



RIAN PENACHI

**UMA INTRODUÇÃO À OTIMIZAÇÃO NÃO LINEAR E A SOLUÇÃO
DE PROBLEMAS SIMÉTRICOS VIA ALGENCAN**

CAMPINAS
2015



UNIVERSIDADE ESTADUAL DE CAMPINAS

Instituto de Matemática, Estatística
e Computação Científica

RIAN PENACHI

UMA INTRODUÇÃO À OTIMIZAÇÃO NÃO LINEAR E A SOLUÇÃO DE PROBLEMAS SIMÉTRICOS VIA ALGENCAN

Dissertação apresentada ao Instituto de Matemática, Estatística e Computação Científica da Universidade Estadual de Campinas como parte dos requisitos exigidos para a obtenção do título de Mestre em Matemática Aplicada e Computational.

Orientador: Luis Felipe Cesar da Rocha Bueno

ESTE EXEMPLAR CORRESPONDE À VERSÃO FINAL DA DIS-
SERTAÇÃO DEFENDIDA PELO ALUNO RIAN PENACHI, E
ORIENTADA PELO PROF. DR. LUIS FELIPE CESAR DA
ROCHA BUENO.

Assinatura do Orientador

A handwritten signature in blue ink, appearing to read "L. F. Bueno".

CAMPINAS
2015

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca do Instituto de Matemática, Estatística e Computação Científica
Ana Regina Machado - CRB 8/5467

P37i Penachi, Rian, 1989-
Uma introdução à otimização não linear e a solução de problemas simétricos via ALGENCAN / Rian Penachi. – Campinas, SP : [s.n.], 2015.
Orientador: Luis Felipe Cesar da Rocha Bueno.
Dissertação (mestrado profissional) – Universidade Estadual de Campinas, Instituto de Matemática, Estatística e Computação Científica.
1. ALGENCAN (Programa de computador). 2. Programação não-linear. 3. Otimização matemática. 4. Simetria (Matemática). 5. Lagrange, Funções de. 6. Didática. I. Bueno, Luis Felipe Cesar da Rocha, 1983-. II. Universidade Estadual de Campinas. Instituto de Matemática, Estatística e Computação Científica. III. Título.

Informações para Biblioteca Digital

Título em outro idioma: An introduction to nonlinear optimization and the solution of symmetric problems through ALGENCAN

Palavras-chave em inglês:

ALGENCAN (Computer file)

Nonlinear programming

Mathematical optimization

Symmetry (Mathematics)

Lagrangian functions

Didactics

Área de concentração: Matemática Aplicada e Computacional

Titulação: Mestre em Matemática Aplicada e Computacional

Banca examinadora:

Luis Felipe Cesar da Rocha Bueno [Orientador]

Sandra Augusta Santos

Francisco Nogueira Calmon Sobral

Data de defesa: 30-03-2015

Programa de Pós-Graduação: Matemática Aplicada e Computacional

Dissertação de Mestrado Profissional defendida em 30 de março de 2015 e
aprovada pela Banca Examinadora composta pelos Profs. Drs.

LFBueno

Prof. (a). Dr (a). LUIS FELIPE CESAR DA ROCHA BUENO

Sandra Augusta Santos

Prof. (a). Dr (a). SANDRA AUGUSTA SANTOS

Francisco N.C. Sobral

Prof. (a). Dr (a). FRANCISCO NOGUEIRA CALMON SOBRAL

Abstract

This work proposes a didactic approach concerning nonlinear unconstrained optimization and nonlinear equality constrained optimization, as well as a guide for the readers who need to install and use the ALGENCAN software. Exploring intuitive ideas of the subject, the focus of the text is to study, often in a geometric way, unconstrained problems and problems with equality constraints. For the case of unconstrained problems, it will be emphasized the relationship between optimization methods and methods for nonlinear systems, highlighting the Newton's Method. The main focus will be on the Augmented Lagrangian Method, particularly on ALGENCAN, which is one of the most well-established implementations in the literature. The difficulties found in computational methods to solve symmetric problems of nonlinear constrained optimization will be also studied. It will be shown, through several simple examples, how the symmetry of the problem affects the good performance of the method and the alternatives to overcome these difficulties. Moreover, more complete tests using CUTEst will be done, which will confirm that the modified algorithm introduced by us is, at least, as efficient as the original version of ALGENCAN.

Keywords: Nonlinear programming, Mathematical optimization, Didactic, Symmetry, Lagrangian functions, ALGENCAN.

Resumo

Este trabalho propõe uma abordagem didática acerca de otimização não linear irrestrita e com restrições de igualdade, assim como um guia para o leitor que necessita instalar e utilizar o software ALGENCAN. Prezando por explorar ideias intuitivas do tema, o texto foca em estudar, muitas vezes geometricamente, problemas irrestritos e problemas com restrições de igualdade. Para o caso sem restrições é enfatizada a relação entre métodos de otimização e métodos para zeros de sistemas não lineares, destacando o Método de Newton. Métodos do tipo Lagrangiano Aumentado são o enfoque principal, particularmente em ALGENCAN, que é uma de suas implementações mais bem estabelecidas na literatura. As dificuldades encontradas em métodos computacionais para resolver problemas simétricos de otimização não linear com restrições também são estudadas. São apresentados vários exemplos simples de como a simetria do problema afeta o bom desempenho do método e as alternativas para contornar estes obstáculos. Além do mais, testes mais completos usando a coleção CUTEst comprovam que o algoritmo modificado que introduzimos é, pelo menos, tão competitivo quanto a versão original de ALGENCAN.

Palavras-chave: Programação não linear, Otimização matemática, Didático, Simetria, Funções de Lagrange, ALGENCAN.

Sumário

Resumo	vii
Introdução	1
1 Conceitos básicos para otimização irrestrita	5
1.1 Condições de otimalidade	5
1.2 Direção de descida	8
1.3 Método do Gradiente	10
1.4 Método de Newton	19
2 Otimização não linear com restrições de igualdade	29
2.1 Restrições lineares	29
2.2 Restrições não lineares	41
3 Métodos para otimização não linear com restrições	49
3.1 Método da Penalidade Quadrática	49
3.2 Método do Lagrangiano Aumentado	63
4 ALGENCAN: instalação e utilização	75
4.1 Instalação do ALGENCAN	75
4.2 Utilização do ALGENCAN	85
4.2.1 Compreendendo o resultado obtido com o ALGENCAN	94
5 Problemas simétricos de otimização não linear	97
6 Alterações no ALGENCAN	111
6.1 Comparação entre os algoritmos na coleção CUTEst	116
7 Conclusão	125
Referências	127
A Comandos em Linux	129
B Exemplo de otimização com o ALGENCAN	131

Agradecimentos

Agradeço à minha mãe Rosely, à minha namorada Luana pelo companheirismo e compreensão e ao meu amigo Juliano pelas várias horas de estudo (inclusive aos domingos).

Ao Professor Francisco Sobral da Universidade Estadual de Maringá pela ajuda na parte computacional e pela correção rigorosa deste trabalho.

Agradeço também à Professora Sandra Augusta Santos pela sua leitura cuidadosa e suas importantes recomendações, contribuindo assim para melhorar muito a qualidade do texto final deste trabalho.

Por fim, mas não menos importante, ao meu orientador Luís Felipe pela paciência e empenho neste trabalho.

Introdução

Com uma breve pesquisa nas referências bibliográficas das ementas da disciplina de otimização não linear das universidades brasileiras nota-se uma carência de textos introdutórios destinados aos discentes de graduação. Existem poucos livros em português voltados a graduação, como por exemplo, em [7] e uma recente publicação em [11]. Em vista disso, propõem-se neste trabalho um material de apoio sobre alguns tópicos do tema, utilizando uma abordagem didática diferenciada dos livros, ressaltando a parte intuitiva, trazendo exemplos e comentários sobre o assunto. Com o intuito apresentado anteriormente, o foco do texto será nos problemas sem restrições e com restrições de igualdade, de modo a destacar as ideias geométricas e aproveitar a simplicidade da parte algébrica.

De modo geral, um problema de otimização consiste em encontrar um minimizador de uma função $f : \mathbb{R}^n \rightarrow \mathbb{R}$ dentro de um conjunto $\Omega \subset \mathbb{R}^n$. Este conjunto é chamado de região factível. O conjunto Ω pode ser todo o \mathbb{R}^n ou pode ser descrito por equações e/ou inequações. No primeiro caso o problema de otimização é dito irrestrito, no segundo caso dizemos que ele apresenta restrições de igualdade e/ou desigualdade. Para manter a simplicidade será assumido que todas as funções envolvidas na formulação do problema são suaves, ou seja, possuem quantas derivadas se precise.

É dito que um ponto $x^* \in \Omega$ é um minimizador global do problema de otimização se $f(x^*) \leq f(x)$ para todo $x \in \Omega$. No caso em que $f(x^*) \leq f(x)$ para todo x pertencente a Ω suficientemente próximo de x^* dizemos que x^* é um mínimo local. Desta forma, todo mínimo global é local. A Figura 1 ilustra este conceito.

Um aluno de Cálculo I responde sem dificuldades que no caso irrestrito de $f : \mathbb{R} \rightarrow \mathbb{R}$ suave todo minimizador local anula a derivada de f . Motivados por isso, neste trabalho são evidenciadas algumas relações entre métodos de otimização e métodos para encontrar zeros de sistemas não lineares.

Para problemas com restrições o enfoque será em métodos do tipo penalidade, sobretudo o Lagrangiano Aumentado. Nesses métodos é resolvida uma sequencia de problemas irrestritos, a fim de buscar uma aproximação para a solução do problema original.

Por fim, a atenção será voltada a estudar as dificuldades encontradas em métodos computacio-

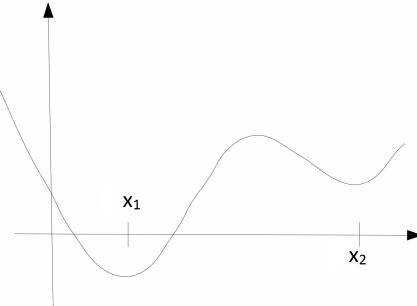


Figura 1: O ponto x_1 é o minimizador global de f e x_2 é um minimizador local.

nais para resolver problemas simétricos de otimização não linear. Ilustraremos através de exemplos simples como a simetria afeta o bom desempenho destes métodos e serão apresentadas as alternativas para contornar estas dificuldades. Além do mais, será usada a coleção CUTEst para verificar o comportamento em problemas gerais dos algoritmos que serão propostos. Esta dissertação está organizada como se segue.

No Capítulo 1 deste trabalho serão abordadas as *condições de optimalidade* de um problema de otimização e também o conceito de *direções de descida*, incluindo exemplos resolvidos e comentados. Além disso, serão vistos os métodos do gradiente e de Newton, dois dos métodos de otimização sem restrições mais utilizados.

No Capítulo 2 será estudada a otimização com restrições de igualdade, que por sua vez, está dividido em duas seções: restrições lineares e restrições não lineares.

No Capítulo 3 serão abordados o *Método de Penalidade Quadrática* com restrições de igualdade e também o *Método do Lagrangiano Aumentado*, que aqui é estudado sob uma perspectiva pouco explorada, a de penalidades deslocadas.

Uma ferramenta poderosa para resolver computacionalmente problemas de otimização não linear e ainda distante de alunos de graduação é o Algencan, que é um programa desenvolvido em Fortran. Neste trabalho se explica sua instalação e a sua correta utilização através de um tutorial simplificado e exemplos, com o intuito de servir como instrumento de aprofundamento nos estudos de otimização não linear e de servir como base em pesquisas científicas.

O Capítulo 4 é dedicado a explicar o funcionamento do Algencan, fazendo com que o leitor consiga instalar e utilizar corretamente esta poderosa ferramenta de otimização.

No Capítulo 5 constam exemplos de problemas simétricos de otimização não linear e testes realizados pelo Algencan com este tipo de problema.

O capítulo 6 é dedicado à criação de cinco modificações de Algencan para tentar superar as dificuldades apresentadas no capítulo anterior. Além disso, uma comparação entre os seis algoritmos (incluindo o original) é feita para analisar o desempenho em problemas gerais. Com isso, propõe-se modificações no Algencan a fim de melhorar seu desempenho em solucionar problemas de otimização não linear.

Conclusões e possibilidades de trabalhos futuros são apresentadas no Capítulo 7.

Capítulo 1

Conceitos básicos para otimização irrestrita

Neste capítulo serão abordadas as condições associadas a um ponto $x^* \in \mathbb{R}^n$, candidato a ser considerado um ponto ótimo (solução) de um problema de minimizar sem restrições. Estas condições são chamadas de *condições de otimalidade* e são importantes porque fornecem ideias de como encontrar possíveis minimizadores. Condições de otimalidade mais fortes permitem classificar x^* como um minimizador de f ou algum outro tipo de ponto crítico.

Além disso, será tratado o modo de escolher uma direção $d \in \mathbb{R}^n$ que diminua o valor de f partindo de $x_0 \in \mathbb{R}^n$, um ponto inicial arbitrário. A isso se dá o nome de *direção de descida*. Este tipo de direção será fundamental para os métodos numéricos que serão explorados no fim deste capítulo.

1.1 Condições de otimalidade

As condições de otimalidade são divididas em duas classes: condições necessárias e condições suficientes. Além disso, a condição é dita de primeira ordem se está ligada à primeira derivada de uma função f e de segunda ordem se está ligada à segunda derivada.

Proposição 1.1.1 (Condição necessária de primeira ordem). Seja $f : \mathbb{R}^n \rightarrow \mathbb{R}$, se x^* é minimizador local de f , então $\nabla f(x^*) = 0$.

Demonstração. Dado um $d \in \mathbb{R}^n$ arbitrário, considere uma função auxiliar $g : \mathbb{R} \rightarrow \mathbb{R}$ tal que $g(\lambda) = f(x^* + \lambda d)$. Como x^* é um minimizador local de f , então $\lambda = 0$ é um minimizador local de $g(\lambda)$. Assim, pelos conceitos clássicos de Cálculo em uma variável deve-se ter $g'(0) = 0$. Pela

regra da cadeia: $g'(\lambda) = \nabla f(x^* + \lambda d)^T d$. Substituindo $\lambda = 0$ vem que

$$0 = g'(0) = \nabla f(x^*)^T d.$$

Como $d \in \mathbb{R}^n$ é arbitrário, então $\nabla f(x^*)$ deve ser ortogonal a todos os outros vetores do espaço. Portanto $\nabla f(x^*) = 0$. ■

Exemplo 1.1.1. Para $f(x_1, x_2) = x_1^6 + x_2^8$ tem-se que $x^* = (0, 0)^T$ é o minimizador global de f , pois $f(x_1, x_2) > f(x^*) \forall x \neq x^* \in \mathbb{R}^2$. Pelo resultado anterior, seria necessário ter que $\nabla f(x^*) = 0$. De fato, $\nabla f(x_1, x_2) = (6x_1^5, 8x_2^7)^T$, portanto, $\nabla f(x^*) = 0$ se, e somente se $x = (0, 0)^T$.

Proposição 1.1.2 (Condição necessária de segunda ordem). Sejam $f : \mathbb{R}^n \rightarrow \mathbb{R}$ e x^* um minimizador local de f , então

- (a) $\nabla f(x^*) = 0$;
- (b) $\nabla^2 f(x^*) \geq 0$ (semi-definida positiva).

Demonstração. A parte (a) já está provada pela Proposição 1.1.1. Para provar (b) será admitido que é sabido das funções de uma variável que se x^* é um ponto de mínimo, então $f''(x^*) \geq 0$. Utilizando a regra da cadeia para obter a segunda derivada de $g(\lambda) = f(x^* + \lambda d)$ tem-se:

$$g''(\lambda) = d^T \nabla^2 f(x^* + \lambda d) d.$$

Assim,

$$g''(0) = d^T \nabla^2 f(x^*) d \geq 0.$$

Como $d \in \mathbb{R}^n$ é arbitrário, então $\nabla^2 f(x^*) \geq 0$. ■

Exemplo 1.1.2. Ainda com a função do exemplo anterior será ilustrada a Proposição 1.1.2. Já foi visto que realmente $\nabla f(x^*) = 0$. Por isso, basta mostrar que $\nabla^2 f(x^*) \geq 0$. Para este exemplo a hessiana vale

$$\nabla^2 f(x) = \begin{pmatrix} 30x_1^4 & 0 \\ 0 & 56x_2^6 \end{pmatrix}.$$

Portanto, no ponto $x^* = (0, 0)$ tem-se

$$\nabla^2 f(x^*) = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \geq 0.$$

Proposição 1.1.3 (Condição suficiente de segunda ordem). Seja $f : \mathbb{R}^n \rightarrow \mathbb{R}$, se $\nabla f(x^*) = 0$ e $\nabla^2 f(x^*) > 0$ então $x^* \in \mathbb{R}^n$ é um minimizador local de f .

Demonstração. Considere $A = \{p \in \mathbb{R}^n | \|p\| = 1\}$, onde $\|\cdot\|$ é a norma Euclidiana, e uma função $\phi : A \rightarrow \mathbb{R}$ tal que

$$\phi(p) = p^T \nabla^2 f(x^*) p.$$

Por ϕ ser contínua e A ser fechado e limitado, a função ϕ possui máximo e mínimo em A . Seja b esse mínimo, então, pelo fato de $\nabla^2 f(x^*) > 0$,

$$\phi(p) \geq b > 0 \quad \forall p \in A.$$

Dado $d \in \mathbb{R}^n$ arbitrário não-nulo, como $\frac{d}{\|d\|} \in A$ então

$$d^T \nabla^2 f(x^*) d \geq b \|d\|^2. \quad (1.1.1)$$

Expandindo f por Taylor em torno de x^* vem

$$f(x^* + d) - f(x^*) = \nabla f(x^*)^T d + \frac{1}{2} d^T \nabla^2 f(x^*) d + E(\|d\|^2),$$

em que $\lim_{d \rightarrow 0} \frac{E(\|d\|^2)}{\|d\|^2} = 0$. Por hipótese, $\nabla f(x^*) = 0$ e portanto, usando (1.1.1), tem-se que

$$f(x^* + d) - f(x^*) \geq \frac{b}{2} \|d\|^2 + E(\|d\|^2).$$

Como $\frac{b}{2} \|d\|^2 > 0$, então para $\|d\|$ suficientemente pequeno $f(x^* + d) - f(x^*) > 0$. Assim, na vizinhança de x^* , com $x \neq x^*$, tem-se

$$f(x) > f(x^*).$$

Portanto, x^* é minimizador local de f . ■

Exemplo 1.1.3. Seja $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ tal que $f(x_1, x_2) = 3x_1^2 + 3x_2^2 - 4x_1x_2 + 2x_1 + 5$. Logo,

$$\nabla f(x_1, x_2) = (6x_1 - 4x_2 + 2, 6x_2 - 4x_1)^T$$

que se anula em $x^* = \left(-\frac{3}{5}, -\frac{2}{5}\right)$. Além disso,

$$\nabla^2 f(x_1, x_2) = \begin{pmatrix} 6 & -4 \\ -4 & 6 \end{pmatrix},$$

a qual é definida positiva. Portanto, x^* é um minimizador local de f .

1.2 Direção de descida

Dado um ponto $x_0 \in \mathbb{R}^n$ com $\nabla f(x_0) \neq 0$, pela Proposição 1, é sabido que x_0 não é minimizador de f . Desta forma, em toda vizinhança de x_0 existe um $\bar{x} \in \mathbb{R}^n$ tal que $f(\bar{x}) < f(x_0)$. Mas em qual direção d se deve caminhar para obter este \bar{x} ? Esta pergunta está associada com a possibilidade de se obter uma d em que a função decresce.

A taxa de variação de f em uma direção d é dada por sua derivada direcional

$$\lim_{t \rightarrow 0^+} \frac{f(x + td) - f(x)}{t} = \nabla f(x)^T d = \cos(\theta) \|\nabla f(x)\| \|d\|, \quad (1.2.1)$$

onde θ é o ângulo entre d e $\nabla f(x)$. A partir disso é possível fazer algumas análises. Como $\|\nabla f(x)\|$ e $\|d\|$ são sempre positivos, então o sinal desta derivada é regulado pelo $\cos(\theta)$. Considerando $\theta \in [0, \pi]$ tem-se:

- Para $\left(0 \leq \theta < \frac{\pi}{2}\right)$, $\cos(\theta) > 0$ o que implica que $\nabla f(x)^T d > 0$,
- Para $\left(\frac{\pi}{2} < \theta \leq \pi\right)$, $\cos(\theta) < 0$ o que implica que $\nabla f(x)^T d < 0$.

Se f tem variação positiva numa direção, isto é, $\cos(\theta) > 0$, então se tem aí uma direção de subida. Mas se $\cos(\theta) < 0$ a direção é de descida, pois a variação é negativa.

A proposição abaixo garante que uma vez que $\nabla f(x)^T d < 0$ então é possível reduzir o valor de f através de um processo simples.

Proposição 1.2.1. Sejam $x \in \mathbb{R}^n$ tal que $\nabla f(x) \neq 0$ e $d \in \mathbb{R}^n$ tal que $\nabla f(x)^T d < 0$. Então existe $\beta > 0$ tal que $f(x + \alpha d) < f(x) \forall \alpha \in (0, \beta]$.

Demonstração. Considere a função auxiliar $\phi(\alpha) \equiv f(x + \alpha d)$. Note que $\phi(0) = f(x)$. Pela regra da cadeia,

$$\phi'(0) = \nabla f(x)^T d = \lim_{\alpha \rightarrow 0^+} \frac{\phi(\alpha) - \phi(0)}{\alpha}.$$

Para um β suficientemente pequeno $0 < \alpha < \beta$, o sinal de $\phi'(0)$ e $\phi(\alpha) - \phi(0)$ deve ser o mesmo. Como $\nabla f(x)^T d < 0$, então $\phi(\alpha) - \phi(0) < 0$ para $0 < \alpha < \beta$. Portanto,

$$f(x + \alpha d) - f(x) < 0.$$

■

Do cálculo de várias variáveis é sabido que $\nabla f(x_0)$ é perpendicular à curva de nível de f que passa por x_0 .

Na Figura 1.1 estão representadas em verde vetores que indicam direções de subida e em azul vetores que indicam direções de descida de uma função $f : \mathbb{R}^2 \rightarrow \mathbb{R}$.

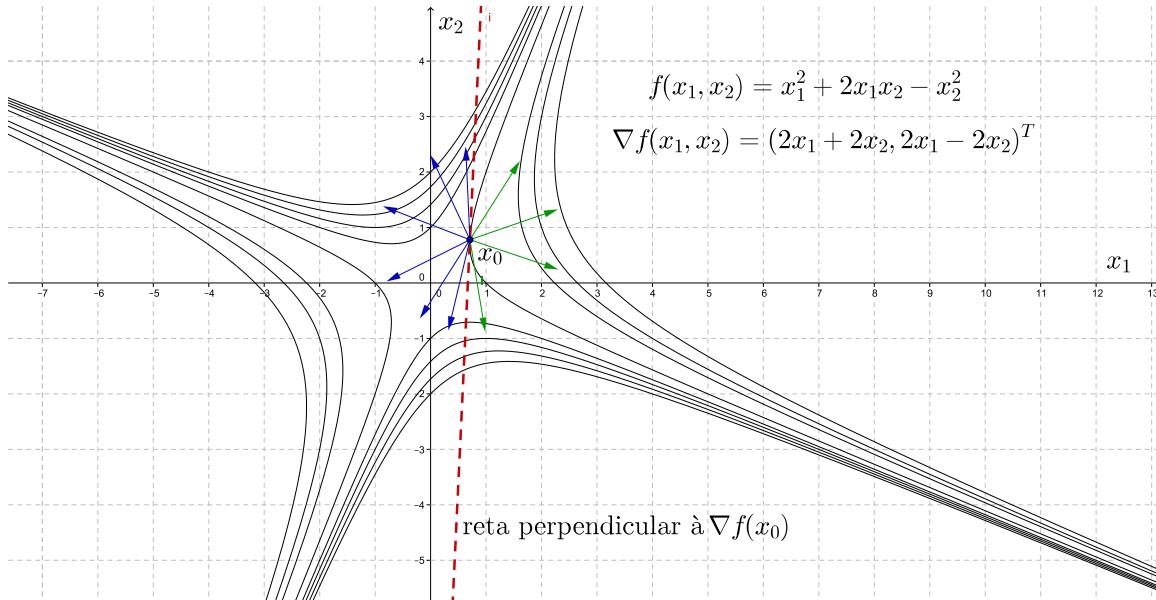


Figura 1.1: Direções de descida e de subida a partir de x_0 .

Observe que para uma direção d de descida não necessariamente tem-se que $f(x_0 + d) < f(x_0)$. Entretanto, para um passo α pequeno nesta direção, a Proposição 1.2.1 garante que $f(x_0 + \alpha d) < f(x_0)$.

Nas próximas seções serão abordados dois métodos muito utilizados para resolver problemas de otimização sem restrições, a saber: Método do Gradiente e Método de Newton. Nestes dois métodos busca-se reduzir o valor de f por meio de movimentos ao longo de direções de descida.

Métodos baseados em direções de descida geram uma sequência que, sob certas condições, converge à solução do problema. Esta sequência será denotada por $\{x_0, x_1, x_2, \dots\}$, onde cada $x_i \in \mathbb{R}^n$. Desta forma, é necessário cuidado para não confundir x_i , a i -ésima iteração de um método computacional, com a coordenada i de um vetor x . De modo geral, pelo contexto, ficará claro para o leitor qual notação estará sendo usada.

1.3 Método do Gradiente

O Método do Gradiente, que é um dos mais antigos de otimização, visa minimizar funções sempre “caminhando” na direção de $-\nabla f(x)$. O vetor oposto ao gradiente se encontra entre as direções de descida, pois, quando $\theta = \pi$ em (1.2.1), vale que $\cos(\theta) = -1 < 0$ e tem-se a uma taxa de decrescimento para f . Na verdade, o motivo de caminhar nesta direção é bastante intuitivo: uma vez que o vetor gradiente aponta sempre para a direção de maior crescimento, então a direção de maior decrescimento é a oposta à do gradiente. Por este motivo, o Método do Gradiente também é conhecido como Método do Declive Máximo (ou *steepest descent*).

Porém, andar na direção de $-\nabla f(x)$ não significa necessariamente caminhar uma distância igual a $\|-\nabla f(x)\|$. Por exemplo, o gráfico da função $f(x) = 2x_1^2 + x_2^2$ é um paraboloide elíptico e as curvas de nível de f estão ilustradas na Figura 1.2. Para o ponto $x_0 = (1, 0)^T$ são ilustradas as direções $\nabla f(x_0)$ e $-\nabla f(x_0)$, que são ortogonais à curva de nível neste ponto e são, respectivamente, as direções de maior crescimento e decrescimento de f . Contudo, se fosse dado o passo completo na direção $-\nabla f(x_0)$ seria obtido o ponto $x_1 = x_0 - \nabla f(x_0) = (-3, 0)^T$, para o qual o valor de f não é reduzido.

Entretanto, dado um ponto inicial x_0 , o próximo ponto x_1 poderá ser encontrado caminhando na direção de $-\nabla f(x_0)$, uma vez que, suficientemente próximo de x_0 é garantido que $f(x_1) < f(x_0)$. Mas quanto caminhar na direção de $-\nabla f(x_0)$? Deve-se tomar um passo $\lambda > 0$ de modo que para

$$x_1 = x_0 - \lambda \nabla f(x_0)$$

valha que $f(x_1) < f(x_0)$.

De modo geral tem-se que

$$x_{k+1} = x_k - \lambda_k \nabla f(x_k).$$

Para efeito de notação se usa $d_k = -\nabla f(x_k)$. Isto é feito assim pois na verdade d_k poderia ser qualquer direção de descida. Normalmente, $\lambda_k > 0$ é dito tamanho do passo e é determinado de

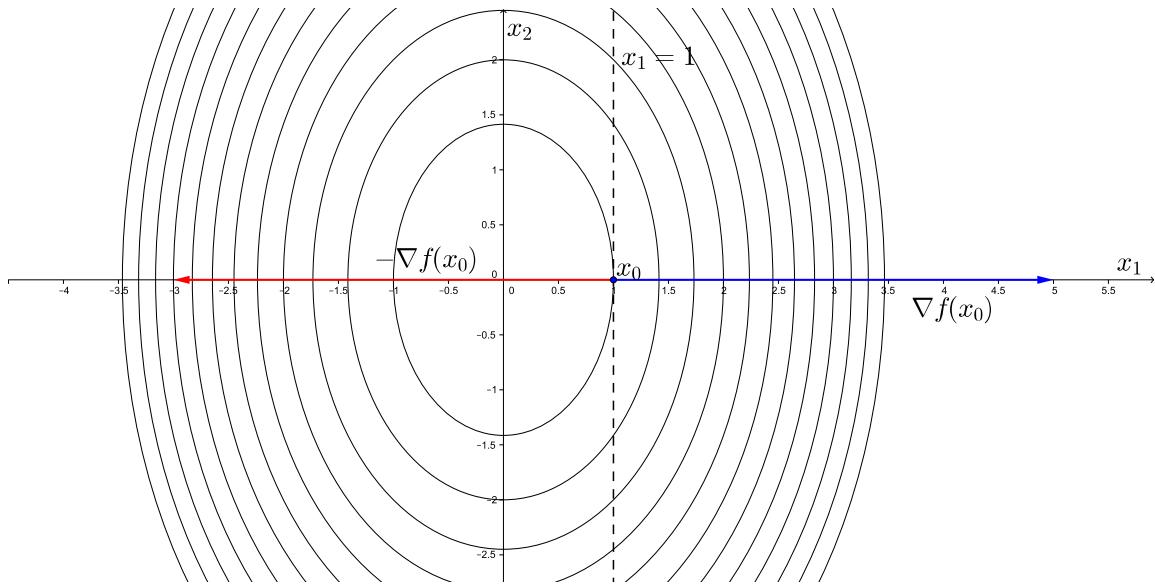


Figura 1.2: As direções $-\nabla f(x_0)$ e $\nabla f(x_0)$ são perpendiculares à reta tangente à curva de nível de f que passa por x_0 . Vale que $-\nabla f(x_0)$ é uma direção de descida para f mas, dando o passo completo, $x_1 = x_0 - \nabla f(x_0)$, obtém-se que $f(x_1) > f(x_0)$.

tal modo que

$$\underbrace{f(x_k + \lambda_k d_k)}_{x_{k+1}} < f(x_k).$$

Achar λ_k é resolver um subproblema chamado de Busca Linear.

Exemplo 1.3.1. Minimizar $f(x_1, x_2) = x_1^2 + x_1 x_2 + 2x_2^2 - 3$.

$$f(x_1, x_2) = x_1^2 + x_1 x_2 + 2x_2^2 - 3 \Rightarrow -\nabla f(x_1, x_2) = (-2x_1 - x_2, -x_1 - 4x_2)^T.$$

Se $x_0 = (1, 1)^T$, então $f(x_0) = f(1, 1) = 1$, $d_0 = -\nabla f(x_0) = (-3, -5)^T$ e tomando $\lambda_0 = 1$ vem que

$$\begin{aligned} x_1 &= x_0 + 1d_0 \\ &= (1, 1)^T + 1(-3, -5)^T \\ &= (-2, -4)^T. \end{aligned}$$

Porém, $f(x_1) = 41 > f(x_0) = 1$, ou seja, a função está crescendo. Deste modo, $\lambda_0 = 1$ não é um bom passo para caminhar na direção de $-\nabla f(x)$. A Figura 1.3 mostra este fato.

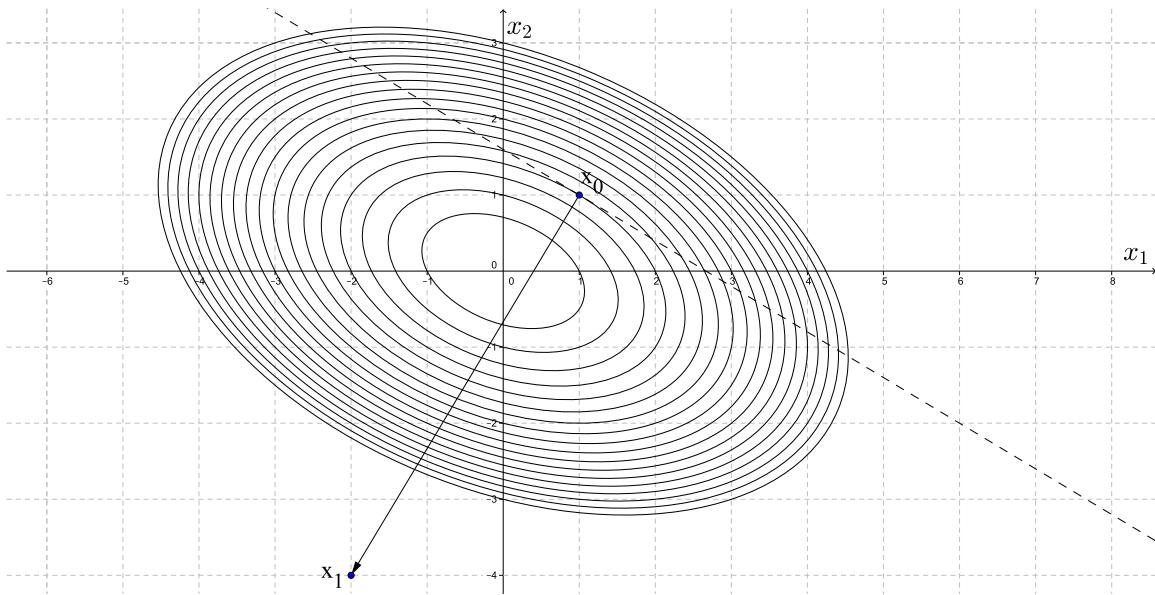


Figura 1.3: O passo completo em uma direção de descida pode não reduzir o valor de f , ou seja, $f(x + d) > f(x)$.

Uma opção agora é diminuir λ_0 , por exemplo tomar $\lambda_0^{(1)} = \frac{\lambda_0}{10}$, e fazer

$$\begin{aligned} x_1 &= x_0 + \lambda_0^{(1)} d_0 \\ &= x_0 + \frac{1}{10} d_0 \\ &= (1, 1)^T + \frac{1}{10} (-3, -5)^T \\ &= \left(\frac{7}{10}, \frac{1}{2} \right)^T. \end{aligned}$$

Agora observa-se $f(x_1) = -1,66 < f(x_0) = 1$ e assim x_1 é aceitável. É dito então, que o passo $\lambda_0^{(1)} = \frac{\lambda_0}{10}$ foi aceito. Para encontrar x_2 é possível se iniciar novamente o processo de Busca Linear

considerando, a priori, $\lambda_1 = 1$. Porém, neste exemplo, continuando com $\lambda_1 = \frac{1}{10}$, obtém-se que

$$\begin{aligned} x_2 &= x_1 + \frac{1}{10}d_1 \\ &= \left(\frac{7}{10}, \frac{1}{2}\right)^T + \frac{1}{10} \left(-\frac{1}{10}, -\frac{13}{10}\right)^T \\ &= \left(\frac{51}{100}, \frac{23}{100}\right)^T, \end{aligned}$$

onde $d_1 = -\nabla f(x_1)$, e portanto $f(x_2) = -2,5168 < f(x_1) = -1,66$. Seguindo para a terceira iteração ainda com $\lambda_2 = \frac{1}{10}$,

$$\begin{aligned} x_3 &= x_2 + \frac{1}{10}d_2 \\ &= \left(\frac{51}{100}, \frac{23}{100}\right)^T + \frac{1}{10} \left(-\frac{3}{4}, -\frac{57}{100}\right)^T \\ &= \left(\frac{77}{200}, \frac{2}{23}\right)^T \end{aligned}$$

e consequentemente novamente a função decresce: $f(x_3) \approx -2,8336 < f(x_2) = -2,5168$. Por observação, tomar $\lambda_k = \frac{1}{10}$ garante, para esta função, que $f(x_{k+1}) < f(x_k)$. Para as 8 primeiras iterações tem-se $f(x_k + d_k) > f(x_k)$ mas $f\left(x_k + \frac{1}{10}d_k\right) < f(x_k)$. Com isso

$$x_4 = \left(\frac{129}{431}, \frac{8}{585}\right); x_5 = \left(\frac{5}{21}, -\frac{1}{46}\right); x_6 = \left(\frac{173}{898}, -\frac{15}{407}\right);$$

$$x_7 = \left(\frac{92}{583}, -\frac{6}{145}\right); x_8 = \left(\frac{851}{6527}, -\frac{187}{4605}\right).$$

Uma possibilidade de se encerrar o método seria quando $x_{k+1} = x_k$, ou pelo menos quando esses valores estivessem tão próximos quanto se queira. Ou seja, para $\varepsilon \in \mathbb{R}_+$,

$$\|x_{k+1} - x_k\| < \varepsilon.$$

Pela Proposição 1.1.1, se x^* é minimizador da função, então $\nabla f(x^*) = 0$. Sendo assim, outro critério de parada seria

$$\|\nabla f(x_k)\| \leq \varepsilon.$$

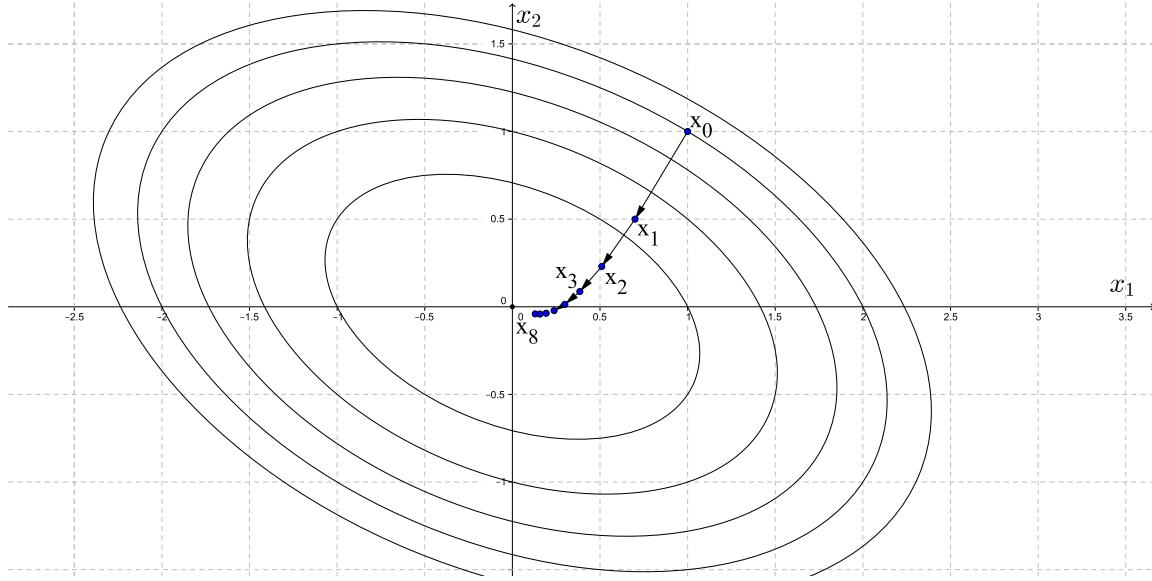


Figura 1.4: Convergência lenta do Método do Gradiente com Busca Linear.

No exemplo apresentado, $x^* = (0, 0)$ e $f(x^*) = -3$. Observa-se que

$$\{x_n\} = \{x_0, x_1, x_2, x_3, \dots\} \rightarrow x^*$$

ainda que muito lentamente.

Uma outra abordagem para determinar o tamanho do passo seria calculá-lo analiticamente de alguma maneira eficiente. Isto significa caminhar na direção de $-\nabla f(x)$ de modo que, nesta direção, $f(x)$ fica minimizada. Na prática, se faz isso criando uma função

$$h(\lambda_k) = f(x_k + \lambda_k d_k)$$

e depois minimizando-a.

No Exemplo 1.3.1:

$$\begin{aligned} x_1 &= x_0 + \lambda_0 d_0 \\ &= (1, 1)^T + \lambda_0 (-3, -5)^T \\ &= (-3\lambda_0 + 1, -5\lambda_0 + 1)^T. \end{aligned}$$

Portanto,

$$\begin{aligned} h(\lambda_0) &= (-3\lambda_0 + 1)^2 + (-3\lambda_0 + 1)(-5\lambda_0 + 1) + 2(-5\lambda_0 + 1)^2 - 3 \\ &= 74\lambda_0^2 - 34\lambda_0 + 1. \end{aligned}$$

Desta forma, para determinar λ_0 é preciso resolver o problema $\min h(\lambda_0)$.

Como $h(\lambda_0)$ é uma parábola com concavidade voltada para cima, então $h(\lambda_0)$ admite ponto de mínimo. Derivando $h(\lambda_0)$ e igualando a zero obtém-se

$$h'(\lambda_0) = 148\lambda_0 - 34 = 0 \Rightarrow \lambda_0 = \frac{17}{74}.$$

Deste modo,

$$\begin{aligned} x_1 &= (-3\lambda_0 + 1, -5\lambda_0 + 1)^T \\ &= \left(-3\left(\frac{17}{74}\right) + 1, -5\left(\frac{17}{74}\right) + 1 \right)^T \\ &= \left(\frac{23}{74}, -\frac{11}{74} \right)^T. \end{aligned}$$

Observe que $f(x_1) < f(x_0)$ e $f(x_1) \approx -2,9$, que é muito próximo de $f(x^*)$. Continuando a fazer as iterações obtém-se,

$$\begin{aligned} x_2 &= x_1 + \lambda_1 d_1 \\ &= \left(\frac{23}{74}, -\frac{11}{74} \right)^T + \lambda_1 \left(-\frac{35}{74}, \frac{21}{74} \right)^T \\ &= \left(\frac{23}{74} - \frac{35}{74}\lambda_1, -\frac{11}{74} + \frac{21}{74}\lambda_1 \right)^T \end{aligned}$$

e

$$h(\lambda_1) = \left(\frac{23}{74} - \frac{35}{74}\lambda_1 \right)^2 + \left(\frac{23}{74} - \frac{35}{74}\lambda_1 \right) \left(-\frac{11}{74} + \frac{21}{74}\lambda_1 \right) + \left(-\frac{11}{74} + \frac{21}{74}\lambda_1 \right)^2.$$

Achando λ_1 que minimiza $h(\lambda_1)$ conclui-se que

$$\lambda_1 = \frac{17}{28}.$$

Assim,

$$\begin{aligned} x_2 &= \left(\frac{23}{74} - \frac{35}{74} \left(\frac{17}{28} \right), -\frac{11}{74} + \frac{21}{74} \left(\frac{17}{28} \right) \right)^T \\ &= \left(\frac{7}{296}, \frac{7}{296} \right)^T. \end{aligned}$$

Tem-se $f(x_2) < f(x_1)$ e $f(x_2) \approx -2,998$. A sequência de pontos será

$$\begin{aligned} x_0 &= (1, 1); \quad x_1 \stackrel{\lambda_0=\frac{17}{74}}{=} (0.310811, -0.148649); \quad x_2 \stackrel{\lambda_1=\frac{17}{28}}{=} (0.023649, 0.023649); \\ x_3 &\stackrel{\lambda_2=\frac{17}{74}}{=} (0.00735, -0.003515); \quad x_4 \stackrel{\lambda_3=\frac{17}{28}}{=} (0.000559, 0.000559) \\ x_5 &\stackrel{\lambda_4=\frac{17}{74}}{=} (0.000174, -0.000083). \end{aligned}$$

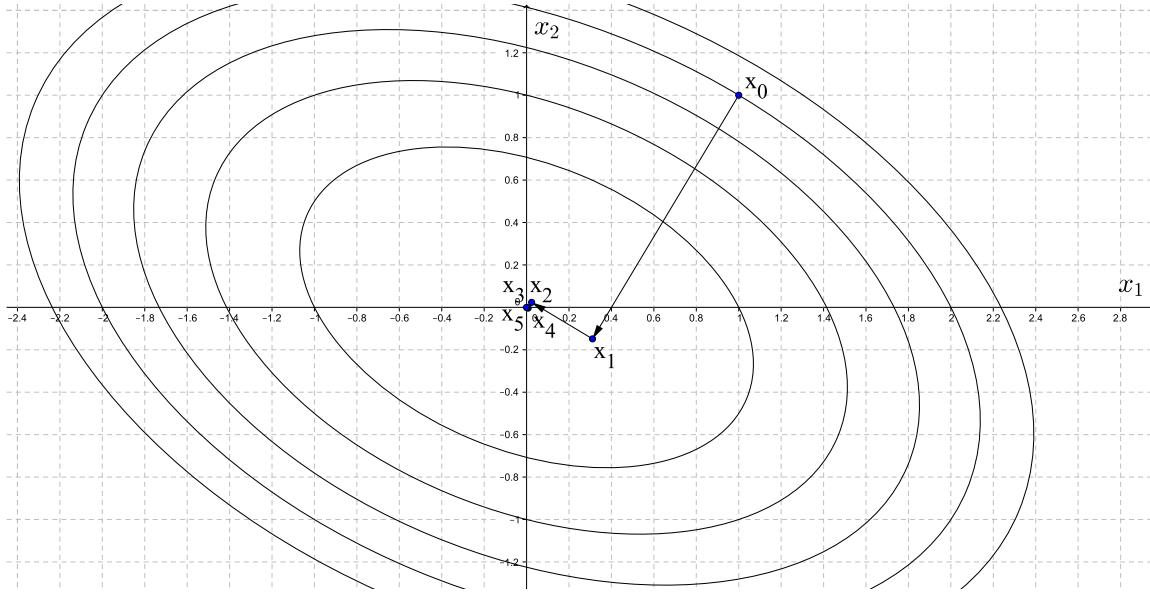


Figura 1.5: Convergência do Método do Gradiente com Busca Linear Exata.

Novamente

$$\{x_n\} = \{x_0, x_1, x_2, x_3, \dots\} \rightarrow x^*.$$

A Figura 1.5 mostra geometricamente a convergência da sequência dos pontos obtidos.

Essa busca linear que visa minimizar $h(\lambda_k)$ é dita Busca Linear Exata. Nesta busca há o

inconveniente de resolver o problema unidimensional de minimizar $h(\lambda_k)$ a cada iteração, que nem sempre é fácil de solucionar. No caso especial que f é quadrática isso pode ser feito sem dificuldades.

