

ADEMIR ALVES RIBEIRO

**CONVERGÊNCIA GLOBAL DOS MÉTODOS DE FILTRO
PARA PROGRAMAÇÃO NÃO LINEAR**

Tese apresentada como requisito parcial à
obtenção do grau de Doutor, pelo Programa
de Pós-Graduação em Métodos Numéricos
em Engenharia, Setores de Ciências Exatas e
Tecnologia, Universidade Federal do Paraná.

Orientador:

Prof. Dr. Celso Carnieri

Co-orientadora:

Prof^a. Dr^a. Elizabeth Wegner Karas.

Curitiba

2005

TERMO DE APROVAÇÃO

*Para ser grande, sê inteiro
Nada teu exagera ou exclui
Sê todo em cada coisa
Põe o quanto és no mínimo que fazes
Assim em cada lago a lua toda brilha,
porque alta vive.*

Fernando Pessoa

*Para
Samira, Juliana e Gabriel.*

Agradecimentos

A conclusão deste trabalho só foi possível graças ao apoio que recebi de várias pessoas e de algumas instituições, que gostaria de agradecer.

Ao meu orientador, Professor Celso Carnieri, pelo que aprendi nos diversos seminários que realizamos e pela liberdade dada trabalhar em um assunto que para mim foi muito interessante.

À Professora Elizabeth Wegner Karas, pela inestimável tarefa de co-orientação e pela amizade, que por várias vezes me impediu de desistir.

De forma muito especial, agradeço ao Professor Clóvis Caesar Gonzaga que, mesmo não sendo meu orientador, dedicou grande parte do seu tempo para ajudar a perceber as idéias principais que fazem parte desta tese.

Aos Professores José Mario Martínez, Raimundo José Borges de Sampaio e Yuan Jin Yun por aceitarem participar da banca julgadora desta tese e pelas importantes considerações que fizeram, tornando mais claras algumas partes deste trabalho.

À Professora Marli Cardia, que considero minha “madrinha acadêmica”, por tudo que já me ensinou, pelo incentivo e também pela leitura cuidadosa do texto, cujas sugestões melhoraram a redação final.

Ao Professor Anselmo Chaves Neto, pelo apoio incondicional que tem me dado, desde a época em que eu cursava o mestrado.

Aos meus amigos, por todo o estímulo que recebi para concluir este trabalho, deixo aqui meus sinceros agradecimentos.

Agradeço também ao Programa de Pós-Graduação em Métodos Numéricos em Engenharia da Universidade Federal do Paraná pela oportunidade de cursar o doutorado e ao Departamento de Matemática pela licença concedida para que eu pudesse me dedicar exclusivamente ao curso.

À minha mãe que, mesmo não estando mais entre nós, tanto me iluminou e me deu forças para continuar.

À todos os meus familiares e em especial à minha esposa, Samira e aos meus filhos, Juliana e Gabriel, que também deixaram de fazer muitas coisas para estar do meu lado, dando apoio para subir este degrau muito importante na minha vida.

Resumo

Discutimos neste trabalho uma classe de métodos para programação não linear. Tais métodos, conhecidos como *métodos de filtro*, foram introduzidos recentemente por Fletcher e Leyffer com o objetivo de evitar o uso de funções de mérito, uma ferramenta comum na maioria dos algoritmos para otimização com restrições. Ao invés disso, a idéia é usar um filtro para induzir a convergência, que interfere o mínimo possível nos passos obtidos pelo algoritmo.

Apresentamos um algoritmo geral de filtro com uma grande liberdade para o cálculo do passo. O algoritmo é muito simples e consiste basicamente em calcular um ponto não proibido pelo filtro, a partir do ponto corrente. Provamos a convergência global do algoritmo proposto, assumindo que o passo seja eficiente, no sentido de que, perto de um ponto viável não estacionário, o decréscimo da função objetivo seja “grande”. Mostramos que esta condição é razoável, exibindo dois algoritmos clássicos para a obtenção do passo, que satisfazem tal hipótese. O primeiro obtém o passo pelo método de restauração inexata de Martínez e Pilotta e a prova de que a condição é satisfeita é dada por Gonzaga, Karas e Vanti. Outra forma de calcular o passo é pelo método de programação quadrática sequencial. Provamos neste trabalho que esta abordagem também cumpre a referida condição.

Discutimos também aspectos de convergência local, como o uso de correção de segunda ordem para evitar o *efeito Maratos*, um sério problema deste e de outros critérios de aceitação de passo que pode comprometer seriamente a eficiência do algoritmo.

Realizamos testes computacionais utilizando uma família de problemas conhecidos como *Hard-Spheres problems*, que fornece um conjunto conveniente de problemas-teste para avaliar e comparar algoritmos de programação não linear.

Abstract

In this work we discuss a class of methods for nonlinear programming. Such methods, known as *filter methods*, were introduced recently by Fletcher and Leyffer, whose aim is to dispense the need for a merit function, a common tool in most algorithms for constrained optimization. Instead, the idea is to use a filter to induce convergence, which interferes as little as possible with the step obtained by the underlying method.

We present a very general filter algorithm that allows a great deal of freedom in the step computation. The algorithm consists basically in computing a point which is not forbidden by the filter, from the current point. We prove its global convergence, assuming that the step must be efficient, in the sense that, near a feasible non-stationary point, the reduction of the objective function is “large”. We show that this condition is reasonable, by presenting two classical ways for performing the step, satisfying this condition. On the first one, the step is obtained by the inexact restoration method of Martínez and Pilotta and the proof of such a condition is given by Gonzaga, Karas and Vanti. Another way for computing the step is by sequential quadratic programming. We prove in this work that this approach also satisfies the efficiency condition.

We also discuss local convergence features, like the usage of a second order correction in order to avoid the *Maratos effect*, a serious defect of this and other step-acceptance criteria, which can reduce the convergence rate of the algorithm.

We perform numerical tests on a family of problems known as *Hard-Spheres problems*, that provides a suitable set of test problems for evaluating nonlinear programming codes.

Sumário

Introdução	1
1 Programação quadrática sequencial	5
1.1 O método de PQS local	6
1.2 Globalização em PQS	12
1.2.1 PQS com busca linear	13
1.2.2 PQS com região de confiança	14
2 Algoritmos de filtro: uma teoria geral de convergência	19
2.1 O algoritmo	23
2.2 Convergência global do algoritmo de filtro	25
2.3 Algoritmos internos para a obtenção do passo	29
2.3.1 Restauração Inexata	29
2.3.2 Programação Quadrática Sequencial	30
2.3.3 Discussão comparativa	46
3 Estudo de convergência local em programação não linear	49
3.1 Efeito Maratos	50
3.2 Correção de segunda ordem	51
3.3 Perspectivas	56
4 Resultados numéricos	57
4.1 O problema “Hard-Spheres”	58
4.2 Métodos de restauração inexata	60
4.2.1 Fase de viabilidade	61
4.2.2 Fase de otimalidade	61
4.2.3 Critério de aceitação do passo	62
4.3 Implementação	62
4.3.1 Experimentos computacionais	62

Conclusões	72
Anexos	75
Referências Bibliográficas	77

Introdução

Discutimos neste trabalho uma classe de métodos para resolver problemas de Programação Não Linear (PNL) da forma

$$(P) \quad \begin{array}{ll} \text{minimizar} & f_0(x) \\ \text{sujeito a} & f_{\mathcal{E}}(x) = 0 \\ & f_{\mathcal{I}}(x) \leq 0, \end{array}$$

onde as funções $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$, $i \in \{0\} \cup \mathcal{E} \cup \mathcal{I}$, são continuamente diferenciáveis. Poderemos considerar $\mathcal{E} = \emptyset$ ou $\mathcal{I} = \emptyset$, conforme estejamos interessados em estudar problemas apenas com restrições de desigualdades ou apenas com igualdades, respectivamente.

Temos aqui dois objetivos conflitantes: reduzir a função objetivo e obter viabilidade. Uma forma de combinar otimalidade e viabilidade é fazer uso das funções de mérito, como proposto por Gomes, Maciel e Martínez [22] em um trabalho que utiliza Programação Quadrática Sequencial (PQS) com região de confiança para obtenção do passo, que é avaliado pela função de mérito.

Byrd, Gilbert e Nocedal [6] e Byrd, Hribar e Nocedal [7] também utilizam função de mérito para avaliar o passo. O algoritmo, denominado NITRO, usa idéias dos métodos de pontos interiores e de programação quadrática sequencial com região de confiança.

Outra forma de utilizar função de mérito é proposta por Martínez [32] e Martínez e Pilotta [33], nos algoritmos de *Restauração Inexata*. Nestes algoritmos cada iteração é decomposta em duas fases: viabilidade e otimalidade. Na fase de otimalidade é aplicado um método de região de confiança. Entretanto o centro da região de confiança é o ponto obtido na fase de viabilidade e não o ponto corrente. A função de mérito avalia então o ponto tentativo obtido na fase de otimalidade. Caso não seja aceito, a região de confiança é reduzida e um novo ponto tentativo é calculado. A diferença em relação aos métodos de PQS com região de confiança, como, por exemplo, aquele proposto em [22], está no fato de que em restauração inexata a fase de

viabilidade não é repetida quando se reduz a região de confiança.

Função de mérito também aparece nos algoritmos propostos em [15, 20, 45, 48], onde o método de Newton é aplicado às condições de KKT do problema, obtendo-se uma direção de busca. É então feita uma busca linear para encontrar um passo que garanta uma redução suficiente na função de mérito.

Apesar de muito bem estabelecidos, os métodos que fazem uso de funções de mérito tem algumas desvantagens. Uma delas reside na não diferenciabilidade de algumas funções de mérito clássicas. Outra, talvez o ponto mais delicado, é a escolha do parâmetro de penalidade. Dependendo do valor deste parâmetro o algoritmo pode recusar uma solução ótima ou, por outro lado, ficar bastante lento.

Como uma alternativa aos métodos que utilizam funções de mérito, Fletcher e Leyffer [17] introduziram os chamados *Métodos de Filtro* para globalização em PNL. Estes métodos são baseados em Programação Quadrática Sequencial (PQS) e caracterizados pelo uso do conceito de *dominância*, da otimização multi-critério.

Em sua forma pura, PQS resolve um problema de PNL através de uma sequência de problemas de programação quadrática, onde minimiza-se um modelo quadrático do lagrangeano, sujeito a linearização das restrições. Pode ser provado [1, 2, 4, 12, 39] que, sob certas condições, este método tem convergência local quadrática. No entanto, para um ponto inicial arbitrário, não se tem garantia de convergência, a menos que se use algum critério de aceitação do passo.

A idéia dos métodos de filtro é interferir o mínimo possível nas iterações de PQS, mas de modo a induzir a convergência. Além disso, evitar o uso de parâmetro de penalidade e ser menos restritivo que os métodos que utilizam função de mérito.

Basicamente, um algoritmo de filtro define uma *região proibida*, armazenando pares $(f_0(x^j), h(x^j))$, escolhidos convenientemente das iterações anteriores, sendo $h(\cdot)$ uma medida de inviabilidade. Um ponto tentativo é aceito quando não for dominado por nenhum elemento do filtro, segundo a regra:

y é dominado por x se, e somente se,

$$f_0(y) \geq f_0(x) \quad \text{e} \quad h(y) \geq h(x).$$

Os métodos de filtro também foram aplicados no contexto de Programação Linear Sequencial (PLS) com região de confiança. Os trabalhos de Chin e Fletcher [9] e Fletcher, Leyffer e Toint [18] apresentam prova de convergência global para o método.

Para os métodos de filtro com PQS, a convergência global foi obtida por Fletcher, Leyffer e Toint [19], assumindo que uma solução exata dos subproblemas

quadráticos era encontrada. Enfraquecendo esta hipótese, ou seja, exigindo apenas uma solução “aproximada” dos subproblemas, Fletcher *et al* [16] também estabelecem convergência global, utilizando uma técnica de decomposição de passo baseada nos trabalhos de Byrd [5] e Omojokun [40]. Outra abordagem do uso de filtro, ainda baseada em PQS, foi proposta por Pu-yan Nie [38], que considera a linearização das restrições como uma penalidade no modelo quadrático ao invés de uma restrição no subproblema. O algoritmo resultante também tem propriedade de convergência global.

Ulbrich, Ulbrich e Vicente [46] aliaram as técnicas de filtro aos algoritmos de pontos interiores, e estabeleceram convergência global do algoritmo. Entretanto eles definem o filtro de um modo diferente do que foi proposto originalmente em [17]. As entradas do filtro, ao invés da função objetivo e medida de inviabilidade, são o lagrangeano e uma medida de inviabilidade que envolve as variáveis primais e duais.

Gonzaga, Karas e Vanti [23] propuseram um algoritmo de filtro globalmente convergente para programação não linear, baseado em restauração inexata [32, 33]. Cada iteração é composta de uma fase de restauração, que reduz uma medida de inviabilidade, e de uma fase de otimalidade, que melhora o valor da função objetivo em uma aproximação tangencial do conjunto viável.

De modo geral, os bons resultados dos algoritmos de filtro motivaram sua utilização na resolução de outros problemas. Gould, Leyffer e Toint [24] apresentam um algoritmo de filtro multi-dimensional para resolução de sistemas de equações não lineares. A dimensão do filtro está relacionada com o número de equações do problema. Gould, Sainvitu e Toint [25] estendem as idéias de filtro multi-dimensional para problemas de minimização irrestrita.

Karas, Ribeiro, Sagastizábal e Solodov [29] aplicaram as técnicas de filtro ao método de feixes proposto em [44], para a resolução de problemas de otimização convexa não diferenciável.

As idéias de filtro também foram utilizadas por Gomes [21] para definir um critério de aceitação de passo que combina filtro e função de mérito em uma *função de mérito linear por partes*, dando origem a um critério mais tolerante que a função de mérito.

Embora os algoritmos de filtro sejam menos restritivos que os algoritmos que usam funções de mérito, eles podem sofrer do efeito Maratos, ou seja, podem recusar um passo essencialmente bom, mesmo arbitrariamente próximo de uma solução. Para remediar isto, podemos incorporar uma correção de segunda ordem ao passo, como sugerido em [17]. Vários trabalhos apresentam estudo de convergência local dos métodos de filtro. Wächter e Biegler [50] propõem um algoritmo de filtro, globalmente conver-

gente, usando busca linear. Em [49] os mesmos autores discutem o uso de correção de segunda ordem para evitar o efeito Maratos e estabelecem convergência superlinear. A mesma técnica é utilizada por Chin [8]. Ulbrich [47] também prova convergência superlinear para métodos de filtro mas sem o uso de correção de segunda ordem. A estratégia está na definição do filtro, onde as entradas são o lagrangeano e uma medida de inviabilidade que envolve as variáveis primais e duais.

Proposta do trabalho. Nós propomos nesta tese um algoritmo geral de filtro para resolução de problemas de programação não linear que consiste basicamente em:

Dado x^k , calcule um ponto x^{k+1} não proibido e atualize o filtro.

Provamos a convergência global do algoritmo proposto, assumindo que o passo seja eficiente, no sentido de que, perto de um ponto viável não estacionário a função objetivo “decreça bastante”. Esta condição é similar ao critério de Polak [41] para convergência global de algoritmos e está formalizada na Hipótese H4, no Capítulo 2.

Mostramos que a referida condição é razoável, exibindo dois algoritmos clássicos para a obtenção do passo, que satisfazem tal hipótese. Um deles calcula o passo usando as idéias de restauração inexata de Martínez [32] e Martínez e Pilotta [33], sendo a prova de que a Hipótese H4 é satisfeita dada por Gonzaga, Karas e Vanti [23].

Outra forma clássica de calcular o passo é através de PQS. Propomos um algoritmo de PQS baseado naquele apresentado por Fletcher *et al* [16] e provamos neste trabalho que o algoritmo proposto cumpre a Hipótese H4.

Organização dos capítulos. No Capítulo 1 apresentamos uma revisão dos métodos de programação quadrática sequencial, enfatizando a relação com o método de Newton. O Capítulo 2 apresenta o resultado principal desta pesquisa. Propomos um algoritmo geral para os métodos de filtro com uma grande liberdade para o cálculo do passo. Discutimos sua convergência global e duas formas de obtenção do passo. No Capítulo 3 discutimos alguns aspectos relacionados com a convergência local dos métodos baseados em PQS, como o uso de correção de segunda ordem para evitar o efeito Maratos. Finalmente, no Capítulo 4, apresentamos testes computacionais, utilizando uma classe de problemas conhecidos como “*Hard-Spheres Problems*”, com o intuito de comparar os algoritmos de restauração inexata com filtro [23] e com função de mérito [32, 33].

Capítulo 1

Programação quadrática sequencial

O objetivo deste capítulo é revisar os conceitos relacionados com programação quadrática sequencial (PQS), que serão úteis no desenvolvimento deste trabalho.

O método PQS é um dos métodos mais eficazes para otimização não linear com restrições e procura obter a solução de um problema de PNL através de uma sequência de problemas de programação quadrática.

O princípio que norteia este método é comum quando se pretende resolver, de forma aproximada, um problema matemático: a solução de um problema “difícil” vai sendo aproximada por uma sequência de pontos obtidos como solução de um problema “fácil”, que muda a cada iteração de acordo com as informações disponíveis no ponto corrente. O método de Newton para a resolução de sistemas de equações não lineares é um exemplo clássico. Neste caso os problemas “fáceis” são obtidos tomando-se a linearização do sistema que queremos resolver, em torno do ponto corrente. Temos assim um sistema de equações lineares, cuja solução (que existe sob certas hipóteses) é tomada como próximo ponto da sequência.

Nos métodos de programação quadrática sequencial a idéia consiste em substituir, a cada iteração, a função objetivo por um modelo quadrático e as restrições por equações ou inequações lineares, aproximações de Taylor de primeira ordem das restrições do problema original em torno do ponto corrente. O modelo quadrático é a aproximação de Taylor de segunda ordem do lagrangeano em torno do ponto corrente.

Veremos que existe um íntima relação entre PQS e o método de Newton, o que permite estabelecer convergência local rápida (superlinear ou quadrática) para os métodos de PQS. Entretanto, do mesmo modo como acontece com o método de Newton, a PQS pura não tem a propriedade de convergência global, isto é, se o ponto inicial estiver longe de uma solução, não se garante que a sequência gerada pelo algoritmo seja convergente, nem mesmo que tenha algum ponto de acumulação. É necessário,

portanto, inserir o método PQS em uma estrutura globalmente convergente. As duas estratégias básicas de globalização são a *busca linear* e *região de confiança*. Neste capítulo discutiremos, brevemente, estas estratégias. Para um estudo mais detalhado sobre o assunto podem ser consultadas inúmeras referências, dentre as quais [4, 12, 22, 26, 34, 39].

1.1 O método de PQS local

Para simplificar a exposição e evidenciar a relação entre PQS e o método de Newton vamos considerar o problema (P), apresentado na introdução deste trabalho, com $\mathcal{I} = \emptyset$, isto é, vamos estudar o problema apenas com restrições de igualdades:

$$(P_E) \quad \begin{array}{ll} \text{minimizar} & f_0(x) \\ \text{sujeito a} & f(x) = 0, \end{array}$$

com $f_0 : \mathbb{R}^n \rightarrow \mathbb{R}$, $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$, duas vezes continuamente diferenciáveis.

Um ponto que satisfaça certas condições necessárias de otimalidade, como por exemplo, as condições de KKT, será chamado de *ponto estacionário*.

Denotamos a matriz jacobiana de f no ponto x por $A(x)$. O lagrangeano associado ao problema (P_E) é dado por

$$x \in \mathbb{R}^n, \lambda \in \mathbb{R}^m \mapsto \ell(x, \lambda) = f_0(x) + \lambda^T f(x), \quad (1.1)$$

onde o vetor λ é chamado multiplicador de Lagrange. A hessiana parcial do lagrangeano, $\nabla_{xx}^2 \ell(x, \lambda)$, é denotada por $H(x, \lambda)$.

Dados x^k e λ^k , o algoritmo básico de PQS consiste em resolver a cada iteração o seguinte problema quadrático

$$(PQ_k) \quad \begin{array}{ll} \text{minimizar} & \ell(x^k, \lambda^k) + \nabla_x \ell(x^k, \lambda^k)^T d + \frac{1}{2} d^T H(x^k, \lambda^k) d \\ \text{sujeito a} & A(x^k) d + f(x^k) = 0, \end{array}$$

que sob certas hipóteses tem solução única d^k . Definimos então $x^{k+1} = x^k + d^k$, estimamos o multiplicador de Lagrange λ^{k+1} e repetimos o processo com o novo problema quadrático (PQ_{k+1}). Veremos agora que este procedimento, quando aplicado em uma vizinhança de um ponto estacionário onde são satisfeitas as condições suficientes de segunda ordem, está bem definido e produz uma sequência que converge quadraticamente para este ponto.

Hipóteses. Para estabelecer a convergência do algoritmo vamos considerar uma solução primal-dual (x^*, λ^*) do problema (P_E) e assumir que valem as seguintes condições.

H1 As funções $\nabla^2 f_i(\cdot)$, $i = 0, 1, \dots, m$, são lipschitzianas em uma vizinhança de x^* .

H2 A matriz jacobiana das restrições, $A(x^*)$, tem posto linha completo e a hessiana parcial $H(x^*, \lambda^*)$ é definida positiva no espaço tangente das restrições, isto é, $d^T H(x^*, \lambda^*) d > 0$ para todo $d \neq 0$, $d \in \mathcal{N}(A(x^*))$.

Vamos descrever de modo mais preciso o algoritmo básico de PQS.

Algoritmo 1.1 *PQS básico*

Dados: $k = 0$, $(x^0, \lambda^0) \in \mathbb{R}^n \times \mathbb{R}^m$,

ENQUANTO $\nabla \ell(x^k, \lambda^k) \neq 0$

 Resolva o subproblema quadrático (PQ_k) ,

 obtendo uma solução primal-dual (d^k, μ^k)

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

$$\lambda^{k+1} = \lambda^k + \mu^k$$

$$k = k + 1$$

FIM.

Um ponto (x^*, λ^*) satisfaz o critério de parada quando

$$\nabla \ell(x^*, \lambda^*) = \begin{bmatrix} \nabla f_0(x^*) + A(x^*)^T \lambda^* \\ f(x^*) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad (1.2)$$

isto é, quando cumpre as condições de otimalidade de primeira ordem para o problema (P_E) . Por outro lado, uma solução primal-dual (d^k, μ^k) do subproblema quadrático (PQ_k) satisfaz

$$\begin{cases} H(x^k, \lambda^k) d^k + A(x^k)^T \mu^k & = -\nabla_x \ell(x^k, \lambda^k) \\ A(x^k) d^k & = -f(x^k). \end{cases} \quad (1.3)$$

Convergência local. Os resultados abaixo estabelecem a convergência local do Algoritmo 1.1 ao mesmo tempo que evidenciam a relação com o método de Newton. O seguinte lema garante que se (x^k, λ^k) está próximo de (x^*, λ^*) , o passo (d^k, μ^k) satisfazendo (1.3) está bem definido e é único.

Lema 1.2 *Seja (x^*, λ^*) uma solução primal-dual do problema (P_E) e suponha que a Hipótese H2 seja satisfeita. Então existe uma vizinhança V_1 de (x^*, λ^*) , tal que se $(x^k, \lambda^k) \in V_1$, o sistema (1.3) tem uma única solução (d^k, μ^k) .*

Demonstração. Usando [4, Proposição 12.1] podemos concluir que a matriz

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

é não singular. Por continuidade, segue que existe uma vizinhança V_1 de (x^*, λ^*) tal que se $(x^k, \lambda^k) \in V_1$, então

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

também é não singular. Mas isto significa que o sistema (1.3), que pode ser escrito como

$$\begin{bmatrix} H(x^k, \lambda^k) & A(x^k)^T \\ A(x^k) & 0 \end{bmatrix} \begin{bmatrix} d^k \\ \mu^k \end{bmatrix} = \begin{bmatrix} -\nabla_x \ell(x^k, \lambda^k) \\ -f(x^k) \end{bmatrix},$$

tem uma única solução (d^k, μ^k) , completando a demonstração. \square

Nas condições do Lema 1.2, o vetor d^k é o único minimizador global do sub-problema (PQ_k) .

Vamos agora ao principal resultado desta seção.

Teorema 1.3 *Seja (x^*, λ^*) uma solução primal-dual do problema (P_E) e suponha que as Hipóteses H1 e H2 sejam satisfeitas. Então existe uma vizinhança V_2 de (x^*, λ^*) , tal que se $(x^0, \lambda^0) \in V_2$, o Algoritmo 1.1 está bem definido e, se o critério de parada não for satisfeito, gera uma sequência $(x^k, \lambda^k)_{k \in \mathbb{N}}$ que converge quadraticamente para esta solução.*

Demonstração. Basta notar que o passo (d^k, μ^k) definido pelo Algoritmo 1.1 é exatamente o passo de Newton para o sistema de equações

$$\nabla \ell(x, \lambda) = \begin{bmatrix} \nabla f_0(x) + A(x)^T \lambda \\ f(x) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \quad (1.4)$$

De fato, a jacobiana da função $(x, \lambda) \mapsto \nabla \ell(x, \lambda)$ é a matriz

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

e assim o passo de Newton para (1.4), (d_N^k, μ_N^k) , é dado por

$$\begin{bmatrix} H(x^k, \lambda^k) & A(x^k)^T \\ A(x^k) & 0 \end{bmatrix} \begin{bmatrix} d_N^k \\ \mu_N^k \end{bmatrix} = - \begin{bmatrix} \nabla f_0(x^k) + A(x^k)^T \lambda^k \\ f(x^k) \end{bmatrix} = \begin{bmatrix} -\nabla_x \ell(x^k, \lambda^k) \\ -f(x^k) \end{bmatrix},$$

ou seja, pelo sistema (1.3).

