura 4.5: Barra da estrutura.

Este modelo pode ser consideravelmente mais difícil de resolver que o Modelo 1. Isto é devido ao aumento do número de restrições de desigualdade pelas restrições nas tensões nas barras. Uma vantagem do Modelo 2 com respeito do Modelo 1 aparece na comparação das soluções achadas. No Modelo 1, quando o problema tem vários estados de carregamento, algumas barras apresentam tensões altas para alguns estados. No Modelo 2 isto não acontece precisamente pela inclusão das restrições nas tensões nas barras.

As Figuras 4.6 e 4.7 correspondem ao problema B-10 apresentado na seção seguinte (seção 4.2.4). O mesmo corresponde a uma configuração de nós e arestas como mostrada na Figura 4.3 com dois estados de carregamento (ver seção 4.2.4). Nas figuras mostram-se as tensões (em módulo) que aparecem nas 10 barras nas treliças soluções dos modelos 1 e 2. Para comparar as soluções, os volumes das barras foram divididos entre o volume final da estrutura de forma de ter dois soluções com o mesmo volume. Na Figura 4.6 podem-se apreciar tensões altas nas barras 3 e 4 para o estado de carregamento 2. O mesmo não acontece na solução do modelo de otimização 2, as tensões para as barras 3 e 4 no estado de carregamento 2 coincidem com as tensões nas barras 1, 8 e 9 para o estado de carregamento 1. Precisamente as barras 1, 3, 4, 8 e 9 são as barras que resultam com área positiva (ver Figura 4.4). As barras que resultam com área nula apresentam tensões menores, isto pode ser utilizado como critério para identificar as barras positivas da solução do Modelo 2.

Nas Figuras 4.8 e 4.9 mostram-se as tensões máximas e mínimas nas barras do problema B-200. As mesmas são apresentadas em ordem crescente.

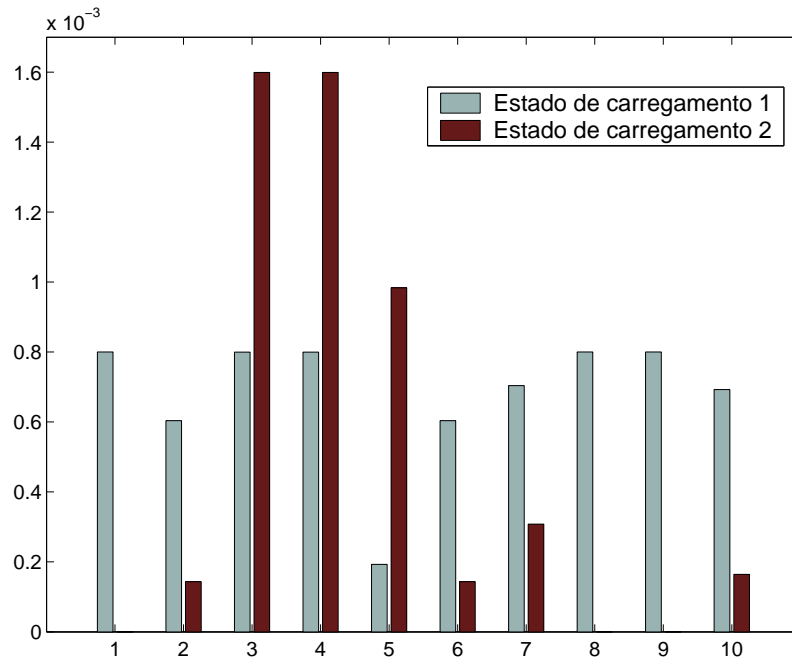


Figura 4.6: Tensões nas barras para a solução do problema B-10, Modelo 1.

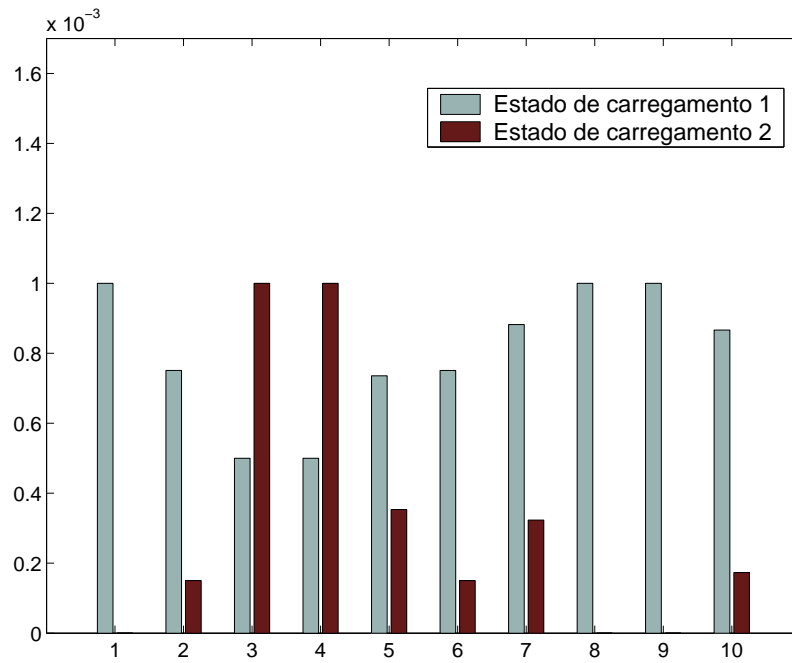


Figura 4.7: Tensões nas barras para a solução do problema B-10, Modelo 2.

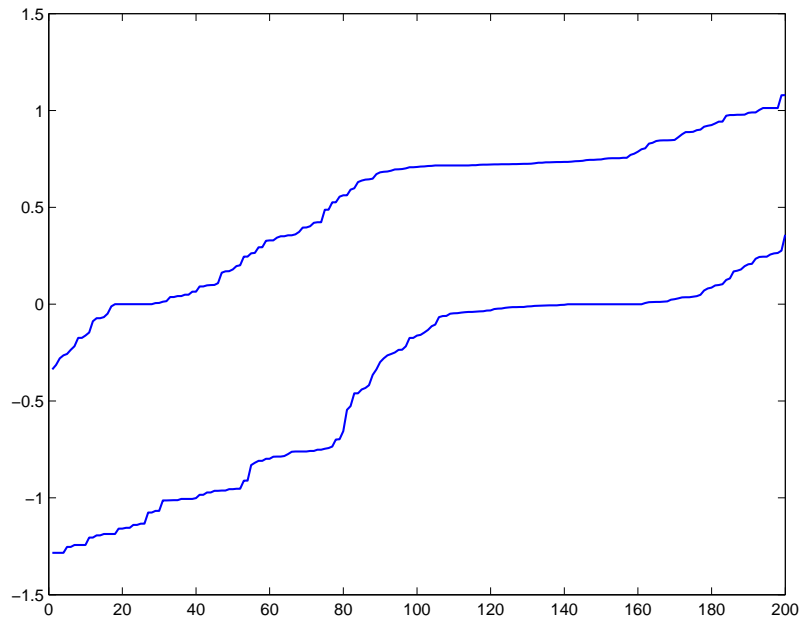


Figura 4.8: Tensões nas barras para a solução do problema B-200, Modelo 1.

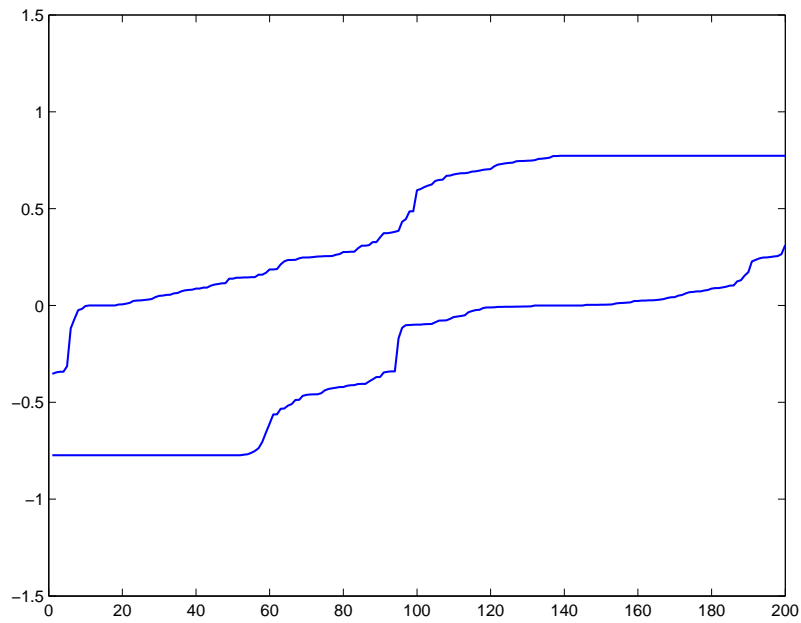


Figura 4.9: Tensões nas barras para a solução do problema B-200, Modelo 2.

### 4.2.3 Modelo 3: Formulação implícita

Seja o seguinte problema equivalente com o problema da seção 4.2.1 para um único estado de carregamento [26]:

$$\begin{aligned} & \text{minimize} && \phi(t) \\ & \text{sujeito a:} && \sum_{i=1}^b t_i \leq V \\ & && t_i \geq 0 \quad i = 1, \dots, b \end{aligned} \tag{4.11}$$

onde a função  $\phi$  é definida como:

$$\phi(t) = \begin{cases} p^T u & \text{se existe } u \text{ tal que } K(t)u = p \\ +\infty & \text{outro caso} \end{cases} \tag{4.12}$$

Neste caso as únicas variáveis de projeto estão dadas pelo vetor  $t$ .

É conhecido que a função  $\phi$  é uma função convexa e própria quando  $K(t)$  não é singular para algum  $t > 0$  [27]. Isto pode ser deduzido a partir da equação seguinte [28]:

$$\tau \geq \phi(t) \Leftrightarrow A(\tau, t) = \begin{pmatrix} \tau & p^T \\ p & K(t) \end{pmatrix} \geq 0$$

onde  $A \geq 0$  indica que a matriz  $A$  é semidefinida positiva. Como o epígrafe da função  $\phi$  coincide com a região dada pelos pontos  $(\tau, t)$  onde a matriz  $A(\tau, t)$  é semidefinida positiva, e, como  $A$  é função afim da variável  $(\tau, t)$ , tem-se que o epígrafe da função  $\phi$  é um domínio convexo.

Esta formulação do problema de otimização topológica de treliças tem a vantagem de definir um problema convexo com um número de variáveis menor. A maior desvantagem é que o problema não é diferenciável, usualmente a função  $\phi$  não é diferenciável no ponto solução.

Embora o problema dado pela Eq. 4.11 não seja diferenciável, pode ser resolvido por algoritmos eficientes, por exemplo utilizando algoritmos de programação semidefinida [28, 29].

Outra forma de resolver o problema, que pode ser aplicada nos exemplos menores, é utilizar algoritmos para resolução de problemas convexas gerais. Estes algoritmos requerem do cálculo de subgradientes. Nos pontos onde  $t \geq 0$  o vetor  $u$  é único e calcula-se pela expressão  $u = u(t) = K(t)^{-1}p$ . Nesses pontos a função  $\phi$  é



diferenciável pois tem-se:

$$\begin{aligned}\frac{\partial\phi(t)}{\partial t_i} &= p^T \frac{\partial u(t)}{\partial t_i} \\ &= u(t)^T K(t) \frac{\partial u(t)}{\partial t_i}\end{aligned}\quad (4.13)$$

Dado que  $K(t)u(t) = p$  tem-se:

$$\frac{\partial K(t)}{\partial t_i} u(t) + K(t) \frac{\partial u(t)}{\partial t_i} = 0 \quad (4.14)$$

Substituindo a Eq. 4.14 na Eq. 4.13 tem-se:

$$\begin{aligned}\frac{\partial\phi(t)}{\partial t_i} &= -u(t)^T \frac{\partial K(t)}{\partial t_i} u(t) \\ &= -u(t)^T K_i u(t)\end{aligned}\quad (4.15)$$

A Eq. 4.15 permite achar um sub-gradiente da função  $\phi$  sem ter que resolver sistemas lineares adicionais.