Teorema 1.3.1. Seja $f : \mathbb{R}^n \rightarrow \mathbb{R}$ quadrática e estritamente convexa, escrita matricialmente como

$$f(x) = \frac{1}{2}x^T Gx + x^T b + c,$$

com $G \in \mathbb{R}^{n \times n}$, $b \in \mathbb{R}^n$ e $c \in \mathbb{R}$. Então em todas as iterações o valor de λ_k é dado por $\lambda_k = \frac{\nabla f(x_k)^T \nabla f(x_k)}{\nabla f(x_k)^T G \nabla f(x_k)}$.

Demonstração. Como $f(x)$ é uma quadrática estritamente convexa, tem-se que G é definida positiva (vide [7] proposição 3.2, página 19). Além disso, derivando f , obtém-se que

$$\nabla f(x) = Gx + b.$$

Definindo a função $h(\lambda) = f(x_k + \lambda d_k)$ obtém-se λ_k minimizando-a. Para isso nota-se que

$$h(\lambda) = f(x_k + \lambda d_k) \Rightarrow h'(\lambda) = \nabla f(x_k + \lambda d_k)^T d_k.$$

Portanto,

$$\nabla f(x_k + \lambda_k d_k)^T d_k = 0 \Rightarrow (G(x_k + \lambda_k d_k) + b)^T d_k = 0 \Rightarrow (Gx_k + b)^T d_k + \lambda_k (Gd_k)^T d_k = 0.$$

Deste modo,

$$\lambda_k d_k^T G d_k = -(Gx_k + b)^T d_k = -\nabla f(x_k)^T d_k. \quad (1.3.1)$$

Como $d_k^T G d_k > 0$, para $d_k \neq 0$, obtém-se

$$\lambda_k = -\frac{d_k^T \nabla f(x_k)}{d_k^T G d_k}.$$

Como $d_k = -\nabla f(x_k)$ tem-se

$$\lambda_k = \frac{\nabla f(x_k)^T \nabla f(x_k)}{\nabla f(x_k)^T G \nabla f(x_k)}$$

e o resultado está demonstrado. ■

Vale observar que se $\nabla f(x_k + \lambda_k d_k) = 0$, então $h'(\lambda_k) = 0$ mas a recíproca não é verdadeira. No exemplo feito $\lambda_1 = \frac{17}{74}$ e com isso $h'(\lambda_1) = 0$, mas $\nabla f(x_2) \neq 0$.

Outro detalhe é que, na Busca Linear Exata, $h'(\lambda_k) = 0$ para todo $k \in \mathbb{N}$. Mas $h'(\lambda_k) = \nabla f(x_k + \lambda_k d_k)^T d_k$, então, $\nabla f(x_k + \lambda_k d_k)^T d_k = 0$. Contudo, $x_k + \lambda_k d_k = x_{k+1}$ e $d_k = -\nabla f(x_k)$, deste modo conclui-se que

$$\nabla f(x_{k+1})^T \nabla f(x_k) = 0.$$

Este resultado quer dizer que em duas iterações sucessivas os gradientes respectivos de f em x_k e x_{k+1} são ortogonais.

Sendo assim, se $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ com hessiana definida positiva é quadrática, então é correto afirmar que em toda iteração cujo índice é par o tamanho do passo a ser dado será igual. O mesmo acontecerá com os índices ímpares. Tal fato é consequência direta da ortogonalidade dos gradientes entre duas iterações sucessivas, pois a cada duas iterações intercaladas os gradientes serão paralelos, isto é

$$\nabla f(x_k) = \alpha \nabla f(x_{k+2}).$$

Deste modo,

$$\lambda_k = \frac{\nabla f(x_k)^T \nabla f(x_k)}{\nabla f(x_k)^T G \nabla f(x_k)} = \frac{(\alpha \nabla f(x_{k+2}))^T (\alpha \nabla f(x_{k+2}))}{(\alpha \nabla f(x_{k+2}))^T G (\alpha \nabla f(x_{k+2}))} = \frac{\alpha^2 \nabla f(x_{k+2})^T \nabla f(x_{k+2})}{\alpha^2 \nabla f(x_{k+2})^T G \nabla f(x_{k+2})} = \lambda_{k+2}.$$

Observe na sequência de passos do Exemplo 1.3.1 que $\lambda_0 = \lambda_2 = \lambda_4$ e $\lambda_1 = \lambda_3$.

Para finalizar, é notória a lentidão da convergência do Método do Gradiente com a Busca Linear Exata devido ao zigue-zague decorrente das direções ortogonais. Esse fenômeno é mais prejudicial caso as curvas de nível da função sejam elipses com maior excentricidade. Além disso, se a técnica da Busca Linear Exata for utilizada para funções não quadráticas tem-se que, a cada iteração, resolver um subproblema que nem sempre tem fácil solução. Neste caso é possível usar um processo iterativo para a determinação de λ_k e tomar como aproximação inicial o passo

$$\lambda_k^{(0)} = \frac{\nabla f(x_k)^T \nabla f(x_k)}{\nabla f(x_k)^T \nabla^2 f(x_k) \nabla f(x_k)}.$$

Caso o minimizador do subproblema unidimensional seja encontrado exatamente, é mostrado (veja, por exemplo, página 241 de [12]) que, sob certas condições, a taxa de convergência depende da razão entre a diferença e a soma do maior e do menor autovalores da hessiana na solução.

1.4 Método de Newton

O Método de Newton para otimização está baseado no fato que os pontos estacionários de $f(x)$ são zeros de sua derivada. Isto significa que buscamos, primeiramente, satisfazer a condição de otimalidade

$$\nabla f(x^*) = 0. \quad (1.4.1)$$

Entretanto, se usa também informações associadas às condições de segunda ordem para ter mais chances que x^* seja um ponto de mínimo, uma vez que satisfazer (1.4.1) é garantia apenas que x^* é um ponto crítico de $f(x)$.

No Cálculo Numérico, como pode ser visto em [15], são estudados métodos para resolver sistemas não lineares ($F(x) = 0$), onde $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$. A aproximação linear de Taylor

$$F(x) \approx F(x_k) + J_F(x_k)(x - x_k),$$

nos dá uma boa estimativa para encontrar o zero de F . Deste modo, quando $J_F(x_k)$ é não singular, a iteração de Newton está bem definida e é obtida zerando a aproximação linear, ou seja

$$x_{k+1} = x_k - J_F(x_k)^{-1}F(x_k). \quad (1.4.2)$$

No caso particular em que $F(x) = \nabla f(x)$ a iteração descrita em (1.4.2) se torna

$$x_{k+1} = x_k - \nabla^2 f(x_k)^{-1}\nabla f(x_k).$$

Fazendo, $d_k = x_{k+1} - x_k$ tem-se que

$$d_k = -\nabla^2 f(x_k)^{-1}\nabla f(x_k). \quad (1.4.3)$$

A direção d_k em (1.4.3) é chamada de direção de Newton para otimização. Para o caso em que valem as condições suficientes de segunda ordem na solução x^* , é verdade que para todo x_k perto de x^* a hessiana $\nabla^2 f(x_k)$ será definida positiva e, portanto, não singular. A seguir será apresentada outra maneira de interpretar essa direção.

Em uma variável, usando o Método de Newton para achar o zero de $g(x)$ se faz a aproximação

$$g(x) \approx g(x_k) + g'(x_k)(x - x_k).$$

Para o caso em que $g(x) = f'(x)$ tem-se

$$f'(x) \approx f'(x_k) + f''(x_k)(x - x_k).$$

Integrando dos dois lados obtém-se

$$f(x) = f'(x_k)x + \frac{1}{2}f''(x_k)(x - x_k)^2 + C.$$

Tomando $C = f(x_k) - f'(x_k)x_k$, obtém-se

$$f(x) = f(x_k) + f'(x_k)(x - x_k) + \frac{1}{2}f''(x_k)(x - x_k)^2.$$

Isto quer dizer que o Método de Newton aplicado à derivada de $f(x)$ é equivalente a aproximar $f(x)$ pela parábola de Taylor e obter seu ponto extremo. Caso $f''(x_k) > 0$ o ponto obtido é o minimizador da aproximação.

Da maneira apresentada, a escolha da direção em (1.4.3) foi feita levando em consideração primordialmente a condição (1.4.1). Entretanto, é conhecido que pontos críticos não necessariamente são minimizadores. Para aumentar as chances de se obter um ponto de mínimo deve-se utilizar informações de segunda ordem. Para isso é importante lembrar que d_k precisa ser uma direção de descida para que seja possível garantir que o valor de f será reduzido perto de x_k , ao longo desta direção. Mas como garantir isso?

Teorema 1.4.1. Se $\nabla^2 f(x_k)$ é definida positiva então a direção do Método de Newton d_k será de descida.

Demonstração. Para garantir que uma direção d_k é de descida, é necessário verificar que

$$\nabla f(x_k)^T d_k < 0.$$

Multiplicando a equação (1.4.3) por $\nabla f(x_k)^T$ tem-se que

$$\nabla f(x_k)^T d_k = \nabla f(x_k)^T (-\nabla^2 f(x_k)^{-1} \nabla f(x_k)) = -\nabla f(x_k)^T \nabla^2 f(x_k)^{-1} \nabla f(x_k).$$

Como $A > 0 \Leftrightarrow A^{-1} > 0$ (já que os autovalores de A^{-1} são os inversos dos de A), então o fato de $\nabla^2 f(x_k) > 0$ garante que $\nabla f(x_k)^T d_k < 0$. ■

Exemplo 1.4.1. Neste exemplo é aplicado o Método de Newton para resolver o problema

$$\min f(x) = x_1^2 + x_1x_2 + 2x_2^2 - 3.$$

Por cálculo direto tem-se que $\nabla f(x) = (2x_1 + x_2, x_1 + 4x_2)^T$ e consequentemente

$$\nabla^2 f(x) = \begin{pmatrix} 2 & 1 \\ 1 & 4 \end{pmatrix}.$$

Como $\nabla^2 f(x)$ é sempre definida positiva, pois seus autovalores são positivos ($a_1 = 3 + \sqrt{2}$; $a_2 = 3 - \sqrt{2}$), então a direção de Newton será uma direção de descida em qualquer ponto. Escolhendo $x_0 = (1, 1)$ obtém-se

$$\begin{aligned} d_0 &= -\nabla^2 f(x_0)^{-1} \nabla f(x_0) \\ &= - \begin{pmatrix} \frac{4}{7} & -\frac{1}{7} \\ -\frac{1}{7} & \frac{2}{7} \end{pmatrix} \begin{pmatrix} 3 \\ 5 \end{pmatrix} = \begin{pmatrix} -1 \\ -1 \end{pmatrix} \end{aligned}$$

e

$$\begin{aligned} x_1 &= x_0 + d_0 \\ &= \begin{pmatrix} 1 \\ 1 \end{pmatrix} + \begin{pmatrix} -1 \\ -1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \end{aligned}$$

Observa-se que $x_1 = x^*$ pois $\nabla f(x_1) = 0$. Assim, o método se encerra e em apenas uma iteração a solução é obtida. Na Figura 1.6 pode ser feita a interpretação geométrica da convergência.

Teorema 1.4.2. Seja $f : \mathbb{R}^n \rightarrow \mathbb{R}$ uma função quadrática e $\nabla^2 f(x)$ definida positiva, então o Método de Newton convergirá em uma única iteração.

Demonstração. Como $f(x)$ é quadrática então pode ser escrita como

$$f(x) = \frac{1}{2}x^T G x + b^T x + c,$$

daí vem que

$$\nabla f(x) = Gx + b$$

e

$$\nabla^2 f(x) = G,$$

com G definida positiva por hipótese.

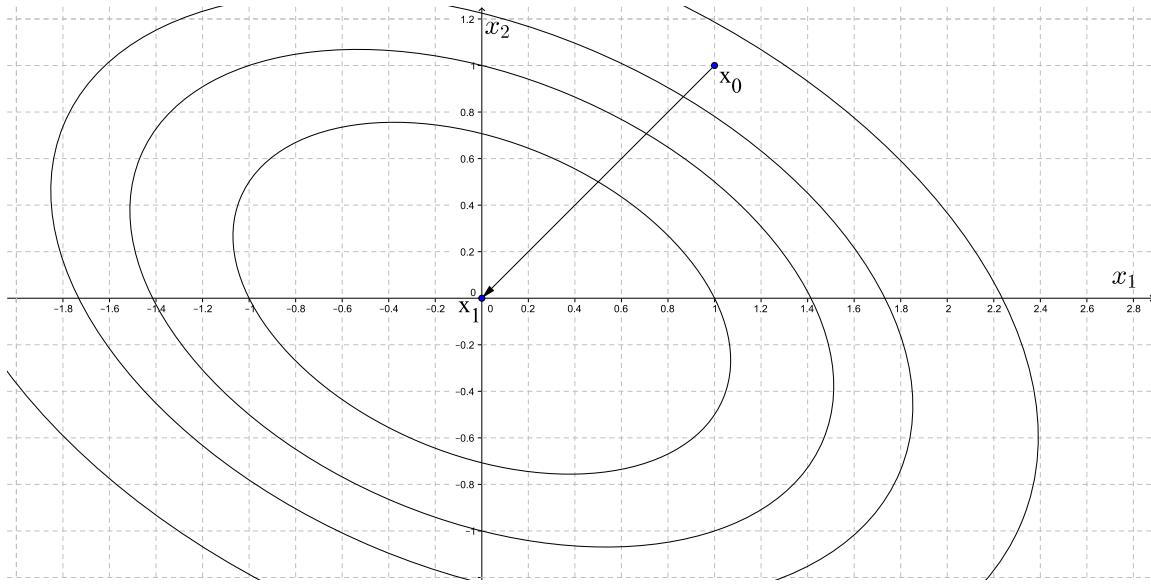


Figura 1.6: Para o Método de Newton aplicado a uma f quadrática tem-se que $x_1 = x^*$.

A iteração se dá por $x_{k+1} = x_k + d_k$. Se $k = 0$ tem-se

$$x_1 = x_0 + d_0 \Rightarrow x_1 = x_0 - G^{-1}(Gx_0 + b) \Rightarrow x_1 = x_0 - x_0 - G^{-1}b \Rightarrow x_1 = -G^{-1}b.$$

De fato $x_1 = x^*$, pois

$$\nabla f(x_1) = Gx_1 + b = G(-G^{-1}b) + b = -Ib + b = -b + b = 0.$$

Como G é definida positiva, as condições suficientes de segunda ordem estão satisfeitas, então $x_1 = x^*$ é ponto de mínimo. ■

No Cálculo Numérico geralmente é visto que o Método de Newton para zero de sistemas não lineares tem convergência local quadrática. Essa informação pode ser encontrada, por exemplo, na Seção 4.2 de [15]. Intuitivamente este resultado quer dizer que, com condições adequadas e partindo próximo da solução, o método dobra o número de casas decimais corretas a cada iteração. O leitor que desejar aprofundar seu conhecimento sobre a teoria de convergência do Método de Newton pode usar como referência o Capítulo 5, em especial a Seção 5.4, de [13]. Desta forma, o Método de Newton para otimização também possui convergência local quadrática, pois é igual

ao Método de Newton para zeros de funções perto de uma solução em que a hessiana é definida positiva.

A fim de aumentar as chances de convergência global (ou seja, a partir de qualquer ponto inicial) do método para funções não quadráticas, inclui-se o passo λ_k . Sendo assim, as iterações serão da forma

$$x_{k+1} = x_k + \lambda_k d_k.$$

O passo λ_k é encontrado de modo a garantir $f(x_{k+1}) < f(x_k)$ e pode ser obtido com uma Busca Linear como a exposta na seção 1.3.

Outra importante ressalva a ser feita é em relação à matriz hessiana associada a $f(x)$. Se $f(x)$ é quadrática e $\nabla^2 f(x)$ é definida positiva, então, como mostrado acima, não há grandes problemas em minimizá-la. Mas quando $f(x)$ não for quadrática, pode ocorrer que $\nabla^2 f(x_k)$ não seja definida positiva. A ideia é então, através de um artifício, usar aproximações quadráticas $q(x)$ com hessianas H_k definida positivas. Para manter o modelo tangente à função a ser aproximada é necessário que

$$q(x) = f(x_k) + \nabla f(x_k)^T(x - x_k) + \frac{1}{2}(x - x_k)^T H_k(x - x_k).$$

Uma condição necessária para que $\nabla^2 f(x_k)$ seja definida positiva é que tenha todos seus autovalores positivos. Por sua vez, os autovalores estão fortemente ligados com a diagonal principal de $\nabla^2 f(x_k)$.

Para tornar H_k definida positiva caso $\nabla^2 f(x_k)$ não o seja (em alguma iteração $k \in \mathbb{N}$), é intuitivo alterar sua diagonal principal.

Considere $m \in \mathbb{R}$ e a matriz

$$H_k = \nabla^2 f(x_k) + mI.$$

Se $v \in \mathbb{R}^n$ é um autovetor de $\nabla^2 f(x_k)$ com um autovalor $a \in \mathbb{R}$ associado a ele, então

$$(\nabla^2 f(x_k) + mI)v = \nabla^2 f(x_k)v + mv = av + mv = (a + m)v.$$

Isto é, somar $m \in \mathbb{R}$ na diagonal principal de $\nabla^2 f(x_k)$ é o mesmo que somar m ao seus autovalores. Deste modo, somando algo maior que o módulo do menor autovalor de $\nabla^2 f(x_k)$ em sua diagonal tem-se que H_k será definida positiva. Computacionalmente, calcular os autovalores de uma matriz é caro. Assim, para detectar se uma matriz é definida positiva se faz a fatoração de Cholesky de H_k , começando com $m = 0$. Se a fatoração falhar, então H_k não é definida positiva e neste caso deve-se aumentar m e tentar novamente refazer a fatoração.

A fim de elucidar como construir modelos quadráticos convexos utilizando a fatoração de Cho-

lesky, considere o exemplo em duas variáveis a seguir.

Exemplo 1.4.2. Minimizar $f(x) = (x_1 - 1)(x_1 - 3)(x_1 - 6) + x_2^2$.

O único minimizador local de f é o ponto $x^* \approx (4.7863; 0)^T$. Ao aplicar o Método de Newton partindo de $x_0 = (2, 11)$, para o qual $f(x_0) = 125$, é necessário verificar se $\nabla^2 f(x_0)$ é definida positiva. Calculando o gradiente e a hessiana de f obtém-se

$$\nabla f(x) = \begin{pmatrix} 3x_1^2 - 20x_1 + 27 \\ 2x_2 \end{pmatrix} \quad \text{e} \quad \nabla^2 f(x) = \begin{pmatrix} 6x_1 - 20 & 0 \\ 0 & 2 \end{pmatrix}.$$

Portanto, o gradiente e a hessiana no ponto inicial são

$$\nabla f(x_0) = \begin{pmatrix} -1 \\ 22 \end{pmatrix} \quad \text{e} \quad \nabla^2 f(x_0) = \begin{pmatrix} -8 & 0 \\ 0 & 2 \end{pmatrix}.$$

Não é possível realizar a fatoração de Cholesky de $\nabla^2 f(x_0)$, ou seja, é impossível obter uma matriz triangular inferior $L \in \mathbb{R}^{2 \times 2}$ tal que $LL^T = \nabla^2 f(x_0)$. Isso acontece porque para determinar L_{11} seria necessário calcular $\sqrt{-8}$, o que não está definido em \mathbb{R} . Somando três à diagonal de $\nabla^2 f(x_0)$ obtém-se

$$H_0 = \nabla^2 f(x_0) + 3I = \begin{pmatrix} -5 & 0 \\ 0 & 5 \end{pmatrix}.$$

Como H_0 ainda não é definida positiva, é necessário continuar a aumentar sua diagonal principal.

Realizando mais dois acréscimos de três na diagonal obtém-se

$$H_0 = \begin{pmatrix} 1 & 0 \\ 0 & 11 \end{pmatrix},$$

a qual é definida positiva e tem fator de Cholesky

$$L = \begin{pmatrix} 1 & 0 \\ 0 & \sqrt{11} \end{pmatrix}.$$

Uma vez obtida $H_0 > 0$, para calcular a direção inicial, é necessário resolver o sistema $H_0 d_0 = -\nabla f(x_0)$. Por já se tratar de um sistema triangular, não é vantajoso utilizar a fatoração obtida para encontrar a solução deste sistema, que é $d_0 = (1, -2)$.

Observe que d_0 realmente é uma direção de descida já que $\nabla f(x_0)^T d_0 = -45 < 0$. Desta forma é possível realizar uma busca linear nesta direção a fim de obter x_1 tal que $f(x_1) < f(x_0)$. Dando o passo completo nesta direção obtém-se $x_0 + d_0 = (3, 9)^T$ e $f(x_0 + d_0) = 81$. Como $f(x_0 + d_0) < f(x_0)$

o passo é aceito e então $x_1 = (3, 9)^T$.

Calculando a hessiana em x_1 obtém-se

$$\nabla^2 f(x_1) = \begin{pmatrix} -2 & 0 \\ 0 & 2 \end{pmatrix}.$$

Novamente o modelo quadrático de Taylor não é convexo, já que $\nabla^2 f(x_1)$ não é definida positiva. Somando quatro à sua diagonal obtém-se a matriz definida positiva

$$H_1 = \begin{pmatrix} 2 & 0 \\ 0 & 6 \end{pmatrix}.$$

Resolvendo $H_1 d_1 = -\nabla f(x_1)$ a direção obtida é $d_1 = (1.5, -3)^T$. Sendo assim, $x_1 + d_1 = (4.5, 6)$ e $f(x_1 + d_1) = 28.125 < f(x_1)$, logo $x_2 = x_1 + d_1$.

A hessiana em x_2 é

$$\nabla^2 f(x_2) = \begin{pmatrix} 7 & 0 \\ 0 & 2 \end{pmatrix},$$

a qual já é definida positiva. Neste caso o modelo quadrático de Taylor já é convexo e, portanto a direção de Newton é calculada resolvendo o sistema $\nabla^2 f(x_2) d_2 = -\nabla f(x_2)$. A direção obtida é $d_2 = (0.321; -6)$ e $x_2 + d_2 = (4.821; 0)$. Como $f(x_2 + d_2) = -8.203 < f(x_2)$ o passo é aceito.

Realizando mais duas iterações não seriam necessários ajustes nos modelos quadráticos e seriam obtidos x_4 e x_5 , que estão muito próximos da solução. O erro em cada um deles em relação à solução exata seria, respectivamente, da ordem de 10^{-4} e 10^{-8} , ilustrando a convergência quadrática.

O próximo exemplo consiste no problema unidimensional relativo à variável x_1 do exemplo anterior. O intuito aqui é ilustrar geometricamente a importância da convexidade do modelo quadrático.

Exemplo 1.4.3. Considere o problema de minimizar $f(x) = (x - 1)(x - 3)(x - 6)$.

Esta função possui somente mínimo local em $x^* \approx 4.7863$.

Começando com $x_0 = 2$, deve-se aproximar $f(x)$ por uma parábola. Para isso se faz necessário aproximar $f(x)$ por Taylor até o termo quadrático em torno de $x_0 = 2$.

Fazendo isso obtém-se

$$q(x) = f(2) + f'(2)(x - 2) + \frac{f''(2)}{2}(x - 2)^2,$$

ou seja,

$$q(x) = 4 - (x - 2) - 4(x - 2)^2.$$

Esta parábola resultante de Taylor tem concavidade voltada para baixo pois $f''(2) = -8$. Este fato em várias variáveis significa que a hessiana não é definida positiva, justamente o que não é desejável. Por exemplo, alterando o valor da segunda derivada do modelo e fazendo-a igual a 1 tem-se uma nova aproximação com termo quadrático positivo

$$q(x) = 4 - (x - 2) + \frac{1}{2}(x - 2)^2.$$

O próximo passo é achar x_1 de modo que $x_1 = \arg \min q(x)$. Efetuando os cálculos obtém-se que $x_1 = 3$. Novamente aproxima-se $f(x)$ por uma parábola, agora em torno de x_1 , com concavidade voltada para cima e encontra-se o mínimo dela. Para isso é preciso novamente corrigir a segunda derivada do modelo de Taylor. Neste caso a correção é feita de modo que $q''(x) = 2$. Deste modo, o ponto encontrado é $x_2 = 4.5$.

Observe que $f(x_0) > f(x_1) > f(x_2)$ e prosseguindo desta maneira a sequencia de pontos obtida converge para x^* . Nas iterações seguintes não seria necessário fazer correções no modelo quadrático de Taylor.

Outro ponto a ser destacado é que se fosse usado um modelo quadrático com segunda derivada pequena então o passo seria grande e possivelmente seria necessário usar a Busca Linear. Por exemplo, se o modelo quadrático em torno de x_0 fosse

$$q(x) = 4 - (x - 2) + \frac{0.1}{2}(x - 2)^2,$$

o minimizador de $q(x)$ seria $\bar{x} = 12$, para o qual $f(\bar{x}) = 594$. Como $f'(x_0) < 0$ é verdade que, localmente, f decresce à direita de x_0 , mas em \bar{x} temos que $f(\bar{x}) > f(x_0)$. Isso acontece porque o valor do passo é proporcional ao inverso da segunda derivada.

A Figura 1.7 mostra $f(x)$ com suas respectivas paráolas de aproximação.

Uma observação importante é que o Método do Gradiente é recuperado se fosse usada H_k igual à matriz identidade em todas as iterações do Método de Newton. Se a diagonal da matriz hessiana for excessivamente aumentada no processo de correção da curvatura do modelo quadrático, a matriz H_k ficaria próxima de um múltiplo da identidade. Desta forma, a direção obtida estaria próxima de ser paralela à direção do gradiente. Nestes casos o Método de Newton não seria tão eficiente.

Por fim, pode-se concluir que o Método de Newton é a priori muito bom, uma vez que possui convergência local quadrática. Porém, a cada iteração é preciso calcular $\nabla^2 f(x_k)$ e resolver

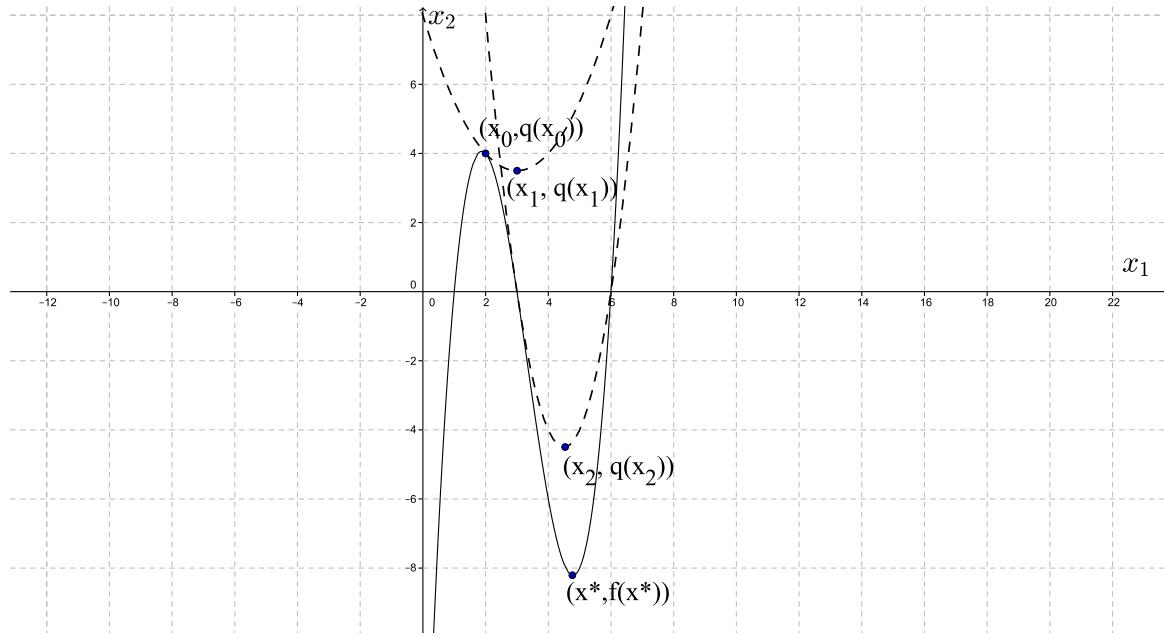


Figura 1.7: Parábolas de aproximação em um método do tipo Newton em tracejado.

$\nabla^2 f(x_k) d_k = -\nabla f(x_k)$. Computacionalmente isto pode ser caro, principalmente se a dimensão de f for muito grande. Além do mais, se $\nabla^2 f(x_k)$ não é definida positiva é preciso substituí-la por uma matriz que seja, o que também envolve a fatoração de matrizes $n \times n$. Esta fatoração pode ser usada na resolução do sistema anterior.

Conforme ressaltado anteriormente, foram tomadas algumas medidas para aumentar as chances de convergência dos métodos a partir de qualquer ponto inicial. Mais formalmente, é dito que um método tem convergência global quando, para qualquer ponto inicial, todo ponto limite da sequência gerada pelo método é um ponto crítico do problema original de minimizar f em \mathbb{R}^n . Pode ser visto no teorema 6.1.6 em [13] que qualquer método que satisfaça as condições a seguir possui convergência global garantida:

- Escolher uma direção d_k suficientemente de descida. Ou seja, as direções devem estar uniformemente afastadas de 90° com o gradiente. Para um parâmetro $\gamma < 0$, então

$$\frac{\nabla f(x_k)^T d_k}{\|\nabla f(x_k)\| \|d_k\|} < \gamma.$$

- O tamanho da direção não pode ser pequeno quando se está longe da solução. Para um

parâmetro $\beta > 0$, $\|d_k\| \geq \beta \|\nabla f(x_k)\|$.

- O passo deve acarretar um decréscimo suficiente na função. Uma das alternativas mais importantes neste sentido é a Busca de Armijo, que exige que o decréscimo seja pelo menos uma fração do previsto pela reta tangente. Para parâmetros $\alpha \in (0, 1)$ e $\sigma > 1$, escolhe-se $\lambda_k \in \left\{1, \frac{1}{\sigma}, \frac{1}{\sigma^2}, \dots\right\}$ o maior possível tal que $f(x_k + \lambda_k d_k) < f(x_k) + \lambda \alpha \nabla f(x_k)^T d_k$.

Capítulo 2

Otimização não linear com restrições de igualdade

Neste capítulo serão abordados problemas de otimização não linear com restrições lineares e não lineares. Em ambos os casos o conjunto factível onde se encontra o minimizador não é mais necessariamente \mathbb{R}^n .

2.1 Restrições lineares

Nesta seção o objetivo é estudar o problema

$$\begin{aligned} & \min f(x) \\ & \text{sujeito a} \\ & Ax = b, \end{aligned} \tag{2.1.1}$$

onde $A \in \mathbb{R}^{m \times n}$, $1 \leq m < n$ e posto de A é m .

O conjunto de factibilidade $R \equiv \{x \in \mathbb{R}^n \mid Ax = b\}$ é uma reta se $m = n - 1$ e um plano se $m = n - 2$.

Se x^* é o minimizador de (2.1.1), então obviamente deverá satisfazer $Ax^* = b$. Isto é, x^* está sobre o conjunto das soluções de $Ax = b$. Por sua vez, o conjunto determinado por $Ax = b$ é paralelo ao obtido com as soluções de $Ax = 0$, que é chamado de núcleo de A e denotado por $Nu(A)$.

Por definição, toda linha de A é ortogonal a $Nu(A)$. Como o posto de A é m , então as m linhas de A formam um subespaço de dimensão m , ortogonal a $Nu(A)$, que é dito imagem de A^T e denotado por $Im(A^T)$. A Figura 2.1 ilustra a restrição $Ax = b$ (quando esta é uma reta), o $Nu(A)$ e $Im(A^T)$.

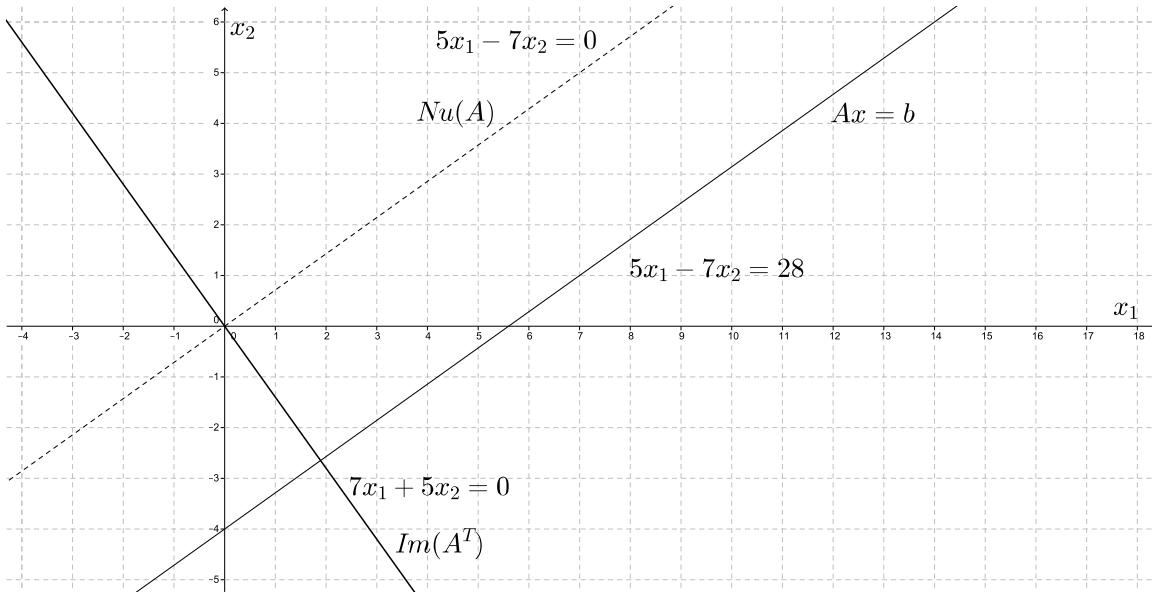


Figura 2.1: Paralelismo entre uma restrição linear e seu núcleo e a ortogonalidade com a imagem de A^T .

Seja $d \in Nu(A)$ uma direção a ser seguida e \bar{x} uma solução factível, então $x = \bar{x} + \alpha d$ também é solução factível, pois $Ax = A(\bar{x} + \alpha d) = A\bar{x} + \alpha Ad = b + 0 = b$. Mais do que isso, se $Ax = b$ e $Ay = b$, então $d = x - y \in Nu(A)$, pois $Ad = A(x - y) = Ax - Ay = b - b = 0$. Observa-se que tomar $d \in Nu(A)$ como direção significa caminhar sobre a restrição $Ax = b$ a partir de uma solução factível conhecida.

Assim, pode-se redefinir

$$R = \{x \in \mathbb{R}^n \mid x = \bar{x} + Z\beta, \quad \beta \in \mathbb{R}^{n-m}\},$$

onde Z é a matriz cujas colunas formam uma base de $Nu(A)$ e \bar{x} é uma solução factível para o problema (2.1.1). Neste caso, $d = Z\beta$.

Exemplo 2.1.1. Encontrar o conjunto das soluções factíveis de um problema que consiste em $\min f$ sujeito à restrição linear

$$\begin{pmatrix} 2 & 5 & 1 \\ -2 & 4 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}.$$

Neste caso,

$$A = \begin{pmatrix} 2 & 5 & 1 \\ -2 & 4 & 2 \end{pmatrix}$$

e $\bar{x} = (0, 0, 1)^T$ é uma solução factível. Achando uma base B para $Nu(A)$

$$\begin{cases} 2x_1 + 5x_2 + x_3 = 0 \\ -2x_1 + 4x_2 + 2x_3 = 0 \end{cases},$$

obtém-se

$$B = x_2(-1, 1, -3)^T \Rightarrow Z = (-1, 1, -3)^T.$$

Assim,

$$x = (0, 0, 1)^T + (-1, 1, -3)^T\beta.$$

Deste modo, para todo $\beta \in \mathbb{R}$ tem-se em x uma solução factível para o problema.

Como toda solução factível é dada por $x = \bar{x} + Z\beta$, então para achar x^* minimizador de f é natural definir $g : \mathbb{R}^{n-m} \rightarrow \mathbb{R}^n$

$$g(\beta) = \bar{x} + Z\beta$$

e ainda $\phi(\beta) : \mathbb{R}^{n-m} \rightarrow \mathbb{R}$

$$\phi(\beta) = f(g(\beta)) = f(\bar{x} + Z\beta).$$

Deste modo, solucionar o problema irrestrito

$$\min_{\beta \in \mathbb{R}^{n-m}} \phi(\beta), \quad (2.1.2)$$

é equivalente a resolver (2.1.1).

Se β^* é o minimizador de $\phi(\beta)$, então $x^* = \bar{x} + Z\beta^*$ é minimizador de f .

Resolvendo (2.1.2), então se deve ter

$$\nabla \phi(\beta^*) = 0.$$

Logo,

$$\begin{aligned} \nabla \phi(\beta^*) &= Z^T \nabla f(g(\beta^*)) \\ &= Z^T \nabla f(\bar{x} + Z\beta^*) \\ &= Z^T \nabla f(x^*) = 0. \end{aligned} \quad (2.1.3)$$

A equação (2.1.3) é uma condição necessária de primeira ordem para que x^* seja minimizador local de f . Esta equação mostra que $\nabla f(x^*)$ é ortogonal ao $Nu(A)$. Logo, $\nabla f(x^*)$ deve estar na $Im(A^T)$, o que no caso de $m = 1$, significa que $\nabla f(x^*)$ é paralelo à linha de A . A Figura 2.2 mostra o caso onde a restrição é $5x_1 - 7x_2 = 28$.

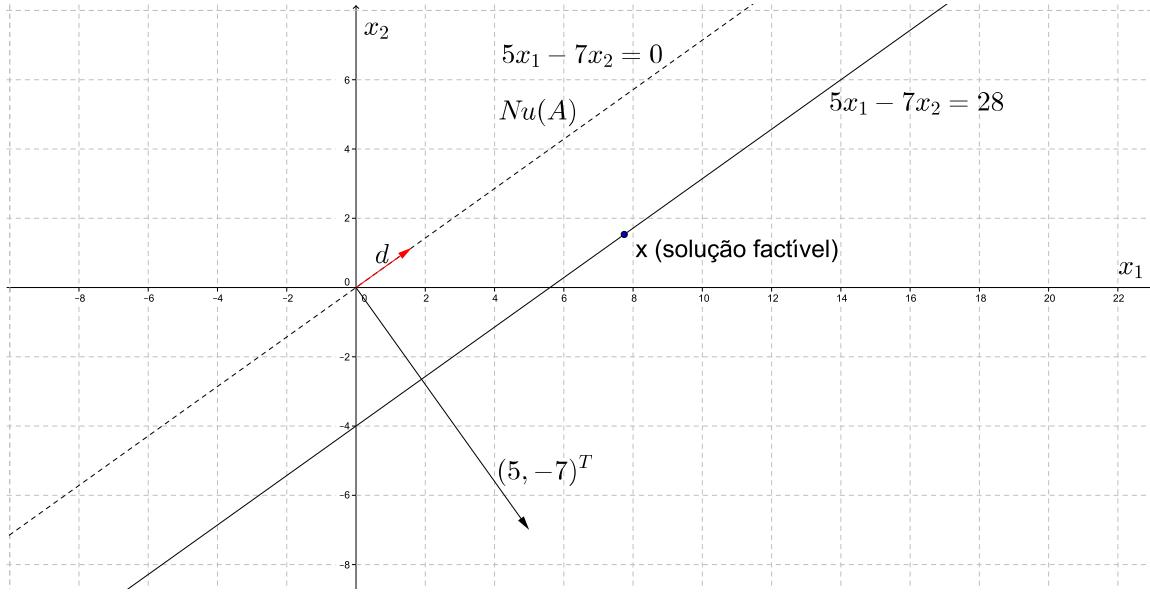


Figura 2.2: O núcleo de A é o conjunto de direções factíveis em R .

Portanto, para o caso geral, existe $\lambda^* \in \mathbb{R}^m$ tal que

$$\nabla f(x^*) = A^T \lambda^*.$$

Resumindo o que foi dito, se x^* é minimizador de f , então existirá o par (x^*, λ^*) que é solução do sistema

$$\begin{cases} \nabla f(x^*) = A^T \lambda^* \\ Ax^* = b \end{cases}.$$

Chama-se de *multiplicador de Lagrange* a cada coordenada do vetor $\lambda^* \in \mathbb{R}^m$.

Porém, resolver este sistema não garante que o ponto x^* encontrado seja um minimizador, mas que seja apenas um ponto crítico do problema de minimizar f sujeito a estas restrições.

A Figura 2.3 permite que se faça a interpretação geométrica do sistema acima. Nesta figura o problema é minimizar $f(x_1, x_2) = x_1^2 + \frac{1}{2}x_1x_2 + x_2^2 - 8x_1 - 4x_2$ sujeito a $\frac{3}{4}x_1 + 2x_2 = 9$.

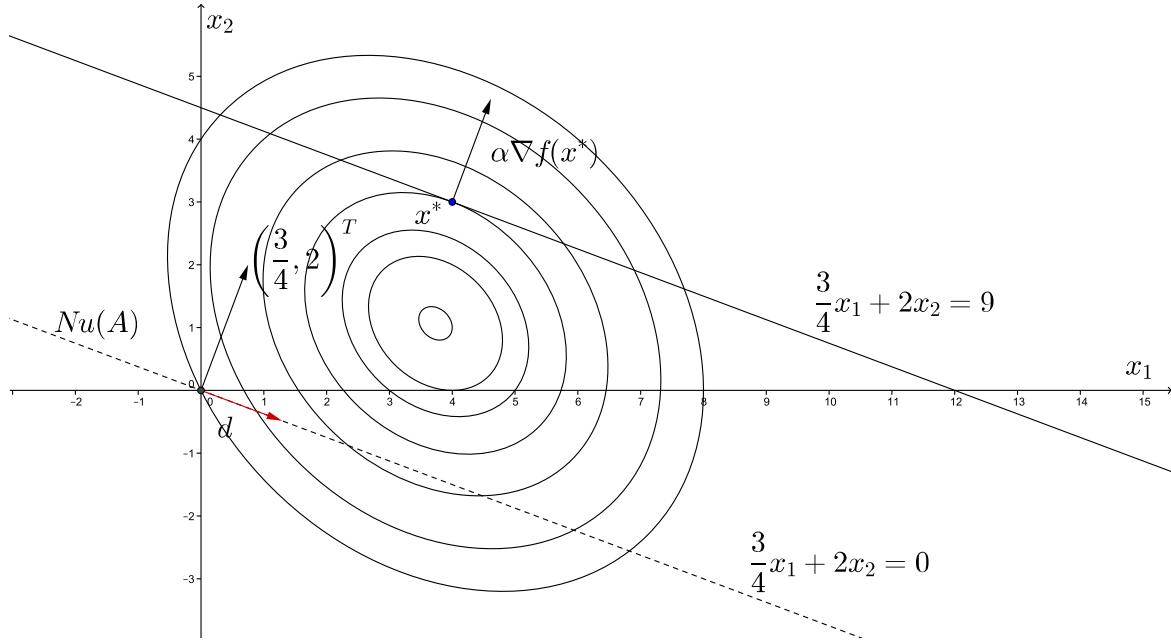


Figura 2.3: Condição de otimalidade de primeira ordem: o vetor gradiente de f em x^* deve estar no espaço gerado pelas linhas de A . No caso de uma única restrição, o vetor $\nabla f(x^*)$ deve ser paralelo à linha de A .

Exemplo 2.1.2. Considere o problema

$$\begin{aligned} & \min x_1^2 + x_2^2 + x_3^2 \\ & \text{sujeito a} \\ & \begin{cases} x_1 + x_2 + x_3 = 1 \\ x_2 - 2x_3 = 2. \end{cases} \end{aligned}$$

As condições necessárias de primeira ordem asseguram que o ponto x^* , solução do problema de optimização, está associado ao vetor $\lambda^* \in \mathbb{R}^2$, de forma que (x^*, λ^*) seja solução do sistema linear

$$\begin{cases} \begin{pmatrix} 2x_1 \\ 2x_2 \\ 2x_3 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 1 & -2 \end{pmatrix} \begin{pmatrix} \lambda_1 \\ \lambda_2 \end{pmatrix} \\ \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & -2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \end{pmatrix} \end{cases}.$$

Resolvendo o sistema vem que $x^* = \left(\frac{1}{2}, 1, -1\right)^T$ e $\lambda^* = (-1, -1)^T$. Como $\nabla^2 f(x)$ é definida positiva, pois

$$\nabla^2 f(x) = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{pmatrix}$$

cujos autovalores são $a_1 = a_2 = a_3 = 2 > 0$, então x^* é o minimizador do problema.

Observe que $\nabla f(x^*) = (1, 2, -1)^T$ não é paralelo a nenhuma linha da matriz A^T , isto é, não é paralelo a $(1, 1, 1)^T$ e nem a $(0, 1, -2)^T$. O que se pode afirmar é que $\nabla f(x^*)$ é uma combinação linear das linhas de A^T . Neste caso,

$$\begin{pmatrix} 1 \\ 2 \\ -1 \end{pmatrix} - 1 \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} - 1 \begin{pmatrix} 0 \\ 1 \\ 2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix},$$

e portanto,

$$\nabla f(x^*) \in \text{Im}(A^T).$$

Exemplo 2.1.3. Considere o problema

$$\begin{aligned} \min & (x_1 - 4)^2 - x_2^2 - 3x_1x_2 \\ \text{sujeito a} & \\ & \begin{pmatrix} 1 & -1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = 0. \end{aligned}$$

Procurando por um ponto estacionário e pelo multiplicador de Lagrange tem-se que

$$\begin{cases} \begin{pmatrix} 2(x_1 - 4) - 3x_2 \\ -2x_2 - 3x_1 \end{pmatrix} = \begin{pmatrix} 1 \\ -1 \end{pmatrix} \lambda \\ \begin{pmatrix} 1 & -1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = 0 \end{cases}.$$

Solucionando o sistema linear, conclui-se que $x_1 = x_2 = -\frac{4}{3}$ e $\lambda = -\frac{20}{3}$.

Neste caso, o sistema resultante das condições de primeira ordem é linear. Isso acontece porque a função objetivo é quadrática e as restrições são lineares. Entretanto, se f não fosse quadrática, o sistema poderia não ser linear.

Porém, é possível mostrar que $x^* = \left(-\frac{4}{3}, -\frac{4}{3}\right)^T$ não é minimizador do problema. A Figura 2.4 mostra geometricamente que o ponto x^* encontrado não é o minimizador.