Se (x^k, λ^k) está na vizinhança dada no Lema 1.2, bem como na região de convergência do método de Newton, então o passo (d_N^k, μ_N^k) coincide com o passo PQS e o Algoritmo 1.1 está bem definido. Além disso, a convergência quadrática segue dos resultados sobre o método de Newton, por exemplo, [4, Teorema 11.6]. \square

Ressaltamos que a convergência quadrática estabelecida no Teorema 1.3 é da sequência $(x^k, \lambda^k)_{k \in \mathbb{N}}$ e isto não implica que a convergência de $(x^k)_{k \in \mathbb{N}}$ seja quadrática. No exemplo abaixo, tirado de [4, Exercício 12.8], temos uma sequência $(x^k, \lambda^k)_{k \in \mathbb{N}}$ convergindo quadraticamente para $(0, 0)$, enquanto que a convergência de $(x^k)_{k \in \mathbb{N}}$ para 0 não é sequer linear.

Exemplo 1.4

Defina $x^0 = \lambda^0 = 1$ e, para $k \geq 1$,

$$x^k = \begin{cases} \beta^{2^k}, & \text{se } k \text{ é ímpar} \\ x^{k-1}, & \text{se } k \text{ é par} \end{cases} \quad \text{e} \quad \lambda^k = \beta^{2^{k-1}},$$

onde $\beta \in (0, 1)$. Temos então $x^k = \lambda^k$, se k é par e $x^k = (\lambda^k)^2 < \lambda^k$, se k é ímpar. Além disso, $\lambda^{k+1} = (\lambda^k)^2$, para todo $k \geq 1$ e portanto,

$$\frac{\|(x^{k+1}, \lambda^{k+1})\|_\infty}{\|(x^k, \lambda^k)\|_\infty^2} = \frac{\lambda^{k+1}}{(\lambda^k)^2} = 1.$$

Por outro lado, temos

$$\frac{x^{k+1}}{x^k} = 1, \text{ se } k \text{ é ímpar.}$$

\square

É possível, entretanto, modificar o Algoritmo 1.1 de modo a transformá-lo em um algoritmo puramente primal e ter convergência quadrática na sequência $(x^k)_{k \in \mathbb{N}}$. Ao invés de calcular $\lambda^{k+1} = \lambda^k + \mu^k$, utilizamos uma estimativa $\lambda^k = \lambda(x^k)$, onde $\lambda(\cdot)$ é uma inversa a direita de $A(\cdot)$. Em [4, Teorema 12.5] podemos encontrar tal abordagem.

O algoritmo PQS, discutido aqui, pode ser interpretado de outro modo. Fazendo $\xi^k = \lambda^k + \mu^k$, a relação (1.3) pode ser reescrita como

$$\begin{cases} H(x^k, \lambda^k)d^k + \nabla f_0(x^k) + A(x^k)^T \xi^k & = 0 \\ A(x^k)d^k + f(x^k) & = 0, \end{cases}$$

que representa as condições de otimalidade do problema quadrático

$$(PQ_k^0) \quad \begin{array}{ll} \text{minimizar} & f_0(x^k) + \nabla f_0(x^k)^T d + \frac{1}{2} d^T H(x^k, \lambda^k) d \\ \text{sujeito a} & A(x^k) d + f(x^k) = 0. \end{array}$$

Podemos assim fazer uma releitura do algoritmo PQS e dizer que minimizamos a cada iteração um modelo quadrático da função objetivo f_0 , sujeito a linearização das restrições. Entretanto, neste modelo quadrático incorporamos na hessiana informações sobre a curvatura das restrições.

É interessante notar que considerando em (PQ_k^0) o modelo

$$f_0(x^k) + \nabla f_0(x^k)^T d + \frac{1}{2} d^T \nabla^2 f_0(x^k) d, \quad (1.5)$$

isto é, a aproximação de Taylor de segunda ordem de f_0 , o algoritmo não funciona, conforme nos mostra o exemplo seguinte.

Exemplo 1.5

Considere o problema [4, Exercício 12.1]

$$\begin{array}{ll} \text{minimizar} & f_0(x) = -\frac{x_1^2}{2} + 2x_2 \\ \text{sujeito a} & f(x) = x_1^2 + x_2^2 - 1 = 0, \end{array} \quad (1.6)$$

cuja solução (única e global) é o ponto $x^* = \begin{pmatrix} 0 \\ -1 \end{pmatrix}$, com multiplicador correspondente $\lambda^* = 1$. Suponha que o ponto corrente seja $x(\delta) = \begin{pmatrix} \delta \\ -\sqrt{1-\delta^2} \end{pmatrix}$. Assim, se utilizarmos (1.5) como modelo, o subproblema quadrático associado a (1.6) fica

$$\begin{array}{ll} \text{minimizar} & -\frac{d_1^2}{2} - \delta d_1 + 2d_2 \\ \text{sujeito a} & \delta d_1 - \sqrt{1-\delta^2} d_2 = 0, \end{array} \quad (1.7)$$

já desconsiderando os termos constantes no modelo. Resolvendo as condições de KKT para (1.7), obtemos

$$d_1 = \frac{2\delta}{\sqrt{1-\delta^2}} - \delta \quad \text{e} \quad d_2 = \frac{2\delta^2}{1-\delta^2} - \frac{\delta^2}{\sqrt{1-\delta^2}}.$$

Para δ suficientemente pequeno o ponto $x(\delta)$ fica muito próximo da solução x^* . No entanto, pode ser mostrado que

$$\|x(\delta) + d - x^*\| \approx 2\|x(\delta) - x^*\|.$$

Ou seja, mesmo estando o ponto corrente arbitrariamente próximo da solução, o passo determinado por (1.7) aproximadamente duplica a distância ao minimizador.

Vamos aproveitar este exemplo para calcular o passo verdadeiro de PQS, solução do subproblema

$$\begin{aligned} &\text{minimizar} \quad \frac{1}{2}(d_1^2 + 2d_2^2) - \delta d_1 + 2d_2 \\ &\text{sujeito a} \quad \delta d_1 - \sqrt{1-\delta^2}d_2 = 0, \end{aligned} \tag{1.8}$$

que é o problema (PQ_k^0) com

$$H(x^k, \lambda^k) = \nabla_{xx}^2 \ell(x(\delta), \lambda^*) = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}.$$

A solução de (1.8) é o vetor

$$d_{\text{pqs}} = \frac{(\sqrt{1-\delta^2} - 2)\delta}{1 + \delta^2} \begin{pmatrix} \sqrt{1-\delta^2} \\ \delta \end{pmatrix}.$$

Neste caso temos

$$\frac{\|x(\delta) + d_{\text{pqs}} - x^*\|}{\|x(\delta) - x^*\|^2} \approx \frac{1}{2},$$

o que está em conformidade com o Teorema 1.3. □

A Figura 1.1 ilustra este exemplo, onde o conjunto viável está representado pela linha circular cheia, as curvas de nível da função objetivo pelas linhas tracejadas e $x^+ = x(\delta) + d_{\text{pqs}}$.

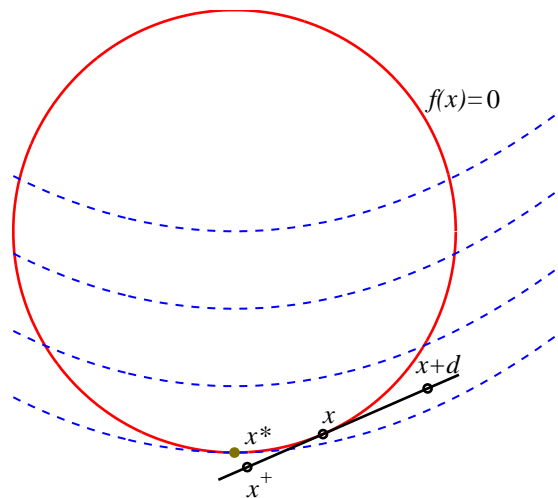


Figura 1.1: O passo de “Pseudo” PQS para o Exemplo 1.5.

1.2 Globalização em PQS

Conforme vimos na seção anterior, a PQS pura gera uma sequência convergente se o ponto inicial estiver próximo de uma solução do problema. Entretanto, para ser prático, um algoritmo deve ser capaz de encontrar uma solução mesmo se o ponto inicial estiver longe desta solução. Vamos discutir agora algumas formas de modificar o algoritmo local de PQS, de modo a atender tal requisito. Chamamos isto de globalização do algoritmo local. Para sermos mais precisos, vamos dar a seguinte definição.

Definição 1.6 *Um algoritmo é dito globalmente convergente quando dada qualquer sequência $(x^k)_{k \in \mathbb{N}}$, gerada pelo algoritmo, todos os pontos de acumulação de $(x^k)_{k \in \mathbb{N}}$ (se existirem) são estacionários.*

Observação. Muitas vezes a expressão *convergência global* é utilizada para expressar que as sequências geradas pelo algoritmo tem *peelo menos um* ponto de acumulação estacionário.

Basicamente, existem duas técnicas de globalização: *busca linear* e *região de confiança*. Em ambas o progresso obtido do ponto corrente x^k para o próximo ponto x^{k+1} é normalmente medido com o auxílio de uma *função de mérito*¹. Para ser aceito, o passo deve provocar um *decréscimo suficiente* na função de mérito.

¹Uma alternativa ao uso de funções de mérito para avaliar o passo foi introduzida por Fletcher e Leyffer [17], com os chamados *métodos de filtro*. Os capítulos seguintes serão dedicados ao estudo de tais métodos.

Diferentemente da otimização irrestrita, onde a própria função objetivo é adequada para avaliar a qualidade de um passo, aqui devemos também considerar a viabilidade. Uma função de mérito muito utilizada para o problema (P_E) é dada por

$$\phi_\sigma(x) = f_0(x) + \sigma \|f(x)\|, \quad (1.9)$$

onde σ é chamado parâmetro de penalidade. É também popular como função de mérito o *lagrangeano aumentado de Fletcher*, definido por

$$\phi_\sigma^F(x) = f_0(x) + \lambda(x)^T f(x) + \sigma \|f(x)\|^2, \quad (1.10)$$

sendo que

$$\lambda(x) = (A(x)A(x)^T)^{-1} A(x)\nabla f_0(x)$$

é uma estimativa para os multiplicadores.

Para a discussão sobre as técnicas de globalização que faremos nas seções seguintes, vamos considerar a função de mérito definida por (1.9).

1.2.1 PQS com busca linear

Neste método calculamos em cada iteração, uma direção de busca d^k , resolvendo o sistema (1.3). Após isto, determinamos, utilizando *backtracking*, um tamanho de passo $\alpha_k \in (0, 1]$ de modo a satisfazer a condição de Armijo

$$\phi_{\sigma_k}(x^k + \alpha_k d^k) \leq \phi_{\sigma_k}(x^k) + \eta \alpha_k D(\phi_{\sigma_k}(x^k), d^k), \quad (1.11)$$

onde $\eta \in (0, \frac{1}{2})$ e $D(\phi_{\sigma_k}(x^k), d^k)$ é a derivada direcional² da função ϕ_{σ_k} no ponto x^k e na direção d^k . Definimos então, $x^{k+1} = x^k + \alpha_k d^k$.

Naturalmente, precisamos garantir que é possível encontrar α_k satisfazendo a condição (1.11), o que não conseguimos se $D(\phi_{\sigma_k}(x^k), d^k) \geq 0$. O teorema seguinte garante que, sob certas hipóteses, d^k é uma direção de descida para ϕ_{σ_k} .

Teorema 1.7 *Suponha que x^k não seja ponto estacionário para o problema (P_E) e que $H(x^k, \lambda^k)$ seja definida positiva no espaço nulo de $A(x^k)$. Então $D(\phi_{\sigma_k}(x^k), d^k) < 0$, para σ_k suficientemente grande.*

Demonstração. [39, Teorema 18.3]. □

²A função ϕ , apesar de não ser diferenciável, tem derivadas direcionais [12, Seção 3.1.4].

Apresentamos agora um algoritmo de PQS com busca linear onde a hessiana $H(x^k, \lambda^k)$ é aproximada por uma matriz B_k , definida positiva.

Algoritmo 1.8 *PQS com busca linear*

Dados: $\eta \in (0, \frac{1}{2})$, $\tau \in (0, 1)$, $(x^0, \lambda^0) \in \mathbb{R}^n \times \mathbb{R}^m$, B_0 definida positiva.

$k = 0$

ENQUANTO $\nabla \ell(x^k, \lambda^k) \neq 0$

Resolva o subproblema quadrático (PQ_k), obtendo uma solução d^k

Escolha σ_k tal que d^k seja direção de descida para ϕ_{σ_k}

$\alpha_k = 1$

ENQUANTO $\phi_{\sigma_k}(x^k + \alpha_k d^k) > \phi_{\sigma_k}(x^k) + \eta \alpha_k D(\phi_{\sigma_k}(x^k), d^k)$

$\alpha_k = \tau \alpha_k$

FIM

$x^{k+1} = x^k + \alpha_k d^k$

$\lambda^{k+1} = - (A(x^{k+1})A(x^{k+1})^T)^{-1} A(x^{k+1}) \nabla f_0(x^{k+1})$

Obtenha B_{k+1} , atualizando B_k por uma fórmula quase-Newton

$k = k + 1$

FIM.

Pode-se provar que o uso de busca linear garante a convergência global em PQS. Veja, por exemplo, [4, Capítulo 15].

1.2.2 PQS com região de confiança

Quando resolvemos o subproblema (PQ_k^0) para dar um passo em direção ao (esperado) minimizador de (P_E), estamos acreditando que o modelo seja representativo. Como sabemos, as aproximações de Taylor de uma função são boas quando estamos perto do ponto em torno do qual elas foram tomadas. A estratégia de região de confiança força esta proximidade, acrescentando uma restrição de bola ao subproblema, de modo a limitar o tamanho do passo. Deste modo, passamos a trabalhar com o seguinte subproblema

$$(PQ_k^{rc}) \quad \begin{array}{ll} \text{minimizar} & f_0(x^k) + \nabla f_0(x^k)^T d + \frac{1}{2} d^T B_k d \\ \text{sujeito a} & A(x^k) d + f(x^k) = 0, \\ & \|d\| \leq \Delta_k, \end{array}$$

onde B_k é uma matriz simétrica, podendo ser $H(x^k, \lambda^k)$.

Se o passo d^k , obtido resolvendo-se o subproblema (PQ_k^{rc}) , provocar um decréscimo suficiente em uma função de mérito, definimos $x^{k+1} = x^k + d^k$ e atualizamos (aumentando ou deixando inalterado) o raio Δ_k para a próxima iteração. Caso contrário, rejeitamos o passo e diminuimos o raio.

Certamente, a inclusão da restrição $\|d\| \leq \Delta_k$ torna o problema consideravelmente mais difícil. Pior ainda, (PQ_k^{rc}) pode ser incompatível se $f(x^k) \neq 0$ e Δ_k for muito pequeno. Note que não basta aumentar o raio Δ_k pois isto fere a filosofia do método de região de confiança quando o modelo não é representativo. Na Figura 1.2 temos um caso de incompatibilidade, onde consideramos um problema em \mathbb{R}^2 com apenas uma restrição. Nesta figura, o conjunto viável e a sua linearização estão representados por linhas cheias, enquanto que as curvas de nível da função objetivo são as linhas tracejadas.

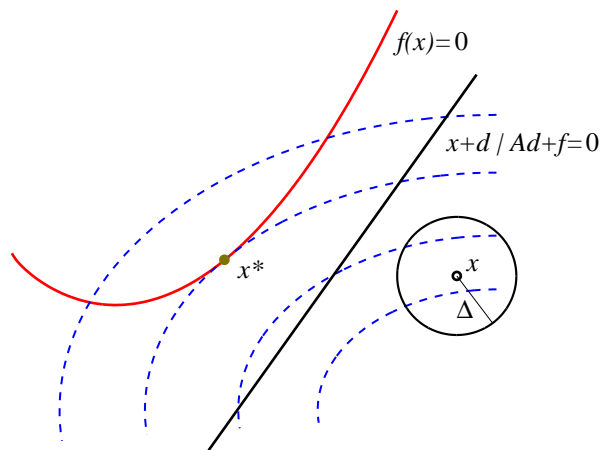


Figura 1.2: Incompatibilidade do subproblema (PQ_k^{rc}) .

Uma das possíveis abordagens para contornar este problema é descrita em [39] e consiste em fazer uma translação das restrições lineares de modo que o novo problema seja compatível. Mais precisamente, devemos calcular um \hat{f}_k conveniente e considerar o problema

$$\begin{aligned} &\text{minimizar} && f_0(x^k) + \nabla f_0(x^k)^T d + \frac{1}{2} d^T B_k d \\ &\text{sujeito a} && A(x^k)d + \hat{f}_k = 0, \\ &&& \|d\| \leq \Delta_k. \end{aligned}$$

A abordagem que veremos é devida a Byrd [5] e Omojokun [40] e se baseia na decomposição do passo em duas componentes, normal e tangencial. Em uma primeira etapa vamos nos preocupar somente com viabilidade, resolvendo o subproblema normal

$$(PN_k^{rc}) \quad \begin{aligned} & \text{minimizar} \quad \|A(x^k)n + f(x^k)\| \\ & \text{sujeito a} \quad \|n\| \leq \xi \Delta_k, \end{aligned}$$

onde $\xi \in (0, 1)$ é um parâmetro que serve para evitar que o passo normal chegue até a fronteira da bola $\|n\| \leq \Delta_k$. Isto é necessário para que esteja bem definida a próxima etapa, que busca a otimalidade por meio do subproblema tangencial

$$\begin{aligned} & \text{minimizar} \quad f_0(x^k) + \nabla f_0(x^k)^T d + \frac{1}{2} d^T B_k d \\ & \text{sujeito a} \quad A(x^k)d = A(x^k)n^k, \\ & \quad \quad \quad \|d\| \leq \Delta_k, \end{aligned}$$

sendo que n^k é a solução de (PN_k^{rc}) .

Fazendo $d = n^k + t$ e desconsiderando os termos constantes, podemos reescrever o problema acima como

$$(PT_k^{rc}) \quad \begin{aligned} & \text{minimizar} \quad (\nabla f_0(x^k) + B_k n^k)^T t + \frac{1}{2} t^T B_k t \\ & \text{sujeito a} \quad A(x^k)t = 0, \\ & \quad \quad \quad \|n^k + t\| \leq \Delta_k, \end{aligned}$$

que também chamaremos de subproblema tangencial. Uma solução de (PT_k^{rc}) será indicada por t^k .

Naturalmente, esta decomposição do passo pode ser usada quando (PQ_k^{rc}) for compatível. Se $\{n \mid A(x^k)n + f(x^k) = 0 \text{ e } \|n\| < \xi \Delta_k\}$ é não vazio, existem infinitas soluções para (PN_k^{rc}) e tomamos neste caso a solução de mínima norma.

Na Figura 1.3 ilustramos a estratégia descrita acima nos dois casos, quando (PQ_k^{rc}) é e quando não é compatível.

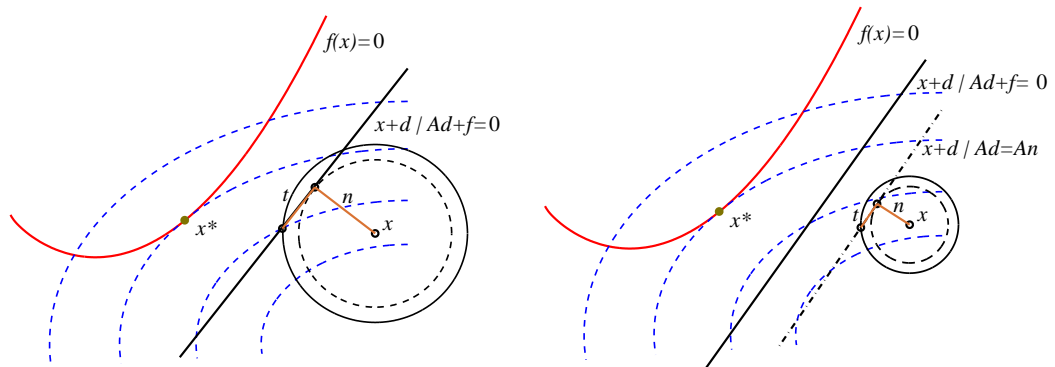


Figura 1.3: Decomposição do passo em normal e tangencial.

Tendo calculado o passo $d^k = n^k + t^k$, resta ver se ele é aceito pelo critério de decréscimo suficiente em alguma função de mérito. Neste estudo, vamos considerar a que foi definida pela relação (1.9), retomada aqui por comodidade,

$$\phi_\sigma(x) = f_0(x) + \sigma \|f(x)\|, \quad (1.12)$$

sendo σ o parâmetro de penalidade.

Um modelo conveniente para ϕ_σ é dado por

$$m_{\phi,\sigma}(d) = m_k(d) + \sigma \|A(x^k)d + f(x^k)\|, \quad (1.13)$$

onde

$$m_k(d) = f_0(x^k) + \nabla f_0(x^k)^T d + \frac{1}{2} d^T B_k d. \quad (1.14)$$

Note que o modelo (1.13) é obtido com a mesma filosofia que usamos para construir o subproblema quadrático a partir do problema original, isto é, tomamos um modelo quadrático para a função objetivo e um modelo linear para as restrições, sendo que no modelo quadrático incorporamos informações sobre a curvatura das restrições. A redução verdadeira na função de mérito é

$$ared = \phi_\sigma(x^k) - \phi_\sigma(x^k + d^k),$$

enquanto que a redução predita pelo modelo é definida por

$$pred = m_k(n^k) - m_k(d^k) + \sigma (\|f(x^k)\| - \|A(x^k)n^k + f(x^k)\|).$$

Esta definição de $pred$ envolve as reduções preditas nos passos normal e tangencial, quando resolvemos (PN_k^{rc}) e (PT_k^{rc}) , respectivamente. No primeiro, o progresso é medido entre 0 e n^k , já no segundo a redução é avaliada entre n^k e $d^k = n^k + t^k$.

Temos agora os ingredientes para apresentar um algoritmo de PQS com região de confiança, para resolver o problema (P_E) .

Algoritmo 1.9 PQS com região de confiança

Dados: $\eta \in (0, \frac{1}{2})$, $\sigma_0 > 0$, $\Delta_0 > 0$, $(x^0, \lambda^0) \in \mathbb{R}^n \times \mathbb{R}^m$, B_0 definida positiva.

$k = 0$

ENQUANTO $\nabla \ell(x^k, \lambda^k) \neq 0$

 Calcule o passo normal n^k , resolvendo (PN_k^{rc})

 Calcule o passo tangencial t^k , resolvendo (PT_k^{rc})

$$d^k = n^k + t^k$$

 Escolha σ_k

$$ared = \phi_\sigma(x^k) - \phi_\sigma(x^k + d^k)$$

$$pred = m_k(n^k) - m_k(d^k) + \sigma (\|f(x^k)\| - \|A(x^k)n^k + f(x^k)\|)$$

 se $ared > \eta pred$,

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

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

$$\lambda^{k+1} = - (A(x^{k+1})A(x^{k+1})^T)^{-1} A(x^{k+1})\nabla f_0(x^{k+1})$$

 Obtenha B_{k+1} , atualizando B_k por uma fórmula quase-Newton

$$k = k + 1$$

senão

$$\Delta_k = \frac{\Delta_k}{2}$$

FIM.

Um estudo detalhado sobre PQS com região de confiança pode ser encontrado em [12, Capítulo 15], onde se estabelece a convergência global do método.

Vimos neste capítulo alguns conceitos relacionados com PQS. No aspecto de convergência local, discutimos a relação entre PQS e o método de Newton. Por outro lado, vimos que o uso de funções de mérito para avaliar o passo permite estabelecer a convergência global do método. O próximo capítulo apresenta outra estratégia de globalização para PNL: os métodos de filtro.

Capítulo 2

Algoritmos de filtro: uma teoria geral de convergência

Este capítulo traz o resultado principal da nossa pesquisa. Propomos uma teoria geral para os chamados *Métodos de Filtro*, que são algoritmos iterativos aplicados na resolução de problemas de Programação Não Linear (PNL). A idéia é apresentar um algoritmo com uma grande liberdade para o cálculo do passo, desde que se cumpra uma condição afirmando que próximo de um ponto viável não estacionário, a função objetivo “cai bastante”. Formalizamos esta condição na Hipótese H4, adiante. Provamos que o algoritmo proposto tem convergência global e em seguida apresentamos duas formas de calcular o passo, de modo a satisfazer tal condição.