No caso de ter vários estados de carregamento, uma formulação que pode ser conveniente é a de minimizar a maior das energias de deformação produzidas pelos mesmos. Sejam por exemplo  $s$  estados de carregamento diferentes. O problema de otimização é:

$$\begin{aligned}\text{minimize} \quad & \max\{\phi_j(t), 1 \leq j \leq s\} \\ \text{sujeito a:} \quad & \sum_{i=1}^b t_i \leq V \\ & t_i \geq 0 \quad \quad \quad i = 1, \dots, b\end{aligned}\quad (4.16)$$

onde as funções  $\phi_j$  são:

$$\phi_j(t) = \begin{cases} p_j^T u & \text{se existe } u \text{ tal que } K(t)u = p_j \\ +\infty & \text{outro caso} \end{cases}\quad (4.17)$$

Este problema é do tipo min-max onde cada função  $\phi_j$  é convexa. A função  $\phi$  é também convexa mas pode não ser diferenciável nos pontos onde  $\phi$  é igual a dois ou mais funções  $\phi_j$ . Nestes pontos pode-se achar um sub-gradiente da função  $\phi$  tomando uma combinação convexa dos gradientes das funções  $\phi_j$  que coincidem com o valor da função  $\phi$ .

Na referência [30] mostram-se alguns resultados numéricos obtidos utilizando o algoritmo para otimização de funções convexas não diferenciáveis apresentado na referência [31].

#### 4.2.4 Resultados obtidos, Modelos 1 e 2

Para testar os algoritmos propostos serão utilizados sete exemplos diferentes chamados B-10, Viga, B-200, B-200b, B-9x9, B-13x9 e B-21x9. Em todas as figuras, os pontos circulares pequenos representam os nós livres, os pontos circulares grandes representam os nós fixos, os nós onde estão aplicadas as forças estão representados em forma de diamante e as arestas que formam parte do problema são representadas em linha tracejada.

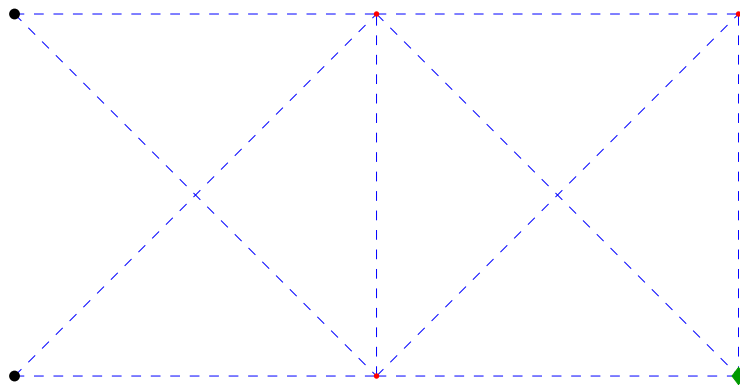


Figura 4.10: Exemplo B-10

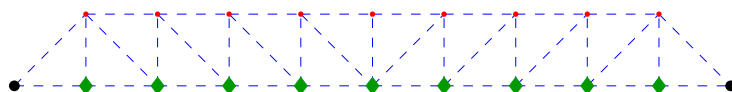


Figura 4.11: Exemplo Viga.

O exemplo B-10 consiste numa configuração de 6 nós, 10 arestas e dois estados de carregamento, o primeiro dado por uma força vertical (de valor 1 para abaixo) no nó em forma de diamante e o outro dado por uma força horizontal (de valor 2 para a direita) no mesmo nó. O exemplo Viga tem 20 nós, 37 arestas e um único estado de carregamento dado por forças verticais iguais nos nós inferiores. O Exemplo B-200 tem 77 nós, 200 arestas e três estados de carregamento, o primeiro por forças horizontais (de valor 1000 para a direita), o segundo por forças verticais (de valor 10000 para abaixo) e o terceiro é a soma dos anteriores. O exemplo B-200b é igual que o anterior mas com um único estado de carregamento, o segundo do exemplo

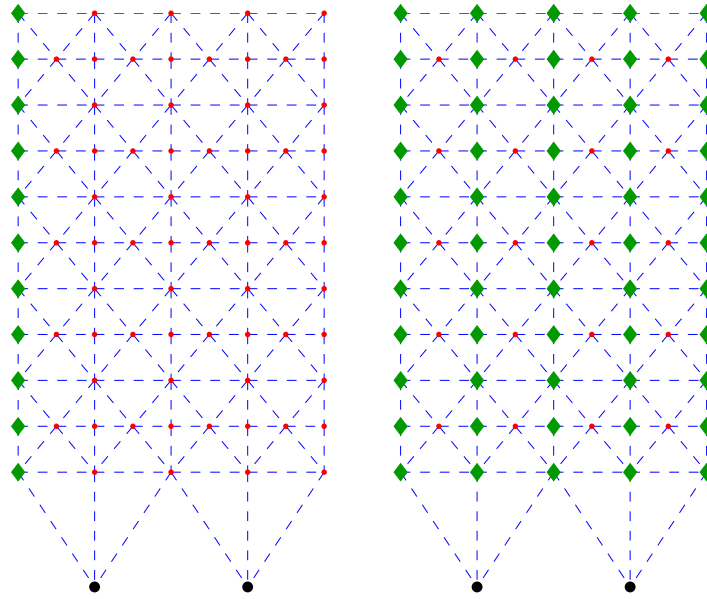


Figura 4.12: Exemplo B-200, estados de carregamento 1 e 2.

anterior. Os exemplos B-9x9, B-13x9 e B-21x9 consistem em malhas retangulares de nós, o lado esquerdo da malha é onde estão os nós fixos e no nó que está no centro do lado direito está aplicada uma força vertical. As arestas do exemplo são todas as possíveis entre os nós de forma que não se solapem e que não unam nós fixos. No exemplo B-9x9 são 2032 arestas, no exemplo B-13x9 são 4208 arestas e no exemplo B-21x9 são 10932 arestas.

Nos exemplos Viga, B-200 e B-200b tem-se uma restrição adicional que deve cumprir a solução do problema: a estrutura deve ser simétrica para um eixo de simetria vertical. Em vez de agregar uma restrição de igualdade para cada par de barras correspondentes pela simetria, opto-se por eliminar do problema as variáveis correspondentes aos volumes de uma das barras dos pares.

Na Tabela 4.5 resumem-se as características dos problemas resolvidos. Na mesma, “NGL” indica o número de graus de liberdade da estrutura, “VA” o número de variáveis correspondentes as arestas (nos problemas Viga, B-200 e B-200b pela restrição de simetria VA é menor ao número de arestas) e “EC” é o número de estados de carregamento considerados. “nvar”, “mvar” e “pvar” são respectivamente o número de variáveis do problema de otimização, o número de restrições de desigualdade e o número de restrições de igualdade.

Tabela 4.5: Características dos problemas resolvidos.

Problema	B-10	Viga	B-200	B-200b	B-9x9	B-13x9	B-21x9
nós	6	20	77	77	81	117	189
NGL	8	36	150	150	144	216	360
arestas	10	37	200	200	2032	4208	10932
VA	10	19	105	105	2032	4208	10932
EC	2	1	3	1	1	1	1
Modelo 1							
nvar	27	56	556	256	2177	4425	11293
mvar	13	21	109	107	2034	4210	10934
pvar	16	36	450	150	144	216	360
Modelo 2							
nvar	26	55	555	255	2176	4424	11292
mvar	50	93	1305	505	6096	12624	32796
pvar	16	36	450	150	144	216	360

As Tabelas 4.6 e 4.7 mostram o número de iterações e o tempo em segundos requerido para obter uma solução.

Tabela 4.6: Número de iterações requerido.

Problema	B-10	Viga	B-200	B-200b	B-9x9	B-13x9	B-21x9
HNP - M1	26	24	40	28	40	52	59
HNPA - M1	24	24	35	28	42	37	38
HNP - M2	26	20	31	23	36	40	(4)
HNPA - M2	22	20	23	20	41	34	(4)

(4) Superou o limite de memória do computador.

Tabela 4.7: Tempo requerido em segundos.

Problema	B-10	Viga	B-200	B-200b	B-9x9	B-13x9	B-21x9
HNP - M1	0.6	0.8	53.1	4.4	69.1	367.0	3251.7
HNPA - M1	0.8	1.0	48.8	5.4	76.7	279.6	2064.5
HNP - M2	1.1	0.8	204.8	6.4	149.3	669.2	-
HNPA - M2	1.0	1.2	168.0	7.3	217.9	604.8	-

As soluções encontradas para ambos os modelos de otimização estão representadas nas figuras seguintes. Nas mesmas aparecem somente as barras de volume positivo. No Anexo 1 estão descritas em forma mais detalhada as características geométricas das treliças e das soluções encontradas.

Na Figura 4.13 mostra-se a solução encontrada para o exemplo B-10 no modelo de otimização 1, no modelo de otimização 2 a solução encontrada é diferente mas não se percebe no desenho. Na Figura 4.14 mostra-se a solução achada para o exemplo Viga, neste caso a solução achada coincide para os dois modelos de otimização.

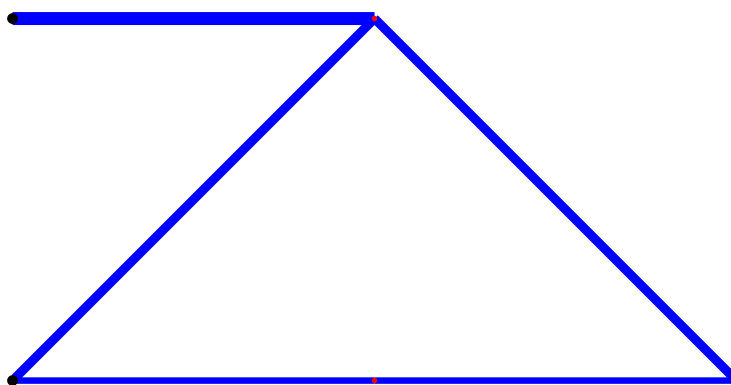


Figura 4.13: Exemplo B-10, solução achada, Modelo 1.

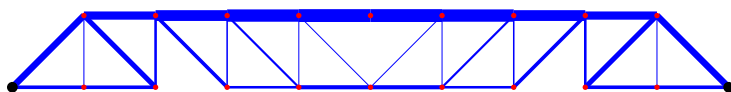


Figura 4.14: Exemplo Viga, solução achada, Modelos 1 e 2.

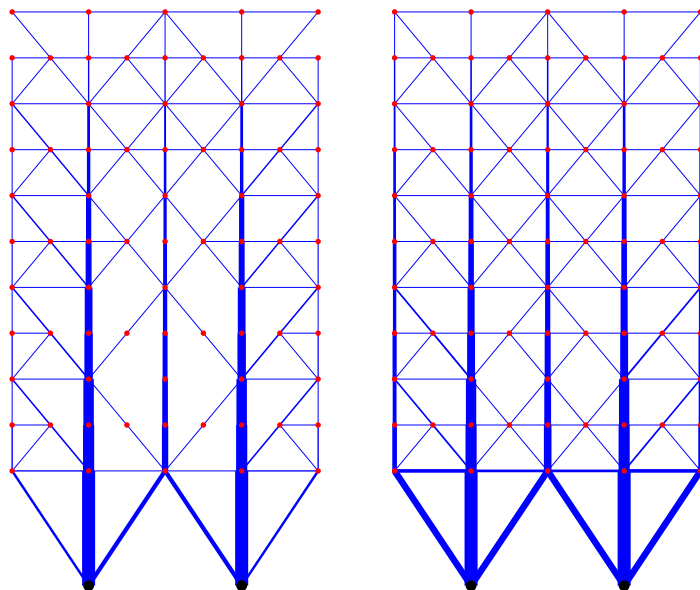


Figura 4.15: Exemplo B-200, solução achada, Modelos 1 e 2.

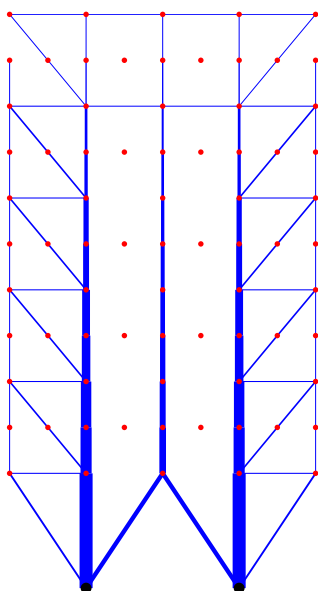


Figura 4.16: Exemplo B-200b, solução achada, Modelos 1 e 2.

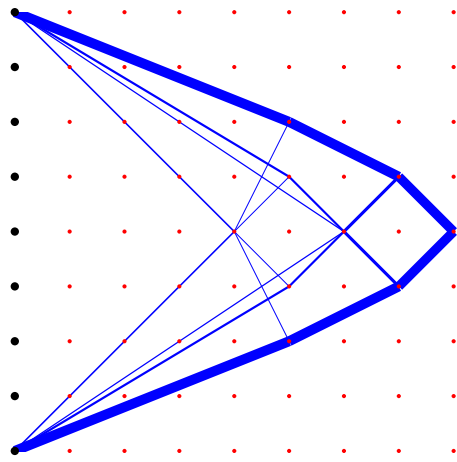


Figura 4.17: Exemplo B-9x9, solução achada, Modelos 1 e 2.

