

Universidade Federal de Santa Catarina  
Curso de Pós-Graduação em Matemática e  
Computação Científica

Análise e Testes Numéricos de um  
Algoritmo de Pontos Interiores  
para Programação Não Linear

Rafael Machado Casali

Orientador: Clóvis Caesar Gonzaga

Florianópolis

Fevereiro de 2002

**Universidade Federal de Santa Catarina**  
**Curso de Pós-Graduação em Matemática e**  
**Computação Científica**

**Análise e Testes Numéricos de um Algoritmo de**  
**Pontos Interiores para Programação Não Linear**

Dissertação submetida ao Curso de Pós-Graduação em Matemática e Computação Científica, do Centro de Ciências Físicas e Matemáticas da Universidade Federal de Santa Catarina, para a obtenção do grau de Mestre em Matemática, com Área de Concentração em Otimização

**Rafael Machado Casali**

**Florianópolis**

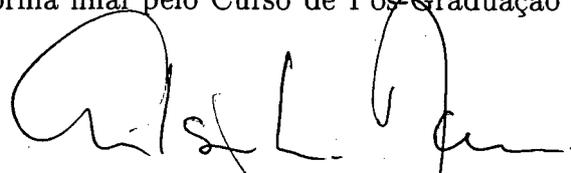
**Fevereiro de 2002**

# Análise e Testes Numéricos de um Algoritmo de Pontos Interiores para Programação Não Linear

por

Rafael Machado Casali

Esta Dissertação foi julgada para a obtenção do Título de "Mestre", Área de Concentração em Otimização, e aprovada em sua forma final pelo Curso de Pós-Graduação em Matemática e Computação Científica.



---

Celso Melchíades Dória  
Coordenador

Comissão Examinadora



---

Prof. Dr. Clóvis Caesar Gonzaga (UFSC - Orientador)



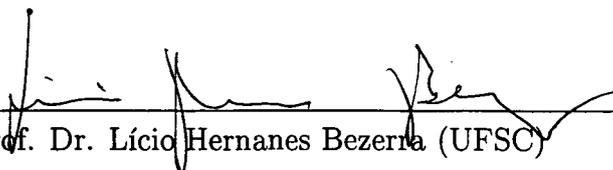
---

Profa. Dra. Sandra A. Santos (UNICAMP)



---

Prof. Dr. Mário César Zambaldi (UFSC)



---

Prof. Dr. Lício Hernanes Bezerra (UFSC)

Florianópolis, 4 de fevereiro de 2002

A Deus  
A meus pais, Adriano e Valda  
A minha noiva, Greice  
Aos meus irmãos, Adriana, Guilherme e Julinana

# Agradecimentos

Ao prof. Clovis C. Gonzaga, quem com paciência, sabedoria, competência e amizade soube orientar a realização deste trabalho.

Aos amigos Elizabeth, Marcia e Matioli pela troca de experiências, pelos seminários e apoio moral nos momentos críticos.

Aos meus colegas de Graduação e Pós-Graduação pela amizade e companhia, principalmente nas festas.

A minha noiva, Greice, e a minha família, pelo incentivo dados em todos os momentos.

À CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) pelo suporte financeiro.

# Sumário

Lista de Figuras	vii
Lista de Tabelas	viii
Lista de Algoritmos	ix
Resumo	x
Abstract	xi
Introdução	1
<b>1 Programação Quadrática Sequencial</b>	<b>3</b>
1.1 Programação Quadrática Sequencial . . . . .	4
1.2 Análise do Passo . . . . .	7
1.3 Função de Mérito . . . . .	8
1.4 Método PQS com Região de Confiança . . . . .	9
<b>2 Pontos Interiores para Programação Não Linear</b>	<b>15</b>
2.1 Método de Barreira . . . . .	15
2.2 Resolução do Problema Penalizado . . . . .	16
2.2.1 Critério de Parada . . . . .	20
2.2.2 Formulação Primal-Dual . . . . .	20
2.2.3 Atualização do Raio da Região de Confiança . . . . .	21
<b>3 Alterações para o Algoritmo de Pontos Interiores com Região de Confiança</b>	<b>22</b>
3.1 Exemplo de Marazzi e Nocedal . . . . .	22
3.2 Formas para a Região de Confiança . . . . .	23

3.3	Critério de Parada . . . . .	26
3.4	Atualização das Folgas . . . . .	27
<b>4</b>	<b>Implementação do Algoritmo de Pontos Interiores</b>	<b>29</b>
4.1	Resolução do Problema Normal . . . . .	29
4.2	Resolução do Problema Tangencial . . . . .	31
4.3	Testes Numéricos . . . . .	33
4.4	Análise dos Resultados Numéricos . . . . .	44
	<b>Referências Bibliográficas</b>	<b>46</b>

# Lista de Figuras

1.1	Restrições incompatíveis no modelo de região de confiança . . . . .	10
1.2	Restrições relaxadas tornam o subproblema compatível . . . . .	11
1.3	O passo $p^k = v^k + Z_k u^k$ do método de região de confiança . . . . .	12
3.1	Primeiras iterações do Algoritmo 4 para o problema (3.2) . . . . .	23
3.2	Região $\ (p_x, S^{-1}p_s)\  \leq \Delta$ , com $s = \frac{1}{2}$ e $\Delta = 0.8$ . . . . .	24
3.3	Região $\ (\frac{p_x}{\Delta}, S^{-1}p_s)\  \leq 1$ , com $s = \frac{1}{2}$ e $\Delta = 0.8$ . . . . .	25
3.4	Resolução do Exemplo 1 utilizando a região $\ (p_x, S^{-1}p_s)\  \leq \Delta$ . . . . .	25
3.5	Resolução do Exemplo 1 utilizando a região $\ (\frac{p_x}{\Delta}, S^{-1}p_s)\  \leq 1$ . . . . .	26
4.1	Trajetória Dogleg . . . . .	30

# Lista de Tabelas

4.1	Conjunto de problemas . . . . .	33
4.2	Resultados numéricos . . . . .	35
4.3	Comparação da atualização da folga . . . . .	44
4.4	Comparação entre as regiões . . . . .	45

# Lista de Algoritmos

1	Método PQS . . . . .	6
2	PQS com região de confiança . . . . .	14
3	Método de barreira . . . . .	16
4	PQS com região de confiança para o problema penalizado . . . . .	19
5	Dogleg . . . . .	31
6	GC-Steihaug . . . . .	32

## Resumo

Neste trabalho apresentamos alguns aspectos computacionais e testes numéricos de um algoritmo de pontos interiores para programação não linear. Este algoritmo transforma as restrições de desigualdades em igualdades incluindo variáveis de folga e aplicam uma barreira logarítmica sobre as folgas. Então minimizam esta função penalizada usando técnicas de programação quadrática sequencial com região de confiança, baseadas nas estratégias de Byrd e Omojokun.

## Abstract

We present some computational details and numerical tests of an interior point algorithm for nonlinear programming. This algorithm transforms the inequality constraints into equality constraints by introducing slack variables and applying a logarithmic barrier to the slack variables. Then they minimize this penalty function using sequential quadratic programming with trust regions, based on the strategy of Byrd and Omojokun.

# Introdução

O objetivo deste trabalho é mostrar e discutir aspectos dos algoritmos de pontos interiores para programação não linear propostos por Byrd, Gilbert e Nocedal [3] e por Byrd, Hribar e Nocedal [4]. Estes algoritmos, essencialmente, mesclam técnicas de pontos interiores (uma função barreira logarítmica) e de Programação Quadrática Sequencial (PQS) com região de confiança, para resolução de cada problema penalizado. Os métodos de PQS têm-se mostrado bastante robustos para problemas com funções não lineares, embora o custo de cada iteração seja potencialmente alto para problemas de grande porte. Por outro lado, os métodos de pontos interiores têm-se mostrado eficientes para problemas lineares de grande porte.

Os métodos de pontos interiores desenvolvidos a partir do método clássico de barreiras, descrito por Fiacco e McCormick [7], tiveram grande evolução após a publicação do algoritmo de Karmarkar [14]. Estes métodos procuram resolver uma seqüência de problemas penalizados, que dependem de um parâmetro de penalização  $\mu$ . Os otimizadores associados ao problema penalizado com cada valor de  $\mu$  definem uma trajetória central. Os métodos de pontos interiores seguem esta trajetória ou geram pontos em alguma vizinhança, possivelmente grande, dela. Estes métodos se destacam em programação linear, o que despertou o interesse em estendê-los para o caso não linear.

Nos métodos que estudaremos, as restrições de desigualdade são transformadas em igualdades pela inclusão de variáveis de folga. Os métodos de pontos interiores, em cada passo, devem resolver um problema de centralização, minimizando uma função penalizada, normalmente com barreira logarítmica sobre as folgas. Estudamos estratégias de globalização para a resolução desses problemas de centralização. As estratégias de globalização visam garantir convergência global do algoritmo, ou seja, convergência mesmo a partir de pontos iniciais distantes da solução ótima. Há basicamente duas formas de globalização: métodos de região de confiança e de busca linear.

Um método de região de confiança com derivadas segundas exatas é usado por

Yamashita, Yabe e Tanabe [28]. As restrições de igualdade são tratadas por uma penalidade  $\ell_1$ . Por outro lado, Forsgren e Gill [8] utilizam uma penalidade quadrática para as restrições de igualdade e uma barreira logarítmica para as desigualdades. A função assim penalizada é minimizada por um método baseado em busca linear. El-Bakry, Tapia, Tsuchiya e Zhang [6] usam uma busca linear e manipulam as desigualdades com variáveis de folga. Eles mostram resultados computacionais com uma função de mérito que é a norma  $\ell_2$  dos resíduos para as condições necessárias de primeira ordem. Com este procedimento, há uma tendência de convergência para pontos críticos que não são minimizadores. Por causa disto, Vanderbei e Shanno [26] preferem usar uma função de mérito que manipula as restrições de igualdade com penalidades quadráticas e as folgas com barreira logarítmica. Seu contexto também é de busca linear e as Hessianas indefinidas são modificadas através de perturbações da diagonal. A proposta de Gay, Overton e Wright em [10] encaixa neste mesmo contexto. A função de mérito é uma função barreira clássica com um Lagrangeano aumentado para manipular as restrições de igualdade. Finalmente, Byrd, Hribar e Nocedal [4] e Byrd, Gilbert e Nocedal [3] usam programação quadrática sequencial com região de confiança e uma função barreira. Essencialmente, as restrições de desigualdade são transformadas em igualdades que são manipuladas explicitamente e as variáveis de folga são incorporadas à função de mérito como termos da barreira logarítmica. Eles usam estratégias de Byrd [2] e Omojokun [18].

Neste trabalho, como dito anteriormente, estaremos interessados nos aspectos computacionais dos algoritmos propostos em [3, 4]. Como estes algoritmos baseiam-se em métodos de programação quadrática sequencial com região de confiança, iremos introduzir este método no Capítulo 1. Em seguida mostramos os algoritmos propostos em [3, 4] fazendo a adaptação do problema penalizado ao método de programação quadrática sequencial (Capítulo 2). No Capítulo 3 discutimos alguns aspectos dos algoritmos, procurando melhorar seus desempenhos computacionais. E, finalmente, no Capítulo 4 damos detalhes de nossa implementação e resultados de alguns testes numéricos.

Neste trabalho propomos duas modificações no método: na forma da região de confiança e na maneira de atualizar as variáveis de folga. Testamos estas duas modificações com a coleção CUTE [1] para identificar a influência no desempenho dos algoritmos. Nossos testes mostraram que estas modificações não apresentam grandes influências.

# Capítulo 1

## Programação Quadrática Sequencial

Uma das maneiras de resolver problemas “difíceis” é resolver uma sequência de problemas “fáceis”. Baseando-se neste fato, Programação Quadrática Sequencial (PQS) resolve problemas de otimização restrita não linear através de uma sequência de subproblemas quadráticos. Este método foi primeiramente proposto por Wilson [27] em 1963 e desenvolvido na década de 70 (Garcia-Palomares e Mangasarian [9], Han [12, 13], Powell [21, 22, 23]).

Num problema geral de otimização temos uma função objetivo e restrições geralmente não lineares. A estratégia básica de (PQS) consiste, em cada passo, fazer uma aproximação quadrática da função objetivo e uma aproximação linear das restrições. A princípio, resolver este problema quadrático é mais simples que resolver o problema original. Esta abordagem é usada juntamente com estratégias de busca linear e região de confiança [5].

Neste capítulo, estamos interessados em apenas mostrar uma visão geral do método de programação quadrática sequencial, mais especificamente, na estratégia com região de confiança. Inicialmente, apresentamos um algoritmo “local” que motiva a abordagem PQS e que nos permite introduzir uma análise do passo. Posteriormente, consideramos o método prático com região de confiança. Para tanto, baseamo-nos em [17].

## 1.1 Programação Quadrática Sequencial

Vamos considerar o problema

$$\begin{aligned} \min \quad & \varphi(x) \\ \text{s.a} \quad & c(x) = 0 \end{aligned} \tag{1.1}$$

onde  $\varphi : \mathbb{R}^n \rightarrow \mathbb{R}$  e  $c : \mathbb{R}^n \rightarrow \mathbb{R}^m$  são funções suaves.

A idéia da Programação Quadrática Sequencial (PQS) é para cada iterando  $x^k \in \mathbb{R}^n$  que não é solução de (1.1), formular e solucionar um subproblema quadrático e definir o minimizador deste subproblema como o iterando  $x^{k+1}$ .

A função Lagrangeano  $\mathcal{L} : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$  associada ao problema é

$$\mathcal{L}(x, \lambda) = \varphi(x) - \lambda^T c(x) \tag{1.2}$$

e vamos denotar  $A(x)$  a matriz Jacobiana das restrições, isto é,

$$A(x)^T = [ \nabla c_1(x) \quad \nabla c_2(x) \quad \dots \quad \nabla c_m(x) ]$$

onde  $c_i(x)$  é a  $i$ -ésima componente do vetor  $c(x)$ . As condições de otimalidade de primeira ordem (condições de Karush Kunh Tucker (KKT)) para o problema (1.1) são

$$F(x, \lambda) = \begin{bmatrix} \nabla \varphi(x) - A(x)^T \lambda \\ c(x) \end{bmatrix} = 0, \tag{1.3}$$

formando um sistema de  $n + m$  equações nas incógnitas  $x$  e  $\lambda$ .

Se  $x^*$  é solução ótima do problema (1.1) e  $A^* = A(x^*)$  tem posto completo, então existe  $\lambda^* \in \mathbb{R}^m$  tal que  $(x^*, \lambda^*)$  satisfaz (1.3). Com isto, podemos aplicar o Método de Newton para a equação não linear (1.3). O Jacobiano de (1.3) é dado por

$$\begin{bmatrix} W(x, \lambda) & -A(x)^T \\ A(x) & 0 \end{bmatrix},$$

em que  $W$  denota a Hessiana do Lagrangeano,

$$W(x, \lambda) = \nabla_{xx}^2 \mathcal{L}(x, \lambda).$$

Assim, o passo de Newton para a iteração  $(x^k, \lambda^k)$  é dado por

$$\begin{bmatrix} x^{k+1} \\ \lambda^{k+1} \end{bmatrix} = \begin{bmatrix} x^k \\ \lambda^k \end{bmatrix} + \begin{bmatrix} p^k \\ p^\lambda \end{bmatrix} \quad (1.4)$$

em que  $p^k$  e  $p^\lambda$  resolvem o sistema KKT

$$\begin{bmatrix} W_k & -A_k^T \\ A_k & 0 \end{bmatrix} \begin{bmatrix} p^k \\ p^\lambda \end{bmatrix} = - \begin{bmatrix} \nabla \varphi_k - A_k^T \lambda^k \\ c_k \end{bmatrix}. \quad (1.5)$$

Esta iteração é bem definida quando a matriz de KKT é não singular. Esta não singularidade é consequência das hipóteses de Newton, que são:

### Hipóteses 1.1

1. A matriz Jacobiana das restrições  $A_k$  tem posto completo;
2. A matriz  $W_k$  é definida positiva no espaço tangente das restrições, isto é,  $d^T W_k d > 0$  para todo  $d \neq 0$  tal que  $A_k d = 0$ .