Os problemas de PNL que consideramos aqui são da forma

$$(P) \quad \begin{array}{ll} \text{minimizar} & f_0(x) \\ \text{sujeito a} & f_{\mathcal{E}}(x) = 0 \\ & f_{\mathcal{I}}(x) \leq 0, \end{array}$$

onde as funções $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$, $i \in \{0\} \cup \mathcal{E} \cup \mathcal{I}$, são continuamente diferenciáveis. Os conjuntos de índices \mathcal{E} e \mathcal{I} se referem às restrições de igualdades e desigualdades, respectivamente.

Um algoritmo para resolver o problema (P) deve lidar com dois objetivos normalmente conflitantes: reduzir a função objetivo e obter viabilidade. Como os algoritmos abordados aqui são iterativos e aceitam pontos inviáveis ao longo das iterações, vamos considerar a função $f^+ : \mathbb{R}^n \rightarrow \mathbb{R}^m$, dada por

$$f_i^+(x) = \begin{cases} f_i(x) & \text{se } i \in \mathcal{E} \\ \max\{0, f_i(x)\} & \text{se } i \in \mathcal{I}, \end{cases}$$

e definir uma medida de inviabilidade $h : \mathbb{R}^n \rightarrow \mathbb{R}$ por

$$h(x) = \|f^+(x)\|, \quad (2.1)$$

onde $\|\cdot\|$ é uma norma arbitrária e m é a cardinalidade de $\mathcal{E} \cup \mathcal{I}$. A função h é uma medida exata de inviabilidade, no sentido de que um ponto x é viável se, e somente se, $h(x) = 0$.

Tradicionalmente, f_0 e h são combinadas em uma única função, como nos métodos que usam função de mérito, discutidos brevemente no Capítulo 1, nos métodos de penalidade, Lagrangeano aumentado e também nos métodos de restauração inexata [32, 33], entre outros.

No entanto, apesar de muito bem estabelecidos, tais métodos sofrem de um sério problema, que é a manipulação do parâmetro de penalidade. Dependendo do valor deste parâmetro o algoritmo pode recusar uma solução ótima ou, por outro lado, ficar bastante lento. Outro ponto desfavorável está na não diferenciabilidade de algumas funções de mérito clássicas.

Ao invés de combinar objetivo e inviabilidade em uma única função, Fletcher e Leyffer [17] resolveram trabalhar separadamente com estes dois objetivos, como é feito na otimização multi-critério, com prioridade para a redução da inviabilidade, já que no final devemos obter um ponto viável. Suas idéias deram origem aos *Métodos de Filtro* para globalização em PNL. Esses métodos são baseados em Programação Quadrática Sequencial (PQS) e caracterizados pelo uso do conceito de *dominância*, segundo a regra:

y é dominado por x se, e somente se,

$$f_0(y) \geq f_0(x) \quad \text{e} \quad h(y) \geq h(x).$$

A idéia dos métodos de filtro é interferir o mínimo possível nas iterações de PQS, mas de modo a induzir a convergência. Além disso, evitar o uso de parâmetro de penalidade e serem menos restritivos que os métodos que utilizam função de mérito.

Basicamente, um algoritmo de filtro define uma *região proibida*, armazenando pares $(f_0(x^j), h(x^j))$, escolhidos convenientemente das iterações anteriores. Um ponto tentativo é aceito quando não for dominado por nenhum elemento do filtro.

A Figura 2.1 ilustra o conceito de filtro, onde a região proibida aparece hachurada¹. Esta região fica determinada pelos seus vértices, isto é, pelos pares (f_0, h)

¹Na realidade, a região proibida está em \mathbb{R}^n e é obtida pela pré-imagem daquela pela função $x \in \mathbb{R}^n \mapsto (f_0(x), h(x))$, mas continuaremos utilizando este abuso de linguagem.

representados pelos pequenos círculos. Também estão indicados² nesta figura um ponto aceitável z e um ponto proibido y .

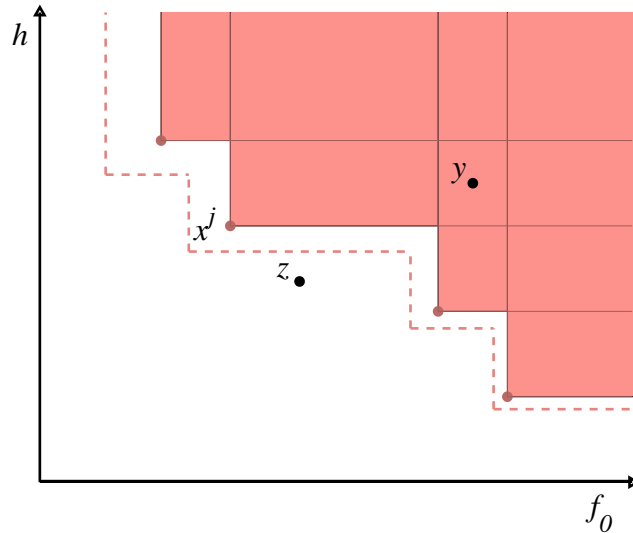


Figura 2.1: O conceito de filtro.

Esta idéia simples precisa, no entanto, ser refinada para que o algoritmo resultante tenha propriedades de convergência global. Sem a inclusão de certas salvaguardas, poderíamos ter uma sequência de pontos se acumulando na vizinhança de um ponto (f_0, h) do filtro, com $h > 0$. Uma sugestão para contornar este problema foi dada pelos próprios autores Fletcher e Leyffer [17] e consiste em criar uma pequena “margem” em torno da região proibida, na qual os pontos também serão considerados proibidos. Na Figura 2.1 esta margem é delimitada pela linha tracejada. Sua função é similar àquela da redução suficiente, exigida para as funções de mérito. Formalizando, dada uma constante $\alpha \in (0, 1)$, diremos que um ponto x é aceitável quando

$$f_0(x) < f_0(x^j) - \alpha h(x^j) \quad \text{ou} \quad h(x) < (1 - \alpha)h(x^j),$$

para todo par $(f_0(x^j), h(x^j))$ do filtro³.

As idéias de filtro não se limitaram às introduzidas por Fletcher e Leyffer [17]. Nos trabalhos de Chin e Fletcher [9] e Fletcher, Leyffer e Toint [18] o filtro foi aplicado no contexto de Programação Linear Sequencial (PLS) com região de confiança e a convergência global para o método foi estabelecida.

No caso de PQS, a convergência global foi obtida por Fletcher, Leyffer e Toint

²Vamos utilizar um abuso de notação, indicando por x o par $(f_0(x), h(x))$.

³Na verdade, são os pares $(\tilde{f}_0, \tilde{h}) = (f_0(x^j) - \alpha h(x^j), (1 - \alpha)h(x^j))$ que vamos armazenar no filtro.

[19], exigindo a solução exata dos subproblemas quadráticos. Sem esta exigência, ou seja, trabalhando apenas com soluções “aproximadas” dos subproblemas (no sentido de ser melhor que a solução de Cauchy), Fletcher *et al* [16] também estabelecem a convergência global do método de filtro, utilizando uma técnica de decomposição de passo baseada nos trabalhos de Byrd [5] e Omojokun [40]. Outra abordagem do uso de filtro, ainda baseada em PQS, foi proposta por Pu-yan Nie [38], que considera a linearização das restrições como uma penalidade no modelo quadrático ao invés de uma restrição no subproblema. O algoritmo resultante também tem propriedade de convergência global.

Ulbrich, Ulbrich e Vicente [46] aliaram as técnicas de filtro aos algoritmos de pontos interiores, e estabeleceram convergência global do algoritmo. Entretanto, as entradas do filtro, ao invés da função objetivo e medida de inviabilidade, são o lagrangeano e uma medida de inviabilidade que envolve as variáveis primais e duais.

Gonzaga, Karas e Vanti [23] propuseram um algoritmo de filtro globalmente convergente para programação não linear baseado em restauração inexata [32, 33]. Cada iteração é composta de uma fase de restauração, que reduz uma medida de inviabilidade, e de uma fase de otimalidade, que melhora o valor da função objetivo em uma aproximação tangencial do conjunto viável.

De modo geral, os bons resultados dos algoritmos de filtro motivaram sua utilização na resolução de outros problemas. Gould, Leyffer e Toint [24] apresentam um algoritmo de filtro multi-dimensional para resolução de sistemas de equações não lineares. A dimensão do filtro está relacionada ao número de equações do problema. Gould, Sainvitu e Toint [25] estendem as idéias de filtro multi-dimensional para problemas de minimização irrestrita.

Karas, Ribeiro, Sagastizábal e Solodov [29] aplicaram as técnicas de filtro ao método de feixes proposto em [44], para a resolução de problemas de otimização convexa não diferenciável.

As idéias de filtro também foram utilizadas por Gomes [21] para definir um critério de aceitação de passo que combina filtro e função de mérito em uma *função de mérito linear por partes*, dando origem a um critério mais tolerante que a função de mérito.

Vamos agora apresentar nosso algoritmo geral de filtro, discutindo em seguida condições que garantam sua convergência global.

2.1 O algoritmo

O algoritmo apresentado abaixo está colocado de uma forma bem genérica, sem especificar o critério de parada e a forma de obtenção do passo. Este é justamente o objetivo deste trabalho: um algoritmo geral com uma grande liberdade para o cálculo do passo. Veremos que se uma determinada condição for satisfeita, o algoritmo terá propriedade de convergência global. Discutiremos adiante duas formas de calcular o passo que cumprem tal condição.

Algoritmo 2.1 *Filtro geral*

Dados: $x^0 \in \mathbb{R}^n$, $F_0 = \emptyset$, $\mathcal{F}_0 = \emptyset$, $\alpha \in (0, 1)$.

$k = 0$

REPITA

$$(\tilde{f}_0, \tilde{h}) = (f_0(x^k) - \alpha h(x^k), (1 - \alpha)h(x^k)).$$

Defina os conjuntos $\bar{F}_k = F_k \cup \{(\tilde{f}_0, \tilde{h})\}$ e

$$\bar{\mathcal{F}}_k = \mathcal{F}_k \cup \{x \in \mathbb{R}^n \mid f_0(x) \geq \tilde{f}_0, h(x) \geq \tilde{h}\}.$$

Passo:

SE x^k é estacionário, pare com sucesso

SENÃO, calcule $x^{k+1} \notin \bar{\mathcal{F}}_k$.

Atualização do filtro:

SE $f_0(x^{k+1}) < f_0(x^k)$,

$$F_{k+1} = F_k, \quad \mathcal{F}_{k+1} = \mathcal{F}_k \quad (\text{iteração } f_0)$$

SENÃO,

$$F_{k+1} = \bar{F}_k, \quad \mathcal{F}_{k+1} = \bar{\mathcal{F}}_k \quad (\text{iteração } h)$$

$k = k + 1$.

No início de cada iteração, uma região proibida é determinada pelo filtro. Esta região é composta de uma parte permanente, denotada \mathcal{F}_k , e outra parte temporária,

$$\left\{ x \in \mathbb{R}^n \mid f_0(x) \geq \tilde{f}_0, h(x) \geq \tilde{h} \right\},$$

que se tornará permanente caso a função objetivo não diminua de x^k para x^{k+1} . Neste caso, o par (\tilde{f}_0, \tilde{h}) é incluído no filtro.

As Figuras 2.2 e 2.3 ilustram (no plano $f_0 \times h$) uma iteração completa do algoritmo, assim como o filtro para a próxima iteração, onde, por abuso de notação, indicamos $x = (f_0(x^k), h(x^k))$ e $x^+ = (f_0(x^{k+1}), h(x^{k+1}))$. Na Figura 2.2 a iteração

é do tipo f_0 e conseqüentemente o par (\tilde{f}_0, \tilde{h}) não é incluído no filtro para a próxima iteração. A região proibida é toda a parte hachurada, sendo que a parte amarela é temporária e deixa de ser proibida para a iteração seguinte. Já na Figura 2.3 a iteração é do tipo h , o par (\tilde{f}_0, \tilde{h}) é incluído no filtro e a região proibida temporária passa a ser permanente.

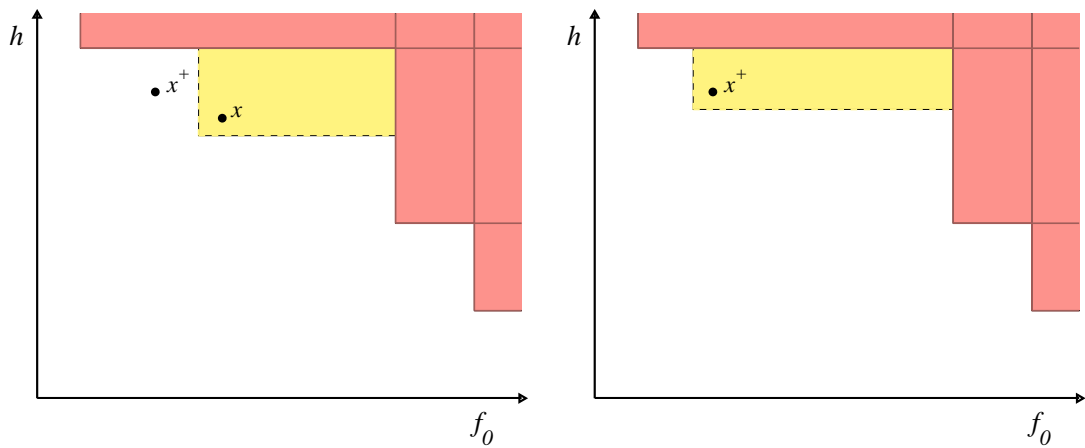


Figura 2.2: Uma iteração f_0 e o filtro para a próxima iteração.

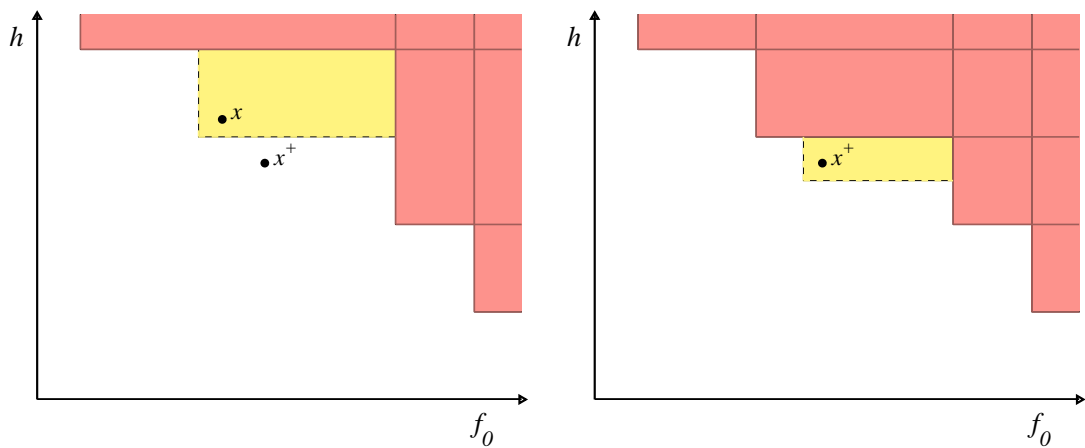


Figura 2.3: Uma iteração h e o filtro para a próxima iteração.

2.2 Convergência global do algoritmo de filtro

Para estabelecer a convergência vamos supor que o algoritmo gera uma sequência infinita $(x^k)_{k \in \mathbb{N}}$ em \mathbb{R}^n e que são satisfeitas as seguintes hipóteses.

H1 As funções $f_i(\cdot)$, $i = 0, 1, \dots, m$, são duas vezes continuamente diferenciáveis, com derivadas primeiras lipschitzianas.

H2 A sequência $(x^k)_{k \in \mathbb{N}}$ permanece em um conjunto convexo e compacto $X \subset \mathbb{R}^n$.

H3 Todo ponto de acumulação viável $\bar{x} \in X$ de $(x^k)_{k \in \mathbb{N}}$ satisfaz a condição de qualificação de Mangasarian-Fromovitz, a saber, os gradientes $\nabla f_i(\bar{x})$, $i \in \mathcal{E}$, são linearmente independentes e existe uma direção $d \in \mathbb{R}^n$ tal que $A_{\mathcal{E}}(\bar{x})d = 0$ e $A_{\bar{\mathcal{I}}}(\bar{x})d < 0$, onde $\bar{\mathcal{I}} = \{i \in \mathcal{I} \mid f_i(\bar{x}) = 0\}$.

H4 Dado um ponto viável não estacionário $\bar{x} \in X$, existem $M > 0$ e uma vizinhança V de \bar{x} tal que se $x^k \in V$, então

$$f_0(x^k) - f_0(x^{k+1}) \geq M v_k,$$

onde $v_k = \min \left\{ 1, \min \left\{ \tilde{h}^j \mid (\tilde{f}_0^j, \tilde{h}^j) \in F_k \right\} \right\}$.

A Hipótese H1 significa que existe uma constante $L > 0$ tal que

$$\|\nabla f_0(y) - \nabla f_0(x)\| \leq L \|y - x\|, \quad (2.2)$$

para todos $x, y \in \mathbb{R}^n$. A Hipótese H2 é comum em problemas com restrições de desigualdades e pode ser garantida adicionando restrições de caixa ao problema. A condição de qualificação assumida em H3 é usual no contexto em que estamos trabalhando. Já a Hipótese H4 define uma importante variável que chamaremos de *altura do filtro*. Nesta hipótese temos a essência da demonstração de convergência do algoritmo. Provaremos adiante que existem algoritmos para PNL que satisfazem esta hipótese.

A Figura 2.4 ilustra a altura do filtro v_k bem como a variável

$$H_k = \min \left\{ 1, \min \left\{ \tilde{h}^j \mid (\tilde{f}_0^j, \tilde{h}^j) \in F_k, \tilde{f}_0^j \leq f_0(x^k) \right\} \right\}, \quad (2.3)$$

chamada de *folga do filtro* em x^k , que foi introduzida em [23]. Esta última também será utilizada neste trabalho.

As primeiras consequências das hipóteses H1-H4 e da construção do algoritmo estão reunidas nos seguintes lemas.

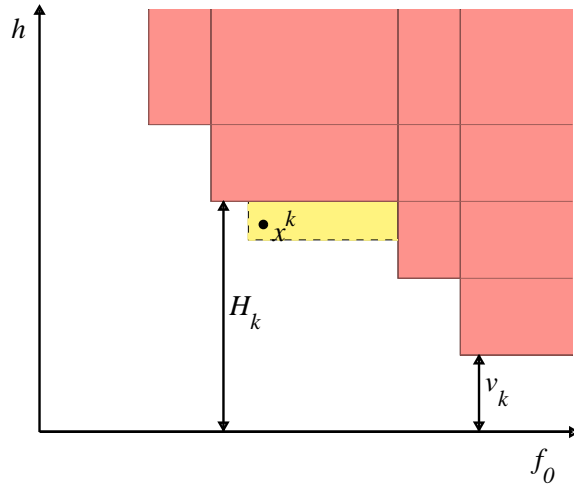


Figura 2.4: A altura e a folga do filtro, v_k e H_k .

Lema 2.2 *Suponha que sejam satisfeitas as Hipóteses H1-H4. Dado $k \in \mathbb{N}$, são válidas as seguintes afirmações:*

- (i) *Propriedade da inclusão: $\mathcal{F}_k \subset \mathcal{F}_{k+1}$.*
- (ii) *$x^{k+p} \notin \mathcal{F}_{k+1}$, para todo $p \geq 1$.*
- (iii) *Pelo menos uma, das duas condições abaixo, é satisfeita:*
 1. $f_0(x^{k+1}) < f_0(x^k) - \alpha h(x^k)$,
 2. $h(x^{k+1}) < (1 - \alpha)h(x^k)$.
- (iv) *$\tilde{h} > 0$ para todo par $(\tilde{f}_0, \tilde{h}) \in F_k$.*

Demonstração. A primeira afirmação segue diretamente do critério de atualização do filtro. Por (i) e pela definição de $\bar{\mathcal{F}}$, temos $\mathcal{F}_{k+1} \subset \mathcal{F}_{k+p} \subset \bar{\mathcal{F}}_{k+p-1}$. A segunda afirmação segue então de $x^{k+p} \notin \bar{\mathcal{F}}_{k+p-1}$. A terceira é imediata, pois $x^{k+1} \notin \bar{\mathcal{F}}_k$. Finalmente, note que se $h(x^k) = 0$, utilizando (iii), obtemos

$$f_0(x^{k+1}) < f_0(x^k) - \alpha h(x^k) = f_0(x^k),$$

ou seja, uma iteração f_0 . Assim o par (\tilde{f}_0, \tilde{h}) pode ser incluído no filtro somente se $h(x^k) > 0$, caso em que $\tilde{h} > 0$. Isto prova a quarta afirmação. \square

Para os lemas seguintes vamos considerar o conjunto das iterações h , isto é, iterações nas quais o filtro cresce, definido por

$$\mathcal{K}_a = \{k \in \mathbb{N} \mid (f_0(x^k) - \alpha h(x^k), (1 - \alpha)h(x^k)) \text{ é adicionado ao filtro}\}. \quad (2.4)$$

Lema 2.3 *Suponha que sejam satisfeitas as Hipóteses H1-H4 e que o conjunto \mathcal{K}_a seja infinito. Então*

$$h(x^k) \xrightarrow{\mathcal{K}_a} 0.$$

Demonstração. Basta provar que dado $\delta > 0$, o conjunto

$$\mathcal{K} = \{k \in \mathcal{K}_a \mid h(x^k) \geq \delta\}$$

é finito. Dado $k \in \mathbb{N}$, denotemos $(f_0(x^k), h(x^k))$ por (f_0^k, h^k) . Temos então que, no plano (f_0, h) , a bola $B((f_0^k, h^k), \alpha\delta)$ está inteiramente contida na região proibida pelo filtro permanente, para todo $k \in \mathcal{K}$. Suponha, por absurdo, que para algum $\delta > 0$ o conjunto \mathcal{K} seja infinito. Pelas Hipóteses H1 e H2, todos os termos da sequência $(f_0^k, h^k)_{k \in \mathcal{K}}$ pertencem ao conjunto compacto $f_0(X) \times h(X)$. Portanto, podemos extrair uma subsequência convergente $(f_0^k, h^k)_{k \in \mathcal{K}_1}$, $\mathcal{K}_1 \subset \mathcal{K}$, o que implica na existência de índices $j, k \in \mathcal{K}_1$, $j < k$ tais que

$$\|(f_0^k, h^k) - (f_0^j, h^j)\| < \alpha\delta,$$

contradizendo o fato de que (f_0^k, h^k) é aceito por (f_0^j, h^j) . \square

Lema 2.4 *Suponha que sejam satisfeitas as Hipóteses H1-H4. Considere $\bar{x} \in X$ um ponto não estacionário. Então existe $\bar{k} \in \mathbb{N}$ e uma vizinhança V de \bar{x} tal que se $k > \bar{k}$ e $x^k \in V$, a iteração k é do tipo f_0 , isto é, $k \notin \mathcal{K}_a$.*

Demonstração. Caso \bar{x} seja viável, a Hipótese H4 garante que existe $M > 0$ e uma vizinhança V de \bar{x} tal que se $x^k \in V$, então

$$f_0(x^k) - f_0(x^{k+1}) \geq Mv_k.$$

Pelo Lema 2.2(iv), temos $v_k > 0$. Portanto, $f_0(x^{k+1}) < f_0(x^k)$, ou seja, uma iteração f_0 . Considere agora o caso em que \bar{x} é inviável e suponha, por absurdo, que exista um conjunto infinito $\mathcal{K} \subset \mathcal{K}_a$ tal que $x^k \xrightarrow{\mathcal{K}} \bar{x}$. Como h é contínua, temos $h(x^k) \xrightarrow{\mathcal{K}} h(\bar{x})$. Por outro lado, o Lema 2.3 garante que $h(x^k) \xrightarrow{\mathcal{K}_a} 0$. Portanto $h(\bar{x}) = 0$, contradizendo o fato de \bar{x} ser inviável. \square

Nós agora podemos apresentar um dos dois principais resultados deste trabalho⁴, onde estabelecemos a convergência do Algoritmo 2.1.