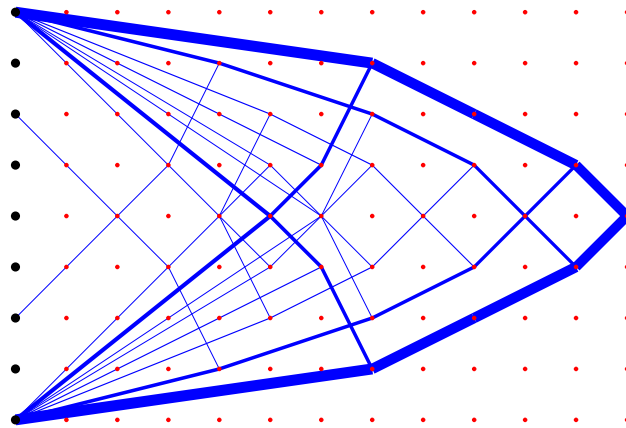


Figura 4.18: Exemplo B-13x9, solução achada, Modelos 1 e 2.

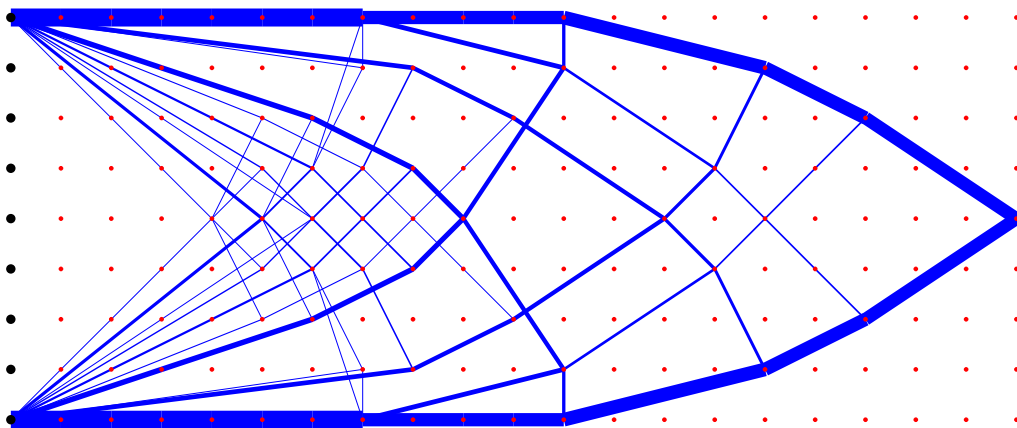


Figura 4.19: Exemplo B-21x9, solução achada, Modelo 1.

Os resultados apresentados nas Tabelas 4.6 e 4.7 foram obtidos para valores da tolerância nas magnitudes  $\|\nabla L_k\|$  e  $\|(\Lambda_k g_k, h_k)\|$  dependentes do problema considerado. Em geral foram utilizados valores da tolerância muito baixos, da ordem de  $10^{-14}$ , isto devido a que a solução do problema apresenta barras de volume muito baixo em comparação com as barras maiores. Isto faz difícil identificar as barras que na solução têm volume nulo das menores que ficam com volume positivo. Para isto é requerida uma solução muito precisa. Por exemplo, no problema B-13x9 o valor da tolerância requerido foi de  $10^{-14}$ , se o valor da tolerância tivesse sido escolhido como sendo  $10^{-8}$  a solução teria sido a representada pela Figura 4.20. Nela são representadas todas as barras que têm volume maior à de menor volume da Figura 4.18.

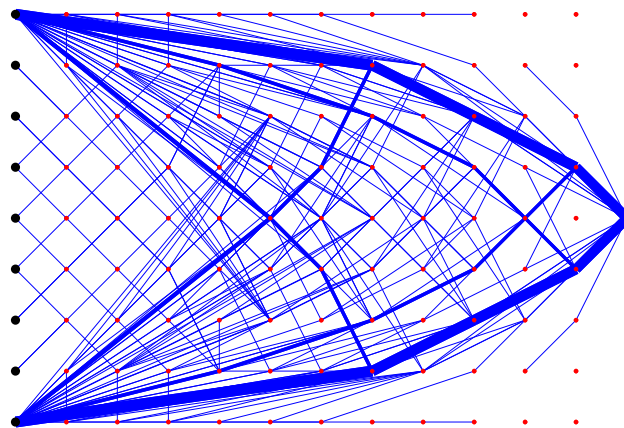


Figura 4.20: Exemplo B-13x9, solução achada, Modelo 1.

As baixas tolerâncias requeridas para chegar a uma solução aceitável somado a que usualmente na solução do problema a matriz de derivadas é singular faz que o problema de otimização de treliças seja muito difícil de resolver, com certeza muito mais difícil que o primeiro exemplo apresentado, o problema do obstáculo.

Não surpreende, portanto, que não se tenha conseguido resolver o problema com alguns dos algoritmos propostos. De forma mais detalhada, os comentários sobre os algoritmos propostos são os seguintes:

1. Os algoritmos de tipo quase-Newton não são convenientes para resolver este tipo de problemas. Como mostrou-se nas Eqs. 4.8 a 4.10 o cômputo da



matriz hessiana tem custo reduzido e portanto carece de sentido prescindir da informação de segunda ordem na resolução destes problemas. As baixas tolerâncias requeridas para obter soluções que não sejam como na mostrada na Figura 4.20 indica que provavelmente a única forma de resolver estes problemas seja por algoritmos baseados no método de Newton. De qualquer forma, tentou-se resolver os problemas com os algoritmos quase-Newton com a intenção de testar seu funcionamento, o único problema resolvido foi o mais pequeno, o problema B-10, e com um número de iterações alto.

2. Como foi mostrado, a matriz hessiana do problema não é definida positiva, antes de modificar a mesma para torná-la definida positiva foram feitos alguns testes preliminares com a intenção de detectar erros de programação. Com grande surpresa vários dos problemas foram resolvidos em forma muito eficiente. A partir desse momento foram modificadas algumas outras características do programa com a intenção de manter a matriz hessiana e resolver os problemas que não foram resolvidos numa primeira instância. As principais modificações realizadas foram as seguintes:

- Modificação da busca linear: Ainda que a matriz hessiana é indefinida, a direção da iteração de Newton,  $d_x^a$ , resultou ser de descida para a função potencial. Lamentavelmente a restrição na escolha de  $\sigma$  dada pela Eq. 3.12 obrigou ao algoritmo a definir  $\sigma$  muito pequeno e com isso, nos casos que o algoritmo falhou, obteve-se passos cada vez menores e convergência a pontos na fronteira da região viável. Com a intenção de manter  $\sigma$  dado pela Eq. 2.37 tentou-se utilizar a função de barreira logarítmica  $b_\sigma$  (ver seção 1.2.2) como função potencial o que melhorou em parte o funcionamento do algoritmo mas tampouco mostrou resultados satisfatórios. Por fim decidiu-se abandonar a busca linear na função potencial permitindo passos maiores o que melhorou significativamente o desempenho do algoritmo.
- Modificação do ponto inicial: É conhecido que os algoritmos de ponto interior podem ter um pobre desempenho e inclusive falhar se o ponto

inicial é desfavorável [32]. Para este exemplo os valores iniciais dos multiplicadores de Lagrange foram escolhidos da seguinte forma:  $\lambda_i = \|\nabla f(x_0)\|/\|\nabla g_i(x_0)\|$ ,  $\mu_i = 0$ . Os valores dos volumes iniciais foram escolhidos como:  $t_i = 0.1V/b$ . Para os deslocamentos foi escolhido o ponto inicial  $u = 0$ .

Com as modificações citadas acima os algoritmos propostos conseguiram resolver os problemas apresentados e os outros problemas testados que não foram descritos neste trabalho.

3. O algoritmo com busca em arco foi melhor nos problemas B-13x9 e B-21x9, precisamente os problemas maiores onde a fatoração da matriz tem custo alto e portanto a redução do número de iterações reduz o tempo de execução. Nos problemas menores a utilização do arco não resultou em tempos de execução menores.
4. O algoritmo que utiliza a função de Fischer-Burmeister, mostrou-se inadequado para este tipo de problemas, não resolveu nenhum dos problemas testados.
5. Nestes problemas a matriz do sistema linear é muito esparsa, por exemplo, para o problema B-200 a matriz é a representada nas Figuras 4.21 e 4.23 para os Modelos 1 e 2 respectivamente. Previamente à fatoração da matriz foi utilizada uma rotina de análise para reordenar linhas e colunas com o objetivo de aumentar a esparsidade dos fatores. A rotina de análise utilizada é a rotina de Matlab SYMAMD (*Symmetric approximate minimum degree permutation*) apropriada para matrizes definidas positivas e que em alguns casos serve também para matrizes indefinidas. O padrão esparsa da matriz logo após reordenamento é mostrada também nas Figuras 4.21 e 4.23. Nas figuras 4.22 e 4.24 mostram-se os fatores da decomposição LU. Essas figuras mostram a razão da diferença de tempo requerido para resolver os problemas de otimização nos Modelos 1 e 2. A matriz do sistema linear, além de ser maior da fatores mais cheios para o Modelo 2 que para o Modelo 1.

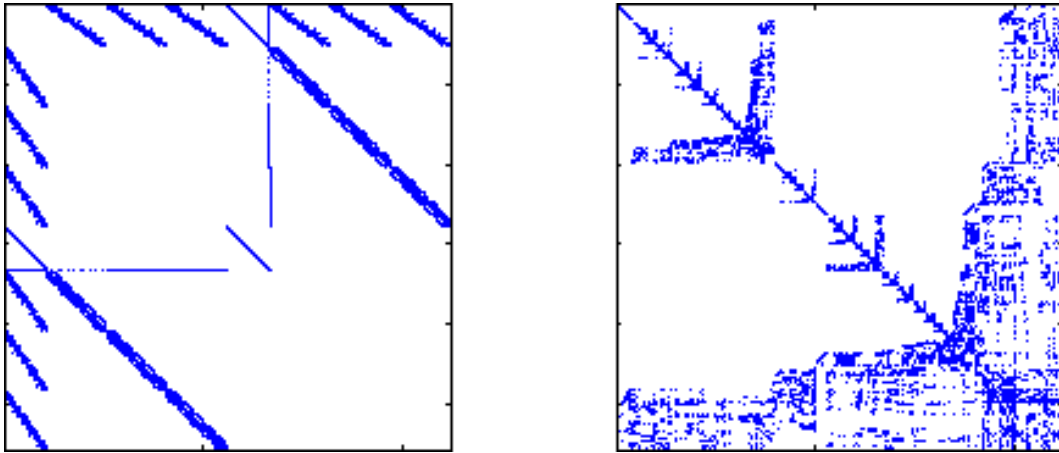


Figura 4.21: Padrão esparsa da matriz do sistema linear do problema B-200 para o modelo de otimização 1, a mesma tem 1115 linhas e colunas e 13613 elementos não nulos. A direita a matriz logo da reordenação de linhas e colunas.

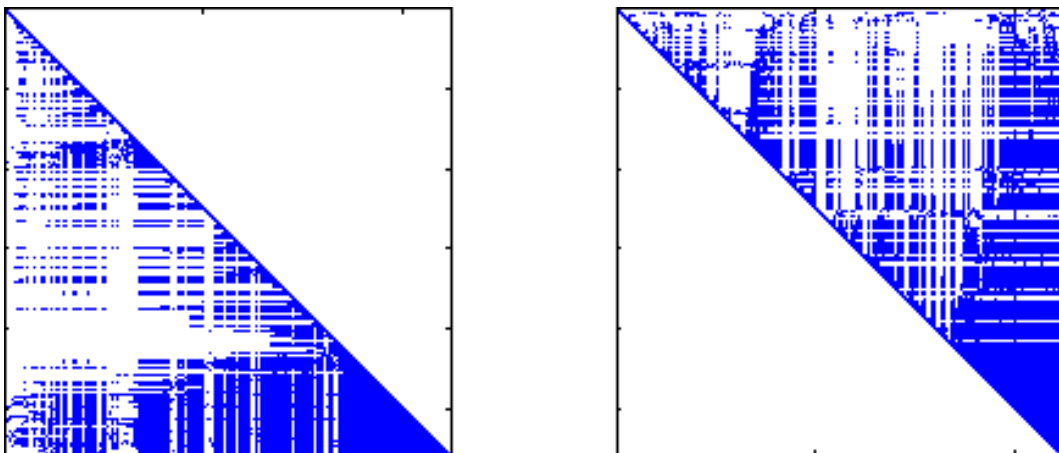


Figura 4.22: Padrão esparsa da matriz dos fatores LU da matriz do sistema linear do problema B-200 para o modelo de otimização 1. O fator L tem 86833 elementos não nulos, o fator U tem 101821 elementos não nulos.