A iteração de Newton (1.4) e (1.5) apresenta convergência quadrática sob as Hipóteses 1.1 e constitui um excelente algoritmo para resolver problemas com restrições de igualdade, desde que o ponto inicial esteja suficientemente próximo de  $x^*$ , uma solução do problema (1.1).

Agora vamos supor que a iteração  $(x^k, \lambda^k)$  não é solução do problema. Definimos inicialmente o problema quadrático

$$\begin{aligned} \min_p \quad & \frac{1}{2} p^T W_k p + \nabla \varphi_k^T p \\ \text{s.a.} \quad & A_k p + c_k = 0. \end{aligned} \quad (1.6)$$

Se as Hipóteses 1.1 são verificadas, então este problema tem solução única  $\bar{p}^k \in \mathbb{R}^n$  e pelas condições de KKT existe  $\bar{\lambda}^k > 0$  tal que

$$\begin{aligned} W_k \bar{p}^k + \nabla \varphi_k - A_k^T \bar{\lambda}^k &= 0 \\ A_k \bar{p}^k + c_k &= 0. \end{aligned}$$

O vetor  $(\bar{p}^k, \bar{\lambda}^k)$  pode ser identificado como uma solução das equações de Newton (1.5), pois se subtraímos  $A_k^T \bar{\lambda}^k$  dos dois lados da primeira equação em (1.5), obtemos

$$\begin{bmatrix} W_k & -A_k^T \\ A_k & 0 \end{bmatrix} \begin{bmatrix} \bar{p}^k \\ \bar{\lambda}^k \end{bmatrix} = - \begin{bmatrix} \nabla \varphi_k \\ c_k \end{bmatrix}. \quad (1.7)$$

Assim, pela não singularidade da matriz dos coeficientes, temos que  $\bar{p} = p^k$  e  $\bar{\lambda} = \lambda^{k+1}$ . Com isto, definimos uma *equivalência entre PQS e Método de Newton*: se as Hipóteses 1.1 são verificadas, então a nova iterada  $(x^{k+1}, \lambda^{k+1})$  pode ser definida a partir da solução do problema quadrático (1.6) ou a partir do método de Newton (1.4) e (1.5) aplicado às condições de otimalidade do problema.

Agora podemos esboçar um algoritmo para o Método de PQS:

---

**Algoritmo 1** Método PQS

---

Escolha  $(x^0, \lambda^0)$  inicial

Para  $k = 0, 1, 2, 3, \dots$  faça

    Calcule  $\varphi_k, \nabla\varphi_k, W_k, c_k$  e  $A_k$

    Resolva (1.6) para obter  $\bar{p}^k$  e o multiplicador  $\bar{\lambda}^k$

$x^{k+1} = x^k + \bar{p}^k$  e  $\lambda^{k+1} = \bar{\lambda}^k$

    Se convergência então

        PARE com solução aproximada  $(x^{k+1}, \lambda^{k+1})$

    fim

---

“Convergência” indica que o critério de parada, geralmente a norma das condições de primeira ordem de KKT, atingiu uma precisão dada.

A equivalência entre o método PQS e o Método de Newton para o sistema não linear  $F(x, \lambda) = 0$  garante a convergência local do algoritmo. Considere  $x^*$  solução de (1.1) com multiplicador  $\lambda^*$ . Se as Hipóteses 1.1 são asseguradas em  $(x^*, \lambda^*)$ , se  $f$  e  $c$  são duas vezes diferenciáveis com derivadas segundas Lipschitz contínuas, e se o ponto inicial  $(x^0, \lambda^0)$  é suficientemente próximo de  $(x^*, \lambda^*)$ , então os passos  $(x^k, \lambda^k)$  gerados pelo Algoritmo 1 convergem quadraticamente para  $(x^*, \lambda^*)$ .

Para tornar o método prático, devemos ser capazes de obter convergência global de um ponto inicial remoto, para problemas convexos ou não. Para isto temos duas estratégias, de busca linear e região de confiança.

Devemos observar que se  $W_k$  é definida positiva no espaço tangente às restrições, então o subproblema quadrático (1.6) tem solução única. Quando isto não ocorre, podemos usar métodos de busca linear, os quais substituem  $W_k$  por uma aproximação definida positiva  $B_k$  ou modificam  $W_k$  diretamente durante o processo de fatoração da matriz. Uma terceira possibilidade é definir  $W_k$  como a Hessiana da função Lagrangeano aumentado, tendo certas propriedades de convexidade. Em todos os casos, o subproblema (1.6) está bem definido.

Outra maneira seria adicionar uma restrição ao subproblema, limitando o passo a uma região em que o modelo (1.6) é considerado confiável. Esta estratégia de usar

métodos de PQS com região de confiança permite que a Hessiana  $W_k$  seja indefinida. Contudo, a inclusão desta restrição pode tornar o subproblema (1.6) incompatível, ou seja, as duas regiões viáveis podem ser disjuntas. Os procedimentos empregados para tornar o problema compatível podem aumentar o custo computacional.

Nenhuma destas estratégias, busca linear ou região de confiança, pode ser considerada claramente superior à outra. A maneira de usar estas estratégias tem um grande impacto na eficiência e robustez dos métodos de PQS, particularmente para problemas grandes. Neste trabalho estaremos interessados em estudar métodos PQS com região de confiança.

Outra questão importante no desenvolvimento do método PQS é a escolha da função de mérito que guia os algoritmos para a solução. Várias funções de mérito podem ser usadas. Os parâmetros destas funções de mérito necessitam ser ajustados em algumas iterações para assegurar que a direção obtida do subproblema seja de fato uma direção de descida com respeito a esta função. As regras para atualizar estes parâmetros requerem considerações cuidadosas, pois têm uma forte influência no comportamento prático dos métodos PQS.

## 1.2 Análise do Passo

Vamos considerar o problema quadrático (1.6) e com  $W_k = \nabla_{xx}^2 \mathcal{L}(x^k, \lambda^k)$ . Se as Hipóteses 1.1 são verificadas, a solução de (1.6) é dada pelo sistema KKT (1.5). Existem várias maneiras de se resolver este sistema KKT, ainda mais sabendo-se que o sistema é derivado de um subproblema de PQS.

Vamos supor que  $Y_k$  e  $Z_k$  geram o espaço imagem de  $A_k^T$  e o espaço nulo de  $A_k$ , respectivamente. Escrevendo

$$p^k = Y_k p_y + Z_k p_z,$$

podemos substituir em (1.5) para obter o seguinte sistema a ser resolvido em  $p_y$  e  $p_z$ :

$$\begin{aligned} (A_k Y_k) p_y &= -c_k \\ (Z_k^T W_k Z_k) p_z &= -Z_k^T W_k Y_k p_y - Z_k^T \nabla \varphi_k. \end{aligned} \quad (1.8)$$

Os multiplicadores de Lagrange  $\lambda^{k+1}$  podem ser obtidos resolvendo-se

$$(A_k Y_k)^T \lambda^{k+1} = Y_k^T (\nabla \varphi_k + W_k p^k) \quad (1.9)$$

com a hipótese de que  $W_k$  é definida positiva, a Hessiana reduzida  $Z_k^T W_k Z_k$  também o é.

Existem variantes para calcular uma aproximação do passo de KKT  $p^k$ . A mais usada é desacoplar os cálculos de  $p^k$  e  $\lambda^{k+1}$ , simplesmente “esquecendo” o termo envolvendo  $p^k$  do lado direito de (1.9)<sup>1</sup>. Os novos multiplicadores serão uma boa estimativa dos multiplicadores da programação quadrática próximo da solução. Se isto acontece, escolhamos  $Y_k = A_k^T$  (se  $A_k$  tem posto completo) e obtemos

$$\hat{\lambda}^{k+1} = (A_k A_k^T)^{-1} A_k \nabla \varphi_k. \quad (1.10)$$

Estes são chamados de multiplicadores de quadrados mínimos, pois podem ser derivados resolvendo-se o problema

$$\min_{\lambda} \|\nabla \varphi_k - A_k^T \lambda\|_2^2.$$

Os multiplicadores de quadrados mínimos são úteis mesmo quando a iteração corrente está longe da solução, pois procuram satisfazer as condições de otimalidade de primeira ordem em (1.5).

### 1.3 Função de Mérito

Para assegurar que o método PQS convirja a partir de um ponto inicial remoto é comum usar uma função de mérito  $\phi$  para controlar o tamanho do passo (em métodos com busca linear) ou determinar quando um passo é aceito e quando o raio da região de confiança precisa ser alterado (nos métodos com região de confiança). Esta função tem o papel da função objetivo na otimização irrestrita, devendo ser reduzida a cada passo.

Uma função de mérito possível é:

$$x \in \mathbb{R}^n, \nu \in \mathbb{R} \mapsto \phi_1(x, \nu) = \varphi(x) + \nu \|c(x)\|_1, \quad (1.11)$$

em que  $\nu$  é chamado de parâmetro de penalidade. Este parâmetro indica o quanto queremos dar importância ou não às restrições (neste caso). Esta função é não diferenciável, especificamente em pontos  $x$  onde uma ou mais componentes de  $c(x)$  são nulas.

---

<sup>1</sup>pois observa-se que  $p^k$  converge para zero quando estamos próximo da solução, enquanto  $\nabla \varphi_k$  normalmente não converge para zero.

Outra função de mérito bastante usada em PQS é a conhecida como função de mérito do Lagrangeano aumentado de Fletcher, definida por

$$x \in \mathbb{R}^n, \nu \in \mathbb{R} \mapsto \phi_F(x, \nu) = \varphi(x) - \lambda(x)^T c(x) + \nu \|c(x)\|^2, \quad (1.12)$$

onde  $\lambda(x)$  é definido como em (1.10) e  $\|\cdot\|$  é a norma  $\ell_2$ .

Em ambos os casos precisamos controlar o parâmetro de penalidade para que realmente exista uma redução possível na função de mérito. As funções de mérito são um ponto crítico dos algoritmos para otimização restrita, pois é difícil medir o quanto e quando devemos penalizar mais ou menos as restrições, ou seja, quando devemos olhar mais para a otimalidade ou mais para a viabilidade. Em geral, o parâmetro  $\nu$  cresce ao longo das iterações. Em [11] os autores trabalham com uma sequência  $\{\nu\}$  não monótono.

## 1.4 Método PQS com Região de Confiança

Esta estratégia tem muitas propriedades atraentes, como permitir um tratamento para o caso em que os gradientes das restrições ativas são linearmente dependentes, e de usar diretamente as informações da derivada segunda.

Vamos considerar o problema (1.1), para o qual o subproblema quadrático é (1.6). Adicionando a restrição de região de confiança, obtemos um novo subproblema

$$\begin{aligned} \min_p \quad & \frac{1}{2} p^T W_k p + \nabla \varphi_k^T p \\ \text{s.a.} \quad & A_k p + c_k = 0 \\ & \|p\| \leq \Delta_k. \end{aligned} \quad (1.13)$$

Vamos supor que  $\|\cdot\|$  denota a norma  $\ell_2$ , embora na prática seja comum trabalhar com a região de confiança escalonada na forma  $\|Dp\| \leq \Delta_k$  (onde  $D$  é uma matriz diagonal). O raio da região de confiança  $\Delta_k$  será atualizado dependendo da relação entre a redução predita na função de mérito comparada à redução “real”. Se existe uma boa relação, o raio da região de confiança é inalterado ou incrementado, caso contrário o raio é diminuído.

A inclusão da restrição de região de confiança torna o subproblema (1.13) mais difícil de ser resolvido do que o subproblema original (1.6). Ainda mais, o novo subproblema pode não ter solução, pois as restrições podem ser incompatíveis, ou seja, o subproblema pode ser inviável, como mostra a Figura 1.1.

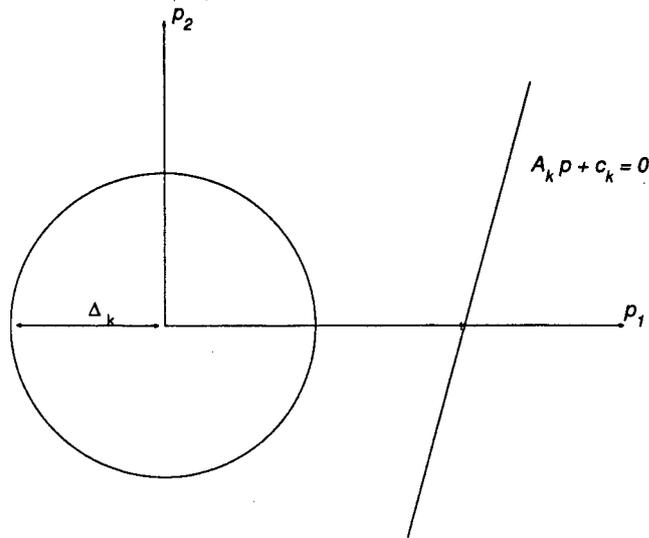


Figura 1.1: Restrições incompatíveis no modelo de região de confiança

Para resolver este conflito não podemos aumentar o raio da região de confiança  $\Delta_k$  até que ambas restrições sejam satisfeitas, pois, em primeiro lugar, isto vai contra a proposta de usar região de confiança como uma maneira de definir a região na qual confiamos no modelo. Outro ponto de vista é questionar o porquê de satisfazer exatamente as restrições de igualdade em cada passo, ou seja, podemos relaxar as restrições de igualdade em cada passo, para satisfazê-las exatamente apenas no limite. Nestes casos, temos várias abordagens: deslocamento das restrições e duas regiões elípticas [17]. Aqui estaremos interessados em apenas um caso no qual deslocamos as restrições, exposto a seguir.

A idéia é substituir a restrição linear de (1.13) pela restrição deslocada

$$A_k p + \theta c_k = 0 \quad (1.14)$$

em que  $\theta \in (0, 1]$  é escolhido suficientemente pequeno tal que o conjunto de restrições de (1.13) seja viável. É fácil encontrar um valor para  $\theta$ , pois  $\theta$  tem o efeito de deslocar a variedade linear correspondente às restrições linearizadas para uma variedade paralela que intercepta a região de confiança, veja a Figura 1.2.

Contudo, não há uma maneira simples de fazer a escolha de  $\theta$  que assegure um bom desempenho do algoritmo na prática, pois  $\theta$  deve controlar simultaneamente o quanto o passo computado  $p$  tende a satisfazer a viabilidade das restrições ou a minimizar a função objetivo. Por exemplo, se escolhermos um valor muito pequeno para

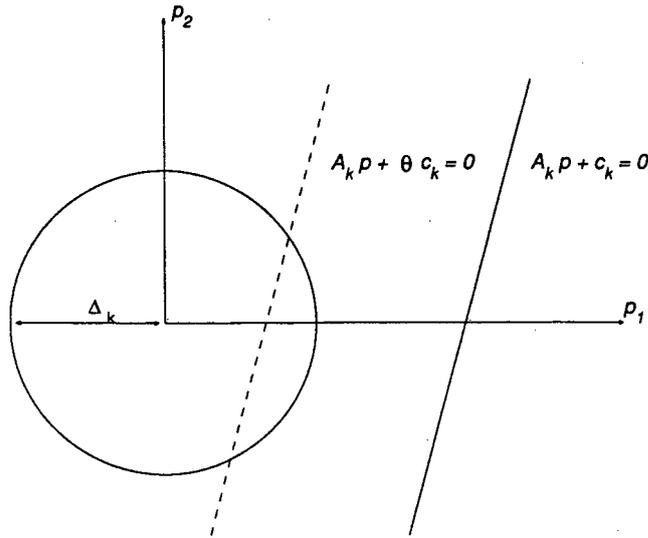


Figura 1.2: Restrições relaxadas tornam o subproblema compatível

$\theta$ , ou seja, se deslocamos as restrições até próximo da iterada corrente  $x^k$ , então o passo computado no subproblema (1.13) tenderá a fazer pequenas melhoras em satisfazer  $c(x) = 0$ , focando na redução da função objetivo  $\varphi$ .