Teorema 2.5 *Suponha que sejam satisfeitas as Hipóteses H1-H4. Então a sequência $(x^k)_{k \in \mathbb{N}}$ tem um ponto de acumulação estacionário.*

Demonstração. Considere o conjunto \mathcal{K}_a definido em (2.4). Caso \mathcal{K}_a seja infinito, existem $\mathcal{K}_1 \subset \mathcal{K}_a$ e $\bar{x} \in X$ tais que $x^k \xrightarrow{\mathcal{K}_1} \bar{x}$, pois, pela Hipótese H2, $(x^k)_{k \in \mathbb{N}}$ é limitada. Pelo Lema 2.4, \bar{x} deve ser estacionário. Agora o caso em que \mathcal{K}_a é finito. Então existe $k_0 \in \mathbb{N}$ tal que toda iteração $k \geq k_0$ é do tipo f_0 . Assim $(f_0(x^k))_{k \geq k_0}$ é decrescente e por H1 e H2 é também limitada. Consequentemente

$$f_0(x^k) - f_0(x^{k+1}) \rightarrow 0. \quad (2.5)$$

Além disso, pela construção do algoritmo, $F_k = F_{k_0}$ para todo $k \geq k_0$. Portanto, a sequência $(v_k)_{k \in \mathbb{N}}$, definida na Hipótese H4, satisfaz

$$v_k = v_{k_0} > 0, \quad (2.6)$$

para todo $k \geq k_0$. Se o conjunto

$$\mathcal{K}_2 = \{k \in \mathbb{N} \mid \alpha h(x^k) < f_0(x^k) - f_0(x^{k+1})\}$$

é infinito, usando (2.5), concluímos que $h(x^k) \xrightarrow{\mathcal{K}_2} 0$. Caso contrário, o Lema 2.2(iii) garante a existência de $k_1 \in \mathbb{N}$ tal que $h(x^{k+1}) < (1 - \alpha)h(x^k)$ para todo $k \geq k_1$, o que implica $h(x^k) \rightarrow 0$. De qualquer modo, $(x^k)_{k \in \mathbb{N}}$ tem um ponto de acumulação viável \bar{x} . Afirmamos que \bar{x} é estacionário. De fato, seja \mathcal{K} o conjunto de índices tal que $x^k \xrightarrow{\mathcal{K}} \bar{x}$. Se \bar{x} não fosse estacionário, concluiríamos, usando a Hipótese H4, que existem $k_2 \in \mathbb{N}$ e $M > 0$ tais que

$$f_0(x^k) - f_0(x^{k+1}) \geq Mv_k,$$

para todo $k \in \mathcal{K}$, $k \geq k_2$, o que juntamente com (2.6) contradiz (2.5). \square

Como vimos acima, a Hipótese H4 é essencial na prova de convergência do algoritmo. Esta hipótese é semelhante à que foi introduzida por Gonzaga, Karas e Vanti [23] e lembra o critério de Polak [41] para convergência global de algoritmos. Na realidade, enfraquecemos a condição de decréscimo assumida em [23], utilizando v_k ao

⁴O outro é o Teorema 2.13, que garante que a Hipótese H4 é satisfeita.

invés de H_k (veja (2.3) e a Figura 2.4). Certamente esta é a hipótese mais forte feita aqui e precisa ser justificada. Este é o propósito da próxima seção.

2.3 Algoritmos internos para a obtenção do passo

Veremos agora dois algoritmos que satisfazem a Hipótese H4. Um deles foi proposto em [23] e segue as idéias de Restauração Inexata de Martínez [32] e Martínez e Pilotta [33]. O outro algoritmo calcula o passo por PQS e integra a contribuição original deste trabalho.

2.3.1 Restauração Inexata

Aqui a obtenção do passo é feita em duas fases: dado o ponto corrente x^k , a primeira fase reduz a medida de inviabilidade⁵, obtendo um ponto z^k , não proibido pelo filtro, tal que $h(z^k) < (1 - \alpha)h(x^k)$, onde $\alpha \in (0, 1)$ é uma constante. Se z^k não for estacionário, a segunda fase procura reduzir o valor da função objetivo em relação a z^k . Uma estratégia de região de confiança é aplicada em torno de z^k para controlar o tamanho do passo. Nesta fase o passo é dado em uma aproximação tangencial do conjunto viável, para não perder demais o ganho em viabilidade obtido na primeira fase.

Do mesmo modo como em [32, 33], as fases de viabilidade e otimalidade são mais independentes em relação a outros métodos de decomposição de passo.

Fase de viabilidade. Para esta fase podemos utilizar um algoritmo que diminua h , obtendo (a partir de um ponto inviável x^k) um ponto não proibido z^k , tal que $h(z^k) < (1 - \alpha)h(x^k)$. A menos que $h(\cdot)$ tenha um ponto estacionário inviável, caso em que o método pára com insucesso, é possível executar esta fase com terminação finita. De fato, pelo Lema 2.2(iv) temos $v_k > 0$, onde v_k é a altura do filtro, definida na Hipótese H4.

Fase de otimalidade. Se z^k não for estacionário, a segunda fase tem por objetivo encontrar um ponto $z^k + t^k$, não proibido, que produza uma redução suficiente em f_0 , com relação a z^k . Este passo é dado no conjunto linearizado

$$\mathcal{L}^G(z^k) = \{z^k + t \in \mathbb{R}^n \mid A_{\mathcal{E}}(z^k)t = 0, f_{\mathcal{I}}(z^k) + A_{\mathcal{I}}(z^k)t \leq f_{\mathcal{I}}^+(z^k)\} \quad (2.7)$$

⁵Se x^k for viável, então $z^k = x^k$.

para que a redução da inviabilidade, obtida na primeira fase, não seja muito prejudicada. O algoritmo proposto em [23] para o passo de otimalidade usa região de confiança e busca uma solução do problema quadrático

$$(PT_k) \quad \begin{aligned} & \text{minimizar} && m(z^k + t) = f_0(z^k) + \nabla f_0(z^k)^T t + \frac{1}{2} t^T B_k t \\ & \text{sujeito a} && z^k + t \in \mathcal{L}^G(z^k) \\ & && \|t\| \leq \Delta, \end{aligned}$$

onde B_k é uma matriz simétrica.

Algoritmo 2.6 *Passo de otimalidade.*

Dados: $\eta \in (0, 1)$, $\Delta_{min} > 0$, $z^k \notin \bar{\mathcal{F}}_k$, $\Delta \geq \Delta_{min}$.

REPITA

Encontre t_Δ , solução “aproximada” do problema (PT_k)

Calcule $ared = f_0(z^k) - f_0(z^k + t_\Delta)$ e $pred = m_k(z^k) - m_k(z^k + t_\Delta)$

SE $z^k + t_\Delta \notin \bar{\mathcal{F}}_k$ E $ared \geq \eta pred$,

$$t^k = t_\Delta$$

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

$$\Delta_k = \Delta, \text{ pare com sucesso}$$

SENÃO,

$$\Delta = \Delta/2.$$

No Algoritmo 2.6, a solução “aproximada” significa um ponto que forneça uma redução no modelo tão boa quanto a redução obtida pelo ponto de Cauchy, que está na direção do gradiente projetado. Prova-se em [23] que a sequência gerada pelo Algoritmo 2.1 com restauração inexata satisfaz a Hipótese H4.

2.3.2 Programação Quadrática Sequencial

Apresentamos aqui um algoritmo de PQS para o cálculo do passo, que também é feito em duas fases. Na primeira fase calculamos um passo de viabilidade satisfazendo a linearização das restrições, que visa reduzir a medida de inviabilidade. Em seguida damos um passo tangencial com o propósito de produzir um decréscimo no modelo quadrático. O algoritmo proposto se baseia naquele apresentado por Fletcher *et al* [16], que por sua vez segue as idéias de Byrd [5] e Omojokun [40]. Aqui, diferentemente do que é feito em restauração inexata, a região de confiança é tomada em torno de x^k . Na Seção 2.3.3 faremos um estudo comparativo mais detalhado entre essas duas abordagens, evidenciando as semelhanças e diferenças.

O cálculo do passo. O passo é calculado por meio do subproblema quadrático com região de confiança

$$(PQ_{x^k}^{rc}) \quad \begin{array}{ll} \text{minimizar} & m_k(x^k + d) \\ \text{sujeito a} & x^k + d \in \mathcal{L}(x^k) \\ & \|d\| \leq \Delta, \end{array}$$

onde

$$m_k(x^k + d) = f_0(x^k) + \nabla f_0(x^k)^T d + \frac{1}{2} d^T B_k d, \quad (2.8)$$

com B_k simétrica, e

$$\mathcal{L}(x^k) = \{x^k + d \in \mathbb{R}^n \mid f_{\mathcal{E}}(x^k) + A_{\mathcal{E}}(x^k)d = 0, f_{\mathcal{I}}(x^k) + A_{\mathcal{I}}(x^k)d \leq 0\}. \quad (2.9)$$

Passo de viabilidade e compatibilidade de $(PQ_{x^k}^{rc})$. O passo de viabilidade n^k deve satisfazer as restrições de $(PQ_{x^k}^{rc})$ e visa reduzir a medida de inviabilidade. Uma forma de calcular n^k é dada por

$$n^k = P_k(x^k) - x^k,$$

onde P_k é a projeção ortogonal no conjunto $\mathcal{L}_k \stackrel{\text{def}}{=} \mathcal{L}(x^k)$. No entanto, não faremos uso desta escolha específica, mas sim de uma condição que garanta uma certa eficiência na fase de viabilidade. Esta condição está formalizada na Hipótese H5, adiante. Além disso, esse passo só pode ser bem aproveitado se estiver *suficientemente dentro* da região de confiança, de modo que haja espaço para o passo tangencial. Vamos então exigir que

$$\|n^k\| \leq \xi \Delta, \quad (2.10)$$

onde $\xi \in (0, 1)$ é uma constante. Quando $\mathcal{L}_k \neq \emptyset$ e a relação (2.10) for satisfeita, diremos que o subproblema $(PQ_{x^k}^{rc})$ é *compatível*.

Para o que segue vamos definir

$$z^k = x^k + n^k \quad (2.11)$$

o ponto obtido na fase de viabilidade. Note que, usando (2.8) e (2.11), temos

$$m_k(z^k) = m_k(x^k + n^k) = f_0(x^k) + \nabla f_0(x^k)^T n^k + \frac{1}{2} n^{kT} B_k n^k. \quad (2.12)$$

Passo tangencial. No caso em que o subproblema $(PQ_{x^k}^{rc})$ é compatível, esperamos uma redução razoável do modelo no passo tangencial t_Δ , solução (aproximada) do problema quadrático

$$(PT_k^{rc}) \quad \begin{aligned} & \text{minimizar} && (\nabla f_0(x^k) + B_k n^k)^T t + \frac{1}{2} t^T B_k t \\ & \text{sujeito a} && A_\mathcal{E}(x^k) t = 0 \\ & && f_\mathcal{I}(x^k) + A_\mathcal{I}(x^k)(n^k + t) \leq 0 \\ & && \|n^k + t\| \leq \Delta, \end{aligned}$$

que é, a menos de um termo constante, equivalente a $(PQ_{x^k}^{rc})$ com $d = n^k + t$.

Fase de restauração. Se o subproblema $(PQ_{x^k}^{rc})$ não for compatível, o algoritmo chama um *procedimento de restauração* com o objetivo de obter um ponto $x^{k+1} \notin \bar{\mathcal{F}}_k$. Isto pode ser feito resolvendo-se o problema

$$\begin{aligned} & \text{minimizar} && h(x) \\ & && x \in \mathbb{R}^n \end{aligned},$$

onde a função h é a medida de inviabilidade definida em (2.1).

A Figura 2.5 ilustra uma iteração completa do algoritmo, onde consideramos um problema em \mathbb{R}^2 com uma restrição de igualdade. O conjunto viável e a sua linearização estão representados por linhas cheias, enquanto que as curvas de nível da função objetivo são as linhas tracejadas. Também representamos nesta figura a região de confiança $\{x^k + d \mid \|d\| \leq \Delta_k\}$ (circunferência maior) e a bola definida por (2.10) (circunferência menor), que limita o tamanho do passo de viabilidade para garantir uma redução razoável do modelo no passo tangencial.

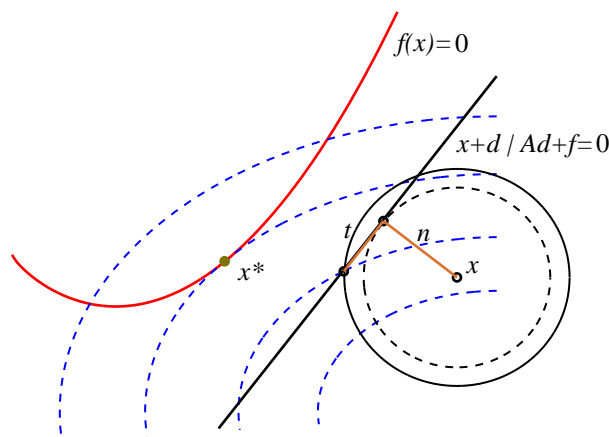


Figura 2.5: Uma iteração de PQS.

Podemos agora resumir a discussão acima no seguinte algoritmo para o cálculo do passo.

Algoritmo 2.7 *Cálculo de $x^{k+1} \notin \bar{\mathcal{F}}_k$*

Dados: $x^k \in \mathbb{R}^n$, $\bar{\mathcal{F}}_k$, $\Delta_{\min} > 0$, $\Delta \geq \Delta_{\min}$, $c_p \in (0, 1)$, $\eta \in (0, 1)$, $\gamma \in (0, 1)$.

SE $\mathcal{L}_k = \emptyset$,

faça uma restauração obtendo $x^{k+1} \notin \bar{\mathcal{F}}_k$.

SENÃO

calcule o passo de viabilidade n^k

REPITA (enquanto o ponto x^{k+1} não for obtido)

SE $\|n^k\| > \xi\Delta$,

faça uma restauração obtendo $x^{k+1} \notin \bar{\mathcal{F}}_k$.

obtenha B_{k+1} simétrica

SENÃO,

calcule o passo tangencial t_Δ e defina $d_\Delta = n^k + t_\Delta$

$ared = f_0(x^k) - f_0(x^k + d_\Delta)$ e $pred = m_k(x^k) - m_k(x^k + d_\Delta)$

SE $\{x^k + d_\Delta \in \bar{\mathcal{F}}_k\}$ OU $\{pred \geq c_p(h(x^k))^2 \text{ E } ared < \eta pred\}$

$\Delta = \gamma\Delta$

SENÃO

$x^{k+1} = x^k + d_\Delta$

obtenha B_{k+1} simétrica

$\Delta_k = \Delta$

O Algoritmo 2.7 foi inspirado no algoritmo de filtro com PQS proposto por Fletcher *et al* [16]. Entretanto, existem algumas diferenças que salientamos agora. Uma delas é que aqui o cálculo do passo é separado do algoritmo principal de filtro, apresentado na Seção 2.1. Isto facilita o estudo das propriedades do passo, deixando o algoritmo principal e a prova de convergência em um formato mais limpo. Outra diferença reside no raio da região de confiança. Cada vez que o Algoritmo 2.7 é chamado para calcular o passo, é tomado um raio inicial $\Delta \geq \Delta_{\min}$, onde $\Delta_{\min} > 0$ é uma constante. Este recurso não é usado em [16], criando uma dificuldade na prova de convergência do algoritmo: próximo de um ponto viável não estacionário se precisava garantir que o passo normal era muito menor que o passo tangencial. Para contornar esta dificuldade, eles impõe uma condição do tipo

$$\|n^k\| \leq c\Delta^{1+\mu}, \quad (2.13)$$

com $c > 0$ e $\mu \in (0, 1)$, para aceitar o passo normal e proceder com o cálculo do passo tangencial. Se a condição acima não for satisfeita, o subproblema $(PQ_{x^k}^{rc})$ é considerado incompatível e o algoritmo recorre para uma fase de restauração com o propósito de obter um novo ponto não proibido pelo filtro, satisfazendo (2.13).

No algoritmo que propomos, esta condição é substituída pela relação (2.10),

$$\|n^k\| \leq \xi \Delta,$$

onde $\xi \in (0, 1)$ é uma constante. Esta exigência é comum nas abordagens de decomposição de passo que estamos considerando.

Observações. Representamos por d_Δ o passo tentativo obtido quando o raio da região de confiança é $\Delta \geq \Delta_k$. Deste modo, o ponto x^{k+1} pode ser calculado de duas maneiras: através de uma restauração ou por $x^{k+1} = x^k + d_{\Delta_k}$. O passo tentativo d_Δ pode ser recusado pelo algoritmo em duas situações:

$$x^k + d_\Delta \in \bar{\mathcal{F}}_k \tag{2.14}$$

ou

$$pred \geq c_p (h(x^k))^2 \text{ e } ared < \eta pred, \tag{2.15}$$

casos em que o raio da região de confiança é reduzido e um novo passo é calculado. Assim, para que o passo d_Δ seja aceito, não basta passar pelo critério do filtro. Ele deve também garantir um decréscimo suficiente na função objetivo quando a redução predita pelo modelo for mais significativa que a violação das restrições. Em particular, se todos os iterandos forem viáveis, a primeira desigualdade em (2.15) será sempre verdadeira, pois $n^k = 0$ neste caso. Além disso, se $ared \geq \eta pred$, então $x^k + d_\Delta \notin \bar{\mathcal{F}}_k$. Deste modo o critério de aceitação do passo fica apenas $ared \geq \eta pred$ e o algoritmo se reduz a um método clássico de região de confiança irrestrito.

Para estabelecer o nosso resultado, isto é, que a Hipótese H4 é satisfeita quando o Algoritmo 2.1 é aplicado ao problema (P) e o passo é obtido pelo Algoritmo 2.7, vamos introduzir uma função que será utilizada como medida de estacionaridade. Dados $x, z \in X$ e o conjunto $\mathcal{L}(x)$ definido em (2.9), chamaremos o vetor

$$d^c(x, z) = P_{\mathcal{L}(x)}(z - \nabla f_0(x)) - z \tag{2.16}$$

de *direção do gradiente projetado* e usaremos a função $\varphi : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$, dada por

$$\varphi(x, z) = -\nabla f_0(x)^T \frac{d^c(x, z)}{\|d^c(x, z)\|}, \quad (2.17)$$

como medida de estacionaridade. De acordo com [23] temos que em um ponto viável \bar{x} , as condições de KKT são equivalentes a $d^c(\bar{x}, \bar{x}) = 0$. Além disso, se \bar{x} não é estacionário, vale $\varphi(\bar{x}, \bar{x}) > 0$.

A direção do gradiente projetado dada acima se baseia em uma direção introduzida por Martínez e Svaiter [35] para definir uma nova condição de otimalidade, chamada de *propriedade AGP* (Approximate Gradient Projection), que fica entre KKT e Fritz-John. Esta propriedade é satisfeita por minimizadores locais de problemas de otimização com restrições, independentemente de condições de qualificação.

Hipóteses relacionadas com o Algoritmo 2.7. Vamos assumir que são satisfeitas as seguintes condições.

H5 Existe uma constante $c_n > 0$ tal que $\|n^k\| \leq c_n h(x^k)$, para todo $k \geq 0$.

H6 Se o subproblema $(PQ_{x^k}^{r^c})$ for compatível, então a redução do modelo no passo tangencial satisfaz

$$m_k(z^k) - m_k(z^k + t_\Delta) \geq m_k(z^k) - m_k(z^k + t^c),$$

com $t^c = \operatorname{argmin} \{m_k(z^k + \lambda d_1^c) \mid \|z^k + \lambda d_1^c - x^k\| \leq \Delta, \lambda \geq 0\}$, $d_1^c = \frac{d^c(x^k, z^k)}{\|d^c(x^k, z^k)\|}$.

H7 As hessianas B_k são uniformemente limitadas, isto é, existe uma constante $\beta > 1$ tal que $\|B_k\| \leq \beta$, para todo $k \geq 0$.

A Hipótese H5 significa que o passo de viabilidade deve ser razoavelmente escalado em relação às restrições. Em particular, $n^k = 0$ quando x^k é viável. Esta hipótese é discutida com mais detalhes em [32], onde é apresentado um algoritmo de viabilidade satisfazendo tal hipótese. A Hipótese H6 diz que o passo tangencial deve ser pelo menos tão bom quanto o *passo de Cauchy generalizado*, t^c .

Os lemas que seguem constituem os degraus para chegarmos ao teorema principal. No primeiro lema fazemos uma estimativa da medida de inviabilidade antes e depois de dar o passo tentativo.

Lema 2.8 *Suponha que sejam satisfeitas as Hipóteses H1-H3 e que $(PQ_{x^k}^{rc})$ seja compatível. Então existe uma constante $c_h > 0$ tal que*

$$h(x^k) \leq c_h \Delta \quad e \quad h(x^k + d_\Delta) \leq c_h \Delta^2,$$

onde d_Δ é o passo tentativo obtido no Algoritmo 2.7.

Demonstração. Pela equivalência de normas, podemos considerar, sem perda de generalidade, a norma do máximo na definição de h , dada em (2.1). Temos então $h(x^k) = |f_i(x^k)|$ para algum $i \in \mathcal{E}$, ou $h(x^k) = f_i(x^k)$ para algum $i \in \mathcal{I}$. Utilizando as relações (2.9) e (2.11), juntamente com a desigualdade de Cauchy-Schwarz, obtemos

$$h(x^k) = |f_i(x^k)| = |-\nabla f_i(x^k)^T n^k| \leq \|\nabla f_i(x^k)\| \|n^k\|$$

no primeiro caso, e

$$h(x^k) = f_i(x^k) \leq -\nabla f_i(x^k)^T n^k \leq \|\nabla f_i(x^k)\| \|n^k\|$$

no segundo caso. Pelas Hipóteses H1 e H2 mais o fato de que $(PQ_{x^k}^{rc})$ é compatível, podemos garantir que existe uma constante $c_1 > 0$ tal que

$$h(x^k) \leq c_1 \Delta. \quad (2.18)$$

Por outro lado, lembrando que d_Δ satisfaz as restrições de $(PQ_{x^k}^{rc})$, temos:

$$f_i(x^k + d_\Delta) = f_i(x^k) + \nabla f_i(x^k)^T d_\Delta + \frac{1}{2} d_\Delta^T \nabla^2 f_i(x^k + \theta_k d_\Delta) d_\Delta = \frac{1}{2} d_\Delta^T \nabla^2 f_i(x^k + \theta_k d_\Delta) d_\Delta$$

se $i \in \mathcal{E}$, ou

$$f_i(x^k + d_\Delta) = f_i(x^k) + \nabla f_i(x^k)^T d_\Delta + \frac{1}{2} d_\Delta^T \nabla^2 f_i(x^k + \theta_k d_\Delta) d_\Delta \leq \frac{1}{2} d_\Delta^T \nabla^2 f_i(x^k + \theta_k d_\Delta) d_\Delta,$$

se $i \in \mathcal{I}$, onde $\theta_k \in (0, 1)$. Deste modo, como $h(x^k + d_\Delta) = |f_i(x^k + d_\Delta)|$ para algum $i \in \mathcal{E}$ ou $h(x^k + d_\Delta) = f_i(x^k + d_\Delta)$ para algum $i \in \mathcal{I}$, podemos concluir, usando H1 e H2, que existe uma constante $c_2 > 0$ tal que

$$h(x^k + d_\Delta) \leq c_2 \Delta^2. \quad (2.19)$$

Portanto, definindo $c_h = \max\{c_1, c_2\}$ e utilizando (2.18) e (2.19), completamos a demonstração. \square

O próximo lema avalia o quanto o modelo e a função objetivo podem piorar quando é dado o passo de viabilidade.

Lema 2.9 *Suponha que sejam satisfeitas as Hipóteses H1-H3, H5 e H7. Dado $\bar{x} \in X$ viável, existem $N > 0$ e uma vizinhança V_1 de \bar{x} tais que se $x^k \in V_1$, então*

$$(i) \quad |m_k(x^k) - m_k(z^k)| \leq Nh(x^k).$$

$$(ii) \quad |f_0(x^k) - f_0(z^k)| \leq Nh(x^k).$$

Demonstração. Seja $c_g = \max \{ \|\nabla f_0(x)\| \mid x \in X \}$. Por (2.12) temos

$$m_k(z^k) = m_k(x^k) + \nabla f_0(x^k)^T n^k + \frac{1}{2} n^{kT} B_k n^k.$$

Assim, usando a desigualdade de Cauchy-Schwarz, H5 e H7, obtemos

$$\begin{aligned} |m_k(x^k) - m_k(z^k)| &\leq \|\nabla f_0(x^k)\| \|n^k\| + \frac{1}{2} \|B_k\| \|n^k\|^2 \\ &\leq c_g \|n^k\| + \frac{1}{2} \beta \|n^k\|^2 \\ &\leq c_g c_n h(x^k) + \frac{1}{2} \beta c_n^2 (h(x^k))^2. \end{aligned}$$

Como $h(\bar{x}) = 0$ e h é contínua, podemos tomar uma vizinhança V_1 de \bar{x} tal que se $x^k \in V_1$, então

$$\frac{1}{2} \beta c_n^2 (h(x^k))^2 \leq c_g c_n h(x^k)$$

e conseqüentemente

$$|m_k(x^k) - m_k(z^k)| \leq 2c_g c_n h(x^k).$$

Por outro lado, pelas Hipóteses H1 e H2, ∇f_0 é contínua e x^k, z^k estão em um compacto. Portanto existe $L > 0$ tal que

$$|f_0(x^k) - f_0(z^k)| \leq L \|x^k - z^k\|,$$

que juntamente com H5 fornece

$$|f_0(x^k) - f_0(z^k)| \leq L c_n h(x^k).$$

Tomando $N = \max \{2c_g c_n, L c_n\}$ completamos a demonstração. \square

Vamos agora provar que os decréscimos do modelo e da função objetivo são grandes quando estamos perto de um ponto viável não estacionário e o filtro não atrapalha. O primeiro lema considera apenas o passo tangencial, enquanto que o segundo olha para o passo inteiro (de viabilidade e tangencial).