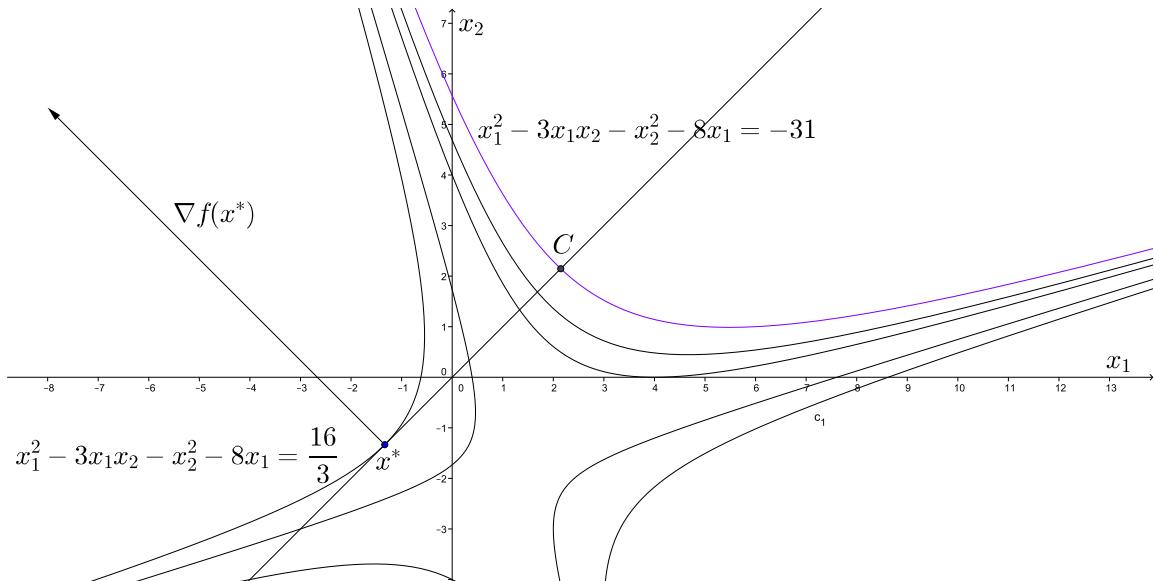


Figura 2.4: Condições de primeira ordem são apenas necessárias: observe que $f(x^*) > f(C)$, logo x^* não pode ser o minimizador de f . De fato, x^* não é nem um minimizador local do problema.

Para garantir que um ponto x^* seja minimizador é preciso que

$$\nabla^2 \phi(\beta^*) > 0 \quad (\text{definida positiva}).$$

Já é sabido que $\nabla\phi(\beta) = Z^T \nabla f(\bar{x} + Z\beta)$. Assim, aplicando novamente a regra da cadeia obtém-se

$$\nabla^2\phi(\beta^*) = Z^T \nabla^2 f(x^*) Z.$$

Então, a condição necessária de segunda ordem implica que

$$\nabla^2\phi(\beta^*) = Z^T \nabla^2 f(x^*) Z \geq 0,$$

e a condição suficiente de segunda ordem diz que

$$\nabla^2\phi(\beta^*) = Z^T \nabla^2 f(x^*) Z > 0.$$

Estas condições significam que $s^T \nabla^2\phi(\beta^*) s \geq 0 \forall s \in \mathbb{R}^{n-m}$ ou $s^T \nabla^2\phi(\beta^*) s > 0 \forall s \in \mathbb{R}^{n-m}$, $s \neq 0$, respectivamente. Portanto $(Zs)^T \nabla^2 f(x^*) (Zs) \geq 0$, logo $d^T \nabla^2 f(x^*) d \geq 0 \forall d \in Nu(A)$.

Por fim, se $x^* \in \mathbb{R}^n$ é solução factível e

- $Z^T \nabla f(x^*) = 0$ ou equivalentemente $\nabla f(x^*) = A^T \lambda^*$,
- $s^T Z^T \nabla^2 f(x^*) Z s > 0 \forall s \in \mathbb{R}^{n-m}$ ou equivalentemente $d^T \nabla^2 f(x^*) d > 0 \forall d \in Nu(A)$,

então x^* é minimizador local de f .

Exemplo 2.1.4. Considere o problema

$$\begin{aligned} \min f(x_1, x_2) &= x_1^2 + 2x_2^2 - x_1 x_2 + x_1 + 2x_2 \\ &\text{sujeito a} \\ &\left(\begin{array}{cc} 1 & \frac{5}{2} \end{array} \right) \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \frac{7}{2}. \end{aligned}$$

Deve-se então resolver o sistema linear

$$\begin{cases} \nabla f(x_1, x_2) = A^T \lambda \\ x_1 + \frac{5}{2}x_2 = \frac{7}{2} \end{cases}$$

a fim de encontrar um possível minimizador. Assim sendo, o sistema será

$$\begin{cases} 2x_1 - x_2 - \lambda = -1 \\ -x_1 + 4x_2 - \frac{5}{2}\lambda = -2 \\ x_1 + \frac{5}{2}x_2 + 0\lambda = \frac{7}{2} \end{cases}.$$

Logo, $x_1 = x_2 = 1$ e $\lambda = 2$. Deste modo, o candidato a minimizador é $x^* = (1, 1)^T$. Para verificar a condição de primeira ordem no formato $Z^T \nabla f(1, 1) = 0$ é necessário encontrar uma base para o $Nu(A)$. Assim,

$$x_1 + \frac{5}{2}x_2 = 0 \Rightarrow x_1 = -\frac{5}{2}x_2.$$

Portanto,

$$Z = \begin{pmatrix} -\frac{5}{2} \\ 1 \end{pmatrix}.$$

Como era de se esperar, $\nabla f(x^*) \perp Nu(A)$, pois

$$\left(-\frac{5}{2} \quad 1 \right) \begin{pmatrix} 2 \\ 5 \end{pmatrix} = 0.$$

Além disso, caso x^* seja um minimizador de (2.1.1) deve-se ter $Z^T \nabla^2 f(x^*) Z > 0$. Para verificar isso tem-se que

$$\nabla^2 f(1, 1) = \begin{pmatrix} 2 & -1 \\ -1 & 4 \end{pmatrix},$$

de onde conclui-se que é definida positiva pois seus autovalores são $a_1 = a_2 = 3 > 0$. Desta forma $d^T \nabla^2 f(1, 1) d > 0$ para todo d não nulo e consequentemente vale o caso particular em que

$$d^T \nabla^2 f(1, 1) d > 0 \quad \forall 0 \neq d \in Nu(A).$$

Sendo assim, $x^* = (1, 1)^T$ é minimizador de $f(x_1, x_2)$ sujeita à restrição linear dada.

Observe na Figura 2.5 que A^T e $\nabla f(x^*)$ são realmente paralelos, fato já mostrado nesta seção.

Exemplo 2.1.5. Considere o problema

$$\begin{aligned} & \min x_1 x_2 \\ & \text{sujeito a} \\ & x_1 - x_2 = 0. \end{aligned}$$

Tem-se que $x^* = (0, 0)$ é o minimizador deste problema. Analisando a Hessiana da função objetivo $f(x_1, x_2) = x_1 x_2$ em x^* ,

$$\nabla^2 f(x^*) = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix},$$

a qual não é definida positiva, já que seus autovalores são $a_1 = 1$ e $a_2 = -1$.

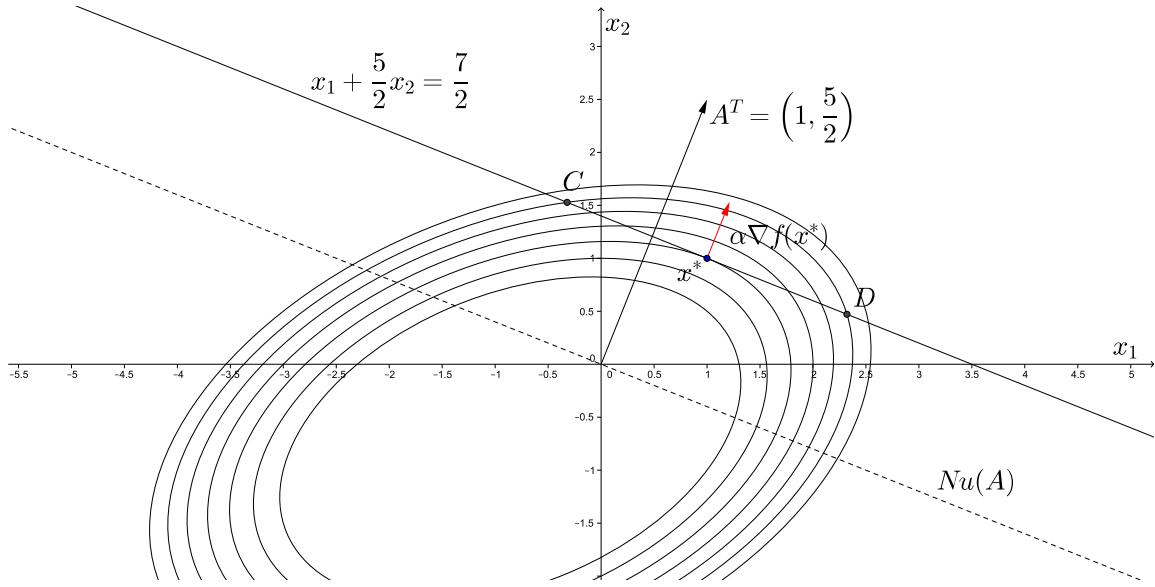


Figura 2.5: Representação geométrica do Exemplo 2.1.4. Os pontos C e D são algumas soluções factíveis do problema de minimização.

Porém, uma base do núcleo da matriz das restrições é

$$Z = \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

Logo,

$$Z^T \nabla f(x^*) Z = 2 > 0$$

e portanto as condições suficientes de segunda ordem são satisfeitas.

O próximo teorema mostra como encontrar um minimizador de uma função quadrática sujeito a restrições lineares.

Teorema 2.1.1. Seja a função quadrática $f(x) = \frac{1}{2}x^T Gx + p^T x + c$ com $G \in \mathbb{R}^{n \times n}$ simétrica, $p \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}$. Assuma também que $A \in \mathbb{R}^{m \times n}$, Z é uma base de $Nu(A)$, $Z^T G Z$ definida positiva e \bar{x} é tal que $A\bar{x} = b$. Então,

$$x^* = \bar{x} - Z(Z^T G Z)^{-1} Z^T (G\bar{x} + p) \quad (2.1.4)$$

é o minimizador de f sujeito a $Ax = b$.

Demonstração. Como o conjunto viável está determinado por restrições lineares, então minimizar f é equivalente ao problema irrestrito de minimizar $\phi(\beta) = f(\bar{x} + Z\beta)$. Se β^* é o minimizador de $\phi(\beta)$, então $x^* = \bar{x} + Z\beta$ é o minimizador do problema original. Como $\phi(\beta)$ também é uma quadrática, pelo Método de Newton e o Teorema 1.4.2, tem-se que

$$\beta^* = \beta - \underbrace{(Z^T G Z)^{-1}}_{\nabla^2 \phi(\beta)^{-1}} \overbrace{Z^T \nabla f(\bar{x} + Z\beta)}^{\nabla \phi(\beta)}.$$

Para $\beta = 0$, pois \bar{x} é factível, tem-se que $\beta^* = -(Z^T G Z)^{-1} Z^T \nabla f(\bar{x})$. Portanto,

$$x^* = \bar{x} + Z(-Z^T G Z)^{-1} Z^T \underbrace{\nabla f(\bar{x})}_{G\bar{x}+p} \Rightarrow x^* = \bar{x} - Z(Z^T G Z)^{-1} Z^T (G\bar{x} + p).$$

■

Exemplo 2.1.6. Considere o problema

$$\begin{aligned} & \min x_1^2 + x_2^2 + x_3^2 + 2x_1 + x_2 - x_3 \\ & \text{sujeito a} \\ & \begin{pmatrix} -1 & 4 & 2 \\ 2 & 1 & -1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0 \\ -2 \end{pmatrix}. \end{aligned}$$

Pela técnica dos multiplicadores de Lagrange, primeiramente é preciso resolver o sistema linear

$$\begin{cases} \begin{pmatrix} 2x_1 + 2 \\ 2x_2 + 1 \\ 2x_3 - 1 \end{pmatrix} = \begin{pmatrix} -1 & 0 \\ 4 & 1 \\ 2 & -1 \end{pmatrix} \begin{pmatrix} \lambda_1 \\ \lambda_2 \end{pmatrix} \\ \begin{pmatrix} -1 & 4 & 2 \\ 2 & 1 & -1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 2 \end{pmatrix} \end{cases},$$

que é equivalente ao sistema

$$\begin{cases} 2x_1 + 0x_2 + 0x_3 + \lambda_1 - 2\lambda_2 = -2 \\ 0x_1 + 2x_2 + 0x_3 - 4\lambda_1 - \lambda_2 = -1 \\ 0x_1 + 0x_2 + 0x_3 - 2\lambda_1 + \lambda_2 = 1 \\ -x_1 + 4x_2 + 2x_3 + 0\lambda_1 + 0\lambda_2 = 0 \\ 2x_1 + x_2 - x_3 + 0\lambda_1 + 0\lambda_2 = -2 \end{cases}.$$

Deste modo, $x^* = \left(-\frac{2}{3}, -\frac{1}{3}, \frac{1}{3}\right)^T$ é candidato a minimizador do problema.

A fim de garantir que x^* seja um minimizador, é preciso verificar a condição suficiente de segunda ordem

$$Z^T \nabla^2 f(x^*) Z > 0.$$

Calculando a hessiana de f tem-se que

$$\nabla^2 f(x^*) = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{pmatrix},$$

que é claramente definida positiva. Portanto, x^* é minimizador e $f(x^*) = -\frac{4}{3}$.

Uma outra abordagem equivalente para o mesmo problema seria verificar que ele atende às condições de Teorema 2.1.1. Para isso, observa-se que uma base do $Nu(A)$ é

$$Z = \begin{pmatrix} -2 \\ 1 \\ -3 \end{pmatrix}$$

e uma solução factível é

$$\bar{x} = \begin{pmatrix} -\frac{4}{3} \\ 0 \\ -\frac{2}{3} \end{pmatrix}.$$

O resultado por (2.1.4) é $x^* = \left(-\frac{2}{3}, -\frac{1}{3}, \frac{1}{3}\right)^T$, o mesmo obtido anteriormente.

Como última observação desta seção, note que denotando $h : \mathbb{R}^n \rightarrow \mathbb{R}^m$ como sendo $h(x) = Ax - b$, as restrições dos problema (2.1.1) podem ser escritas como $h(x) = 0$. Neste caso as equações das condições de primeira ordem podem ser reescritas como $\nabla f(x^*) = J_h(x^*)^T \lambda^*$ e $h(x^*) = 0$, onde

$J_h(x)$ denota o jacobiano de $h(x)$. Este fato motiva o estudo de condições para o problema com restrições não lineares, que é estudado na próxima seção.

2.2 Restrições não lineares

Nesta seção será estudado o problema

$$\begin{aligned} & \min f(x) \\ & \text{sujeito a} \\ & h(x) = 0, \end{aligned} \tag{2.2.1}$$

onde $h : \mathbb{R}^n \rightarrow \mathbb{R}^m$ é uma função suave e $m < n$. De modo geral será considerado que a restrição é não linear.

Em \mathbb{R}^2 , se U é a região factível, então

$$U \equiv \{x \in \mathbb{R}^2 \mid h(x) = 0\}.$$

Observe que diferentemente das restrições lineares, neste caso, pode não haver direções factíveis a partir de uma solução factível conhecida, pois em \mathbb{R}^2 , $h(x) = 0$ é geralmente uma curva e os movimentos a partir de um \bar{x} factível devem ser curvilíneos.

Seja $x(t) : (a, b) \rightarrow U$, onde $U \equiv \{x \in \mathbb{R}^n \mid h(x) = 0\}$, uma parametrização de um arco factível, então $h(x(t)) = 0$ para todo $t \in (a, b)$. Derivando $h(x(t))$ em relação a t , obtém-se

$$\nabla h_i(x(t))^T x'(t) = 0 \quad \forall t \in (a, b)$$

e $1 \leq i \leq m$.

Em particular, para $x(\bar{t}) = \bar{x}$, obtém-se

$$\nabla h_i(\bar{x})^T x'(\bar{t}) = 0, \quad 1 \leq i \leq m.$$

Isto significa que os vetores gradientes das restrições em \bar{x} devem ser ortogonais ao vetor tangente de qualquer arco factível que passe por \bar{x} . Entretanto, nem sempre é verdade que se v é ortogonal a todos os gradientes das restrições em \bar{x} então v é tangente a alguma curva viável que passa por \bar{x} .

Se $\{\nabla h_1(\bar{x}), \dots, \nabla h_m(\bar{x})\}$ é linearmente independente e se $h(\bar{x}) = 0$, então é dito que \bar{x} é um ponto regular às restrições.

No Teorema 2.2.4 de [13] é demonstrado que se \bar{x} é um ponto regular da superfície U , então o respectivo plano tangente T é dado por

$$T = \{y \in \mathbb{R}^n \mid J_h(\bar{x})y = 0\}.$$

Seja $\phi : \mathbb{R} \rightarrow \mathbb{R}$ dada por $\phi(t) = f(x(t))$, a condição necessária de primeira ordem diz que $\phi'(t) = 0$. Usando a regra da cadeia, tem-se que $0 = \phi'(t) = \nabla f(x)^T x'(t) \forall x(t)$. Se x^* é regular, isto significa que $\nabla f(x^*)$ deve ser ortogonal a qualquer vetor de T , plano tangente às restrições. Desta forma, se x^* é um ponto regular e minimizador local, então $\nabla f(x^*) \in \text{Im}(J_h(x^*)^T)$, ou seja

$$\nabla f(x^*) = \sum_{i=1}^m \lambda_i^* \nabla h_i(x^*), \quad (2.2.2)$$

onde $\lambda^* \in \mathbb{R}^m$ é um vetor de multiplicadores de Lagrange. O leitor pode consultar [7] ou [13] para aprofundamento no tema.

A condição (2.2.2) revela que o vetor gradiente de f é uma combinação linear dos vetores gradientes das restrições. No caso de apenas uma restrição a condição necessária de primeira ordem pode ser escrita como

$$\begin{cases} \nabla f(x^*) = \lambda^* \nabla h(x^*) \\ h(x^*) = 0. \end{cases}$$

Exemplo 2.2.1. O vetor $x^* \approx (-1.773, -0.141)^T$ é o minimizador do problema

$$\begin{aligned} \min f(x) &= x_1 + x_2^2 - 10 \\ \text{sujeito a} \\ x_1^2 + x_2 &= 3. \end{aligned}$$

Sendo assim, x^* verifica

$$\nabla f(x^*) = \lambda \nabla h(x^*).$$

De fato,

$$\nabla f(x^*) = \begin{pmatrix} 1 \\ -0,282 \end{pmatrix} = -0,282 \begin{pmatrix} -3,564 \\ 1 \end{pmatrix} = \lambda \nabla h(x^*),$$

$$\text{onde } \lambda \approx -0,282 \text{ e } \nabla h(x^*) = \begin{pmatrix} -3,564 \\ 1 \end{pmatrix}.$$

Na Figura 2.6 as letras maiúsculas do alfabeto representam soluções factíveis para o problema e em vermelho está a restrição $h(x) = x_1^2 + x_2 - 3 = 0$.

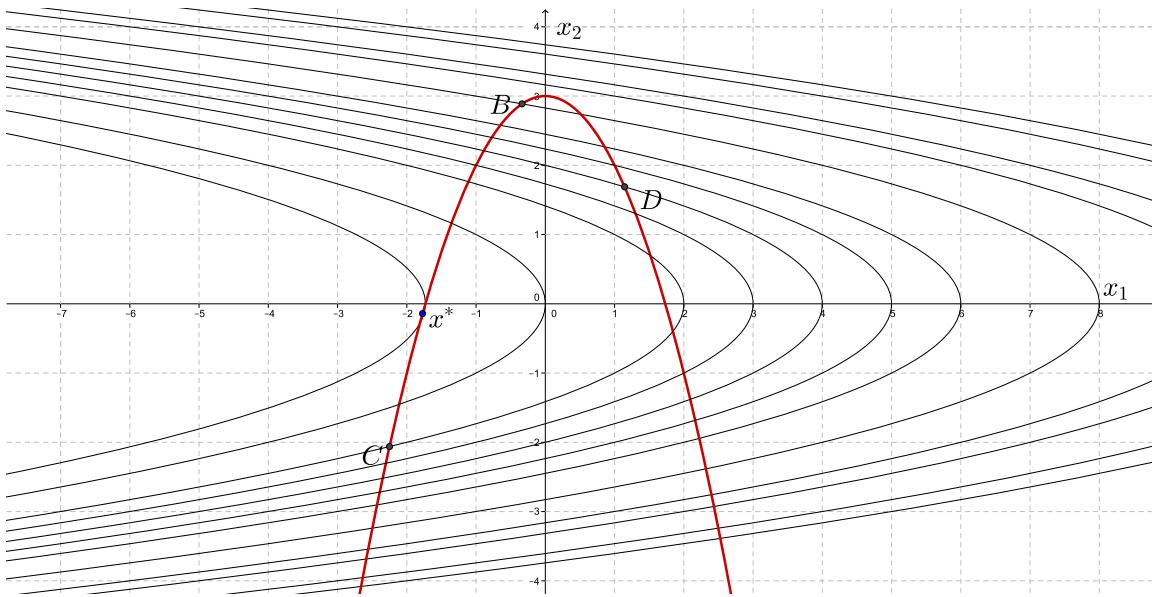


Figura 2.6: Otimização com restrições não lineares: representação geométrica do Exemplo 2.2.1. As letras maiúsculas representam soluções factíveis e x^* é a solução do problema.

No caso de mais de uma restrição, em um ponto regular x^* , as condições necessárias garantem que deve existir $\lambda^* \in \mathbb{R}^m$ de forma que (x^*, λ^*) seja solução de

$$\begin{cases} \nabla f(x) = \lambda_1 \nabla h_1(x) + \lambda_2 \nabla h_2(x) + \dots + \lambda_m \nabla h_m(x) \\ h_1(x) = 0 \\ h_2(x) = 0 \\ \vdots \\ h_m(x) = 0. \end{cases} \quad (2.2.3)$$

Definição 2.2.1. Para problemas de otimização não linear que trazem restrições de igualdade, um ponto $x^* \in \mathbb{R}^n$ é dito *KKT* (Karush-Kuhn-Tucker) se existe $\lambda^* \in \mathbb{R}^m$ tal que (x^*, λ^*) satisfaz (2.2.3).

Observação 2.2.1. Nem todo minimizador é um ponto KKT. Para $f(x) = x$ e $h(x) = x^2$ tem-se que $x^* = 0$ é a solução do problema de minimizar $f(x)$ sujeito a $h(x) = 0$, porém não existe λ tal que $f'(x^*) = \lambda h'(x^*)$. Isto só é possível porque x^* não é regular. Entretanto é possível mostrar que todo minimizador é aproximadamente KKT (AKKT). Ou seja, existe \bar{x} tão perto de x^* quanto se queira, para o qual existe $\bar{\lambda}$ de forma que o sistema (2.2.3) seja satisfeito por $(\bar{x}, \bar{\lambda})$ também com

a precisão $\epsilon > 0$ que se deseja. Vide [14] e [1] para maiores detalhes.

Mais uma vez, resolver o sistema não linear (2.2.3) não garante que a solução x^* seja um minimizador do problema original (2.2.1), mas apenas um ponto crítico. Para obter mais condições para analisar este ponto crítico, deve-se agregar informações de segunda ordem. É possível mostrar que a condição necessária de segunda ordem é que

$$y^T \left(\nabla^2 f(x^*) + \sum_{i=1}^m \lambda_i^* \nabla^2 h_i(x^*) \right) y = y^T \nabla_{xx}^2 L(x^*, \lambda^*) y \geq 0 \quad \forall y \in T,$$

onde

$$L(x, \lambda) = f(x) + \lambda^T h(x), \quad x \in \mathbb{R}^n \text{ e } \lambda \in \mathbb{R}^m$$

é dita *função lagrangiana*. Veja Teorema 2.2.8 de [13], página 19.

Esta condição indica que é necessário olhar a curvatura não só de f mas também das restrições no caso não linear. Por exemplo, se $f(x_1, x_2) = x_2$, então $(0, 0)$ é um minimizador de $f(x_1, x_2)$ restrito a $x_2 = x_1^2$ mas é um maximizador se a restrição for $x_2 = -x_1^2$, mesmo com o núcleo do jacobiano neste ponto sendo o mesmo para as duas restrições. Isto acontece pela curvatura das restrições, conforme ilustrado na Figura 2.7.

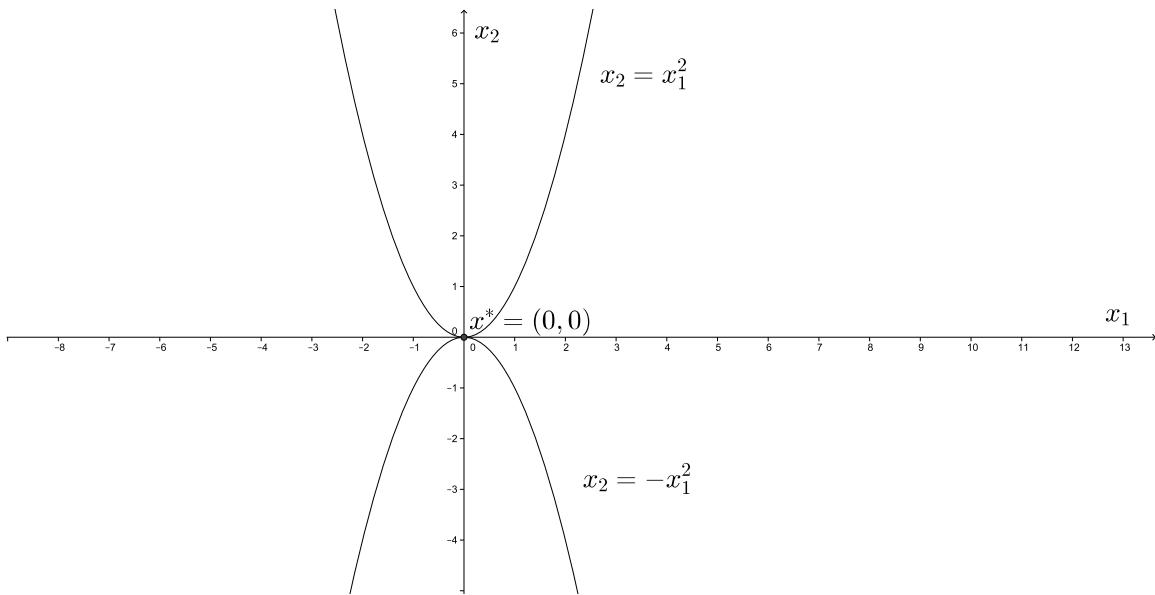


Figura 2.7: A relevância da curvatura das restrições.

Por fim, são apresentadas as condições suficientes de segunda ordem. Se x^* é um ponto regular, $\lambda^* \in \mathbb{R}^m$ um multiplicador de Lagrange e

- $\begin{cases} \nabla f(x^*) = \sum_{i=1}^m \lambda_i^* \nabla h_i(x^*) \\ h_i(x^*) = 0, i = 1, \dots, m \end{cases}$
- $y^T \nabla_x^2 L(x^*, \lambda^*) y > 0 \quad \forall y \in T - \{0\}$

então, x^* é minimizador local estrito de f sobre as m restrições não lineares. Vide Teorema 2.2.9 de [13], página 19.

Observação 2.2.2. Note que as condições suficientes de segunda ordem não obrigam que $\nabla_x^2 L(x^*, \lambda^*)$ seja definida positiva. Isso acontece porque só interessam os arcos factíveis, o que restringe as possibilidades dos vetores tangentes y .

Exemplo 2.2.2. Considere o problema

$$\begin{aligned} \min \quad & 3x_1^2 - x_2^2 \\ \text{sujeito a} \quad & x_1^2 + x_2^2 = 4. \end{aligned}$$

Para encontrar $x^* \in \mathbb{R}^2$ minimizador do problema é necessário resolver o seguinte sistema não linear

$$\begin{cases} \nabla f(x) = \lambda \nabla h(x) \\ h(x) = 0 \end{cases},$$

onde $f(x) = 3x_1^2 - x_2^2$ e $h(x) = x_1^2 + x_2^2 - 4$. Deste modo, tem-se

$$\begin{cases} \begin{pmatrix} 6x_1 \\ -2x_2 \end{pmatrix} = \lambda \begin{pmatrix} 2x_1 \\ 2x_2 \end{pmatrix} \\ x_1^2 + x_2^2 = 4 \end{cases}.$$

As soluções são:

- Para $\lambda = 3$ os pontos são $P_1 = (2, 0)$ e $P_2 = (-2, 0)$.
- Para $x_1 = 0$, os pontos são $P_3 = (0, 2)$ e $P_4 = (0, -2)$ e então $\lambda = -1$.

Como $f(P_1) = f(P_2) > f(P_3) = f(P_4)$, então os candidatos a minimizadores são P_3 e P_4 .

Para garantir que eles sejam minimizadores é necessário verificar a condição suficiente de segunda ordem para cada um dos candidatos. É preciso achar a matriz Z_3 de modo que suas colunas formem uma base para $Nu(J_h(P_3))$ e mostrar que

$$Z_3^T \nabla_x^2 L(P_3, -1) Z_3 > 0.$$

Deste modo,

$$J_h(x) = \begin{pmatrix} 2x_1 & 2x_2 \end{pmatrix} \Rightarrow J_h(P_3) = \begin{pmatrix} 0 & 4 \end{pmatrix}.$$

Uma base para $Nu(J_h(P_3))$ é

$$Z_3 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}.$$

Calculando as derivadas da função lagrangiana tem-se que

$$\nabla_x L(x, \lambda) = \begin{pmatrix} 6x_1 + 2\lambda x_1 \\ -2x_2 + 2\lambda x_2 \end{pmatrix} \Rightarrow \nabla_x^2 L(x, \lambda) = \begin{pmatrix} 6 + 2\lambda & 0 \\ 0 & -2 + 2\lambda \end{pmatrix}.$$

Como $\lambda = -1$, então

$$\nabla_x^2 L(x, -1) = \begin{pmatrix} 4 & 0 \\ 0 & -4 \end{pmatrix},$$

qualquer que seja x , inclusive para $x = P_3$ ou $x = P_4$.

Portanto,

$$Z_3^T \nabla_x^2 L(P_3, -1) Z_3 = 4 > 0.$$

Observe que $Nu(J_h(P_3)) = Nu(J_h(P_4))$ e consequentemente

$$Z_4^T \nabla_x^2 L(P_4, -1) Z_4 = Z_3^T \nabla_x^2 L(P_3, -1) Z_3 = 4 > 0.$$

Por fim, conclui-se que P_3 e P_4 são minimizadores.

A Figura 2.8 mostra que P_3 e P_4 são minimizadores de f sobre a circunferência.

Pelas condições de primeira ordem, encontrar candidatos a minimizadores é equivalente a resolver um sistema não linear. Para isso é possível fazer uso do Método de Newton estudado em Cálculo Numérico. Entretanto, como no Método de Newton para otimização sem restrições, é necessário tomar algumas medidas para garantir que se está de fato minimizando a função e não apenas encontrando pontos estacionários. Esta abordagem dá origem aos métodos do tipo SQP

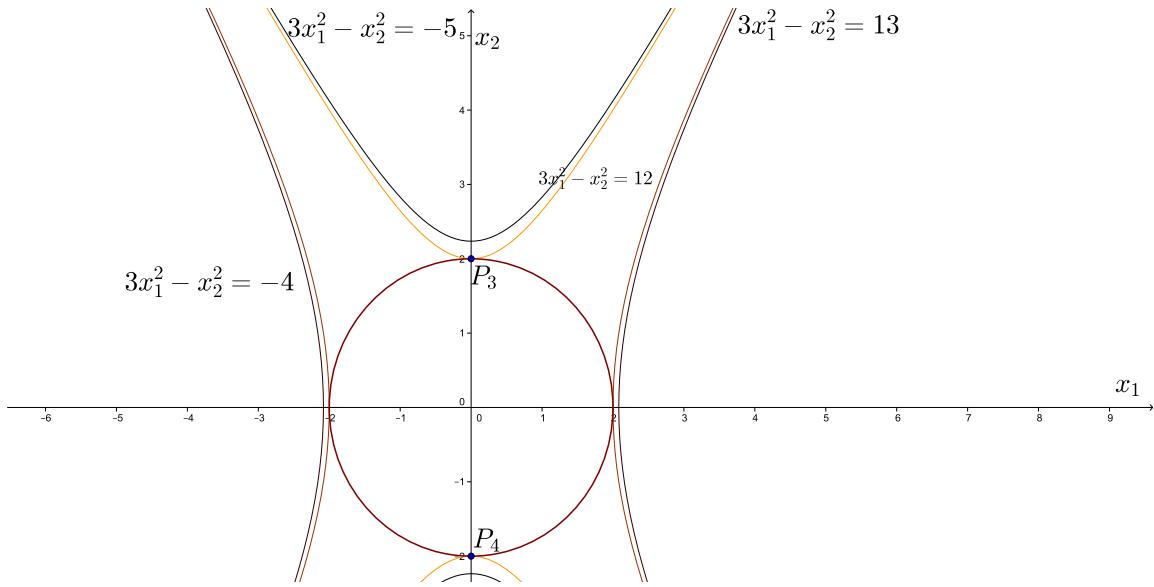


Figura 2.8: Pontos P_3 e P_4 são minimizadores do Exemplo 3.2.2.

(Programação Quadrática Sequencial) que é uma das famílias de métodos mais importantes da otimização com restrições. Neste trabalho será dado enfoque em métodos do tipo penalidade, o que será apresentado no capítulo subsequente. A ideia central destes métodos é resolver o problema com restrições mediante a solução de uma sequência de problemas irrestritos, para os quais é possível usar o método de Newton apresentado no Capítulo 2.

Capítulo 3

Métodos para otimização não linear com restrições

Serão estudados neste capítulo dois importantes métodos para otimização não linear com restrições. Primeiramente o Método da Penalidade Quadrática e posteriormente o Método do Lagrangiano Aumentado.

O problema de interesse neste capítulo será

$$\begin{aligned} & \min f(x) \\ & \text{sujeito a} \\ & h(x) = 0. \end{aligned} \tag{3.0.1}$$

O problema (3.0.1) será estudado sob um ponto de vista distinto do utilizado no Capítulo 2 deste trabalho. Nos métodos computacionais citados a ideia é aproximar a solução deste problema restrito através de outro problema associado, mas irrestrito.

3.1 Método da Penalidade Quadrática

Métodos do tipo penalidade consideram uma medida de inviabilidade do problema original (3.0.1) e a incluem no termo a ser minimizado em um problema aproximado. Isso pode ser feito utilizando um parâmetro de penalidade $\rho \in \mathbb{R}_+$ e $\|\cdot\|$, uma norma qualquer de \mathbb{R}^n , de forma a construir o problema

$$\min_x f(x) + \rho \|h(x)\|. \tag{3.1.1}$$

De modo geral, para garantir que o termo associado à inviabilidade tenha derivadas em todos

os pontos, usa-se a seguinte formulação

$$\min_x f(x) + \frac{\rho}{2} \|h(x)\|^2, \quad (3.1.2)$$

onde $\rho \in \mathbb{R}_+$ é um número possivelmente muito grande e $\|\cdot\|$ é a norma Euclidiana.

Como é possível observar em (3.1.2), a penalidade está associada ao termo $h(x)$. Com isso, paga-se um preço quando se obtém x tal que $h(x) \neq 0$. Desta forma, é esperado que tomando valores cada vez maiores de ρ o termo $h(x)$ seja forçado a ir para zero. Porém, isto só será possível se ρ tender ao infinito.

Para uma notação mais simplificada, em todas as demonstrações e interpretações dos resultados apresentados a seguir, será considerada apenas uma restrição, ou seja, $h : \mathbb{R}^n \rightarrow \mathbb{R}$. Com isso será adotada a função

$$q(\rho, x) = f(x) + \frac{\rho}{2} h(x)^2.$$

Sendo assim, o problema penalizado passa a ser

$$\min_x q(\rho, x). \quad (3.1.3)$$

O problema com mais restrições ($h : \mathbb{R}^n \rightarrow \mathbb{R}^m$) pode ser estudado de maneira muito semelhante e todos os resultados podem ser obtidos com as mesmas ideias aqui apresentadas. Desta forma, a situação geral será considerada nos enunciados dos lemas e teoremas.

Resolver problemas de otimização sem restrições, como (3.1.3), já foi estudado no Capítulo 2. A condição necessária de primeira ordem diz que

$$\nabla_x q(\rho, x) = 0,$$

que implica em

$$\nabla_x q(\rho, x) = \nabla f(x) + \nabla h(x)(\rho h(x)) = 0.$$

Assim, ao resolver (3.1.3), se espera que

$$\nabla f(x) = (-\rho h(x)) \nabla h(x).$$

Parece natural que, quando ρ tende a infinito, tenha-se que $h(x)$ convirja a zero e, consequentemente, $-\rho h(x)$ convirja a λ , onde $\lambda \in \mathbb{R}$ é um multiplicador de Lagrange associado à solução.

De modo geral, o Método da Penalidade Quadrática é um processo iterativo onde, em uma

iteração $k \in \mathbb{N}$, fixa-se ρ_k e encontra-se x_k resolvendo o problema

$$\min_x q(\rho_k, x) = f(x) + \frac{\rho_k}{2} \|h(x)\|^2.$$

À medida que k cresce, o parâmetro ρ_k é atualizado de modo que a sequência $\{\rho_k\}$ seja crescente e ilimitada. Desta forma, novamente considerando $m = 1$, os iterandos x_k satisfazem as equações

$$\begin{cases} \nabla f(x_k) = \nabla h(x_k)(-\rho_k h(x_k)) \\ h(x_k) = \varepsilon_k \end{cases},$$

onde se espera que ε_k tenda a zero quando ρ_k tende a infinito. Ou seja, denotando $\lambda_k = -\rho_k h(x_k)$, ao considerar o par (x_k, λ_k) , o sistema KKT está sendo resolvido aproximadamente e com isso os pontos limites da sequência gerada pelo método serão AKKT [1].

Uma observação importante é que nem sempre seria possível garantir que todos os subproblemas estão bem definidos. Por exemplo, para o problema de minimizar $-2x^2$ sujeito a $x = 0$, utilizando $\rho_k = 2$, o subproblema irrestrito seria minimizar $-x^2$, o qual não tem solução. Uma forma de contornar isso seria acrescentar um conjunto de restrições simples, geralmente uma caixa $l_i \leq x_i \leq u_i$, $l, u \in \mathbb{R}^n$, que não seria penalizado e garantisse que o domínio dos subproblemas fosse um conjunto compacto. Neste caso seria necessário conhecer algum método prático para resolver problemas de otimização com restrições simples. Pela maior simplicidade, neste texto é assumido que x_k está sempre bem definido e é encontrado por algum algoritmo de otimização irrestrita.

O próximo teorema mostra que se $h(x_k) = 0$, numa iteração $k \in \mathbb{N}$ qualquer, então x_k é a solução do problema (3.0.1).

Teorema 3.1.1. Seja $x^* \in \mathbb{R}^n$ o minimizador do problema (3.0.1) e $x_k \in \mathbb{R}^n$ o minimizador de $q(\rho_k, x)$. Se $h(x_k) = 0$, então $x_k = x^*$.

Demonstração. Por simplicidade, é assumido nesta demonstração que $m = 1$. Pelo fato de x_k ser um minimizador de $q(\rho_k, x)$ é válido que

$$f(x_k) + \frac{\rho_k}{2} h(x_k)^2 \leq f(x) + \frac{\rho_k}{2} h(x)^2 \quad \forall x \in \mathbb{R}^n.$$

Assim, se $h(x_k) = 0$, então

$$f(x_k) \leq f(x) + \frac{\rho_k}{2} h(x)^2.$$

Para um $x \in \mathbb{R}^n$ factível tem-se que $h(x) = 0$, logo

$$f(x_k) \leq f(x) \Rightarrow x_k = x^*.$$

■

Se a cada iteração o parâmetro de penalidade está sendo aumentado, é intuitivo pensar que o valor da função q seja maior ou igual ao valor desta função na iteração anterior. O lema seguinte estabelece tal fato.

Lema 3.1.1. Se $x_k \in \mathbb{R}^n$ é minimizador de $q(\rho_k, x)$, $\rho_{k+1} > \rho_k$ e $x_{k+1} \in \mathbb{R}^n$ minimizador de $q(\rho_{k+1}, x)$, então

$$q(\rho_k, x_k) \leq q(\rho_{k+1}, x_{k+1}).$$

Demonstração. Mais uma vez é assumido na demonstração que $m = 1$. Usando o fato que $q(\rho_k, x_k) \leq q(\rho_k, x)$ para todo $x \in \mathbb{R}^n$ e $\rho_{k+1} \geq \rho_k$, tem-se que

$$\begin{aligned} q(\rho_{k+1}, x_{k+1}) &= f(x_{k+1}) + \frac{\rho_{k+1}}{2} h(x_{k+1})^2 \\ &\geq f(x_{k+1}) + \frac{\rho_k}{2} h(x_{k+1})^2 \geq f(x_k) + \frac{\rho_k}{2} h(x_k)^2 \\ &= q(\rho_k, x_k) \end{aligned} \tag{3.1.4}$$

Portanto,

$$q(\rho_{k+1}, x_{k+1}) \geq q(\rho_k, x_k).$$

■

O lema seguinte mostra que o valor de $\|h(x_{k+1})\|^2$ é menor ou igual ao valor de $\|h(x_k)\|^2$. No caso em que $m = 1$ isto significa que a cada iteração o valor de $h(x)$ diminui ou permanece igual ao valor da iteração anterior.

Lema 3.1.2. Se $x_k \in \mathbb{R}^n$ é minimizador de $q(\rho_k, x)$ e $x_{k+1} \in \mathbb{R}^n$ é um minimizador de $q(\rho_{k+1}, x)$, com $\rho_{k+1} > \rho_k$, então

(a) $\|h(x_{k+1})\|^2 \leq \|h(x_k)\|^2$

(b) $f(x_{k+1}) \geq f(x_k)$.

Demonstração. Lembre-se que nas demonstrações dos resultados desta seção está sendo assumido que $m = 1$. Pelo fato de x_k ser um minimizador de $q(\rho_k, x)$ é valido que

$$f(x_{k+1}) + \frac{\rho_k}{2} h(x_{k+1})^2 \geq f(x_k) + \frac{\rho_k}{2} h(x_k)^2 \tag{3.1.5}$$

e, como x_{k+1} é um minimizador de $q(\rho_{k+1}, x)$, tem-se que

$$f(x_k) + \frac{\rho_{k+1}}{2} h(x_k)^2 \geq f(x_{k+1}) + \frac{\rho_{k+1}}{2} h(x_{k+1})^2. \quad (3.1.6)$$

Somando (3.1.5) e (3.1.6) e manipulando algebraicamente obtém-se que

$$\left(\frac{\rho_{k+1}}{2} - \frac{\rho_k}{2} \right) h(x_k)^2 \geq \left(\frac{\rho_{k+1}}{2} - \frac{\rho_k}{2} \right) h(x_{k+1})^2$$

e portanto $h(x_k)^2 \geq h(x_{k+1})^2$.

Desta última relação em (3.1.4), segue que $f(x_{k+1}) \geq f(x_k)$. ■

Pelo teorema anterior tem-se que $\{|h(x_k)|\}$ (ou $\{\|h(x_k)\|\}$ no caso de m restrições) é uma sequencia decrescente, a qual, por ser limitada, é convergente. Se $|h(x_k)|$ convergisse para um número positivo, então implicaria que $\rho_k h(x_k)^2$ e $q(\rho_k, x_k)$ tenderiam ao infinito, o que é absurdo, já que $q(\rho_k, x_k) \leq f(x^*)$ para todo $k \in \mathbb{N}$. Desta forma tem-se que $h(x_k)$ tende a 0 e, pelo Teorema 3.1.1, é natural esperar que x_k tenda a x^* .

Para o caso geral de $h : \mathbb{R}^n \rightarrow \mathbb{R}^m$, um critério de parada pode se dar quando, para alguma iteração $k \in \mathbb{N}$,

$$\|h(x_k)\| \leq \varepsilon,$$

onde ε é um número real positivo tão perto do zero quanto se queira. Neste caso as condições KKT estariam sendo satisfeitas aproximadamente com certa tolerância. Entretanto, observe que a demonstração do fato de $h(x_k)$ convergir a zero foi baseada na existência de um ponto viável para o problema original, no caso a solução do problema x^* . Desta forma, outro critério de parada razoável seria se o parâmetro de penalidade estiver muito grande e mesmo assim a medida de inviabilidade não esteja próxima de zero, indicando que o problema original é provavelmente inviável.

Exemplo 3.1.1. Considere o mesmo problema do Exemplo 2.1.4

$$\begin{aligned} \min \quad & x_1^2 + 2x_2^2 - x_1x_2 + x_1 + 2x_2 \\ \text{sujeto a} \quad & x_1 + \frac{5}{2}x_2 = \frac{7}{2}, \end{aligned}$$

com $f(x) = x_1^2 + 2x_2^2 - x_1x_2 + x_1 + 2x_2$ e $h(x) = x_1 + \frac{5}{2}x_2 - \frac{7}{2}$.

Como este problema será resolvido pelo Método da Penalidade Quadrática é necessário achar um $\tilde{x}_k \in \mathbb{R}^2$ que seja o minimizador de $q(\rho_k, x)$.

Seja $\rho_k = 10^{k+2}$ para todo $k \in \mathbb{N}$. Para $k = 1$,

$$q(1000, x) = f(x) + \frac{1000}{2}h(x)^2 = x_1^2 + 2x_2^2 - x_1x_2 + x_1 + 2x_2 + \frac{1000}{2} \left(x_1 + \frac{5}{2}x_2 - \frac{7}{2} \right)^2.$$

O subproblema consiste em

$$\min q(1000, x).$$

A condição necessária de primeira ordem diz que $\nabla_x q(1000, x) = 0$, deste modo,

$$\nabla_x q(1000, x) = \begin{pmatrix} 2x_1 - x_2 + 1 + 1000 \left(x_1 + \frac{5}{2}x_2 - \frac{7}{2} \right) \\ 4x_2 - x_1 + 2 + 1000 \left(x_1 + \frac{5}{2}x_2 - \frac{7}{2} \right) \frac{5}{2} \end{pmatrix} = 0.$$

Com isso,

$$\nabla_x q(1000, x) = 0 \Leftrightarrow \begin{cases} x_1 = \frac{1653}{1654} \\ x_2 = \frac{1791}{1792} \end{cases}.$$

Observação 3.1.1. Pelo Teorema 1.4.2 o mesmo resultado seria encontrado se fosse aplicado o Método de Newton para otimização sem restrições.