Para driblar esta dificuldade, calculamos o passo em dois estágios, seguindo a estratégia proposta por Byrd [2] e Omojokun [18]. No primeiro estágio, ignoramos a função objetivo de modo geral, e analisamos o quanto podemos melhorar as restrições linearizadas de (1.13) enquanto ficamos dentro da região de confiança. Isto é, precisamos resolver o seguinte “subproblema normal”

$$\begin{aligned} \min_v \quad & \|A_k v + c_k\|_2 \\ \text{s.a} \quad & \|v\|_2 \leq \zeta \Delta_k \end{aligned} \quad (1.15)$$

onde  $\zeta \in (0, 1)$  (usualmente  $\zeta = 0,8$ ). Chamamos a solução  $v^k$  deste subproblema de *passo normal*. Como soluções exatas não são necessárias, este subproblema pode ser resolvido aproximadamente pelo Método de Dogleg, com algumas modificações, pois o passo de Newton não é unicamente definido. Para mais detalhes, ver [17].

Agora “forçamos” o passo total  $p^k$  do método de região de confiança a satisfazer as restrições linearizadas como no passo normal. Ou seja, substituímos  $\theta c_k$  por  $-A_k v_k$  em (1.14), obtendo um novo subproblema

$$\begin{aligned}
\min_p \quad & \frac{1}{2} p^T W_k p + \nabla \varphi_k^T p \\
\text{s.a} \quad & A_k p = A_k v^k \\
& \|p\| \leq \Delta_k.
\end{aligned} \tag{1.16}$$

É claro que as restrições de (1.16) são consistentes, pois a escolha  $p = v^k$  satisfaz ambas as restrições. Para resolver isto, usaremos o fato de que o passo normal  $v^k$  pertence ao espaço imagem de  $A_k^T$ , assim expressamos o passo total  $p^k$

$$p^k = v^k + Z_k u^k \tag{1.17}$$

para algum  $u^k \in \mathbb{R}^{n-m}$  e  $Z_k$  uma matriz cujas colunas formam uma base para o espaço nulo de  $A_k$ . Isto é, fixamos a componente de  $p^k$  no espaço imagem de  $A_k^T$  para ser exatamente  $v^k$  e não permitimos outros movimentos nesta direção, como mostra a Figura 1.3.

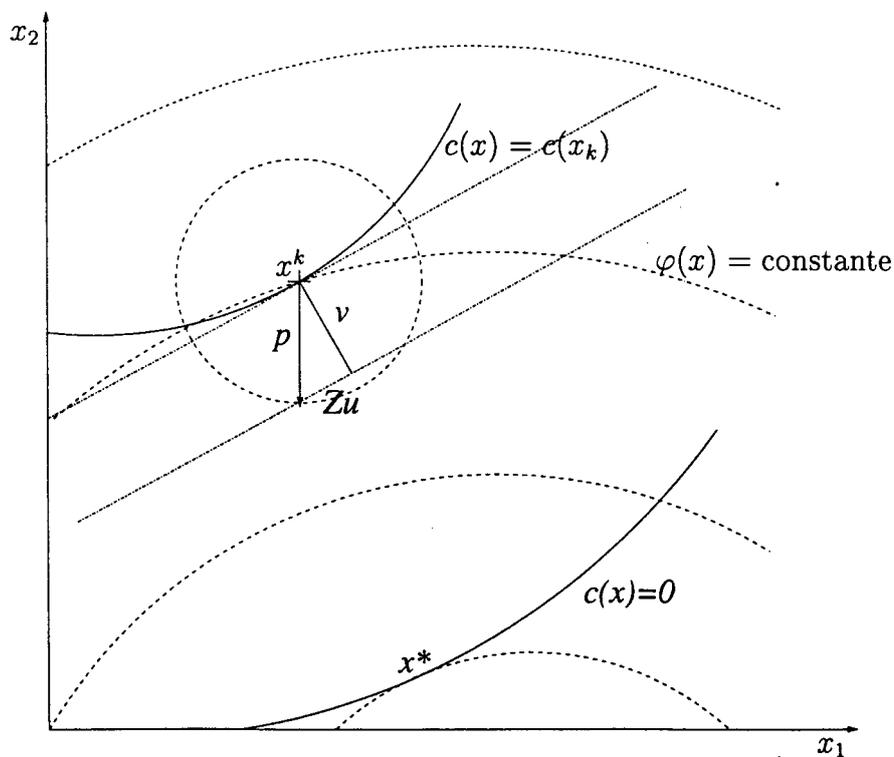


Figura 1.3: O passo  $p^k = v^k + Z_k u^k$  do método de região de confiança

Substituindo (1.17) em (1.16), obtemos o seguinte *subproblema tangencial* na variável  $u$

$$\begin{aligned} \min_u \quad m_k(u) &= (\nabla\varphi_k + W_k v^k)^T Z_k u + \frac{1}{2} u^T Z_k^T W_k Z_k u \\ \text{s.a} \quad \|Z_k u\|_2 &\leq \sqrt{\Delta_k^2 - \|v^k\|_2^2}. \end{aligned} \quad (1.18)$$

A forma simples da restrição da região de confiança é uma consequência do fato de que as componentes  $v^k$  e  $Z_k u^k$  em (1.17) são ortogonais. Porém, não podemos tratar a restrição de (1.18) como uma restrição de região de confiança escalonada, pois a matriz  $Z_k$  nem sempre é quadrada. No entanto, podemos escrever esta restrição como

$$u^T D_k^2 u \leq \Delta_k,$$

com  $D_k^2 = Z_k^T Z_k$ , em que  $D_k$  é definida positiva, pois  $Z_k$  tem colunas linearmente independentes.

Chamamos a solução de (1.18) de passo tangencial. Este problema não pode ser computado pelo Método de Dogleg, a menos que tenhamos a Hessiana reduzida  $Z_k^T W_k Z_k$  definida positiva. Mas o Método de Gradiente Conjugado (Steihaug [25]), pode sempre ser aplicado a este subproblema.

Uma componente importante nos Métodos PQS é a escolha da função de mérito, pois esta indicará quando um passo é aceito. Uma função de mérito possível é a função  $\ell_2$  não diferenciável

$$\phi(x, \nu) = \varphi(x) + \nu \|c(x)\|_2. \quad (1.19)$$

O uso da norma  $\ell_2$  é consistente no subproblema normal, mas outras funções também podem ser usadas. A “redução real” nesta função gerada pelo passo candidato  $p^k = v^k + Z_k u^k$  é definida como

$$\mathbf{ared} = \phi(x^k, \nu^k) - \phi(x^k + p^k, \nu^k),$$

enquanto a “redução predita” é

$$\mathbf{pred} = m_k(0) - m_k(u) + \nu^k \mathbf{vpred},$$

em que  $m_k$  é definido em (1.18) e  $\mathbf{vpred}$  é a redução no modelo (1.15) obtido pelo passo normal  $v^k$ , dado por

$$\mathbf{vpred} = \|c(x)\| - \|c(x) + A_k v^k\|.$$

Exigimos que  $\nu^k$  seja pequeno suficiente de modo a garantir que **pred** seja positivo e grande em relação a **vpred**, isto é,

$$\mathbf{pred} \geq \rho \nu^k \mathbf{vpred}, \quad (1.20)$$

em que  $0 < \rho < 1$  (por exemplo  $\rho = 0,3$ ). Podemos forçar esta desigualdade escolhendo o parâmetro de penalidade  $\nu^k$  tal que

$$\nu^k \geq \frac{m_k(0) - m_k(u)}{(1 - \rho)\mathbf{vpred}}. \quad (1.21)$$

A seguir descrevemos mais precisamente o algoritmo PQS com região de confiança para problemas com restrições de igualdade.

---

**Algoritmo 2** PQS com região de confiança

---

Dados  $\varepsilon > 0, \eta, \zeta, \gamma \in (0, 1), x^0 \in \mathbb{R}^n, k = 0$  e  $\Delta_0 > 0$

**Repita**

Calcule  $\varphi_k, c_k, \nabla\varphi_k, A_k$

Calcule as estimativas dos multiplicadores  $\hat{\lambda}_k$  por (1.10)

Se  $\|\nabla\varphi_k - A_k^T \hat{\lambda}_k\|_\infty \leq \varepsilon$  e  $\|c_k\| < \varepsilon$  **então**

PARE com a solução aproximada  $x^k$

Resolva o subproblema normal (1.15) para  $v^k$

Calcule  $Z_k$  com as colunas formando base para o espaço nulo de  $A_k$

Calcule ou atualize  $W_k$

Resolva o subproblema tangencial (1.18) para  $u^k$

$p^k = v^k + Z_k u^k$

Calcule  $\rho_k = \frac{\mathbf{ared}}{\mathbf{pred}}$

Se  $\rho_k > \eta$  **então**

$x^{k+1} = x^k + p^k$

Escolha  $\Delta_{k+1}$  satisfazendo  $\Delta_{k+1} \geq \Delta_k$

**senão**

$x^{k+1} = x^k$

Escolha  $\Delta_{k+1}$  satisfazendo  $\Delta_{k+1} \leq \gamma \|p^k\|$

$k = k + 1$

**Até convergência**

---

No próximo capítulo iremos mostrar como utilizar estes conceitos junto com métodos de pontos interiores.

## Capítulo 2

# Pontos Interiores para Programação Não Linear

Neste capítulo apresentaremos o método de pontos interiores proposto por Byrd, Gilbert e Nocedal [3], e Byrd, Hribar e Nocedal [4]. O método combina as técnicas de pontos interiores [15], programação quadrática sequencial e região de confiança. Esta combinação se deve ao fato que métodos de pontos interiores conseguem tratar restrições de desigualdades, e programação quadrática sequencial explora eficientemente a não linearidade das restrições. A região de confiança permite trabalhar com problemas convexos ou não, e fazer o uso da segunda derivada:

### 2.1 Método de Barreira

Vamos considerar o problema

$$\begin{aligned} \min_x & f_0(x) \\ \text{s.a.} & f_i(x) \leq 0, \quad i = 1, 2, \dots, m \end{aligned} \tag{2.1}$$

onde  $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$  para  $i = 0, 1, \dots, m$  são funções com derivadas segundas contínuas. Supomos que existe um ponto interior, isto é, um ponto  $x$  tal que  $f_i(x) < 0$ .

Seguindo as estratégias de pontos interiores, acrescentamos variáveis de folga e uma penalidade logarítmica, obtendo o seguinte problema de barreira associado ao problema (2.1)

$$\begin{aligned}
& \min_{x,s} f_0(x) - \mu \sum_{i=1}^m \log s_i \\
& \text{s.a. } f(x) + s = 0 \\
& \quad s > 0,
\end{aligned} \tag{2.2}$$

em que  $\mu > 0$  é o parâmetro de barreira e os vetores das variáveis de folga  $s = (s_1, s_2, \dots, s_m)^T$  são positivos.

O método de barreira para resolver o problema de desigualdade (2.1) é estabelecido abaixo.

---

**Algoritmo 3** Método de barreira

---

Dado  $\mu > 0$

**Repita**

Resolva “aproximadamente” o problema (2.2).

Reduza  $\mu$ .

**Até** convergência

---

Para resolver aproximadamente o problema penalizado (2.2), as maneiras mais usadas são a aplicação do método de Newton às condições de KKT do problema (2.2) (veja [24, 26]), e aplicação de técnicas de programação quadrática sequencial com região de confiança [3, 4], o que apresentaremos a seguir.

## 2.2 Resolução do Problema Penalizado

Para resolver o problema penalizado (2.2) usaremos as estratégias expostas no Capítulo 1. Para simplificar a notação, denotamos

$$z = (x, s), \quad \hat{\varphi}(z) = f_0(x) - \mu \sum_{i=1}^m \log s_i, \quad \hat{c}(z) = f(x) + s$$

reescrevendo o problema (2.2) como:

$$\begin{aligned}
& \min_z \hat{\varphi}(z) \\
& \text{s.a. } \hat{c}(z) = 0.
\end{aligned} \tag{2.3}$$

Temos uma única e importante diferença entre os problemas (1.1) e (2.3): é que neste último temos que garantir a positividade das variáveis de folga.

Seguindo a metodologia de PQS, definimos o Lagrangeano associado ao problema (2.2):

$$\mathcal{L}(x, s, \lambda) = f_0(x) - \mu \sum_{i=1}^m \log s_i + \lambda^T (f(x) + s), \quad (2.4)$$

e minimizamos o modelo quadrático do Lagrangeano sujeito às restrições linearizadas e a uma região de confiança para garantir convergência a partir de um ponto remoto. Com isto, ficamos com o problema da forma

$$\begin{aligned} \min_p \quad & \nabla \hat{\varphi}(z)^T p + \frac{1}{2} p^T H p \\ \text{s.a.} \quad & \hat{A}(z)p + \hat{c}(z) = 0 \\ & p \in \Omega, \end{aligned} \quad (2.5)$$

em que  $p = \begin{pmatrix} p_x \\ p_s \end{pmatrix}$ ,  $H$  é a Hessiana do Lagrangeano (2.4) que é definida por blocos

$$H = \nabla_{zz}^2 \mathcal{L}(x, s, \lambda) = \begin{bmatrix} \nabla_{xx}^2 \mathcal{L}(x, s, \lambda) & 0 \\ 0 & \mu S^{-2} \end{bmatrix},$$

$S$  é a matriz diagonal da folgas,  $\hat{A}(z)$  é a Jacobiana de  $\hat{c}$  dada por

$$\hat{A}(z) = \begin{bmatrix} A(x) & I \end{bmatrix},$$

e  $\Omega$  é a região de confiança.

Para as variáveis  $x$ , a região de confiança  $\|p_x\| \leq \Delta_x$  relaciona-se à precisão do modelo quadrático do Lagrangeano. Com relação às variáveis  $s$ , a restrição  $s > 0$  faz com que o formato mais conveniente para a região de confiança seja o dos elipsóides de Dikin, isto é,  $\|S^{-1} p_s\| \leq \Delta_s$ . Byrd, Gilbert e Nocedal [3] e Byrd, Hribar e Nocedal [4] definem a região de confiança em  $z = (x, s)$  fazendo

$$\|(p_x, S^{-1} p_s)\| \leq \Delta \quad (2.6)$$

em (2.5). Discutiremos outras formas para  $\Omega$  adiante.

Geralmente, a região definida em (2.6) não assegura que as novas variáveis de folga  $s + p_s$  sejam positivas. Para isto, introduzimos a relação

$$s + p_s \geq (1 - \tau)s,$$

com  $\tau \in (0, 1)$  fixo (geralmente próximo de 1), para limitar o passo em  $s$ . Assim, o problema quadrático (2.5) é reformulado como

$$\begin{aligned} \min_p \quad & \nabla \hat{\varphi}(z)^T p + \frac{1}{2} p^T H p \\ \text{s.a.} \quad & \hat{A}(z)p + \hat{c}(z) = 0 \\ & \|(p_x, S^{-1} p_s)\| \leq \Delta \\ & p_s \geq -\tau s. \end{aligned} \tag{2.7}$$

Agora PQS, para facilitar a resolução deste problema, reescreve  $p$  como a soma de um vetor pertencente ao espaço imagem de  $\hat{A}^T(z)$  (passo normal) e outro pertencente ao espaço nulo de  $\hat{A}(z)$  (passo tangencial). Ou seja, fazemos  $p = v + w$  e calculamos:

**Passo normal:**  $v$  é a solução do problema normal

$$\begin{aligned} \min_v \quad & \|\hat{A}(z)v + \hat{c}(z)\| \\ \text{s.a.} \quad & \|(v_x, S^{-1} v_s)\| \leq \zeta \Delta \\ & v_s \geq -\zeta \tau s, \end{aligned} \tag{2.8}$$

sendo  $\zeta \in (0, 1)$  o parâmetro de contração da região (geralmente  $\zeta = 0,8$ ).