Lema 2.10 *Suponha que sejam satisfeitas as Hipóteses H1-H3, H5-H7 e que $(PQ_{x^k}^{rc})$ seja compatível. Dados $\bar{x} \in X$ viável não estacionário e $\bar{\eta} \in (0, 1)$, considere V_1 a vizinhança dada no Lema 2.9 e Δ_{\min} a constante dada no Algoritmo 2.7. Então existem $\Delta_\rho \in (0, \Delta_{\min}]$, $\tilde{c} > 0$ e uma vizinhança $V_2 \subset V_1$ de \bar{x} tais que se $x^k \in V_2$ e t_Δ é o passo tentativo tangencial obtido no Algoritmo 2.7, temos*

$$(i) \quad m_k(z^k) - m_k(z^k + t_\Delta) \geq \tilde{c}\Delta, \text{ para todo } \Delta \in (0, \Delta_\rho].$$

$$(ii) \quad m_k(z^k) - m_k(z^k + t_\Delta) \geq \tilde{c}\Delta', \text{ para todos } \Delta, \Delta' \text{ tais que } \Delta' \leq \min\{\Delta, \Delta_\rho\}.$$

$$(iii) \quad f_0(z^k) - f_0(z^k + t_\Delta) \geq \bar{\eta} (m_k(z^k) - m_k(z^k + t_\Delta)), \text{ para todo } \Delta \in (0, \Delta_\rho].$$

Demonstração. Seja $\zeta = 1 - \xi$, onde ξ é a constante dada em (2.10). Usando H6, (2.8) e (2.11) temos

$$\begin{aligned} m_k(z^k) - m_k(z^k + t_\Delta) &\geq m_k(z^k) - m_k(z^k + t^c) \\ &\geq m_k(z^k) - m_k(z^k + \zeta \Delta d_1^c) \\ &= \nabla f_0(x^k)^T n^k + \frac{1}{2} n^{kT} B_k n^k - \nabla f_0(x^k)^T (n^k + \zeta \Delta d_1^c) \\ &\quad - \frac{1}{2} (n^k + \zeta \Delta d_1^c)^T B_k (n^k + \zeta \Delta d_1^c) \\ &= \zeta \Delta \left[-\nabla f_0(x^k)^T d_1^c - n^{kT} B_k d_1^c - \frac{1}{2} \zeta \Delta d_1^{cT} B_k d_1^c \right], \end{aligned}$$

que por H7 e (2.17) implica

$$m_k(z^k) - m_k(z^k + t_\Delta) \geq \zeta \Delta \left[\varphi(x^k, z^k) - \|n^k\| \beta - \frac{1}{2} \zeta \Delta \beta \right]. \quad (2.20)$$

Como \bar{x} é viável e não estacionário, a função contínua φ , definida em (2.17), satisfaz $\varphi(\bar{x}, \bar{x}) > 0$. Seja $\bar{V}_1 \subset V_1$ uma vizinhança de \bar{x} tal que

$$\varphi(x, z) \geq \frac{1}{2} \varphi(\bar{x}, \bar{x}), \quad (2.21)$$

para todos $x, z \in \bar{V}_1$. Pela Hipótese H5 temos $\|x^k - z^k\| \leq c_n h(x^k)$, o que permite considerar uma vizinhança $V_2 \subset \bar{V}_1$ tal que se $x^k \in V_2$, então $z^k \in \bar{V}_1$. Além disso, podemos restringir V_2 , se necessário, e escolher $\Delta_\rho^0 \in (0, \Delta_{\min}]$ tal que se $x^k \in V_2$ e $\Delta \in (0, \Delta_\rho^0]$, então

$$\|n^k\| \beta + \frac{1}{2} \zeta \Delta \beta \leq \frac{1}{4} \varphi(\bar{x}, \bar{x}). \quad (2.22)$$

Portanto, combinando (2.20), (2.21) e (2.22), obtemos

$$m_k(z^k) - m_k(z^k + t_\Delta) \geq \frac{1}{4} \zeta \varphi(\bar{x}, \bar{x}) \Delta,$$

se $x^k \in V_2$ e $\Delta \in (0, \Delta_\rho^0]$. Isto demonstra (i) para $\Delta_\rho \leq \Delta_\rho^0$ e $\tilde{c} = \frac{1}{4}\zeta\varphi(\bar{x}, \bar{x})$. Analogamente, como $\Delta' \leq \Delta$, temos

$$\begin{aligned} m_k(z^k) - m_k(z^k + t_\Delta) &\geq m_k(z^k) - m_k(z^k + t^c) \\ &\geq m_k(z^k) - m_k(z^k + \zeta\Delta'd_1^c) \\ &= \zeta\Delta' \left[-\nabla f_0(x^k)^T d_1^c - n^{kT} B_k d_1^c - \frac{1}{2}\zeta\Delta'd_1^{cT} B_k d_1^c \right]. \end{aligned}$$

Deste modo, se $x^k \in V_2$, podemos usar (2.21), o fato de que $\Delta' \leq \Delta_\rho$ e (2.22) para concluir que

$$m_k(z^k) - m_k(z^k + t_\Delta) \geq \tilde{c}\Delta',$$

donde segue (ii). Para demonstrar (iii) note que, pelo Teorema do Valor Médio,

$$ared_{z^k} \stackrel{\text{def}}{=} f_0(z^k) - f_0(z^k + t_\Delta) = -\nabla f_0(z^k + \theta t_\Delta)^T t_\Delta,$$

sendo $\theta \in (0, 1)$. Por outro lado,

$$pred_{z^k} \stackrel{\text{def}}{=} m_k(z^k) - m_k(z^k + t_\Delta) = -\nabla f_0(x^k)^T t_\Delta - n^{kT} B_k t_\Delta - \frac{1}{2}t_\Delta^T B_k t_\Delta.$$

Deste modo, usando (2.2) e H7, obtemos

$$\begin{aligned} |ared_{z^k} - pred_{z^k}| &= \left| (\nabla f_0(x^k) - \nabla f_0(z^k + \theta t_\Delta))^T t_\Delta + n^{kT} B_k t_\Delta + \frac{1}{2}t_\Delta^T B_k t_\Delta \right| \\ &\leq L \|z^k - x^k + \theta t_\Delta\| \|t_\Delta\| + \beta \|n^k\| \|t_\Delta\| + \frac{1}{2}\beta \|t_\Delta\|^2 \\ &\leq L \|n^k\| \Delta + L\Delta^2 + \beta \|n^k\| \Delta + \frac{1}{2}\beta\Delta^2 \\ &= (L + \beta) \|n^k\| \Delta + (L + \frac{1}{2}\beta) \Delta^2. \end{aligned}$$

Podemos restringir a vizinhança V_2 , se necessário, e tomar $\Delta_\rho \leq \Delta_\rho^0$ de modo que se $x^k \in V_2$ e $\Delta \in (0, \Delta_\rho]$, então

$$\frac{(L + \beta) \|n^k\|}{\tilde{c}} \leq \frac{1 - \bar{\eta}}{2} \quad \text{e} \quad \frac{(L + \frac{1}{2}\beta) \Delta}{\tilde{c}} \leq \frac{1 - \bar{\eta}}{2},$$

e consequentemente, usando (i),

$$\left| \frac{ared_{z^k}}{pred_{z^k}} - 1 \right| = \left| \frac{ared_{z^k} - pred_{z^k}}{pred_{z^k}} \right| \leq \frac{(L + \beta) \|n^k\| \Delta + (L + \frac{1}{2}\beta) \Delta^2}{\tilde{c}\Delta} \leq 1 - \bar{\eta},$$

completando a demonstração. \square

Lema 2.11 *Suponha que sejam satisfeitas as Hipóteses H1-H3, H5-H7 e que $(PQ_{x^k}^{rc})$ seja compatível. Dados $\bar{x} \in X$ viável não estacionário e $0 < \eta < \bar{\eta} < 1$, considere a constante γ dada no Algoritmo 2.7, a vizinhança V_2 e a constante Δ_ρ dadas no Lema 2.10. Então existe uma vizinhança $V_3 \subset V_2$ de \bar{x} tais que se $x^k \in V_3$ e t_Δ é o passo tentativo tangencial obtido no Algoritmo 2.7, temos*

$$(i) \quad m_k(x^k) - m_k(z^k + t_\Delta) \geq \frac{1}{2}\tilde{c}\Delta,$$

$$(ii) \quad f_0(x^k) - f_0(z^k + t_\Delta) \geq \eta (m_k(x^k) - m_k(z^k + t_\Delta)),$$

para todo $\Delta \in [\gamma^2\Delta_\rho, \Delta_\rho]$.

Demonstração. Sejam N e \tilde{c} as constantes dadas nos Lemas 2.9 e 2.10, respectivamente, $\tau = \frac{\bar{\eta} - \eta}{\bar{\eta} + \eta}$ e $V_3 \subset V_2$ uma vizinhança de \bar{x} tal que

$$Nh(x) \leq \min \left\{ \frac{1}{2}\tilde{c}\gamma^2\Delta_\rho, \tau\bar{\eta}\tilde{c}\gamma^2\Delta_\rho \right\}, \quad (2.23)$$

para todo $x \in V_3$. Assim, se $x^k \in V_3$ e $\Delta \in [\gamma^2\Delta_\rho, \Delta_\rho]$, podemos aplicar os Lemas 2.9 e 2.10 para concluir que

$$|m_k(x^k) - m_k(z^k)| \leq Nh(x^k) \leq \frac{1}{2}\tilde{c}\gamma^2\Delta_\rho \leq \frac{1}{2}\tilde{c}\Delta$$

e

$$m_k(x^k) - m_k(z^k + t_\Delta) = m_k(x^k) - m_k(z^k) + m_k(z^k) - m_k(z^k + t_\Delta) \geq \frac{1}{2}\tilde{c}\Delta.$$

Isto demonstra (i). Para demonstrar (ii) vamos aplicar novamente os Lemas 2.9 e 2.10, obtendo

$$|f_0(x^k) - f_0(z^k)| \leq Nh(x^k) \leq \tau\bar{\eta}\tilde{c}\gamma^2\Delta_\rho \leq \tau\bar{\eta}\tilde{c}\Delta \leq \tau (f_0(z^k) - f_0(z^k + t_\Delta))$$

e conseqüentemente

$$\begin{aligned} f_0(x^k) - f_0(z^k + t_\Delta) &= f_0(x^k) - f_0(z^k) + f_0(z^k) - f_0(z^k + t_\Delta) \\ &\geq (1 - \tau) (f_0(z^k) - f_0(z^k + t_\Delta)). \end{aligned} \quad (2.24)$$

Estes mesmos lemas também implicam

$$m_k(x^k) - m_k(z^k) \leq Nh(x^k) \leq \tau\tilde{c}\gamma^2\Delta_\rho \leq \tau\tilde{c}\Delta \leq \tau (m_k(z^k) - m_k(z^k + t_\Delta)).$$

Portanto

$$\begin{aligned} m_k(x^k) - m_k(z^k + t_\Delta) &= m_k(x^k) - m_k(z^k) + m_k(z^k) - m_k(z^k + t_\Delta) \\ &\leq (1 + \tau) (m_k(z^k) - m_k(z^k + t_\Delta)). \end{aligned} \quad (2.25)$$

Deste modo, se $x^k \in V_3$ e $\Delta \in [\gamma^2 \Delta_\rho, \Delta_\rho]$, podemos usar (2.24), Lema 2.10(iii) e (2.25) para obter

$$\begin{aligned} f_0(x^k) - f_0(z^k + t_\Delta) &\geq (1 - \tau) \bar{\eta} (m_k(z^k) - m_k(z^k + t_\Delta)) \\ &\geq \frac{(1 - \tau) \bar{\eta}}{(1 + \tau)} (m_k(x^k) - m_k(z^k + t_\Delta)) \\ &= \eta (m_k(x^k) - m_k(z^k + t_\Delta)), \end{aligned}$$

completando a demonstração. \square

Para os dois últimos resultados vamos utilizar a *folga do filtro* H_k , definida em (2.3). Provaremos primeiro que, perto de um ponto viável não estacionário, a recusa de um passo se deve ao fato de que a inviabilidade aumentou muito.

Lema 2.12 *Suponha que sejam satisfeitas as Hipóteses H1-H3, H5-H7 e que $(PQ_{x^k}^{rc})$ seja compatível. Dado $\bar{x} \in X$ viável não estacionário, considere as constantes γ e Δ_ρ dadas no Algoritmo 2.7 e no Lema 2.10, respectivamente e a vizinhança V_3 dada no Lema 2.11. Então existe uma vizinhança $V \subset V_3$ de \bar{x} tais que se $x^k \in V$ e t_Δ é o passo tentativo tangencial obtido no Algoritmo 2.7, temos*

$$h(z^k + t_\Delta) \geq H_k,$$

para todo $\Delta \in [\gamma^2 \Delta_\rho, \Delta_\rho]$ que foi recusado pelo Algoritmo 2.7.

Demonstração. Sejam α , η , N e \tilde{c} as constantes dadas nos Algoritmos 2.1 e 2.7 e nos Lemas 2.9 e 2.10, respectivamente. Considere $V \subset V_3$ uma vizinhança de \bar{x} tal que

$$Nh(x) \leq \frac{1}{2} \tilde{c} \gamma^2 \Delta_\rho \quad \text{e} \quad \alpha h(x) \leq \frac{1}{2} \eta \tilde{c} \gamma^2 \Delta_\rho, \quad (2.26)$$

para todo $x \in V$. Assim, se $x^k \in V$ e $\Delta \in [\gamma^2 \Delta_\rho, \Delta_\rho]$, podemos aplicar o Lema 2.9 para concluir que

$$|m_k(x^k) - m_k(z^k)| \leq Nh(x^k) \leq \frac{1}{2} \tilde{c} \gamma^2 \Delta_\rho \leq \frac{1}{2} \tilde{c} \Delta,$$

que juntamente com o Lema 2.10 fornece

$$m_k(x^k) - m_k(z^k + t_\Delta) \geq \frac{1}{2}\tilde{c}\Delta \geq \frac{1}{2}\tilde{c}\gamma^2\Delta_\rho. \quad (2.27)$$

Usando agora o Lema 2.11, (2.27) e a segunda desigualdade de (2.26), obtemos

$$\begin{aligned} f_0(x^k) - f_0(z^k + t_\Delta) &\geq \eta (m_k(x^k) - m_k(z^k + t_\Delta)) \\ &\geq \frac{1}{2}\eta\tilde{c}\gamma^2\Delta_\rho \\ &\geq \alpha h(x^k). \end{aligned}$$

Como $z^k + t_\Delta = x^k + n^k + t_\Delta = x^k + d_\Delta$, podemos concluir imediatamente que

$$f_0(x^k) - f_0(x^k + d_\Delta) \geq \eta (m_k(x^k) - m_k(x^k + d_\Delta))$$

e

$$f_0(x^k + d_\Delta) \leq f_0(x^k) - \alpha h(x^k).$$

Se além disso o passo tentativo foi recusado pelo Algoritmo 2.7, devemos ter necessariamente

$$h(z^k + t_\Delta) \geq H_k,$$

completando a demonstração. \square

Vamos agora apresentar o outro resultado principal deste trabalho⁶: *A Hipótese H4 é satisfeita.* Na realidade provaremos uma condição suficiente para garantir a Hipótese H4. Conforme vimos no Teorema 2.5, este resultado consiste na principal ferramenta que permite provar a convergência do Algoritmo 2.1.

Para o que segue, lembramos que o conjunto \mathcal{L}_k é definido em (2.9) e definimos o conjunto \mathcal{K}_r , das iterações onde houve restauração, por

$$\mathcal{K}_r = \{k \in \mathbb{N} \mid \mathcal{L}_k = \emptyset \text{ ou } \|n^k\| > \xi\Delta_k\}. \quad (2.28)$$

Teorema 2.13 *Suponha que o Algoritmo 2.1 é aplicado ao problema (P), com o passo obtido no Algoritmo 2.7, e que sejam satisfeitas as Hipóteses H1-H3, H5-H7. Dado um ponto viável não estacionário $\bar{x} \in X$, existem $M > 0$ e uma vizinhança V de \bar{x} tais que se $x^k \in V$, então*

$$f_0(x^k) - f_0(x^{k+1}) \geq M\sqrt{H_k}.$$

⁶O primeiro foi o Teorema 2.5, provado na Seção 2.2.

Em particular, como $\sqrt{H_k} \geq v_k$, a Hipótese H4 se verifica.

Demonstração. Seja \bar{x} um ponto viável não estacionário. Considere a vizinhança V dada no Lema 2.12 e a constante Δ_ρ dada no Lema 2.10. Sem perda de generalidade podemos supor que

$$\Delta_\rho \leq \min \left\{ \frac{\xi\gamma^2}{c_n c_h}, \frac{\tilde{c}\gamma^2}{2Nc_h}, \frac{\tilde{c}\gamma^2}{2c_p c_h} \right\}, \quad (2.29)$$

sendo ξ a constante dada em (2.10), γ e c_p dadas no Algoritmo 2.7, c_n dada na Hipótese H5 e c_h , N e \tilde{c} dadas nos Lemas 2.8, 2.9 e 2.10, respectivamente. Pela Hipótese H3, podemos assumir que se $x^k \in V$, então $\mathcal{L}_k \neq \emptyset$. Deste modo, o Algoritmo 2.7 começa com um raio $\Delta \geq \Delta_{\min}$ e termina com o raio $\Delta_k = \gamma^{nr} \Delta$, onde nr é o número de vezes que o raio foi reduzido no loop do algoritmo.

Vamos considerar dois casos, sendo o primeiro $\Delta_k \geq \gamma^2 \Delta_\rho$. Neste caso, usando a Hipótese H5 e restringindo a vizinhança V se necessário, temos

$$\|n^k\| \leq c_n h(x^k) \leq \xi \gamma^2 \Delta_\rho \leq \xi \Delta_k, \quad (2.30)$$

o que significa que o Algoritmo 2.7 não chama a fase de restauração durante a iteração k , isto é, $k \notin \mathcal{K}_r$. Portanto, aplicando o Lema 2.10(ii) com $\Delta' = \gamma^2 \Delta_\rho$, obtemos

$$m_k(z^k) - m_k(x^{k+1}) = m_k(z^k) - m_k(z^k + t_{\Delta_k}) \geq \tilde{c}\gamma^2 \Delta_\rho. \quad (2.31)$$

Por outro lado, pelo Lema 2.9(i), existe uma constante $N > 0$ tal que

$$|m_k(x^k) - m_k(z^k)| \leq N h(x^k). \quad (2.32)$$

Podemos restringir novamente a vizinhança V se necessário, de modo que

$$N h(x^k) \leq \frac{1}{2} \tilde{c}\gamma^2 \Delta_\rho, \quad c_p (h(x^k))^2 \leq \frac{1}{2} \tilde{c}\gamma^2 \Delta_\rho \quad \text{e} \quad h(x^k) \leq 1. \quad (2.33)$$

A primeira desigualdade em (2.33) junto com (2.31) e (2.32) fornece

$$pred_k \stackrel{\text{def}}{=} m_k(x^k) - m_k(x^{k+1}) = m_k(x^k) - m_k(z^k) + m_k(z^k) - m_k(x^{k+1}) \geq \frac{1}{2} \tilde{c}\gamma^2 \Delta_\rho,$$

que implica $pred_k \geq c_p (h(x^k))^2$ em virtude da segunda desigualdade em (2.33). Pelo mecanismo do Algoritmo 2.7, temos então que

$$f_0(x^k) - f_0(x^{k+1}) \stackrel{\text{def}}{=} ared_k \geq \eta pred_k = \frac{1}{2} \eta \tilde{c}\gamma^2 \Delta_\rho \geq \frac{1}{2} \eta \tilde{c}\gamma^2 \Delta_\rho \sqrt{H_k}, \quad (2.34)$$

pois $H_k \leq 1$.

Considere agora o segundo caso, isto é, $\Delta_k < \gamma^2 \Delta_\rho$. Neste caso, novamente vamos analisar duas possibilidades. Na primeira vamos supor que $h(x^k + d_\Delta) \geq H_k$, para todo $\Delta \leq \gamma \Delta_\rho$ tal que o passo tentativo d_Δ foi calculado. Como

$$\frac{\Delta_k}{\gamma} < \gamma \Delta_\rho < \Delta_{\min},$$

existiu o passo tentativo $\tilde{d} = d_{\tilde{\Delta}}$, obtido com o raio $\tilde{\Delta} = \frac{\Delta_k}{\gamma}$, e $h(x^k + \tilde{d}) \geq H_k$. Por outro lado, pelo Lema 2.8, temos $h(x^k + \tilde{d}) \leq c_h \tilde{\Delta}^2$, donde segue que

$$c_h \Delta_k^2 = c_h \gamma^2 \tilde{\Delta}^2 \geq \gamma^2 h(x^k + \tilde{d}) \geq \gamma^2 H_k \geq \gamma^2 h(x^k) \quad (2.35)$$

Usando a Hipótese H5, (2.29) e (2.35), obtemos

$$\|n^k\| \leq c_n h(x^k) \leq \frac{c_n c_h}{\gamma^2} \Delta_k^2 \leq \xi \Delta_k, \quad (2.36)$$

o que significa que o Algoritmo 2.7 não chama a fase de restauração durante a iteração k , isto é, $k \notin \mathcal{K}_r$. Portanto, o passo t_{Δ_k} foi calculado e, pelo Lema 2.10(i), satisfaz

$$m_k(z^k) - m_k(x^{k+1}) = m_k(z^k) - m_k(z^k + t_{\Delta_k}) \geq \tilde{c} \Delta_k. \quad (2.37)$$

Além disso, (2.32) continua válida neste caso e junto com (2.29) e (2.35) fornece

$$|m_k(x^k) - m_k(z^k)| \leq N h(x^k) \leq \frac{N c_h}{\gamma^2} \Delta_k^2 \leq \frac{1}{2} \tilde{c} \Delta_k. \quad (2.38)$$

Combinando (2.37) e (2.38), obtemos

$$pred_k = m_k(x^k) - m_k(x^{k+1}) = m_k(x^k) - m_k(z^k) + m_k(z^k) - m_k(x^{k+1}) \geq \frac{1}{2} \tilde{c} \Delta_k, \quad (2.39)$$

que junto com (2.29), (2.35) e a última desigualdade em (2.33) implica

$$pred_k \geq \frac{c_p c_h}{\gamma^2} \Delta_k^2 \geq c_p h(x^k) \geq c_p (h(x^k))^2.$$

Portanto, pelo mecanismo do Algoritmo 2.7, (2.35) e (2.39),

$$f_0(x^k) - f_0(x^{k+1}) = a red_k \geq \eta pred_k \geq \frac{1}{2} \eta \tilde{c} \Delta_k \geq \frac{\eta \tilde{c} \gamma}{2 \sqrt{c_h}} \sqrt{H_k}. \quad (2.40)$$

Vejam agora a segunda possibilidade, ou seja, que existe $\Delta \leq \gamma\Delta_\rho$ tal que o passo tentativo d_Δ foi calculado e $h(x^k + d_\Delta) < H_k$. Sejam $\hat{\Delta}$ o primeiro Δ com esta propriedade e $\bar{d} = d_{\hat{\Delta}}$ o passo tentativo obtido com o raio $\bar{\Delta} = \frac{\hat{\Delta}}{\gamma}$. Afirmamos que

$$h(x^k + \bar{d}) \geq H_k. \quad (2.41)$$

De fato, caso $\bar{\Delta} \leq \gamma\Delta_\rho$, a definição de $\hat{\Delta}$ garante a afirmação. Caso $\bar{\Delta} > \gamma\Delta_\rho$, temos $\bar{\Delta} \in [\gamma^2\Delta_\rho, \Delta_\rho]$ e assim podemos aplicar o Lema 2.12, obtendo

$$h(x^k + \bar{d}) = h(z^k + t_{\bar{\Delta}}) \geq H_k.$$

Assim, a desigualdade (2.41) é verdadeira. Procedendo agora de modo inteiramente análogo ao que fizemos acima e também no Lema 2.11, podemos mostrar que

$$c_h \hat{\Delta}^2 \geq \gamma^2 H_k \geq \gamma^2 h(x^k), \quad (2.42)$$

$$\text{pred}_{\hat{\Delta}} \stackrel{\text{def}}{=} m_k(x^k) - m_k(z^k + t_{\hat{\Delta}}) \geq \frac{1}{2} \tilde{c} \hat{\Delta}, \quad (2.43)$$

$$\text{ared}_{\hat{\Delta}} \stackrel{\text{def}}{=} f_0(x^k) - f_0(z^k + t_{\hat{\Delta}}) \geq \eta \text{pred}_{\hat{\Delta}} \geq \frac{1}{2} \eta \tilde{c} \hat{\Delta} \quad (2.44)$$

e

$$\text{ared}_{\hat{\Delta}} \geq \alpha h(x^k), \quad (2.45)$$

onde α é a constante dada no Algoritmo 2.1. A definição de $\hat{\Delta}$ e (2.45) garantem que $x^k + d_{\hat{\Delta}} (= z^k + t_{\hat{\Delta}})$ é aceito pelo filtro. Portanto, usando (2.44), concluímos que $x^k + d_{\hat{\Delta}} = x^{k+1}$. Além disso, (2.42) e (2.44) implicam

$$f_0(x^k) - f_0(z^k + t_{\hat{\Delta}}) \geq \frac{1}{2} \eta \tilde{c} \hat{\Delta} \geq \frac{\eta \tilde{c} \gamma}{2 \sqrt{c_h}} \sqrt{H_k},$$

isto é,

$$f_0(x^k) - f_0(x^{k+1}) \geq \frac{\eta \tilde{c} \gamma}{2 \sqrt{c_h}} \sqrt{H_k}. \quad (2.46)$$

Como (2.34), (2.40) e (2.46) esgotam todas as possibilidades, definindo

$$M = \min \left\{ \frac{1}{2} \eta \tilde{c} \gamma^2 \Delta_\rho, \frac{\eta \tilde{c} \gamma}{2 \sqrt{c_h}} \right\},$$

completamos a demonstração. □

Apresentamos abaixo um estudo comparativo entre as duas abordagens discutidas aqui.