Como

$$\nabla_x^2 q(1000, x) = \begin{pmatrix} 1002 & 2499 \\ 2499 & 6254 \end{pmatrix}$$

é definida positiva, então o par $\tilde{x}_1 = \left(\frac{1653}{1654}, \frac{1791}{1792} \right)^T$ é o minimizador de $q(1000, x)$. Nesta etapa,

$$\tilde{\lambda}_1 = -\rho_1 h(\tilde{x}_1) = -1000 \left(-\frac{38}{19003} \right) = \frac{6333}{3167} \approx 1,99968.$$

Como $h(\tilde{x}_1) \neq 0$ tem-se que \tilde{x}_1 não é o minimizador do problema restrito, isto é, $\tilde{x}_1 \neq x^*$. Agora é preciso atualizar o termo de penalidade. Assim, para $k = 2$ se tem $\rho_2 = 10^4 = 10000$ e o novo subproblema é

$$\min q(10000, x) = x_1^2 + 2x_2^2 - x_1x_2 + x_1 + 2x_2 + \frac{10000}{2} \left(x_1 + \frac{5}{2}x_2 - \frac{7}{2} \right)^2.$$

A condição necessária de primeira ordem diz que $\nabla_x q(10000, x) = 0$, deste modo,

$$\nabla_x q(10000, x) = \begin{pmatrix} 2x_1 - x_2 + 1 + 10000 \left(x_1 + \frac{5}{2}x_2 - \frac{7}{2} \right) \\ 4x_2 - x_1 + 2 + 10000 \left(x_1 + \frac{5}{2}x_2 - \frac{7}{2} \right) \frac{5}{2} \end{pmatrix} = 0.$$

Com isso,

$$\nabla_x q(10000, x) = 0 \Leftrightarrow \begin{cases} x_1 = \frac{16538}{16539} \\ x_2 = \frac{17916}{17917} \end{cases}.$$

Como $\nabla_x^2 q(10000, x)$ é definida positiva para todo x , então o par $\tilde{x}_2 = \left(\frac{16538}{16539}, \frac{17916}{17917}\right)^T$ é o minimizador de $q(10000, x)$. Nesta etapa,

$$\tilde{\lambda}_2 = -\rho_2 h(\tilde{x}_2) = -10000 \left(-\frac{9}{45001}\right) = \frac{44999}{22500} \approx 1,99995.$$

Observando os resultados, é fácil ver que numericamente, quando $\rho \rightarrow \infty$, então

$$\{\tilde{x}_k\} = \{\tilde{x}_1, \tilde{x}_2, \dots\} \rightarrow x^* = (1, 1)^T.$$

Pela convergência do método e como $h(x^*) = 0$, então conclui-se que $x^* = (1, 1)^T$ é o minimizador do problema restrito. Além disso,

$$\{\tilde{\lambda}_k\} \rightarrow \lambda^* = 2.$$

Por isso, o multiplicador de Lagrange associado ao problema restrito é $\lambda^* = 2$.

Para verificar os resultados obtidos é necessário que o par (x^*, λ^*) satisfaça o sistema

$$\begin{cases} \nabla f(x^*) = \lambda^* \nabla h(x^*) \\ h(x^*) = 0 \end{cases}.$$

De fato,

$$\begin{cases} \begin{pmatrix} 2 \\ 5 \end{pmatrix} = 2 \begin{pmatrix} 1 \\ \frac{5}{2} \end{pmatrix} \\ h(1, 1) = 0 \end{cases},$$

e os resultados estão verificados.

Exemplo 3.1.2. Considere o problema

$$\begin{aligned} & \min x_1^2 + 5x_2^2 \\ & \text{sujeito a} \\ & x_1 + x_2^2 = -1. \end{aligned}$$

Ao aplicar o Método da Penalidade Quadrática, o subproblema passa a ser

$$\min q(\rho_k, x) = x_1^2 + 5x_2^2 + \frac{\rho_k}{2} (x_1 + x_2^2 + 1)^2.$$

Fazendo $\rho_k = 10^{k+1}$ para todo $k \in \mathbb{N}$, então $\rho_1 = 100$. Com isso, o primeiro subproblema a ser resolvido é

$$\min q(100, x) = x_1^2 + 5x_2^2 + \frac{100}{2} (x_1 + x_2^2 + 1)^2.$$

A condição necessária de primeira ordem diz que

$$\nabla_x q(100, x) = 0,$$

logo,

$$\nabla_x q(100, x) = \begin{pmatrix} 102x_1 + 100x_2^2 + 100 \\ 210x_2 + 200x_1x_2 + 200x_2^3 \end{pmatrix} = 0.$$

Como o sistema é não linear, então é recomendado aplicar o Método de Newton para obter um ponto estacionário do subproblema. Porém, neste exemplo, quando $x_2 = 0$, então a segunda linha do vetor gradiente e o termo $100x_2$ da primeira se anulam. Com isso o valor de x_1 resulta da equação

$$102x_1 + 100 = 0.$$

Logo,

$$\tilde{x}_1 = \left(-\frac{100}{102}, 0 \right)^T$$

é um candidato a minimizador pois anula o gradiente de $q(100, x)$.

A condição suficiente de segunda ordem determina que \tilde{x}_1 será minimizador se $\nabla_x^2 q(100, \tilde{x}_1)$ for definida positiva. Deste modo,

$$\nabla_x^2 q(100, x) = \begin{pmatrix} 102 & 200x_2 \\ 200x_2 & 210 + 200x_1 + 600x_2^2 \end{pmatrix}.$$

Avaliando a hessiana em \tilde{x}_1 obtém-se

$$\nabla_x^2 q(100, \tilde{x}_1) = \begin{pmatrix} 102 & 0 \\ 0 & \frac{200}{51} \end{pmatrix},$$

que é definida positiva, uma vez que seus autovalores são 102 e $\frac{200}{51}$, ambos positivos.

Assim, $\tilde{x}_1 = \left(-\frac{100}{102}, 0 \right)^T$ é um minimizador local do subproblema irrestrito $\min q(100, x)$.

Nesta etapa,

$$\tilde{\lambda}_1 = -\rho_1 h(\tilde{x}_1) = -100 \left(\frac{1}{51} \right) = -\frac{100}{51}.$$

Observe que na teoria apresentada era necessário encontrar o minimizador global a cada iteração. Entretanto, na prática isso é muitas vezes inviável e portanto na maioria das implementações se contenta em encontrar minimizadores locais.

Atualizando o termo de penalidade, pois $h(\tilde{x}_1) \neq 0$, tem-se que $\rho_2 = 1000$. Agora o novo subproblema é

$$\min q(1000, x) = x_1^2 + 5x_2^2 + \frac{1000}{2} (x_1 + x_2^2 + 1)^2$$

A condição necessária de primeira ordem diz que

$$\nabla_x q(1000, x) = 0,$$

logo,

$$\nabla_x q(1000, x) = \begin{pmatrix} 1002x_1 + 1000x_2^2 + 1000 \\ 2010x_2 + 2000x_1x_2 + 2000x_2^3 \end{pmatrix} = 0.$$

É fácil ver que $\tilde{x}_2 = \left(-\frac{1000}{1002}, 0 \right)^T$ é um candidato a minimizador deste subproblema, pois $\nabla_x q(1000, \tilde{x}_2) = 0$.

Avaliando a Hessiana em \tilde{x}_2 obtém-se

$$\nabla_x^2 q(1000, \tilde{x}_2) = \begin{pmatrix} 1002 & 0 \\ 0 & \frac{7010}{501} \end{pmatrix},$$

que é definida positiva.

Logo, $\tilde{x}_2 = \left(-\frac{1000}{1002}, 0 \right)^T$ é um minimizador de $q(1000, x)$. Nesta etapa,

$$\tilde{\lambda}_2 = -\rho_2 h(\tilde{x}_2) = -1000 \left(\frac{1}{501} \right) = -\frac{1000}{501}.$$

Atualizando o termo de penalidade e resolvendo os respectivos subproblemas, a sequência de pontos obtida será

$$\tilde{x}_3 = \left(-\frac{10000}{10002}, 0 \right); \tilde{x}_4 = \left(-\frac{100000}{100002}, 0 \right); \tilde{x}_5 = \left(-\frac{1000000}{1000002}, 0 \right); \dots$$

que converge claramente para $x^* = (-1, 0)^T$. Como $h(x^*) = 0$ então x^* é o minimizador do

problema restrito. Além disso,

$$\{\tilde{\lambda}_k\} \rightarrow \lambda^* = -2.$$

Por isso, o multiplicador de Lagrange associado ao problema restrito é $\lambda^* = -2$.

De forma geral, para este exemplo, observa-se que os minimizadores dos subproblemas são da forma

$$\tilde{x}_k = \left(-\frac{\rho_k}{\rho_k + 2}, 0 \right).$$

Isto ocorre pois, ao aplicar o Método de Penalidade Quadrática no problema original restrito tem-se o problema irrestrito

$$\min q(\rho_k, x) = x_1^2 + 5x_2^2 + \frac{\rho_k}{2} (x_1 + x_2^2 + 1)^2.$$

A condição necessária de primeira ordem diz que

$$\nabla_x q(\rho_k, x) = 0,$$

logo,

$$\nabla_x q(\rho_k, x) = \begin{pmatrix} 2x_1 + \rho_k(x_1 + x_2^2 + 1) \\ 10x_2 + 2x_2\rho_k(x_1 + x_2^2 + 1) \end{pmatrix} = 0.$$

Quando $x_2 = 0$ a segunda linha do vetor gradiente se anula e o valor de x_1 resulta da equação

$$2x_1 + \rho_k(x_1 + 1) = 0.$$

Daí vem que

$$\begin{aligned} 2x_1 + \rho_k x_1 + \rho_k &= 0 \\ x_1 (\rho_k + 2) + \rho_k &= 0 \\ x_1 &= -\frac{\rho_k}{\rho_k + 2}. \end{aligned}$$

Conclui-se então que

$$\tilde{x}_k = \left(-\frac{\rho_k}{\rho_k + 2}, 0 \right).$$

Na Figura 3.1 tem-se em vermelho a restrição do problema e as letras maiúsculas são pontos referentes a algumas das soluções factíveis.

Na Figura 3.2 é possível visualizar os minimizadores dos problemas irrestritos.

Exemplo 3.1.3. Considere o problema proposto em [10] na página 238 que consiste em

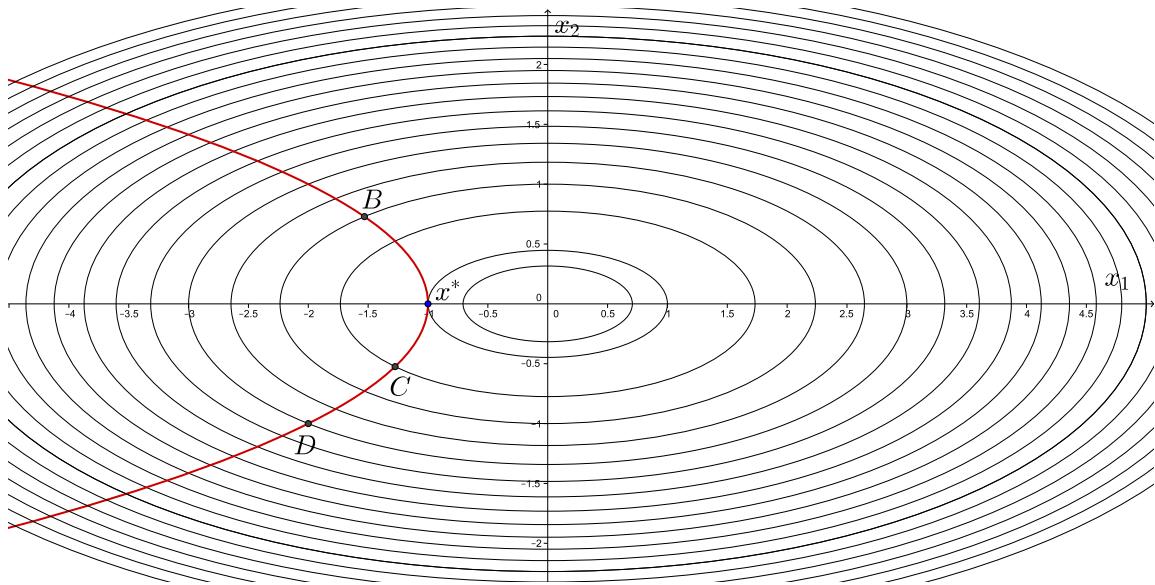


Figura 3.1: Otimização com restrições não lineares: representação geométrica do Exemplo 3.1.2. Os pontos com letras maiúscula indicam pontos viáveis e x^* é solução do problema.

$$\begin{aligned} \min & \quad 2x_1^2 + (x_2 - 1)^2 \\ \text{sujeto a} & \\ & 2x_1 + x_2 = 0. \end{aligned}$$

Aplicando o Método de Penalidade Quadrática, tem-se

$$\min q(\rho_k, x) = 2x_1^2 + (x_2 - 1)^2 + \frac{\rho_k}{2} (2x_1 + x_2)^2.$$

Fazendo $\rho_k = 10^{k+1}$ para todo $k \in \mathbb{N}$, então para a primeira iteração $k = 1$ é necessário solucionar o seguinte subproblema

$$\min_{\rho_1} q(100, x) = 2x_1^2 + (x_2 - 1)^2 + \frac{100}{2} (2x_1 + x_2)^2.$$

A condição necessária de primeira ordem diz que

$$\nabla_x q(100, x) = 0,$$

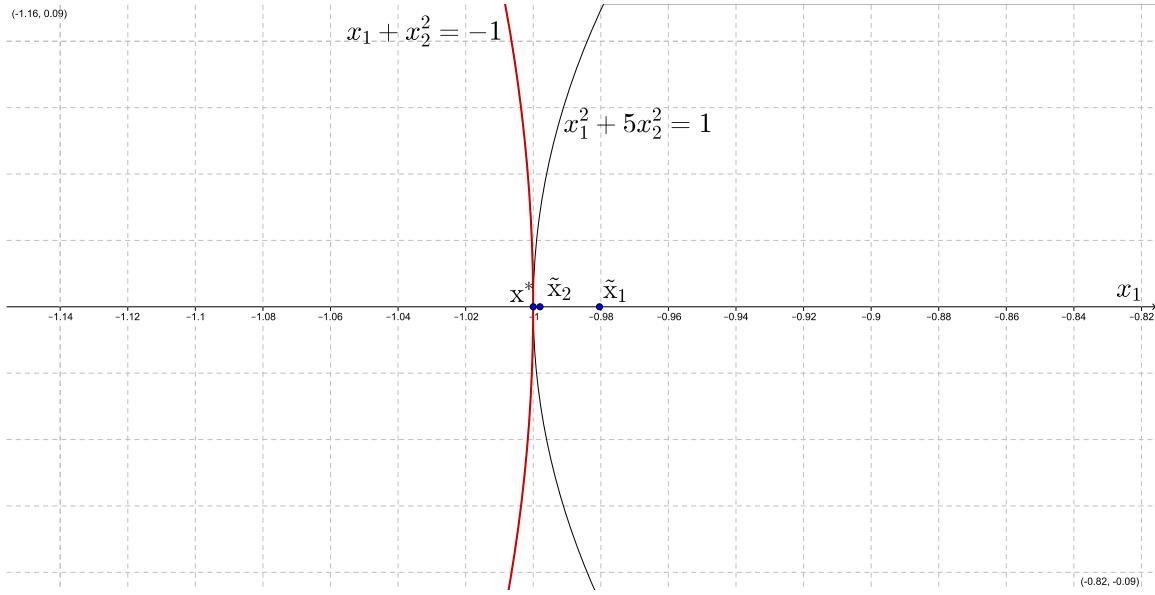


Figura 3.2: Otimização com restrições não lineares: representação geométrica de \tilde{x}_1 e \tilde{x}_2 do Exemplo 3.1.2.

logo,

$$\nabla_x q(100, x) = \begin{pmatrix} 404x_1 + 200x_2 \\ 200x_1 + 102x_2 - 2 \end{pmatrix} = 0.$$

Com isso,

$$\nabla_x q(100, x) = 0 \Leftrightarrow \begin{cases} x_1 = -\frac{50}{151} \\ x_2 = \frac{101}{151} \end{cases}.$$

Como

$$\nabla_x^2 q(100, x) = \begin{pmatrix} 404 & 200 \\ 200 & 102 \end{pmatrix}$$

é definida positiva, então o par $\tilde{x} = \left(-\frac{50}{151}, \frac{101}{151}\right)^T$ é o minimizador de $q(100, x)$. Nesta etapa,

$$\tilde{\lambda}_1 = -\rho_1 h(\tilde{x}_1) = -100 \left(\frac{1}{151}\right) = -\frac{100}{151}.$$

Mas $h(\tilde{x}_1) \neq 0$, então \tilde{x}_1 não é o minimizador do problema restrito. Agora é preciso atualizar o termo de penalidade. Assim, para $k = 2$ se tem $\rho_2 = 10^3 = 1000$ e o novo subproblema a ser

resolvido é

$$\min q(\rho_2, x) = 2x_1^2 + (x_2 - 1)^2 + \frac{1000}{2} (2x_1 + x_2)^2.$$

A condição necessária de primeira ordem diz que $\nabla_x q(1000, x) = 0$, deste modo,

$$\nabla_x q(1000, x) = \begin{pmatrix} 4004x_1 + 2000x_2 \\ 2000x_1 + 1002x_2 - 2 \end{pmatrix} = 0.$$

Com isso,

$$\nabla_x q(1000, x) = 0 \Leftrightarrow \begin{cases} x_1 = -\frac{500}{1501} \\ x_2 = \frac{1001}{1501} \end{cases}.$$

Como $\nabla_x^2 q(1000, x)$ é definida positiva para todo x , então o par $\tilde{x}_2 = \left(-\frac{500}{1501}, \frac{1010}{1501}\right)^T$ é o minimizador de $q(1000, x)$. Nesta etapa,

$$\tilde{\lambda}_2 = -\rho_2 h(\tilde{x}_2) = -1000 \left(-\frac{1}{1501}\right) = -\frac{1000}{1501}.$$

Como $h(\tilde{x}_2) \neq 0$ tem-se \tilde{x}_2 não é o minimizador do problema restrito. Para $k = 3$ se tem $\rho_3 = 10^4 = 10000$ e o novo subproblema a ser resolvido é

$$\min q(\rho_3, x) = 2x_1^2 + (x_2 - 1)^2 + \frac{10000}{2} (2x_1 + x_2)^2.$$

Pela condição necessária de primeira ordem

$$\nabla_x q(10000, x) = \begin{pmatrix} 40004x_1 + 20000x_2 \\ 20000x_1 + 10002x_2 - 2 \end{pmatrix} = 0.$$

Com isso,

$$\nabla_x q(10000, x) = 0 \Leftrightarrow \begin{cases} x_1 = -\frac{5000}{15001} \\ x_2 = \frac{10001}{15001} \end{cases}.$$

Como $\nabla_x^2 q(10000, x)$ é definida positiva para todo x , então o par $\tilde{x}_3 = \left(-\frac{5000}{15001}, \frac{10001}{15001}\right)^T$ é o minimizador de $q(10000, x)$. Nesta etapa,

$$\tilde{\lambda}_3 = -\rho_3 h(\tilde{x}_3) = -10000 \left(-\frac{1}{15001}\right) = -\frac{10000}{15001}.$$

É fácil ver que quando $\rho \rightarrow \infty$, então

$$\{\tilde{x}_k\} = \{\tilde{x}_1, \tilde{x}_2, \dots\} \rightarrow x^* = \left(-\frac{1}{3}, \frac{2}{3}\right)^T.$$

Pela convergência do método e como $h(x^*) = 0$, então conclui-se que $x^* = \left(-\frac{1}{3}, \frac{2}{3}\right)^T$ é o minimizador do problema restrito. Além disso,

$$\{\tilde{\lambda}_k\} \rightarrow \lambda^* = -\frac{2}{3}.$$

Por isso, o multiplicador de Lagrange associado ao problema restrito é $\lambda^* = -\frac{2}{3}$. A Figura 3.3 mostra algumas curvas de níveis da função objetivo, em azul a restrição e em vermelho o ponto ótimo do problema.

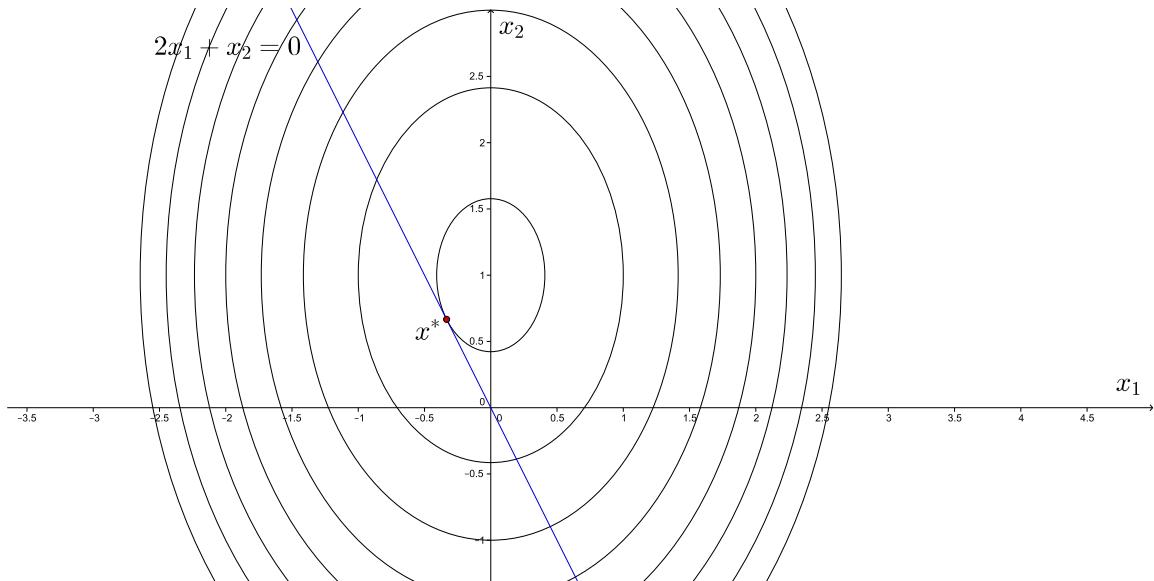


Figura 3.3: Interpretação geométrica do Exemplo 3.1.3, destacando que a curva de nível da função objetivo é tangente à restrição na solução.

O Método de Penalidade Quadrática é simples e intuitivo, além disso, possui propriedades teóricas satisfatórias. Porém, até se chegar em uma boa aproximação de x^* o termo de penalidade pode ser um número muito grande. Isto implica diretamente nos subproblemas que deverão ser

resolvidos. Com o termo de penalidade cada vez maior, a resolução dos subproblemas fica, computacionalmente, cada vez mais difícil. Isso acontece pois o termo referente à função objetivo fica numericamente desprezível em relação ao termo da inviabilidade penalizada. Deste modo, métodos mais sofisticados que limitam o termo de penalidade são mais utilizados, entre eles, o Método do Lagrangiano Aumentado, que será estudado na próxima seção.

3.2 Método do Lagrangiano Aumentado

Nesta seção será considerado novamente o problema

$$\begin{aligned} & \min f(x) \\ & \text{sujeito a} \\ & h(x) = 0. \end{aligned} \tag{3.2.1}$$

Suponha que foi aplicado o Método de Penalidade Quadrática no problema (3.2.1) com um parâmetro de penalidade $\rho_1 \in \mathbb{R}_+$ e que foi encontrado x_1 minimizador do primeiro subproblema conforme Figura 3.4.

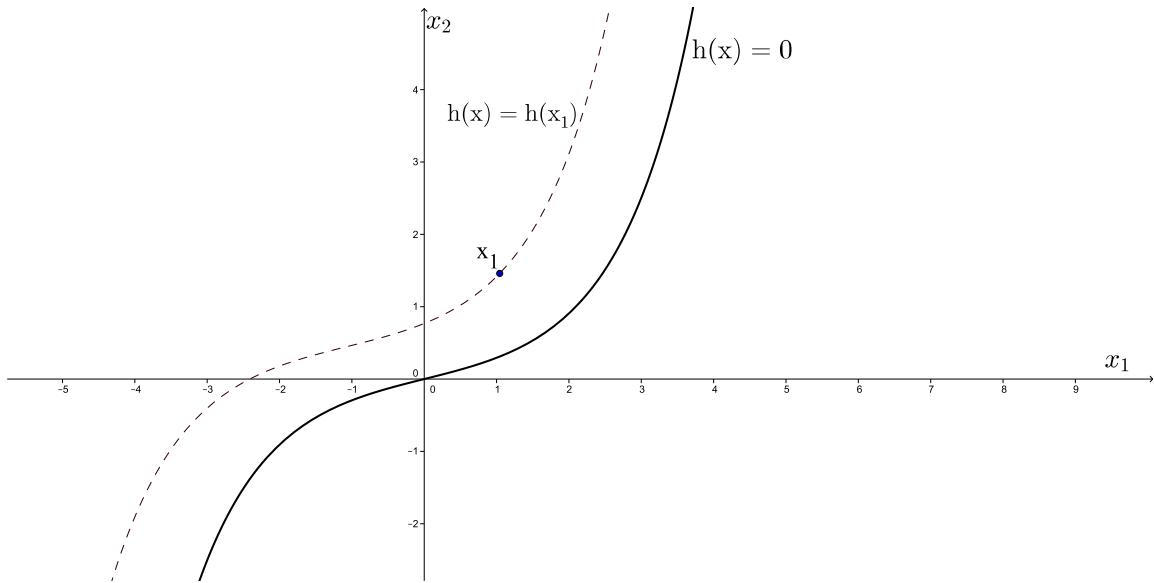


Figura 3.4: O minimizador do problema penalizado x_1 não é uma solução factível.

Como $h(x_1) \neq 0$, então o próximo passo seria atualizar ρ_1 para ρ_2 . Considere agora que

o Método da Penalidade Quadrática fosse aplicado não mais com o parâmetro de penalidade atualizado mas sim com uma restrição deslocada do tipo $h(x) + h(x_1) = 0$, que está representada na Figura 3.5.

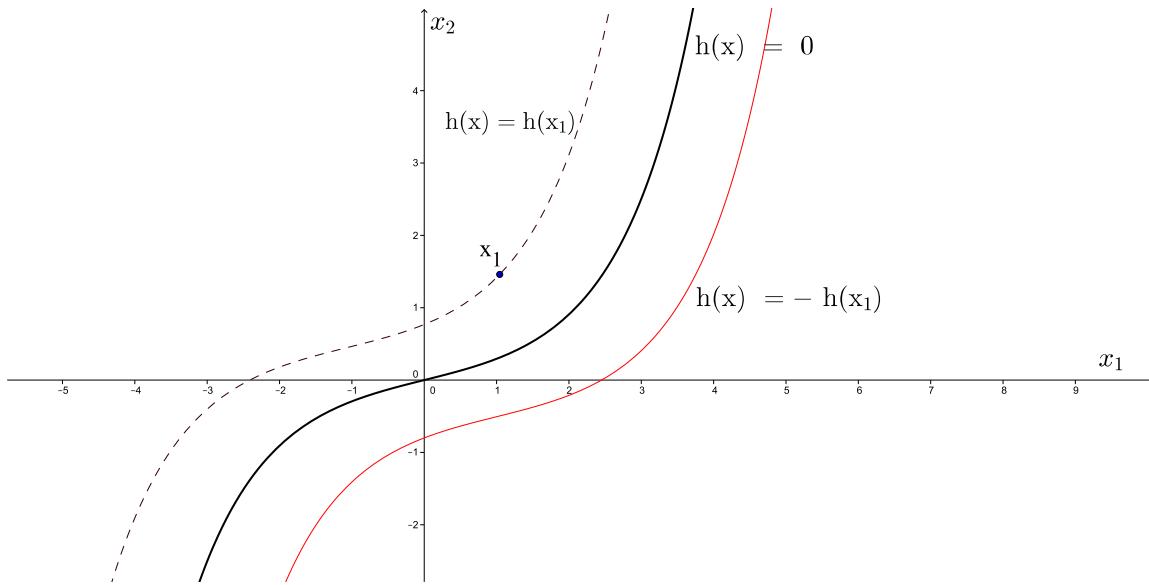


Figura 3.5: Restrição $h(x) = 0$ deslocada.

Se a medida de inviabilidade se mantiver, o minimizador x_2 deste subproblema irrestrito será um ponto factível. Esta é a situação ideal, na prática se espera pelo menos um ponto muito próximo da curva $h(x) = 0$.

O problema solucionado no Método de Penalidade Quadrática para encontrar x_k era

$$\min_x q(\rho_k, x) = f(x) + \frac{\rho_k}{2} \|h(x)\|^2.$$

Assim, para um $\rho_1 \in \mathbb{R}_+$, tinha-se

$$\min_x q(\rho_1, x) = f(x) + \frac{\rho_1}{2} \|h(x)\|^2. \quad (3.2.2)$$

No caso de apenas uma restrição e aplicando o que foi visto até aqui, tem-se numa segunda iteração o problema

$$\min q(\rho_1, x) = f(x) + \frac{\rho_1}{2} (h(x) + h(x_1))^2, \quad (3.2.3)$$

onde $x_1 \in \mathbb{R}^n$ é o minimizador do problema (3.2.2).

Desenvolvendo o produto notável $(h(x) + h(x_1))^2$, tem-se

$$\begin{aligned} q(\rho_1, x) &= f(x) + \frac{\rho_1}{2}(h(x)^2 + 2h(x)h(x_1) + h(x_1)^2) \\ &= f(x) + \rho_1 h(x_1)h(x) + \frac{\rho_1}{2}h(x)^2 + \underbrace{C}_{\frac{\rho_1}{2}h(x_1)^2}. \end{aligned}$$

Voltando ao problema, é necessário resolver então

$$\min_x q(\rho_1, x) = f(x) + \rho_1 h(x_1)h(x) + \frac{\rho_1}{2}h(x)^2,$$

uma vez que o termo C é constante e não interfere na otimização.

Observe ainda que $\rho_1 h(x_1) \in \mathbb{R}$ também é constante. Mais que isso, $\rho_1 h(x_1)$ é o valor usado no Método da Penalidade Quadrática para estimar o multiplicador de Lagrange λ_1 . Desta forma,

$$q(\rho_1, x) = f(x) + \lambda_1 h(x) + \frac{\rho_1}{2}h(x)^2 = L(x, \lambda_1) + \frac{\rho_1}{2}h(x)^2.$$

Da resolução de (3.2.3) encontra-se x_2 minimizador.

Mais uma vez é possível deslocar a penalidade da restrição para tentar corrigir a possível inviabilidade de x_2 . Assim, na próxima iteração o problema consiste em

$$\min f(x) + \frac{\rho_1}{2}(h(x) + h(x_1) + h(x_2))^2.$$

Este problema pode ser escrito como

$$\min f(x) + (\rho_1 h(x_1) + \rho_1 h(x_2))h(x) + \frac{\rho_1}{2}h(x)^2$$

ou

$$\min f(x) + \lambda_2 h(x) + \frac{\rho_1}{2}h(x)^2.$$

Deste modo uma nova estimativa para o multiplicador de Lagrange seria

$$\lambda_2 = \lambda_1 + \rho_1 h(x_2).$$

Analizando o que ocorrerá em cada iteração, é possível concluir que o vetor multiplicador de Lagrange será atualizado, sendo

$$\lambda_{k+1} = \lambda_k + \rho_1 h(x_{k+1}).$$

Observação 3.2.1. Na verdade não significa que o parâmetro de penalidade não precisará mais ser atualizado. Em casos onde o deslocamento não estiver sendo suficiente para se obter pontos viáveis é necessário aumentar ρ_k .

Assim, de modo geral, o Método do Lagrangiano Aumentado se dá por encontrar x_{k+1} resolvendo o subproblema

$$\min_x L(x, \lambda_k) + \frac{\rho_{k+1}}{2} \|h(x)\|^2, \quad (3.2.4)$$

onde $L(x, \lambda) = f(x) + \lambda h(x)$ é a função lagrangiana.

Outra maneira de reescrever o problema ao aplicar o Método do Lagrangiano Aumentado para o caso de várias variáveis é

$$\min_x f(x) + \frac{\rho_{k+1}}{2} \left\| h(x) + \frac{\lambda_k}{\rho_{k+1}} \right\|^2. \quad (3.2.5)$$

Uma das vantagens que este método leva sobre o Método da Penalidade Quadrática é o fato de considerar informação quanto à curvatura das restrições através da função lagrangiana. Outra vantagem, que será vista mais adiante, é de não ter que atualizar a cada iteração o parâmetro de penalidade, possivelmente deixando-o limitado. Pelo que foi estudado até então, este método pode ser entendido como um método de deslocamento de restrições.

Os critérios de parada podem ser os mesmos vistos no Método de Penalidade Quadrática. Para $\varepsilon \in \mathbb{R}_+$ um número tão perto de zero quanto se queria, então o método se encerra em um $x_k \in \mathbb{R}^n$ se $\|h(x_k)\| \leq \varepsilon$. Neste caso o ponto encontrado satisfaz aproximadamente as condições KKT para o problema original. Outro critério de parada seria se o parâmetro de penalidade estiver muito grande e mesmo assim a tolerância em relação à inviabilidade não estiver sendo satisfeita. Neste último caso o problema é declarado como possivelmente inviável.

Considere a seguinte notação para o Lagrangiano Aumentado

$$L(x, \lambda, \rho) \equiv f(x) + \lambda h(x) + \frac{\rho}{2} \|h(x)\|^2.$$

Como $L(x, \lambda, \rho)$ depende de três variáveis, não há risco que se faça confusão com a função lagrangiana, também denotada por L mas que possui apenas duas variáveis. Desta forma será usado o subproblema

$$\min_x L(x, \lambda, \rho)$$

para indicar o uso do Método do Lagrangiano Aumentado. Observe que se $\lambda = 0$, então este problema é equivalente ao subproblema usado quando aplicado o Método de Penalidade Quadrática,

pois

$$\min_x L(x, 0, \rho) \equiv \min_x f(x) + \frac{\rho}{2} \|h(x)\|^2.$$

O Algoritmo 3.2 apresenta os passos para o Método do Lagrangiano Aumentado.

Algoritmo 3.1 Método do Lagrangiano Aumentado

Escolha $\bar{\lambda}_0 \in \mathbb{R}^m$, $\rho_1 > 0$, $\bar{\lambda}_{\min} \leq \bar{\lambda}_{\max}$, $r \in (0, 1)$ e $\gamma > 1$. Tome $k = 0$.

1. Calcule $x_{k+1} \in \mathbb{R}^n$ de modo que x_{k+1} seja solução de (3.2.4) com $\lambda_k = \bar{\lambda}_k$;
2. Tome $\lambda_{k+1} = \bar{\lambda}_k + \rho_{k+1}h(x_{k+1})$ e $\bar{\lambda}_{k+1}$ sendo a projeção de λ_{k+1} em $[\bar{\lambda}_{\min}, \bar{\lambda}_{\max}]$;
3. Se $\|h(x_{k+1})\| \geq r\|h(x_k)\|$, então tome $\rho_{k+2} > \gamma\rho_{k+1}$;
4. Tome $k = k + 1$ e volte para o Passo 1.

Será assumido que todos os subproblemas do Método do Lagrangiano Aumentado estão bem definidos, ou seja, é possível encontrar a solução de cada um deles.

Os teoremas que seguem garantem importantes resultados sobre o Método do Lagrangiano Aumentado.

Teorema 3.2.1. Se $\{x_k\}$ fica em um compacto, então uma das possibilidades abaixo acontecerá

- (i) Se o parâmetro de penalidade é limitado, então x^* é viável.
- (ii) O ponto x^* é estacionário de $\|h(x)\|^2$.

Demonstração. A demonstração é um caso particular do Teorema 4.1 encontrado em [2] na página 1291. ■

O teorema anterior assegura que um outro critério natural de parada para o Algoritmo 3.2 seja que $\rho_k > \rho_{\max}$, onde $\rho_{\max} \in \mathbb{R}_+$ é um número grande, declarando que o problema é provavelmente inviável e o ponto encontrado é estacionário da inviabilidade.

O teorema abaixo garante que o Método do Lagrangiano Aumentado encontra soluções que satisfazem a condição de primeira ordem, tal qual o Método de Penalidade Quadrática.

Teorema 3.2.2. Suponha que a sequência $\{x_k\}$ gerada pelo Método do Lagrangiano Aumentado converge a x^* . Assumindo que x^* seja um ponto viável e regular, então existe λ^* tal que

$$\nabla L(x^*, \lambda^*) = 0.$$

Mais do que isso, se λ^* estiver entre $\bar{\lambda}_{\min}$ e $\bar{\lambda}_{\max}$ então, a sequência $\{\lambda_k\}$ converge a λ^* .

Demonstração. A demonstração é um caso particular do Teorema 4.2 encontrado em [2] na página 1293. ■

Um critério de parada natural para o Algoritmo 3.2, que é motivado pelo Teorema 3.2.2, é que o ponto encontrado seja aproximadamente KKT.

O teorema que segue garante que o parâmetro de penalidade fica limitado e este fato melhora numericamente o desempenho em relação ao Método de Penalidade Quadrática.

Teorema 3.2.3. Suponha que a sequência $\{x_k\}$ gerada pelo Método do Lagrangiano Aumentado converge para x^* , que x^* é factível, regular, e que satisfaz as condições suficientes de segunda ordem associado a λ^* com $\lambda_k^* \in [\bar{\lambda}_{\min}, \bar{\lambda}_{\max}]$, então ρ_k fica limitado.

Demonstração. A demonstração é um caso particular do Teorema 5.4 encontrado em [2] nas páginas 1297-1298. ■

A seguir é apresentado um exemplo de como é possível resolver um problema de otimização utilizando o Método do Lagrangiano Aumentado.

Exemplo 3.2.1. Seja o problema proposto em [10] na página 238 dado por

$$\begin{aligned} & \min x_1^2 + 2x_2^2 \\ & \text{sujeito a} \\ & 3x_1 + x_2 = 3. \end{aligned}$$

Aplicando o Método do Lagrangiano Aumentado, tem-se que

$$\min_x L(x, \lambda, \rho) = x_1^2 + 2x_2^2 + \lambda(3x_1 + x_2 - 3) + \frac{\rho}{2}(3x_1 + x_2 - 3)^2.$$

Adotando $\lambda_0 = 0$ e $\rho_1 = 100$ tem-se o subproblema irrestrito

$$\min L(x, 0, 100) = x_1^2 + 2x_2^2 + \frac{100}{2}(3x_1 + x_2 - 3)^2.$$

A condição de primeira ordem diz que $\nabla L(x, 0, 100) = 0$, logo

$$\begin{pmatrix} 2x_1 + 100(3x_1 + x_2 - 3)3 \\ 4x_2 + 100(3x_1 + x_2 - 3) \end{pmatrix} = 0 \Leftrightarrow \begin{cases} x_1 = \frac{225}{238} \\ x_2 = \frac{75}{476} \end{cases}.$$

Então, $\tilde{x}_1 = \left(\frac{225}{238}, \frac{75}{476}\right)^T$ é candidato a minimizador de $L(x, 0, 100)$.

A condição suficiente de segunda ordem diz que se

$$\nabla^2 L(\tilde{x}_1, 0, 100) > 0$$

então \tilde{x}_1 será o minimizador. Avaliando a matriz hessiana obtém-se

$$\nabla^2 L(x, 0, 100) = \begin{pmatrix} 902 & 300 \\ 300 & 104 \end{pmatrix},$$

que é definida positiva para todo x . Assim, \tilde{x}_1 é minimizador de $L(x, 0, 100)$.

Atualizando o multiplicador de Lagrange, obtém-se

$$\lambda_1 = \lambda_0 + \rho_1 h(\tilde{x}_1) \Rightarrow \lambda_1 = -\frac{75}{119}.$$

Conservando o parâmetro de penalidade $\rho_2 = \rho_1 = 100$, o novo subproblema é

$$\min L\left(x, -\frac{75}{119}, 100\right) = x_1^2 + 2x_2^2 - \frac{75}{119}(3x_1 + x_2 - 3) + \frac{100}{2}(3x_1 + x_2 - 3)^2.$$

A condição de primeira ordem diz que $\nabla L\left(x, -\frac{75}{119}, 100\right) = 0$, logo

$$\begin{pmatrix} 2x_1 - \frac{225}{119} + 100(3x_1 + x_2 - 3)3 \\ 4x_2 - \frac{75}{119} + 100(3x_1 + x_2 - 3) \end{pmatrix} = 0 \Leftrightarrow \begin{cases} x_1 = \frac{5509}{5815} \approx 0.947377 \\ x_2 = \frac{415}{2629} \approx 0.157855 \end{cases}.$$

Então, $\tilde{x}_2 = \left(\frac{5509}{5815}, \frac{415}{2629}\right)^T$ é minimizador de $L\left(x, -\frac{75}{119}, 100\right)$, pois

$$\nabla^2 L\left(\tilde{x}_2, -\frac{75}{119}, 100\right) = \nabla^2 L(\tilde{x}_1, 0, 100) > 0.$$

Observação 3.2.2. A igualdade entre as matrizes hessianas acima ocorre porque

$$\nabla^2 L(x, \lambda, \rho) = \begin{pmatrix} 9\rho + 2 & 3\rho \\ 3\rho & \rho + 4 \end{pmatrix}$$

depende exclusivamente de ρ . Como o parâmetro de penalidade se mantém inalterado, isto é, $\rho_k = 100$, então a igualdade ocorrerá.

Atualizando o multiplicador de Lagrange, obtém-se

$$\lambda_2 = \lambda_1 + \rho_2 h(\tilde{x}_2) \Rightarrow \lambda_2 = -\frac{869}{1376} \approx -0.631541.$$

Note que $h(\tilde{x}_2) \approx -1,2886 \times 10^{-5}$. Se o erro admitido é da ordem de 10^{-4} , isto é, $\varepsilon = 10^{-4}$, então o método se encerra nesta iteração. Então,

$$\{\tilde{x}_k\} = \{\tilde{x}_1, \tilde{x}_2, \dots\} \rightarrow x^* = \left(\frac{18}{19}, \frac{3}{19}\right)^T \approx (0.947368, 0.157895)^T.$$

Além disso,

$$\{\lambda_k\} = \{\lambda_1, \lambda_2, \dots\} \rightarrow \lambda^* = -\frac{12}{19} \approx -0.631579.$$

De fato, x^* é a solução do problema, pois

$$\begin{cases} \nabla f(x^*) + \lambda^* h(x^*) = 0 \\ h(x^*) = 0 \end{cases}$$

e

$$\nabla^2 L(x^*, \lambda^*) > 0.$$

A Figura 3.6 possibilita a interpretação geométrica do Exemplo 3.2.1.

Exemplo 3.2.2. Considere o problema proposto em [12] na página 432 dado por

$$\begin{aligned} & \min x_1^2 + x_1 x_2 + x_2^2 - 2x_2 \\ & \text{sujeito a} \\ & \quad x_1 + x_2 = 2. \end{aligned}$$

Aplicando o Método do Lagrangiano Aumentado, tem-se o problema irrestrito

$$\min_x L(x, \lambda, \rho) = x_1^2 + x_2 x_2 + x_2^2 - 2x_2 + \lambda(x_1 + x_2 - 2) + \frac{\rho}{2}(x_1 + x_2 - 2)^2.$$

Tomando $\lambda_0 = 0$ e $\rho_1 = 100$, então na primeira iteração é necessário

$$\min L(x, 0, 100).$$

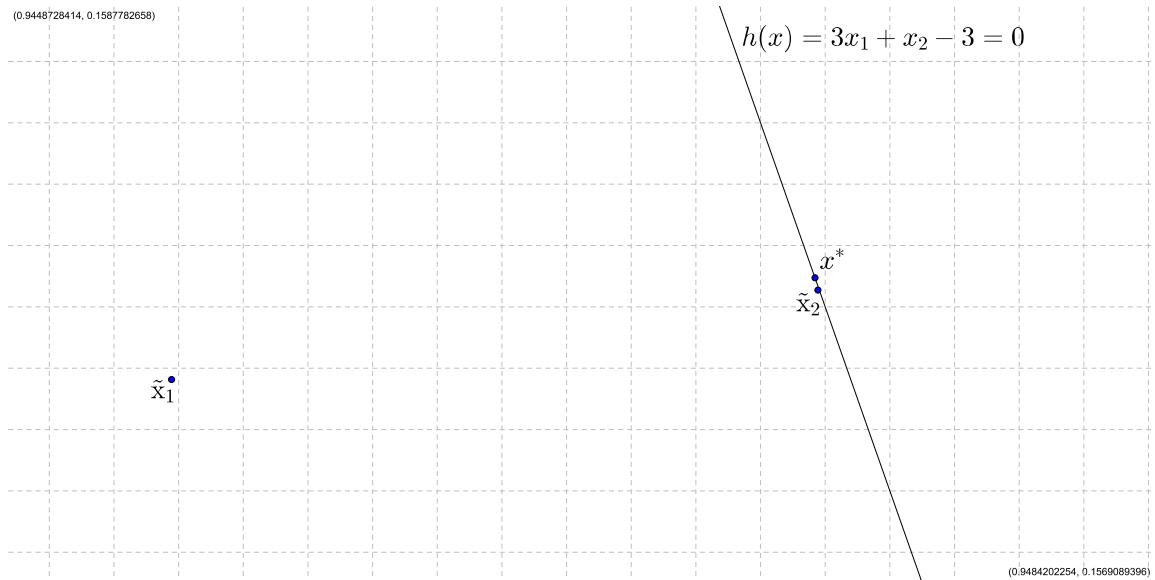


Figura 3.6: Ilustração das iterações do método de penalidade convergindo para a solução à medida que ρ aumenta: representação geométrica do Exemplo 3.2.1.