**Passo tangencial:** fazemos  $w = Zu$ , em que  $Z = \begin{pmatrix} Z_x^T & Z_s^T \end{pmatrix}^T$  é a matriz cujas colunas geram o espaço nulo de  $\hat{A}$  e  $u$  é a solução do problema tangencial, pertencente ao espaço nulo de  $\hat{A}$ :

$$\begin{aligned} \min_u \quad & (\nabla f_o^k + W v_x)^T Z_x u - \mu (S_k^{-1} e - S_k^{-2} v_s)^T Z_s u \\ & + \frac{1}{2} u^T (Z_x^T W Z_x + \mu Z_s^T S_k^{-2} Z_s) u \\ \text{s.a.} \quad & \|(Z_x u, S^{-1} Z_s u)\| \leq (\Delta^2 - \|(v_x, S^{-1} v_s)\|^2)^{\frac{1}{2}} \\ & S^{-1} (v_s + Z_s u) \geq -\tau, \end{aligned} \tag{2.9}$$

em que  $W = \nabla_{xx}^2 \mathcal{L}(x, s, \lambda)$ .

Note que, em ambos os problemas, estamos limitando o passo em  $s$ , sendo esta a principal diferença em relação aos problemas normal e tangencial do Capítulo 1, o que dificulta a sua resolução. Byrd, Hribar e Nocedal [4] sugerem pequenas modificações nos métodos de Dogleg e Steihaug para solucionar estes problemas, como veremos adiante.

Para analisar se o passo é aceito ou não, verificamos a redução na função de

mérito definida por

$$\phi(x, s, \nu) = f_0(x) - \mu \sum_{i=1}^m \log s_i + \nu \|f(x) + s\| \quad (2.10)$$

onde

$$\begin{aligned} \text{ared} &= \phi(x, s, \nu) - \phi(x + d_x, s + d_s, \nu) \\ \text{vpred} &= \|f(x) + s\| - \|f(x) + s + \hat{A}v\| \\ \text{pred} &= -\nabla \hat{\phi}(z)^T p - \frac{1}{2} p^T H p + \nu \text{vpred} \end{aligned}$$

e exigimos que  $\nu$  satisfaça a relação (1.20),

$$\text{pred} \geq \rho \nu \text{vpred}.$$

Com isto, podemos reescrever o Algoritmo 2 para resolver o problema penalizado (2.2), para um dado  $\mu > 0$ .

---

**Algoritmo 4** PQS com região de confiança para o problema penalizado

---

Dados  $\varepsilon_\mu > 0, \eta, \zeta, \tau, \gamma \in (0, 1), x^0 \in \mathbb{R}^n, s^0 \in \mathbb{R}^m, \nu > 0, k = 0$  e  $\Delta_0 > 0$ .

**Repita**

Calcule as estimativas dos multiplicadores  $\hat{\lambda}_k$  por (1.10).

Resolva o subproblema normal (2.8) para  $v^k$ .

Calcule  $Z_k$  tal que suas colunas formem base para o espaço nulo de  $\hat{A}_k$ .

Calcule ou atualize  $H$ .

Resolva (2.9) para  $u^k$ .

$$p^k = v^k + Z_k u^k.$$

Calcule  $\rho_k = \frac{\text{ared}}{\text{pred}}$ .

Se  $\rho_k > \eta$  então

$$z^{k+1} = z^k + p^k.$$

Escolha  $\Delta_{k+1}$  satisfazendo  $\Delta_{k+1} \geq \Delta_k$ .

senão

$$z^{k+1} = z^k.$$

Escolha  $\Delta_{k+1}$  satisfazendo  $\Delta_{k+1} \leq \gamma \|p^k\|$ .

$$k = k + 1.$$

**Até** que algum critério de parada seja satisfeito.

---

Temos que chamar atenção para alguns detalhes: necessitamos fazer com que o algoritmo tenha características primais-duais, descrever a atualização do raio da região de confiança e escolher um critério de parada.

### 2.2.1 Critério de Parada

Um dos critérios mais usados parte das condições de KKT do problema (2.2). Em [4] define-se a função  $E(\cdot)$  por

$$E(x, s, \mu) = \max \{ \|\nabla f_0(x) + A\lambda\|_\infty, \|s\lambda - \mu e\|_\infty, \|f(x) + s\|_\infty \}, \quad (2.11)$$

onde  $\lambda$  é calculado através de mínimos quadrados (1.10), e em [3]

$$E(x, s, \mu) = \max \{ \|\nabla f_0(x) + A\lambda\|, \|f(x) + s\| \}, \quad (2.12)$$

com  $\lambda = \mu s^{-1}$ ,  $\|\cdot\|$  é a norma euclidiana.

Ambos os critérios de parada têm um ponto negativo: são fortemente dependentes de escala.

### 2.2.2 Formulação Primal-Dual

O Algoritmo 4 como proposto tem uma formulação conhecida como primal, que fica estabelecida pelo segundo bloco  $\Sigma$  da Hessiana do Lagrangeano

$$\nabla_{zz}^2 \mathcal{L}(x, s, \lambda) = \begin{bmatrix} \nabla_{xx}^2 \mathcal{L}(x, s, \lambda) & 0 \\ 0 & \Sigma \end{bmatrix}$$

em relação à variável  $s$ , neste caso,

$$\Sigma = \mu S^{-2}.$$

Em [4], os autores sugerem uma formulação primal-dual usando uma certa aproximação desta Hessiana

$$\Sigma = S^{-1}\Lambda,$$

em que  $\Lambda$  é a matriz diagonal de ordem  $m$  cujos elementos diagonais são os multiplicadores de Lagrange  $\lambda_i$ . Esta escolha é feita com base na eficiência dos métodos de pontos interiores para programação linear considerando a condição de KKT perturbada para o problema de barreira (2.2).

Em [4] os autores fazem uma comparação numérica entre as duas formulações e concluem que a formulação primal-dual é claramente superior a formulação primal.

### 2.2.3 Atualização do Raio da Região de Confiança

A relação entre a redução predita e a redução verdadeira

$$\rho = \frac{\text{ared}}{\text{pred}}$$

fornece um critério para aceitação do passo e também para atualização da região de confiança. De modo geral, quando esta relação está próxima de 1, existe uma boa concordância entre o modelo e a função em torno do ponto corrente, de modo que parece seguro aumentar a região na iteração seguinte. Se a relação não está próxima de 1, não se altera o tamanho da região. Mas se o passo foi recusado e a relação está próxima de 0, então é necessário reduzir a região e confiar menos no modelo.

Em [4], a seguinte regra é adotada para atualização do raio  $\Delta$

$$\Delta^+ = \begin{cases} \max\{7\|p\|, \Delta\} & \text{se } \rho \geq 0.9 \\ \max\{2\|p\|, \Delta\} & \text{se } 0.3 \leq \rho < 0.9 \\ \Delta & \text{se } \eta \leq \rho < 0.3 \end{cases}$$

em que  $\Delta^+$  indica o raio atualizado,  $p$  o passo calculado a partir do ponto corrente  $z$  e  $\eta > 0$  geralmente  $10^{-8}$ . Quando o passo é rejeitado, o raio é reduzido de tal forma que o novo raio seja no máximo  $\frac{\Delta}{2}$ , mas não inferior a um décimo do comprimento do passo  $p$ . Para determinar o fator de contração de  $\Delta$ , usa-se interpolação linear ou quadrática, cujos detalhes são apresentados em [19]. Ajusta-se também o raio  $\Delta$  quando o parâmetro de penalidade  $\mu$  é reduzido, usando a regra de atualização  $\Delta^+ = \max\{5\Delta, 1\}$ .

## Capítulo 3

# Alterações para o Algoritmo de Pontos Interiores com Região de Confiança

Neste capítulo mostraremos um problema no qual o método de pontos interiores para programação não linear descrito no capítulo anterior tem um desempenho insatisfatório. Este fato nos levou a analisar alguns detalhes sobre o método. Primeiramente, questionamos a região de confiança usada. Em seguida, sugerimos um novo critério de parada para o método. Finalmente, fazemos uma breve análise sobre a atualização das variáveis de folga.

### 3.1 Exemplo de Marazzi e Nocedal

O método PQS com região de confiança [3, 4] possui boas propriedades de convergência. No entanto, é possível encontrar exemplos para os quais o método tem um desempenho insatisfatório. Marazzi e Nocedal [16] exibem um tal exemplo:

**Exemplo 1** *Considere o problema linear na reta:*

$$\begin{aligned} \min \quad & x \\ \text{s.a} \quad & x \geq b \\ & x \in \mathbb{R}, \end{aligned} \tag{3.1}$$

em que  $b$  é uma constante, digamos  $b = 500$ . Dado  $\mu > 0$ , o problema de barreira é dado por

$$\begin{aligned} \min \quad & x - \mu \log(s) \\ \text{s.a.} \quad & x - b - s = 0 \\ & s > 0 \end{aligned} \tag{3.2}$$

cuja solução ótima é  $(x, s) = (b + \mu, \mu)$ .

Ilustramos na Figura (3.1) as primeiras iterações geradas pelo Algoritmo 4 aplicado ao problema acima com  $\mu = 0.1$ , e ponto inicial  $x^0 = 1$  e  $s^0 = \mu$  e considerando a região de confiança inativa nos problemas normal e tangencial.

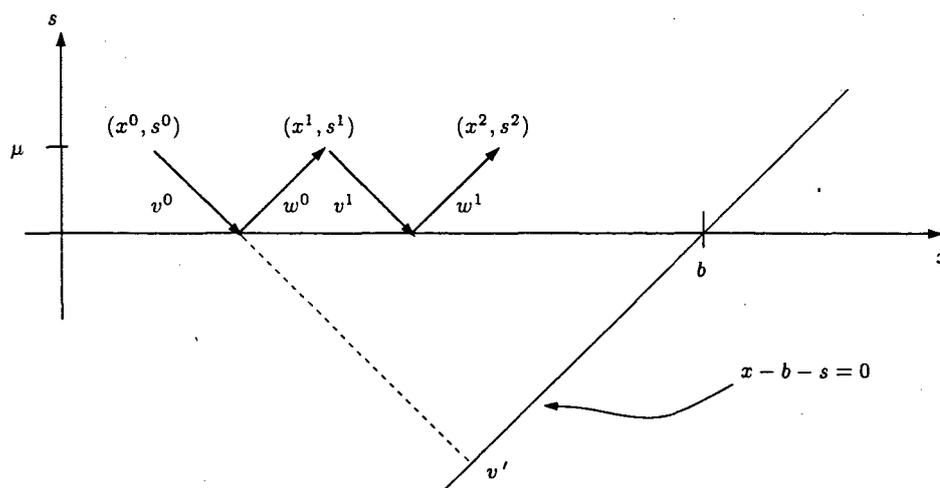


Figura 3.1: Primeiras iterações do Algoritmo 4 para o problema (3.2)

Na resolução exata do passo normal (2.8), com região de confiança inativa, teríamos como solução  $v'$ . Entretanto, com a positividade da folga, obtemos  $v^0$  sem uma redução significativa da inviabilidade. Como no cálculo do passo tangencial a inviabilidade é fixada, obtemos um passo total  $v^0 + w^0$  curto. O mesmo ocorre nas demais iterações, o que torna necessário um grande número de iterações para atingir viabilidade.

Este exemplo nos levou ao estudo de novas formas de região de confiança. Voltaremos a este exemplo adiante.

## 3.2 Formas para a Região de Confiança

Como a região de confiança restringe o tamanho do passo. A sua forma influencia o desempenho do método de pontos interiores para programação não linear

(apresentado no capítulo anterior). A região apresentada a seguir (adotada por [3, 4]) vincula o tamanho do passo nas variáveis  $x$  às variáveis de folga  $s$ , e é altamente dependente de escala nas variáveis  $x$ , pois usa o elipsóide de Dikin apenas nas variáveis de folga. A Figura 3.2 exemplifica esta região para um problema em  $\mathbb{R}^2$  com uma restrição.

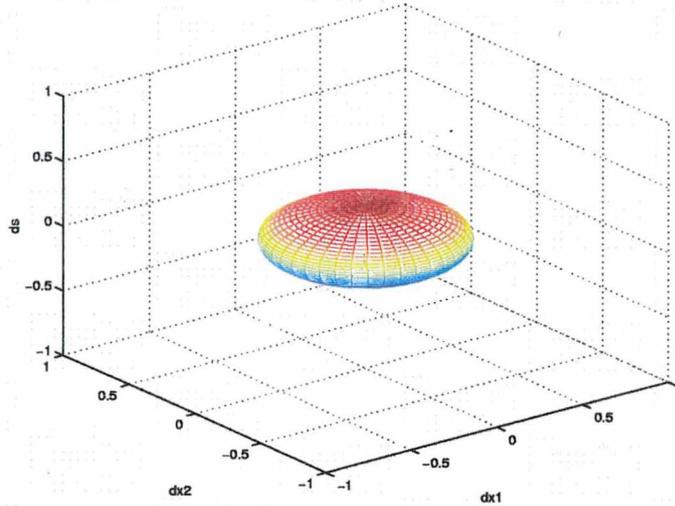


Figura 3.2: Região  $\|(p_x, S^{-1}p_s)\| \leq \Delta$ , com  $s = \frac{1}{2}$  e  $\Delta = 0.8$ .

Como já mencionamos no capítulo anterior, gostaríamos de que a região de confiança para as variáveis  $x$  fosse da forma  $\|p_x\| \leq \Delta_x$  e para as variáveis  $s$ , da forma  $\|S^{-1}p_s\| \leq \Delta_s$ . Como as restrições são lineares em relação a  $s$ , a região de confiança para essas variáveis pode ser grande. Com  $\Delta_s = 1$  obtemos os elipsóides de Dikin totalmente contidos no primeiro octante. Em  $\mathbb{R}^{n+m}$ , podemos fazer uma composição dessas regiões com

$$\left\| \left( \frac{p_x}{\Delta}, S^{-1}p_s \right) \right\| \leq 1. \quad (3.3)$$

Em ambos os casos precisamos assegurar que existirá o passo tangencial, ou seja, precisamos incluir as restrições de  $p_s$  para que este não chegue muito próximo da fronteira, sem deixar espaço para o próximo passo (normal ou tangencial).

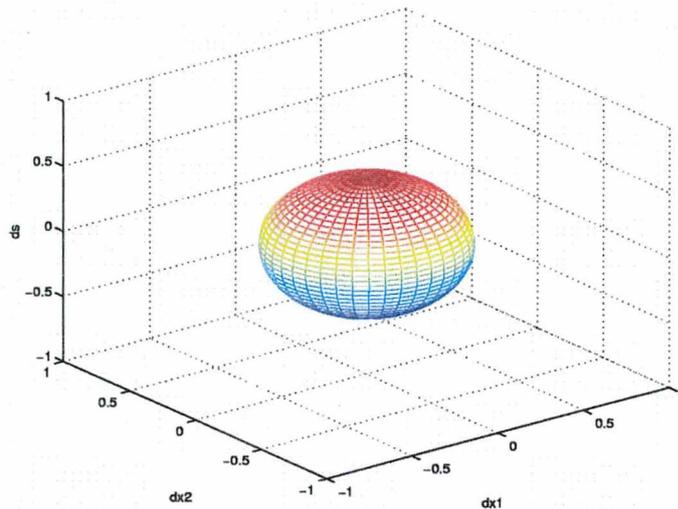


Figura 3.3: Região  $\|(\frac{p_x}{\Delta}, S^{-1}p_s)\| \leq 1$ , com  $s = \frac{1}{2}$  e  $\Delta = 0.8$ .