2.3.3 Discussão comparativa

O objetivo desta seção é destacar as principais semelhanças e diferenças entre os algoritmos de filtro com restauração inexata [23], que indicaremos por GKV, e filtro com PQS, indicado por FPQS (Algoritmos 2.1 e 2.7). Conforme vimos anteriormente, ambos satisfazem uma propriedade de convergência global, garantindo que pelo menos um ponto da sequência gerada seja estacionário ([23, Teorema 2.7] e Teorema 2.5).

Obtenção do passo. Quanto a obtenção do passo, existem diferenças substanciais entre GKV e FPQS. No primeiro, o passo é obtido usando as idéias de restauração inexata de Martínez [32] e Martínez e Pilotta [33], onde cada iteração é decomposta em duas fases independentes. Existe uma certa liberdade para os algoritmos internos (viabilidade e otimalidade), desde que satisfaçam hipóteses razoáveis sobre a sua eficiência. Por outro lado, o Algoritmo FPQS calcula o passo usando PQS com região de confiança, também decompondo em passo de viabilidade e passo tangencial, mas de modo similar aos métodos de Byrd [5] e Omojokun [40]. Neles a região de confiança tem centro no ponto corrente x^k , enquanto que em GKV o centro é o ponto z^k , obtido na fase de viabilidade. Neste último, a vantagem está no fato de que a fase de viabilidade não é repetida quando se reduz a região de confiança (pelo passo ter sido recusado).

Na Figura 2.6 representamos uma iteração de cada algoritmo. O conjunto viável está representado pela linha curva cheia, enquanto que as curvas de nível da função objetivo são as linhas tracejadas. Indicamos o ponto corrente por x , o ponto obtido após a fase de viabilidade e o ponto tentativo para GKV por z e x_G^+ , os passos de viabilidade e tangencial e o ponto tentativo para FPQS por n , t e x_F^+ , respectivamente. Note nesta mesma figura os conjuntos linearizados para cada algoritmo. Um deles é definido por

$$\mathcal{L}_k^G = \{z^k + t \in \mathbb{R}^n \mid A_{\mathcal{E}}(z^k)t = 0, f_{\mathcal{I}}(z^k) + A_{\mathcal{I}}(z^k)t \leq f_{\mathcal{I}}^+(z^k)\},$$

com as funções calculadas em z^k , e o outro por

$$\mathcal{L}_k^F = \{x^k + d \in \mathbb{R}^n \mid f_{\mathcal{E}}(x^k) + A_{\mathcal{E}}(x^k)d = 0, f_{\mathcal{I}}(x^k) + A_{\mathcal{I}}(x^k)d \leq 0\},$$

com as funções calculadas em x^k .

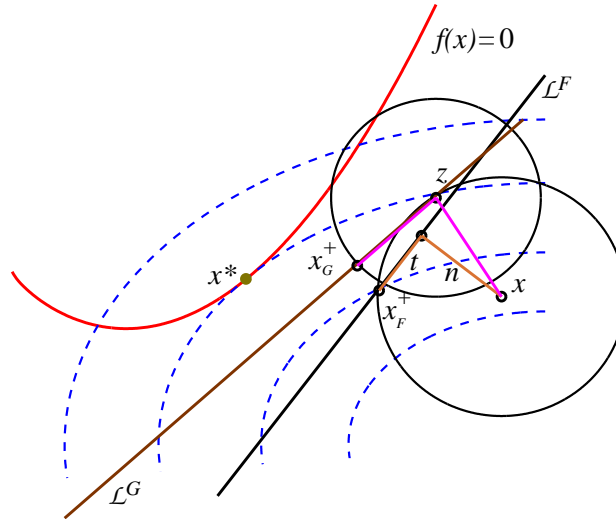


Figura 2.6: Comparação entre GKV e FPQS.

Aproveitando a notação da Figura 2.6, salientamos que o critério de região de confiança, baseado nos valores *ared* e *pred*, avalia o progresso entre z e x_G^+ no Algoritmo GKV e entre x e x_F^+ para FPQS.

Convergência global para *todo* ponto de acumulação. Uma observação importante a respeito da convergência do Algoritmo GKV é que se pode obter um resultado mais forte fazendo uma pequena modificação no critério de atualização do filtro no Algoritmo 2.1, conforme descrito abaixo.

Atualização do filtro: ($\varepsilon > 0$ é uma constante)

$$\text{SE } f_0(x^{k+1}) < f_0(x^k) - \min\{(h(x^k))^2, \varepsilon\},$$

$$F_{k+1} = F_k, \quad \mathcal{F}_{k+1} = \mathcal{F}_k \quad (\text{iteração } f_0)$$

SENÃO,

$$F_{k+1} = \bar{F}_k, \quad \mathcal{F}_{k+1} = \bar{\mathcal{F}}_k \quad (\text{iteração } h)$$

Com esta modificação mais pontos podem ser incluídos no filtro, o que torna o algoritmo mais restritivo. O seguinte resultado é provado em [23, Teorema 4.5].

Teorema 2.14 *Todo ponto de acumulação da sequência (x^k) é estacionário.*

Acreditamos que isto também seja válido para o Algoritmo FPQS, mas este assunto ainda é objeto de pesquisa no presente momento.

Um outro comentário que gostaríamos de fazer, apesar de menos importante, é que quando incluímos o par (\tilde{f}_0, \tilde{h}) no filtro, pode acontecer que algum ponto já

existente seja dominado por este. É conveniente, mas não necessário, eliminar os pontos dominados. Isto evita que sejam feitas comparações desnecessárias quando vamos testar se um ponto é proibido. Apresentamos nos Anexos uma subrotina para fazer este teste sem ter que percorrer todos os pontos do filtro, supondo que eles estejam ordenados pelas abscissas. Temos também uma subrotina para atualizar o filtro, que elimina os pontos dominados e faz a referida ordenação.

Neste capítulo foram abordados apenas os aspectos de convergência global dos métodos de filtro, que embora sejam menos restritivos que os algoritmos que usam função de mérito, podem sofrer do efeito Maratos. Isto significa que podem ser recusados passos essencialmente bons, mesmo arbitrariamente próximos de uma solução, comprometendo a eficiência de tais métodos. No próximo capítulo estudaremos a convergência local bem como uma técnica para evitar esse indesejável fenômeno.

Capítulo 3

Estudo de convergência local em programação não linear

Discutimos nos capítulos anteriores algoritmos com propriedades de convergência global para resolver problemas de PNL. A convergência global de um algoritmo é uma característica importante, pois garante que um ponto estacionário pode ser obtido, independentemente da escolha do ponto inicial. Entretanto, para fins práticos, o tempo para que o algoritmo encontre a solução não pode ser muito grande. Precisamos portanto estudar a velocidade de convergência da sequência gerada pelo algoritmo. Como os algoritmos discutidos aqui são baseados no método de Newton, é razoável esperar que as boas propriedades de convergência (superlinear ou quadrática) sejam herdadas. Para minimização irrestrita isto realmente acontece. Um algoritmo de região de confiança, que utiliza a própria função objetivo como função de mérito, aceita passos completos de Newton, assintoticamente. Este fato decorre de [12, Teorema 6.5.5]. No caso de minimização com restrições, porém, isto não ocorre. Algoritmos que utilizam certas funções de mérito, como a definida em (1.9), bem como os algoritmos de filtro, podem rejeitar passos completos de Newton, mesmo que o ponto corrente esteja arbitrariamente próximo da solução e que o passo seja muito bom. Este fenômeno, conhecido como *efeito Maratos* [31], pode inibir uma rápida convergência, esperada para os métodos aqui abordados.

Estudaremos neste capítulo uma técnica para evitar o efeito Maratos, chamada *correção de segunda ordem*, que adiciona ao passo de Newton uma correção com o objetivo de promover um decréscimo na medida de inviabilidade. Antes, porém, apresentaremos o clássico exemplo, devido a Powell [42], do efeito Maratos.

3.1 Efeito Maratos

O fenômeno conhecido como efeito Maratos ocorre quando o passo obtido na resolução do subproblema (PQ_k) , definido na Seção 1.1, é rejeitado mesmo que o ponto corrente x^k esteja arbitrariamente próximo de uma solução x^* . Uma situação em que isto ocorre é quando o passo provoca um aumento tanto na função objetivo quanto na medida de inviabilidade. Deste modo, se o critério de aceitação do passo utiliza, por exemplo, a função de mérito (1.9), temos

$$\phi_{\sigma_k}(x^k + d^k) > \phi_{\sigma_k}(x^k),$$

o que significa que o passo d^k é rejeitado. Isto implica na redução do tamanho do passo no caso de busca linear, ou na redução do raio da região de confiança. Como consequência, a convergência superlinear ou quadrática pode ficar comprometida.

O mesmo acontece com os algoritmos de filtro, já que

$$f_0(x^k + d^k) > f_0(x^k) \quad \text{e} \quad h(x^k + d^k) > h(x^k)$$

implicam que o ponto $x^k + d^k$ é proibido.

Vamos agora ilustrar este fato com o conhecido exemplo de Powell.

Exemplo 3.1

Considere o problema

$$\begin{aligned} \text{minimizar} \quad & f_0(x) = -x_1 + 2(x_1^2 + x_2^2 - 1) \\ \text{sujeito a} \quad & f(x) = x_1^2 + x_2^2 - 1 = 0. \end{aligned} \tag{3.1}$$

É fácil ver que a solução (única e global) é o ponto $x^* = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$, com multiplicador correspondente $\lambda^* = -\frac{3}{2}$. A hessiana do lagrangeano é $H(x, \lambda^*) = I$.

Suponha que o ponto corrente (viável) seja $x^k = \begin{pmatrix} \cos \theta \\ \cos \theta \end{pmatrix}$ e tome o multiplicador $\lambda^k = \lambda^* = -\frac{3}{2}$. Assim, o subproblema quadrático (PQ_k) , associado a (3.1), fica

$$\begin{aligned} \text{minimizar} \quad & (\cos \theta - 1)d_1 + (\sin \theta)d_2 + \frac{1}{2}(d_1^2 + d_2^2) \\ \text{sujeito a} \quad & (\cos \theta)d_1 + (\sin \theta)d_2 = 0, \end{aligned} \tag{3.2}$$

sendo que desconsideramos os termos constantes no modelo. Resolvendo (3.2), obtemos $d_1^k = \sin^2 \theta$ e $d_2^k = -\sin \theta \cos \theta$.

Uma manipulação algébrica nos fornece

$$\begin{aligned} f_0(x^k + d^k) &= -\cos \theta + \sin^2 \theta > -\cos \theta = f_0(x^k), \\ h(x^k + d^k) &= \sin^2 \theta > 0 = h(x^k), \end{aligned}$$

se θ não é múltiplo de π . Isto significa que o passo d^k não pode ser aceito, considerando os critérios discutidos acima.

Note que podemos tomar x^k tão próximo da solução quanto desejarmos, bastando para isso escolher $|\theta|$ pequeno o suficiente. Além disso vale a relação quadrática

$$\|x^k + d^k - x^*\| = 1 - \cos \theta = \frac{\|x^k - x^*\|^2}{2},$$

ou seja, o passo d^k fornece um ótimo progresso em direção à solução. □

Este exemplo é ilustrado na Figura 3.1, onde o conjunto viável está representado pela linha cheia, as curvas de nível da função objetivo pelas linhas tracejadas e as curvas de nível do lagrangeano pelas circunferências. O passo causou uma perda de viabilidade e também provocou um aumento na função objetivo.

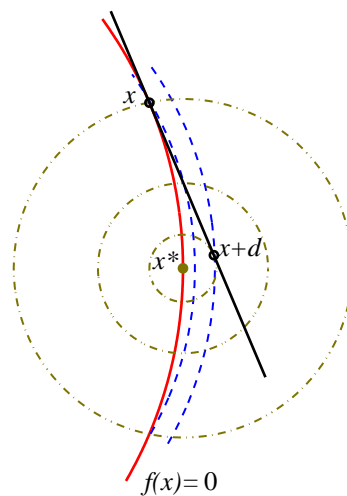


Figura 3.1: Efeito Maratos para o Exemplo 3.1.

3.2 Correção de segunda ordem

Uma das maneiras de contornar o efeito Maratos é tentar corrigir o passo de Newton, quando este não for aceito, por meio de um outro (pequeno) passo, chamado correção de segunda ordem. O objetivo deste segundo passo é reduzir a inviabilidade,

fazendo uma restauração em $x^k + d^k$. Esta restauração pode ser vista como um “passo de Newton” para $f(x) = 0$ no ponto $x^k + d^k$.

Para tornar precisas estas idéias, vamos retomar nosso problema de PNL

$$(P_E) \quad \begin{array}{ll} \text{minimizar} & f_0(x) \\ \text{sujeito a} & f(x) = 0, \end{array}$$

com $f_0 : \mathbb{R}^n \rightarrow \mathbb{R}$, $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ duas vezes continuamente diferenciáveis. Lembremos também do subproblema quadrático associado

$$(PQ_k) \quad \begin{array}{ll} \text{minimizar} & \ell(x^k, \lambda^k) + \nabla_x \ell(x^k, \lambda^k)^T d + \frac{1}{2} d^T H(x^k, \lambda^k) d \\ \text{sujeito a} & A(x^k) d + f(x^k) = 0. \end{array}$$

Conforme vimos no Capítulo 1, resolver (PQ_k) é equivalente a resolver o subproblema

$$\begin{array}{ll} \text{minimizar} & f_0(x^k) + \nabla f_0(x^k)^T d + \frac{1}{2} d^T H(x^k, \lambda^k) d \\ \text{sujeito a} & A(x^k) d + f(x^k) = 0, \end{array}$$

cujas condições de KKT são dadas por

$$\begin{bmatrix} H(x^k, \lambda^k) & A(x^k)^T \\ A(x^k) & 0 \end{bmatrix} \begin{bmatrix} d^k \\ \xi^k \end{bmatrix} = - \begin{bmatrix} \nabla f_0(x^k) \\ f(x^k) \end{bmatrix}, \quad (3.3)$$

sendo que o multiplicador ξ^k pode ser tomado como λ^{k+1} .

O efeito Maratos surge porque a linearização das restrições não leva em conta a curvatura, isto é, não possui o termo quadrático. É possível, portanto, que a aproximação

$$f(x^k + d^k) \approx f(x^k) + A(x^k) d^k$$

não seja confiável, pois o erro (proveniente da curvatura das restrições)

$$f(x^k + d^k) - f(x^k) - A(x^k) d^k = O(\|d^k\|^2) \quad (3.4)$$

pode ser muito grande.

Seria bastante razoável se o erro cometido quando aproximamos $f(x^k + d^k)$, fosse $o(\|d^k\|^2)$. Isto poderia ser obtido tomando-se os três primeiros termos da aproximação de Taylor de f . Entretanto, além do altíssimo custo computacional, o subproblema deixaria de ser um problema de programação quadrática.

Ao invés disso, vamos adicionar ao passo d^k uma correção d_{soc}^k de modo que

$$f(x^k + d^k + d_{\text{soc}}^k) = o(\|d^k\|^2). \quad (3.5)$$

Entretanto, vamos exigir que esta correção não seja muito grande quando comparada com o passo de Newton d^k , já que não queremos perder suas boas propriedades de convergência. A condição

$$d_{\text{soc}}^k = O(\|d^k\|^2) \quad (3.6)$$

atende tal requisito. De fato, supondo (3.6) e

$$\|x^k + d^k - x^*\| = O(\|x^k - x^*\|^2),$$

temos

$$\|d^k\| \leq \|x^k + d^k - x^*\| + \|x^k - x^*\| = O(\|x^k - x^*\|),$$

e portanto

$$\|x^k + d^k + d_{\text{soc}}^k - x^*\| \leq \|x^k + d^k - x^*\| + \|d_{\text{soc}}^k\| = O(\|x^k - x^*\|^2). \quad (3.7)$$

As relações (3.5) e (3.6) justificam o nome correção de segunda ordem. O índice “soc” vem do inglês: “second-order correction”.

Veremos agora algumas formas de se obter um passo de correção de segunda ordem. Para mais detalhes, citamos [12, Seção 15.3.2.3].

Uma abordagem consiste em fazer uma restauração a partir de $x^k + d^k$, resolvendo o sistema

$$f(x^k + d^k) + A(x^k + d^k)d_{\text{soc}}^k = 0, \quad (3.8)$$

e tomando a solução de mínima norma. Considerando a norma euclidiana, pode-se mostrar [12, Seção 4.4.2] que se $A(x^k + d^k)$ tem posto linha completo, então esta solução é dada por

$$d_{\text{soc}}^k = -A(x^k + d^k)^T (A(x^k + d^k)A(x^k + d^k)^T)^{-1} f(x^k + d^k), \quad (3.9)$$

ou, equivalentemente

$$\begin{bmatrix} I & A(x^k + d^k)^T \\ A(x^k + d^k) & 0 \end{bmatrix} \begin{bmatrix} d_{\text{soc}}^k \\ \lambda_{\text{soc}}^k \end{bmatrix} = - \begin{bmatrix} 0 \\ f(x^k + d^k) \end{bmatrix}. \quad (3.10)$$

Note que (3.3) e (3.4) implicam

$$f(x^k + d^k) = O(\|d^k\|^2), \quad (3.11)$$

e portanto, usando (3.9), concluímos que (3.6) é satisfeita. Além disso o Teorema de Taylor juntamente com a relação (3.8) nos fornecem

$$f(x^k + d^k + d_{\text{soc}}^k) = f(x^k + d^k) + A(x^k + d^k)d_{\text{soc}}^k + O(\|d_{\text{soc}}^k\|^2) = O(\|d_{\text{soc}}^k\|^2). \quad (3.12)$$

Utilizando então (3.6), obtemos

$$f(x^k + d^k + d_{\text{soc}}^k) = O(\|d^k\|^4),$$

o que implica (3.5). Assim a solução de mínima norma de (3.8) é de fato uma correção de segunda ordem.

Uma pequena variante desta abordagem reside na escolha de uma outra norma ao invés da euclidiana. Por exemplo, tomando uma matriz H definida positiva e a norma

$$x \mapsto \|x\|_H = \sqrt{x^T H x},$$

a relação (3.10) fica

$$\begin{bmatrix} H & A(x^k + d^k)^T \\ A(x^k + d^k) & 0 \end{bmatrix} \begin{bmatrix} d_{\text{soc}}^k \\ \lambda_{\text{soc}}^k \end{bmatrix} = - \begin{bmatrix} 0 \\ f(x^k + d^k) \end{bmatrix}, \quad (3.13)$$

e novamente podemos concluir que esta solução é uma correção de segunda ordem. Observe que (3.10) é um caso particular de (3.13), com $H = I$.

Em termos de custo computacional, o cálculo de d_{soc}^k através de (3.13) pode ser muito caro, pois exige a avaliação da jacobiana das restrições em $x^k + d^k$. Isto pode comprometer a eficiência do algoritmo. Podemos, no entanto, conseguir um passo de correção satisfazendo (3.5) e (3.6) aproveitando a jacobiana já calculada em x^k .

Para tanto, utilizamos a relação $A(x^k + d^k) = A(x^k) + O(\|d^k\|)$ em (3.12), obtendo

$$f(x^k + d^k + d_{\text{soc}}^k) = f(x^k + d^k) + A(x^k)d_{\text{soc}}^k + O(\|d^k\|)d_{\text{soc}}^k + O(\|d_{\text{soc}}^k\|^2), \quad (3.14)$$

e escolhemos d_{soc}^k como o vetor de mínima H -norma satisfazendo

$$f(x^k + d^k) + A(x^k)d_{\text{soc}}^k = 0, \quad (3.15)$$

ou seja, tal que

$$\begin{bmatrix} H & A(x^k)^T \\ A(x^k) & 0 \end{bmatrix} \begin{bmatrix} d_{\text{soc}}^k \\ \lambda_{\text{soc}}^k \end{bmatrix} = - \begin{bmatrix} 0 \\ f(x^k + d^k) \end{bmatrix}. \quad (3.16)$$

Para escolhas convenientes da matriz H , como por exemplo $H = I$ ou $H = H(x^k, \lambda^k)$,¹ temos que a matriz aumentada em (3.16) tem inversa uniformemente limitada em uma vizinhança de uma solução de (P_E) , onde valem as condições suficientes de segunda ordem. Deste modo, concluímos de (3.11) e (3.16) que

$$d_{\text{soc}}^k = O(\|d^k\|^2). \quad (3.17)$$

Substituindo então (3.15) e (3.17) em (3.14), obtemos

$$f(x^k + d^k + d_{\text{soc}}^k) = O(\|d^k\|^3), \quad (3.18)$$

o que juntamente com (3.17) caracteriza d_{soc}^k como uma correção de segunda ordem.

A Figura 3.2 ilustra o passo de correção de segunda ordem, definido por (3.16) e $H = I$, para o Exemplo 3.1.

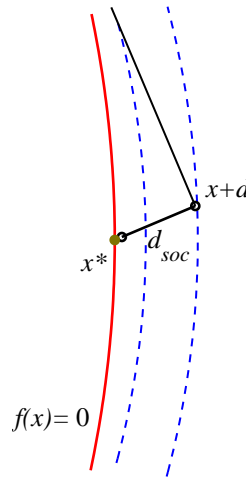


Figura 3.2: Correção de segunda ordem para o Exemplo 3.1.

Pode-se provar que o uso de correção de segunda ordem é efetivo para PQS com busca linear [4, Proposição 15.7], para PQS com região de confiança e função de mérito [12, Teorema 15.3.7], bem como em alguns algoritmos de filtro com busca linear [8, 49].

¹A matriz $H = H(x^k, \lambda^k)$ pode não definir uma norma, mas se for definida positiva no núcleo de $A(x^k)$, o sistema (3.16) está bem definido.

Ulbrich [47] também prova convergência superlinear para métodos de filtro mas sem o uso de correção de segunda ordem. A estratégia está na definição do filtro, onde as entradas são o lagrangeano e uma medida de inviabilidade que envolve as variáveis primais e duais.

3.3 Perspectivas

No presente momento, Gonzaga e Karas desenvolvem pesquisas para estabelecer resultados de convergência local para o algoritmo de filtro proposto em [23], baseado em restauração inexata. Um dos resultados garante que, sob certas hipóteses sobre a fase de viabilidade, o algoritmo “puro”² de restauração inexata herda as propriedades de convergência local Q-quadrática do método PQS.

O outro resultado refere-se à globalização usando filtro, incorporando passos de correção de segunda ordem para evitar o efeito Maratos e permanecer com as boas propriedades de velocidade de convergência.

Usando outra técnica, Birgin e Martínez [3] provam convergência R-linear para o algoritmo de restauração inexata. A convergência R-quadrática também é obtida, com hipóteses mais fortes.

²Isto é, sem um critério de aceitação de passo.

Capítulo 4

Resultados numéricos

Apresentamos neste capítulo resultados numéricos de um trabalho realizado em co-autoria com a professora Elizabeth Karas do Departamento de Matemática da Universidade Federal do Paraná e com o professor Elvio Pilotta do Departamento de Matemática da Universidade Nacional de Córdoba, Argentina.

Este estudo computacional tem o objetivo de avaliar a performance do algoritmo de restauração inexata com filtro [23], bem como compará-lo com o algoritmo de restauração inexata com função de mérito [32, 33]. Também reportamos resultados com o método de lagrangeano aumentado implementado no conhecido pacote LANCELOT [11], descrito em [10]. Conforme salientado em [32, 33], os algoritmos de restauração inexata devem ser aplicados levando-se em conta a especificidade do problema a ser resolvido, pois, em virtude da liberdade para as fases de viabilidade e de otimalidade, muitos problemas permitem que seja aplicado um algoritmo interno muito mais simples e barato. Este é o caso dos “*Hard-Spheres Problems*” (HSP), que descrevemos aqui e usamos para os testes.