A condição necessária de primeira ordem diz que $\nabla L(x, 0, 100) = 0$, logo

$$\begin{pmatrix} 2x_1 + x_2 + 100(x_1 + x_2 - 2) \\ x_1 + 2x_2 - 2 + 100(x_1 + x_2 - 2) \end{pmatrix} = 0 \Leftrightarrow \begin{cases} x_1 = -\frac{2}{203} \\ x_2 = \frac{404}{203} \end{cases} .$$

Então, $\tilde{x}_1 = \left(-\frac{2}{203}, \frac{404}{203}\right)$ é o candidato a minimizador de $L(x, 0, 100)$.

A condição suficiente de segunda ordem diz que se

$$\nabla^2 L(\tilde{x}_1, 0, 100) > 0$$

então \tilde{x}_1 será o minimizador do problema irrestrito. Avaliando a matriz hessiana obtém-se

$$\nabla^2 L(x, 0, 100) = \begin{pmatrix} 102 & 101 \\ 101 & 102 \end{pmatrix}$$

que é definida positiva, para todo vetor $x \in \mathbb{R}^n$. Assim, \tilde{x}_1 é minimizador de $L(x, 0, 100)$.

Atualizando o multiplicador de Lagrange, obtém-se

$$\lambda_1 = \lambda_0 + \rho_1 h(\tilde{x}_1) \Rightarrow \lambda_1 = -\frac{400}{203}.$$

Conservando o parâmetro de penalidade, o novo subproblema é

$$\min L\left(x, -\frac{400}{203}, 100\right) = x_1^2 + x_1 x_2 + x_2^2 - 2x_2 - \frac{400}{203}(x_1 + x_2 - 2) + \frac{100}{2}(x_1 + x_2 - 2)^2.$$

A condição de primeira ordem diz que $\nabla L\left(x, -\frac{400}{203}, 100\right) = 0$, logo

$$\begin{pmatrix} 102x_1 + 101x_2 - \frac{6867}{34} \\ 101x_1 + 102x_2 - \frac{6935}{34} \end{pmatrix} = 0 \Leftrightarrow \begin{cases} x_1 = -\frac{1}{6902} \\ x_2 = \frac{13803}{6902} \end{cases}.$$

Então, $\tilde{x}_2 = \left(-\frac{1}{6902}, \frac{13803}{6902}\right)^T$ é minimizador de $L\left(x, -\frac{400}{203}, 100\right)$, pois

$$\nabla^2 L\left(\tilde{x}_2, -\frac{400}{203}, 100\right) = \nabla^2 L(\tilde{x}_1, 0, 100) > 0.$$

Observação 3.2.3. A igualdade entre as matrizes hessianas acima ocorre porque

$$\nabla^2 L(x, \lambda, \rho) = \begin{pmatrix} \rho + 2 & 1 + \rho \\ 1 + \rho & \rho + 2 \end{pmatrix}$$

que depende exclusivamente de ρ . Como o parâmetro de penalidade se mantém inalterado, isto é, $\rho_k = 100$, então a igualdade entre as hessianas ocorrerá.

Atualizando o multiplicador de Lagrange, obtém-se

$$\lambda_2 = \lambda_1 + \rho_2 h(\tilde{x}_2) \Rightarrow \lambda_2 = -\frac{6900}{3451}.$$

Assim, o próximo subproblema a ser resolvido, conservando o parâmetro de penalidade, é

$$\min L\left(x, -\frac{6900}{3451}, 100\right) = x_1^2 + x_1 x_2 + x_2^2 - 2x_2 - \frac{6900}{3451}(x_1 + x_2 - 2) + \frac{100}{2}(x_1 + x_2 - 2)^2.$$

A condição de primeira ordem diz que $\nabla L\left(x, -\frac{6900}{3451}, 100\right) = 0$, logo

$$\begin{pmatrix} 102x_1 + 101x_2 - \frac{346851}{1726} \\ 101x_1 + 102x_2 - \frac{352103}{1726} \end{pmatrix} = 0 \Leftrightarrow \begin{cases} x_1 = -\frac{1}{350378} \\ x_2 = \frac{700755}{350378} \end{cases}.$$

O ponto $\tilde{x}_2 = \left(-\frac{1}{350378}, \frac{700755}{350378}\right)^T$ é minimizador de $L\left(x, -\frac{6900}{3451}, 100\right)$, pois

$$\nabla^2 L\left(x, -\frac{6900}{3451}, 100\right) = \nabla^2 L\left(\tilde{x}_2, -\frac{400}{203}, 100\right) = \nabla^2 L(\tilde{x}_1, 0, 100) > 0.$$

Note que $h(\tilde{x}_3) \approx 0,000011416$. Se $\varepsilon = 10^{-3}$ por exemplo, então o método se encerra.

Observe que

$$\{\tilde{x}_k\} = \{\tilde{x}_1, \tilde{x}_2, \dots\} \rightarrow x^* = (0, 2)^T.$$

Além disso,

$$\{\lambda_k\} = \{\lambda_1, \lambda_2, \dots\} \rightarrow \lambda^* = -2.$$

Neste caso, $f(x^*) = 0$, $h(x^*) = 0$,

$$\nabla f \begin{pmatrix} 0 \\ 2 \end{pmatrix} - 2\nabla h \begin{pmatrix} 0 \\ 2 \end{pmatrix} = 0$$

e

$$\nabla^2 L(x^*, \lambda^*) > 0.$$

Portanto, x^* é solução do problema original.

Capítulo 4

ALGENCAN: instalação e utilização

ALGENCAN é um programa desenvolvido em Fortran com o propósito de resolver problemas de otimização não linear com restrições (desigualdade e igualdade), utilizando para isso o Método do Lagrangiano Aumentado. Os realizadores deste projeto são os professores E. G. Birgin da Universidade de São Paulo (Departamento de Ciência da Computação do Instituto de Matemática e Estatística) e José Mario Martínez da Universidade Estadual de Campinas (Departamento de Matemática Aplicada do Instituto de Matemática, Estatística e Computação Científica).

Este capítulo está dividido em duas seções: instalação e utilização do ALGENCAN. Como o próprio nome diz, a primeira seção se dedicará à instalação do ALGENCAN, desde o *download* do programa até a execução do seu arquivo teste. Na segunda seção, sobre a utilização, serão abordados os principais comandos do ALGENCAN, a fim de que o usuário possa de forma autônoma realizar a otimização do seu próprio problema.

4.1 Instalação do ALGENCAN

Primeiramente é necessário compreender que o ALGENCAN foi desenvolvido a partir do Linux. Embora não seja necessário, na experiência realizada, se nota que é de extrema importância que o usuário tenha este sistema no computador que pretende desenvolver a otimização. Se o usuário preferir é possível o funcionamento do pacote ALGENCAN com a virtualização do Linux. Neste trabalho o Linux foi utilizado através do sistema operacional Ubuntu, versão 14.04.1 LTS.

Antes de executar a sequência de passos abaixo, recomenda-se uma leitura prévia para melhor compreensão do que está sendo feito. Em resumo o que será feito é:

1. instalar o GFortran;
2. baixar o ALGENCAN compactado;

3. copiar o ALGENCAN compactado no diretório **Pasta pessoal**;
4. descompactar o ALGENCAN;
5. gerar a biblioteca do ALGENCAN;
6. criar a pasta *myfolder*;
7. copiar o arquivo **toyprob.f90** na pasta *myfolder*;
8. criar uma variável de ambiente;
9. compilar **toyprob.f90**;
10. executar o arquivo compilado.

Entre no ambiente Linux, abra o Terminal e digite

```
sudo apt-get install gfortran
```

para instalar o software GFortran que será essencial para compilar o teste. Uma tela parecida com a Figura 4.1 deverá aparecer.

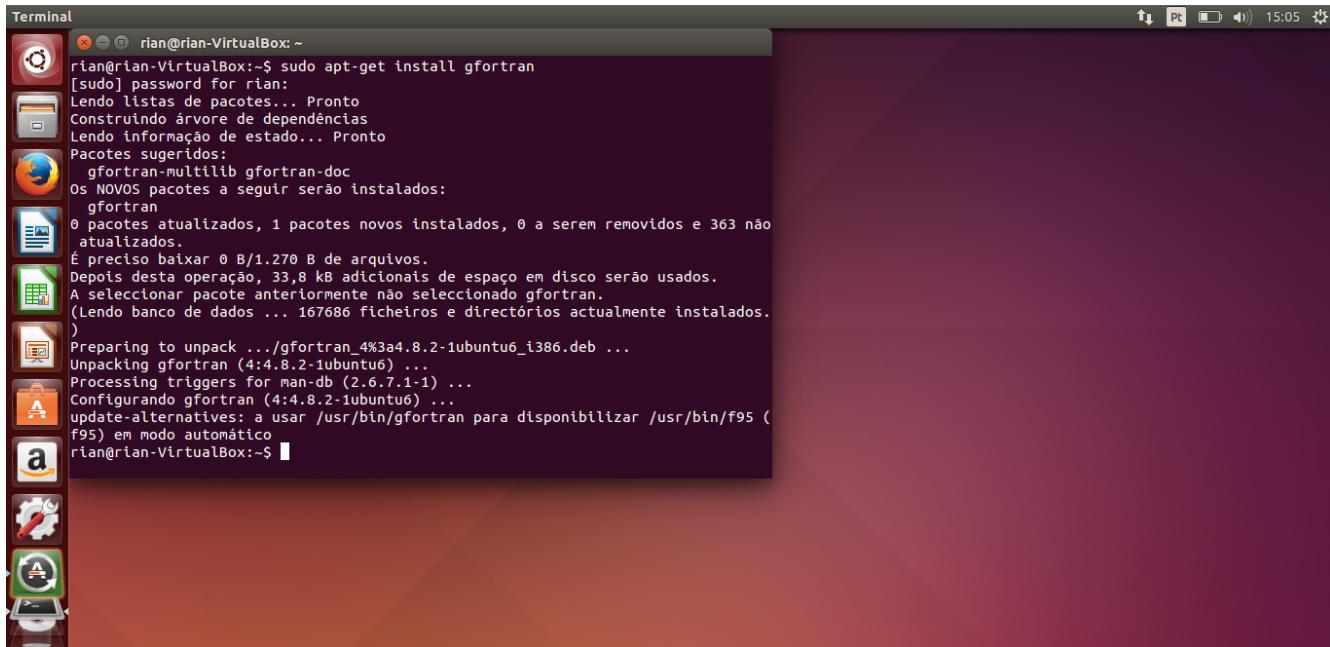


Figura 4.1: Instalação do GFortran.

Faça o download do ALGENCAN em

<http://www.ime.usp.br/~egbirgin/tango/downloads.php>,

conforme mostra Figura 4.2. A versão de ALGENCAN utilizada neste trabalho é 3.0.0. Se houver alguma atualização, então a versão que será disponibilizada será diferente da adotada neste trabalho. Um possível número para a próxima versão de ALGENCAN é 3.0.1. Neste caso, provavelmente as instruções aqui descritas continuarão válidas trocando 3.0.0 por 3.0.1.

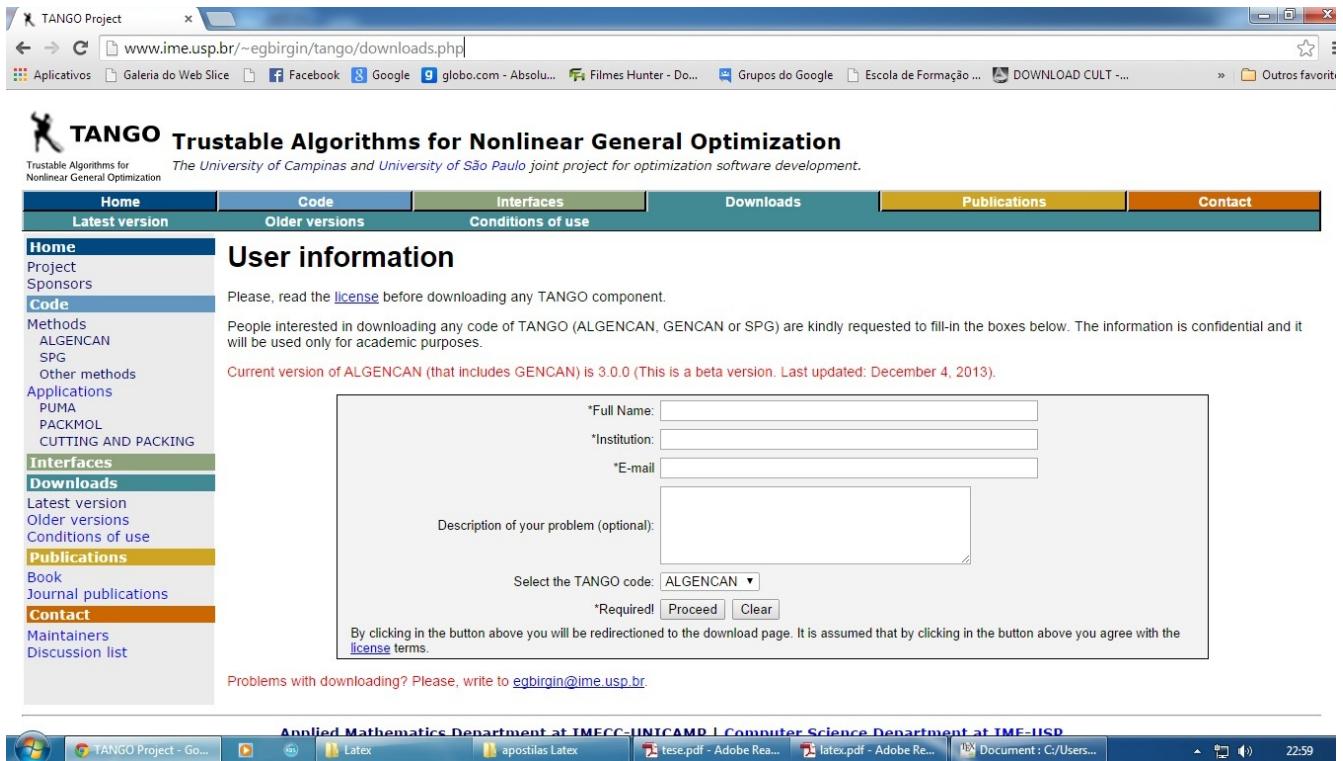


Figura 4.2: Página de download do pacote ALGENCAN.

Em via de regra, o arquivo baixado se encontra no diretório **Downloads**, repare que este está compactado. Digite no Terminal

```
cd Downloads
```

para ir para o diretório onde se encontra o ALGENCAN. Na sequência digite

```
ls -la
```

para se certificar que o ALGENCAN está lá. Na saída deste comando deverá aparecer o arquivo ‘algencan-3.0.0-beta.tgz’, conforme Figura 4.3.

Ainda em **Downloads** digite

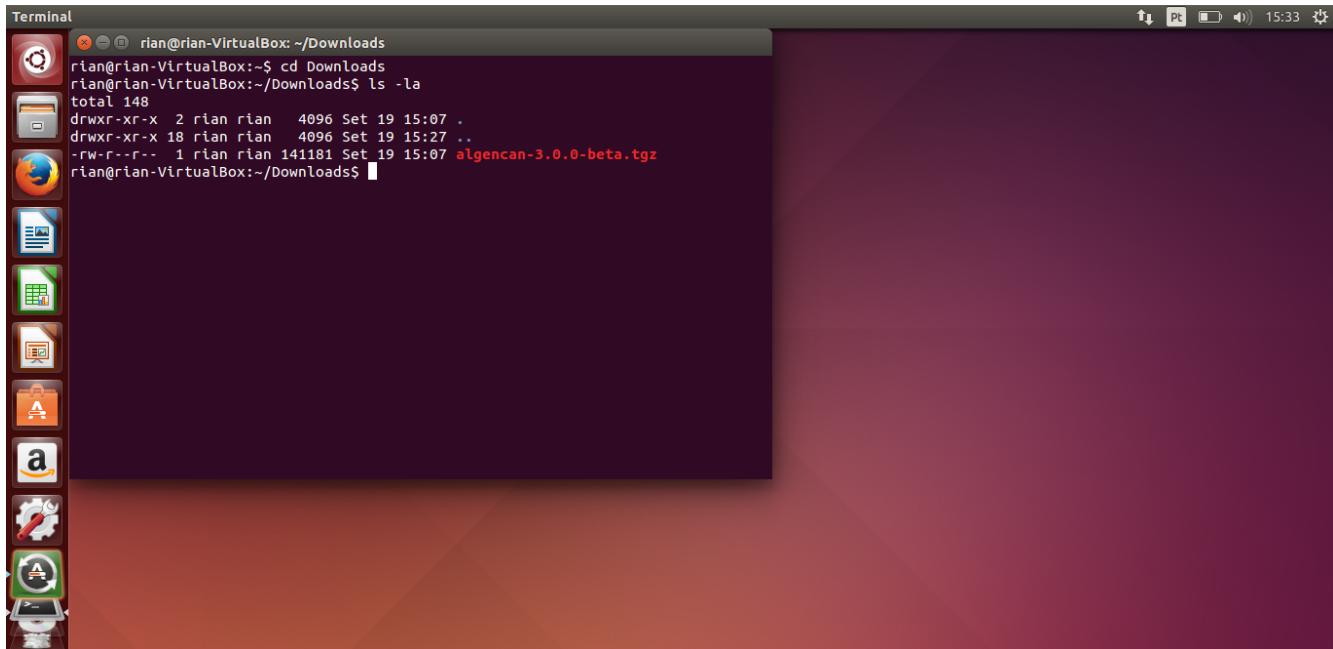


Figura 4.3: Averiguação se o pacote ALGENCAN se encontra em **Downloads**.

```
cp algencan-3.0.0-beta.tgz ../../
```

para copiar o ALGENCAN na **Pasta pessoal**, conforme Figura 4.4.

Depois digite

```
cd ..
```

para sair do diretório **Downloads**. Digite

```
tar -zxvf algencan-3.0.0-beta.tgz
```

para descompactar o ALGENCAN na pasta pessoal. Na saída deste comando deverá aparecer uma lista com vários arquivos. Toda linha desta lista deve se iniciar com ‘algencan-3.0.0’, conforme Figura 4.5.

Observe que, ao descompactar, criou-se uma pasta chamada *algencan-3.0.0* no diretório **Pasta pessoal**. Digite agora

```
cd algencan-3.0.0
```

para entrar na pasta recém criada. Para se certificar que está na pasta correta, digite

```
pwd
```

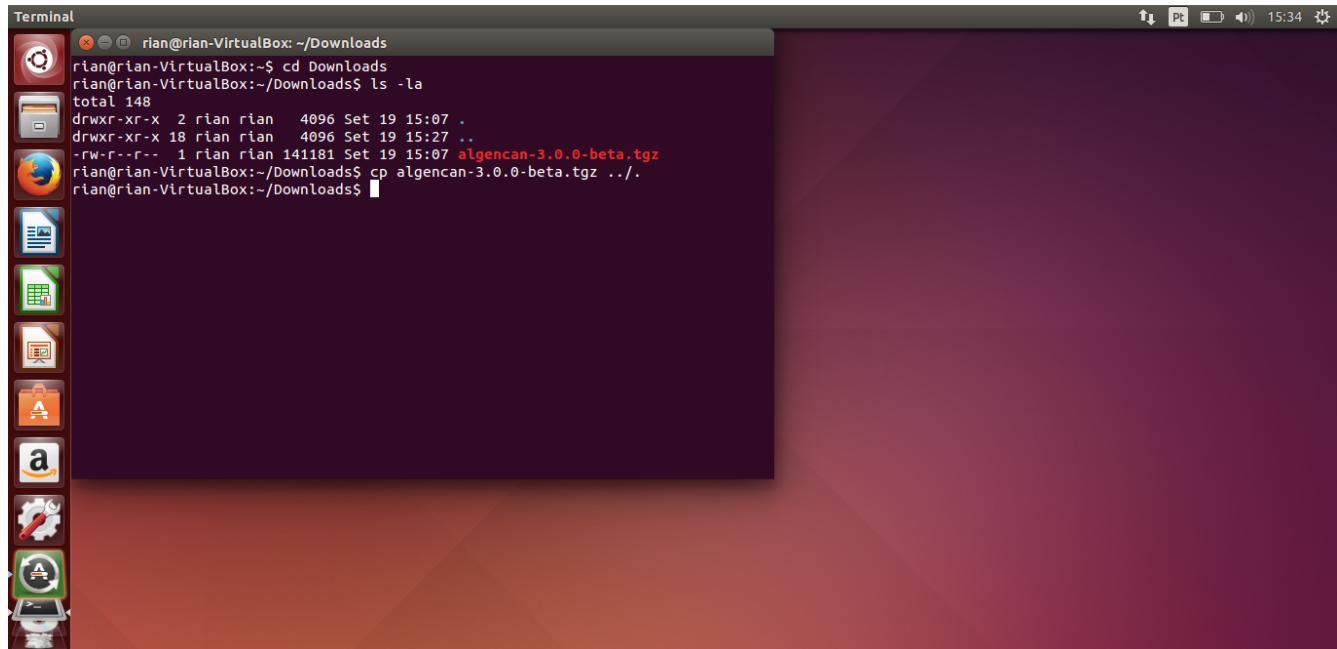


Figura 4.4: Copiando ALGENCAN na Pasta pessoal.

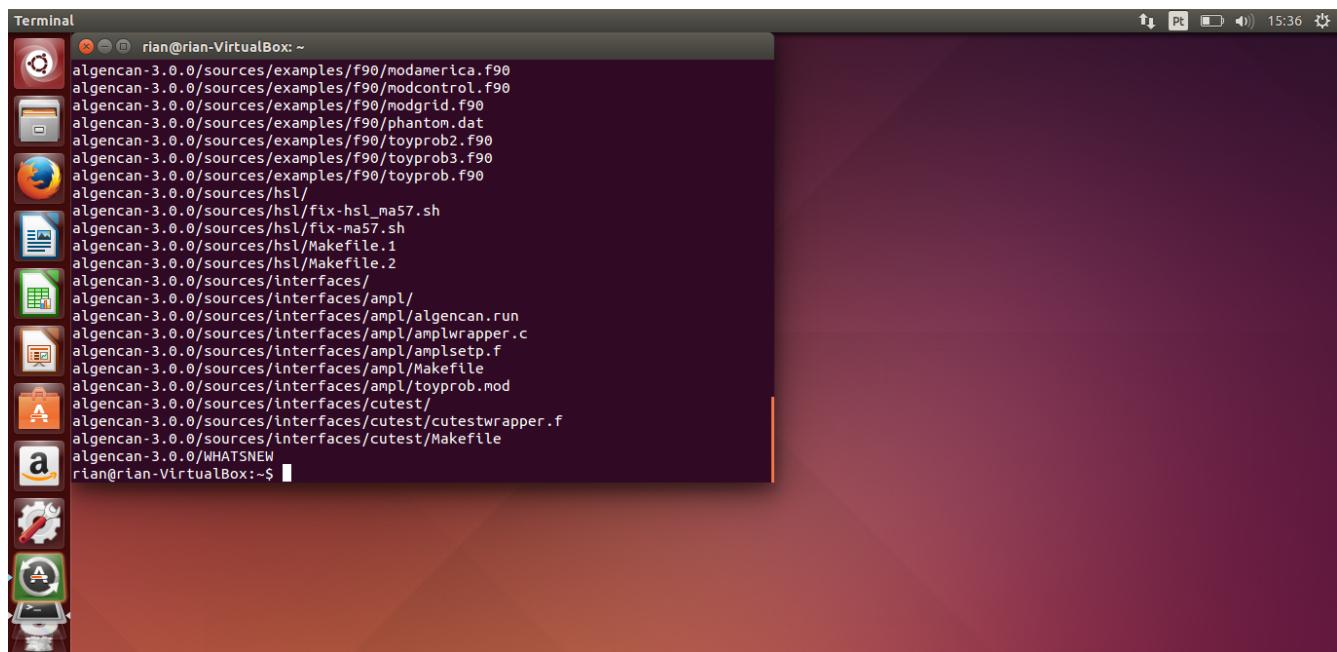


Figura 4.5: Descompactação do pacote ALGENCAN.

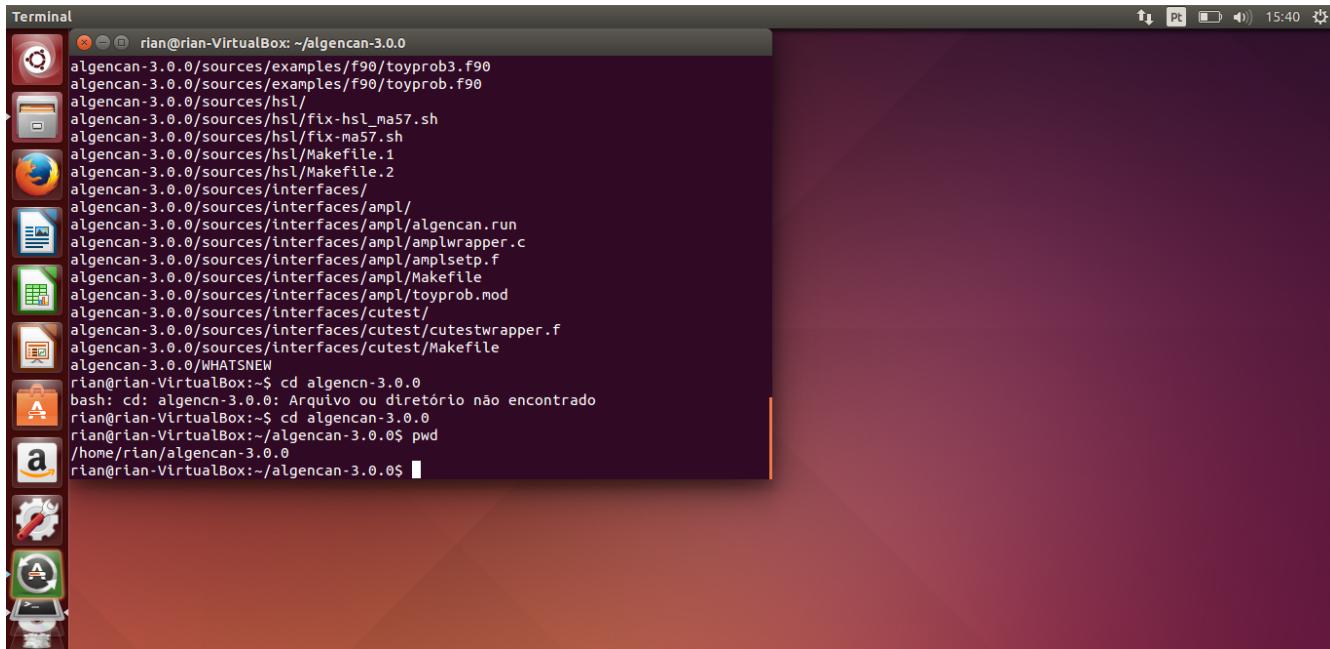


Figura 4.6: Na saída do comando `pwd` deve constar o diretório onde o usuário está trabalhando.

e na saída deverá aparecer ‘/home/<usuário>/algencan-3.0.0’, conforme Figura 4.6.

Na sequência, digite

`make`

e a saída deverá estar em conformidade com a Figura 4.7. Este comando irá fazer uma leitura do arquivo “Makefile” contido na pasta `algencan-3.0.0` que por vez irá compilar os arquivos onde a programação do ALGENCAN está escrita.

Agora será criada uma pasta dentro da pasta `algencan-3.0.0` para que seja possível trabalhar e manter uma organização, por exemplo, a pasta chamada `myfolder`. Digite

`mkdir myfolder`

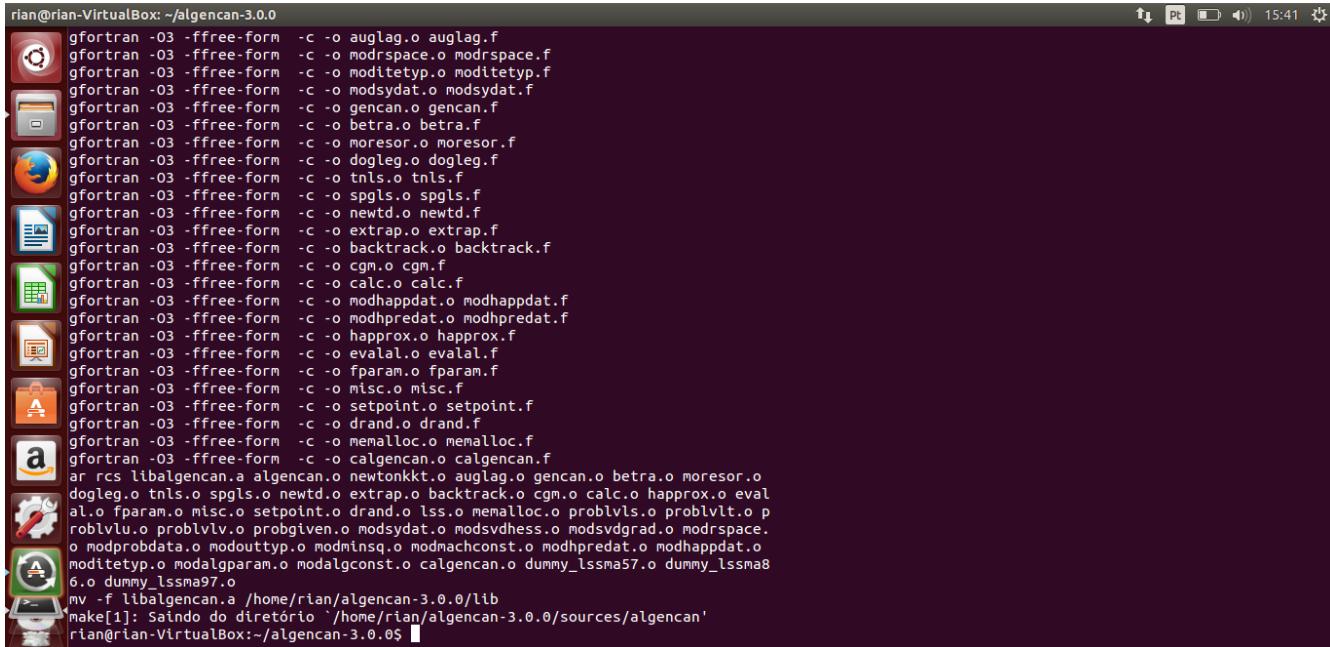
conforme Figura 4.8.

Observe que dentro da pasta `algencan-3.0.0` foi criada a pasta `myfolder`. Digite agora

`cd myfolder`

para entrar nesta pasta.

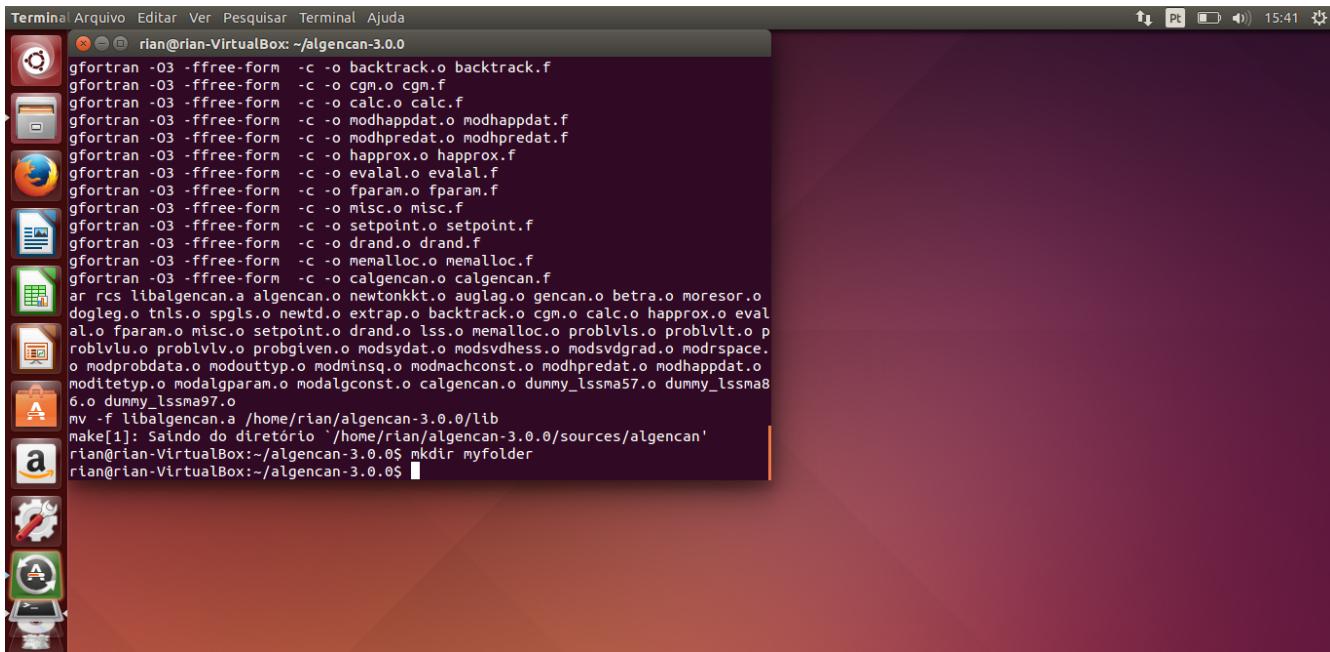
Um exemplo de problema a ser resolvido no ALGENCAN se encontra no arquivo `toyprob.f90`. Este será o arquivo com o qual o teste irá ser realizado. Então, digite



```

rian@rian-VirtualBox: ~/algencan-3.0.0
gfortran -O3 -ffree-form -c -o auglag.o auglag.f
gfortran -O3 -ffree-form -c -o modrspace.o modrspace.f
gfortran -O3 -ffree-form -c -o moditotyp.o moditotyp.f
gfortran -O3 -ffree-form -c -o modsydat.o modsydat.f
gfortran -O3 -ffree-form -c -o gencan.o gencan.f
gfortran -O3 -ffree-form -c -o betra.o betra.f
gfortran -O3 -ffree-form -c -o moresor.o moresor.f
gfortran -O3 -ffree-form -c -o dogleg.o dogleg.f
gfortran -O3 -ffree-form -c -o tnls.o tnls.f
gfortran -O3 -ffree-form -c -o spgls.o spgls.f
gfortran -O3 -ffree-form -c -o newtd.o newtd.f
gfortran -O3 -ffree-form -c -o extrap.o extrap.f
gfortran -O3 -ffree-form -c -o backtrack.o backtrack.f
gfortran -O3 -ffree-form -c -o cgm.o cgm.f
gfortran -O3 -ffree-form -c -o calc.o calc.f
gfortran -O3 -ffree-form -c -o modhappdat.o modhappdat.f
gfortran -O3 -ffree-form -c -o modhpredat.o modhpredat.f
gfortran -O3 -ffree-form -c -o approx.o approx.f
gfortran -O3 -ffree-form -c -o evalal.o evalal.f
gfortran -O3 -ffree-form -c -o fparam.o fparam.f
gfortran -O3 -ffree-form -c -o misc.o misc.f
gfortran -O3 -ffree-form -c -o setpoint.o setpoint.f
gfortran -O3 -ffree-form -c -o drand.o drand.f
gfortran -O3 -ffree-form -c -o memalloc.o memalloc.f
gfortran -O3 -ffree-form -c -o calgencan.o calgencan.f
ar rcs libalgencan.a algencan.o newtonkkt.o auglag.o gencan.o betra.o moresor.o
dogleg.o tnls.o spgls.o newtd.o extrap.o backtrack.o cgm.o calc.o approx.o evalal.o
fparam.o misc.o setpoint.o drand.o lss.o memalloc.o problvlt.o p
roblvlu.o problvlv.o probgivten.o modsydat.o modsvdhess.o modsvdgrad.o modrspace.o
modprobdata.o modouttyp.o modminsq.o modmachconst.o modhpredat.o modhappdat.o
moditotyp.o modalgparam.o modalgconst.o calgencan.o dummy_lssma57.o dummy_lssma8
6.o dummy_lssma97.o
mv -f libalgencan.a /home/rian/algencan-3.0.0/lib
make[1]: Saindo do diretório '/home/rian/algencan-3.0.0/sources/algencan'
rian@rian-VirtualBox:~/algencan-3.0.0$ 

```

Figura 4.7: Saída do comando `make`.


```

Terminal Arquivo Editar Ver Pesquisar Terminal Ajuda
rian@rian-VirtualBox: ~/algencan-3.0.0
gfortran -O3 -ffree-form -c -o backtrack.o backtrack.f
gfortran -O3 -ffree-form -c -o cgm.o cgm.f
gfortran -O3 -ffree-form -c -o calc.o calc.f
gfortran -O3 -ffree-form -c -o modhappdat.o modhappdat.f
gfortran -O3 -ffree-form -c -o modhpredat.o modhpredat.f
gfortran -O3 -ffree-form -c -o approx.o approx.f
gfortran -O3 -ffree-form -c -o evalal.o evalal.f
gfortran -O3 -ffree-form -c -o fparam.o fparam.f
gfortran -O3 -ffree-form -c -o misc.o misc.f
gfortran -O3 -ffree-form -c -o setpoint.o setpoint.f
gfortran -O3 -ffree-form -c -o drand.o drand.f
gfortran -O3 -ffree-form -c -o memalloc.o memalloc.f
gfortran -O3 -ffree-form -c -o calgencan.o calgencan.f
ar rcs libalgencan.a algencan.o newtonkkt.o auglag.o gencan.o betra.o moresor.o
dogleg.o tnls.o spgls.o newtd.o extrap.o backtrack.o cgm.o calc.o approx.o evalal.o
fparam.o misc.o setpoint.o drand.o lss.o memalloc.o problvlt.o p
roblvlu.o problvlv.o probgivten.o modsydat.o modsvdhess.o modsvdgrad.o modrspace.o
modprobdata.o modouttyp.o modminsq.o modmachconst.o modhpredat.o modhappdat.o
moditotyp.o modalgparam.o modalgconst.o calgencan.o dummy_lssma57.o dummy_lssma8
6.o dummy_lssma97.o
mv -f libalgencan.a /home/rian/algencan-3.0.0/lib
make[1]: Saindo do diretório '/home/rian/algencan-3.0.0/sources/algencan'
mkdir myfolder
rian@rian-VirtualBox:~/algencan-3.0.0$ 

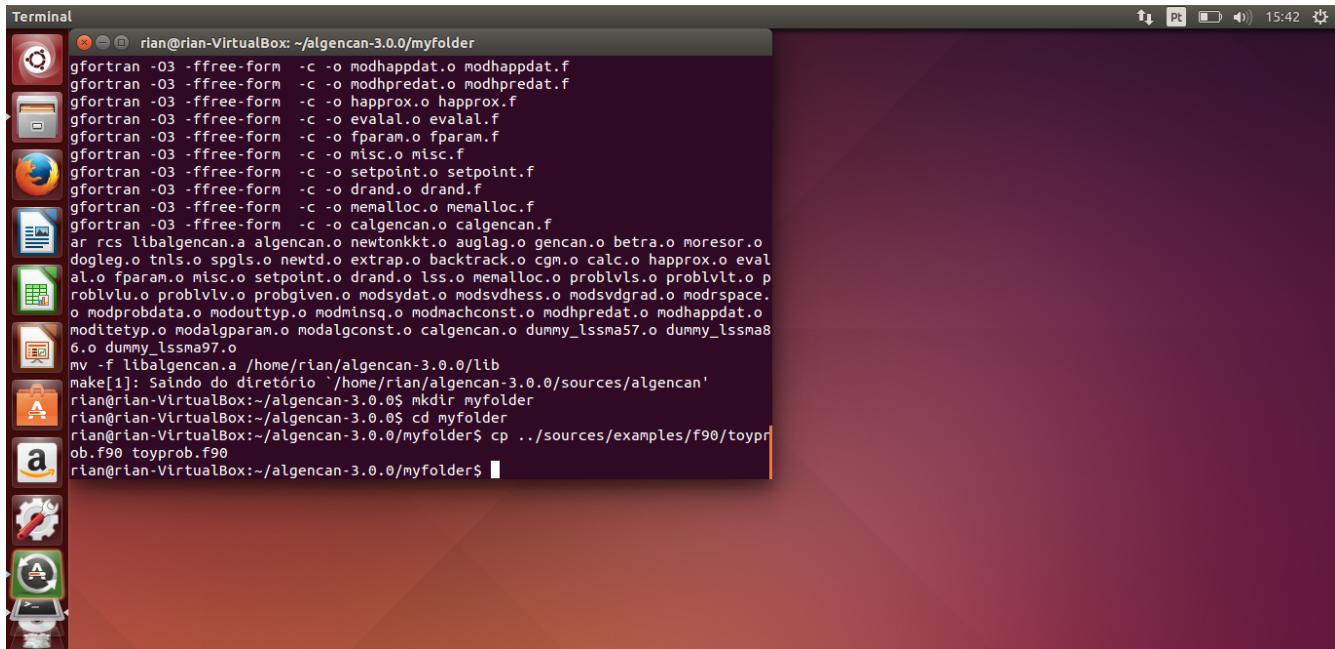
```

Figura 4.8: Saída do comando `mkdir myfolder`.

```
cp ../sources/examples/f90/toyprob.f90 .
```

para copiar o arquivo **toyprob.f90** na pasta *myfolder*. Note que existe um ponto final no final do comando.

Veja Figura 4.9 para uma ilustração do comando realizado.



```
Terminal
rian@rian-VirtualBox: ~/algencan-3.0.0/myfolder
gfortran -O3 -ffree-form -c -o modhappdat.o modhappdat.f
gfortran -O3 -ffree-form -c -o modhpredat.o modhpredat.f
gfortran -O3 -ffree-form -c -o approx.o approx.f
gfortran -O3 -ffree-form -c -o evalal.o evalal.f
gfortran -O3 -ffree-form -c -o fparam.o fparam.f
gfortran -O3 -ffree-form -c -o misc.o misc.f
gfortran -O3 -ffree-form -c -o setpoint.o setpoint.f
gfortran -O3 -ffree-form -c -o drand.o drand.f
gfortran -O3 -ffree-form -c -o memalloc.o memalloc.f
gfortran -O3 -ffree-form -c -o algencan.o algencan.f
ar rcs libalgencan.a algencan.o newtonkkt.o auglag.o gencan.o betra.o moresor.o
dogleg.o tnls.o spgl.o newtd.o extrap.o backtrack.o cgm.o calc.o approx.o eval
al.o fparam.o misc.o setpoint.o drand.o lss.o memalloc.o probolvls.o probolvlt.o p
roblvlu.o probolv.o probgivin.o modsydat.o modsvdhess.o modsvdgrad.o modrspace.
o modprobdata.o modouttyp.o modminsq.o modmachconst.o modhpredat.o modhappdat.o
moditetyp.o modalgparam.o modalgconst.o algencan.o dummy_lssma57.o dummy_lssma8
6.o dummy_lssma97.o
mv -f libalgencan.a /home/rian/algencan-3.0.0/lib
make[1]: Saindo do diretório `/home/rian/algencan-3.0.0/sources/algencan'
rian@rian-VirtualBox:~/algencan-3.0.0$ mkdir myfolder
rian@rian-VirtualBox:~/algencan-3.0.0$ cd myfolder
rian@rian-VirtualBox:~/algencan-3.0.0/myfolder$ cp ../sources/examples/f90/toypr
ob.f90 toyprob.f90
rian@rian-VirtualBox:~/algencan-3.0.0/myfolder$
```

Figura 4.9: Copiando **toyprob.f90** na pasta *myfolder*.

Agora digite

```
export ALGENCAN=/home/<usuário>/algencan-3.0.0
```

para criar uma variável de ambiente. Basicamente isto é feito para indicar onde está o ALGENCAN que o usuário irá usar, conforme Figura 4.10.

Digite

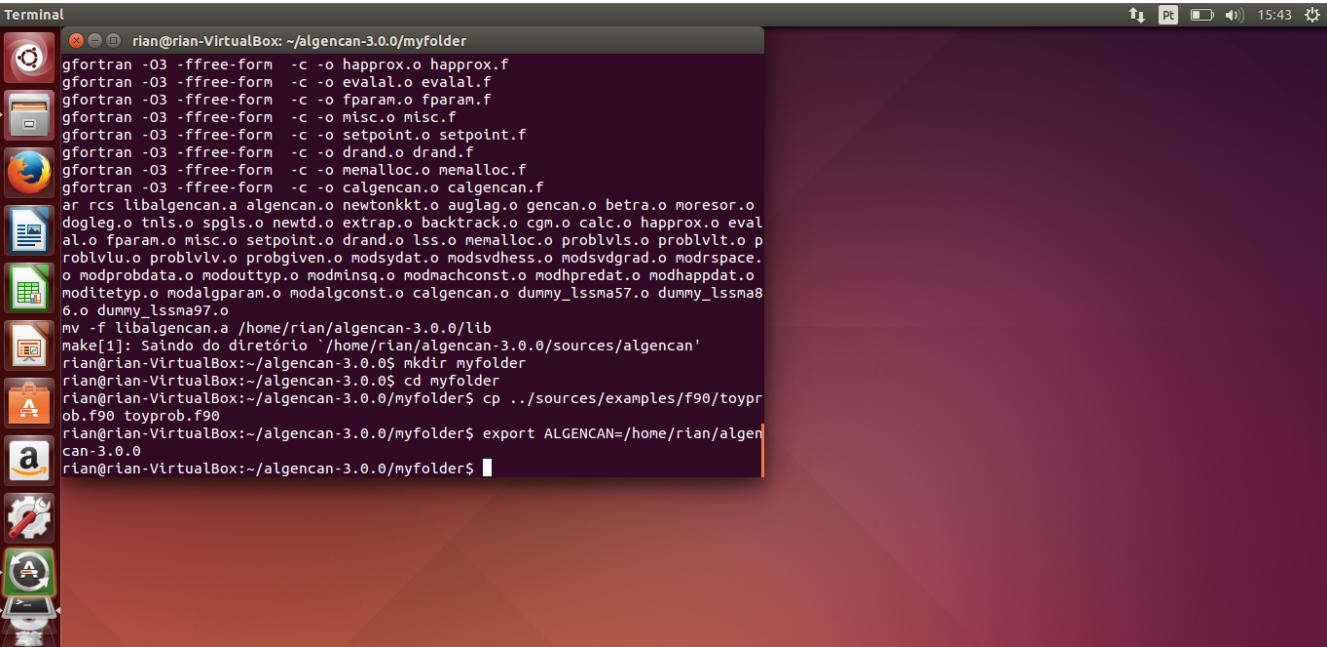
```
echo $ALGENCAN
```

para se certificar que a variável definida está correta. Deverá aparecer na saída

```
home/<usuário>/algencan-3.0.0,
```

conforme Figura 4.11.