A Figura 3.3 representa um problema em  $\mathbb{R}^2$  com uma restrição. Note que a região de [3, 4] está contida na região definida por (3.3). Adiante apresentaremos o desempenho destas regiões com alguns testes numéricos. Um grande motivo que nos levou a testar esta forma de região foi a resolução do exemplo de Marazzi e Nocedal [16], como mostram as Figuras 3.4 e 3.5.

As Figuras 3.4 e 3.5 mostram os passos normais e tangenciais (representados pela linha na figura) e os pontos (representados por um \* nas figuras) gerados pelo Algoritmo 4 aplicado ao problema (3.1).

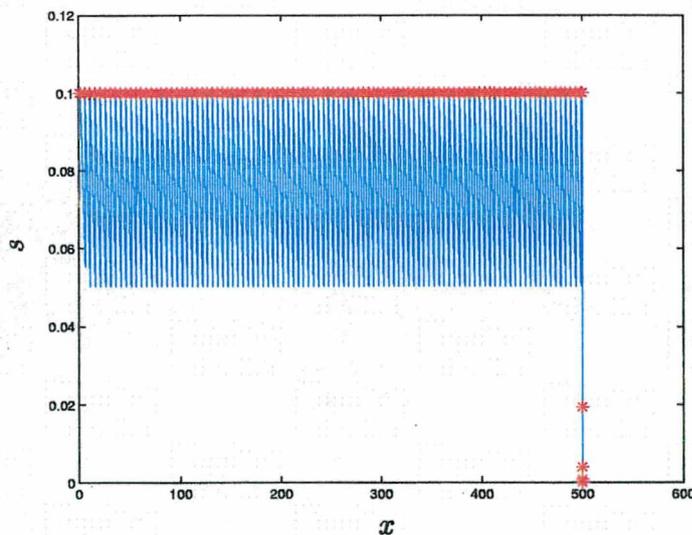


Figura 3.4: Resolução do Exemplo 1 utilizando a região  $\|(p_x, S^{-1}p_s)\| \leq \Delta$

A Figura 3.4 mostra os passos gerados usando a região de confiança definida por (2.6). Neste caso o método levou 107 iterações internas para chegar na solução.

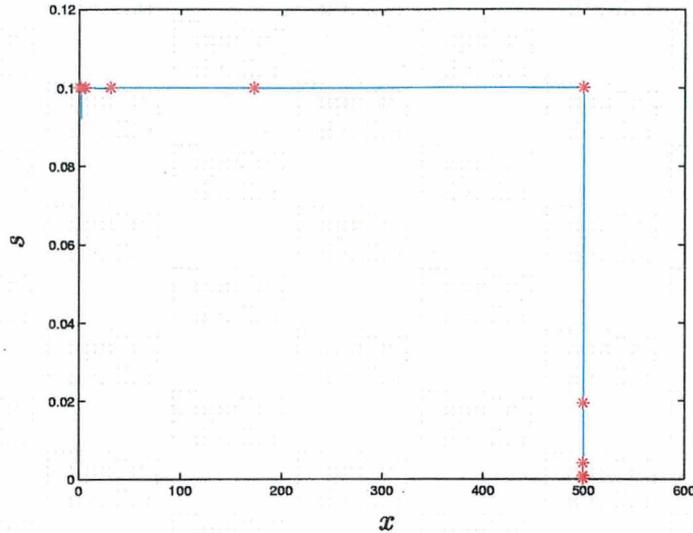


Figura 3.5: Resolução do Exemplo 1 utilizando a região  $\|(\frac{p_x}{\Delta}, S^{-1}p_s)\| \leq 1$

Na Figura 3.5 temos os passos gerados usando a região de confiança definida por (3.3). Com esta região, o número de iterações internas é 11, o que leva a acreditar, numa primeira impressão, que a região (3.3) tenha um melhor desempenho.

Outras regiões possíveis seriam

$$\|p_x\| \leq \Delta \text{ e } \|S^{-1}p_s\| \leq 1,$$

ou seja, bolas em  $x$  e elipsóides de Dikin em  $s$ , assim desvinculando as variáveis  $x$  e  $s$ . Outra alternativa é usar bola em  $x$  e caixa em  $s$ , fazendo

$$\|p_x\| \leq \Delta \text{ e } p_s \geq -\tau s,$$

dando total liberdade a  $s$ . E ainda transformar tudo em caixa, ou seja, o uso da norma infinito. Os testes não foram realizados para estas últimas propostas, devido às dificuldades de implementação.

### 3.3 Critério de Parada

Quando falamos em critério de parada, geralmente a primeira coisa que vem à mente é que os erros devem ser pequenos. Mas quão pequenos? Teoricamente

pensamos em erros de ordem  $10^{-5}$ ,  $10^{-7}$ , entretanto na prática isto não é verdade. Imagine um problema com grandeza em bilhões, um erro aceitável seria na ordem de milhares ( $10^3$ ).

O critério de parada adotado por [4] para cada problema penalizado e para o problema geral é baseado na função  $E(x, s, \mu)$  definida em (2.11), a qual é dependente de escala. O critério para resolver o problema geral exige que as normas das condições de KKT fiquem menores que um  $\varepsilon_{tol}$  (nos teste  $\varepsilon_{tol} = 10^{-5}$ ), ou seja, exige que a otimalidade, complementaridade e a viabilidade sejam reduzidas até  $\varepsilon_{tol}$ .

A nossa proposta é, para cada problema penalizado, usar um critério de parada proporcional aos valores iniciais. Para isto definimos uma nova função erro

$$E_p(x, s, \mu) = \max \left\{ \mu_0 \|\nabla f_0(x) + A^T \lambda\|_\infty, 2\|S\lambda - \mu e\|_\infty, \frac{5\mu_0 \|f(x) + s\|_\infty}{(\|f(x_0) + s_0\|_\infty + 1)} \right\} \quad (3.4)$$

na qual, para cada problema penalizado, gostaríamos que  $E_p \leq \mu$ . E para o problema geral paramos o algoritmo quando  $\mu$  for pequeno (nos testes paramos quando  $\mu < 10^{-5}$ ). Com isto estamos querendo que as normas das condições de KKT tenham uma redução da ordem de  $10^{-5}$ , exemplo usado nos testes numéricos.

Para analisar estes dois critérios imagine que começamos com um erro na viabilidade da ordem de  $10^4$ . Com o critério de parada usado por [4] este erro teria que cair até  $10^{-5}$  (exemplo usado nos testes), ou seja, uma redução da ordem de  $10^{-9}$ . Com a nossa proposta, este erro teria que ter uma redução de  $10^{-5}$ , ou seja, ele deveria ficar na ordem de  $10^{-1}$ .

### 3.4 Atualização das Folgas

Em [4], as folgas são atualizadas por

$$s^{k+1} = s^k + p_s. \quad (3.5)$$

Por outro lado, os autores de [3] propõem uma maneira diferente de atualizar as folgas. Quando alguma restrição  $f_i$  é viável em  $x^{k+1}$ , ou seja,  $f_i(x^{k+1}) \leq 0$ , temos mais liberdade na escolha da variável de folga correspondente  $s_i^{k+1}$  e a forma de atualizá-la é

$$s_i^{k+1} = \max\{s_i^k + p_{s_i}, -f_i(x^{k+1})\} \quad (3.6)$$

o que garante que a folga  $s_i^{k+1}$  não seja desnecessariamente pequena.

Este procedimento é seguro pois este novo valor de  $s$  produzirá uma redução na função de mérito, o que garante que este ponto também seria aceito, ou seja

$$\phi(x^{k+1}, s^{k+1}, \nu) \leq \phi(x^{k+1}, s^k + p_s, \nu).$$

Vamos verificar este fato usando a função de mérito definida em (2.10). Como  $s_i^{k+1} \geq s_i^k + p_{s_i}$ , temos que  $\log((s_i^k + p_{s_i})/s_i^{k+1}) \leq 0$  e  $\|f^{k+1} + s^{k+1}\| \leq \|f^{k+1} + s^k + p_s\|$ , pois as componentes de  $f^{k+1} + s^{k+1}$  ou são iguais a zero ou são iguais às componentes de  $f^{k+1} + s^k + p_s$ .

Logo,

$$\begin{aligned} \phi(x^{k+1}, s^{k+1}, \nu) - \phi(x^{k+1}, s^k + p_s, \nu) &= \mu \sum_{i=1}^m \log \frac{s_i^k + p_{s_i}}{s_i^{k+1}} + \\ &\quad \nu (\|f^{k+1} + s^{k+1}\| - \|f^{k+1} + s^k + p_s\|) \\ &\leq 0. \end{aligned}$$

Nos testes numéricos comparamos as duas formas de atualização das variáveis de folga: (3.5) e (3.6).

# Capítulo 4

## Implementação do Algoritmo de Pontos Interiores

Para implementação do algoritmo de pontos interiores para programação não linear, utilizamos o Matlab v6.0 R12. Implementamos o algoritmo com a característica primal-dual. Implementamos também duas rotinas importantes na resolução do problema, uma para resolução do problema normal (1.15) e outra para resolução do problema tangencial (1.18).

### 4.1 Resolução do Problema Normal

O problema normal (1.15) e (2.8) pode ser visto como um problema da forma: dado  $\Delta > 0$

$$\begin{aligned} \min_d \quad & Q_D(d) = b^T d + \frac{1}{2} d^T H d \\ \text{s.a} \quad & \|d\| \leq \zeta \Delta. \end{aligned} \tag{4.1}$$

Um método computacionalmente barato e eficiente para calcular a solução deste tipo de problema é o método dogleg [20], o qual apresentaremos a seguir.

Vamos analisar o comportamento da solução em função do raio da região de confiança  $\Delta$ . Denotamos a solução do problema (4.1) por  $d^*(\Delta)$  para enfatizar a sua dependência de  $\Delta$ . Supomos que  $H$  é definida positiva.

Se considerarmos o problema de minimizar uma quadrática sem restrições, a solução seria o passo completo  $d^H = -H^{-1}b$ , pois  $H$  é definida positiva. Assim, para  $\Delta \geq \|d^H\|$ , a solução do problema (4.1) será  $d^H$ . Com isto temos:

$$d^*(\Delta) = d^H, \text{ quando } \Delta \geq \|d^H\|.$$

Quando  $\Delta$  é pequeno, a restrição  $\|d\| \leq \Delta$  faz com que o termo quadrático do problema (4.1) tenha pouca influência na solução, isto é, a solução estará próxima da direção de máximo declive. Assim temos

$$d^*(\Delta) \simeq -\Delta \frac{b}{\|b\|}, \text{ quando } \Delta \text{ é pequeno.}$$

Ao variarmos  $\Delta$  teremos um trajetória de soluções ótimas como na Figura 4.1. O método dogleg faz uma aproximação desta trajetória por dois segmentos de reta. O primeiro liga a origem da região de confiança ao minimizador irrestrito ao longo da direção de maior declive (ponto de Cauchy) definido por

$$d^u = -\frac{b^T b}{b^T H b} b. \quad (4.2)$$

O segundo segmento vai de  $d^u$  ao ponto do passo completo  $d^H$ . Este caminho é chamado de *trajetória dogleg* (veja Figura 4.1). O método escolhe  $d$  ao longo desta trajetória sujeito à região de confiança para ser o minimizador aproximado do problema (4.1). Esta escolha é o ponto de intersecção entre a trajetória dogleg e a fronteira da região de confiança quando  $\|d^H\| > \Delta$ . No caso contrário será o passo completo  $d^H$ .

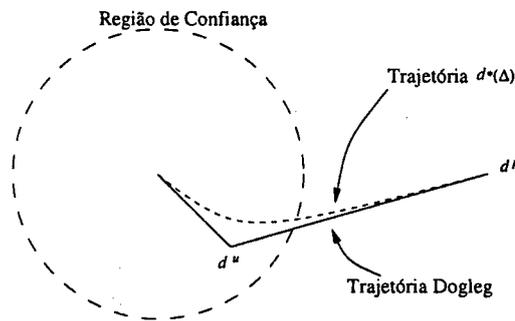


Figura 4.1: Trajetória Dogleg

Para achar o ponto de intersecção da trajetória dogleg com a fronteira da região de confiança, basta calcular um valor apropriado de  $\omega$  para o qual  $\|\tilde{d}(\omega)\| = \Delta$ , onde

$$\tilde{d}(\omega) = \begin{cases} \omega d^u, & 0 \leq \omega \leq 1 \\ d^u + (\omega - 1)(d^H - d^u), & 1 \leq \omega \leq 2 \end{cases}$$

Assim temos o seguinte algoritmo:

---

**Algoritmo 5 Dogleg**

---

Calcule o passo completo  $d^H = -H^{-1}b$

Se  $d^H$  é viável então

$$d = d^H$$

senão

Calcule  $d^u$  por (4.2)

Se  $d^u$  não é viável então

Ache  $\omega \in [0, 1]$  tal que  $\omega d^u$  é viável

$$d = \omega d^u$$

senão

Ache  $\omega \in [1, 2]$  tal que  $d^u + (\omega - 1)(d^H - d^u)$  é viável

$$d = d^u + (\omega - 1)(d^H - d^u)$$

---

O método dogleg pode ser adaptado para lidar com  $H$  indefinida, mas não há razão para fazer isto porque o passo completo  $d^H$  não é o minimizador irrestrito da quadrática neste caso.

Alguns autores ainda testam se este ponto da interseção é melhor do que o ponto da interseção da região com a direção de Newton, ficando com o melhor dos dois. Ou seja, calculam alguns pontos e fazem uma busca para achar o ponto de interseção com a região.

No algoritmo de pontos interiores para programação não linear, apresentado no Capítulo 2, o problema normal (2.8) apresenta uma restrição para que a folga não se torne negativa. Isto afeta o dogleg apenas na busca da interseção com a região. Ou seja, devemos fazer a busca nos preocupando com a região e a positividade das folgas.

## 4.2 Resolução do Problema Tangencial

Para o problema tangencial (1.18) usamos o método de gradientes conjugados de Steihaug [25] que, como o método de dogleg, minimiza uma quadrática numa região de confiança.

O método de Steihaug é baseado no algoritmo de gradientes conjugados, um algoritmo para resolver sistemas lineares com matriz de coeficientes simétrica definida positiva. A proposta de Steihaug [25] é de incluir três diferentes critérios de parada no método de gradientes conjugados (GC) padrão. Primeiramente, terminar quando se tem uma aproximação suficientemente boa do passo de Newton. O segundo critério consiste em terminar quando a norma da aproximação é muito grande. Neste caso

tomamos uma combinação linear da iterada anterior e da corrente. O terceiro consiste em terminar ao encontrar uma direção de curvatura negativa quando nos movemos para a fronteira.

Considerando o problema (4.1), podemos escrever o algoritmo seguinte:

---

**Algoritmo 6 GC-Steihaug**

---

Dados:  $\epsilon > 0$

$$d_0 = 0, r_0 = b, q_0 = -r_0$$

Se  $\|r_0\| < \epsilon$  então

Retorne com  $d = d_0$

Para  $i = 0, 1, 2, \dots$  faça

Se  $q_i^T H q_i \leq 0$  então

Ache  $\tau$  tal que  $d = d_i + \tau q_i$  seja solução de (4.1) e satisfaça  $\|d\| = \Delta$

$d = d_i + \tau q_i$ , fim.

$$\alpha_i = \frac{r_i^T r_i}{q_i^T H q_i}$$

$$d_{i+1} = d_i + \alpha_i q_i$$

Se  $\|d_{i+1}\| \geq \Delta$  então

Ache  $\tau \geq 0$  tal que  $d = d_i + \tau q_i$  satisfaz  $\|d\| = \Delta$

$d = d_i + \tau q_i$ , fim.

$$r_{i+1} = r_i + \alpha_i H q_i$$

Se  $\|r_{i+1}\| < \epsilon \|r_0\|$  então

$d = d_{i+1}$ , fim.

$$\beta_{i+1} = \frac{r_{i+1}^T r_{i+1}}{r_i^T r_i}$$

$$q_{i+1} = r_{i+1} + \beta_{i+1} q_i$$

fim

---

A inicialização de  $d_0$  em zero é importante para o algoritmo, pois após a primeira iteração (assumindo  $\|r_0\|_2 \geq \epsilon$ ), temos

$$d_1 = \alpha_0 q_0 = \frac{r_0^T r_0}{q_0^T H q_0} q_0 = -\frac{b^T b}{b^T H b} b,$$

o que é exatamente o ponto de Cauchy, que fornece uma condição necessária para convergência global.