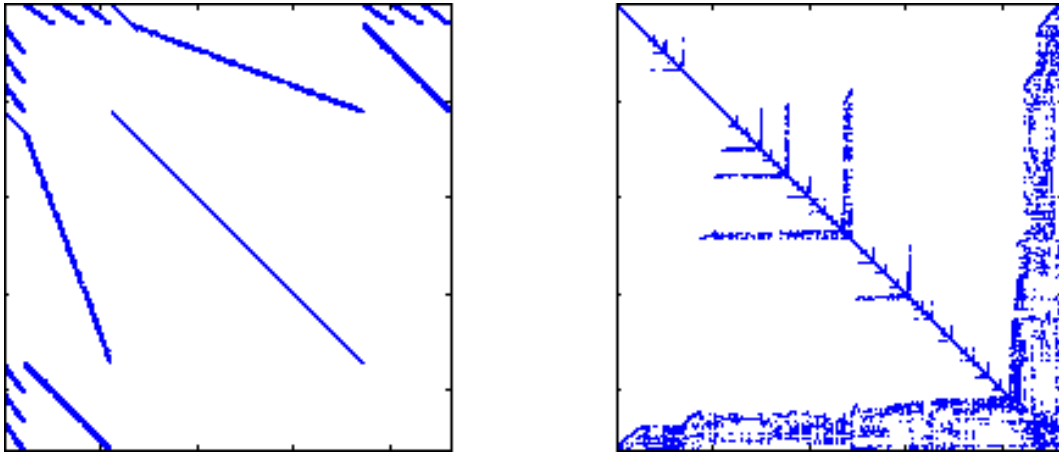


Figura 4.23: Padrão esparsa da matriz do sistema linear do problema B-200 para o modelo de otimização 2, a mesma tem 2310 linhas e colunas e 21055 elementos não nulos. A direita a matriz logo da reordenação de linhas e colunas.

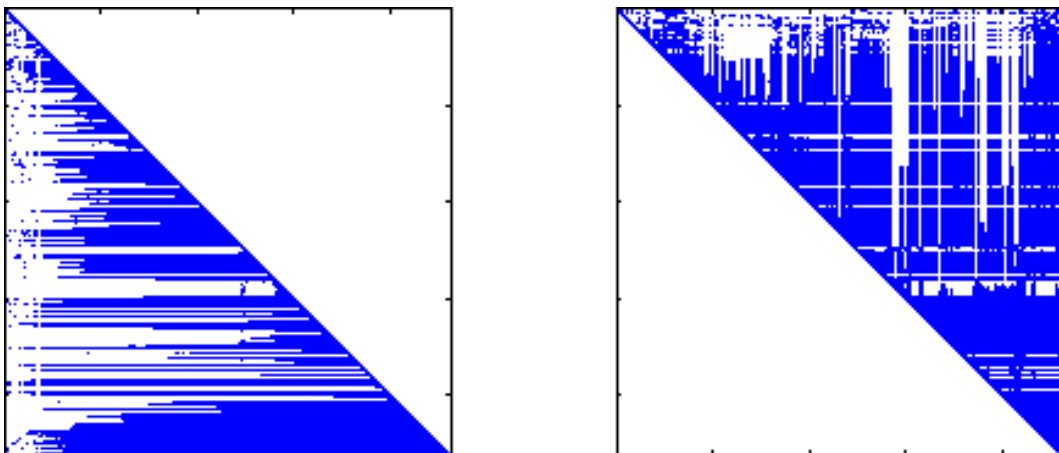


Figura 4.24: Padrão esparsa da matriz dos fatores LU da matriz do sistema linear do problema B-200 para o modelo de otimização 2. O fator L tem 356597 elementos não nulos, o fator U tem 433119 elementos não nulos.

6. As Figuras 4.21 a 4.24 mostram que os fatores LU tem consideravelmente mais elementos não nulos que a matriz original. Para resolver problemas maiores que os problemas de exemplo apresentados deverão ser feitas algumas modificações na implementação utilizada como por exemplo:

- Simetrização dos sistemas lineares e utilização de rotinas de análise e fatoração específicas para matrizes simétricas indefinidas.
- Redução do sistema linear.

Na Tabela 4.8 mostram-se os tempos requeridos para resolver os problemas correspondentes aos Modelos 1 e 2, para o Algoritmo HNP (já mostrado na Tabela 5.6) e para o Modelo 2 do algoritmo HNP com redução do sistema linear, chamado HNPR. A redução dos sistemas lineares foi feita de acordo com a seção 3.2.3.

Tabela 4.8: Tempo requerido em segundos.

Problema	B-10	Viga	B-200	B-200b	B-9x9	B-13x9	B-21x9
HNP - M1	0.6	0.8	53.1	4.4	69.1	367.0	3251.7
HNP - M2	1.1	0.8	204.8	6.4	149.3	669.2	-
HNPR - M2	0.9	0.8	166.5	5.0	106.3	349.8	-

Os resultados da Tabela 4.8 mostram que a redução do sistema linear é conveniente para reduzir o custo computacional da resolução do Modelo 2. Utilizando a técnica de redução dos sistemas lineares descrita na seção 3.2.3 o custo da resolução do Modelo 2 ficou próximo do custo da resolução do Modelo 1.

## 5 Conclusões

Foi desenvolvido um algoritmo para resolução de sistemas de equações não lineares que respeita em cada iteração um conjunto de restrições de desigualdades dadas. Baixo as mesmas hipóteses de convergência do método de Newton foi mostrado que o algoritmo converge à solução do problema com velocidade de convergência quadrática.

Baseados no algoritmo anterior foram desenvolvidos algoritmos para solução de problemas de otimização não linear, o Algoritmo 3.2 tem algumas das características do algoritmo FAIPA e do algoritmo Primal-dual, o Algoritmo 3.3 foi pensado para solução de problemas degenerados onde o Algoritmo 3.2 pode apresentar dificuldades.

No Capítulo 4 foram utilizados dois exemplos de otimização para testar os algoritmos propostos. De acordo com os resultados obtidos são feitas as seguintes conclusões:

1. Os Algoritmo 3.2 mostrou-se eficaz na solução dos exemplos apresentados.

Contudo, os resultados obtidos sugerem estudar os seguintes temas:

- Escolha do ponto inicial: Sobretudo das variáveis duais, já que em geral não é dado pelo usuário e, portanto devem ser definidas pelo algoritmo. No exemplo de otimização topológica de treliças a escolha das variáveis duais iniciais mostrou-se importante.
- Modificação da matriz hessiana: No mesmo exemplo obteve-se um algoritmo eficiente ao utilizar a matriz hessiana do problema. Por outro lado existe a conveniência teórica da utilização de uma matriz definida positiva no seu lugar. É de interesse então o estudo de técnicas, gerais ou específicas para este problema, de modificação da matriz hessiana para torná-la definida positiva.
- Estratégia de globalização: No mesmo problema teve de ser abandonada a busca linear na função potencial utilizada, mesmo que se obteve convergência para a solução do problema em todas as estruturas testadas,

deve ser estudada alguma técnica para assegurar a convergência global do método, como por exemplo, escolha de uma função potencial mais apropriada ou estratégias de redução não monótona da função potencial, tema que vem tendo muito interesse na atualidade.

2. Em quanto à eficiência do Algoritmo 3.2 pode-se dizer que se obteve em geral bons resultados respeito do número de iterações realizado. A eficiência do mesmo poderia ser maior se fosse mais eficiente a solução dos sistemas lineares. A redução dos sistemas lineares apresentada na seção 3.2.3 mostrou-se eficaz para reduzir o custo computacional. Espera-se obter melhores resultados logo de adicionar aos algoritmos técnicas de análise e solução de sistemas lineares indefinidos e esparsos. A correção de segunda ordem estudada na seção 3.2.2 mostrou-se eficaz tanto no exemplo da mínima superfície com obstáculo como nos exemplos maiores do problema de otimização de treliças.
3. O Algoritmo 3.3 não mostrou a eficácia do Algoritmo 3.2. Em problemas pequenos e degenerados como o do exemplo do Capítulo 3 o Algoritmo 3.3 mostrou os resultados desejados de redução do número de iterações necessários e de redução do número de condição dos sistemas lineares. Em problemas com um grande número de variáveis e restrições como os testados no Capítulo 4 o Algoritmo 3.3 apresenta dificuldades sobretudo nas primeiras iterações, os passos pequenos dados pelo algoritmo o fizeram em todos os casos menos eficiente que o Algoritmo 3.2.

O problema de otimização topológica de treliças tem interesse próprio, ele é um dos problemas mais estudados da otimização estrutural, os resultados obtidos na solução do Modelo 2 são promissores, porém, é necessário um estudo profundo do problema para estabelecer um método que mantendo a eficiência do método apresentado tenha convergência global a uma solução do problema.

## Referências

- [1] Benson, H.Y., Shanno, D.F., Vanderbei, R.J., *A comparative study of large-scale nonlinear optimization algorithms*, Technical Report ORFE 01-04, Department of Operations Research and Financial Engineering, Princeton University, Princeton, NJ 08544, Dec. 2001.
- [2] Dolan, E.D., Moré, J.J., *Benchmarking optimization software with performance profiles*, Mathematics and Computer Science Technical Report ANL/MCS-P861-1200, Argonne National Laboratory, Argonne, Illinois, USA, 2001.
- [3] Morales, J.L., Nocedal, J., Waltz, R.A., Liu, G., Goux, J.P., “Assessing the potential of interior methods for nonlinear optimization”, In: *Proceedings of the First Sandia Workshop on Large-Scale PDE-Constrained Optimization*, O. Ghattas, editor, Springer Verlag, Heidelberg, Berlin, New York, 2002.
- [4] Karmarkar, N., “A new polynomial-time algorithm for linear programming”, *Combinatorica*, v. 4, n. 4, pp 73–395, 1984.
- [5] Gould, N.I.M., “Some reflections on the current state of active-set and interior-point methods for constrained optimization”, *SIAG/OPT Views-and-News*, v. 14, n. 1, pp 2–7, Abr. 2003.
- [6] Tapia, R.A., *On the fundamental role of interior-point methodology in constrained optimization*, Technical Report TR97-09, Department of Computational and Applied Mathematics, Rice University, 1997.
- [7] Herskovits, J., “Feasible direction interior-point technique for nonlinear optimization”, *Journal of Optimization Theory and Applications*, v. 99, n. 1, pp 121–146, Oct. 1998.
- [8] Nocedal, J., Wright, S.J., *Numerical Optimization*, New York, Springer-Verlag, 1999.



- [9] Benson, H.Y., Shanno, D.F., Vanderbei, R.J., *Interior-point methods for non-convex nonlinear programming: complementarity constraints*, Technical Report ORFE 02-02. Department of Operations Research and Financial Engineering, Princeton University, Princeton, NJ, 08544. Jul. 2002.
- [10] Byrd, R.H., Nocedal, J., Waltz, R.A., “Feasible interior methods using slacks for nonlinear optimization”, *Computational Optimization and Applications* v. 26, n. 1, pp 35–61, Oct. 2003.
- [11] Maratos, N., *Exact penalty function algorithms for finite dimensional optimization problems with equality and inequality constraints*, Ph.D. thesis, Imperial College of Science and Technology, London, 1978.
- [12] El-Bakry, A.S., Tapia, R.A., Tsuchiya, T., Zhang, E., “On the formulation and theory of the Newton interior-point method for nonlinear programming”, *Journal of Optimization Theory and Applications*, v. 89, n. 3, pp 507–541, Jun. 1996.
- [13] Shanno, D.F., Vanderbei, R.J., “An interior-point algorithm for nonconvex nonlinear programming”. *Computational Optimization and Applications*, v. 13, pp 231–252, 1999.
- [14] Bazararaa, M.S., Sherali, H.D., Shetty, C.M., *Nonlinear Programming*, 2nd edition, Wiley, 1993.
- [15] Figueiredo, D.G., *Análise I*, 2 ed. Rio de Janeiro, LTC, 1996.
- [16] Griva, I., Shanno, D.V., Vanderbei, R.J., *Convergence analysis of a primal-dual interior-point method for nonlinear programming*, Technical Report ORFE-04-07, Department of Operations Research and Financial Engineering, Princeton University, Princeton, NJ 08544. Jul. 2004.
- [17] Byrd, R.H., Nocedal, J., Schnabel, R.B., “Representations of quasi-Newton matrices and their use in limited memory methods”, *Mathematical Programming*, v. 63, n. 2, pp 129–156, Jan. 1994.