Os “Hard-Spheres Problems” constituem uma classe de problemas muito interessantes em matemática. Estes problemas desafiantes, com enunciado extremamente simples, já despertaram interesse em muitos matemáticos respeitáveis (p. ex. Newton, Hilbert, Gregory), mas muitos deles ainda permanecem em aberto. Além disso, eles tem algumas relações com uma grande quantidade de problemas práticos em química, biologia e física, veja por exemplo [13, 43]. Cada HSP pode ser parametrizado por dois inteiros positivos, p e n . O problema consiste em maximizar a mínima distância (dois a dois) entre p pontos em uma esfera do \mathbb{R}^n e pode ser formulado como um problema de PNL não convexo, difícil e potencialmente com um grande número de pontos (não ótimos) satisfazendo as condições necessárias de otimalidade. Temos assim um conjunto de problemas indexados por p e n , que pode ser utilizado para testar algoritmos

de otimização. Aliás, isto já foi feito no caso do ALBOX [30], um algoritmo de Lagrangeano aumentado que usa aproximações de Gauss-Newton das hessianas em cada iteração interna.

4.1 O problema “Hard-Spheres”

Considerando, sem perda de generalidade, que a esfera é unitária e centrada na origem, podemos formular diretamente este problema como

$$\begin{aligned} \max \min_{i \neq j} \|y^i - y^j\| \\ \text{sujeito a } \|y^k\| = 1, \quad k = 1, \dots, p, \end{aligned} \quad (4.1)$$

onde y^k , $k = 1, \dots, p$ são vetores em \mathbb{R}^n e $\|\cdot\|$ é a norma Euclidiana.

A Figura 4.1 ilustra o problema “Hard-Spheres” no caso $p = 6$ e $n = 2$. A do lado esquerdo representa uma solução viável (não ótima) e a do lado direito representa uma solução ótima para o problema (4.1).

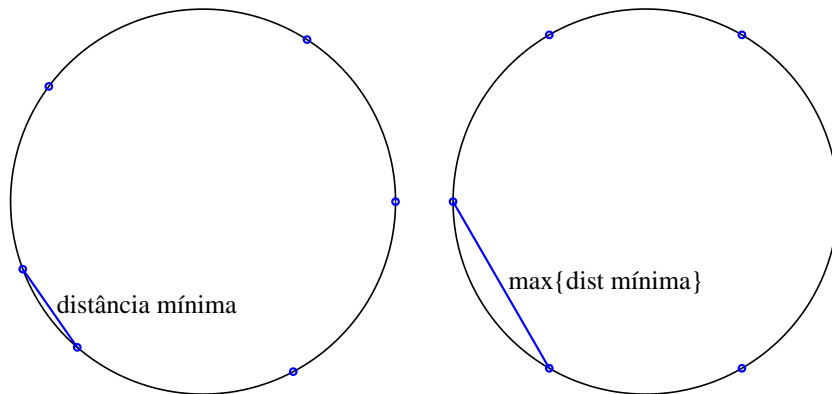


Figura 4.1: O problema “Hard-Spheres”, com $p = 6$ e $n = 2$.

Usando as propriedades do produto interno usual $\langle \cdot, \cdot \rangle$, é fácil ver que o problema (4.1) é equivalente a

$$\begin{aligned} \min \max_{i \neq j} \langle y^i, y^j \rangle \\ \text{sujeito a } \|y^k\| = 1, \quad k = 1, \dots, p. \end{aligned} \quad (4.2)$$

Aplicando agora o clássico truque para transformar um problema minimax em

outro de minimização, nós reduzimos (4.2) ao problema de PNL

$$\begin{aligned} & \text{minimizar } \xi \\ & \text{sujeito a } \langle y^k, y^k \rangle - 1 = 0, \quad k = 1, \dots, p, \\ & \quad \langle y^i, y^j \rangle - \xi \leq 0, \quad \forall i \neq j. \end{aligned} \quad (4.3)$$

É neste problema que concentraremos nossa atenção. Ele tem $(np+1)$ variáveis que serão representadas pelo vetor $x = (y^1, \dots, y^p, \xi)$ em \mathbb{R}^{np+1} e está no formato do problema (P) , definido no Capítulo 2. A função objetivo é $f_0(x) = \xi$ e a função f , que define as restrições, é dada por

$$\begin{aligned} f_k(x) &= \langle y^k, y^k \rangle - 1, \quad k = 1, \dots, p, \\ f_{ij}(x) &= \langle y^i, y^j \rangle - \xi, \quad i \neq j. \end{aligned} \quad (4.4)$$

Antes de passarmos à discussão dos métodos para resolver o problema “Hard-Spheres”, vamos salientar uma relação com um outro famoso problema na área que é o “Kissing Number Problem”. Neste problema o objetivo é determinar o número máximo \mathcal{K}_n de esferas com raio unitário que tangenciam, sem sobreposição, a esfera $S = \{x \in \mathbb{R}^n \mid \|x\| = 1\}$. Dados n e p , não é difícil verificar que a distância obtida na solução do problema (4.1) é maior ou igual a um se, e somente se, $\mathcal{K}_n \geq p$.

Na Figura 4.2 ilustramos o problema “Kissing Number” para $n = 2$. No lado esquerdo representamos a solução $\mathcal{K}_2 = 6$, enquanto que no lado direito temos uma solução ótima do problema “Hard-Spheres” com $p = 7 > \mathcal{K}_2$, caso em que não conseguimos fazer as sete esferas tangenciarem S , sem sobreposição.

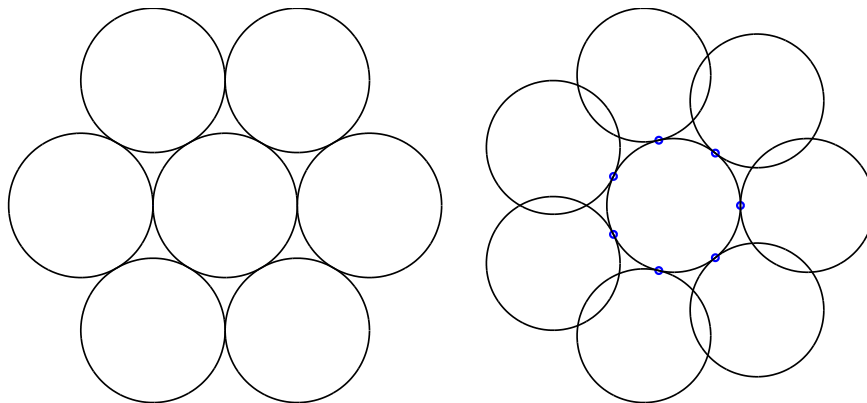


Figura 4.2: O problema “Kissing Number”, com $n = 2$.

Para alguns valores de n , o número \mathcal{K}_n é conhecido. No entanto há outros valores de n para os quais se conhece, ou se conjectura, apenas um intervalo que

contém \mathcal{K}_n . Esses valores ou limites são apresentados na Tabela 4.1, obtida a partir de [13] e [37].

n	\mathcal{K}_n
1	2
2	6
3	12
4	24
5	40–46
6	72–82
7	126–140
8	240
9	306–380
10	500–595
11	582–915
12	840–1416

Tabela 4.1: Valores/limites conhecidos de \mathcal{K}_n .

4.2 Métodos de restauração inexata

Nos métodos de restauração inexata, propostos por Martínez [32] e Martínez e Pilotta [33], cada iteração é decomposta em duas fases. A primeira delas visa reduzir uma medida de inviabilidade, enquanto que a segunda, chamada fase de otimalidade, procura reduzir o valor da função objetivo em uma aproximação tangencial do conjunto viável. Na fase de otimalidade é aplicado um método de região de confiança. Entretanto o centro da região de confiança é o ponto obtido na fase de viabilidade e não o ponto corrente. A diferença em relação aos métodos de PQS com região de confiança está no fato de que em restauração inexata somente o passo de otimalidade é recalculado quando se reduz a região de confiança. Deste modo, as fases de viabilidade e otimalidade se tornam mais independentes.

Nos métodos propostos por Martínez e Pilotta uma função de mérito é utilizada para avaliar o passo. Gonzaga, Karas e Vanti [23], inspirados nas idéias de Martínez e Pilotta, propuseram um algoritmo de restauração inexata que usa filtro como critério de aceitação do passo. Eles também propõe, neste mesmo trabalho, uma pequena modificação no critério de atualização do filtro¹ e melhoram o resultado

¹Conforme discutido na Seção 2.3.3.

de convergência, mostrando que todo ponto de acumulação da sequência gerada pelo algoritmo é estacionário.

Descrevemos em seguida algumas condições razoáveis para cada fase destes métodos.

4.2.1 Fase de viabilidade

Dado o ponto corrente $x^k \in \mathbb{R}^n$, o algoritmo calcula um ponto intermediário “menos inviável” $z^k \in \mathbb{R}^n$, que deve satisfazer as seguintes condições

$$h(z^k) \leq (1 - \alpha)h(x^k) \quad (4.5)$$

e

$$\|z^k - x^k\| \leq \beta h(x^k) \quad (4.6)$$

onde h é definida por (2.1) e $\alpha \in (0, 1)$ e $\beta > 0$ são parâmetros dados, independentes de k . A condição (4.5) exige uma redução na medida de inviabilidade. Já a condição (4.6) faz um controle no tamanho do passo de viabilidade. Em particular, impõe que z^k deve ser igual a x^k se este for viável.

4.2.2 Fase de otimalidade

Vamos associar a cada ponto $z \in \mathbb{R}^n$ o conjunto

$$L(z) = \{x \in \mathbb{R}^n \mid A_{\mathcal{E}}(z) (x - z) = 0, f_{\mathcal{I}}(z) + A_{\mathcal{I}}(z) (x - z) \leq f_{\mathcal{I}}^+(z)\} \quad (4.7)$$

que é a linearização do conjunto $\{x \in \mathbb{R}^n \mid f_{\mathcal{E}}(x) = f_{\mathcal{E}}(z), f_{\mathcal{I}}(x) \leq f_{\mathcal{I}}^+(z)\}$. Em um ponto viável z , $L(z)$ é a linearização do conjunto viável do problema (P).

O algoritmo de otimalidade deve encontrar x^{k+1} no linearizado $L(z^k)$, tal que $f_0(x^{k+1}) \leq f_0(z^k)$ e o critério de aceitação de passo seja satisfeito. Para calcular o ponto tentativo de modo eficiente, podemos aplicar um algoritmo razoável ao problema

$$\begin{aligned} &\text{minimizar} && f_0(x) \\ &\text{sujeito a} && x \in L(z^k) \\ &&& \|x - z^k\| \leq \Delta, \end{aligned}$$

onde Δ é o raio da região de confiança.

4.2.3 Critério de aceitação do passo

A diferença essencial entre os métodos propostos em [23] e [32, 33] reside no critério de aceitação do passo. No primeiro se utiliza filtro, enquanto que no segundo o progresso é medido por uma função de mérito. Deste modo, uma comparação entre as performances destes dois métodos parece bastante oportuna.

4.3 Implementação

Usamos a solução conhecida do problema “Kissing Number” tridimensional para ajustar os parâmetros dos algoritmos a serem testados. Os testes foram feitos com valores de p e n que podem ampliar ou reforçar as conjecturas existentes sobre os valores de \mathcal{K}_n , apresentados na Tabela 4.1.

Fase de viabilidade. Pela especificidade do problema (4.3), podemos aplicar uma fase de viabilidade simples, barata e que reduz a medida de inviabilidade para zero, isto é, dado $x = (y^1, \dots, y^p, \xi)$, obtemos facilmente um ponto $z = (w^1, \dots, w^p, \zeta)$ que satisfaça todas as restrições. De fato, basta definir

$$w^i = \frac{y^i}{\|y^i\|}, \quad i = 1, \dots, p \quad \text{e} \quad \zeta = \max\{\langle w^i, w^j \rangle, \quad i \neq j\}.$$

Deste modo vemos que o uso dos “Hard-Spheres Problems” é conveniente para testar os algoritmos, pois a fase de viabilidade, que normalmente é um ponto delicado na implementação, aqui é trivial.

Fase de otimalidade. Nesta fase usamos o conhecido pacote MINOS [36] para minimização com restrições lineares. O problema a ser resolvido pelo MINOS consiste em minimizar a variável ξ na interseção do politopo definido pela linearização das restrições de (4.3) com a região de confiança (caixa) em torno de z .

4.3.1 Experimentos computacionais

Com o propósito de comparar os algoritmos de restauração inexata com filtro e com função de mérito, resolvemos 30 problemas da família HSP usando o algoritmo original proposto por Gonzaga, Karas e Vanti [23] (Filtro A), a sua versão modificada (Filtro B) e o algoritmo proposto por Martínez e Pilotta [33] (IR). Também resolvemos os problemas pelo algoritmo de Lagrangeano aumentado implementado no pacote LANCELOT [11], descrito em [10].

Rodamos os quatro códigos usando 50 pontos iniciais para cada um dos 30 problemas indexados por p e n . Os resultados estão apresentados nas tabelas que seguem. Além disso, para facilitar a análise dos resultados, dispomos as informações em gráficos de desempenho, de acordo com os introduzidos por Dolan e Moré [14]. Consideramos tempo de CPU, número de iterações internas e número de iterações externas.

As Tabelas 4.2-4.4 listam os 30 problemas, com o número de variáveis, de restrições, uma informação estatística contendo a distância mínima entre dois pontos (mínimo, máximo e média) e o tempo de CPU (mínimo, máximo e médio), usando Filtro A, Filtro B, IR e LANCELOT, representados nas quatro linhas de cada conjunto, respectivamente.

As Figuras 4.3 e 4.4 mostram os desempenhos para o tempo de CPU. Na Figura 4.3 temos os desempenhos dos quatro algoritmos. Em particular, IR resolve 90% dos problemas usando 2,34 vezes o tempo de CPU gasto pelo melhor algoritmo, que neste caso é o Filtro A. LANCELOT resolve todos os problemas usando em torno de dez vezes o tempo de CPU gasto pelo melhor algoritmo. Por outro lado, a Figura 4.4 apresenta apenas os desempenhos das duas variantes do algoritmo de filtro.

As Tabelas 4.5-4.7 listam os 30 problemas, com o número de variáveis, de restrições e uma informação estatística contendo o número de iterações externas e internas (mínimo, máximo e média) usando Filtro A, Filtro B, IR, representados nas três linhas de cada conjunto, respectivamente.

As Figuras 4.5 e 4.6 mostram os desempenhos para o número de iterações externas e internas, respectivamente. Aqui também temos um bom desempenho do filtro. Em particular, IR resolve 80% (respectivamente, 70%) dos problemas usando 2,82 vezes (respectivamente, 2,52 vezes) o número de iterações externas (respectivamente, internas) gasto pelo melhor algoritmo.

Os resultados obtidos aqui atestam uma boa performance da abordagem que usa filtro, mas não evidenciam uma diferença significativa, pelo menos neste conjunto de problemas, entre o algoritmo original de filtro (Filtro A) e a sua versão modificada (Filtro B).

Problema			Distância mínima entre 2 pontos			Tempo de CPU em seg.		
p	var.	restr.	min.	max.	média	min.	max.	média
8	25	36	1.2155625	1.2155625	1.2155625	0.14	0.21	0.17
			1.2155625	1.2155625	1.2155625	0.14	0.21	0.17
			1.2155625	1.2155625	1.2155625	0.13	0.40	0.32
			1.2155657	1.2155657	1.2155657	0.31	0.74	0.48
9	28	45	1.1547005	1.1547005	1.1547005	0.17	0.29	0.21
			1.1547005	1.1547005	1.1547005	0.17	0.29	0.21
			1.1547005	1.1547005	1.1547005	0.15	0.50	0.41
			1.1547034	1.1547034	1.1547034	0.40	1.01	0.66
10	31	55	1.0514622	1.0914263	1.0886192	0.19	0.33	0.24
			1.0514622	1.0914263	1.0886192	0.19	0.33	0.24
			1.0514622	1.0914263	1.0897560	0.24	0.67	0.50
			1.0514656	1.0914302	1.0901244	0.43	1.70	0.89
11	34	66	1.0514622	1.0514622	1.0514622	0.25	0.39	0.33
			1.0514622	1.0514622	1.0514622	0.25	0.42	0.33
			1.0514622	1.0514622	1.0514622	0.65	0.87	0.74
			1.0514656	1.0514656	1.0514656	0.58	2.70	1.09
12	37	78	0.9472196	1.0514622	1.0493774	0.29	0.55	0.43
			0.9472196	1.0514622	1.0493774	0.29	0.55	0.41
			1.0514622	1.0514622	1.0514622	0.80	1.08	0.91
			0.9463826	1.0514766	1.0388557	0.60	1.70	1.12
13	40	91	0.9281797	0.9535789	0.9485253	0.30	0.57	0.41
			0.9281797	0.9535789	0.9485253	0.30	0.50	0.41
			0.9463817	0.9564136	0.9510276	0.80	1.14	0.96
			0.9427937	0.9564099	0.9502761	1.38	5.43	2.21
14	43	105	0.9087888	0.9338626	0.9279113	0.39	0.71	0.53
			0.9087888	0.9338626	0.9279113	0.39	0.71	0.52
			0.9161167	0.9338626	0.9323456	1.08	1.37	1.19
			0.9025187	0.9338629	0.9299379	1.25	5.32	2.71
15	46	120	0.8745439	0.9026562	0.9006951	0.50	0.83	0.64
			0.8745439	0.9026562	0.9006951	0.48	0.83	0.64
			0.8850043	0.9026562	0.9021272	0.69	1.73	1.39
			0.8978196	0.9026516	0.9019241	1.92	5.67	3.75
16	49	136	0.8650118	0.8805741	0.8746946	0.60	1.03	0.77
			0.8650118	0.8805741	0.8746946	0.60	1.03	0.76
			0.8732431	0.8805741	0.8777909	0.55	1.84	1.55
			0.8628557	0.8805793	0.8772524	2.13	7.00	4.23
17	52	153	0.8426190	0.8624449	0.8609066	0.63	1.13	0.91
			0.8426190	0.8624449	0.8609066	0.63	1.10	0.89
			0.8437469	0.8624449	0.8617040	0.64	2.31	1.80
			0.8399286	0.8624500	0.8604780	2.51	10.74	5.25

Tabela 4.2: Distâncias mínimas e tempo de CPU para $n = 3$ usando Filtro A, Filtro B, IR e LANCELOT.

Problema			Distância mínima entre 2 pontos			Tempo de CPU em seg.		
p	var.	restr.	min.	max.	média	min.	max.	média
20	81	210	1.0351487	1.0637118	1.0614469	2.26	3.76	2.99
			1.0351487	1.0637111	1.0609267	2.29	4.99	3.00
			1.0351487	1.0637118	1.0625693	1.99	8.88	6.24
			1.0351522	1.0637105	1.0609359	5.79	26.87	10.79
21	85	231	0.9936595	1.0282209	1.0224987	2.43	4.38	3.14
			0.9936595	1.0282209	1.0227355	2.32	4.35	3.20
			1.0000000	1.0282209	1.0256053	2.61	7.60	5.65
			0.9933378	1.0282218	1.0242247	9.78	44.45	21.34
22	89	253	0.9796009	1.0019895	0.9956077	2.36	5.98	3.67
			0.9833497	1.0019895	0.9959161	2.39	5.50	3.74
			0.9868542	1.0019895	0.9990601	3.18	10.42	6.44
			0.9825273	1.00119880	0.9956835	11.02	52.30	24.42
23	93	276	0.9713685	1.0000000	0.9837457	2.56	8.62	4.53
			0.9710819	1.0000000	0.9831386	2.59	7.32	4.45
			0.9730502	1.0000000	0.9862260	3.42	10.84	7.18
			0.9674813	0.9918568	0.9840495	12.44	65.33	31.48
24	97	300	0.9579707	1.0000000	0.9739830	3.30	8.71	5.12
			0.9579707	1.0000000	0.9737856	3.32	9.28	5.20
			0.9630490	1.0000000	0.9762215	4.38	14.77	8.50
			0.9586866	0.9828733	0.9736303	14.75	83.83	38.52
25	101	325	0.9494635	0.9617108	0.9575574	3.72	9.30	5.66
			0.9480549	0.9617108	0.9576500	3.78	8.51	5.85
			0.9548308	0.9619044	0.9589259	3.79	17.82	9.16
			0.9473436	0.9616652	0.9574605	18.80	96.79	43.32
26	105	351	0.9364391	0.9583427	0.9485589	4.32	11.49	6.62
			0.9364391	0.9583427	0.9491374	4.15	10.68	6.59
			0.9363436	0.9583427	0.9506996	4.62	19.41	9.99
			0.9383603	0.9583423	0.9506288	22.94	135.84	55.30
27	109	378	0.9277902	0.9406473	0.9349250	4.43	11.86	7.74
			0.9277902	0.9389306	0.9348058	5.11	11.64	7.74
			0.9293616	0.9406472	0.9364135	4.96	23.23	11.48
			0.9274373	0.9387758	0.9352882	31.00	146.21	65.95
28	113	406	0.9123780	0.9301962	0.9236410	5.54	11.43	8.50
			0.9123788	0.9301962	0.9238224	5.72	15.95	8.79
			0.9148390	0.9300958	0.9247347	5.16	22.80	12.16
			0.9168860	0.9302258	0.9247872	36.86	174.75	83.30
29	117	435	0.9021564	0.9207228	0.9148940	5.93	15.10	9.90
			0.9021564	0.9209968	0.9145187	4.63	15.45	10.24
			0.9085787	0.9210383	0.9160910	5.46	27.38	13.56
			0.9068517	0.9212383	0.9148001	31.55	160.80	80.21

Tabela 4.3: Distâncias mínimas e tempo de CPU para $n = 4$ usando Filtro A, Filtro B, IR e LANCELOT.

Problema			Distância mínima entre 2 pontos			Tempo de CPU em seg.		
p	var.	restr.	min.	max.	média	min.	max.	média
35	176	630	1.0047565	1.0190406	1.0122777	25.75	100.08	43.06
			1.0044725	1.0190031	1.0121760	22.41	78.05	44.76
			1.0118353	1.0202388	1.0153848	67.54	201.59	102.26
			1.0027761	1.0190191	1.0122799	144.26	835.69	379.40
36	181	666	0.9959127	1.0126686	1.0053548	30.89	119.33	52.43
			0.9959127	1.0128296	1.0052988	27.84	113.92	53.02
			1.0012557	1.0142040	1.0084049	77.53	163.33	110.00
			0.9977374	1.0127487	1.0045874	127.28	670.49	343.48
37	186	703	0.9922334	1.0033574	0.9984867	31.05	81.82	53.86
			0.9922334	1.0031985	0.9982152	32.50	104.10	53.63
			0.9949556	1.0054290	1.0010692	86.09	218.34	125.36
			0.9929753	1.0041514	0.9986303	136.80	968.82	449.49
38	191	741	0.9826614	1.0037492	0.9918239	32.75	167.94	62.87
			0.9857376	0.9986589	0.9916255	38.52	136.68	63.03
			0.9912413	1.0016036	0.9963429	99.67	395.94	149.08
			0.9848350	1.0002300	0.9925158	183.91	1024.40	453.80
39	196	780	0.9770637	0.9928665	0.9865952	44.90	106.35	67.10
			0.9776045	0.9928768	0.9868756	47.80	124.80	87.83
			0.9847763	0.9937400	0.9899947	104.57	280.71	156.59
			0.9979153	0.9919577	0.9864628	193.66	925.18	508.23
40	201	820	0.9736250	0.9887873	0.9822140	49.70	293.44	88.32
			0.9703412	0.9887802	0.9815535	39.99	169.77	81.34
			0.9778864	0.9923987	0.9849359	118.81	394.10	184.07
			0.9714525	0.9881295	0.9811602	167.10	1319.65	588.87
41	206	861	0.9667732	0.9838715	0.9761358	52.35	441.97	103.04
			0.9689059	0.9813675	0.9755642	49.75	193.66	89.28
			0.9734068	0.9824129	0.9787404	130.51	403.71	197.84
			0.9665506	0.9828632	0.9754543	236.51	1649.94	777.65
42	211	946	0.9595536	0.9781159	0.9693056	54.96	235.39	104.80
			0.9595536	0.9781159	0.9694242	55.14	264.36	106.54
			0.9660170	0.9796514	0.9720144	139.98	357.97	207.57
			0.9638880	0.9746999	0.9697488	297.13	2527.59	775.10
43	216	946	0.9556184	0.9666692	0.9621177	59.56	270.20	107.77
			0.9553271	0.9675972	0.9623982	72.24	266.27	111.07
			0.9616707	0.9728198	0.9660902	157.15	406.32	237.52
			0.9549450	0.9715040	0.9636488	331.96	1498.10	817.49
44	221	990	0.9518362	0.9611590	0.9573401	64.47	180.64	118.75
			0.9514074	0.9609389	0.9570276	67.22	939.06	137.38
			0.9556165	0.9691699	0.9606071	176.19	479.46	247.72
			0.9513464	0.9721522	0.9580631	351.86	2513.80	1085.30

Tabela 4.4: Distâncias mínimas e tempo de CPU para $n = 5$ usando Filtro A, Filtro B, IR e LANCELOT.