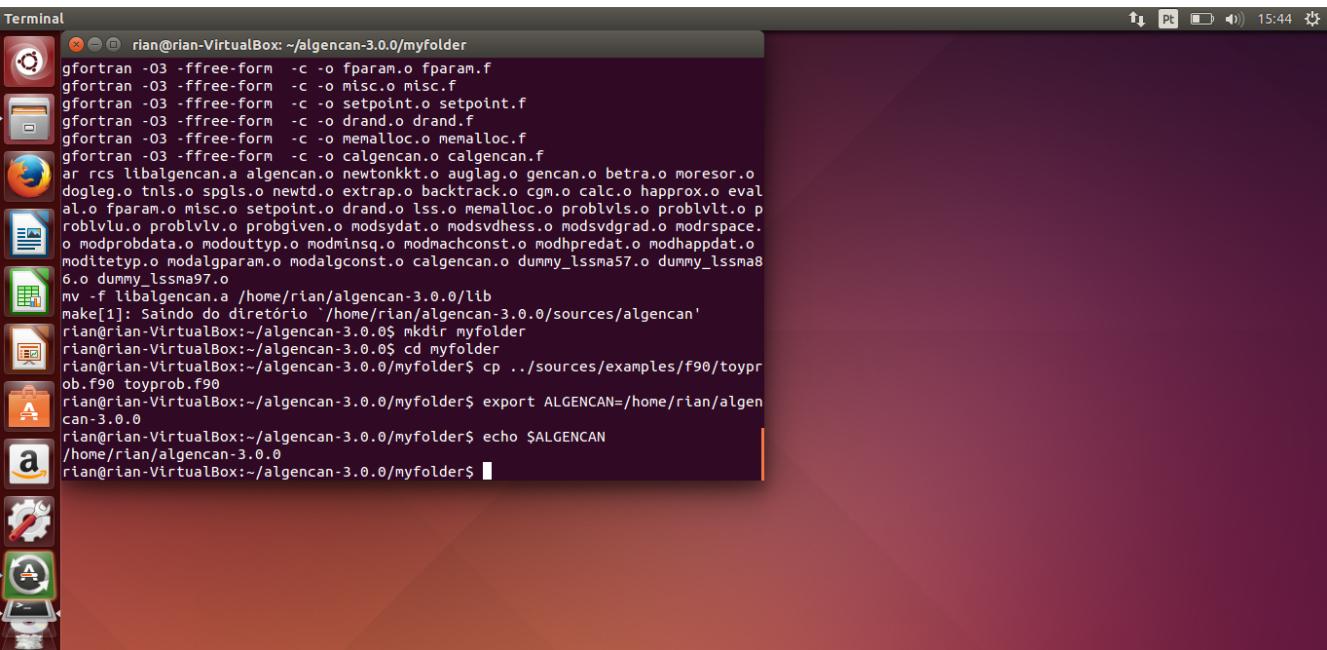
Agora digite



```

Terminal
rian@rian-VirtualBox: ~/algencan-3.0.0/myfolder
gfortran -O3 -ffree-form -c -o happen.o happen.f
gfortran -O3 -ffree-form -c -o eval.o eval.f
gfortran -O3 -ffree-form -c -o fparam.o fparam.f
gfortran -O3 -ffree-form -c -o misc.o misc.f
gfortran -O3 -ffree-form -c -o setpoint.o setpoint.f
gfortran -O3 -ffree-form -c -o drand.o drand.f
gfortran -O3 -ffree-form -c -o memalloc.o memalloc.f
gfortran -O3 -ffree-form -c -o calgencan.o calgencan.f
ar rcs libalgencan.a algencan.o newtonkt.o auglag.o gencan.o betra.o moresor.o
dogleg.o tnls.o spgl.o newtd.o extrap.o backtrack.o cgm.o calc.o happen.o eval
al.o fparam.o misc.o setpoint.o drand.o lss.o memalloc.o probvlvls.o probvlvt.o p
robvlvlu.o probvlv.o probgivin.o modsdat.o modsvdhess.o modsvdgrad.o modrspace.
o modprobdata.o modouttyp.o modminsq.o modmachconst.o modhpredat.o modhappdat.o
moditety.o modalparam.o modalconst.o calgencan.o dummy_lssma57.o dummy_lssma8
6.o dummy_lssma97.o
mv -f libalgencan.a /home/rian/algencan-3.0.0/lib
make[1]: Saindo do diretório '/home/rian/algencan-3.0.0/sources/algencan'
rian@rian-VirtualBox:~/algencan-3.0.0$ mkdir myfolder
rian@rian-VirtualBox:~/algencan-3.0.0$ cd myfolder
rian@rian-VirtualBox:~/algencan-3.0.0/myfolder$ cp ../sources/examples/f90/toypr
ob.f90 toyprob.f90
rian@rian-VirtualBox:~/algencan-3.0.0/myfolder$ export ALGENCAN=/home/rian/alg
can-3.0.0
rian@rian-VirtualBox:~/algencan-3.0.0/myfolder$ 
```

Figura 4.10: Criando uma varável de ambiente.



```

Terminal
rian@rian-VirtualBox: ~/algencan-3.0.0/myfolder
gfortran -O3 -ffree-form -c -o fparam.o fparam.f
gfortran -O3 -ffree-form -c -o misc.o misc.f
gfortran -O3 -ffree-form -c -o setpoint.o setpoint.f
gfortran -O3 -ffree-form -c -o drand.o drand.f
gfortran -O3 -ffree-form -c -o memalloc.o memalloc.f
gfortran -O3 -ffree-form -c -o calgencan.o calgencan.f
ar rcs libalgencan.a algencan.o newtonkt.o auglag.o gencan.o betra.o moresor.o
dogleg.o tnls.o spgl.o newtd.o extrap.o backtrack.o cgm.o calc.o happen.o eval
al.o fparam.o misc.o setpoint.o drand.o lss.o memalloc.o probvlvls.o probvlvt.o p
robvlvlu.o probvlv.o probgivin.o modsdat.o modsvdhess.o modsvdgrad.o modrspace.
o modprobdata.o modouttyp.o modminsq.o modmachconst.o modhpredat.o modhappdat.o
moditety.o modalparam.o modalconst.o calgencan.o dummy_lssma57.o dummy_lssma8
6.o dummy_lssma97.o
mv -f libalgencan.a /home/rian/algencan-3.0.0/lib
make[1]: Saindo do diretório '/home/rian/algencan-3.0.0/sources/algencan'
rian@rian-VirtualBox:~/algencan-3.0.0$ mkdir myfolder
rian@rian-VirtualBox:~/algencan-3.0.0$ cd myfolder
rian@rian-VirtualBox:~/algencan-3.0.0/myfolder$ cp ../sources/examples/f90/toypr
ob.f90 toyprob.f90
rian@rian-VirtualBox:~/algencan-3.0.0/myfolder$ export ALGENCAN=/home/rian/alg
can-3.0.0
rian@rian-VirtualBox:~/algencan-3.0.0/myfolder$ echo $ALGENCAN
/home/rian/algencan-3.0.0
rian@rian-VirtualBox:~/algencan-3.0.0/myfolder$ 
```

Figura 4.11: Certificando que a varável de ambiente foi criada.

```

Terminal rian@rian-VirtualBox: ~/algencan-3.0.0/myfolder
gfortran -O3 -ffree-form -c -o setpoint.o setpoint.f
gfortran -O3 -ffree-form -c -o drand.o drand.f
gfortran -O3 -ffree-form -c -o memalloc.o memalloc.f
gfortran -O3 -ffree-form -c -o calgencan.o calgencan.f
ar rcs libalgencan.a algencan.o newtonkt.o auglag.o gencan.o betra.o moresr.o
dogleg.o tlns.o spgls.o newtd.o extrap.o backtrack.o cgm.o calc.o happrox.o eval
al.o fparam.o misc.o setpoint.o drand.o lss.o memalloc.o problvls.o problvlt.o p
roblvlu.o problvlv.o probgivin.o modsydat.o modsvdhess.o modsvdgrad.o modrspace.
o modprobdata.o modouttyp.o modmnsq.o modmachconst.o modhpredat.o modhappdat.o
moditetyp.o modalparam.o modalconst.o calgencan.o dummy_lssma57.o dummy_lssma8
6.o dummy_lssma97.o
mv -f libalgencan.a /home/rian/algencan-3.0.0/lib
make[1]: Saindo do diretório '/home/rian/algencan-3.0.0/sources/algencan'
rian@rian-VirtualBox:~/algencan-3.0.0$ mkdir myfolder
rian@rian-VirtualBox:~/algencan-3.0.0$ cd myfolder
rian@rian-VirtualBox:~/algencan-3.0.0$ cp ..sources/examples/f90/toypr
ob.f90 toyprob.f90
rian@rian-VirtualBox:~/algencan-3.0.0$ export ALGENCAN=/home/rian/alg
can-3.0.0
rian@rian-VirtualBox:~/algencan-3.0.0$ echo $ALGENCAN
/home/rian/algencan-3.0.0
rian@rian-VirtualBox:~/algencan-3.0.0$ gfortran toyprob.f90 -L$ALGENCAN
/lib -lalgencan -o teste
rian@rian-VirtualBox:~/algencan-3.0.0$ 
```

Figura 4.12: Compilando o arquivo **toyprob.f90**.

```
gfortran toyprob.f90 -L$ALGENCAN/lib -lalgencan -o teste
```

para compilar, conforme Figura 4.12.

Observe que será criado um arquivo executável com o nome ‘teste’ na pasta *myfolder*. Digite

```
./teste
```

para executá-lo. Na saída deverá aparecer o mesmo que na Figura 4.13, isto significa que o problema de otimização contido no arquivo **toyprob.f90** foi resolvido com sucesso.

Observação 4.1.1. Cuidado ao digitar os comandos, especialmente com os espaços entre palavras e com os caracteres.

Observação 4.1.2. Depois de digitar cada comando é necessário teclar Enter.

Observação 4.1.3. Toda vez que o usuário fechar o Terminal e depois desejar compilar o aquivo para executá-lo, deverá se dirigir até a pasta *myfolder* e criar novamente a variável de ambiente, isto é, deverá digitar novamente **export ALGENCAN=/home/<usuário>/algencan-3.0.0**.

Para a melhor compreensão dos comandos acima descritos, vide Apêndice A na página 129.

```

rivan@rivan-S425-G-BC34P1:~/algenca-3.0.0/myfolder$ ./teste
=====
This is ALGENCAN 3.0.0.
ALGENCAN, an Augmented Lagrangian method for nonlinear programming, is part of
the TANGO Project: Trustable Algorithms for Nonlinear General Optimization.
See http://www.ime.usp.br/~egbirgin/tango/ for details.
=====

There are no strings to be processed in the array of parameters.

The specification file is not being used.

Available HSL subroutines = NONE

ALGENCAN PARAMETERS:

firstde      =          T
seconde      =          T
truehpr      =          T
htype in TN =      TRUEHP
lsslvr in TR =    NONE/NONE
lsslvr in NW =    NONE/NONE
lsslvr in ACCPROC =    NONE/NONE
innslvr      =      TN
accproc      =          F
rmfixv      =          T
slacks      =          F
scale      =          T
epsfeas      =  1.0000D-08
epsopt      =  1.0000D-08
efstain      =  1.0000D-04
eostain      =  1.0000D-12
efacc      =  1.0000D-04
eoacc      =  1.0000D-04
iprint      =          10
ncomp      =          6

Specification filename =  ''
Output filename =  ''
Solution filename =  ''

```

Figura 4.13: Finalização do teste com o ALGENCAN.

4.2 Utilização do ALGENCAN

Como foi dito na seção anterior, o arquivo **toyprob.f90** é um exemplo de problema de otimização a ser resolvido pelo ALGENCAN quando executado. Logo, para que o usuário possa realizar suas próprias otimizações é importante entender as principais linhas de programação contidas no arquivo **toyprob.f90** a fim de alterá-las quando necessário.

Vá na pasta *myfolder*, criada para a realização do teste, e abra o arquivo **toyprob.f90**, conforme Figura 4.14.

Observe que aparece a linha de comando

$$n = 2,$$

isto significa que o número de variáveis (*Number of variables*) trabalhadas é dois. Ou seja, o problema conta com x_1 e x_2 .

Mais abaixo aparece

```

toyprob.f90 (~/algencan-3.0.0/myfolder) - gedit
! ****
! ****
program algencanna
  implicit none
! LOCAL SCALARS
  logical :: checkder
  integer :: allocerr,hnnzmax,hnnzmax1,hnnzmax2,hnnzmax3, &
             inform,jcnzmax,m,n,vparam
  real(kind=8) :: cnorm,efacc,efstain,eoacc,eostain,epsfeas,epsopt, &
                 f,nlpupn,snorm
! LOCAL ARRAYS
  character(len=15) :: strtmp
  character(len=80) :: specfnm,outputfnm,vparam(10)
  logical :: coded(11)
  logical, pointer :: equatn(:,linear(:))
  real(kind=8), pointer :: l(:,lambda(:,u(:,x(:)))
! EXTERNAL SUBROUTINES
  external :: myevalf,myevalg,myevalh,myevaljac,myevalhc, &
             myevalfc,myevalgjac,myevalgjcp,myevalhl,myevalhlp
! Number of variables
n = 2
! Set lower bounds, upper bounds, and initial guess
allocate(x(n),l(n),u(n),stat=allocerr)
if ( allocerr .ne. 0 ) then
  write(*,*) 'Allocation error in main program'
  stop
end if

```

Fortran 95 ▾ Largura da tabulação: 8 ▾ Lin 1, Col 1 INS

Figura 4.14: Arquivo **toyprob.f90**.

```

l(1) = -10.0d0
u(1) = 10.0d0
l(2) = -1.0d+20
u(2) = 1.0d+20
x(1:2) = 0.0d0

```

O que está sendo feito é a limitação do valor das variáveis. A notação $-10.0d0$ representa $-10 \times 10^0 = -10$ e a notação $-1.0d+20$ representa $-1 \times 10^{20} = -10^{20}$. Já as letras “l” e “u” significam *lower* e *upper* respectivamente. O equivalente matematicamente é

$$\begin{aligned}-10 \leq x_1 &\leq 10 \\ -10^{20} \leq x_2 &\leq 10^{20}.\end{aligned}$$

Já a expressão

$$x(1:2) = 0.0d0$$

diz que o valor inicial das variáveis é zero.

Observação 4.2.1. Note que x_2 estar restringido entre -10^{20} e 10^{20} é praticamente deixar esta variável livre. Isto é feito porque a teoria do ALGENCAN exige que as variáveis estejam em um compacto.

Na sequência aparece o número de restrições m , neste caso, $m = 2$. A linha de comando

```
equatn(1:m) = .false.
```

significa que as restrições de 1 até m não são de igualdade (portanto, ambas são de desigualdade). Em ALGENCAN as desigualdades são sempre consideradas da forma menor ou igual a zero. Note que ainda não foram determinadas quais são essas restrições propriamente ditas nem qual é a função objetivo, isto será feito mais a frente.

Como para cada restrição é vinculado um valor de lambda (multiplicador de Lagrange), então a linha

```
lambda(1:m) = 0.0d0
```

significa que os m (igual a 2) valores de lambda começam com valor inicial igual a zero.

É preciso também informar a linearidade das restrições. As linhas

```
linear(1) = .false.
linear(2) = .true.
```

dizem que a primeira restrição não é linear e a segunda é.

Mais abaixo, no arquivo **toyprob.f90**, em “Coded Subroutines”,

```
coded(1:6) = .true.
coded(7:11) = .false.
```

o usuário informará quais subrotinas estarão sendo passadas para o programa. No exemplo, estão ativas as 6 primeiras subrotinas. Já da sétima até a décima primeira estão desativadas. Estes comandos não precisarão ser modificados em problemas usuais de otimização. Eles indicam que o usuário fornecerá uma subrotina que registra a função objetivo, outra que registra o gradiente da função objetivo, outra que registra a sua Hessiana, outra registra as restrições, outra o Jacobiano das restrições e a sexta registra a Hessiana das restrições.

A linha

```
jcnnzmax = 4
```

informa que quatro é o espaço de memória reservado para armazenar os elementos não nulos do jacobiano das restrições. Desta forma, `jcnnzmax` deve ser maior ou igual ao número máximo destes elementos.

O próximo parâmetro necessário em ALGENCAN é `hnnzmax`. Este parâmetro registra o espaço de memória reservado para os elementos não nulos das hessianas da função objetivo e de cada uma das m restrições. No exemplo referente ao arquivo `toyprob.f90` a definição de `hnzzmax` é feita usando as variáveis auxiliares `hnnzmax1`, `hnnzmax2` e `hnnzmax3`.

A linha

```
hnnzmax1 = 0
```

diz que há no máximo zero elementos não nulos na matriz hessiana da função objetivo. Isto é, todos os elementos são nulos.

A linha

```
hnnzmax2 = 1
```

diz que há no máximo um elemento não nulo na matriz hessiana da primeira restrição.

A linha

```
hnnzmax3 = 6
```

diz que o espaço de memória reservado para armazenar os elementos não nulos na matriz hessiana da segunda restrição é de seis posições. Isso significa que há no máximo seis destes elementos, pois caso contrário os demais não poderiam ser armazenados. Vale notar que no exemplo referente ao arquivo `toyprob.f90` esta restrição é linear e, portanto sua hessiana não possui elementos não nulos. Desta forma seria possível colocar `hnnzmax3 = 0`, mas, em problemas pequenos, não há problemas em reservar uma quantidade maior de memória do que a que realmente será usada. Observe que $0 \leq jcnnzmax \leq nm$ e $0 \leq hnnzmax \leq n^2 + mn^2$. Sendo assim, o usuário que não tenha informações sobre a esparsidade do seu problema poderia alocar o limitante superior para cada um destes parâmetros.

Se o problema a ser minimizado tiver mais que duas restrições então, caso o usuário deseje utilizar as variáveis auxiliares, ele deverá declarar `hnnzmax4`, `hnnzmax5`, etc. juntamente com as já declaradas, deverá defini-las na sequência das três já definidas e deverá acrescentá-las como parcelas que tem como soma `hnnzmax`.

Caso o usuário preferir definir `hnnzmax` diretamente não há necessidade de usar as variáveis auxiliares `hnnzmax1`, `hnnzmax2`, etc.

Em “Checking derivatives?” é necessário informar ao programa se o usuário deseja que o valor das derivadas informadas seja conferido através do método das diferenças finitas. Caso a resposta seja positiva, deve-se fazer `checkder = .true.` e no caso negativo `checkder = .false..`

A primeira subrotina passada para o ALGENCAN chama-se “myevalf” e é nela onde o usuário informará qual é a função objetivo do problema de otimização. Isto se dá na linha

$$f = x(2),$$

que significa matematicamente que a função a ser minimizada é $f(x) = x_2$.

A segunda subrotina chama-se “myevalg”. Nela deverá ser informado o gradiente relativo à função objetivo. Em

$$\begin{aligned} g(1) &= 0.0d0 \\ g(2) &= 1.0d0 \end{aligned}$$

tem-se que a derivada da função objetivo em relação à primeira variável é zero. Matematicamente, $\frac{\partial f}{\partial x_1} = 0$. Já a derivada em relação à segunda variável é 1. Isto pode ser facilmente comprovado algebricamente.

Na terceira subrotina, intitulada “myevalh”, o usuário deverá informar o número máximo de elementos não nulos da matriz hessiana da função objetivo. Como a função objetivo neste exemplo é $f(x) = x_2$ então $\nabla^2 f(x) = 0$, o que se traduz à linha

$$hnnz = 0.$$

Na quarta subrotina, “myevalc”, o usuário introduzirá a expressão das restrições. Esta subrotina calcula em c o valor da ind-ésima restrição. A linha

```
if ( ind .eq. 1 ) then
```

diz que se trata da primeira restrição por causa no número 1 contido nela. Logo abaixo há a linha

$$c = x(1) ** 2 + 1 - x(n),$$

que diz que a restrição é $x_1^2 + 1 - x_2 \leq 0$. Isso vem do fato que $n = 2$ e que, como visto, ambas as restrições são de desigualdade. A simbologia `**` é de potenciação.

De forma análoga, tem-se que a segunda restrição é $2 - x_1 - x_2 \leq 0$. O usuário deverá ser cuidadoso se precisar incluir ou excluir uma restrição. Para que isto ocorra é necessário que as linhas, similares à Figura 4.15, sejam incluídas ou excluídas com seus respectivos `ind`'s associados.

A quinta subrotina chama-se “myevaljac”, nela o usuário informará os valores não nulos da matriz jacobiana das restrições. Observe a Figura 4.16.

Em

```

if ( ind .eq. 1 ) then
c = x(1) ** 2 + 1 - x(n)
return
end if

```

Figura 4.15: Identificação de uma restrição em ALGENCAN.

```

if ( ind .eq. 1 ) then
jcnnz = 2

if ( jcnnz .gt. lim ) then
lmem = .true.
return
end if

jcvar(1) = 1
jcval(1) = 2.0d0 * x(1)
jcvar(2) = 2
jcval(2) = - 1.0d0

return
end if

if ( ind .eq. 2 ) then
jcnnz = 2

if ( jcnnz .gt. lim ) then
lmem = .true.
return
end if

jcvar(1) = 1
jcval(1) = - 1.0d0
jcvar(2) = 2
jcval(2) = - 1.0d0

return
end if

```

Figura 4.16: Informação das matrizes jacobianas das restrições.

```

if ( ind .eq. 1 ) then
jcnnz = 2

```

é dito que a linha da matriz jacobiana relativa à primeira restrição tem no máximo dois elementos não nulos. A seguir é dito como passar a posição e a expressão de cada um destes elementos.

Como ainda se está dentro do condicional em que $\text{ind} = 1$, serão fornecidas posições da primeira linha da matriz jacobiana. Em

```
jcvar(1) = 1
```

é dito que o primeiro elemento não nulo, indicado por `jcvar(1)` está na primeira coluna. Já em

```
jcval(1) = 2.0d0 * x(1)
```

é dito que o elemento vale $2x_1$.

Em

```
jcvar(2) = 2
jcval(2) = - 1.0d0
```

é dito que na primeira linha, ainda indicado por `ind. eq. 1`, e segunda coluna, agora indicado por `jcvar(2) = 2`, da matriz jacobiana das restrições, o elemento vale -1 .

Em

```
if ( ind .eq. 2 ) then
jcnnz = 2
```

é dito que a matriz jacobiana relativa à segunda restrição tem no máximo dois elementos não nulos.

A expressão

```
jcvar(1) = 1
jcval(1) = - 1.0d0
```

diz que o valor da segunda linha e primeira coluna é -1 .

Em

```
jcvar(2) = 2
jcval(2) = - 1.0d0
```

é dito que o valor da segunda linha e segunda coluna é -1 .

Observação 4.2.2. Na subrotina “myevaljac” são passados apenas os valores não nulos da matriz jacobiana. Isso é feito assim porque é comum em problemas reais de grande porte que vários dos elementos do jacobiano sejam nulos. Este fato é o que é chamado de esparsidade e o uso desta informação permite que a memória utilizada seja reduzida e também que as operações envolvendo esta matriz sejam feitas de forma eficiente, evitando a realização de cálculos desnecessários. Os números entre parênteses não se referem às variáveis mas sim a qual elemento não nulo se está inserindo e, portanto, devem estar compreendidos entre 1 e `jcnnz`.

Derivando as restrições é fácil de se comprovar que as posições e os valores passados estão corretos. Sejam h_1 e h_2 as duas restrições, onde

$$\begin{aligned} h_1 &: x_1^2 + 1 - x_2 \leq 0 \\ h_2 &: 2 - x_1 - x_2 \leq 0, \end{aligned}$$

então,

$$J_h = \begin{pmatrix} 2x_1 & -1 \\ -1 & -1 \end{pmatrix}.$$

Mais uma vez o usuário deverá tomar cuidado se for modificar esta subrotina, isto é, quando for inserir os valores dos elementos da matriz jacobiana. O usuário deverá sempre respeitar a estrutura da programação.

Por fim, a última subrotina ativada chama-se “myevalhc”. Nela o usuário colocará o valor dos elementos não nulos das matrizes hessianas das restrições. Observe a Figura 4.17.

Em

```
if ( ind .eq. 1 ) then
    hcnnz = 1
```

é dito que a matriz hessiana associada à primeira restrição (`ind .eq. 1`) possui no máximo um elemento não nulo (`hcnnz = 1`).

Em

```
hcrow(1) = 1
hccol(1) = 1
hcval(1) = 2.0d0
```

é dito que o elemento ocupa a primeira linha, indicada pelo lado direito de `hcrow(1) = 1`, e a primeira coluna, indicada pelo lado direito de `hccol(1) = 1` e seu valor é 2, indicado por `hcval(1) = 2`.

Em

```
if ( ind .eq. 2 ) then
    hcnnz = 0
```

é dito que a matriz hessiana associada à segunda restrição possui no máximo zero elementos não nulos, ou seja, é uma matriz nula.

```

      subroutine myevalhc
(n,x,ind,hcrow,hccol,hcval,hcnnz,lim,lmem,flag)

      implicit none

! SCALAR ARGUMENTS
      logical, intent(out) :: lmem
      integer, intent(in)  :: ind,lim,n
      integer, intent(out) :: flag,hcnnz

! ARRAY ARGUMENTS
      integer,      intent(out) :: hccol(lim),hcrc
      real(kind=8), intent(in)  :: x(n)
      real(kind=8), intent(out) :: hcval(lim)

      flag = 0
      lmem = .false.

      if ( ind .eq. 1 ) then
hcnnz = 1

      if ( hcnnz .gt. lim ) then
      lmem = .true.
      return
      end if

      hcrow(1) = 1
      hccol(1) = 1
      hcval(1) = 2.0d0

      return
      end if

      if ( ind .eq. 2 ) then
hcnnz = 0

```

Figura 4.17: Informação das matrizes hessianas das restrições.

Observação 4.2.3. Na subrotina “myevalhc” os números entre parênteses não se referem às variáveis mas sim a qual elemento não nulo se está inserindo e devem estar compreendidos entre 1 e hcnnz. Além disso, somente é necessário informar os elementos h_{ij} da forma $i \geq j$, uma vez que hessianas de funções suaves são simétricas.

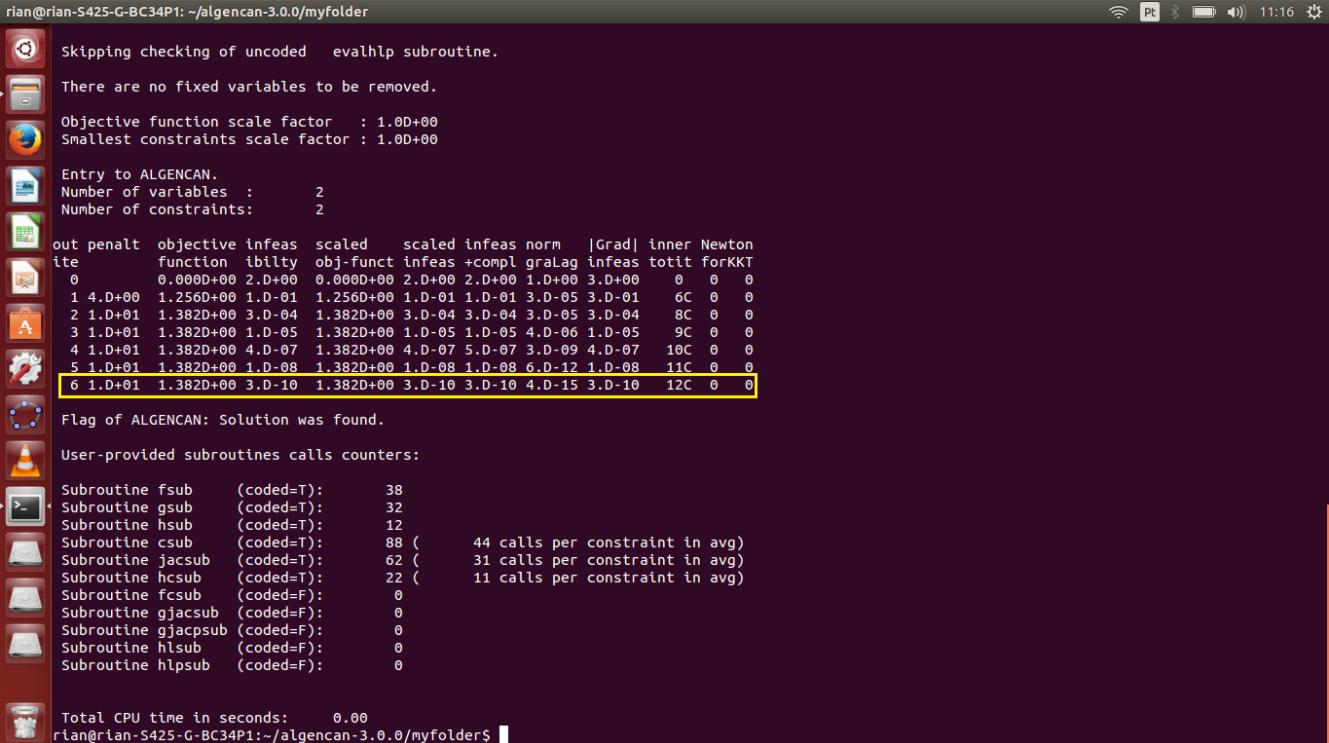
Matematicamente, o problema que o ALGENCAN está solucionando no arquivo **toyprob.f90** é

$$\begin{aligned}
 & \min x_2 \\
 & \text{sujeito a} \\
 & x_1^2 + 1 - x_2 \leq 0 \\
 & 2 - x_1 - x_2 \leq 0 \\
 & -10 \leq x_1 \leq 10 \\
 & -10^{20} \leq x_2 \leq 10^{20}
 \end{aligned}$$

A partir daqui o usuário está apto a modificar o arquivo **toyprob.f90** a fim de resolver seu próprio problema de otimização.

4.2.1 Compreendendo o resultado obtido com o ALGENCAN

Após executar o teste com o comando `./teste` observe a tela de saída, conforme Figura 4.18.



```

rian@rian-S425-G-BC34P1:~/algenkan-3.0.0/myfolder
Skipping checking of uncoded evalhlp subroutine.
There are no fixed variables to be removed.
Objective function scale factor : 1.0D+00
Smallest constraints scale factor : 1.0D+00
Entry to ALGENCAN.
Number of variables : 2
Number of constraints: 2
out penalt objective infeas scaled infeas norm |Grad| inner Newton
ite function iblty obj-funct infeas +compl graLag infeas totit forKKT
 0 0.000D+00 2.D+00 0.000D+00 2.D+00 2.D+00 1.D+00 3.D+00 0 0 0
 1 4.D+00 1.256D+00 1.D-01 1.256D+00 1.D-01 1.D-01 3.D-05 3.D-01 6C 0 0
 2 1.D+01 1.382D+00 3.D-04 1.382D+00 3.D-04 3.D-04 3.D-05 3.D-04 8C 0 0
 3 1.D+01 1.382D+00 1.D-05 1.382D+00 1.D-05 1.D-05 4.D-06 1.D-05 9C 0 0
 4 1.D+01 1.382D+00 4.D-07 1.382D+00 4.D-07 5.D-07 3.D-09 4.D-07 10C 0 0
 5 1.D+01 1.382D+00 1.D-08 1.382D+00 1.D-08 1.D-08 6.D-12 1.D-08 11C 0 0
 6 1.D+01 1.382D+00 3.D-10 1.382D+00 3.D-10 3.D-10 4.D-15 3.D-10 12C 0 0
Flag of ALGENCAN: Solution was found.

User-provided subroutines calls counters:
Subroutine fsub (coded=T): 38
Subroutine gsub (coded=T): 32
Subroutine hsub (coded=T): 12
Subroutine csub (coded=T): 88 ( 44 calls per constraint in avg)
Subroutine jacsu (coded=T): 62 ( 31 calls per constraint in avg)
Subroutine hcsu (coded=T): 22 ( 11 calls per constraint in avg)
Subroutine fcsub (coded=F): 0
Subroutine gjacsub (coded=F): 0
Subroutine gjacpsub (coded=F): 0
Subroutine hlsu (coded=F): 0
Subroutine hlpsub (coded=F): 0

Total CPU time in seconds: 0.00
rian@rian-S425-G-BC34P1:~/algenkan-3.0.0/myfolder$ 

```

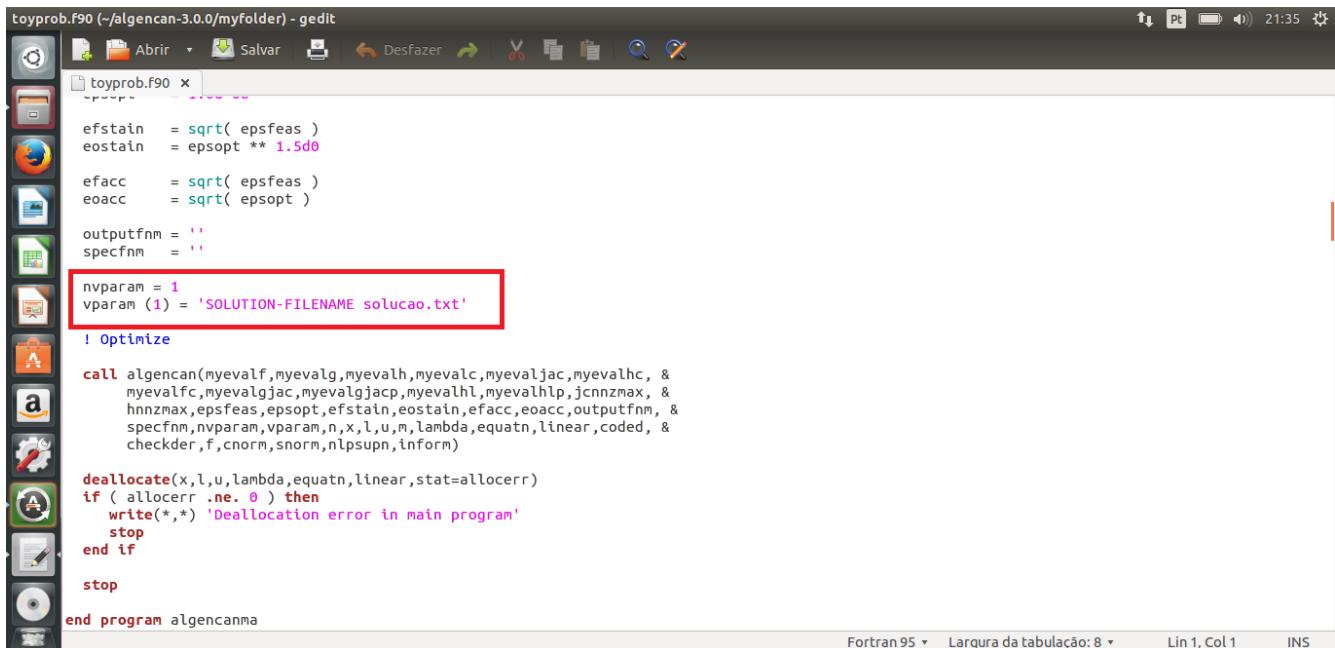
Figura 4.18: Tela de saída do comando `./teste`.

No retângulo em destaque se encontram, nas quatro primeiras colunas, o número de iterações para solucionar o problema de otimização, neste caso 6, o valor do parâmetro de penalidade, neste

caso 10, o valor da função objetivo, neste exemplo, 1,38 e a inviabilidade da resposta encontrada, neste caso é 3×10^{-10} .

Caso o usuário tenha optado por fazer `checkder = .true.`, ele deverá informar se quer visualizar os testes das derivadas nas subrotinas “`evalg`”, “`evaljac`”, “`evalh`” e “`evalhc`”, uma por vez e nesta ordem. Caso o usuário deseje verificar essas subrotinas deverá teclar Y e Enter, caso deseje abortar deverá teclar A e Enter (neste caso a tela com o resultado será exibida diretamente) e se quiser apenas pular a visualização da subrotina deverá teclar S e Enter.

Note que o valor das variáveis x_1 e x_2 não aparecem explícitas na tela, assim como não aparecem os valores dos multiplicadores de Lagrange. Para obter estas informações é necessário fazer mais alterações no arquivo `toyprob.f90`, conforme mostra a Figura 4.19.



```

toyprob.f90 (~/algencan-3.0.0/myFolder) - gedit
Abrir Salvar Desfazer Recuperar Desfazer Recuperar Busca
toyprob.f90 x
efstain = sqrt( epsfeas )
eostain = epsopt ** 1.5d0
efacc = sqrt( epsfeas )
eoacc = sqrt( epsopt )
outputfnm =
specfnm =
nvparam = 1
vparam (1) = 'SOLUTION-FILENAME solucao.txt'
! Optimize
call algencan(myevalf,myevalg,myevalh,myevalc,myevaljac,myevalhc, &
myevalfc,myevaljac,myevaljacp,myevalhl,myevalhlp,jcnnzmax, &
hnnzmax,epsfeas,epsopt,efstain,eostain,efacc,eoacc,outputfnm, &
specfnm,nvparam,vparam,n,x,l,u,m,lambda,equatn,linear,coded, &
checkder,f,cnorm,snorm,nlpsupn,inform)
deallocate(x,l,u,lambda,equatn,linear,stat=allocerr)
if ( allocerr .ne. 0 ) then
  write(*,*) 'Deallocation error in main program'
  stop
end if
stop
end program algencanma

```

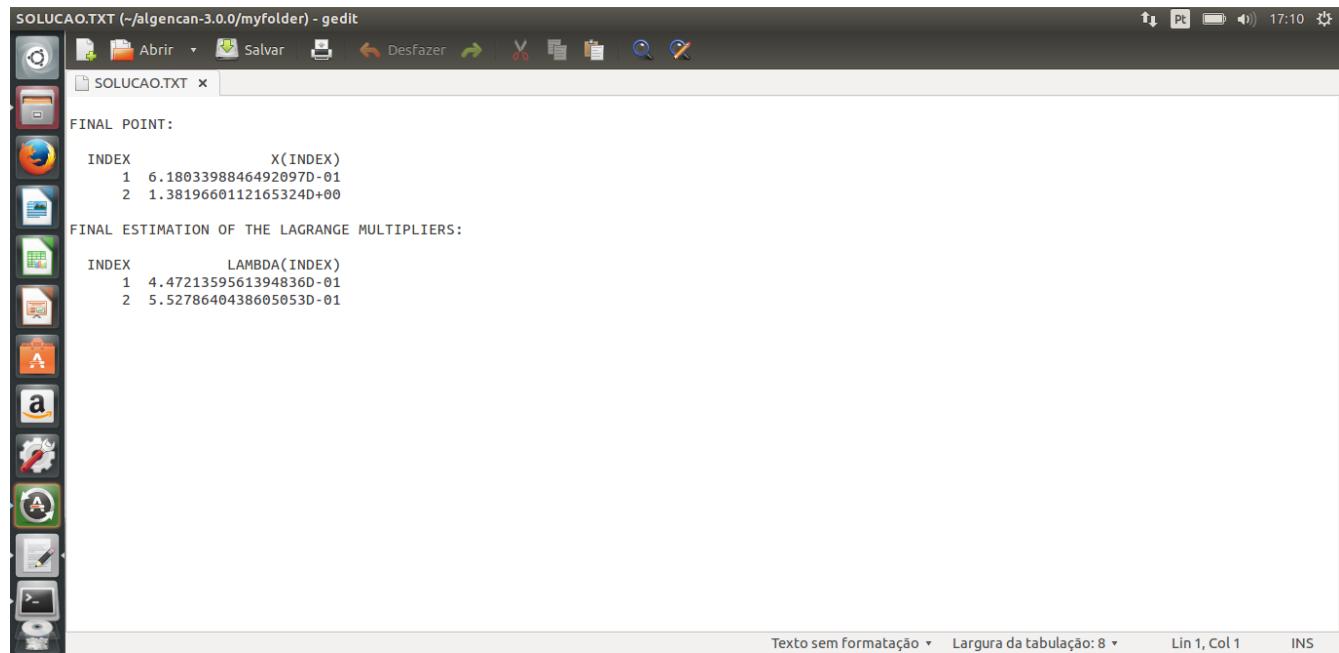
Fortran 95 ▾ Largura da tabulação: 8 ▾ Lin 1, Col 1 INS

Figura 4.19: Alteração no arquivo `toyprob.f90` para habilitar o arquivo com as soluções.

Com este comando, será criado na pasta `myfolder` o arquivo “`SOLUCAO.TXT`” e nele poderá ser encontrado o valor de x_1 e x_2 que minimiza a função objetivo e os valores dos multiplicadores de Lagrange, conforme Figura 4.20.

Analizando o arquivo “`SOLUCAO.TXT`” obtido pode-se concluir que, neste exemplo, $x_1 \approx 0,618$ e $x_2 \approx 1,38$. Além disso, $\lambda_1 \approx 0,447$ e $\lambda_2 \approx 0,552$.

Saídas mais detalhadas podem ser obtidas inserindo outras palavras chaves diferentes de “`SOLUTION-FILENAME`”. Algumas opções são



```
SOLUCAO.TXT (~/algencan-3.0.0/myfolder) - gedit
FINAL POINT:
INDEX          X(INDEX)
 1  6.1803398846492097D-01
 2  1.3819660112165324D+00

FINAL ESTIMATION OF THE LAGRANGE MULTIPLIERS:
INDEX          LAMBDA(INDEX)
 1  4.4721359561394836D-01
 2  5.5278640438605053D-01

Texto sem formatação ▾ Largura da tabulação: 8 ▾ Lin 1, Col 1 INS
```

Figura 4.20: Arquivo “SOLUCAO.TXT” em *myfolder*.

- PENALTY-PARAMETER-INITIAL-VALUE
- LINEAR-SYSTEMS-SOLVER-IN-TRUST-REGIONS

entre outras. Para incluir mais palavras-chave é necessário aumentar o valor de `nvparam`. Mais detalhes podem ser encontrados em [3] no Capítulo 10.

Capítulo 5

Problemas simétricos de otimização não linear

Neste capítulo serão estudados problemas simétricos em otimização não linear e sua resolução com ALGENCAN. Será mostrado que muitas vezes esses problemas possuem pontos atratores de ALGENCAN que não são soluções do problema. Além disso, para cada exemplo será construída uma tabela com os resultados obtidos por ALGENCAN ao solucionar o problema e os principais resultados serão analisados.

O significado de simetria pode ser formalmente escrito com a próxima definição.

Definição 5.0.1 (Simetria entre dois pontos). Dois pontos H e H' são ditos simétricos em relação ao ponto O se

- $d(H, O) = d(H', O)$;
- Dada uma reta r de modo que $H, O \in r$, então $H' \in r$. Isto é, os pontos H , H' e O estão alinhados.

Neste caso, O é chamado de *centro de simetria*.

Exemplo 5.0.1. Considere o problema

$$\begin{aligned} & \min x_1^2 + 3x_2^2 \\ & \text{sujeito a} \\ & 3x_1^2 + x_2^2 - 1 = 0. \end{aligned}$$

e sua representação geométrica na Figura 5.1.

Aplicando a teoria dos multiplicadores de Lagrange estudada no Capítulo 2, Seção 2.2, são encontrados analiticamente os pontos $A = \left(-\frac{\sqrt{3}}{3}, 0\right)$ e $B = \left(\frac{\sqrt{3}}{3}, 0\right)$, que são os minimizadores deste problema, e os pontos $C = (0, 1)$ e $D = (0, -1)$, que são os maximizadores.

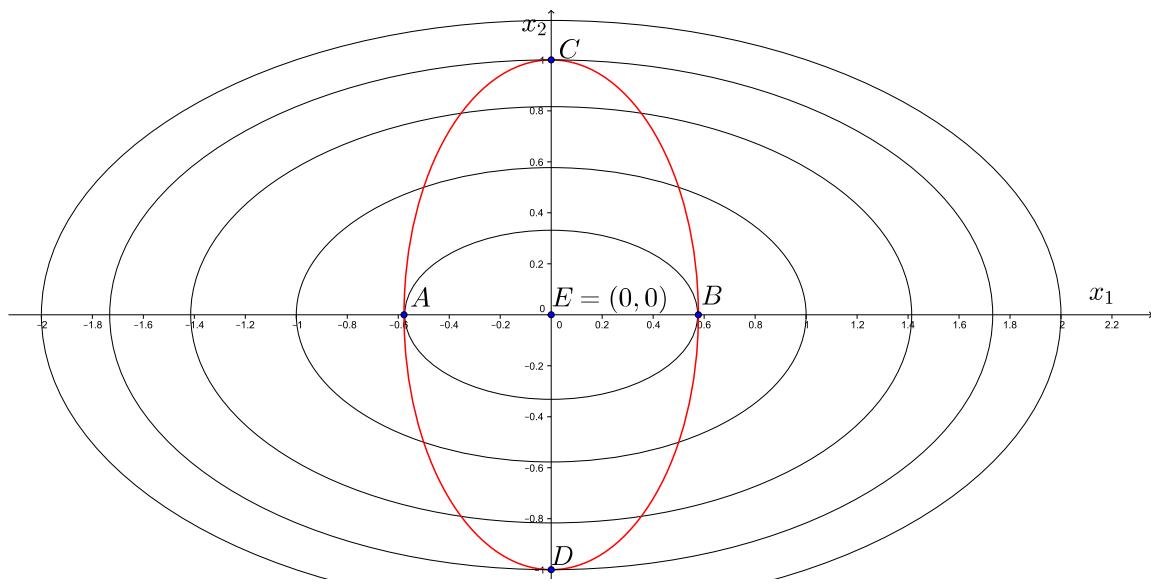


Figura 5.1: Representação geométrica do problema simétrico do Exemplo 5.0.1.

É possível observar que este problema é simétrico em relação aos eixos coordenados, pois $f(x_1, x_2) = f(-x_1, x_2) = f(x_1, -x_2) = f(-x_1, -x_2)$. O mesmo ocorre para a restrição.

Observe que a distância entre os minimizadores e o ponto $E = (0, 0)$ é igual, isto é

$$d(A, E) = d(B, E).$$

Como os pontos E , A , B estão sobre a reta $x_2 = 0$, então, pela definição 5.0.1, é possível concluir que os minimizadores são simétricos em relação ao ponto E . É correto dizer também que o ponto A é o reflexo do ponto B através do ponto E e vice-versa.