- [18] Armand, P., Gilbert, J.Ch., Jan-Jégou, S., “A feasible BFGS interior point algorithm for solving strongly convex minimization problems”, *SIAM Journal on Optimization* v. 11, pp 199–222, 2000.
- [19] Canelas, A., Dubeux, V.J.C, Herskovits, J., “Um algoritmo de gradiente conjugado preconditionado para sistemas lineares baseado na técnica quase-Newton de memória limitada”, *XXV CILAMCE*, In CD, Recife, Brasil, Nov., 2004.
- [20] Sorensen, D.C., “An example concerning quasi-Newton estimates of a sparse Hessian”, *SIGNUM Newsletter*, v. 16, n. 2, pp 8–10, 1981.
- [21] Goulart, E., Herskovits, J., “Sparse quase-Newton matrix for large-scale problems in non-linear optimization”, *XXV CILAMCE*, In CD, Recife, Brasil, Nov., 2004.
- [22] Facchinei, F., Lucidi, S., “A Newton type algorithm for the solution of inequality constrained minimization problems”, In: *Operations Research Proceedings 1994*, U. Derigs, A. Bachem and A. Drexel (eds.), pp 33–39, Springer-Verlag, Berlin, Heidelberg, 1995.
- [23] Wright, S.J., “Constraint identification and algorithm stabilization for degenerate nonlinear programs”, *Mathematical Programming*, v. 95, n. 1, pp 137–160, Jan. 2003.
- [24] Chen, X.J., “Smoothing methods for complementarity problems and their applications: A survey”, *Journal of The Operations Research Society of Japan*, v. 43, n. 1, pp 32–47, Mar. 2000.
- [25] Vicente, L.N., Wright, S.J., “Local convergence of a primal-dual method for degenerate nonlinear programming”, *Computational Optimization and Applications* v. 22, n. 3, pp 311–328, Sep. 2002.
- [26] Bendsøe, M.P., *Optimization of structural topology, shape, and material*, ISBN 3-540-59057-9, Springer-Verlag, 1995.

- [27] Svanberg, K., “On local and global minima in structural optimization”. In: *New Directions in Optimum Structural Design*, pp 327–341, Atrek A., Gallagher R.H., Ragsdell K.M., Zienkiewicz O.C., ed., Wiley, New York, 1984.
- [28] Boyd, S., Vandenberghe L., “Semidefinite Programming”, *Siam Review*, v. 38, n. 1, pp 49–95, Mar. 1996.
- [29] Ben-Tal, A., Nemirovski, A., “Robust truss topology design via semidefinite programming”, *Siam Journal On Optimization*, v. 7, n. 4, pp 991–1016, Nov. 1997.
- [30] Canelas, A., Herskovits, J., “Robust truss topologic design by a new interior-point algorithm for non-smooth convex programming”, *XXV CIL-AMCE*, In CD, Recife, Brasil, Nov., 2004.
- [31] Passarella, W., *Um algoritmo de direções viáveis para otimização convexa não diferenciável*, D. Sc. thesis, COPPE/UFRJ, Rio de Janeiro, RJ-Brasil, 2005.
- [32] Gertz, M., Nocedal, J., Sartenaer, A., “A starting point strategy for nonlinear interior methods”, *Applied Mathematics Letters*, v. 17, n. 8, pp 945–952, Ago. 2004.

# Anexo 1

## Modelo 1

Nas tabelas seguintes detalham-se as características geométricas das treliças resolvidas e os volumes das barras para a solução obtida. O volume total das barras consideradas com volume positivo é dado no final da última coluna.

Problema B-10

Nodo	X	Y	-	Elemento	Nodo 1	Nodo 2	Volume
1	2	1		1	5	3	2.49935880e-001
2	2	0		2	6	4	1.25094525e-001
3	1	1		3	4	2	1.25092978e-001
4	1	0		4	6	3	2.49935057e-001
5	0	1		5	3	2	2.49935265e-001
6	0	0					
							9.99993705e-001

### Problema Viga

Nodo	X	Y	-	Elemento	Nodo 1	Nodo 2	Volume
1	0	0		1	1	2	2.01149425e-002
2	1	0		2	1	3	5.17241379e-002
3	1	1		3	2	3	5.74712644e-003
4	2	0		4	2	4	2.01149425e-002
5	2	1		5	3	4	4.02298851e-002
6	3	0		6	3	5	4.59770115e-002
7	3	1		7	4	5	1.43678161e-002
8	4	0		8	5	6	2.87356322e-002
9	4	1		9	5	7	6.03448276e-002
10	5	0		10	6	7	8.62068966e-003
11	5	1		11	6	8	1.43678161e-002
12	6	0		12	7	8	1.72413793e-002
13	6	1		13	7	9	6.89655172e-002
14	7	0		14	8	9	2.87356322e-003
15	7	1		15	8	10	2.29885057e-002
16	8	0		16	9	10	5.74712644e-003
17	8	1		17	9	11	7.18390805e-002
18	9	0		18	10	12	2.29885057e-002
19	9	1		19	10	13	5.74712644e-003
20	10	0		20	11	13	7.18390805e-002
				21	12	13	2.87356322e-003
				22	12	14	1.43678161e-002
				23	12	15	1.72413793e-002
				24	13	15	6.89655172e-002
				25	14	15	8.62068966e-003
				26	14	17	2.87356322e-002
				27	15	17	6.03448276e-002
				28	16	17	1.43678161e-002
				29	16	18	2.01149425e-002
				30	16	19	4.02298851e-002
				31	17	19	4.59770115e-002
				32	18	19	5.74712644e-003
				33	18	20	2.01149425e-002
				34	19	20	5.17241379e-002
							1.00000000e+000

Problema B-200

Nodo	X	Y	-	Elemento	Nodo 1	Nodo 2	Volume
1	0	1800		1	1	2	2.62944552e-003
2	240	1800		2	2	3	2.62944552e-003
3	480	1800		3	3	4	2.62944552e-003
4	720	1800		4	4	5	2.62944552e-003
5	960	1800		5	1	7	3.40575376e-003
6	0	1656		6	2	8	2.00995332e-003
7	120	1656		7	3	9	2.35820754e-004
8	240	1656		8	3	10	1.87102289e-003
9	360	1656		9	3	11	2.35820754e-004
10	480	1656		10	4	12	2.00995332e-003
11	600	1656		11	5	13	3.40575376e-003
12	720	1656		12	6	7	1.18437636e-004
13	840	1656		13	7	8	2.02571163e-005
14	960	1656		14	8	9	2.02571163e-005
15	0	1512		15	9	10	1.65764134e-005
16	240	1512		16	10	11	1.65764134e-005
17	480	1512		17	11	12	2.02571163e-005
18	720	1512		18	12	13	2.02571163e-005
19	960	1512		19	13	14	1.18437636e-004
20	0	1368		20	6	15	2.00995360e-003
21	120	1368		21	7	15	1.53586561e-004
22	240	1368		22	7	16	3.53092945e-003
23	360	1368		23	8	16	4.01990665e-003
24	480	1368		24	9	16	2.46688643e-004
25	600	1368		25	9	17	1.16782338e-005
26	720	1368		26	10	17	3.88097621e-003
27	840	1368		27	11	17	1.16782338e-005
28	960	1368		28	11	18	2.46688643e-004
29	0	1224		29	12	18	4.01990665e-003
30	240	1224		30	13	18	3.53092945e-003
31	480	1224		31	13	19	1.53586561e-004
32	720	1224		32	14	19	2.00995360e-003
33	960	1224		33	15	16	4.17653392e-003
34	0	1080		34	16	17	1.41015290e-003
35	120	1080		35	17	18	1.41015290e-003
36	240	1080		36	18	19	4.17653392e-003
37	360	1080		37	15	20	7.63003669e-004
38	480	1080		38	15	21	5.42517407e-003
39	600	1080		39	16	22	8.18664256e-003
40	720	1080		40	16	23	5.18567633e-005
41	840	1080		41	17	23	4.45265903e-004
42	960	1080		42	17	24	5.62654989e-003
43	0	936		43	17	25	4.45265903e-004
44	240	936		44	18	25	5.18567633e-005
45	480	936		45	18	26	8.18664256e-003
46	720	936		46	19	27	5.42517407e-003
47	960	936		47	19	28	7.63003669e-004
48	0	792		48	20	21	1.18437636e-004
49	120	792		49	21	22	6.47198900e-006
50	240	792		50	22	23	6.47198900e-006
51	360	792		51	23	24	3.12625173e-006
52	480	792		52	24	25	3.12625173e-006
53	600	792		53	25	26	6.47198900e-006

Problema B-200

Nodo	X	Y	-	Elemento	Nodo 1	Nodo 2	Volume
54	720	792		54	26	27	6.47198900e-006
55	840	792		55	27	28	1.18437636e-004
56	960	792		56	20	29	2.76108096e-003
57	0	648		57	21	29	1.47440985e-004
58	240	648		58	21	30	5.53849358e-003
59	480	648		59	22	30	1.01964645e-002
60	720	648		60	23	30	4.49790308e-004
61	960	648		61	23	31	5.54904398e-005
62	0	504		62	24	31	7.63650321e-003
63	120	504		63	25	31	5.54904398e-005
64	240	504		64	25	32	4.49790308e-004
65	360	504		65	26	32	1.01964645e-002
66	480	504		66	27	32	5.53849358e-003
67	600	504		67	27	33	1.47440985e-004
68	720	504		68	28	33	2.76108096e-003
69	840	504		69	29	30	4.42462629e-003
70	960	504		70	32	33	4.42462629e-003
71	0	360		71	29	34	1.36006030e-003
72	240	360		72	29	35	5.70802715e-003
73	480	360		73	30	36	1.56707601e-002
74	720	360		74	30	37	8.85002085e-004
75	960	360		75	31	37	6.32485692e-005
76	240	0		76	31	38	9.60040932e-003
77	720	0		77	31	39	6.32485692e-005
				78	32	39	8.85002085e-004
				79	32	40	1.56707601e-002
				80	33	41	5.70802715e-003
				81	33	42	1.36006030e-003
				82	34	35	1.18437636e-004
				83	35	36	4.59491502e-006
				84	36	37	4.59491502e-006
				85	39	40	4.59491502e-006
				86	40	41	4.59491502e-006
				87	41	42	1.18437636e-004
				88	34	43	3.35427584e-003
				89	35	43	1.46474033e-004
				90	35	44	5.81758690e-003
				91	36	44	1.76789767e-002
				92	37	44	6.79180732e-005
				93	37	45	8.87905668e-004
				94	38	45	1.16103626e-002
				95	39	45	8.87905668e-004
				96	39	46	6.79180732e-005
				97	40	46	1.76789767e-002
				98	41	46	5.81758690e-003
				99	41	47	1.46474033e-004
				100	42	47	3.35427584e-003
				101	43	44	4.74045531e-003
				102	46	47	4.74045531e-003
				103	43	48	1.81599259e-003
				104	43	49	6.11437160e-003
				105	44	50	2.31366356e-002
				106	45	51	1.05018463e-003

Problema B-200

Nodo	X	Y	-	Elemento	Nodo 1	Nodo 2	Volume
				107	45	52	1.28398436e-002
				108	45	53	1.05018463e-003
				109	46	54	2.31366356e-002
				110	47	55	6.11437160e-003
				111	47	56	1.81599259e-003
				112	48	49	1.18437636e-004
				113	55	56	1.18437636e-004
				114	48	57	3.76257252e-003
				115	49	57	1.44493919e-004
				116	49	58	6.22596963e-003
				117	50	58	2.51459335e-002
				118	51	58	1.05018464e-003
				119	52	59	1.48497969e-002
				120	53	60	1.05018464e-003
				121	54	60	2.51459335e-002
				122	55	60	6.22596963e-003
				123	55	61	1.44493919e-004
				124	56	61	3.76257252e-003
				125	57	58	4.70006359e-003
				126	60	61	4.70006359e-003
				127	57	62	2.22463879e-003
				128	57	63	6.03627802e-003
				129	58	64	3.12521745e-002
				130	58	65	1.76930802e-003
				131	59	66	1.68597502e-002
				132	60	67	1.76930802e-003
				133	60	68	3.12521745e-002
				134	61	69	6.03627802e-003
				135	61	70	2.22463879e-003
				136	62	63	1.18437636e-004
				137	69	70	1.18437636e-004
				138	62	71	4.17204408e-003
				139	63	71	1.44493921e-004
				140	63	72	6.14050776e-003
				141	64	72	3.32590065e-002
				142	65	73	1.76930804e-003
				143	66	73	1.88697035e-002
				144	67	73	1.76930804e-003
				145	68	74	3.32590065e-002
				146	69	74	6.14050776e-003
				147	69	75	1.44493921e-004
				148	70	75	4.17204408e-003
				149	71	72	6.57672082e-003
				150	72	73	1.94062663e-003
				151	73	74	1.94062663e-003
				152	74	75	6.57672082e-003
				153	71	76	2.20553827e-002
				154	72	76	9.71993153e-002
				155	73	76	3.85209246e-002
				156	73	77	3.85209246e-002
				157	74	77	9.71993153e-002
				158	75	77	2.20553827e-002
							9.99999999e-001