Outra propriedade do método é que, em cada iteração, o passo  $d_i$  é maior, em norma, que seu predecessor, conseqüência da inicialização  $d_0 = 0$ . Além disso, o algoritmo produz pontos  $d_i$  em algum caminho interpolado de  $d_1$  a  $d$ , um caminho que a cada passo aumenta a distância total ao ponto de partida. Quando  $H$  é definida positiva, este caminho pode ser comparado com o caminho do método dogleg, porque ambos os métodos movem-se do passo de Cauchy ao passo completo  $d^H$ , até a fronteira

da região de confiança intervir.

Como no problema normal, o problema tangencial para a programação não linear também inclui uma restrição para que a folga seja estritamente positiva. Em [4] eles resolvem este problema calculando o gradiente conjugado normalmente e verificam se o ponto encontrado satisfaz a positividade das folgas. Caso isso não ocorra, restauram o último ponto viável (em relação às folgas) e a direção calculada a partir deste ponto. Então procuram o maior passo possível nesta direção tal que continue viável.

### 4.3 Testes Numéricos

O objetivo dos testes numéricos é analisar a influência, no desempenho do algoritmo, da atualização das folgas (3.6) e da forma da região de confiança. A análise será feita considerando os dois critérios de paradas discutidos na Seção 3.3. Para isto rodamos 8 programas para cada problema. Ou seja, para cada critério temos duas regiões e para cada região podemos ou não, atualizar a folga por (3.6).

Para os testes numéricos usamos um conjunto de 35 problemas da coleção CUTE [1], todos com apenas restrições de desigualdades, cujas características são descritas na Tabela 4.1. Inicialmente tínhamos selecionado 55 problemas, porém estamos listando apenas aqueles problemas que nunca atingiram o número máximo de iterações (100 nos testes) na resolução do problema penalizado (2.2) e com número de restrições menor que 50.

Tabela 4.1: Conjunto de problemas

Problema	# variáveis	# restrições	tipo das restrições	objetivo
CB2	3	3	3 não lineares	linear
CB3	3	3	3 não lineares	linear
CHACONN1	3	3	3 não lineares	linear
CHACONN2	3	3	3 não lineares	linear
COSHFUN	10	3	3 não lineares	linear
DEMYMALO	3	3	2 lineares 1 não linear	linear
EXPFITA	5	22	22 lineares	não linear
GIGOMEZ1	3	3	2 lineares 1 não linear	linear
GOFFIN	51	50	50 lineares	linear
HS10	2	1	1 não linear	linear
HS12	2	1	1 não linear	não linear
HS22	2	2	1 não linear 1 linear	não linear

continua na próxima página

Tabela 4.1: Conjunto de problemas (continuação)

Problema	# variáveis	# restrições	tipo das restrições	objetivo
HS29	3	1	1 não linear	não linear
HS43	4	3	3 não linear	não linear
HS268	5	5	5 lineares	não linear
KIWCRESC	3	2	2 não lineares	linear
MADSEN	3	6	6 1 não lineares	linear
MIFFLIN1	3	2	1 linear 1 não linear	linear
MIFFLIN2	3	2	2 não lineares	linear
MINMAXRB	3	4	2 não lineares 2 lineares	linear
OET1	3	6	6 lineares	não linear
OET2	3	6	6 não lineares	linear
OET3	4	6	6 lineares	linear
OET4	4	6	6 não lineares	linear
OET5	5	6	6 não lineares	linear
POLAK1	3	2	2 não lineares	linear
PT	2	3	3 lineares	linear
ROSENMMX	5	4	4 não lineares	linear
SIPOW1	2	20	20 lineares	linear
SIPOW1M	2	20	20 lineares	linear
SIPOW2M	2	20	20 lineares	linear
TIF1	3	11	11 não lineares	não linear
TIF2	3	11	11 lineares	linear
TIF3	3	11	11 lineares	não linear
WOMFLET	3	3	3 não lineares	linear

O CUTE já vem com uma interface para o Matlab, entretanto foi necessária uma pequena modificação nesta. Em sua versão original, a interface não disponibilizava a ferramenta CIDH, a qual permitia o cálculo da Hessiana de apenas uma função, seja objetivo ou de uma restrição.

Um problema que encontramos em nossas implementações foi o fato de que em problemas com muitas restrições o algoritmo ficava bastante lento. Isto ocorre pelo fato de usarmos funções do Matlab para resolução de sistemas lineares e para encontrar matrizes de base do espaço nulo, sem explorar a esparsidade das matrizes.

Resolvemos estes problemas com os seguintes valores iniciais:  $\mu = 0, 1$ ,  $\Delta_0 = 1$ ,  $\varepsilon_{tol} = 10^{-5}$ ,  $s_0 = [1, 1, \dots, 1]^T \in \mathbb{R}^m$ . A seguir temos a Tabela 4.2 de resultados numéricos cujas colunas contêm:

1. o critério de parada usado: BHN para o critério (2.11) proposto em [4] e “Sugestão”, o critério (3.4) proposto neste trabalho,

2. a região de confiança,
3. se atualizamos a folga por (3.5) ou (3.6),
4. o valor atingido pela função objetivo,
5. o número total de iterações internas,
6. o número total de iterações externas (número de vezes que reduzimos o  $\mu$ ),
7. o número de cálculos de funções (contando objetivo e cada uma das restrições),
8. o número de cálculos de gradientes,
9. o número de cálculos de Hessianas.

Tabela 4.2: Resultados numéricos

CB2

Critério	Região	atual. S	obj.	#int	#ext	#funções	# $\nabla$	# $\nabla^2$
BHN	$\ (p_x, S^{-1} p_s)\  \leq \Delta$	(3.5)	1.952	16	8	100	100	64
		(3.6)	1.952	16	8	100	100	64
	$\ (\frac{p_x}{\Delta}, S^{-1} p_s)\  \leq 1$	(3.5)	1.952	16	8	100	100	64
		(3.6)	1.952	16	8	100	100	64
Sugestão	$\ (p_x, S^{-1} p_s)\  \leq \Delta$	(3.5)	1.952	14	6	84	84	56
		(3.6)	1.952	14	6	84	84	56
	$\ (\frac{p_x}{\Delta}, S^{-1} p_s)\  \leq 1$	(3.5)	1.952	14	6	84	84	56
		(3.6)	1.952	14	6	84	84	56

CB3

Critério	Região	atual. S	obj.	#int	#ext	#funções	# $\nabla$	# $\nabla^2$
BHN	$\ (p_x, S^{-1} p_s)\  \leq \Delta$	(3.5)	2	16	7	96	96	64
		(3.6)	2	16	7	96	96	64
	$\ (\frac{p_x}{\Delta}, S^{-1} p_s)\  \leq 1$	(3.5)	2	15	7	92	92	60
		(3.6)	2	15	7	92	92	60
Sugestão	$\ (p_x, S^{-1} p_s)\  \leq \Delta$	(3.5)	2	13	6	80	80	52
		(3.6)	2	13	6	80	80	52
	$\ (\frac{p_x}{\Delta}, S^{-1} p_s)\  \leq 1$	(3.5)	2	13	6	80	80	52
		(3.6)	2	13	6	80	80	52

CHACONN1

Critério	Região	atual. S	obj.	#int	#ext	#funções	# $\nabla$	# $\nabla^2$
BHN	$\ (p_x, S^{-1} p_s)\  \leq \Delta$	(3.5)	1.952	14	7	88	88	56
		(3.6)	1.952	14	7	88	88	56
	$\ (\frac{p_x}{\Delta}, S^{-1} p_s)\  \leq 1$	(3.5)	1.952	16	7	96	96	64
		(3.6)	1.952	16	7	96	96	64
Sugestão	$\ (p_x, S^{-1} p_s)\  \leq \Delta$	(3.5)	1.952	13	6	80	80	52
		(3.6)	1.952	13	6	80	80	52
	$\ (\frac{p_x}{\Delta}, S^{-1} p_s)\  \leq 1$	(3.5)	1.952	13	6	80	80	52
		(3.6)	1.952	13	6	80	80	52

continua...

Tabela 4.2: Resultados numéricos (continuação)

CHACONN2

Critério	Região	atual. S	obj.	#int	#ext	#funções	# $\nabla$	# $\nabla^2$
BHN	$\ (p_x, S^{-1} p_s)\  \leq \Delta$	(3.5)	2	15	7	92	92	60
		(3.6)	2	15	7	92	92	60
	$\ (\frac{p_x}{\Delta}, S^{-1} p_s)\  \leq 1$	(3.5)	2	15	7	92	92	60
		(3.6)	2	15	7	92	92	60
Sugestão	$\ (p_x, S^{-1} p_s)\  \leq \Delta$	(3.5)	2	14	6	84	84	56
		(3.6)	2	14	6	84	84	56
	$\ (\frac{p_x}{\Delta}, S^{-1} p_s)\  \leq 1$	(3.5)	2	13	6	80	80	52
		(3.6)	2	13	6	80	80	52

COSHFUN

Critério	Região	atual. S	obj.	#int	#ext	#funções	# $\nabla$	# $\nabla^2$
BHN	$\ (p_x, S^{-1} p_s)\  \leq \Delta$	(3.5)	-0.6614	20	7	112	108	76
		(3.6)	-0.6614	20	7	112	108	76
	$\ (\frac{p_x}{\Delta}, S^{-1} p_s)\  \leq 1$	(3.5)	-0.6614	19	7	108	104	72
		(3.6)	-0.6614	19	7	108	104	72
Sugestão	$\ (p_x, S^{-1} p_s)\  \leq \Delta$	(3.5)	-0.6614	15	6	88	84	56
		(3.6)	-0.6614	15	6	88	84	56
	$\ (\frac{p_x}{\Delta}, S^{-1} p_s)\  \leq 1$	(3.5)	-0.6613	14	6	84	80	52
		(3.6)	-0.6613	14	6	84	80	52

DEMYMALO

Critério	Região	atual. S	obj.	#int	#ext	#funções	# $\nabla$	# $\nabla^2$
BHN	$\ (p_x, S^{-1} p_s)\  \leq \Delta$	(3.5)	-3	17	7	104	96	64
		(3.6)	-3	20	7	116	108	76
	$\ (\frac{p_x}{\Delta}, S^{-1} p_s)\  \leq 1$	(3.5)	-3	18	7	104	100	68
		(3.6)	-3	18	7	108	100	68
Sugestão	$\ (p_x, S^{-1} p_s)\  \leq \Delta$	(3.5)	-3	19	6	108	100	72
		(3.6)	-3	18	6	104	96	68
	$\ (\frac{p_x}{\Delta}, S^{-1} p_s)\  \leq 1$	(3.5)	-3	15	6	88	88	60
		(3.6)	-3	22	6	120	112	84

EXPFITA

Critério	Região	atual. S	obj.	#int	#ext	#funções	# $\nabla$	# $\nabla^2$
BHN	$\ (p_x, S^{-1} p_s)\  \leq \Delta$	(3.5)	0.001164	24	7	736	736	552
		(3.6)	0.001165	18	7	598	598	414
	$\ (\frac{p_x}{\Delta}, S^{-1} p_s)\  \leq 1$	(3.5)	0.001164	31	7	897	897	713
		(3.6)	0.001163	21	7	667	667	483
Sugestão	$\ (p_x, S^{-1} p_s)\  \leq \Delta$	(3.5)	0.001304	40	6	1081	943	782
		(3.6)	0.001304	17	6	552	552	391
	$\ (\frac{p_x}{\Delta}, S^{-1} p_s)\  \leq 1$	(3.5)	0.001315	91	6	2254	2254	2093
		(3.6)	0.001317	88	6	2185	2185	2024

continua...

Tabela 4.2: Resultados numéricos (continuação)

## GIGOMEZ1

Critério	Região	atual. S	obj.	#int	#ext	#funções	# $\nabla$	# $\nabla^2$
BHN	$\ (p_x, S^{-1} p_s)\  \leq \Delta$	(3.5)	-3	27	7	148	128	96
		(3.6)	-3	21	7	120	112	80
	$\ (\frac{p_x}{\Delta}, S^{-1} p_s)\  \leq 1$	(3.5)	-3	20	7	112	108	76
		(3.6)	-3	20	7	112	108	76
Sugestão	$\ (p_x, S^{-1} p_s)\  \leq \Delta$	(3.5)	-3	29	6	152	132	104
		(3.6)	-3	20	6	112	104	76
	$\ (\frac{p_x}{\Delta}, S^{-1} p_s)\  \leq 1$	(3.5)	-3	16	6	92	88	60
		(3.6)	-3	18	6	100	96	68

## GOFFIN

Critério	Região	atual. S	obj.	#int	#ext	#funções	# $\nabla$	# $\nabla^2$
BHN	$\ (p_x, S^{-1} p_s)\  \leq \Delta$	(3.5)	0.00032	18	7	1326	1326	918
		(3.6)	0.0003235	53	7	3111	3111	2703
	$\ (\frac{p_x}{\Delta}, S^{-1} p_s)\  \leq 1$	(3.5)	0.0001139	66	8	3825	3825	3366
		(3.6)	0.0001203	92	8	5151	5151	4692
Sugestão	$\ (p_x, S^{-1} p_s)\  \leq \Delta$	(3.5)	0.0016	16	6	1173	1173	816
		(3.6)	0.0016	33	6	2040	2040	1683
	$\ (\frac{p_x}{\Delta}, S^{-1} p_s)\  \leq 1$	(3.5)	0.002244	54	6	3111	3111	2754
		(3.6)	0.002108	75	6	4182	4182	3825

## HS10

Critério	Região	atual. S	obj.	#int	#ext	#funções	# $\nabla$	# $\nabla^2$
BHN	$\ (p_x, S^{-1} p_s)\  \leq \Delta$	(3.5)	-1	14	7	44	44	28
		(3.6)	-1	14	7	44	44	28
	$\ (\frac{p_x}{\Delta}, S^{-1} p_s)\  \leq 1$	(3.5)	-1	14	7	44	44	28
		(3.6)	-1	14	7	44	44	28
Sugestão	$\ (p_x, S^{-1} p_s)\  \leq \Delta$	(3.5)	-1	12	6	38	38	24
		(3.6)	-1	12	6	38	38	24
	$\ (\frac{p_x}{\Delta}, S^{-1} p_s)\  \leq 1$	(3.5)	-1	14	6	42	42	28
		(3.6)	-1	14	6	42	42	28

## HS12

Critério	Região	atual. S	obj.	#int	#ext	#funções	# $\nabla$	# $\nabla^2$
BHN	$\ (p_x, S^{-1} p_s)\  \leq \Delta$	(3.5)	-30	11	7	38	36	20
		(3.6)	-30	16	7	48	46	30
	$\ (\frac{p_x}{\Delta}, S^{-1} p_s)\  \leq 1$	(3.5)	-30	10	7	36	36	20
		(3.6)	-30	11	7	38	38	22
Sugestão	$\ (p_x, S^{-1} p_s)\  \leq \Delta$	(3.5)	-30	10	6	34	32	18
		(3.6)	-30	14	6	42	40	26
	$\ (\frac{p_x}{\Delta}, S^{-1} p_s)\  \leq 1$	(3.5)	-30	12	6	38	38	24
		(3.6)	-30	9	6	32	32	18

continua...