O mesmo ocorre com os maximizadores, i.e.,

$$d(C, E) = d(D, E).$$

Como os pontos E , C e D estão sobre a reta $x_1 = 0$, então o ponto C pode ser entendido como o reflexo do ponto D através do ponto E .

Observe ainda que o ponto E , que é o centro de simetria dos minimizadores e dos maximizadores, é o centro das elipses que correspondem às curvas de nível da função objetivo e é também o centro da elipse que representa a restrição.

Os testes realizados com o Exemplo 5.0.1 se refletem na Tabela 5.1.

Valores iniciais			Valores obtidos			Valor da função objetivo
x_1	x_2	λ	x_1	x_2	λ	$f(x_1, x_2)$
0	0	0	0	0	0	0
10^{-10}	0	0	0,5773	0	-0,3333	0,3333
0	10^{-10}	0	0	1	-3	3
30	-20	0	0,5773	0	-0,3333	0,3333

Tabela 5.1: Testes referentes ao Exemplo 5.0.1. Note que $\frac{\sqrt{3}}{3} \approx 0.5773$, portanto, com x_0 fora do eixo de simetria, ALGENCAN encontra um dos minimizadores.

Quando $x_0 = (0, 0)$ ou $x_0 = (0, 10^{-10})$ o ALGENCAN não consegue encontrar a solução do problema. No primeiro caso, o ALGENCAN fica estacionado e no segundo converge para um maximizador.

A seguir será estudado o que está ocorrendo nestes casos. ALGENCAN está solucionando a cada iteração o problema

$$\min L(x, \lambda, \rho) = x_1^2 + 3x_2^2 + \frac{\rho}{2} \left(3x_1^2 + x_2^2 - 1 + \frac{\lambda}{\rho} \right)^2,$$

que nada mais é do que o Método do Lagrangiano Aumentado aplicado no problema original. Com isso,

$$\nabla L(x, \lambda, \rho) = \begin{pmatrix} 2x_1 + 6x_1\rho \left(3x_1^2 + x_2^2 - 1 + \frac{\lambda}{\rho} \right) \\ 6x_2 + 2x_2\rho \left(3x_1^2 + x_2^2 - 1 + \frac{\lambda}{\rho} \right) \end{pmatrix}.$$

Mas em $x_0 = (0, 0)$ tem-se $\nabla L(x_0, \lambda, \rho) = 0$. Logo, x_0 é um ponto estacionário da função $L(x, \lambda, \rho)$ e o programa não sai deste ponto. O ALGENCAN alerta que um ponto estacionário da inviabilidade foi encontrado, porém provavelmente um ponto não factível. Observe que este resultado independe de ρ e λ .

Já os pontos iniciais escolhidos da forma $x_0 = (0, x_2)$ fazem com que os maximizadores se tornem atratores para o ALGENCAN. Isto ocorre pois, ALGENCAN escolhe como direção de descida avaliada neste ponto inicial a direção do Método de Newton aplicado ao problema irrestrito. Desta forma,

$$\nabla^2 L(x_0, \lambda, \rho) \begin{pmatrix} d_1 \\ d_2 \end{pmatrix} = -\nabla L(x_0, \lambda, \rho),$$

que resulta em

$$\begin{pmatrix} 2 + 6\rho \left(x_2^2 - 1 + \frac{\lambda}{\rho} \right) \rho & 0 \\ 0 & 6 + 2\rho \left(x_2^2 - 1 + \frac{\lambda}{\rho} \right) + 4x_2^2 \rho \end{pmatrix} \begin{pmatrix} d_1 \\ d_2 \end{pmatrix} = - \begin{pmatrix} 0 \\ 6x_2 + 2x_2 \rho \left(x_2^2 - 1 + \frac{\lambda}{\rho} \right) \end{pmatrix}.$$

Sendo assim, o vetor de descida $d = (d_1, d_2)^T$ tem $d_1 = 0$ e consequentemente esta direção está sobre a reta $x_1 = 0$, o que leva o ALGENCAN para um maximizador. Em uma análise geométrica este resultado era esperado. Pelo fato dos minimizadores estarem equidistantes do eixo de simetria, não há motivo para o ALGENCAN preferir x_1 positivo ou negativo, o que o impede de convergir para eles.

Para pontos iniciais fora do eixo x_2 , por exemplo $(30, -20)$, ALGENCAN obtém um dos minimizadores como solução.

A seguir é estudado o caso de minimizar uma função convexa, cujas curvas de nível são elipses, sujeito a uma restrição que determina um conjunto viável que também é uma elipse. Será considerado que os eixos da elipse determinada pela restrição são paralelos aos eixos das curvas de nível da função objetivo e que todas as elipses estão centradas no mesmo ponto. Além disso, é verdade que se $\alpha, \beta > 0$ então minimizar $f(x)$ sujeito a $h(x) = 0$ é equivalente a minimizar $\alpha f(x)$ sujeito a $\beta h(x) = 0$. Usando este fato e escolhendo o sistema de coordenadas adequadamente, sempre é possível escrever um problema equivalente ao problema de interesse da seguinte forma

$$\begin{aligned} & \min x_1^2 + ax_2^2 \\ & \text{sujeito a} \\ & x_1^2 + bx_2^2 - c = 0, \end{aligned} \tag{5.0.1}$$

com $a, b, c \in \mathbb{R}_+$.

Para que $x^* \in \mathbb{R}^2$ seja minimizador de (5.0.1) é necessário que x^* seja solução do seguinte sistema não linear

$$\begin{cases} \nabla f(x) + \lambda \nabla h(x) = 0 \\ h(x) = 0. \end{cases}$$

Neste caso, tem-se o sistema

$$\begin{cases} 2x_1 + 2\lambda x_1 = 0 \\ 2ax_2 + 2\lambda bx_2 = 0 \\ x_1^2 + bx_2^2 - c = 0. \end{cases}$$

Da primeira equação vem que

$$x_1 = 0 \text{ ou } \lambda = -1.$$

Se $x_1 = 0$, então $x_2 = \pm\sqrt{\frac{c}{b}}$ com $\lambda = -\frac{a}{b}$. Assim, os primeiros candidatos a mínimo são

$$\begin{aligned} P_1 &= \left(0, \sqrt{\frac{c}{b}}\right) \\ P_2 &= \left(0, -\sqrt{\frac{c}{b}}\right) \end{aligned}$$

com $f(P_1) = f(P_2) = \frac{ac}{b}$.

Se $\lambda = -1$, e $a = b$ então todo ponto viável seria solução do problema (5.0.1) e o valor da função objetivo seria igual a c em todos eles. Se $\lambda = -1$, e $a \neq b$ então $x_2 = 0$ e $x_1 = \pm\sqrt{c}$. Assim,

$$\begin{aligned} P_3 &= (\sqrt{c}, 0) \\ P_4 &= (-\sqrt{c}, 0), \end{aligned}$$

com $f(P_3) = f(P_4) = c$.

Desta forma, se $a < b$, então os minimizadores são P_1 e P_2 , pois $f(P_1) = f(P_2) < f(P_3) = f(P_4)$. Senão, os minimizadores são P_3 e P_4 .

Observação 5.0.4. O Exemplo 5.0.1 se encaixa neste caso, basta dividir a equação da restrição por 3.

A seguir outra situação onde ocorre a simetria com elipses.

Exemplo 5.0.2. Considere o problema

$$\begin{aligned} \min \quad & x_1^2 + 2x_2^2 \\ \text{sujeito a} \quad & (x_1 - 1)^2 + 3x_2^2 = 1 \end{aligned}$$

e sua representação geométrica, conforme Figura 5.2.

Neste exemplo, o ponto $x^* = (0, 0)$ é o minimizador do problema e o ponto $B = (2, 0)$ é o maximizador.

Aqui ocorre que o problema é simétrico em relação ao eixo das abscissas, pois $f(x_1, x_2) = f(x_1, -x_2)$ e ainda, $h(x_1, x_2) = h(x_1, -x_2)$.

Note que $C = (1, 0)$ é o centro de simetria entre x^* e B , pois $d(x^*, C) = d(C, B)$ e x^* , C e B estão sobre a reta $x_2 = 0$. Além disso, $C = (1, 0)$ é o centro da elipse que representa a restrição.

Para os testes feitos no Exemplo 5.0.2 foi construída a Tabela 5.2.

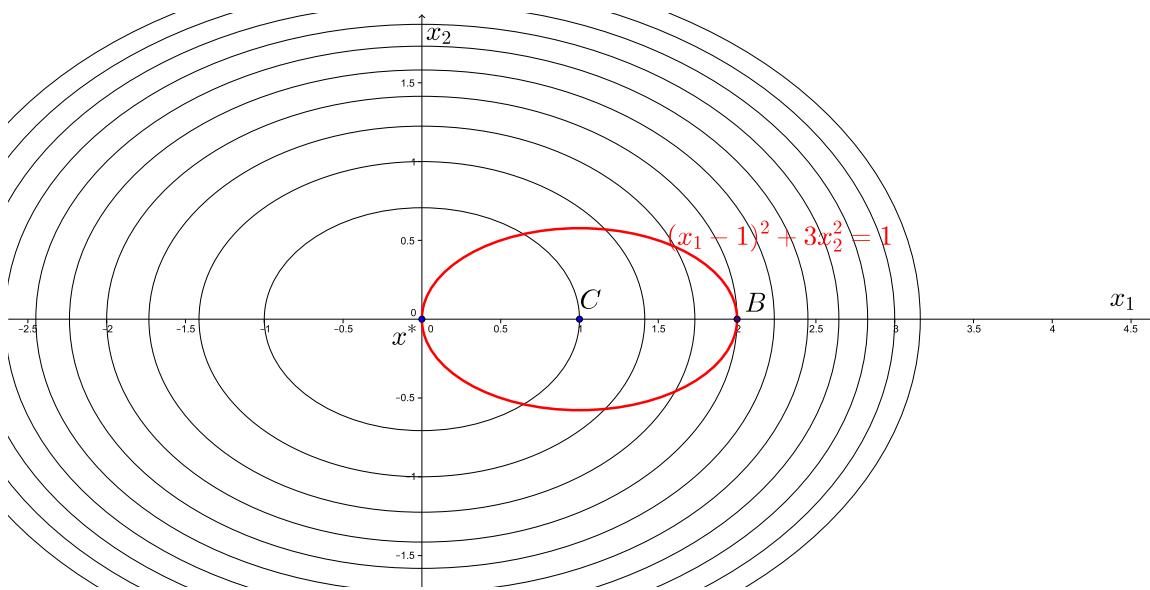


Figura 5.2: Representação geométrica do problema simétrico do Exemplo 5.0.2.

Quando $x_0 = (1, 0)$, que é o centro de simetria entre o maximizador e o minimizador, o ALGENCAN para e quando $x_0 = (1.5, 0)$ ele converge para o maximizador. Observe que no primeiro caso, o ponto não atende nem a condição de primeira ordem, isto é, não existe multiplicador de Lagrange de modo que

$$\nabla f(x_0) + \lambda \nabla h(x_0) = 0,$$

pois

$$\begin{pmatrix} 2 \\ 0 \end{pmatrix} \neq -\lambda \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

Mais que isso, o gradiente do Lagrangiano Aumentado não é nulo. Isto ocorre pois, embora inviável, $x_0 = (1, 0)$ é um ponto estacionário da inviabilidade, isto é,

$$\nabla \|h(x_0)\|^2 = 0.$$

Embora o Teorema 3.2.1 afirme que pontos deste tipo possam ser encontrados por ALGENCAN, isso só deveria acontecer se ρ fosse para infinito. Como não é isso que acontece, percebe-se que ALGENCAN tem um critério de parada extra: para por encontrar um ponto estacionário da inviabilidade. A discussão deste critério será retomada no próximo capítulo.

Valores iniciais			Valores obtidos			Valor da função objetivo
x_1	x_2	λ	x_1	x_2	λ	$f(x_1, x_2)$
1	0	0	1	0	0	1
0.5	0	0	0	0	0	0
1,5	0	0	2	0	-2	4
1,000001	0	0	0	0	0	0
1	0.1	0	0	0	0	0
1.5	0.2	0	0	0	0	0
1	0.5	0	0	0	0	0
2	0.1	0	0	0	0	0
1.9	0.1	0	0	0	0	0

Tabela 5.2: Testes referentes ao Exemplo 5.0.2.

Quando o ponto inicial é $x_0 = (1.5, 0)$, o ALGENCAN indica como solução o ponto $(2, 0)$. As razões disso são discutidas a seguir. Considerando $x_2 = 0$ e $\lambda = 0$ então, para qualquer x_1 , a direção advinda do Método de Newton para otimização aplicado ao Método do Lagrangiano Aumentado é da forma $(d_1, 0)$. Sendo assim, os iterandos sempre estarão sobre a reta $x_1 = 0$. Os pontos $(0, 0)$ e $(2, 0)$ são os únicos pontos viáveis sobre esta reta. Desta forma, se fosse considerado apenas o termo de inviabilidade na função minimizada, valeria que, com pontos iniciais no intervalo $[0, 1]$, a sequência gerada convergiria para $(0, 0)$ e, com pontos iniciais em $(1, 2]$, a sequência convergiria para o ponto $(2, 0)$. Note que quanto mais próximo de $(2, 0)$ maior seria o preço pago ao se deslocar para a esquerda, já que a medida de inviabilidade cresceria mais rapidamente. Ao acrescentar a função objetivo do problema original no subproblema, o intervalo de convergência seria deslocado, fazendo com que alguns pontos com abscissa maior do que um também convergissem para $(0, 0)$. O valor de ρ regularia o tamanho deste deslocamento. Claramente, quanto maior o valor de ρ , menor seria o deslocamento, já que o problema penalizado estaria cada vez mais parecido com o problema onde a função objetivo é desprezada. Analisando o resultado obtido por ALGENCAN para $x_0 = (1.5, 0)$ é possível perceber que com $\rho = 10$ o intervalo de convergência não teria sido deslocado além da metade do intervalo original $(1, 2]$.

No exemplo a seguir, a simetria ocorre com uma parábola representando a restrição.

Exemplo 5.0.3. Considere o problema

$$\min 4x_1^2 + 9(x_2 - 4)^2$$

sujeito a

$$x_1^2 - x_2 = 0.$$

Sua representação geométrica se dá conforme Figura 5.3.

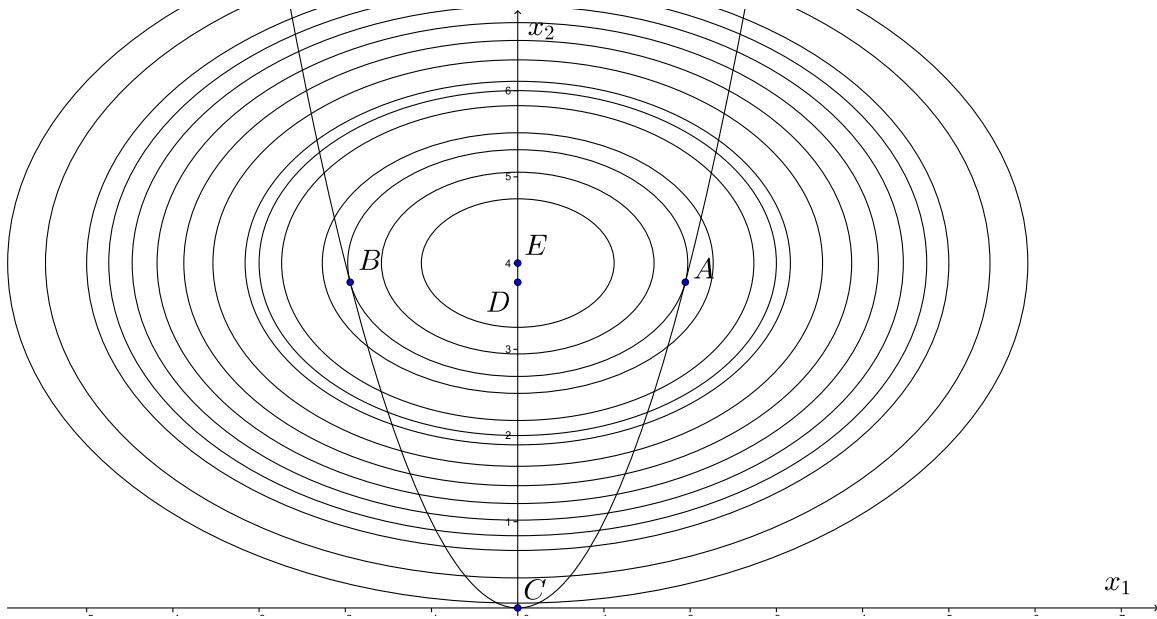


Figura 5.3: Representação geométrica do problema simétrico Exemplo 5.0.3.

Os minimizadores são $A = \left(\frac{\sqrt{34}}{3}, \frac{34}{9}\right)$ e $B = \left(-\frac{\sqrt{34}}{3}, \frac{34}{9}\right)$. Um maximizador local é o ponto $C = (0, 0)$ e o ponto $D = \left(0, \frac{34}{9}\right)$ é o centro de simetria entre os minimizadores. O ponto $E = (0, 4)$ representa o centro de simetria das curvas de nível da função objetivo.

Neste exemplo ocorre que a reta $x_1 = 0$ é o eixo de simetria do problema, pois $f(x_1, x_2) = f(-x_1, x_2)$ assim como $h(x_1, x_2) = h(-x_1, x_2)$.

Os resultados dos testes realizados para o Exemplo 5.0.3 podem ser acompanhados na Tabela 5.3.

Quando $x_0 = (0, 0)$ ou $x_0 = (0, 4)$ o ALGENCAN converge para o maximizador. De modo geral, em pontos da forma $x_0 = (0, x_2)$ o ALGENCAN convergirá para o maximizador. O motivo disso ocorrer é semelhante ao exemplo anterior. Ao aplicar o Método de Newton no problema

Valores iniciais			Valores obtidos			Valor da função objetivo
x_1	x_2	λ	x_1	x_2	λ	$f(x_1, x_2)$
0	0	0	0	0	-72	144
0	4	0	0	0	-72	144
2	4	0	1.9436	3.7777	-4	15.56
10^{-3}	4	0	1.9436	3.7777	-4	15.56
10^{-3}	0	0	1.9436	3.7777	-4	15.56
-10^{-3}	0	0	1.9436	3.7777	-4	15.56
0	$\frac{34}{9}$	0	0	0	-72	144
0	10^{-3}	0	0	0	-72	144
0	-10^{-3}	0	0	0	-72	144

Tabela 5.3: Testes referentes ao Exemplo 5.0.3. Note que $\frac{34}{9} \approx 3.7777$ e $\frac{\sqrt{34}}{9} \approx 1,9436$, portanto, com x_0 fora do eixo de simetria, ALGENCAN encontra um dos minimizadores do problema.

irrestrito, obtém-se a direção $d = (d_1, d_2)^T$, onde $d_1 = 0$. Porém, quando há uma perturbação no valor de x_1 o programa encontra a solução desejada.

O Exemplo 5.0.4 a seguir trás como restrição que os pontos viáveis estejam sobre uma hipérbole.

Exemplo 5.0.4. Considere o problema

$$\begin{aligned} & \min (x_1 - 7)^2 + 4x_2^2 + 3 \\ & \text{sujeito a} \\ & x_1^2 - x_2^2 - 1 = 0, \end{aligned}$$

conforme a Figura 5.4.

Neste exemplo os minimizadores são $A = \left(\frac{7}{5}, \frac{2\sqrt{6}}{5}\right)$, $B = \left(\frac{7}{5}, -\frac{2\sqrt{6}}{5}\right)$, $D = (1, 0)$ e $E = (-1, 0)$. Os pontos A e B são minimizadores globais do problema.

O eixo de simetria do problema é $x_2 = 0$, pois observe que $f(x_1, x_2) = f(x_1, -x_2)$ e $h(x_1, x_2) = h(x_1, -x_2)$. O centro de simetria dos minimizadores globais é $O = \left(\frac{7}{5}, 0\right)$ e o centro de simetria dos outros dois minimizadores é $O' = (0, 0)$.

Os testes feitos no Exemplo 5.0.4 resultaram na Tabela 5.4.

Novamente o ALGENCAN para em um ponto inicial que não atende a condição necessária de primeira ordem, neste caso, quando $x_0 = (0, 0)$. Isto ocorre pois $x_0 = (0, 0)$ é um ponto estacionário da inviabilidade.

Para pontos da forma $x_0 = (x_1, 0)$ com $x_1 \neq 0$ o ALGENCAN converge para os minimizadores locais. Quando há uma perturbação na coordenada x_2 o programa converge para os minimizadores globais.

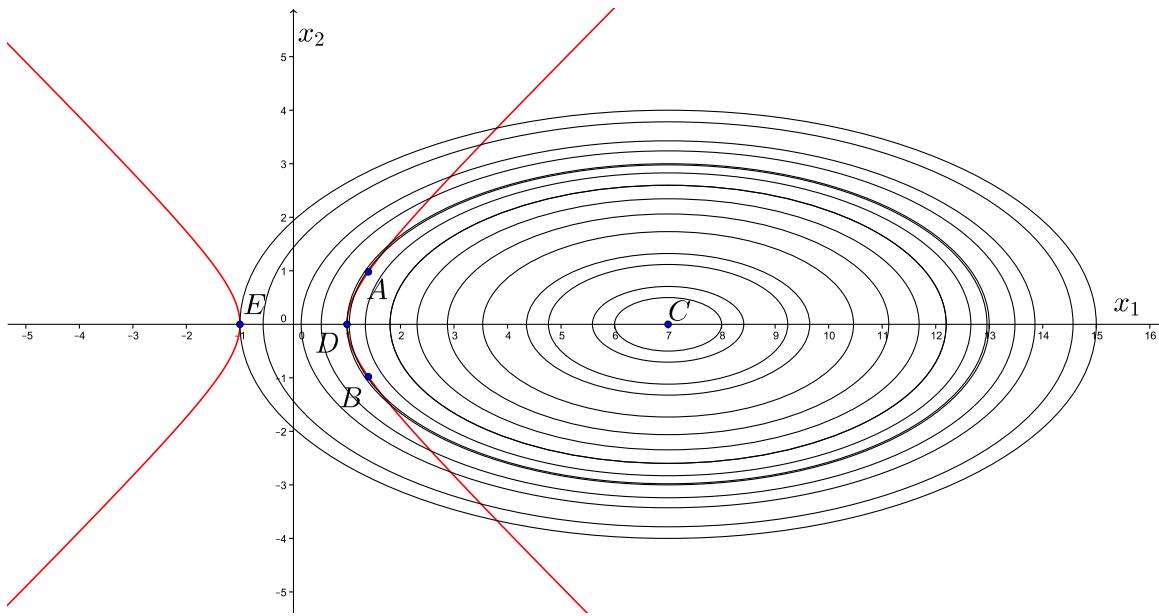


Figura 5.4: Representação geométrica do problema simétrico do Exemplo 5.0.4.

O Exemplo 5.0.5 apresenta mais um caso onde ocorre simetria ainda com hipérbole na restrição.

Exemplo 5.0.5. Considere o problema

$$\begin{aligned} \min \quad & 3x_1^2 + 4x_1x_2 + 3x_2^2 - 20x_1 - 20x_2 \\ \text{sujeito a} \quad & x_1x_2 - 5 = 0. \end{aligned}$$

A Figura 5.5 ilustra este problema. Geometricamente é fácil ver que os minimizadores são $A = (\sqrt{5}, \sqrt{5})$ e $B = (-\sqrt{5}, -\sqrt{5})$. O ponto A é o minimizador global. Neste caso, a reta $x_2 = x_1$ é o eixo de simetria do problema e o ponto $O = (0, 0)$ é o centro de simetria dos minimizadores.

Para os testes relativos ao Exemplo 5.0.5 foi construída a Tabela 5.5.

Quando $x_0 = (0, 0)$ o ALGENCAN fica estacionado em um ponto inicial que não atende a condição necessária de primeira ordem. Novamente, isto ocorre pois $x_0 = (0, 0)$ é um ponto estacionário da inviabilidade. Para pontos iniciais cujas coordenadas não são nulas, ALGENCAN converge para os minimizadores, seja o local ou o global.

Ainda com a mesma função objetivo do exemplo anterior, o Exemplo 5.0.6 é um caso específico e importante de problemas estudados em otimização não linear chamados de Problemas de

Valores iniciais			Valores obtidos			Valor da função objetivo
x_1	x_2	λ	x_1	x_2	λ	$f(x_1, x_2)$
0	0	0	0	0	0	52
7	0	0	1	0	6	39
1.4	0	0	1	0	6	39
-2	3	0	1.4	0.9797	4	38.2
0	1	0	1.4	0.9797	4	38.2
-0,5	0	0	-1	0	-8	67
3	0	0	1	0	6	39
3	10^{-4}	0	1.4	0.9797	4	38.2

Tabela 5.4: Testes referentes ao Exemplo 5.0.4. Note que $\frac{2\sqrt{6}}{5} \approx 0.9797$ e $\frac{7}{5} = 1.4$ portanto, com x_0 fora do eixo de simetria, ALGENCAN encontra um dos minimizadores.

Valores iniciais			Valores obtidos			Valor da função objetivo
x_1	x_2	λ	x_1	x_2	λ	$f(x_1, x_2)$
0	0	0	0	0	0	0
2	3	0	2.2360	2.2360	-1.0557	-39.44
-4	-4	0	-2.2360	-2.2360	-1.8944	139.4
3	5	0	2.2360	2.2360	-1.0557	-39.44
-2	2	0	2.2360	2.2360	-1.0557	-39.44

Tabela 5.5: Testes referentes ao Exemplo 5.0.5. Note que $\sqrt{5} \approx 2.2360$ portanto, com x_0 fora do eixo de simetria, ALGENCAN encontra um dos minimizadores.

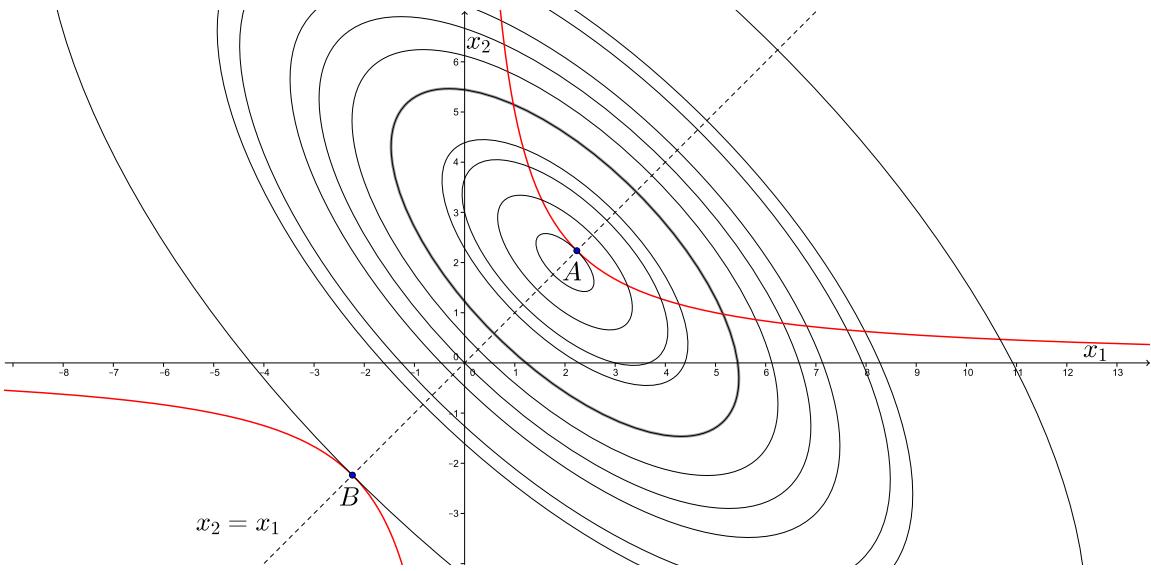


Figura 5.5: Representação geométrica do problema simétrico do Exemplo 5.0.5.

Complementaridade, conforme pode ser visto em [4]. Estes problemas aparecem em situações que envolvem a modelagem com restrições do tipo “uma coisa ou outra”. Um exemplo disso ocorre em otimização com restrições de desigualdade do tipo $g(x) \leq 0$ com $g : \mathbb{R}^n \rightarrow \mathbb{R}^p$. Nesta situação cada multiplicador de Lagrange só é diferente de zero se for associado a uma restrição satisfeita como igualdade. Ou seja, deve-se ter $g_i(x)\lambda_i = 0$, o que significa que $g_i(x) = 0$ ou $\lambda_i = 0$, $i = 1, \dots, p$.

Exemplo 5.0.6. Considere o problema

$$\begin{aligned} & \min 3x_1^2 + 4x_1x_2 + 3x_2^2 - 20x_1 - 20x_2 \\ & \text{sujeito a} \\ & x_1x_2 = 0, \end{aligned}$$

conforme Figura 5.6.

Os minimizadores são $A = \left(\frac{10}{3}, 0\right)$ e $B = \left(0, \frac{10}{3}\right)$. O eixo de simetria do problema é a reta $x_1 = x_2$ e o centro de simetria entre os minimizadores é o ponto $D = \left(\frac{5}{3}, \frac{5}{3}\right)$.

Para os testes relativos ao Exemplo 5.0.6 a Tabela 5.6 foi construída.

Para pontos iniciais sobre o eixo de simetria, o ALGENCAN falha em encontrar o minimizador. Nesses pontos o ALGENCAN informa erroneamente que encontrou uma solução. Se fosse exigido maior precisão nos critérios de parada de ALGENCAN, seria possível ver que, para estes pontos

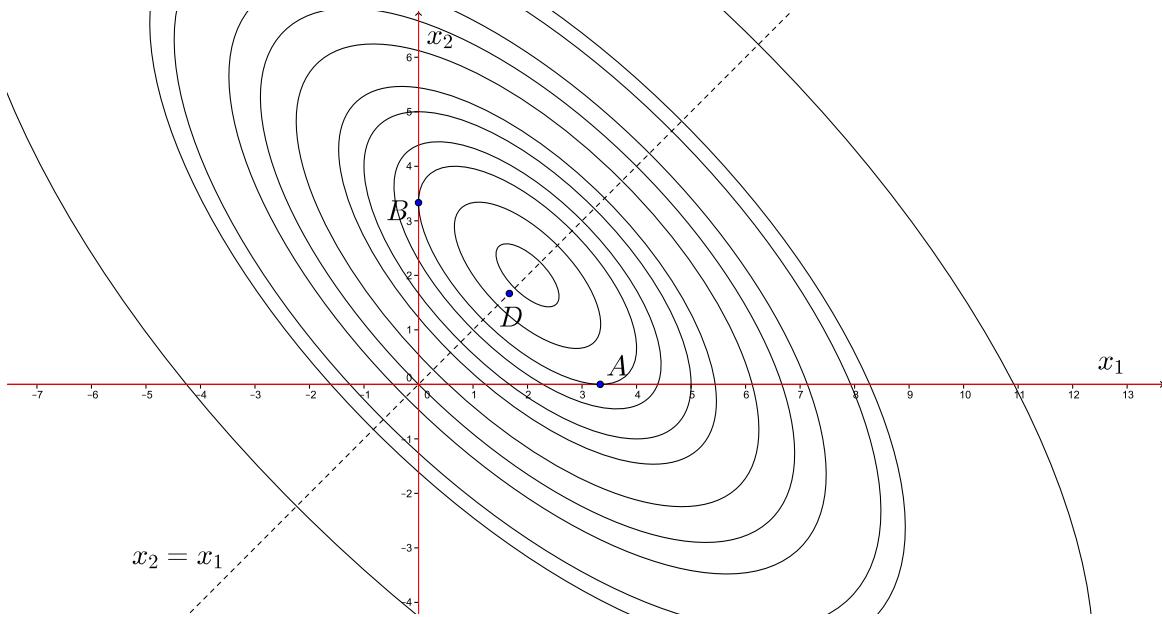


Figura 5.6: Representação geométrica do problema simétrico do Exemplo 5.0.6.

iniciais, a solução encontrada estaria cada vez mais próxima do ponto $(0, 0)$. Entretanto o ponto $(0, 0)$ é viável mas não é KKT para o problema. Isso não contradiz o Teorema (3.2.2) porque o ponto $(0, 0)$ não é um ponto regular. Caso o ponto inicial não esteja sobre a reta $x_1 = x_2$, então o ALGENCAN obtém a solução ótima.

Valores iniciais			Valores obtidos			Valor da função objetivo	
x_1	x_2	λ	x_1	x_2	λ	$f(x_1, x_2)$	
0	0	0	7.45×10^{-5}	7.45×10^{-5}	2.68×10^5	-0.00298	
2	0	0	3.33	0	2	-33.33	
-1	3	0	0	3.33...	2	-33.33	
1	1	0	8.46×10^{-5}	8.46×10^{-5}	2.36×10^5	-0.00384	
-1	1	0	0	3.33	2	-33.33	

Tabela 5.6: Testes referentes ao Exemplo 5.0.6. Note que $\frac{10}{3} \approx 3.33$ portanto, com x_0 fora do eixo de simetria, ALGENCAN encontra um dos minimizadores.

Capítulo 6

Alterações no ALGENCAN

No capítulo anterior foram vistos alguns problemas simétricos específicos de otimização não linear que acarretam dificuldades para o ALGENCAN. Neste capítulo, propõem-se alternativas para sanar estas dificuldades.

A primeira ideia vem da observação de que fora do eixo de simetria, geralmente, ALGENCAN tem um bom desempenho. Desta forma, uma das propostas é evitar começar em pontos que estejam na região problemática.

Motivado por situações onde não existem pontos factíveis, como a restrição é $x^2 + 1 = 0$, por exemplo, então o Agencan testa e usa como critério de parada a inviabilidade das restrições. Isso é feito na tentativa de acelerar a identificação de problemas inviáveis, evitando que ρ vá para infinito e antecipando o processo se um ponto é estacionário da soma dos quadrados das restrições. De modo simples, dado um $\tilde{x} \in \mathbb{R}^n$ tal que $\|h(\tilde{x})\| > \varepsilon$ se

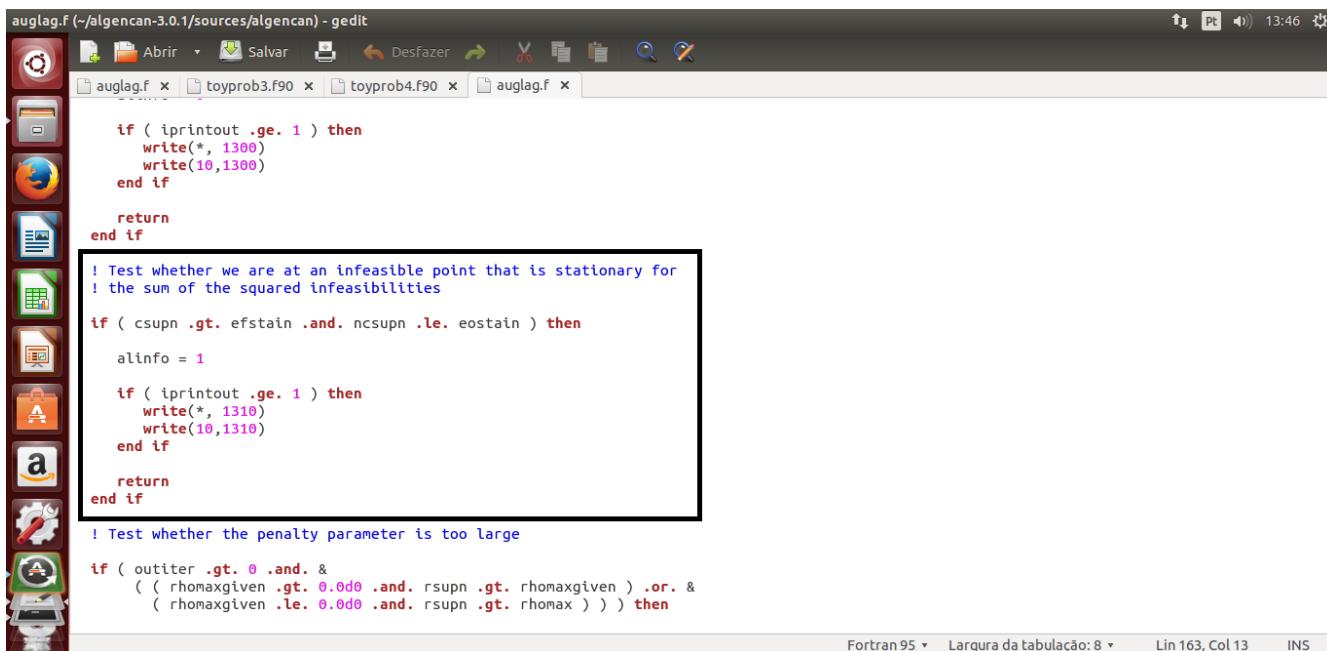
$$\nabla \left(h_1(\tilde{x})^2 + h_2(\tilde{x})^2 + \dots + h_m(\tilde{x})^2 \right) \leq \varepsilon,$$

para um ε pequeno, então o ALGENCAN para em \tilde{x} . Neste caso, ALGENCAN exibe como resposta e mostra em sua tela de saída a mensagem que um ponto estacionário da inviabilidade foi encontrado, porém, provavelmente um ponto não viável.

Este critério se encontra no arquivo “auglag.f” na pasta *sources*, dentro de *algencan-3.0.0*, conforme mostra Figura 6.1. Porém, pode ser modificado em **toyprob.f90** com as variáveis “efstain” e “eostain”. Ver [3], página 117.

Assim, nos exemplos onde o ponto inicial era estacionário da inviabilidade, o ALGENCAN fica parado mesmo que o gradiente do subproblema seja não nulo.

Cada item abaixo representa um tipo de modificação feita no ALGENCAN, sempre com o objetivo de sanar as dificuldades encontradas pelo programa ou, pelo menos, minimizá-las.



```

auglag.f (~/algenkan-3.0.1/sources/algenkan) - gedit
Abrir Salvar Desfazer Desfazer Novo Busca
auglag.f x toyprob3.f90 x toyprob4.f90 x auglag.f x
if ( iprintout .ge. 1 ) then
    write(*, 1300)
    write(10,1300)
end if

return
end if

! Test whether we are at an infeasible point that is stationary for
! the sum of the squared infeasibilities

if ( csupn .gt. efstain .and. ncsupn .le. eostain ) then
    alinfo = 1

    if ( iprintout .ge. 1 ) then
        write(*, 1310)
        write(10,1310)
    end if

    return
end if

! Test whether the penalty parameter is too large

if ( outiter .gt. 0 .and. (
    ( rhomaxgiven .gt. 0.0d0 .and. rsupn .gt. rhomaxgiven ) .or. &
    ( rhomaxgiven .le. 0.0d0 .and. rsupn .gt. rhomax ) ) ) then

```

Fortran 95 ▾ Largura da tabulação: 8 ▾ Lin 163, Col 13 INS

Figura 6.1: Critérios de parada de ALGENCAN em pontos estacionários da inviabilidade.

Perturbação do ponto inicial A ideia desta modificação é de sair da região problemática para o ALGENCAN, onde ele não tem um bom desempenho. O objetivo desta modificação é bastante simples: perturbar um pouco o ponto inicial fornecido pelo usuário para que este saia da região ruim, que é o eixo de simetria do problema. A perturbação se deu no arquivo “algenkan.f” e tem suas linhas de comando conforme Figura 6.2.

Observe que a perturbação, como na figura, se deu na ordem de 10^{-8} . Outros valores também poderiam ser utilizados e de fato foram testados os valores 10^{-8} , 10^{-7} e 10^{-4} . Primeiramente ALGENCAN toma um número aleatório entre 0 e 1 com o comando “drand(seed)”. Depois, é subtraído 1 do dobro deste número aleatório gerando um número entre -1 e 1. Em seguida, multiplica o resultado obtido com o máximo entre 1 e o módulo do ponto inicial fornecido pelo usuário para obter uma perturbação proporcional às dimensões do problema. Depois, multiplica por 10^{-8} , para que a perturbação seja pequena, e somada com o ponto inicial dado pelo usuário. Isto é feito tantas vezes quanto a quantidade de variáveis do problema, isto é, a modificação garante que as coordenadas $x_1, x_2, x_3, \dots, x_n$ serão perturbadas. Por fim, a perturbação feita é projetada nos valores de máximo e mínimo, “uorig(i)” e “lorig(i)” respectivamente, também fornecidos.

Exclusão do teste de inviabilidade A ideia desta modificação é que o ALGENCAN não realize

```

! =====
! Initialization
! =====

! SET A PERTURBATIONS OF X (INITIAL GUESS)

seed = 123456.0d0

do i = 1,norig
    xorig(i) = xorig(i) + 1.0d-8 * ( 2.0d0 * drand(seed) - 1.0d0 ) * &
        max( 1.0d0, abs( xorig(i) ) )
    xorig(i) = max( lorig(i), min( xorig(i), uorig(i) ) )
end do

! Set machine-dependent constants (module machconst)
call inimachconst()

! Output control (module outtyp)
call iniouttyp()

! =====
! Set default values in 'prob' modules (probgiven, problvl[s|t|u|v])
! =====

```

Figura 6.2: Perturbação no ponto inicial dado pelo usuário.

mais o teste da inviabilidade das restrições e consequentemente não pare em pontos que são estacionários da inviabilidade mas que não são estacionários.

Teste de inviabilidade modificado O conceito básico desta modificação é que quando o ALGENCAN, em uma iteração qualquer, se deparar com um ponto estacionário da inviabilidade ele não mais pare mas sim atualize o valor do termo de penalidade com um valor elevado e continue a fazer as iterações.

Se o problema realmente não tem pontos factíveis, então este critério original é realmente muito bom, pois evita que o ALGENCAN procure por pontos que não existem. Caso contrário o critério é ruim, pois o programa parará em um ponto que não é factível sendo que a restrição pode ser satisfeita. Em resumo, pelo que está posto, a existência deste critério é ruim para uma das duas situações. Assim, a intenção é deixar o critério, mas que este não seja de parada mas sim de atualização do termo de penalidade, acelerando a busca de pontos viáveis.

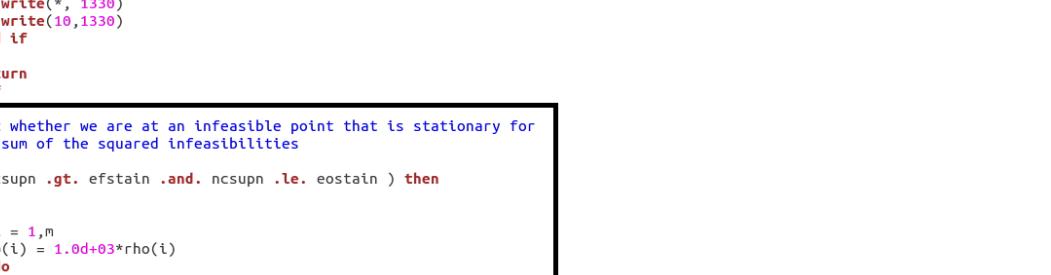
Imagine que o problema de minimização não admite pontos viáveis. Assim, quando o ALGENCAN encontrar um ponto estacionário da inviabilidade ele atualizará o termo de penalidade para um valor grande, com o intuito de procurar pontos factíveis. Como eles não existem, pelo critério do tamanho do termo de penalidade, o ALGENCAN parará. Agora

suponha uma situação oposta, ou seja, o problema de minimização admite pontos factíveis. Se o ponto inicial for estacionário da inviabilidade mas o gradiente do subproblema for não nulo, então o ALGENCAN não mais parará mas sim atualizará o termo de penalidade para um valor grande de modo a buscar um ponto factível.

Importante também ressaltar a necessidade da troca das ordens dos critérios. É necessário que o critério de parada que verifica o tamanho do termo de penalidade venha antes do critério de atualização. Se não for assim, quando atualizado o termo de penalidade o ALGENCAN já poderia parar pois acusaria um termo muito grande.

A modificação se deu conforme Figura 6.3. Nela, como é possível observar, quando um ponto é classificado como estacionário da inviabilidade pelo teste, então faz-se

$$\rho = 10^3 \rho_*$$



The screenshot shows a Gedit text editor window with a black background. The title bar reads "auglag.f (~/algencan-3.0.3/sources/algencan) - gedit". The status bar at the bottom right shows "20:26" and "Fortran 95". The code in the editor is as follows:

```
writeln(*, 1330)
writeln(10,1330)
end if

return
end if

! Test whether we are at an infeasible point that is stationary for
! the sum of the squared infeasibilities

if ( csupn .gt. efstain .and. ncsupn .le. eostain ) then

    do i = 1,m
        rho(i) = 1.0d+03*rho(i)
    end do
    !alinfo = 1

    !if ( iprintout .ge. 1 ) then
    !    writeln(*, 1310)
    !    writeln(10,1310)
    !end if

    !return
end if

=====
! Near the solution, try to solve the KKT system by Newton's method
=====
```

Figura 6.3: Modificação no teste de inviabilidade.

Perturbação do ponto inicial com a exclusão do teste de inviabilidade Esta modificação consiste em perturbar o ponto inicial fornecido pelo usuário e retirar o critério de parada que consiste em encerrar as iterações se um ponto é estacionário da inviabilidade.

Perturbação do ponto inicial com o teste de inviabilidade modificado Esta alteração tem

por objetivo unir a perturbação feita no ponto inicial informada pelo usuário com a modificação no critério de parada, explicada anteriormente.

Dadas estas modificações e a partir do ALGENCAN original, foram construídos cinco novos algoritmos. Então, no total, são seis algoritmos baseados no ALGENCAN. São eles:

A0 para o algoritmo original;

A1 para o algoritmo com a perturbação do ponto inicial;

A2 para o algoritmo sem o critério de parada;

A3 para o algoritmo com o critério de parada modificado;

A4 para o algoritmo com a perturbação e sem o critério de parada;

A5 para o algoritmo com a perturbação e com o critério de parada modificado.

Observação 6.0.5. A partir daqui serão usadas as notações descritas acima para se referir a um determinado algoritmo.

A motivação para a construção destes algoritmos é obter um desempenho melhor em problemas simétricos em comparação ao original. De fato, para os exemplos exibidos no Capítulo 5, as modificações foram bem sucedidas para superar as situações a que se propõe. Partindo dos pontos onde o ALGENCAN original (A0) apresentava um mau desempenho, foi verificado o comportamento de cada algoritmo modificado. Para cada algoritmo e cada exemplo são apresentados na Tabela 6.1 x_0 e x^* , respectivamente, o ponto inicial considerado e a solução encontrada. Na Tabela 6.2 é possível verificar quais algoritmos encontraram a solução de cada problema.