Problema B-200b

Nodo	X	Y	-	Elemento	Nodo 1	Nodo 2	Volume
1	0	1800		1	1	2	2.84329339e-003
2	240	1800		2	2	3	2.84329342e-003
3	480	1800		3	3	4	2.84329342e-003
4	720	1800		4	4	5	2.84329339e-003
5	960	1800		5	1	7	3.46881793e-003
6	0	1656		6	2	8	2.04717166e-003
7	120	1656		7	3	10	2.04717158e-003
8	240	1656		8	4	12	2.04717166e-003
9	360	1656		9	5	13	3.46881793e-003
10	480	1656		10	6	15	2.04717198e-003
11	600	1656		11	7	16	3.46881861e-003
12	720	1656		12	8	16	4.09434323e-003
13	840	1656		13	10	17	4.09434314e-003
14	960	1656		14	12	18	4.09434323e-003
15	0	1512		15	13	18	3.46881861e-003
16	240	1512		16	14	19	2.04717198e-003
17	480	1512		17	15	16	5.41579542e-003
18	720	1512		18	16	17	2.57250159e-003
19	960	1512		19	17	18	2.57250159e-003
20	0	1368		20	18	19	5.41579542e-003
21	120	1368		21	15	20	1.94970016e-004
22	240	1368		22	15	21	6.60727113e-003
23	360	1368		23	16	22	8.18868648e-003
24	480	1368		24	17	24	6.14151460e-003
25	600	1368		25	18	26	8.18868648e-003
26	720	1368		26	19	27	6.60727113e-003
27	840	1368		27	19	28	1.94970016e-004
28	960	1368		28	20	29	2.24214158e-003
29	0	1224		29	21	30	6.60727155e-003
30	240	1224		30	22	30	1.02358580e-002
31	480	1224		31	24	31	8.18868616e-003
32	720	1224		32	26	32	1.02358580e-002
33	960	1224		33	27	32	6.60727155e-003
34	0	1080		34	28	33	2.24214158e-003
35	120	1080		35	29	30	5.41579678e-003
36	240	1080		36	32	33	5.41579678e-003
37	360	1080		37	29	34	3.89938954e-004
38	480	1080		38	29	35	6.60727252e-003
39	600	1080		39	30	36	1.61824030e-002
40	720	1080		40	31	38	1.02358576e-002
41	840	1080		41	32	40	1.61824030e-002
42	960	1080		42	33	41	6.60727252e-003
43	0	936		43	33	42	3.89938954e-004
44	240	936		44	34	43	2.43711052e-003
45	480	936		45	35	44	6.60727282e-003
46	720	936		46	36	44	1.82295746e-002
47	960	936		47	38	45	1.22830292e-002
48	0	792		48	40	46	1.82295746e-002
49	120	792		49	41	46	6.60727282e-003
50	240	792		50	42	47	2.43711052e-003

Problema B-200b

Nodo	X	Y	-	Elemento	Nodo 1	Nodo 2	Volume
51	360	792		51	43	44	5.41579753e-003
52	480	792		52	46	47	5.41579753e-003
53	600	792		53	43	48	5.84907487e-004
54	720	792		54	43	49	6.60727333e-003
55	840	792		55	44	50	2.41761203e-002
56	960	792		56	45	52	1.43302006e-002
57	0	648		57	46	54	2.41761203e-002
58	240	648		58	47	55	6.60727333e-003
59	480	648		59	47	56	5.84907487e-004
60	720	648		60	48	57	2.63207905e-003
61	960	648		61	49	58	6.60727355e-003
62	0	504		62	50	58	2.62232919e-002
63	120	504		63	52	59	1.63773722e-002
64	240	504		64	54	60	2.62232919e-002
65	360	504		65	55	60	6.60727355e-003
66	480	504		66	56	61	2.63207905e-003
67	600	504		67	57	58	5.41579802e-003
68	720	504		68	60	61	5.41579802e-003
69	840	504		69	57	62	7.79875749e-004
70	960	504		70	57	63	6.60727385e-003
71	0	360		71	58	64	3.21698381e-002
72	240	360		72	59	66	1.84245436e-002
73	480	360		73	60	68	3.21698381e-002
74	720	360		74	61	69	6.60727385e-003
75	960	360		75	61	70	7.79875749e-004
76	240	0		76	62	71	2.82704732e-003
77	720	0		77	63	72	6.60727403e-003
				78	64	72	3.42170097e-002
				79	66	73	2.04717152e-002
				80	68	74	3.42170097e-002
				81	69	74	6.60727403e-003
				82	70	75	2.82704732e-003
				83	71	72	5.41579840e-003
				84	74	75	5.41579840e-003
				85	71	76	1.76013453e-002
				86	72	76	1.00408890e-001
				87	73	76	4.06591009e-002
				88	73	77	4.06591009e-002
				89	74	77	1.00408890e-001
				90	75	77	1.76013453e-002
							9.99999974e-001

Problema B-9x9

Nodo	X	Y	-	Elemento	Nodo 1	Nodo 2	Volume
1	0	0		1	1	11	8.60288432e-003
2	0	1		2	1	30	1.74239024e-002
3	0	2		3	1	48	2.05965909e-001
4	0	3		4	1	49	4.96580694e-002
5	0	4		5	9	17	8.60288108e-003
6	0	5		6	9	34	1.74238744e-002
7	0	6		7	9	51	4.96581245e-002
8	0	7		8	9	52	2.05965909e-001
9	0	8		9	11	21	8.60288432e-003
10	1	0		10	17	25	8.60288108e-003
11	1	1		11	21	31	8.60288432e-003
12	1	2		12	25	33	8.60288108e-003
13	1	3		13	30	59	1.74239024e-002
14	1	4		14	31	41	8.60288431e-003
15	1	5		15	33	41	8.60288108e-003
16	1	6		16	34	59	1.74238744e-002
17	1	7		17	41	48	7.10227271e-003
18	1	8		18	41	49	2.92106291e-003
19	2	0		19	41	51	2.92106615e-003
20	2	1		20	41	52	7.10227271e-003
21	2	2		21	48	67	8.52272727e-002
22	2	3		22	49	59	1.16842516e-002
23	2	4		23	51	59	1.16842646e-002
24	2	5		24	52	69	8.52272727e-002
25	2	6		25	59	67	1.70454545e-002
26	2	7		26	59	69	1.70454545e-002
27	2	8		27	67	77	5.11363636e-002
28	3	0		28	69	77	5.11363636e-002
29	3	1					
30	3	2					
31	3	3					
32	3	4					
33	3	5					
34	3	6					
35	3	7					
36	3	8					
37	4	0					
38	4	1					
39	4	2					
40	4	3					

Problema B-9x9

Nodo	X	Y	-	Elemento	Nodo 1	Nodo 2	Volume
41	4	4					
42	4	5					
43	4	6					
44	4	7					
45	4	8					
46	5	0					
47	5	1					
48	5	2					
49	5	3					
50	5	4					
51	5	5					
52	5	6					
53	5	7					
54	5	8					
55	6	0					
56	6	1					
57	6	2					
58	6	3					
59	6	4					
60	6	5					
61	6	6					
62	6	7					
63	6	8					
64	7	0					
65	7	1					
66	7	2					
67	7	3					
68	7	4					
69	7	5					
70	7	6					
71	7	7					
72	7	8					
73	8	0					
74	8	1					
75	8	2					
76	8	3					
77	8	4					
78	8	5					
79	8	6					
80	8	7					
81	8	8					
							1.00000000e+000

Problema B-13x9

Nodo	X	Y	-	Elemento	Nodo 1	Nodo 2	Volume
1	0	0		1	1	11	1.07524895e-003
2	0	1		2	1	20	5.09498509e-003
3	0	2		3	1	30	2.58048839e-003
4	0	3		4	1	38	2.88715821e-002
5	0	4		5	1	48	1.23128806e-002
6	0	5		6	1	49	8.38036753e-003
7	0	6		7	1	50	5.22235970e-002
8	0	7		8	1	65	1.52849552e-001
9	0	8		9	3	13	7.27855010e-004
10	1	0		10	7	15	7.27855010e-004
11	1	1		11	9	17	1.07524681e-003
12	1	2		12	9	26	5.09498509e-003
13	1	3		13	9	34	2.58046980e-003
14	1	4		14	9	44	2.88715821e-002
15	1	5		15	9	50	5.22235970e-002
16	1	6		16	9	51	8.38040399e-003
17	1	7		17	9	52	1.23128806e-002
18	1	8		18	9	71	1.52849552e-001
19	2	0		19	11	21	1.07524895e-003
20	2	1		20	13	23	7.27855009e-004
21	2	2		21	15	23	7.27855009e-004
22	2	3		22	17	25	1.07524680e-003
23	2	4		23	20	39	5.09498508e-003
24	2	5		24	21	31	1.07524895e-003
25	2	6		25	23	31	7.27855009e-004
26	2	7		26	23	33	7.27855009e-004
27	2	8		27	25	33	1.07524680e-003
28	3	0		28	26	43	5.09498508e-003
29	3	1		29	30	59	2.58048838e-003
30	3	2		30	31	38	1.21309168e-003
31	3	3		31	31	41	8.32630610e-004
32	3	4		32	33	41	8.32628466e-004
33	3	5		33	33	44	1.21309168e-003
34	3	6		34	34	59	2.58046980e-003
35	3	7		35	38	66	2.18356503e-002
36	3	8		36	39	58	5.09498508e-003
37	4	0		37	41	48	4.24582087e-004
38	4	1		38	41	49	4.92962796e-004
39	4	2		39	41	51	4.92964941e-004
40	4	3		40	41	52	4.24582087e-004

Problema B-13x9

Nodo	X	Y	-	Elemento	Nodo 1	Nodo 2	Volume
41	4	4		41	43	60	5.09498508e-003
42	4	5		42	44	70	2.18356503e-002
43	4	6		43	48	67	5.09498507e-003
44	4	7		44	49	59	1.97185118e-003
45	4	8		45	50	58	1.01899702e-002
46	5	0		46	50	60	1.01899702e-002
47	5	1		47	51	59	1.97185976e-003
48	5	2		48	52	69	5.09498507e-003
49	5	3		49	58	65	1.52849552e-002
50	5	4		50	59	66	2.18356503e-003
51	5	5		51	59	67	1.01899701e-003
52	5	6		52	59	69	1.01899701e-003
53	5	7		53	59	70	2.18356503e-003
54	5	8		54	60	71	1.52849552e-002
55	6	0		55	65	84	4.58548657e-002
56	6	1		56	66	85	1.52849552e-002
57	6	2		57	67	77	3.05699104e-003
58	6	3		58	69	77	3.05699104e-003
59	6	4		59	70	87	1.52849552e-002
60	6	5		60	71	88	4.58548657e-002
61	6	6		61	77	85	3.05699104e-003
62	6	7		62	77	87	3.05699104e-003
63	6	8		63	84	103	4.58548657e-002
64	7	0		64	85	95	9.17097313e-003
65	7	1		65	87	95	9.17097313e-003
66	7	2		66	88	105	4.58548657e-002
67	7	3		67	95	103	9.17097313e-003
68	7	4		68	95	105	9.17097313e-003
69	7	5		69	103	113	2.75129194e-002
70	7	6		70	105	113	2.75129194e-002
71	7	7					
72	7	8					
73	8	0					
74	8	1					
75	8	2					
76	8	3					
77	8	4					
78	8	5					
79	8	6					
80	8	7					

Problema B-13x9

Nodo	X	Y	-	Elemento	Nodo 1	Nodo 2	Volume
81	8	8					
82	9	0					
83	9	1					
84	9	2					
85	9	3					
86	9	4					
87	9	5					
88	9	6					
89	9	7					
90	9	8					
91	10	0					
92	10	1					
93	10	2					
94	10	3					
95	10	4					
96	10	5					
97	10	6					
98	10	7					
99	10	8					
100	11	0					
101	11	1					
102	11	2					
103	11	3					
104	11	4					
105	11	5					
106	11	6					
107	11	7					
108	11	8					
109	12	0					
110	12	1					
111	12	2					
112	12	3					
113	12	4					
114	12	5					
115	12	6					
116	12	7					
117	12	8					
							9.99999999e-001