Tabela 4.2: Resultados numéricos (continuação)

HS22

Critério	Região	atual. S	obj.	#int	#ext	#funções	# $\nabla$	# $\nabla^2$
BHN	$\ (p_x, S^{-1} p_s)\  \leq \Delta$	(3.5)	1	11	7	57	57	33
		(3.6)	1	12	7	60	60	36
	$\ (\frac{p_x}{\Delta}, S^{-1} p_s)\  \leq 1$	(3.5)	1	12	7	60	60	36
		(3.6)	1	12	7	60	60	36
Sugestão	$\ (p_x, S^{-1} p_s)\  \leq \Delta$	(3.5)	1	11	6	54	54	33
		(3.6)	1	11	6	54	54	33
	$\ (\frac{p_x}{\Delta}, S^{-1} p_s)\  \leq 1$	(3.5)	1	13	6	60	60	39
		(3.6)	1	13	6	60	60	39

HS29

Critério	Região	atual. S	obj.	#int	#ext	#funções	# $\nabla$	# $\nabla^2$
BHN	$\ (p_x, S^{-1} p_s)\  \leq \Delta$	(3.5)	-22.63	11	7	38	38	22
		(3.6)	-22.63	11	7	38	38	22
	$\ (\frac{p_x}{\Delta}, S^{-1} p_s)\  \leq 1$	(3.5)	-22.63	11	7	38	38	22
		(3.6)	-22.63	11	7	38	38	22
Sugestão	$\ (p_x, S^{-1} p_s)\  \leq \Delta$	(3.5)	-22.63	9	6	32	32	18
		(3.6)	-22.63	10	6	34	34	20
	$\ (\frac{p_x}{\Delta}, S^{-1} p_s)\  \leq 1$	(3.5)	-22.63	9	6	32	32	18
		(3.6)	-22.63	10	6	34	34	20

HS43

Critério	Região	atual. S	obj.	#int	#ext	#funções	# $\nabla$	# $\nabla^2$
BHN	$\ (p_x, S^{-1} p_s)\  \leq \Delta$	(3.5)	-44	14	7	88	88	56
		(3.6)	-44	14	7	88	88	56
	$\ (\frac{p_x}{\Delta}, S^{-1} p_s)\  \leq 1$	(3.5)	-44	15	7	92	92	60
		(3.6)	-44	18	7	104	104	72
Sugestão	$\ (p_x, S^{-1} p_s)\  \leq \Delta$	(3.5)	-44	15	6	88	88	60
		(3.6)	-44	15	6	88	88	60
	$\ (\frac{p_x}{\Delta}, S^{-1} p_s)\  \leq 1$	(3.5)	-44	15	6	88	88	60
		(3.6)	-44	17	6	96	96	68

HS268

Critério	Região	atual. S	obj.	#int	#ext	#funções	# $\nabla$	# $\nabla^2$
BHN	$\ (p_x, S^{-1} p_s)\  \leq \Delta$	(3.5)	3.713e-06	15	7	138	138	90
		(3.6)	3.708e-06	16	7	144	144	96
	$\ (\frac{p_x}{\Delta}, S^{-1} p_s)\  \leq 1$	(3.5)	3.712e-06	21	7	174	174	126
		(3.6)	3.72e-06	17	7	150	150	102
Sugestão	$\ (p_x, S^{-1} p_s)\  \leq \Delta$	(3.5)	1.773e-05	15	6	132	132	90
		(3.6)	1.778e-05	15	6	132	132	90
	$\ (\frac{p_x}{\Delta}, S^{-1} p_s)\  \leq 1$	(3.5)	1.78e-05	20	6	162	162	120
		(3.6)	1.779e-05	17	6	144	144	102

continua...

Tabela 4.2: Resultados numéricos (continuação)

KIWCRES

Critério	Região	atual. S	obj.	#int	#ext	#funções	# $\nabla$	# $\nabla^2$
BHN	$\ (p_x, S^{-1} p_s)\  \leq \Delta$	(3.5)	1.28e-05	24	7	96	81	57
		(3.6)	1.88e-05	35	7	162	81	57
	$\ (\frac{p_x}{\Delta}, S^{-1} p_s)\  \leq 1$	(3.5)	1.28e-05	22	7	90	81	57
		(3.6)	1.273e-05	25	7	114	75	51
Sugestão	$\ (p_x, S^{-1} p_s)\  \leq \Delta$	(3.5)	6.481e-05	18	6	75	60	39
		(3.6)	9.256e-05	33	6	147	78	57
	$\ (\frac{p_x}{\Delta}, S^{-1} p_s)\  \leq 1$	(3.5)	6.394e-05	16	6	69	60	39
		(3.6)	6.408e-05	18	6	75	66	45

MADSEN

Critério	Região	atual. S	obj.	#int	#ext	#funções	# $\nabla$	# $\nabla^2$
BHN	$\ (p_x, S^{-1} p_s)\  \leq \Delta$	(3.5)	0.6164	20	7	196	182	126
		(3.6)	0.6164	21	7	203	189	133
	$\ (\frac{p_x}{\Delta}, S^{-1} p_s)\  \leq 1$	(3.5)	0.6164	19	7	189	182	126
		(3.6)	0.6164	20	7	196	182	126
Sugestão	$\ (p_x, S^{-1} p_s)\  \leq \Delta$	(3.5)	0.6165	16	6	161	154	105
		(3.6)	0.6165	18	6	175	161	112
	$\ (\frac{p_x}{\Delta}, S^{-1} p_s)\  \leq 1$	(3.5)	0.6165	18	6	175	154	105
		(3.6)	0.6165	18	6	175	161	112

MIFFLIN1

Critério	Região	atual. S	obj.	#int	#ext	#funções	# $\nabla$	# $\nabla^2$
BHN	$\ (p_x, S^{-1} p_s)\  \leq \Delta$	(3.5)	-1	11	7	57	57	33
		(3.6)	-1	11	7	57	57	33
	$\ (\frac{p_x}{\Delta}, S^{-1} p_s)\  \leq 1$	(3.5)	-1	13	7	63	63	39
		(3.6)	-1	13	7	63	63	39
Sugestão	$\ (p_x, S^{-1} p_s)\  \leq \Delta$	(3.5)	-0.9999	9	6	48	48	27
		(3.6)	-0.9999	9	6	48	48	27
	$\ (\frac{p_x}{\Delta}, S^{-1} p_s)\  \leq 1$	(3.5)	-0.9999	11	6	54	54	33
		(3.6)	-0.9999	11	6	54	54	33

MIFFLIN2

Critério	Região	atual. S	obj.	#int	#ext	#funções	# $\nabla$	# $\nabla^2$
BHN	$\ (p_x, S^{-1} p_s)\  \leq \Delta$	(3.5)	-1	26	7	108	90	66
		(3.6)	-1	26	7	108	90	66
	$\ (\frac{p_x}{\Delta}, S^{-1} p_s)\  \leq 1$	(3.5)	-1	21	7	87	81	57
		(3.6)	-1	21	7	87	81	57
Sugestão	$\ (p_x, S^{-1} p_s)\  \leq \Delta$	(3.5)	-0.9999	21	6	90	72	51
		(3.6)	-0.9999	21	6	90	72	51
	$\ (\frac{p_x}{\Delta}, S^{-1} p_s)\  \leq 1$	(3.5)	-0.9999	22	6	99	69	48
		(3.6)	-0.9999	22	6	99	69	48

continua...

Tabela 4.2: Resultados numéricos (continuação)

MINMAXRB

Critério	Região	atual. S	obj.	#int	#ext	#funções	# $\nabla$	# $\nabla^2$
BHN	$\ (p_x, S^{-1} p_s)\  \leq \Delta$	(3.5)	2.56e-05	38	7	235	175	135
		(3.6)	2.56e-05	66	7	445	280	240
	$\ (\frac{p_x}{\Delta}, S^{-1} p_s)\  \leq 1$	(3.5)	3.903e-05	53	7	335	220	180
		(3.6)	3.92e-05	53	7	340	240	200
Sugestão	$\ (p_x, S^{-1} p_s)\  \leq \Delta$	(3.5)	0.000128	42	6	260	175	140
		(3.6)	0.000128	73	6	490	295	260
	$\ (\frac{p_x}{\Delta}, S^{-1} p_s)\  \leq 1$	(3.5)	0.000128	73	6	485	260	225
		(3.6)	0.00016	75	6	500	290	255

OET1

Critério	Região	atual. S	obj.	#int	#ext	#funções	# $\nabla$	# $\nabla^2$
BHN	$\ (p_x, S^{-1} p_s)\  \leq \Delta$	(3.5)	0.4039	14	7	154	154	98
		(3.6)	0.4039	14	7	154	154	98
	$\ (\frac{p_x}{\Delta}, S^{-1} p_s)\  \leq 1$	(3.5)	0.4039	14	7	154	154	98
		(3.6)	0.4039	14	7	154	154	98
Sugestão	$\ (p_x, S^{-1} p_s)\  \leq \Delta$	(3.5)	0.4039	11	6	126	126	77
		(3.6)	0.4039	12	6	133	133	84
	$\ (\frac{p_x}{\Delta}, S^{-1} p_s)\  \leq 1$	(3.5)	0.4039	12	6	133	133	84
		(3.6)	0.4039	12	6	133	133	84

OET2

Critério	Região	atual. S	obj.	#int	#ext	#funções	# $\nabla$	# $\nabla^2$
BHN	$\ (p_x, S^{-1} p_s)\  \leq \Delta$	(3.5)	0.07145	15	7	161	161	105
		(3.6)	0.07145	17	7	175	168	112
	$\ (\frac{p_x}{\Delta}, S^{-1} p_s)\  \leq 1$	(3.5)	0.07145	17	7	175	168	112
		(3.6)	0.07145	17	7	175	168	112
Sugestão	$\ (p_x, S^{-1} p_s)\  \leq \Delta$	(3.5)	0.07152	12	6	133	133	84
		(3.6)	0.07152	14	6	147	140	91
	$\ (\frac{p_x}{\Delta}, S^{-1} p_s)\  \leq 1$	(3.5)	0.07152	14	6	147	140	91
		(3.6)	0.07152	15	6	154	147	98

OET3

Critério	Região	atual. S	obj.	#int	#ext	#funções	# $\nabla$	# $\nabla^2$
BHN	$\ (p_x, S^{-1} p_s)\  \leq \Delta$	(3.5)	3.84e-05	14	7	154	154	98
		(3.6)	3.84e-05	14	7	154	154	98
	$\ (\frac{p_x}{\Delta}, S^{-1} p_s)\  \leq 1$	(3.5)	4.871e-05	20	7	196	196	140
		(3.6)	4.87e-05	20	7	196	196	140
Sugestão	$\ (p_x, S^{-1} p_s)\  \leq \Delta$	(3.5)	0.000192	12	6	133	133	84
		(3.6)	0.000192	12	6	133	133	84
	$\ (\frac{p_x}{\Delta}, S^{-1} p_s)\  \leq 1$	(3.5)	0.000275	16	6	161	161	112
		(3.6)	0.0002751	16	6	161	161	112

continua...

Tabela 4.2: Resultados numéricos (continuação)

OET4

Critério	Região	atual. S	obj.	#int	#ext	#funções	# $\nabla$	# $\nabla^2$
BHN	$\ (p_x, S^{-1} p_s)\  \leq \Delta$	(3.5)	3.84e-05	18	7	182	168	112
		(3.6)	3.84e-05	18	7	182	168	112
	$\ (\frac{p_x}{\Delta}, S^{-1} p_s)\  \leq 1$	(3.5)	4.879e-05	24	7	224	210	154
		(3.6)	4.911e-05	24	7	224	210	154
Sugestão	$\ (p_x, S^{-1} p_s)\  \leq \Delta$	(3.5)	0.000192	14	6	147	133	84
		(3.6)	0.000192	16	6	161	147	98
	$\ (\frac{p_x}{\Delta}, S^{-1} p_s)\  \leq 1$	(3.5)	0.0002356	20	6	189	175	126
		(3.6)	0.0001989	20	6	189	175	126

OET5

Critério	Região	atual. S	obj.	#int	#ext	#funções	# $\nabla$	# $\nabla^2$
BHN	$\ (p_x, S^{-1} p_s)\  \leq \Delta$	(3.5)	3.84e-05	24	7	224	196	140
		(3.6)	3.84e-05	18	7	182	168	112
	$\ (\frac{p_x}{\Delta}, S^{-1} p_s)\  \leq 1$	(3.5)	4.877e-05	25	7	231	224	168
		(3.6)	4.859e-05	25	7	231	210	154
Sugestão	$\ (p_x, S^{-1} p_s)\  \leq \Delta$	(3.5)	0.000192	17	6	168	154	105
		(3.6)	0.000192	15	6	161	147	98
	$\ (\frac{p_x}{\Delta}, S^{-1} p_s)\  \leq 1$	(3.5)	0.0002179	29	6	259	224	175
		(3.6)	0.0002631	18	6	175	175	126

POLAK1

Critério	Região	atual. S	obj.	#int	#ext	#funções	# $\nabla$	# $\nabla^2$
BHN	$\ (p_x, S^{-1} p_s)\  \leq \Delta$	(3.5)	2.718	18	7	78	78	54
		(3.6)	2.718	18	7	78	78	54
	$\ (\frac{p_x}{\Delta}, S^{-1} p_s)\  \leq 1$	(3.5)	2.718	16	7	72	72	48
		(3.6)	2.718	16	7	72	72	48
Sugestão	$\ (p_x, S^{-1} p_s)\  \leq \Delta$	(3.5)	2.718	15	6	66	66	45
		(3.6)	2.718	15	6	66	66	45
	$\ (\frac{p_x}{\Delta}, S^{-1} p_s)\  \leq 1$	(3.5)	2.718	15	6	66	66	45
		(3.6)	2.718	15	6	66	66	45

PT

Critério	Região	atual. S	obj.	#int	#ext	#funções	# $\nabla$	# $\nabla^2$
BHN	$\ (p_x, S^{-1} p_s)\  \leq \Delta$	(3.5)	0.1429	11	7	76	76	44
		(3.6)	0.1429	11	7	76	76	44
	$\ (\frac{p_x}{\Delta}, S^{-1} p_s)\  \leq 1$	(3.5)	0.1429	11	7	76	76	44
		(3.6)	0.1429	11	7	76	76	44
Sugestão	$\ (p_x, S^{-1} p_s)\  \leq \Delta$	(3.5)	0.1429	12	6	76	76	48
		(3.6)	0.1429	12	6	76	76	48
	$\ (\frac{p_x}{\Delta}, S^{-1} p_s)\  \leq 1$	(3.5)	0.1429	12	6	76	76	48
		(3.6)	0.1429	12	6	76	76	48

continua...