	5.0.1	5.0.2	5.0.3	5.0.4	5.0.5	5.0.6
x_0	(0, 0)	(1, 0)	(0, 0)	(0, 0)	(0, 0)	(0, 0)
A0	(0, 0)	(1, 0)	(0, 0)	(0, 0)	(0, 0)	(0, 0)
A1	(0.58, 0)	(0, 0)	(1.94, 3.78)	(1.4, 0.98)	(2.24, 2.24)	(3.33, 0)
A2	(0, 0)	(0, 0)	(0, 0)	(1.4, 0.98)	(2.24, 2.24)	(3.33, 0)
A3	(0, 0)	(0, 0)	(0, 0)	(1.4, 0.98)	(2.24, 2.24)	(3.33, 0)
A4	(0.58, 0)	(0, 0)	(1.94, 3.78)	(1.4, 0.98)	(2.24, 2.24)	(3.33, 0)
A5	(0.58, 0)	(0, 0)	(1.94, 3.78)	(1.4, 0.98)	(2.24, 2.24)	(3.33, 0)

Tabela 6.1: Pontos iniciais tomados e as soluções encontradas com os algoritmos para cada exemplo.

Observe que, entre os algoritmos modificados, apenas A2 e A3 não resolveram todos os problemas nos exemplos propostos. Os únicos problemas não resolvidos por estes algoritmos foram os

Algoritmos	Exemplos					
	5.0.1	5.0.2	5.0.3	5.0.4	5.0.5	5.0.6
A0	não	não	não	não	não	não
A1	sim	sim	sim	sim	sim	sim
A2	não	sim	não	sim	sim	sim
A3	não	sim	não	sim	sim	sim
A4	sim	sim	sim	sim	sim	sim
A5	sim	sim	sim	sim	sim	sim

Tabela 6.2: Algoritmos modificados testados nos exemplos de problemas simétricos do Capítulo 5. A indicação “sim” na tabela significa que a solução foi encontrada e “não” significa que a solução não foi encontrada.

exemplos 5.0.1 e 5.0.3. A falha do ALGENCAN original nestes problemas não estava relacionada com a parada em pontos estacionários da inviabilidade e, portanto, realmente não era esperado que os algoritmos A2 e A3 fossem superiores a A0.

Com base nesses resultados, será verificado se este comportamento se repete em uma coleção padrão de problemas gerais de otimização. Espera-se assim possibilitar ao usuário uma nova versão superior ou, pelo menos, tão competitiva quanto a do algoritmo já disponibilizado.

6.1 Comparação entre os algoritmos na coleção CUTEst

Com o intuito de comparar o desempenho dos seis algoritmos (o original mais os cinco criados) de modo satisfatório utiliza-se neste trabalho a principal biblioteca de problemas de otimização não linear suave: a coleção CUTEst [5].

O CUTEst consiste em 1158 problemas de otimização com e sem restrições de variadas dimensões. Esta biblioteca de problemas é muito utilizada para testes em algoritmos que visam solucionar problemas de otimização, como neste caso. Por isso se mostra de essencial importância para este trabalho. Embora a motivação para os algoritmos propostos tenha sido originada de problemas simétricos com restrições de igualdade, todos os problemas da CUTEst foram utilizados nos experimentos computacionais. Isso foi feito assim porque o intuito com estes testes é verificar o comportamento das modificações propostas em problemas gerais. Uma instrução de como baixar e instalar a coleção CUTEst no Linux está em [5].

Para comparar os seis algoritmos foram construídos gráficos do tipo perfil de desempenho [6]. Como já foi dito, A1, A4 e A5 são modificações de ALGENCAN que passaram por perturbações em seus pontos iniciais. Foram consideradas três perturbações, uma na ordem de 10^{-4} , outra na ordem de 10^{-7} e a terceira na ordem de 10^{-8} . Por consequência direta, este trabalho possui

três gráficos comparativos, um para cada perturbação. O critério de inviabilidade adotado neste trabalho é na ordem de 10^{-8} .

Neste trabalho, um problema da coleção CUTEST é considerado resolvido por um algoritmo quando o valor da função objetivo deste se encontra a uma diferença de 1% do menor valor encontrado por algum algoritmo. Obviamente, este menor valor deverá advir de algum problema considerado viável.

Gráficos do tipo perfil de desempenho (ver [6]), usados para comparar *softwares*, foram confecionados da seguinte forma: no eixo das abscissas tem-se uma razão entre o tempo que o algoritmo demorou para solucionar um determinado problema pelo menor tempo de todos os algoritmos que resolveram este problema. O eixo das ordenadas está associado à quantidade de problemas resolvidos até uma determinada razão. Para uma melhor visualização dos resultados, o eixo das ordenadas é escalado de forma a medir a taxa de problemas resolvidos. Isso é feito dividindo o número de problemas resolvidos pelo número total de problemas. Foi considerada neste trabalho uma razão de valor até 3, isto é, problemas que um algoritmo demorou até 3 vezes mais para ser resolvido do que o algoritmo que o resolveu mais rapidamente.

Os testes foram realizados em um computador com 8GB de memória RAM, 1TB de Disco Rígido e sistema operacional Windows 8. Conforme já mencionado anteriormente, o sistema Linux foi virtualizado nesta máquina. Foi utilizado o *gfortran* na sua versão 4.8.2 para a compilação dos arquivos em que se encontram os problemas de otimização. Foi permitido o tempo máximo de um minuto para a resolução de cada problema do CUTEST. Se ao fim deste tempo nenhum dos critérios de parada de ALGENCAN tenham sido atingidos, a execução foi interrompida e foi considerado como solução obtida o último ponto encontrado por ALGENCAN.

Observação 6.1.1. O ALGENCAN retorna na saída o tempo que ele demorou para resolver o problema com duas casas decimais, por isso, alguns problemas retornaram zero segundos para serem solucionados, o que na prática é absurdo. Quando zero segundos foi classificado como o menor dos tempos, então foi considerado 0,005 segundos como tempo de resolução. A quantidade de casas decimais que o ALGENCAN retorna para o tempo pode ser modificada no arquivo “*auglag.f*”. Acredita-se que considerar mais casas decimais no tempo não tenha grandes implicações no resultado das comparações. De toda forma, esta verificação pode ser feita em um trabalho futuro.

As Figuras 6.4, 6.5 e 6.6 referem-se à comparação feita quando a perturbação é da ordem de 10^{-4} , 10^{-7} e 10^{-8} respectivamente. Estes valores foram escolhidos por causa do critério de parada do ALGENCAN, que usa como tolerância 10^{-8} .

Como as alterações propostas são bem pontuais, a expectativa era que elas só tivessem impacto

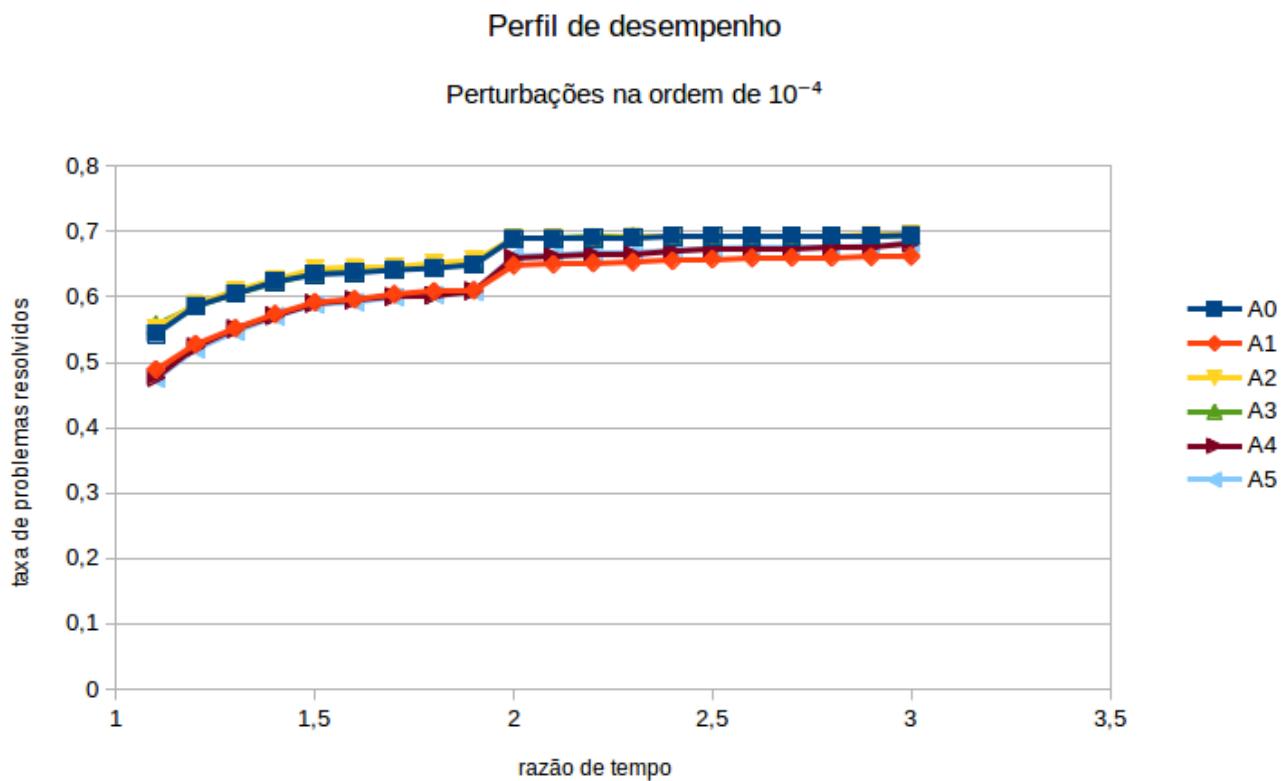


Figura 6.4: Perfil de desempenho com A1, A4 e A5 perturbados na ordem de 10^{-4} .

num número reduzido de problemas. Desta forma, era esperado que os gráficos em relação aos seis algoritmos fossem muito semelhantes e que estivessem praticamente sobrepostos. Entretanto é visível que os gráficos dos algoritmos com perturbações são praticamente um deslocamento dos outros três. Isso dá a impressão que um grupo de problemas deixou de ser resolvido com esta modificação enquanto nos outros não houve grande impacto.

Observando os gráficos feitos é notório que A0, A2 e A3 se destacam de A1, A4 e A5, ou seja, são mais eficientes e mais robustos. Destes, A3 tem uma pequena vantagem em termos de eficiência. Na Tabela 6.3 são apresentados os números referentes à eficiência e à robustez do algoritmos. A eficiência se refere ao número de problemas em que os algoritmos encontram a solução do problema no menor tempo possível e a robustez se refere ao número de problemas resolvidos após três vezes o tempo do algoritmo mais eficiente.

Como já foi posto, os algoritmos A2 e A3 alteram de alguma forma o critério de parada do ponto estacionário da inviabilidade. O A2 exclui este critério e A3 o modifica não mais encerrando

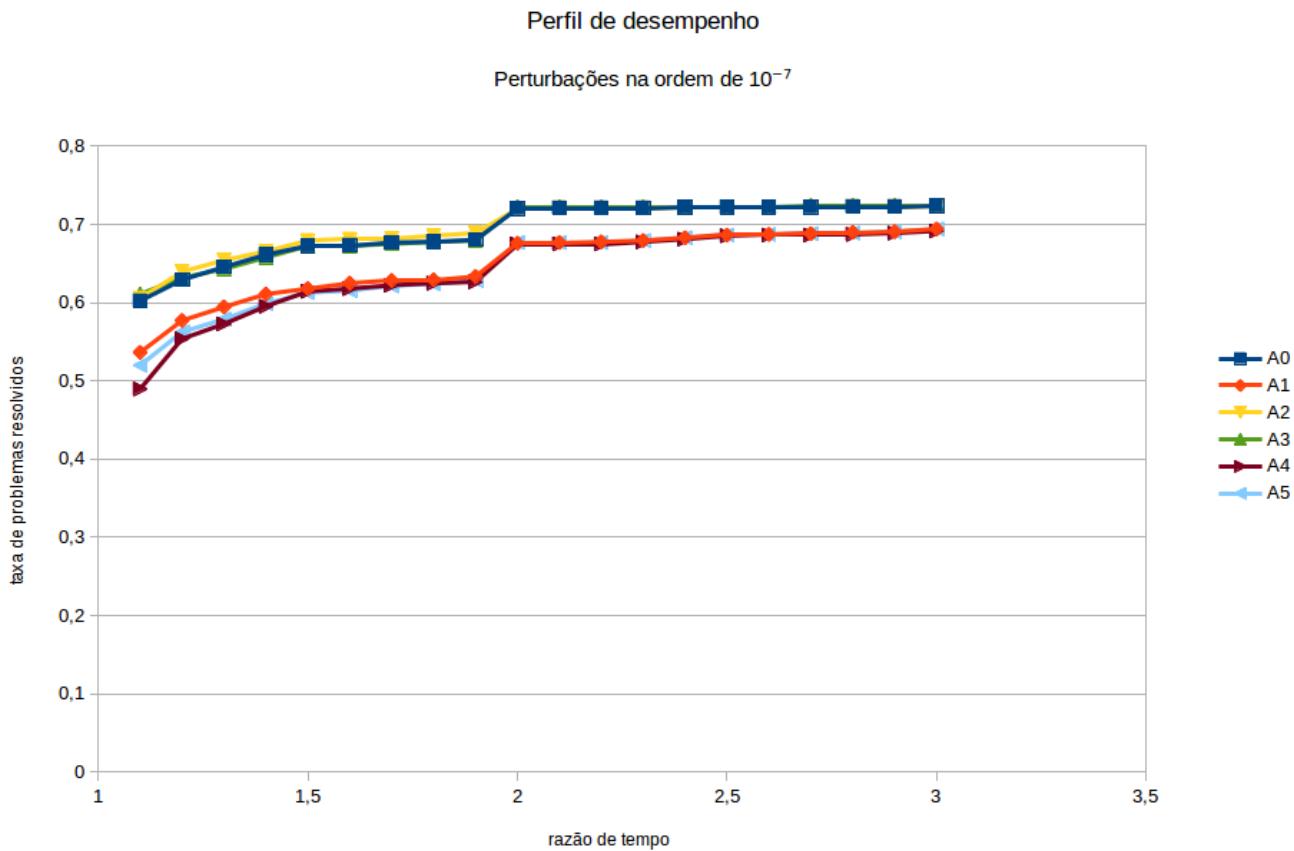


Figura 6.5: Perfil de desempenho com A1, A4 e A5 perturbados na ordem de 10^{-7} .

o processo, mas sim atualizando o termo de penalidade. Obviamente, com estes resultados apresentados é questionável a necessidade deste critério no ALGENCAN, uma vez que A3 revelou uma pequena superioridade a A0 e A2.

Lembre-se que A1, A4 e A5 são os algoritmos que foram submetidos a perturbações em seus pontos iniciais. Apesar de serem eficientes para solucionar problemas simétricos de otimização não linear, não se apresentam eficazes para problemas variados de otimização.

Para uma análise um pouco mais detalhada do porquê do mau comportamento de A1, A4 e A5, foi construída a Tabela 6.4. Foram tomados cinco problemas do CUTEST onde A1, A4 e A5 não acharam a solução enquanto A0, A2 e A3 sim. Por simplicidade são apresentados na Tabela 6.4 apenas os resultados de A0 e A1, já que os resultados dos demais são semelhantes. Esta tabela foi feita sobre os algoritmos que sofreram perturbação na ordem de 10^{-8} .

A Tabela 6.4 indica que os algoritmos apresentaram valores de função objetivo praticamente

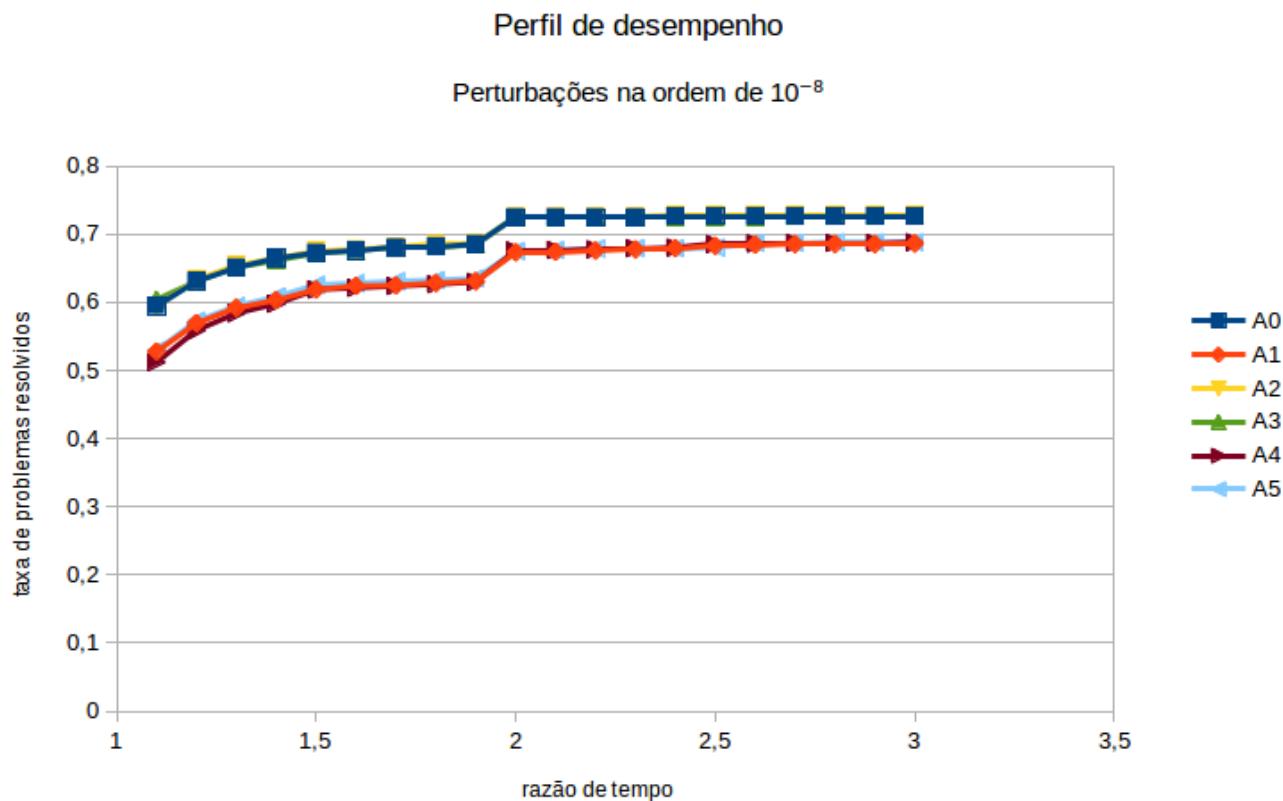


Figura 6.6: Perfil de desempenho com A1, A4 e A5 perturbados na ordem de 10^{-8} .

iguais. Acontece que alguns problemas foram considerados inviáveis. Observe, por exemplo, o problema “HAIFAL”. Nele, o valor de f é $-12,8$ para A0 e $-12,79999$ para A1, mas pelo critério adotado de inviabilidade, este problema é classificado como inviável para o A1 e viável para A0. Este fato nos leva a questionar se a solução encontrada pelos dois algoritmos foi praticamente a mesma, mas com valor das restrições diferente. Os algoritmos A4 e A5 também indicam a solução deste problema como inviável. Os algoritmos que indicaram a inviabilidade da solução são justamente os algoritmos que sofreram perturbações em seus pontos iniciais.

Uma possível explicação para o desempenho dos algoritmos que sofreram perturbação não ter sido tão satisfatório é que talvez a família de problemas não resolvidos possua certa instabilidade nas restrições. Desta forma, os problemas deixam de ser resolvidos quando é alterado o bom ponto inicial sugerido no CUTEst.

Os problemas “A0NNDNSL” e “A2NSDSSL” se originam da reformulação de problemas de complementaridade. Já o problema “DEGENLPB” está associado à ciclagem em programação

	A0	A1	A2	A3	A4	A5
Eficiência	495	485	507	564	478	460
Robustez	838	792	841	838	794	794

Tabela 6.3: Comparação entre os algoritmos propostos e o original em relação à eficiência e à robustez. É apresenta a quantidade de problemas da coleção CUTEst que cada algoritmo resolveu com menor tempo (Eficiência) e em um tempo de até três vezes o tempo gasto pelo algoritmo mais eficiente (Robustez).

Problema	A0		A1	
	f	inv	f	inv
A0NNDSL	0	$5,7 \times 10^{-9}$	$4,22 \times 10^{-10}$	4,9
A2NSDSSL	$4,2 \times 10^{-21}$	$8,3 \times 10^{-9}$	$6,59 \times 10^{-18}$	8×10^{-4}
DEGENLPB	-30,7	$1,6 \times 10^{-9}$	-30,8	$1,5 \times 10^{-6}$
HAIFAL	-12,8	$1,5 \times 10^{-11}$	-12,79	$6,7 \times 10^{-6}$
MSQRTA	0	$5,6 \times 10^{-9}$	0	1,6

Tabela 6.4: Comparação entre A0 e A1 em cinco problemas da coleção CUTEst resolvidos pelos algoritmos sem perturbação no ponto inicial mas não resolvidos pelos que realizam a perturbação.

linear. Estes fatos podem trazer instabilidades para as restrições. Já no problema denominado “CHAIN”, constam discretizações de equações diferenciais em suas restrições. Para isso é utilizado diferenças finitas, que também podem refletir em uma grande sensibilidade nas restrições.

A Tabela 6.5 foi construída de tal modo que é possível observar os valores de x^* encontrado por A0 e A1 para o problema “DEGENLPB”.

Note que apenas duas das vinte variáveis do problema possuem valores diferentes em A0 e A1, são elas: x_{11} e x_{17} . Este fato já faz o problema ser viável em A0, mas não em A1. Entretanto, o erro relativo entre as soluções encontradas é da ordem de 10^{-10} . Além disso, é possível observar que o problema é mal escalado. Se todos os valores fossem divididos, por exemplo, por 10^{-12} , as soluções estariam visivelmente muito próximas, conforme é apresentado na Tabela 6.6.

Considerando a precisão na ordem de 10^{-8} os novos valores de x_{11} e x_{17} são iguais. É possível que este mau escalamento seja responsável para que as alterações em x_{11} e x_{17} influenciem no valor das restrições e façam com que A0 declare o ponto encontrado como viável mas A1 não. Este tipo de situação pode ocorrer em outros problemas e talvez isso explique o mau comportamento dos algoritmos perturbados.

	A0	A1
x_1	2,50E+15	2,50E+15
x_2	8,17E+12	8,17E+12
x_3	2,81E+14	2,81E+14
x_4	9,98E+14	9,98E+14
x_5	3,97E+10	3,97E+10
x_6	1,36E+12	1,36E+12
x_7	4,83E+12	4,83E+12
x_8	5,67E+12	5,67E+12
x_9	1,02E+13	1,02E+13
x_{10}	1,96E+15	1,96E+15
x_{11}	1,95E+02	0,00E+00
x_{12}	0,00E+00	0,00E+00
x_{13}	0,00E+00	0,00E+00
x_{14}	0,00E+00	0,00E+00
x_{15}	0,00E+00	0,00E+00
x_{16}	2,00E+13	2,00E+13
x_{17}	2,92E+04	0,00E+00
x_{18}	1,66E+10	1,66E+10
x_{19}	3,93E+13	3,93E+13
x_{20}	1,20E+13	1,20E+13

Tabela 6.5: Valores de x^* para o problema “DEGENLPB”.

	A0	A1
x_1	2,50E+03	2,50E+03
x_2	8,17	8,17
x_3	2,81E+02	2,81E+02
x_4	9,98E+02	9,98E+02
x_5	3,97E-02	3,97E-02
x_6	1,36	1,36
x_7	4,83	4,83
x_8	5,67	5,67
x_9	1,02E+01	1,02E+01
x_{10}	1,96E+03	1,96E+03
x_{11}	1,95E-10	0,00E+00
x_{12}	0,00E+00	0,00E+00
x_{13}	0,00E+00	0,00E+00
x_{14}	0,00E+00	0,00E+00
x_{15}	0,00E+00	0,00E+00
x_{16}	2,00E+01	2,00E+01
x_{17}	2,92E-08	0,00E+00
x_{18}	1,66E-02	1,66E-02
x_{19}	3,93E+01	3,93E+01
x_{20}	1,20E+01	1,20E+01

Tabela 6.6: Valores escalados de x^* para o problema “DEGENLPB”.

Capítulo 7

Conclusão

Neste trabalho é reforçada a necessidade de textos de apoio a discentes de graduação sobre otimização não linear e espera-se que os capítulos dedicados a isso auxiliem o aluno a superar possíveis dificuldades encontradas. Para isso, o enfoque é diferenciado dos livros, procurando evitar formalismos das demonstrações. São destacadas as interpretações geométricas e as relações com o Método de Newton estudado em Cálculo Numérico. Esta abordagem pode motivar o aluno, inclusive os que participam de Iniciação Científica e porventura ainda não cursaram a disciplina de otimização não linear, a aprofundar os estudos sobre o tema. Desta forma, é possível que estes alunos façam uso do ALGENCAN, podendo assim compreender também a importância de métodos computacionais implementados para resolver problemas de otimização.

Em nenhum momento a pretensão deste trabalho é substituir os livros já utilizados como referências bibliográficas nos cursos de otimização não linear que já foram citados no corpo deste trabalho, mas sim de servir como apoio a eles. A visão apresentada neste texto traz o que julgamos, durante o desenvolvimento deste projeto de Mestrado Profissional, ser uma maneira adequada de transmitir conteúdos de Matemática para alunos que vão ter o primeiro contato com o assunto abordado. Nesta perspectiva são ressaltados os aspectos intuitivos e geométricos mas, ao mesmo tempo, não são feitas afirmações incompatíveis com formalismo matemático. Durante todo o trabalho, os resultados discutidos tiveram hipóteses simplificadas de forma que, mesmo perdendo um pouco da profundidade, a leitura fique bem mais acessível para um primeiro contato. Evidentemente, um livro mais técnico seria adequado para uma segunda leitura de pessoas interessadas em se aprofundar no assunto.

Através de exemplos simples foi estudada uma classe de problemas de otimização não linear, os ditos problemas simétricos. Foi visto que, dependendo do ponto inicial tomado, esse tipo de problema pode trazer dificuldades a um algoritmo bastante utilizado em otimização, o ALGENCAN. Para que essas dificuldades fossem sanadas ou, pelo menos, minimizadas, foram feitas modifica-

ções na programação do ALGENCAN. Desta forma, outras cinco versões foram criadas, cada um com uma motivação, porém unidas pelo mesmo objetivo. As modificações feitas baseavam-se em perturbações do ponto inicial e em alterações no critério de parada em casos de pontos inviáveis estacionários da inviabilidade. Todos esses algoritmos se mostraram eficazes para solucionar pelo menos alguns dos problemas simétricos apresentados. Para um estudo mais científico sobre o comportamento dos algoritmos propostos, utilizamos a coleção CUTEst para uma análise em testes mais completos. Como resultado, o algoritmo nomeado de A3, que modifica o critério de parada para um ponto ser classificado como estacionário da inviabilidade, se mostrou o mais eficiente, sendo sutilmente superior ao ALGENCAN original.

Para trabalhos futuros a intenção é incluir tópicos de otimização não linear não tratados neste trabalho, como por exemplo restrições de desigualdade, com o objetivo de se tornar cada vez mais próximo das ementas das disciplinas de otimização ofertadas nas universidades brasileiras. Adicionar as rotinas de Álgebra Linear em ALGENCAN e considerar mais casas decimais no tempo gasto para o ALGENCAN solucionar um determinado problema também são possibilidades de estudos futuros. Outro ponto chave que também merece ser estudado futuramente é o comportamento de métodos que utilizem perturbações mais elaboradas, que possam ser feitas ao longo das iterações. Além do mais, uma análise mais detalhada dos problemas da coleção CUTEst que não foram solucionadas pelos algoritmos perturbados se faz necessária para um entendimento mais profundo deste comportamento.

Referências Bibliográficas

- [1] Roberto Andreani, Gabriel Haeser e José Mario Martínez. “On sequential optimality conditions for smooth constrained optimization”. Em: *Optimization* 60.5 (2011), pp. 627–641.
- [2] Roberto Andreani et al. “On Augmented Lagrangian methods with general lower-level constraints”. Em: *SIAM Journal on Optimization* 18.4 (2007), pp. 1286–1309.
- [3] Ernesto G. Birgin e José Mario Martínez. *Practical Augmented Lagrangian Methods for Constrained Optimization*. Philadelphia: SIAM, 2014.
- [4] Richard W. Cottle, Jong-Shi Pang e Richard E. Stone. *The Linear Complementarity Problem*. Philadelphia: SIAM, 2009.
- [5] *CUTEst: a Constrained and Unconstrained Testing Environment on steroids*. Nov. de 2014. URL: <http://ccpforge.cse.rl.ac.uk/gf/project/cutest/wiki/>.
- [6] Elizabeth D. Dolan e Jorge J. Moré. “Benchmarking optimization software with performance profiles”. Em: *Mathematical programming* 91.2 (2002), pp. 201–213.
- [7] Ana Friedlander. *Elementos de Programação Não-Linear*. Campinas: Editora da UNICAMP, 1994.
- [8] Nicholas IM Gould, Dominique Orban e Philippe L. Toint. “CUTEst: a Constrained and Unconstrained Testing Environment with safe threads”. Em: *Cahier du GERAD G* 2013.27 (2013).
- [9] Igor Griva, Stephen G. Nash e Ariela Sofer. *Linear and nonlinear optimization*. 2^a ed. Philadelphia: SIAM, 2009.
- [10] Alexey Izmailov e Mikhail Solodov. *Otimização - volume 2: Métodos Computacionais*. 4^a ed. Rio de Janeiro: IMPA, 2012.
- [11] Elizabeth Wegner Karas e Ademir Alves Ribeiro. *Otimização Contínua – Aspectos Teóricos e Computacionais*. São Paulo: Cengage Learning, 2014.
- [12] David G. Luenberger e Yinyu Ye. *Linear and nonlinear programming*. 4^a ed. New York: Springer, 2008.
- [13] José Mario Matínez e Sandra Augusta Santos. *Métodos Computacionais de Otimização*. Rio de Janeiro: SBM, IMPA, 1995.
- [14] Liqun Qi e Zengxin Wei. “On the constant positive linear dependence condition and its application to SQP methods”. Em: *SIAM Journal on Optimization* 10 (2000), pp. 963–981.

- [15] Márcia A. Gomes Ruggiero e Vera Lúcia da Rocha Lopes. *Cálculo Numérico – Aspectos Teóricos e Computacionais*. 2^a ed. São Paulo: Makron Books do Brasil, 1996.

Apêndice A

Comandos em Linux

Abaixo estão listados os comandos em Linux utilizados para a instalação e utilização do pacote Algencan e seus respectivos significados

- `sudo apt-get install` - utilizado para instalar softwares. A sigla “apt” significa *advanced package tool*;
- `cd` - utilizado para a navegação entre pastas do computador. Este comando acessa e muda de diretório corrente. Quando após este comando é acrescentado dois pontos finais, então acessa-se o nível superior. A sigla “cd” significa *change directory*;
- `ls -la` - utilizado para listar os arquivos contidos dentro de um determinado diretório e para obter informações mais detalhadas sobre estes;
- `cp` - utilizado para copiar arquivos. A sigla “cp” significa *copy*;
- `tar -zxf` - utilizado para descompactar os arquivos compactados com gzip. Cada letra após o hífen tem um significado. O “f”, por exemplo, diz que o argumento que vier escrito após ele é o arquivo que deverá ser descompactado, o “x” dá o comando de extração. A sigla “tar” significa *tape archive*;
- `pwd` - utilizado para imprimir o nome do diretório onde o usuário se encontra. A sigla “pwd” significa *print working directory*;
- `make` - utilizado para ler e executar o arquivo “Makefile” e compilar arquivos;
- `mkdir` - utilizado para criar um diretório dentro do qual o usuário está. A sigla “mkdir” significa *make directory*;

- `export ALGENCAN` - utilizada para definir a variável ALGENCAN num diretório específico;
- `echo $ALGENCAN` - utilizado para verificar o conteúdo da variável de ambiente ALGENCAN;
- `gfortran toyprob.f90` - utilizado para compilar o arquivo **toyprob.f90** em fortran e criar um executável cujo nome é definido pelo usuário;
- `./teste` - executa o arquivo **teste** criado ao compilar o arquivo **toyprob.f90**;

Apêndice B

Exemplo de otimização com o ALGENCAN

Considere o Exemplo 2.1.4 da página 36 que consiste em

$$\begin{aligned} \min f(x_1, x_2) &= x_1^2 + 2x_2^2 - x_1x_2 + x_1 + 2x_2 \\ \text{sujeito a} \\ \begin{pmatrix} 1 & 5 \\ 2 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} &= \frac{7}{2}. \end{aligned}$$

Abaixo segue o código em fortran para realizar esta otimização se valendo do pacote ALGENCAN.

```
1 ! ****
2 ! ****
3
4 program algencanma
5
6 implicit none
7
8 ! LOCAL SCALARS
9 logical      :: checkder
10 integer       :: allocerr, hnnzmax, hnnzmax1, hnnzmax2, hnnzmax3, &
11      inform, jcnnzmax, m, n, nvparam
12 real(kind=8) :: cnorm, efacc, efstain, eoacc, eostain, epsfeas, epsopt, &
13      f, nlpsupn, snorm
14
15 ! LOCAL ARRAYS
```

```

16   character(len=15)      :: strtmp
17   character(len=80)       :: specfnm, outputfnm, vparam(10)
18   logical                  :: coded(11)
19   logical,     pointer :: equatn(:), linear(:)
20   real(kind=8), pointer :: l(:), lambda(:), u(:), x(:)

21
22 ! EXTERNAL SUBROUTINES
23 external :: myevalf, myevalg, myevalh, myevalc, myevaljac, myevalhc, &
24           myevalfc, myevalgjac, myevalgjacp, myevalhl, myevalhlp
25
26 ! Number of variables (numero de variaveis)
27
28 n = 2
29
30 ! Set lower bounds, upper bounds, and initial guess
31
32 allocate(x(n),l(n),u(n),stat=allocerr)
33 if ( allocerr .ne. 0 ) then
34   write(*,*) 'Allocation error in main program'
35   stop
36 end if
37
38 l(1) = - 1.0d+20 !x(1) maior ou igual a -10^20
39 u(1) = 1.0d+20 !x(1) menor ou igual a 10^20
40
41 l(2) = - 1.0d+20 !x(2) maior ou igual a -10^20
42 u(2) = 1.0d+20 !x(2) menor ou igual a 10^20
43
44 x(1:2) = 0.0d0 !os valores iniciais sao x(1)=x(2)=0
45
46 ! Constraints (numero de restricoes)
47
48 m = 1
49

```

```

50  allocate(equatn(m),linear(m),lambda(m),stat=allocerr)
51  if ( allocerr .ne. 0 ) then
52    write(*,*) 'Allocation error in main program'
53    stop
54  end if
55
56  equatn(1:m) = .true. ! A restricao e de igualdade
57  lambda(1:m) = 0.0d0 ! O valor inicial de lambda e zero
58
59  linear(1) = .true. ! A restricao 1 e linear
60
61  ! Coded subroutines
62
63  coded(1:6) = .true. ! fsub, gsub, hsub, csub, jacsub, hcsub
64  coded(7:11) = .false. ! fcsub, gjacsub, gjacpsub, hlsub, hlpsub
65
66  ! Upper bounds on the number of sparse-matrices non-null elements
67
68  ! numero maximo de elementos nao nulos do jacobiano da restricao
69  jcnnzmax = 4
70  ! numero maximo de elementos nao nulos da hessiana da funcao objetivo
71  hnnzmax1 = 4
72  ! numero maximo de elementos nao nulos da hessiana da restricao
73  hnnzmax2 = 4
74
75  hnnzmax = hnnzmax1 + hnnzmax2
76
77  ! Checking derivatives? (checar as derivadas)
78
79  checkder = .true.
80
81  ! Parameters setting
82
83  epsfeas = 1.0d-08

```

```

84    epsopt      = 1.0d-08
85
86    efstain     = sqrt( epsfeas )
87    eostain     = epsopt ** 1.5d0
88
89    efacc       = sqrt( epsfeas )
90    eoacc       = sqrt( epsopt )
91
92    outputfnm   =
93    specfnm     =
94
95    nvparam     = 1
96    ! produz um arquivo com os valores de x(1), x(2) e lambda otimos
97    vparam(1) = 'SOLUTION-FILENAME_solucao.txt'
98
99
100   ! Optimize
101
102  call algencan(myevalf,myevalg,myevalh,myevalc,myevaljac,myevalhc, &
103                  myevalfc,myevalgjac,myevalgjacp,myevalhl,myevalhlp,jcnnzmax, &
104                  hnnzmax,epsfeas,epsopt,efstain,eostain,efacc,eoacc,outputfnm, &
105                  specfnm,nvparam,vparam,n,x,l,u,m,lambda,equatn,linear,coded, &
106                  checkder,f,cnorm,snorm,nlpsupn,inform)
107
108  deallocate(x,l,u,lambda,equatn,linear,stat=allocerr)
109  if ( allocerr .ne. 0 ) then
110      write(*,*) 'Deallocation_error_in_main_program'
111      stop
112  end if
113
114  stop
115
116 end program algencanma
117

```

```
118 ! ****
119 ! ****
120
121 subroutine myevalf(n,x,f,flag)
122
123     implicit none
124
125     ! SCALAR ARGUMENTS
126     integer,      intent(in)   :: n
127     integer,      intent(out)  :: flag
128     real(kind=8), intent(out)  :: f
129
130     ! ARRAY ARGUMENTS
131     real(kind=8), intent(in)   :: x(n)
132
133     flag = 0
134
135     ! funcao objetivo
136     f = x(1)**2 + 2.0d0*x(1) - x(1)*x(2) + x(1) + 2.0d0*x(2)
137
138 end subroutine myevalf
139
140 ! ****
141 ! ****
142
143 subroutine myevalg(n,x,g,flag)
144
145     implicit none
146
147     ! SCALAR ARGUMENTS
148     integer,      intent(in)   :: n
149     integer,      intent(out)  :: flag
150
151     ! ARRAY ARGUMENTS
```

```

152     real(kind=8), intent(in) :: x(n)
153     real(kind=8), intent(out) :: g(n)
154
155     flag = 0
156
157     ! gradiente da funcao objetivo
158     g(1) = 2.0d0*x(1) - x(2) + 1.0d0
159     g(2) = 4.0d0*x(2) - x(1) + 2.0d0
160
161 end subroutine myevalg
162
163 ! ****
164 !
165
166 subroutine myevalh(n,x,hrow,hcol,hval,hnnz,lim,lmem,flag)
167
168 implicit none
169
170 ! SCALAR ARGUMENTS
171 logical,      intent(out) :: lmem
172 integer,       intent(in)  :: lim,n
173 integer,       intent(out) :: flag,hnnz
174
175 ! ARRAY ARGUMENTS
176 integer,       intent(out) :: hcol(lim),hrow(lim)
177 real(kind=8), intent(in)  :: x(n)
178 real(kind=8), intent(out) :: hval(lim)
179
180 flag = 0
181 lmem = .false.
182
183 hnnz = 3 ! numero maximo de elementos nao nulos
184           ! da hessiana da funcao objetivo que
185           ! o usuario precisa informar

```

```

186
187      hrow(1) = 1
188      hcol(1) = 1
189      hval(1) = 2.0d0
190      hrow(2) = 2
191      hcol(2) = 1
192      hval(2) = - 1.0d0
193      hrow(3) = 2
194      hcol(3) = 2
195      hval(3) = 4.0d0
196
197  end subroutine myevalh
198
199 ! ****
200 !
201
202 subroutine myevalc(n,x,ind,c,flag)
203
204     implicit none
205
206     ! SCALAR ARGUMENTS
207     integer , intent(in) :: ind,n
208     integer , intent(out) :: flag
209     real(kind=8) , intent(out) :: c
210
211     ! ARRAY ARGUMENTS
212     real(kind=8) , intent(in) :: x(n)
213
214     flag = 0
215
216     if ( ind .eq. 1 ) then
217         c = x(1) + 25d-01*x(2) - 35d-01 !restricao
218         return
219     end if

```

```

220
221
222 end subroutine myevalc
223
224 ! ****
225 ! ****
226
227 subroutine myevaljac(n,x,ind,jcvar,jcval,jcnnz,lim,lmem,flag)
228
229 implicit none
230
231 ! SCALAR ARGUMENTS
232 logical, intent(out) :: lmem
233 integer, intent(in) :: ind,lim,n
234 integer, intent(out) :: flag,jcnnz
235
236 ! ARRAY ARGUMENTS
237 integer, intent(out) :: jcvar(lim)
238 real(kind=8), intent(in) :: x(n)
239 real(kind=8), intent(out) :: jcval(lim)
240
241 flag = 0
242 lmem = .false.
243
244 if ( ind .eq. 1 ) then
245     jcnnz = 2 ! numero maximo de elementos nao
246                 ! nulos do jacobiano da restricao
247
248     if ( jcnnz .gt. lim ) then
249         lmem = .true.
250         return
251     end if
252
253     jcvar(1) = 1

```

```

254     jcval(1) = 1
255     jcvar(2) = 2
256     jcval(2) = 25d-01
257
258     return
259   end if
260
261
262 end subroutine myevaljac
263
264 ! ****
265 !
266
267 subroutine myevalhc(n,x,ind,hcrow,hccol,hcval,hcnnz,lim,lmem,flag)
268
269 implicit none
270
271 ! SCALAR ARGUMENTS
272 logical, intent(out) :: lmem
273 integer, intent(in) :: ind, lim, n
274 integer, intent(out) :: flag, hcnnz
275
276 ! ARRAY ARGUMENTS
277 integer, intent(out) :: hccol(lim), hcrow(lim)
278 real(kind=8), intent(in) :: x(n)
279 real(kind=8), intent(out) :: hcval(lim)
280
281 flag = 0
282 lmem = .false.
283
284 if ( ind .eq. 1 ) then
285   hcnnz = 0 ! numero maximo de elementos
286           ! nao nulos da hessiana da restricao
287           ! que o usuario precisa informar

```

```
288
289
290     if ( hcnnz .gt. lim ) then
291         lmem = .true.
292         return
293     end if
294
295
296     return
297 end if
298
299
300
301 end subroutine myevalhc
302
303 ! *****
304 !
305 subroutine myevalfc(n,x,f,m,c,flag)
306
307 implicit none
308
309 ! SCALAR ARGUMENTS
310
311 integer, intent(in) :: m,n
312 integer, intent(out) :: flag
313 real(kind=8), intent(out) :: f
314
315 ! ARRAY ARGUMENTS
316
317 real(kind=8), intent(in) :: x(n)
318 real(kind=8), intent(out) :: c(m)
319
320 flag = - 1
321
322 end subroutine myevalfc
```

```

322
323 ! ****
324 ! ****
325
326 subroutine myevalgjac(n,x,g,m,jcfun,jcvar,jcval,jcnnz,lim,lmem,flag)
327
328 implicit none
329
330 ! SCALAR ARGUMENTS
331 logical, intent(out) :: lmem
332 integer, intent(in) :: lim,m,n
333 integer, intent(out) :: flag,jcnnz
334
335 ! ARRAY ARGUMENTS
336 integer, intent(out) :: jcfun(lim),jcvar(lim)
337 real(kind=8), intent(in) :: x(n)
338 real(kind=8), intent(out) :: g(n),jcval(lim)
339
340 flag = - 1
341
342 end subroutine myevalgjac
343
344 ! ****
345 ! ****
346
347 subroutine myevalgjacp(n,x,g,m,p,q,work,gotj,flag)
348
349 implicit none
350
351 ! SCALAR ARGUMENTS
352 logical, intent(inout) :: gotj
353 integer, intent(in) :: m,n
354 integer, intent(out) :: flag
355 character, intent(in) :: work

```

```

356
357      ! ARRAY ARGUMENTS
358      real(kind=8), intent(in)    :: x(n)
359      real(kind=8), intent(inout) :: p(m),q(n)
360      real(kind=8), intent(out)   :: g(n)
361
362      flag = - 1
363
364 end subroutine myevalgjacp
365
366 ! ****
367 ! ****
368
369 subroutine myevalhl(n,x,m,lambda,sf,sc,hlrow,hlcol,hlval,hlnnz,lim,&
370                      lmem,flag)
371
372 implicit none
373
374      ! SCALAR ARGUMENTS
375      logical,      intent(out) :: lmem
376      integer,      intent(in)  :: lim,m,n
377      integer,      intent(out) :: flag,hlnnz
378      real(kind=8), intent(in)  :: sf
379
380      ! ARRAY ARGUMENTS
381      integer,      intent(out) :: hlcol(lim),hlrow(lim)
382      real(kind=8), intent(in)  :: lambda(m),sc(m),x(n)
383      real(kind=8), intent(out) :: hlval(lim)
384
385      flag = - 1
386
387 end subroutine myevalhl
388
389 ! ****

```

```

390 ! ****
391
392 subroutine myevalhlp(n,x,m,lambda,sf,sc,p,hp,goth,flag)
393
394 implicit none
395
396 ! SCALAR ARGUMENTS
397 logical, intent(inout) :: goth
398 integer, intent(in) :: m,n
399 integer, intent(out) :: flag
400 real(kind=8), intent(in) :: sf
401
402 ! ARRAY ARGUMENTS
403 real(kind=8), intent(in) :: lambda(m),p(n),sc(m),x(n)
404 real(kind=8), intent(out) :: hp(n)
405
406 flag = - 1
407
408 end subroutine myevalhlp

```

A solução encontrada pelo ALGENCAN pode ser verificada no arquivo “SOLUCAO.TXT” que é gerado pelo programa após sua execução e se encontra na pasta *myfolder*.

```

1
2 FINAL POINT:
3
4 INDEX X(INDEX)
5     1  9.999999769143622D-01
6     2  9.999999858234667D-01
7
8 FINAL ESTIMATION OF THE LAGRANGE MULTIPLIERS:
9
10 INDEX LAMBDA(INDEX)
11    1 -1.999999981028336D+00

```

Note que a solução encontrada pelo ALGENCAN é muito próxima da solução analítica, pois analiticamente $x^* = (1, 1)$ e pelo ALGENCAN

$$x^* = (0, 9999999769143622, 0, 9999999858234667).$$