Problema B-21x9

Nodo	X	Y	-	Elemento	Nodo 1	Nodo 2	Volume
1	0	0		1	1	10	1.54850909e-002
2	0	1		2	1	11	5.56287826e-004
3	0	2		3	1	20	4.34046029e-003
4	0	3		4	1	29	1.48856220e-002
5	0	4		5	1	30	1.80318388e-003
6	0	5		6	1	48	5.12682648e-003
7	0	6		7	1	49	7.05258495e-003
8	0	7		8	1	50	1.76327854e-002
9	0	8		9	1	65	2.43589719e-003
10	1	0		10	1	74	3.23188661e-002
11	1	1		11	9	17	5.56287815e-004
12	1	2		12	9	18	1.54850909e-002
13	1	3		13	9	26	4.34046029e-003
14	1	4		14	9	34	1.80318378e-003
15	1	5		15	9	35	1.48856220e-002
16	1	6		16	9	50	1.76327854e-002
17	1	7		17	9	51	7.05258515e-003
18	1	8		18	9	52	5.12682648e-003
19	2	0		19	9	71	2.43589718e-003
20	2	1		20	9	80	3.23188661e-002
21	2	2		21	10	19	1.54850909e-002
22	2	3		22	11	21	5.56287777e-004
23	2	4		23	17	25	5.56287765e-004
24	2	5		24	18	27	1.54850909e-002
25	2	6		25	19	28	1.54850909e-002
26	2	7		26	20	39	4.34046017e-003
27	2	8		27	21	31	5.56287751e-004
28	3	0		28	25	33	5.56287739e-004
29	3	1		29	26	43	4.34046017e-003
30	3	2		30	27	36	1.54850909e-002
31	3	3		31	28	37	1.54850908e-002
32	3	4		32	29	57	1.48856219e-002
33	3	5		33	30	59	1.80318380e-003
34	3	6		34	31	41	5.56287716e-004
35	3	7		35	33	41	5.56287704e-004
36	3	8		36	34	59	1.80318370e-003
37	4	0		37	35	61	1.48856219e-002
38	4	1		38	36	45	1.54850908e-002
39	4	2		39	37	46	1.54850907e-002
40	4	3		40	39	58	4.34046011e-003
41	4	4		41	41	48	1.76787068e-004
42	4	5		42	41	49	4.14858048e-004
43	4	6		43	41	51	4.14858059e-004
44	4	7		44	41	52	1.76787068e-004
45	4	8		45	43	60	4.34046011e-003
46	5	0		46	45	54	1.54850907e-002
47	5	1		47	46	55	1.54850905e-002
48	5	2		48	48	67	2.12144542e-003
49	5	3		49	49	59	1.65943172e-003
50	5	4		50	50	57	1.48856219e-003



Problema B-21x9

Nodo	X	Y	-	Elemento	Nodo 1	Nodo 2	Volume
51	5	5		51	50	58	2.24969376e-003
52	5	6		52	50	60	2.24969376e-003
53	5	7		53	50	61	1.48856219e-003
54	5	8		54	51	59	1.65943177e-003
55	6	0		55	52	69	2.12144542e-003
56	6	1		56	54	63	1.54850905e-002
57	6	2		57	55	64	1.54850903e-002
58	6	3		58	57	76	1.04199353e-002
59	6	4		59	58	64	8.96315650e-004
60	6	5		60	58	65	1.70512580e-003
61	6	6		61	58	68	2.08398776e-003
62	6	7		62	59	67	2.21425736e-003
63	6	8		63	59	69	2.21425736e-003
64	7	0		64	60	68	2.08398776e-003
65	7	1		65	60	71	1.70512579e-003
66	7	2		66	60	72	8.96315661e-004
67	7	3		67	61	78	1.04199353e-002
68	7	4		68	63	72	1.54850903e-002
69	7	5		69	64	65	7.30768045e-004
70	7	6		70	64	73	1.13968083e-002
71	7	7		71	64	101	1.69942562e-002
72	7	8		72	67	74	2.98327996e-003
73	8	0		73	67	77	6.76211119e-004
74	8	1		74	68	76	2.08398765e-003
75	8	2		75	68	78	2.08398765e-003
76	8	3		76	69	77	6.76211119e-004
77	8	4		77	69	80	2.98327996e-003
78	8	5		78	71	72	7.30768042e-004
79	8	6		79	72	81	1.13968083e-002
80	8	7		80	72	107	1.69942562e-002
81	8	8		81	73	82	1.13968074e-002
82	9	0		82	74	93	8.45262650e-003
83	9	1		83	76	86	6.25196092e-003
84	9	2		84	77	85	6.76211094e-004
85	9	3		85	77	87	6.76211094e-004
86	9	4		86	78	86	6.25196092e-003
87	9	5		87	80	97	8.45262650e-003
88	9	6		88	81	90	1.13968074e-002
89	9	7		89	82	91	1.13968059e-002
90	9	8		90	85	93	6.76210132e-004
91	10	0		91	86	101	1.35459089e-002
92	10	1		92	86	107	1.35459089e-002
93	10	2		93	87	97	6.76210132e-004
94	10	3		94	90	99	1.13968059e-002
95	10	4		95	91	100	1.13968058e-002
96	10	5		96	93	122	1.31860971e-002
97	10	6		97	97	122	1.31860971e-002
98	10	7		98	99	108	1.13968058e-002
99	10	8		99	100	101	2.84919996e-003
100	11	0		100	100	137	4.84363999e-002

Problema B-21x9

Nodo	X	Y	-	Elemento	Nodo 1	Nodo 2	Volume
101	11	1		101	101	130	8.29686990e-003
102	11	2		102	107	108	2.84919996e-003
103	11	3		103	107	132	8.29686990e-003
104	11	4		104	108	143	4.84363999e-002
105	11	5		105	122	130	4.05726075e-003
106	11	6		106	122	132	4.05726075e-003
107	11	7		107	130	137	5.69839998e-003
108	11	8		108	130	140	2.05142382e-003
109	12	0		109	132	140	2.05142382e-003
110	12	1		110	132	143	5.69839998e-003
111	12	2		111	137	156	2.56428000e-002
112	12	3		112	140	148	2.05142385e-003
113	12	4		113	140	150	2.05142385e-003
114	12	5		114	143	160	2.56428000e-002
115	12	6		115	148	156	2.05142386e-003
116	12	7		116	150	160	2.05142386e-003
117	12	8		117	156	185	4.00027656e-002
118	13	0		118	160	185	4.00027656e-002
119	13	1					
120	13	2					
121	13	3					
122	13	4					
123	13	5					
124	13	6					
125	13	7					
126	13	8					
127	14	0					
128	14	1					
129	14	2					
130	14	3					
131	14	4					
132	14	5					
133	14	6					
134	14	7					
135	14	8					
136	15	0					
137	15	1					
138	15	2					
139	15	3					
140	15	4					
141	15	5					
142	15	6					
143	15	7					
144	15	8					
145	16	0					
146	16	1					
147	16	2					
148	16	3					
149	16	4					
150	16	5					

Problema B-21x9

Nodo	X	Y	-	Elemento	Nodo 1	Nodo 2	Volume
151	16	6					
152	16	7					
153	16	8					
154	17	0					
155	17	1					
156	17	2					
157	17	3					
158	17	4					
159	17	5					
160	17	6					
161	17	7					
162	17	8					
163	18	0					
164	18	1					
165	18	2					
166	18	3					
167	18	4					
168	18	5					
169	18	6					
170	18	7					
171	18	8					
172	19	0					
173	19	1					
174	19	2					
175	19	3					
176	19	4					
177	19	5					
178	19	6					
179	19	7					
180	19	8					
181	20	0					
182	20	1					
183	20	2					
184	20	3					
185	20	4					
186	20	5					
187	20	6					
188	20	7					
189	20	8					
							9.99999702e-001

## Modelo 2

Nas tabelas seguintes detalham-se as características geométricas das treliças resolvidas e os volumes das barras para a solução obtida. O volume total das barras consideradas com volume positivo é dado no final da última coluna. Para comparação com o modelo anterior os valores finais dos volumes foram divididos pelo volume total achado. Para estes volumes é dado o valor da tensão máxima. Em todos os problemas foi considerado  $-\sigma_{min} = \sigma_{max}$ .

Problema B-10  $\sigma_{max} = 1.00000000e-003$

Elemento	Nodo 1	Nodo 2	Volume
1	5	3	1.99999994e-001
2	6	4	1.99999997e-001
3	4	2	1.99999999e-001
4	6	3	1.99999988e-001
5	3	2	1.99999998e-001
			9.99999976e-001

Problema Viga  $\sigma_{max} = 1.74000000e-002$

Elemento	Nodo 1	Nodo 2	Volume
1	1	2	2.01149425e-002
2	1	3	5.17241379e-002
3	2	3	5.74712644e-003
4	2	4	2.01149425e-002
5	3	4	4.02298851e-002
6	3	5	4.59770115e-002
7	4	5	1.43678161e-002
8	5	6	2.87356322e-002
9	5	7	6.03448276e-002
10	6	7	8.62068966e-003
11	6	8	1.43678161e-002
12	7	8	1.72413793e-002
13	7	9	6.89655172e-002
14	8	9	2.87356323e-003
15	8	10	2.29885057e-002
16	9	10	5.74712644e-003
17	9	11	7.18390804e-002
18	10	12	2.29885057e-002
19	10	13	5.74712644e-003
20	11	13	7.18390804e-002
21	12	13	2.87356323e-003
22	12	14	1.43678161e-002
23	12	15	1.72413793e-002
24	13	15	6.89655172e-002
25	14	15	8.62068966e-003
26	14	17	2.87356322e-002
27	15	17	6.03448276e-002
28	16	17	1.43678161e-002
29	16	18	2.01149425e-002
30	16	19	4.02298851e-002
31	17	19	4.59770115e-002
32	18	19	5.74712644e-003
33	18	20	2.01149425e-002
34	19	20	5.17241379e-002
			1.00000000e+000

Problema B-200  $\sigma_{max} = 7.73561529e-001$

Elemento	Nodo 1	Nodo 2	Volume
1	1	2	5.53788090e-004
2	2	3	5.53788221e-004
3	3	4	5.53788221e-004
4	4	5	5.53788090e-004
5	1	6	1.50785848e-003
6	1	7	7.68401908e-004
7	2	8	1.86152051e-003
8	3	9	1.89633394e-004
9	3	10	1.78468701e-003
10	3	11	1.89633394e-004
11	4	12	1.86152051e-003
12	5	13	7.68401908e-004
13	5	14	1.50785848e-003
14	6	7	1.55126869e-004
15	7	8	1.05754450e-004
16	8	9	1.05754450e-004
17	9	10	1.06260634e-004
18	10	11	1.06260634e-004
19	11	12	1.05754450e-004
20	12	13	1.05754450e-004
21	13	14	1.55126869e-004
22	6	15	3.36937816e-003
23	7	15	2.37111666e-004
24	7	16	1.00551346e-003
25	8	16	3.72304018e-003
26	9	16	2.11394225e-004
27	9	17	3.55185032e-005
28	10	17	3.64620662e-003
29	11	17	3.55185032e-005
30	11	18	2.11394225e-004
31	12	18	3.72304018e-003
32	13	18	1.00551346e-003
33	13	19	2.37111666e-004
34	14	19	3.36937816e-003
35	15	16	1.34160615e-003
36	16	17	7.35593827e-004
37	17	18	7.35593827e-004
38	18	19	1.34160615e-003
39	15	20	4.28475060e-003
40	15	21	2.03583685e-003
41	16	22	6.13303737e-003
42	16	23	2.09014261e-005
43	17	23	4.77146945e-004
44	17	24	5.32412603e-003
45	17	25	4.77146945e-004
46	18	25	2.09014261e-005
47	18	26	6.13303737e-003
48	19	27	2.03583685e-003
49	19	28	4.28475060e-003
50	20	21	1.55126869e-004