Tabela 4.2: Resultados numéricos (continuação)

ROSEMMX

Critério	Região	atual. S	obj.	#int	#ext	#funções	# $\nabla$	# $\nabla^2$
BHN	$\ (p_x, S^{-1} p_s)\  \leq \Delta$	(3.5)	-44	46	7	305	215	175
		(3.6)	-44	50	7	350	225	185
	$\ (\frac{p_x}{\Delta}, S^{-1} p_s)\  \leq 1$	(3.5)	-44	62	7	430	245	205
		(3.6)	-44	48	7	330	225	185
Sugestão	$\ (p_x, S^{-1} p_s)\  \leq \Delta$	(3.5)	-44	45	6	295	205	170
		(3.6)	-44	52	6	365	220	185
	$\ (\frac{p_x}{\Delta}, S^{-1} p_s)\  \leq 1$	(3.5)	-44	59	6	410	225	190
		(3.6)	-44	42	6	295	190	155

SIPOW1

Critério	Região	atual. S	obj.	#int	#ext	#funções	# $\nabla$	# $\nabla^2$
BHN	$\ (p_x, S^{-1} p_s)\  \leq \Delta$	(3.5)	-1	9	7	357	357	189
		(3.6)	-1	12	7	420	399	231
	$\ (\frac{p_x}{\Delta}, S^{-1} p_s)\  \leq 1$	(3.5)	-1	11	7	399	399	231
		(3.6)	-1	11	7	399	399	231
Sugestão	$\ (p_x, S^{-1} p_s)\  \leq \Delta$	(3.5)	-1	9	6	336	336	189
		(3.6)	-1	11	6	378	357	210
	$\ (\frac{p_x}{\Delta}, S^{-1} p_s)\  \leq 1$	(3.5)	-1	10	6	357	357	210
		(3.6)	-1	11	6	378	378	231

SIPOW1M

Critério	Região	atual. S	obj.	#int	#ext	#funções	# $\nabla$	# $\nabla^2$
BHN	$\ (p_x, S^{-1} p_s)\  \leq \Delta$	(3.5)	-1.012	11	7	399	399	231
		(3.6)	-1.012	16	8	525	504	315
	$\ (\frac{p_x}{\Delta}, S^{-1} p_s)\  \leq 1$	(3.5)	-1.012	13	7	441	441	273
		(3.6)	-1.012	13	7	441	441	273
Sugestão	$\ (p_x, S^{-1} p_s)\  \leq \Delta$	(3.5)	-1.012	11	6	378	378	231
		(3.6)	-1.012	14	6	441	420	273
	$\ (\frac{p_x}{\Delta}, S^{-1} p_s)\  \leq 1$	(3.5)	-1.012	12	6	399	399	252
		(3.6)	-1.012	13	6	420	420	273

SIPOW2M

Critério	Região	atual. S	obj.	#int	#ext	#funções	# $\nabla$	# $\nabla^2$
BHN	$\ (p_x, S^{-1} p_s)\  \leq \Delta$	(3.5)	-1	11	7	399	378	210
		(3.6)	-1	11	7	399	378	210
	$\ (\frac{p_x}{\Delta}, S^{-1} p_s)\  \leq 1$	(3.5)	-1	10	7	378	378	210
		(3.6)	-1	10	7	378	378	210
Sugestão	$\ (p_x, S^{-1} p_s)\  \leq \Delta$	(3.5)	-1	12	6	399	378	231
		(3.6)	-1	14	6	441	420	273
	$\ (\frac{p_x}{\Delta}, S^{-1} p_s)\  \leq 1$	(3.5)	-1	9	6	336	336	189
		(3.6)	-1	9	6	336	336	189

continua...

Tabela 4.2: Resultados numéricos (continuação)

## TFI1

Critério	Região	atual. S	obj.	#int	#ext	#funções	# $\nabla$	# $\nabla^2$
BHN	$\ (p_x, S^{-1} p_s)\  \leq \Delta$	(3.5)	5.335	13	7	252	252	156
		(3.6)	5.335	13	7	252	252	156
	$\ (\frac{p_x}{\Delta}, S^{-1} p_s)\  \leq 1$	(3.5)	5.335	24	7	420	300	204
		(3.6)	5.335	18	7	348	252	156
Sugestão	$\ (p_x, S^{-1} p_s)\  \leq \Delta$	(3.5)	5.335	12	6	228	228	144
		(3.6)	5.335	13	6	240	240	156
	$\ (\frac{p_x}{\Delta}, S^{-1} p_s)\  \leq 1$	(3.5)	5.335	24	6	408	288	204
		(3.6)	5.335	18	6	336	240	156

## TFI2

Critério	Região	atual. S	obj.	#int	#ext	#funções	# $\nabla$	# $\nabla^2$
BHN	$\ (p_x, S^{-1} p_s)\  \leq \Delta$	(3.5)	0.6479	17	7	300	300	204
		(3.6)	0.6479	17	7	300	300	204
	$\ (\frac{p_x}{\Delta}, S^{-1} p_s)\  \leq 1$	(3.5)	0.6479	20	7	336	336	240
		(3.6)	0.6479	20	7	336	336	240
Sugestão	$\ (p_x, S^{-1} p_s)\  \leq \Delta$	(3.5)	0.648	17	6	288	288	204
		(3.6)	0.648	17	6	288	288	204
	$\ (\frac{p_x}{\Delta}, S^{-1} p_s)\  \leq 1$	(3.5)	0.648	18	6	300	300	216
		(3.6)	0.648	18	6	300	300	216

## TFI3

Critério	Região	atual. S	obj.	#int	#ext	#funções	# $\nabla$	# $\nabla^2$
BHN	$\ (p_x, S^{-1} p_s)\  \leq \Delta$	(3.5)	4.301	14	8	276	276	168
		(3.6)	4.301	14	8	276	276	168
	$\ (\frac{p_x}{\Delta}, S^{-1} p_s)\  \leq 1$	(3.5)	4.301	20	8	348	348	240
		(3.6)	4.301	18	7	312	312	216
Sugestão	$\ (p_x, S^{-1} p_s)\  \leq \Delta$	(3.5)	4.301	13	6	240	240	156
		(3.6)	4.301	13	6	240	240	156
	$\ (\frac{p_x}{\Delta}, S^{-1} p_s)\  \leq 1$	(3.5)	4.301	17	6	288	288	204
		(3.6)	4.301	17	6	288	288	204

## WOMFLET

Critério	Região	atual. S	obj.	#int	#ext	#funções	# $\nabla$	# $\nabla^2$
BHN	$\ (p_x, S^{-1} p_s)\  \leq \Delta$	(3.5)	1.919e-05	27	7	140	116	84
		(3.6)	1.92e-05	29	7	164	128	96
	$\ (\frac{p_x}{\Delta}, S^{-1} p_s)\  \leq 1$	(3.5)	1.922e-05	31	7	164	128	96
		(3.6)	1.92e-05	31	7	164	136	104
Sugestão	$\ (p_x, S^{-1} p_s)\  \leq \Delta$	(3.5)	9.598e-05	23	6	120	96	68
		(3.6)	9.6e-05	35	6	196	136	108
	$\ (\frac{p_x}{\Delta}, S^{-1} p_s)\  \leq 1$	(3.5)	9.6e-05	24	6	124	104	76
		(3.6)	9.72e-05	31	6	160	132	104

## 4.4 Análise dos Resultados Numéricos

Para melhor análise destes dados, montamos duas tabelas: uma comparando a atualização da folga e outra comparando as regiões. As tabelas indicam o número de problemas em que o método teve o menor número de iterações internas ou externas, cálculo de funções, de gradientes e de Hessianas, em relação ao outro. E também indica quantas vezes os dois tiveram os mesmos números.

Tabela 4.3: Comparação da atualização da folga

Parada	Região	Atual. S	#int	#ext	#f	#grad	#hess
BHN	$\ (p_x, S^{-1} p_s)\  \leq \Delta$	(3.5)	13	1	13	12	12
		iguais	19	34	19	20	20
		(3.6)	3	0	3	3	3
	$\ (\frac{p_x}{\Delta}, S^{-1} p_s)\  \leq 1$	(3.5)	5	0	7	5	5
		iguais	25	34	23	23	23
		(3.6)	5	1	5	7	7
Sugestão	$\ (p_x, S^{-1} p_s)\  \leq \Delta$	(3.5)	15	0	15	15	15
		iguais	16	35	16	16	16
		(3.6)	4	0	4	4	4
	$\ (\frac{p_x}{\Delta}, S^{-1} p_s)\  \leq 1$	(3.5)	11	0	11	12	12
		iguais	18	35	18	17	17
		(3.6)	6	0	6	6	6

Analisando a Tabela 4.3, a qual se refere a atualização (3.6) da variável de folga, percebemos que, geralmente, esta atualização não traz, praticamente, nenhuma melhora para o algoritmo, tanto para região proposta por [4] como para a região (3.3) proposta neste trabalho.

Na Tabela 4.4 verificamos que a região (3.3) também não traz melhoras quando atualizamos as folgas por (3.6). Entretanto para o critério de parada proposto em [4] as regiões são bastantes equilibradas quando possibilitamos o aumento da variável de folga por (3.6). A região proposta neste trabalho resolveu com muito sucesso o exemplo do Marazzi e Nocedal, entretanto este fenômeno não ocorreu nos testes numéricos.

Embora os testes mostrem que as alterações propostas no Capítulo 3 não são muito promissoras é bom lembrar que Byrd, Gilbert e Nocedal [3] e Byrd, Hribar e Nocedal [4] propõem os algoritmos para problemas de grande porte, enquanto nossos problemas testes são de pequeno porte. Acreditamos ainda que as demais regiões

Tabela 4.4: Comparação entre as regiões

Critério	Atual. S	Região	#int	#ext	#funcao	#grad	#hess
BHN	(3.5)	$\ (p_x, S^{-1} p_s)\  \leq \Delta$	20	1	19	20	20
		iguais	6	34	7	10	10
		$\ (\frac{p_x}{\Delta}, S^{-1} p_s)\  \leq 1$	9	0	9	5	5
	(3.6)	$\ (p_x, S^{-1} p_s)\  \leq \Delta$	13	1	12	12	12
		iguais	8	32	9	12	12
		$\ (\frac{p_x}{\Delta}, S^{-1} p_s)\  \leq 1$	14	2	14	11	11
Sugestão	(3.5)	$\ (p_x, S^{-1} p_s)\  \leq \Delta$	22	0	22	20	20
		iguais	7	35	7	9	9
		$\ (\frac{p_x}{\Delta}, S^{-1} p_s)\  \leq 1$	6	0	6	6	6
	(3.6)	$\ (p_x, S^{-1} p_s)\  \leq \Delta$	17	0	17	15	15
		iguais	9	35	9	10	10
		$\ (\frac{p_x}{\Delta}, S^{-1} p_s)\  \leq 1$	9	0	9	10	10

propostas, as quais desvinculam o passo  $p_x$  do passo  $p_s$ , sejam mais promissoras, mas requerem um estudo mais aprofundado.

Uma dificuldade encontrada foi quanto à implementação das duas rotinas, dogleg e gradientes conjugados. Embora estas rotinas estejam bem estudadas teoricamente, suas implementações requerem certos detalhes que geralmente não estão bem descritos na literatura.

# Referências Bibliográficas

- [1] I. Bongartz, N. I. M. Gould, A. R. Conn, and Ph. L. Toint. CUTE: Constrained and unconstrained testing environment. *ACM Transactions on Mathematical Software*, 21:123–160, 1995.
- [2] R. H. Byrd. Robust trust region methods for constrained optimization. Third SIAM Conference on Optimization, 1987.
- [3] R. H. Byrd, J. C. Gilbert, and J. Nocedal. A trust region method based on interior point techniques for nonlinear programming. *Mathematical Programming*, 89(1):149–186, 2000.
- [4] R. H. Byrd, M. H. Hribar, and J. Nocedal. An interior point algorithm for large-scale nonlinear programming. *SIAM Journal on Optimization*, 9(4):877–900, 1999.
- [5] M. R. Celis, J. E. Dennis, and R. A. Tapia. A trust region strategy for nonlinear equality constrained optimization. In P. T. Boggs, R. H. Byrd, and R. B. Schnabel, editors, *Numerical Optimization 1984*, pages 71 – 82. SIAM, Philadelphia, 1985.
- [6] A. S. El-Bakry, R. A. Tapia, T. Tsuchya, and Y. Zhang. On the formulation and theory of the Newton interior-point method for nonlinear programming. *Journal of Optimization Theory and Applications*, 89:507–541, 1996.
- [7] A. V. Fiacco and G. P. McCormick. *Nonlinear Programming : Sequential Unconstrained Minimization Techniques*. John Wiley & Sons, New York, 1968. Reprint : Volume 4 of *SIAM Classics in Applied Mathematics*, SIAM Publications, Philadelphia, PA 19104–2688, USA, 1990.
- [8] A. Forsgren and P. E. Gill. Primal-dual interior methods for nonconvex nonlinear programming. *SIAM Journal on Optimization*, 8(4):1132–1152, 1998.

- [9] U. M. Garcia-Palomares and O. L. Mangasarian. Superlinearly convergent quasi-newton methods for nonlinearly constrained optimization problems. *Mathematical Programming*, 11:1–13, 1976.
- [10] D. M. Gay, M. L. Overton, and M. H. Wright. A primal-dual interior method for nonconvex nonlinear programming. In Y. Yuan, editor, *Advances in Nonlinear Programming*, pages 31–56. Kluwer Academic Publishers, Dordrecht, 1998.
- [11] F. M. A. Gomes, M. C. Maciel, and J.M. Martínez. Nonlinear programming algorithms using trust region and augmented lagrangians with nonmonotone penalty parameters. *Mathematical Programming*, 84(1):161–200, 1999.
- [12] S-P. Han. Superlinearly convergent variable metri algorithms for general nonlinear programming problems. *Mathematical Programming*, 11:263–282, 1976.
- [13] S-P. Han. A globally convergent method for nonlinear programming. *Journal of Optimization Theory and Applications*, 22:297–309, 1977.
- [14] N. K. Karmarkar. A new polynomial-time algorithm for linear programming. *Combinatorica*, 4:373–395, 1984.
- [15] N. K. Karmarkar. A new polynomial-time algorithm for linear programming. *Proceedings of the 16th Annual ACM Symposium on Theory of Computing*, pages 302–311, 1984.
- [16] M. Marazzi and J. Nocedal. Feasibility control in nonlinear optimization. Technical Report OTC 2000/04, Optimization Technology Center, March 2000.
- [17] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer Series in Operations Research. Springer-Verlag, 1999.
- [18] E. Omojokun. *Trust Region Algorithms for Optimization with Nonlinear Equality and Inequality Constraints*. PhD thesis, Dept. of Computer Science, University of Colorado, 1991.
- [19] T. Plantenga. *Large-Scale Nonlinear Constrained Optimization using Trust Regions*. PhD thesis, EECS Department, Northwestern University, 1994.
- [20] M. J. D. Powell. A hybrid method for nonlinear equations. In P. Rabinowitz, editor, *Numerical Methods for Nonlinear Algebraic Equations*. Gordon and Breach, London, 1970.

- [21] M. J. D. Powell. A fast algorithm for nonlinearly constrained optimization calculations. In G. A. Watson, editor, *Numerical Analysis Dundee*, pages 144–157. Springer-Verlag, Berlin, 1977.
- [22] M. J. D. Powell. Algorithms for nonlinear constraints that use Lagrangian functions. *Mathematical Programming*, 14:224–248, 1978.
- [23] M. J. D. Powell. The convergence of variable metric methods for nonlinearly constrained optimization calculations. In R. R. Meyer O. L. Mangasarian and S. M. Robinson, editors, *Nonlinear Programming 3*, pages 27–64. Academic Press, New York, 1978.
- [24] D. F. Shanno and R. J. Vanderbei. Interior-point methods for nonconvex nonlinear programming: Orderings and higher-order methods. *Mathematical Programming*, 87(2):303–316, 2000.
- [25] T. Steihaug. The conjugate gradient method and trust regions in large scale optimization. *SIAM J. Numer. Anal.*, 20:626–637, 1983.
- [26] R. J. Vanderbei and D. F. Shanno. An interior-point algorithm for nonconvex nonlinear programming. *Computational Optimization and Applications*, 13:231–252, 1999.
- [27] R. B. Wilson. *A Simplicial Algorithm for Concave Programming*. PhD thesis, Graduate School of Business Administration - Harvard University, 1963.
- [28] H. Yamashita, H. Yabe, and T. Tanabe. A globally and superlinearly convergent primal- dual interior point trust region method for large scale constrained optimization. Technical report, Mathematical Systems Inc., Shinjuku-ku, Tokyo, Japan, July 1997.