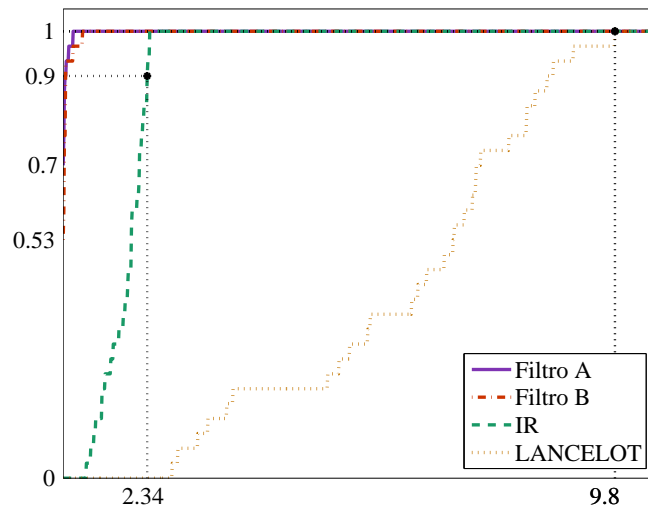


Figura 4.3: Desempenho dos quatro algoritmos - tempo de CPU.

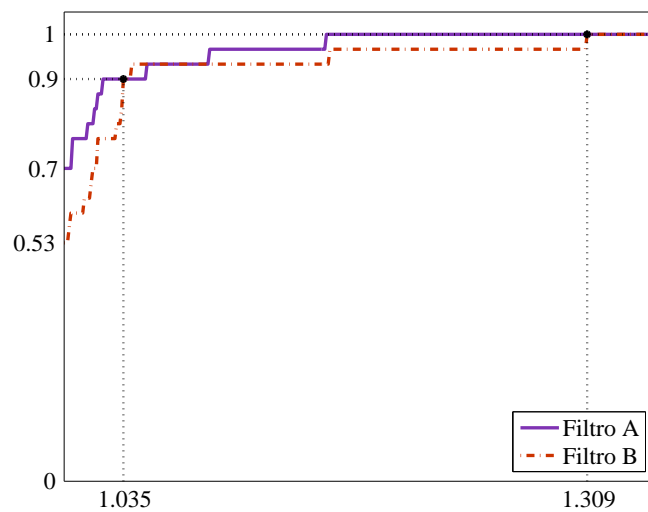


Figura 4.4: Desempenho dos algoritmos de filtro - tempo de CPU.

Problema			Iterações externas			Iterações internas		
p	var.	restr.	min.	max.	média	min.	max.	média
8	25	36	12	20	14	27	40	31
			11	19	14	26	39	31
			10	39	31	22	71	54
9	28	45	11	19	14	24	41	31
			11	19	14	24	41	31
			10	40	32	22	71	56
10	31	55	10	19	13	23	39	29
			10	19	13	23	39	29
			14	39	33	28	69	56
11	34	66	11	17	13	25	35	29
			11	17	13	25	36	29
			32	39	34	53	69	59
12	37	78	11	19	14	24	39	30
			11	19	13	24	39	29
			32	40	35	52	68	58
13	40	91	10	19	13	23	40	29
			10	17	13	23	37	29
			32	41	36	53	71	61
14	43	105	9	18	14	22	37	30
			9	16	13	22	35	30
			35	42	37	57	71	62
15	46	120	11	19	14	25	38	31
			11	17	14	25	37	31
			13	43	37	29	77	62
16	49	136	11	21	14	25	44	31
			11	21	14	25	44	31
			11	42	35	23	71	60
17	52	153	11	22	14	25	44	31
			11	17	14	25	37	30
			10	46	36	21	77	61

Tabela 4.5: Iterações externas e internas para $n = 3$, usando Filtro A, Filtro B e IR.

Problema			Iterações externas			Iterações internas		
p	var.	restr.	min.	max.	média	min.	max.	média
20	81	210	13	24	17	28	46	34
			13	23	16	29	45	34
			10	60	42	21	99	70
21	85	231	12	27	18	26	49	37
			12	26	18	26	47	37
			15	58	41	30	92	68
22	89	253	13	30	18	28	54	37
			13	30	18	28	53	37
			17	63	41	31	111	69
23	93	276	13	26	17	28	50	36
			12	24	16	28	43	35
			15	59	38	30	102	64
24	97	300	12	40	19	27	58	37
			12	40	19	27	66	37
			19	68	40	33	120	69
25	101	325	13	34	19	28	55	37
			13	35	19	28	57	38
			12	68	39	26	127	67
26	105	351	14	46	19	30	64	38
			13	46	19	29	64	38
			15	65	38	28	118	64
27	109	378	13	37	19	29	64	38
			13	35	19	29	57	38
			15	75	37	29	132	65
28	113	406	13	32	18	28	55	37
			12	30	18	27	53	37
			14	63	35	28	108	61
29	117	435	12	39	19	27	61	37
			12	31	19	28	54	38
			14	70	36	28	127	63

Tabela 4.6: Iterações externas e internas para $n = 4$, usando Filtro A, Filtro B e IR.

Problema			Iterações externas			Iterações internas		
p	var.	restr.	min.	max.	média	min.	max.	média
35	176	630	14	81	27	30	102	49
			13	48	26	29	72	49
			66	129	82	104	253	139
36	181	666	15	86	31	34	113	53
			15	78	31	33	102	54
			67	104	81	103	183	136
37	186	703	16	45	27	33	69	49
			14	70	27	30	91	50
			68	126	82	106	237	139
38	191	741	15	95	31	32	124	53
			16	75	29	33	99	52
			69	156	86	108	335	148
39	196	780	17	49	28	35	78	51
			20	49	31	48	111	72
			70	124	84	106	231	142
40	201	820	19	143	36	38	163	58
			13	79	31	29	104	55
			71	139	86	110	284	149
41	206	861	16	158	37	31	176	60
			16	78	32	31	102	57
			70	147	87	107	287	150
42	211	946	16	88	35	34	111	58
			16	98	36	34	126	60
			71	114	86	108	216	146
43	216	946	15	91	31	32	114	53
			15	82	32	32	105	55
			72	123	88	110	229	150
44	221	990	16	55	31	33	82	54
			19	264	36	37	283	60
			74	136	87	114	269	147

Tabela 4.7: Iterações externas e internas para $n = 5$, usando Filtro A, Filtro B e IR.

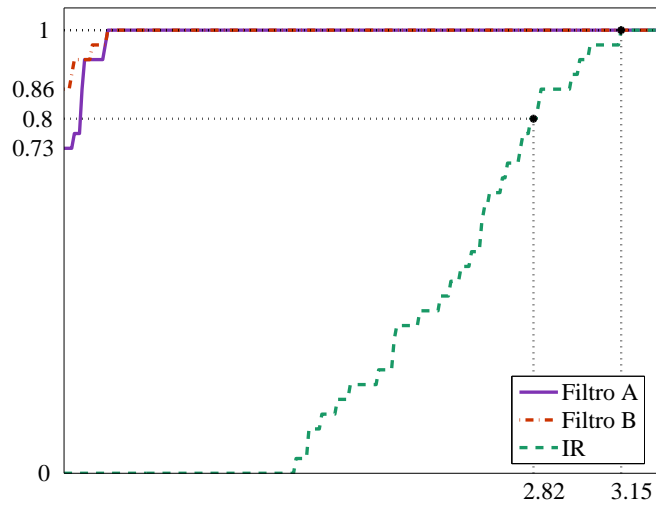


Figura 4.5: Desempenho dos algoritmos Filtro A, Filtro B e IR - iterações externas.

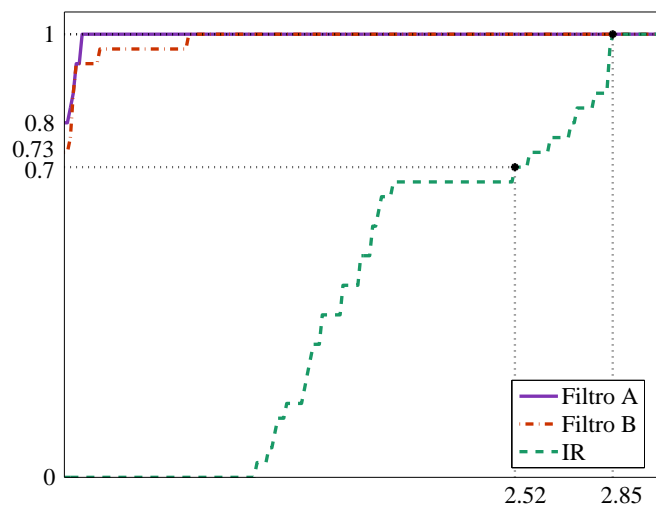


Figura 4.6: Desempenho dos algoritmos Filtro A, Filtro B e IR - iterações internas.

Conclusões

Estudamos ao longo deste trabalho os métodos de filtro para programação não linear. São métodos relativamente recentes, datando de 1997, cujo objetivo é garantir a convergência global de algoritmos, sem fazer uso das funções de mérito, uma ferramenta comum na maioria dos algoritmos para otimização com restrições. Desde sua introdução, por Fletcher e Leyffer [17], surgiram inúmeros artigos, teses, dissertações, etc., apresentando os resultados de pesquisas desenvolvidas no assunto, tanto computacionais quanto teóricas. Esperamos que esta tese possa contribuir com a literatura existente.

Nosso objetivo foi o de apresentar um algoritmo geral de filtro, permitindo uma grande liberdade para o cálculo do passo. A única exigência (além das hipóteses clássicas) é que, perto de um ponto viável não estacionário, o decréscimo da função objetivo seja “grande”, conforme formalizado na Hipótese H4, no Capítulo 2. Com isto provamos a convergência global do algoritmo proposto. Para que o trabalho ficasse completo, também mostramos que a exigência feita sobre o passo é razoável, exibindo dois algoritmos clássicos para a obtenção do passo, que satisfazem tal hipótese.

Apresentamos testes computacionais com o objetivo de comparar os critérios de filtro e função de mérito, que consistem na diferença essencial entre os métodos propostos em [23] e [33]. Escolhemos para tal fim problemas da família dos “Hard-Spheres Problems” que, além de serem interessantes por si só, permitem uma fase de viabilidade simples, barata e exata. Os testes que foram realizados são, naturalmente, insuficientes para fazer afirmações mais gerais. Eles apenas confirmam outros resultados obtidos na avaliação do desempenho dos algoritmos de filtro (por exemplo, [17, 24, 25, 27, 28, 29]).

Perspectivas para outros trabalhos. Apresentamos em seguida algumas questões que abrem caminhos para novas pesquisas.

- **Convergência global para *todo* ponto de acumulação.** O algoritmo de filtro com restauração inexata proposto em [23] tem uma propriedade mais forte sobre convergência, garantindo que *todo* ponto de acumulação da sequência gerada pelo al-

goritmo é estacionário. Para tanto, é feita uma pequena modificação no critério de atualização do filtro, de modo que mais pontos possam ser adicionados no filtro, tornando o algoritmo mais restritivo.

Acreditamos que isto também seja válido para o algoritmo proposto neste trabalho, com os passos calculados por PQS.

• **Convergência local de algoritmos de filtro.** Como vimos no Capítulo 3, os algoritmos de filtro, apesar de serem mais tolerantes que os baseados em função de mérito, também podem sofrer do efeito Maratos, ou seja, podem recusar um passo essencialmente bom, mesmo arbitrariamente próximo de uma solução. Tal fenômeno pode comprometer seriamente a eficiência do algoritmo. Dentre os vários algoritmos de filtro que têm sido propostos, poucos deles abordam convergência local (ou velocidade de convergência). Uma estratégia para evitar o efeito Maratos consiste no uso de correção de segunda ordem, aplicada com sucesso no contexto de busca linear [8, 49]. Atualmente, Gonzaga e Karas estão desenvolvendo pesquisas para estabelecer resultados de convergência local para o algoritmo de filtro proposto em [23], baseado em restauração inexata. Ulbrich [47] também estabelece convergência local superlinear para métodos de filtro, mas sem o uso de correção de segunda ordem. A estratégia está na definição do filtro, onde as entradas são o lagrangeano e uma medida de inviabilidade que envolve as variáveis primais e duais.

No entanto, ainda não temos resultados definitivos, referentes à convergência local, para algoritmos de filtro baseados em PQS com região de confiança. Conforme sugerido por Fletcher e Leyffer [17], o uso de correção de segunda ordem é um caminho possível para evitar o efeito Maratos.

• **Busca linear para obtenção do passo para o Algoritmo 2.1.** As abordagens discutidas aqui utilizam a estratégia de região de confiança para controlar o tamanho do passo. Uma questão que permanece aberta até o presente momento é se os algoritmos baseados em busca linear também se encaixam na nossa teoria geral, ou seja, se eles satisfazem a Hipótese H4.

• **Algoritmo de pontos interiores com filtro.** Nos algoritmos de pontos interiores propostos por Byrd, Gilbert e Nocedal [5] e Byrd, Hribar e Nocedal [7] uma função de mérito é usada para avaliar o passo. Nos trabalhos de Ulbrich, Ulbrich e Vicente [46] e Wächter e Biegler [50] o filtro é utilizado, ao invés da função de mérito, para avaliar o passo. Em [46] o filtro é definido de um modo diferente do que foi proposto originalmente em [17]. As entradas do filtro são o lagrangeano e uma medida de inviabilidade que envolve as variáveis primais e duais. Por outro lado, em [50] a estratégia

de controle do passo é a busca linear.

Uma linha de trabalho neste assunto consiste em utilizar o filtro, do mesmo modo que o sugerido por Fletcher e Leyffer, em algoritmos de pontos interiores com região de confiança.

Anexos

Apresentamos aqui duas subrotinas auxiliares para os algoritmos de filtro discutidos no Capítulo 2. A primeira testa se o ponto (f_0y, hy) é proibido e se caracteriza por não precisar fazer as comparações com todos os pontos do filtro. Isto porque os pontos do filtro são ordenados pelas abscissas e assim basta encontrar j tal que $\tilde{f}_0^j \leq f_0y < \tilde{f}_0^{j+1}$. Para este j a dominação fica determinada pela condição $hy \geq \tilde{h}^j$.

A outra subrotina atualiza o filtro, incluindo o par (\tilde{f}_0, \tilde{h}) . Além disso, os pontos dominados são excluídos e os pontos restantes são ordenados pelas abscissas. A principal característica é obter j tal que $\tilde{f}_0^j < \tilde{f}_0 \leq \tilde{f}_0^{j+1}$. Com isto, a determinação dos pontos dominados fica muito simples, sendo feita com $dom + 1$ comparações do tipo $\tilde{h} \leq \tilde{h}^{j+i+1}$, onde dom representa o número de pontos dominados.

Algoritmo 4.1 *Teste do filtro.*

Dados: $F_k, (\tilde{f}_0, \tilde{h}), nF = comprimento(F_k), (f_0y, hy)$

SE $f_0y \geq \tilde{f}_0$ E $hy \geq \tilde{h}$,

proib = 1

SENÃO

$j = nF$;

ENQUANTO $j > 0$ E $f_0y < \tilde{f}_0^j$,

$j = j - 1$;

FIM

SE $j > 0$ E $hy \geq \tilde{h}^j$,

proib = 1.

Algoritmo 4.2 *Atualização do filtro.*

Dados: $F_k, (\tilde{f}_0, \tilde{h}), nF = \text{comprimento}(F_k)$

$j = nF; \text{dom} = 0;$

ENQUANTO $j > 0$ E $\tilde{f}_0 \leq \tilde{f}_0^j,$

$j = j - 1;$

FIM

ENQUANTO $(j + \text{dom} + 1) \leq nF$ E $\tilde{h} \leq \tilde{h}^{j+\text{dom}+1},$

$\text{dom} = \text{dom} + 1;$

FIM

$nF = nF + 1 - \text{dom};$

SE $nF > j + 1$

SE $\text{dom} = 0$

PARA $i = nF$ ATÉ $j + 2, \text{passo} = -1,$

$\tilde{f}_0^i = \tilde{f}_0^{i-1};$

$\tilde{h}^i = \tilde{h}^{i-1};$

FIM

SE $\text{dom} > 1$

PARA $i = j + 2$ ATÉ $nF,$

$\tilde{f}_0^i = \tilde{f}_0^{i+\text{dom}-1};$

$\tilde{h}^i = \tilde{h}^{i+\text{dom}-1};$

FIM

$\tilde{f}_0^{j+1} = \tilde{f}_0;$

$\tilde{h}^{j+1} = \tilde{h}.$

Referências Bibliográficas

- [1] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty. *Nonlinear Programming Theory and Algorithms*. John Wiley, New York, 2nd edition, 1993.
- [2] D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, Belmont, Massachusetts, 1995.
- [3] E. G. Birgin and J. M. Martínez. Local convergence of an inexact-restoration method and numerical experiments. *Journal of Optimization Theory and Applications*, 127:229–247, 2005.
- [4] J. F. Bonnans, J. C. Gilbert, C. Lemaréchal, and C. A. Sagastizábal. *Numerical Optimization: Theoretical and Practical Aspects*. Springer Verlag, Berlin, 2002.
- [5] R. H. Byrd. Robust trust region methods for constrained optimization. Third SIAM Conference on Optimization, 1987.
- [6] R. H. Byrd, J. C. Gilbert, and J. Nocedal. A trust region method based on interior point techniques for nonlinear programming. *Mathematical Programming*, 89(1):149–185, 2000.
- [7] R. H. Byrd, M. E. Hribar, and J. Nocedal. An interior point algorithm for large-scale nonlinear programming. *SIAM Journal on Optimization*, 9(4):877–900, 1999.
- [8] C. M. Chin. A local convergence theory of a filter line search method for nonlinear programming. Technical report, Numerical Optimization Report, Department of Statistics, University of Oxford, England, January 2003.
- [9] C. M. Chin and R. Fletcher. On the global convergence of an SLP-filter algorithm that takes EQP steps. *Mathematical Programming*, 96(1):161–177, 2003.
- [10] A. R. Conn, N. I. M. Gould, and P. L. Toint. Global convergence of a class of trust region algorithms for optimization with simple bounds. *SIAM Journal*

- on Numerical Analysis*, 25:433–460, 1988. See also *SIAM Journal on Numerical Analysis* 26:764–767, 1989.
- [11] A. R. Conn, N. I. M. Gould, and P. L. Toint. *LANCELOT: a Fortran Package for Large-Scale Nonlinear Optimization (Release A)*. Springer Series in Computational Mathematics 17, Springer Verlag, New York, 1992.
- [12] A. R. Conn, N. I. M. Gould, and P. L. Toint. *Trust-Region Methods*. MPS-SIAM Series on Optimization, SIAM, Philadelphia, 2000.
- [13] J. H. Conway and N. J. C. Sloane. *Sphere Packings, Lattices and Groups*. Springer-Verlag, New York, 1988.
- [14] E. D. Dolan and J. J. Moré. Benchmarking optimization software with performance profiles. *Mathematical Programming*, 91:201–213, 2002.
- [15] A. S. El-Bakry, R. A. Tapia, T. Tsuchya, and Y. Zhang. On the formulation and theory of the Newton interior-point method for nonlinear programming. *Journal of Optimization Theory and Applications*, 89:507–541, 1996.
- [16] R. Fletcher, N. Gould, S. Leyffer, P. Toint, and A. Wächter. Global convergence of trust-region and SQP-filter algorithms for general nonlinear programming. *SIAM J. Optimization*, 13(3):635–659, 2002.
- [17] R. Fletcher and S. Leyffer. Nonlinear programming without a penalty function. *Mathematical Programming - Ser. A*, 91(2):239–269, 2002.
- [18] R. Fletcher, S. Leyffer, and P. L. Toint. On the global convergence of an SLP-filter algorithm. Technical Report NA/183, Dundee University, Dept. of Mathematics, 1998.
- [19] R. Fletcher, S. Leyffer, and P. L. Toint. On the global convergence of a filter-SQP algorithm. *SIAM J. Optimization*, 13(1):44–59, 2002.
- [20] D. M. Gay, M. L. Overton, and M. H. Wright. A primal-dual interior method for nonconvex nonlinear programming. In Yuan Y, editor, *Advances in Nonlinear Programming*, pages 31–56. Kluwer Academic Publishers, Dordrecht, 1998.
- [21] F. A. M. Gomes. A sequential quadratic programming algorithm that combines merit function and filter ideas. Technical report, University of Campinas, IMECC, Department of Applied Mathematics, Brazil, 2004.

- [22] F. A. M. Gomes, M. C. Maciel, and J. M. Martínez. Nonlinear programming algorithms using trust regions and augmented Lagrangians with nonmonotone penalty parameters. *Mathematical Programming*, 84(1):161–200, 1999.
- [23] C. C. Gonzaga, E. W. Karas, and M. Vanti. A globally convergent filter method for nonlinear programming. *SIAM J. Optimization*, 14(3):646–669, 2003.
- [24] N. I. M. Gould, S. Leyffer, and P. L. Toint. A multidimensional filter algorithm for nonlinear equations and nonlinear least-squares. *SIAM Journal on Optimization*, 15(1):17–38, 2005.
- [25] N. I. M. Gould, C. Sainvitu, and P. L. Toint. A filter-trust-region method for unconstrained optimization. *SIAM Journal on Optimization*, 2005. To appear.
- [26] N. I. M. Gould and P. L. Toint. SQP methods for large-scale nonlinear programming. Technical report, CCLRC Rutherford Appleton Laboratory, England, August 1999.
- [27] N. I. M. Gould and P. L. Toint. The filter idea and its application to the nonlinear feasibility problem. Technical report, CCLRC Rutherford Appleton Laboratory, England, September 2003.
- [28] N. I. M. Gould and P. L. Toint. FILTRANE, a fortran 95 filter-trust-region package for solving nonlinear feasibility problems. Technical report, CCLRC Rutherford Appleton Laboratory, England, September 2003.
- [29] E. W. Karas, A. A. Ribeiro, C. Sagastizábal, and M. Solodov. A bundle-filter method for nonsmooth convex constrained optimization. *Mathematical Programming*, submitted, 2005.
- [30] N. Krejić, J. M. Martínez, M. Mello, and E. A. Pilotta. Validation of an augmented lagrangian algorithm with a gauss-newton hessian approximation using a set of hard-spheres problems. *Computational Optimization and Applications*, 16:247–263, 2000.
- [31] N. Maratos. *Exact Penalty Function Algorithms for Finite Dimensional and Control Optimization Problems*. PhD thesis, Imperial College Science Technology, University of London, 1978.

- [32] J. M. Martínez. Inexact-restoration method with Lagrangian tangent decrease and a new merit function for nonlinear programming. *Journal of Optimization Theory and Applications*, 111:39–58, 2001.
- [33] J. M. Martínez and E. A. Pilotta. Inexact restoration algorithm for constrained optimization. *Journal of Optimization Theory and Applications*, 104:135–163, 2000.
- [34] J. M. Martínez and S. A. Santos. Métodos computacionais de otimização. 20.^o Colóquio Brasileiro de Matemática - IMPA, 1995.
- [35] J. M. Martínez and B. F. Svaiter. A practical optimality condition without constraint qualifications for nonlinear programming. *Journal of Optimization Theory and Applications*, 118:117–133, 2003.
- [36] R. B. Murtagh and M. A. Saunders. Minos 5.4 user's guide. Technical Report SOL-83-20, Stanford University, 1995.
- [37] O. R. Musin. The kissing number in four dimensions. Technical report, Institute for Mathematical Study of Complex Systems, Moscow State University, Russia, April 2005.
- [38] P. Nie. Sequential penalty quadratic programming filter methods for nonlinear programming. Technical report, Department of Mathematics, Jinan University, China, 2003.
- [39] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer Series in Operations Research. Springer-Verlag, 1999.
- [40] E. Omojokun. *Trust Region Algorithms for Optimization with Nonlinear Equality and Inequality Constraints*. PhD thesis, Dept. of Computer Science, University of Colorado, 1991.
- [41] E. Polak. *Computational Methods in Optimization: A Unified Approach*. Academic Press, New York, 1971.
- [42] M. J. D. Powell. Convergence properties of algorithms for nonlinear optimization. *SIAM Review*, 28:487–500, 1986.
- [43] E. B. Saff and A.B.J. Kuijlaars. Distributing many points on a sphere. *Mathematical Intelligencer*, 19:5–11, 1997.

- [44] C. Sagastizábal and M. Solodov. An infeasible bundle method for nonsmooth convex constrained optimization without a penalty function or a filter. *SIAM J. Optimization*, 16:146–169, 2005.
- [45] D. F. Shanno and R. J. Vanderbei. Interior-point methods for nonconvex nonlinear programming: Orderings and higher-order methods. *Mathematical Programming*, 87(2):303–316, 2000.
- [46] M. Ulbrich, S. Ulbrich, and L. N. Vicente. A globally convergent primal-dual interior-point filter method for nonlinear programming. *Mathematical Programming, Ser. A*, 100(2):379–410, 2004.
- [47] S. Ulbrich. On the superlinear local convergence of a filter-SQP method. *Mathematical Programming, Ser. B*, 100(1):217–245, 2004.
- [48] R. J. Vanderbei and D. F. Shanno. An interior-point algorithm for nonconvex nonlinear programming. *Computational Optimization and Applications*, 13:231–252, 1999.
- [49] A. Wächter and L. T. Biegler. Line search filter methods for nonlinear programming: Local convergence. *SIAM Journal on Optimization*, 16(1):32–48, 2005.
- [50] A. Wächter and L. T. Biegler. Line search filter methods for nonlinear programming: Motivation and global convergence. *SIAM Journal on Optimization*, 16(1):1–31, 2005.