Problema B-200  $\sigma_{max} = 7.73561529e-001$

Elemento	Nodo 1	Nodo 2	Volume
51	21	22	1.21538338e-004
52	22	23	1.21538338e-004
53	23	24	1.22170846e-004
54	24	25	1.22170846e-004
55	25	26	1.21538338e-004
56	26	27	1.21538338e-004
57	27	28	1.55126869e-004
58	20	29	6.14627027e-003
59	21	29	2.65988098e-004
60	21	30	2.30182474e-003
61	22	30	7.99455704e-003
62	23	30	4.93557199e-004
63	23	31	4.79893515e-005
64	24	31	7.18564568e-003
65	25	31	4.79893515e-005
66	25	32	4.93557199e-004
67	26	32	7.99455704e-003
68	27	32	2.30182474e-003
69	27	33	2.65988098e-004
70	28	33	6.14627027e-003
71	29	30	2.05507672e-003
72	30	31	6.27269554e-004
73	31	32	6.27269554e-004
74	32	33	2.05507672e-003
75	29	34	6.60525577e-003
76	29	35	3.06711577e-003
77	30	36	1.12510464e-002
78	30	37	3.90583585e-004
79	31	37	3.70530366e-004
80	31	38	8.90720838e-003
81	31	39	3.70530366e-004
82	32	39	3.90583585e-004
83	32	40	1.12510464e-002
84	33	41	3.06711577e-003
85	33	42	6.60525577e-003
86	34	35	1.55126869e-004
87	35	36	1.32755607e-004
88	36	37	1.32755607e-004
89	37	38	1.31365172e-004
90	38	39	1.31365172e-004
91	39	40	1.32755607e-004
92	40	41	1.32755607e-004
93	41	42	1.55126869e-004
94	34	43	8.46677544e-003
95	35	43	2.93086617e-004
96	35	44	3.36020188e-003
97	36	44	1.31125661e-002
98	37	44	3.80372463e-004
99	37	45	4.14024032e-004
100	38	45	1.07687280e-002

Problema B-200  $\sigma_{max} = 7.73561529e-001$

Elemento	Nodo 1	Nodo 2	Volume
101	39	45	4.14024032e-004
102	39	46	3.80372463e-004
103	40	46	1.31125661e-002
104	41	46	3.36020188e-003
105	41	47	2.93086617e-004
106	42	47	8.46677544e-003
107	43	44	3.78309221e-003
108	44	45	1.60596880e-003
109	45	46	1.60596880e-003
110	46	47	3.78309221e-003
111	43	48	7.83962994e-003
112	43	49	5.28696786e-003
113	44	50	1.66078073e-002
114	45	51	1.24011327e-003
115	45	52	1.21643964e-002
116	45	53	1.24011327e-003
117	46	54	1.66078073e-002
118	47	55	5.28696786e-003
119	47	56	7.83962994e-003
120	48	49	1.55126869e-004
121	49	50	1.53219161e-004
122	50	51	1.53219161e-004
123	51	52	1.53990301e-004
124	52	53	1.53990301e-004
125	53	54	1.53219161e-004
126	54	55	1.53219161e-004
127	55	56	1.55126869e-004
128	48	57	9.70114961e-003
129	49	57	3.14653776e-004
130	49	58	5.60162116e-003
131	50	58	1.84693270e-002
132	51	58	1.27284744e-003
133	51	59	3.10835737e-005
134	52	59	1.40259161e-002
135	53	59	3.10835737e-005
136	53	60	1.27284744e-003
137	54	60	1.84693270e-002
138	55	60	5.60162116e-003
139	55	61	3.14653776e-004
140	56	61	9.70114961e-003
141	57	58	3.59410467e-003
142	60	61	3.59410467e-003
143	57	62	8.99669654e-003
144	57	63	5.04784427e-003
145	58	64	2.41352495e-002
146	58	65	1.88146979e-003
147	59	65	3.44641534e-005
148	59	66	1.58765664e-002
149	59	67	3.44641534e-005
150	60	67	1.88146979e-003



Problema B-200  $\sigma_{max} = 7.73561529e-001$

Elemento	Nodo 1	Nodo 2	Volume
151	60	68	2.41352495e-002
152	61	69	5.04784427e-003
153	61	70	8.99669654e-003
154	62	63	1.55126869e-004
155	63	64	9.35866555e-005
156	64	65	9.35866555e-005
157	65	66	7.29658528e-005
158	66	67	7.29658528e-005
159	67	68	9.35866555e-005
160	68	69	9.35866555e-005
161	69	70	1.55126869e-004
162	62	71	1.08582162e-002
163	63	71	2.57810927e-004
164	63	72	5.30565501e-003
165	64	72	2.59967692e-002
166	65	72	8.75879559e-005
167	65	73	1.92936618e-003
168	66	73	1.77380861e-002
169	67	73	1.92936618e-003
170	67	74	8.75879559e-005
171	68	74	2.59967692e-002
172	69	74	5.30565501e-003
173	69	75	2.57810927e-004
174	70	75	1.08582162e-002
175	71	72	1.39645842e-002
176	72	73	1.05023932e-002
177	73	74	1.05023932e-002
178	74	75	1.39645842e-002
179	71	76	4.56890546e-002
180	72	76	7.60946615e-002
181	73	76	4.58930116e-002
182	73	77	4.58930116e-002
183	74	77	7.60946615e-002
184	75	77	4.56890546e-002
			9.99999983e-001

Problema B-200b  $\sigma_{max} = 7.03409527e-001$

Elemento	Nodo 1	Nodo 2	Volume
1	1	2	2.84329383e-003
2	2	3	2.84329383e-003
3	3	4	2.84329383e-003
4	4	5	2.84329383e-003
5	1	7	3.46881846e-003
6	2	8	2.04717163e-003
7	3	10	2.04717161e-003
8	4	12	2.04717163e-003
9	5	13	3.46881846e-003
10	6	15	2.04717168e-003
11	7	16	3.46881857e-003
12	8	16	4.09434322e-003
13	10	17	4.09434320e-003
14	12	18	4.09434322e-003
15	13	18	3.46881857e-003
16	14	19	2.04717168e-003
17	15	16	5.41579751e-003
18	16	17	2.57250364e-003
19	17	18	2.57250364e-003
20	18	19	5.41579751e-003
21	15	20	1.94968948e-004
22	15	21	6.60727307e-003
23	16	22	8.18868642e-003
24	17	24	6.14151477e-003
25	18	26	8.18868642e-003
26	19	27	6.60727307e-003
27	19	28	1.94968948e-004
28	20	29	2.24214054e-003
29	21	30	6.60727314e-003
30	22	30	1.02358580e-002
31	24	31	8.18868636e-003
32	26	32	1.02358580e-002
33	27	32	6.60727314e-003
34	28	33	2.24214054e-003
35	29	30	5.41579772e-003
36	32	33	5.41579772e-003
37	29	34	3.89937705e-004
38	29	35	6.60727329e-003
39	30	36	1.61824039e-002
40	31	38	1.02358579e-002
41	32	40	1.61824039e-002
42	33	41	6.60727329e-003
43	33	42	3.89937705e-004
44	34	43	2.43710929e-003
45	35	44	6.60727333e-003
46	36	44	1.82295755e-002
47	38	45	1.22830295e-002
48	40	46	1.82295755e-002
49	41	46	6.60727333e-003
50	42	47	2.43710929e-003

Problema B-200b  $\sigma_{max} = 7.03409527e-001$

Elemento	Nodo 1	Nodo 2	Volume
51	43	44	5.41579784e-003
52	46	47	5.41579784e-003
53	43	48	5.84906398e-004
54	43	49	6.60727341e-003
55	44	50	2.41761215e-002
56	45	52	1.43302011e-002
57	46	54	2.41761215e-002
58	47	55	6.60727341e-003
59	47	56	5.84906398e-004
60	48	57	2.63207799e-003
61	49	58	6.60727345e-003
62	50	58	2.62232931e-002
63	52	59	1.63773727e-002
64	54	60	2.62232931e-002
65	55	60	6.60727345e-003
66	56	61	2.63207799e-003
67	57	58	5.41579792e-003
68	60	61	5.41579792e-003
69	57	62	7.79875050e-004
70	57	63	6.60727349e-003
71	58	64	3.21698392e-002
72	59	66	1.84245442e-002
73	60	68	3.21698392e-002
74	61	69	6.60727349e-003
75	61	70	7.79875050e-004
76	62	71	2.82704664e-003
77	63	72	6.60727352e-003
78	64	72	3.42170108e-002
79	66	73	2.04717158e-002
80	68	74	3.42170108e-002
81	69	74	6.60727352e-003
82	70	75	2.82704664e-003
83	71	72	5.41579798e-003
84	74	75	5.41579798e-003
85	71	76	1.76013434e-002
86	72	76	1.00408892e-001
87	73	76	4.06591023e-002
88	73	77	4.06591023e-002
89	74	77	1.00408892e-001
90	75	77	1.76013434e-002
			9.99999996e-001

Problema B-9x9  $\sigma_{max} = 1.95555556e-003$

Elemento	Nodo 1	Nodo 2	Volume
1	1	11	8.60300412e-003
2	1	30	1.74223001e-002
3	1	48	2.05965908e-001
4	1	49	4.96601758e-002
5	9	17	8.60300493e-003
6	9	34	1.74223071e-002
7	9	51	4.96601620e-002
8	9	52	2.05965908e-001
9	11	21	8.60300409e-003
10	17	25	8.60300490e-003
11	21	31	8.60300407e-003
12	25	33	8.60300488e-003
13	30	59	1.74223001e-002
14	31	41	8.60300403e-003
15	33	41	8.60300484e-003
16	34	59	1.74223071e-002
17	41	48	7.10227255e-003
18	41	49	2.92118681e-003
19	41	51	2.92118600e-003
20	41	52	7.10227255e-003
21	48	67	8.52272724e-002
22	49	59	1.16847473e-002
23	51	59	1.16847441e-002
24	52	69	8.52272724e-002
25	59	67	1.70454545e-002
26	59	69	1.70454545e-002
27	67	77	5.11363635e-002
28	69	77	5.11363635e-002
			9.99999995e-001

Problema B-13x9  $\sigma_{max} = 3.63465616e-003$

Elemento	Nodo 1	Nodo 2	Volume
1	1	11	1.07524412e-003
2	1	20	5.09498701e-003
3	1	30	2.58052613e-003
4	1	38	2.88715722e-002
5	1	48	1.23128770e-002
6	1	49	8.38032144e-003
7	1	50	5.22235924e-002
8	1	65	1.52849544e-001
9	3	13	7.27854667e-004
10	7	15	7.27854667e-004
11	9	17	1.07524413e-003
12	9	26	5.09498701e-003
13	9	34	2.58052615e-003
14	9	44	2.88715722e-002
15	9	50	5.22235924e-002
16	9	51	8.38032141e-003
17	9	52	1.23128770e-002
18	9	71	1.52849544e-001
19	11	21	1.07524376e-003
20	13	23	7.27854597e-004
21	15	23	7.27854597e-004
22	17	25	1.07524376e-003
23	20	39	5.09498613e-003
24	21	31	1.07524356e-003
25	23	31	7.27854571e-004
26	23	33	7.27854571e-004
27	25	33	1.07524356e-003
28	26	43	5.09498613e-003
29	30	59	2.58052560e-003
30	31	38	1.21309099e-003
31	31	41	8.32625251e-004
32	33	41	8.32625252e-004
33	33	44	1.21309099e-003
34	34	59	2.58052561e-003
35	38	66	2.18356424e-002
36	39	58	5.09498569e-003
37	41	48	4.24581417e-004
38	41	49	4.92960124e-004
39	41	51	4.92960122e-004
40	41	52	4.24581417e-004

Problema B-13x9  $\sigma_{max} = 3.63465616e-003$

Elemento	Nodo 1	Nodo 2	Volume
41	43	60	5.09498569e-003
42	44	70	2.18356424e-002
43	48	67	5.09498357e-003
44	49	59	1.97184040e-003
45	50	58	1.01899697e-002
46	50	60	1.01899697e-002
47	51	59	1.97184039e-003
48	52	69	5.09498357e-003
49	58	65	1.52849543e-002
50	59	66	2.18356436e-003
51	59	67	1.01899683e-003
52	59	69	1.01899683e-003
53	59	70	2.18356436e-003
54	60	71	1.52849543e-002
55	65	84	4.58548632e-002
56	66	85	1.52849509e-002
57	67	77	3.05699021e-003
58	69	77	3.05699021e-003
59	70	87	1.52849509e-002
60	71	88	4.58548632e-002
61	77	85	3.05699017e-003
62	77	87	3.05699017e-003
63	84	103	4.58548549e-002
64	85	95	9.17097075e-003
65	87	95	9.17097075e-003
66	88	105	4.58548549e-002
67	95	103	9.17097089e-003
68	95	105	9.17097089e-003
69	103	113	2.75129140e-002
70	105	113	2.75129140e-002
			9.9999855e-001